POLITECNICO DI TORINO

Master's Degree in Computer Engineering
Curriculum Software

Master's Degree Thesis

Development of a Web Application for Risk Management

Supervisor

Prof. Maurizio MORISIO

Company Tutor

Angelo NESTOLA

Candidate

Marta CORCIONE

October 2023

Abstract

With the increase in complexity of modern business operations, effective risk management is one of the key factors in ensuring the success of organizations. The integration of software applications has revolutionized the way companies identify, assess, and mitigate risks. This study deals with the design, development, and deployment of a comprehensive Risk Management Tool.

The Web Application was implemented in Orbyta Tech with the objective of providing a better tool with respect to the ones already on the market. Current solutions often focus on specific risk domains and historical data, making it difficult to adapt to different organizational structures and predict risks in real-time. Our software is based on real-time data and Machine Learning to identify risks and trends. The tool offers customization and an intuitive interface allowing a pleasant User Experience. Our Risk Management Tool is composed of different common sections and modules. This was possible thanks to the Software as a Service (SaaS) model, which is typically based on a multi-tenant architecture.

The Customer has the possibility to select one or more modules by starting a subscription. The Dashboard section displays statistics, allowing users to filter by date. The Project Management section is composed of the elements necessary for registering a Project in the application. In the Risk Management section, users can create, modify, and delete risks, causes, impacts, risk categories, Key Risk Indicators, Risk Breakdown Matrix entries, and mitigations. These features offer comprehensive control over risk-related data, including searching, filtering, and Excel exporting capabilities. The Loss Events History section contains the Loss Events table for managing the company's previous damaging events. The additional modules that were implemented as a start are Data Loss Prevention and Asset Loss and Logistics Risk Management. The former's objective is to support the user in managing and mitigating Cyber-related risks, especially the ones related to data losses and data breaches. The latter, on the other hand, has the goal of managing risks that may impact the transportation (or storage) of goods by using risk assessment founded both on external and internal factors. It also gives the possibility to manage warehouses and shipments and see when a transit crosses a "Dangerous Zone".

The users of this application are Risk and Project Managers, but also IT and Logistics Unit operators who have different roles and authorizations.

The Agile methodology guides the development process, ensuring flexibility and quality.

The C# back-end with Entity Framework Core as the database provider accommodates business logic with a code-first approach, while the Angular front-end offers an intuitive interface. The ABP.io framework provides a robust layered architecture and maintains adherence to Domain Driven Design best practices.

Machine Learning was incorporated into the application using ML.NET. Forecast Algorithms were used to show trends of risk events based on past and present data. Classifier Algorithms were used to categorize events into predefined classes and perform predictions on newly inserted user data. Part of the study focused on researching useful and accurate data sources.

Lastly, the application was tested by adopting an Agile testing approach and deployed with Azure DevOps.

In conclusion, this study's aim was to emphasize the importance of effective risk management software and provide a powerful yet simple solution.

Table of Contents

List of Tables

List of Figures

Acronyms

DDD

Domain Driven Design

ML

Machine Learning

SaaS

Software as a Service

ERM

Enterprise Risk Management

KRIs

Key Risk Indicators

RBM

Risk Breakdown Matrix

WBS

Work Breakdown Structure

RBS

Risk Breakdown Structure

ISV

Independent Software Vendor

SLA

Service Level Agreement

DoD

Definition of Done

PoC

Proof of Concept

UAT

User Acceptance Testing

CI/CD

Continuous Integration and Continuous Delivery

CI

Continuous Integration

CD

Continuous Delivery

UI

User Interface

UX

User Experience

OOP

Object Oriented Programming

ER

Entity-Relationship

GUIs

Graphical User Interfaces

DTOs

Data Transfer Objects

ORM

Object-Relational Mapping

DOM

Document Object Model

MAE

Mean Absolute Error

RMSE

Root Mean Squared Error

e2e

End-to-End

Chapter 1

Introduction

With the increase in complexity of modern business operations, effective risk management is one of the key factors in ensuring the success of organizations. The integration of software applications has revolutionized the way companies identify, assess, and mitigate risks. This study deals with the design, development, and deployment of a comprehensive Risk Management Tool software.
The Web Application was implemented in Orbyta Tech with the objective of providing a better tool with respect to the ones already on the market. Current solutions often focus on specific risk domains and historical data, making it difficult to adapt to different organizational structures and predict risks. Our objective was to exploit real-time data to identify risks and trends in a customizable way and with an intuitive interface for a pleasant User Experience.

One of the main goals of this project, was to create highly modular software, by separating different risk categories and contexts. This was possible thanks to the Software as a Service (SaaS) model, which is typically based on a multi-tenant architecture where a single instance of the application, with different and customized versions based on customer subscriptions, is shared to all customers, keeping their personal data and settings isolated. Our Risk Management Tool is composed of different sections and modules. The sections are common to every customer and subscription. The customer has the possibility to use one or more modules by starting a subscription for each module separately or for a number of them. Common sections are:

- **Dashboard**: The Dashboard contains charts and insights on user-inserted data. It's divided into three tabs:
  - **Risk Dashboard**: The information shown here is related to risks inserted in the system and their assessment. This tab includes the Impact-Likelihood matrix, the foundation of modern Risk Management;

  - **Project Dashboard**: Here the user can see charts related to projects that are/will be assessed;
  - **Loss Events Dashboard**: This tab contains information about the Loss Events History of the company.

- **Project Management**: The Project Management section is composed of the elements necessary for registering a Project in the application. These elements are Project, Sections (sections are parts of a project), and Work Items (composing a section). After adding or editing a Project with its hierarchy, a Gantt diagram is generated. Here it's also possible to add a new Mitigation Plan: a project with the aim of mitigating a risk.

- **Risk Management**: This section is the main part of the application. Here users can actually manage and assess risks using Impact, Likelihood, and Velocity values. It includes the Risk Register where it's possible to connect a risk to other attributes previously added in other subsections such as Causes, Impacts, and Risk Categories (Reputational, Financial, Operational...), and LossTypes associated with the risk. Another functionality consists in managing KRIs and showing them on a Speedometer. Here it's also possible to add new Mitigations (the basis for Mitigation plans) and perform the risk assessment of specific Work Items through the RBM Entries.

- **Loss Events History**: This section contains the Loss Events table for managing the company's previous damaging events and a Loss Type subsection for categorizing Events on their specific type (e.g Hacking or Theft).

The additional modules that were implemented as a start are:

- **Data Loss Prevention**: Data Loss Prevention's objective is to protect sensitive information from being lost, stolen, or exposed to unauthorized parties. With the increasing amount of data being generated and shared across various platforms, it has become more critical than ever to safeguard valuable information. This module has the goal of supporting the user in managing and mitigating Cyber-related risks, especially the ones related to data losses and data breaches. For the moment, it is formed of two subsections: Cyber Security Statistics and Fields. The Statistics one provides an up-to-date analysis of the current security situation and supports the user in decision-making. The Fields subsection allows the user to manage different organization Fields in which it's possible to have cyber attacks, in order to have a better idea of where to apply safety measures.

- **Asset Loss & Logistics Risk Management**: This section has the goal of identifying, assessing, and controlling risks that may impact the transportation (or storage) of goods and allows the user to view statistics and manage Logistics expeditions according to the risk assessment. It is composed of different subsections such as Asset Statistics, showing the current risk assessment based on the data analysis performed in section 4.6.2; Transits and Stops to manage shipments and see when a transit crosses a "Dangerous Zone"; Warehouses and Countries to manage the risks related to items which are not in transit; Products, to adapt security plans based also on the value of transported goods.

For each entity in the system, an intuitive and coherent browsing experience allows searching, filtering, and Excel exporting, giving the user the possibility to easily collect data for producing formal Risk Reports.

The users of this application are Risk and Project Managers, but also IT and Logistics Unit operators who have different roles and authorizations.

By exploiting data-driven insights, the software can provide more accurate predictions and assist the Risk Manager in making informed decisions. Part of the study focused on researching useful and accurate data sources and performing historical data analysis. However, it's essential to note that this is just one part of the complete picture. After the software implementation, historical data will be seamlessly integrated with real-time data, enabling more accurate predictions and real-time informed decisions. Currently, data updates are based on API sources and data streams. By exploiting data-driven insights, the software can provide more accurate predictions and assist the Risk Manager in making informed decisions.Machine Learning models were developed, trained, and integrated into the software. These models will continuously learn from new data, adapting through changes. Forecast Algorithms were used to show trends of risk events based on past and present data. Also, Classifier Algorithms were used to categorize events into predefined classes and perform predictions on newly inserted user data. Part of the study focused on researching useful and accurate data sources.

The main technology used for the development was the ABP.io framework, which provides a robust layered architecture and maintains adherence to Domain Driven Design best practices. The C# back-end, with Entity Framework Core as the database provider, accommodates business logic with a code-first approach, while the Angular front-end offers an intuitive interface. Machine Learning was incorporated into the application using ML.NET, a powerful framework for developing, training, and testing ML Models in C# without particular Machine Learning knowledge needed. The User Interface was enriched with many charts using the Chart.js library and maps for logistics were implemented using Leaflet. Mermaid APIs were used to generate Gantt diagrams starting from project-related information inserted by the user in forms. The information was translated into Markdown Language by a simple algorithm and passed to the APIs. In this way, the user can construct Gantt diagrams without having to perform additional actions.

The Agile methodology guided the development process, emphasizing flexibility and adaptability throughout the software development life-cycle, and the development process was significantly enhanced through the implementation of Continuous Integration/Continuous Development DevOps pipelines. Lastly, the application was tested by adopting an Agile testing approach and deployed with Azure.

The remainder of this thesis is structured as follows: Chapter 2 describes the company in which the thesis work has taken place, Orbyta Tech, to provide the context necessary to understand the needs that led to the creation of the tool. Chapter 3 provides a literature review of existing risk management methodologies and software solutions. Chapter 4 details the methodology and design principles that guided the development process. Chapter 5 focuses on the implementation details, including the technologies used and the architectural choices made. Chapter 6 discusses the evaluation process and shows the results obtained from testing the software in real-world scenarios by experts in the field. Finally, Chapter 7 draws conclusions, and outlines potential areas for future research and development.

In conclusion, this study's aim was to emphasize the importance of effective risk management software and provide a powerful yet simple solution.

Chapter 2

Orbyta Tech

2.1 Provider profile

2.1.1 Type of provider (Vendor/System integrator) and general presentation

Orbyta Tech is the technology company of the Orbyta group, which is made up of seven companies. Thanks to know-how and individual verticalizations, Orbyta is able to provide its customers with support and advice at 360°, thus covering all areas of corporate interest.

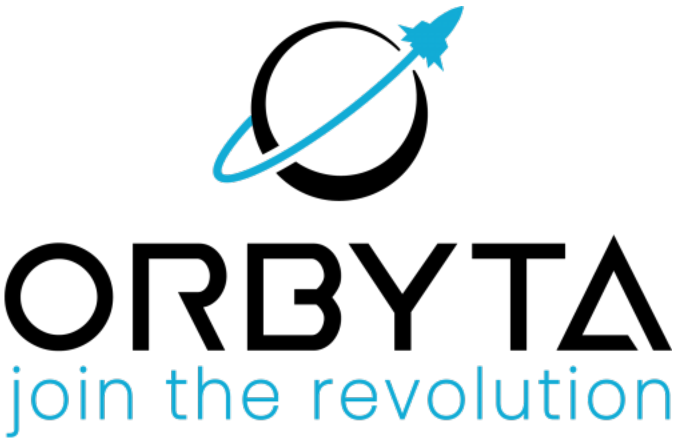


Figure 2.1: The Orbyta Group

Two macro-brands encompass the group's offer based on target customers and sector of action of individual companies:

- **Orbyta Technologies**: specialized in IT consultancy in the application and systemic field. Realizes highly complex projects with the most modern technologies and exploiting the most innovative methodologies. Deals with design, implementation, delivery, integration, application maintenance of software, hardware and IOT systems. The following companies are part of the technologies area:

  - **Orbyta Tech**: Specialized in software development and assistance of systems. It's capable of creating integrated solutions and designing information infrastructures offering consultancy and turnkey projects.
  - **Orbyta Infogest**: deals with designing, supplying and reselling HW, installing and assisting PCs, servers, storage, internetworking in heterogeneous operational environments.

- **Orbyta Business Partner**: The BP area provides essential services for businesses. They support customers in compliance and engineering design, in accounting, administrative, fiscal and financial matters, payroll processing and provide HR management services in outsourcing, up to providing extrajudicial and judicial legal assistance.

The strong synergy between the companies of the group allows the company to be seen as a reliable, prepared and complete partner. The following companies are part of the business partner area:

- **Orbyta Engineering**: Offers skilled performance in designing and in managing constructions in the engineering field and identifies the best customized solutions, in concordance with legal requirements and with the maximum simplification of corporate compliance.

- **Orbyta Tax&Finance**: Specialized in tax and corporate consultancy. It is able to provide assistance to entrepreneurial activities in matters concerning managerial management, accounting and all civil and tax obligations.

- **Orbyta People**: Offers advice on labor matters, administration and human resource management, payroll processing, time management benefits, welfare and union relations.

- **Orbyta Legal**: Offers assistance and judicial and extrajudicial legal advice, even continuous, with particular regard to management and business development.

- **Orbyta Strategy**: It is the internal company of the group which through its staff contributes to the constant improvement of business processes and provides integrated services for organization and growth, setting the logic and managing the Group dynamics.

2.1.2 Key Provider Information

Orbyta is a constantly growing group, with a 2022 turnover of 15 million euros. The group has about 250 employees and is located in various locations: Turin, Milan, Rome, Lecce. These offices and the presence of consultants located in other areas allow the company to cover geographically the whole Italian territory. The process of analyzing and evaluating investments and acquisitions in foreign offices, particularly in Germany, it's currently happening.

2.2 Provider organization overview

At Orbyta Technologies people experiment with the avant-garde and create innovation. Partners are guided in the conception, design and development of interactive and immersive technological processes. The company is characterized by a highly specialized multidisciplinary expertise in the design, development and implementation of complex information systems and innovative digital solutions. Moreover, it has a dedicated team with the goal to support partners and businesses. In Orbyta experience becomes innovation. The offer is divided into:

- **Digital Transformation**: guide and support partners in a path of custom development and digital transformation by locating technological solutions and IT architectures consistent with growth objectives. Manage every aspect of the path and coordinate every single activity by constantly monitoring performance, thanks to important transversal skills and the ability of the Consulting team to go beyond the schemes, combined with an overview of business processes.

- **Software Development**: develop tailor made technological solutions implementing a wide range of IT products and projects in multiple areas of intervention, with carefully composed teams with specialist skills ranging from project management to the most up-to-date ICT training. The aim is to become a reference point for the IT architecture of each partner thanks to the planning and management capacity of information systems and subsystems of the team in our software house, Area 51.

- **Design & Strategy**: Orbyta Tech's creative team is XLAB, ready to accompany digital growth and develop innovative ideas and omnichannel strategies with maximum impact. Generate connections with the right mix of User Experience, digital interface design, Creative Communication and digital marketing. Work on the boundaries between business, technology and design in all its forms (Thinking, Human centered, System, Service, Futures, User Experience, User Interface...)

The group is composed of pixel perfect and enthusiastic futurists, collaborating with partners at all stages of the project from analysis to design up to prototyping and testing phase. The client is a team member, an irreplaceable project partner in the co-creation of the best digital product. The company's skills are: User research, UI/UX Design, Brand Design, Brand Strategy, 3D Design, Creative & Integrated Communication, Web Experience Development and Metaverse Creation. The company's approach:

- **Collaboration**: smart working, hybrid mode, in presence: being efficient in every situation, using the best collaboration tools like Trello, FigJam, InVision, Zeplin and many more.
- **Design**: using the best design and development tools every day such as Figma, Sketch, Adobe, Blender, Webflow.
- **Innovation**: novelties and experiments, oriented towards creating new 3D environments in the Metaverse to devise new business models.
- **Infrastructure Networking**: providing networking and enhancement services to the security status of the corporate network. Identifying the needs connecting elements, making the project operational. Working from both remote and on-site, thanks to the consolidated experience of the Base2 team and constantly looking for the most innovative technologies that guarantee a systemic consultancy support and management at large centers data processing in the banking, insurance and industrial sectors.
- **Hardware Reselling**: creating customized hardware infrastructures implemented from a continuous path of high added value consultancy that becomes promoter of change and development in terms of organization and management of business flows. Acquiring the need, building the structure, leading in integration with operations. Building safe and reliable solutions evaluating from time to time the best technologies available to the Infogest team which makes use of consolidated partnerships with the main players of reference on the market such as HP, Microsoft, Fortinet, Vmware, Veeam, Arcserve.

The current organization of Orbyta Tech in 4 Units and Dedicated Teams for customers and projects with similar technology stacks, makes possible the parallel and coordinated development of initiatives:

- **Intelligent Platform**: Design of complex and resilient Cloud Native architectures, Data Analytics, ML and AI.
- **Process Automation**: Design and development of software modules on the Microsoft DotNet stack, Java, Node, Javascript and Python.

- **Digital & App Innovation**: Design and development of web client, desktop and mobile applications, with different targets and development stacks, such as Angular, React, Vue, Flutter, ReactNative, Swift, Kotlin.
- **Business Consulting**: Governance and management of complex projects, with the application of the best methodologies and development of automated test phases.

2.3 Business Provider Strategy and Profile

Orbyta Tech operates in the area as a System Integrator and offers consultancy to large Corporate client companies from various fields, including:

- **Banking & Insurance**: Design products for every branch of business, from digital payment services to fraud control, web security and encryption services, from a template predictive decision-making on financing to an operations asset management software, up to the creation of an application for managing the migration of a complex set of data.
- **Automotive & Industrial**: Work in synergy with partners, international companies of recognized fame, for the development of: high-speed data streaming and display mechanisms towards remote customers; a complete modeling of the life cycle of software with complex functions of predictive maintenance, intrusion detection, mitigation and firmware over the air update; platforms for the management of complete technical documentation of products with data profiling and automation capabilities for use by teams; application for the cross-management of stock availability and supplies purchase in relation to production times.
- **Transportation**: Carry out innovative technological projects that contribute to the relevant need of the transport and logistics sector to carry on a process of digitization of systems to promote increasingly integrated mobility; to return punctual and in real time information, to maintain the attractiveness for users of the services.
- **Manufacturing**: Structure solutions capable of integrating, harmonizing and aggregating data from multiple sources with the aim of extracting value and optimizing workflows. It's about projects of high strategic value that facilitate monitoring, verification and control and provide important forward-looking data.
- **Textile & Fashion**: Design digital solutions of great strategic impact that intervene in all phases of the production processes. Technology becomes an

essential resource for being competitive in a sector strongly permeated by craftsmanship and element crucial to consolidate the presence on the market and satisfy, if not even anticipate customer needs.

- **Gaming**: Conceive and develop proposals that are characterized as augmented and virtual experiences, totally immersive, also through the creation and use of avatars. Design solutions that through gaming elements are oriented to improve the company performance through user engagement strategies aimed at multiple goals.

Orbyta Tech mainly deals with: Technical consulting, Business analysis, Research and development, Software development and operations, Process management and support, Digital transformation, Data analysis, Cloud, lean processes & new digital core, IOT and connected services. Orbyta Tech is historically Gold Partner of Microsoft, cultivates further expertise in the public Cloud area also with Amazon Web Services (AWS) and Google cloud. As part of the management of multi-cloud native cloud platforms, the simplification of IT operations and the improving of software product efficiency, Orbyta Tech is a partner of the Mia Platform company.

Chapter 3

Risk Management Approaches and Tools

3.1 Risk Management Overview

In modern businesses, risk management has emerged as a critical practice for organizations wanting to ensure project success. This chapter provides an overview of risk management, its meaning, key components, and the steps involved. But what constitutes a risk? According to the Stanford University, a risk is «The possibility that the occurrence of an event will adversely affect the achievement of the organization's objectives» [1]. These can range from external factors like market changes to internal factors such as employee management. The concept of risk has been synthesized through the following mathematical formula:

$$R = p \times I \tag{3.1}$$

where:

- p is the probability that a specific unfavorable event for the project's development occurs.
- I is the impact that a specific risk can have within a project. Similarly, impact can be considered as the magnitude that an event triggers and the resulting damage that can occur throughout the entire project. [2]

Risk management, on the other hand, is the process of identifying, evaluating, and prioritizing potential risks that could negatively impact an organization or an individual, and then developing and implementing strategies to minimize or mitigate those risks. In simple terms, it's like planning for the worst case scenarios and taking steps to prevent or lessen the harm that could result from them. The goal of risk management is to ensure that an organization or an individual can still achieve their goals and objectives even if unexpected events or challenges occur. There are many approaches to risk management and several frameworks and standards have been published. The most important standards are the ISO 31000 [3] and COSO's Enterprise Risk Management (ERM) [4].

3.1.1 The ISO 31000 standard

Released in 2009 and revised in 2018, the ISO standard includes a list of ERM principles. It is a framework to help organizations apply risk management mechanisms to operations, and a process for identifying, evaluating, prioritizing and mitigating risk. The more recent 2018 standard includes a more strategic approach than the original. It highlights the important role of senior management in risk management and the integration of risk management throughout the whole organization [5].

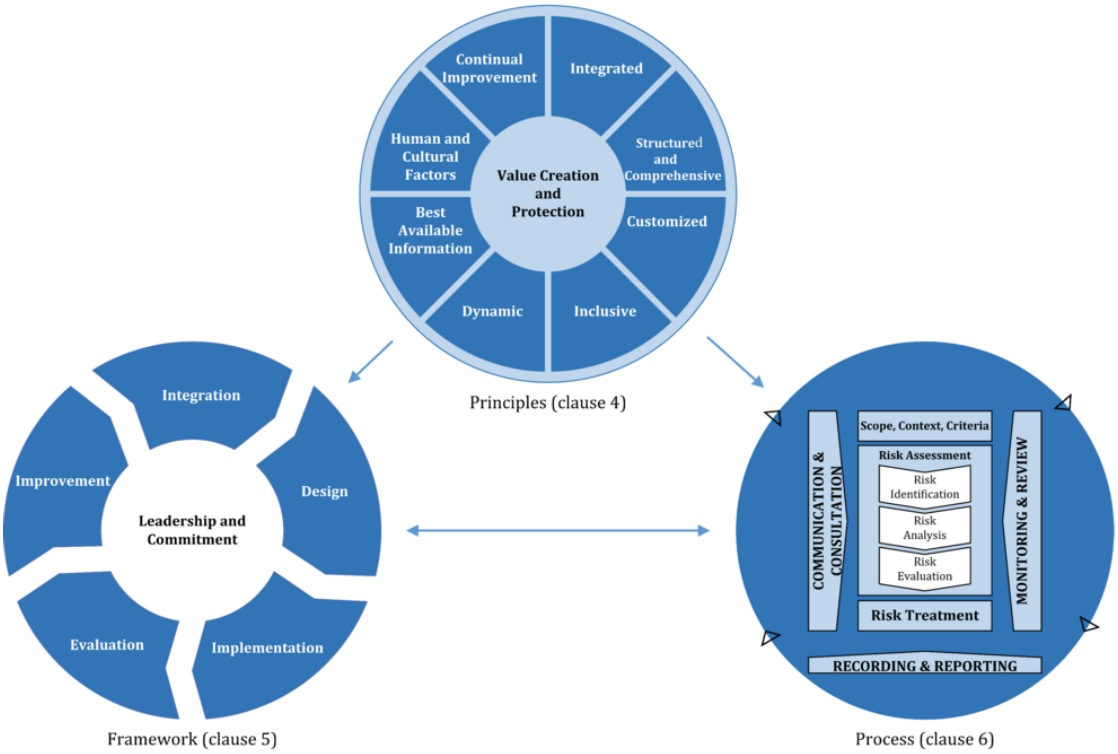


Figure 3.1: Principles, framework and process [3]

3.1.2 COSO ERM Framework

Launched in 2004 and updated in 2017, the COSO framework's goal is to address the increasing complexity of ERM. It defines key concepts and principles of ERM and provides clear direction for managing risk. Developed with input from COSO's five-member organizations and external advisors, the framework is a set of 20 principles divided into five components:

- governance and culture
- strategy and objective-setting
- performance
- review and revision
- information, communication and reporting

According to the framework, organizing risks by categories can also be helpful. The COSO defines the following categories:

- strategic risk (e.g., reputation, customer relations, technical innovations);
- financial and reporting risk (e.g., market, tax, credit);
- compliance and governance risk (e.g., ethics, regulatory, international trade, privacy); and
- operational risk (e.g., IT security and privacy, supply chain, labor issues, natural disasters)[5].

3.1.3 COSO's ERM framework vs ISO 31000 standard

While both COSO and ISO 31000 guide risk management, they differ in their approach and focus. COSO is more focused on internal controls and financial reporting, while ISO 31000 is more comprehensive and emphasizes the importance of a risk management process that is integrated into an organization's overall management system. Ultimately, the choice between these frameworks will depend on an organization's specific needs and requirements.

3.1.4 The 5 components of risk management

There are several ways to structure a risk management process, but all of them should at least include the following foundation steps.

1. **Risk Identification**: Risk identification is the first step in managing risk. Potential risks the business might face are documented and then categorized. Identifying risks is fundamental to reducing the likelihood of missing a risk source, which should be dejected to reduce a risk. Not only current risks should be considered, but also risks that might happen in the future. This allows the company to work on present causes.

2. **Risk Analysis**: The next steps consist of evaluating the two main risk factors: likelihood and impact. The likelihood is the probability that a risk might happen, while the impact is the effect of it, also in monetary terms. If the latter is not possible, different zones can be identified and assigned to factors (e.g. "low", "medium","high"). The combination of impact and likelihood values determines mitigation or acceptance. For example, a risk with a high impact cost but with a very low likelihood could be deprioritized.

3. **Response Planning**: Response planning consists of deciding, first of all, if there's the intention of mitigating the risk, according to previous analysis values. Next, how to act to reduce it.

4. **Risk Mitigation**: The previous planning, will now constitute a Mitigation Plan, a project to mitigate the risk. The organization can determine if it's better to act directly on the likelihood value or reduce the impact. Mitigation plans should be practical and they should aim at reducing the risk score in the best way possible.

5. **Risk Monitoring**: The potential impact and probability of occurrence can change over time, a risk that now does not constitute a threat to the company, might damage revenues in the future. Risk monitoring is the process of constantly assessing the risk over time.

It's important to consider these steps as a process and not as independent operations. Every stage is crucial in assessing and mitigating the risk, and underestimating one could lead to the failure of the whole plan [6].

3.1.5 Challenges and Limitations of Risk Management

Obviously, risk management is not an exact science. It's mainly based on predictions and assumptions that might not always hold true. Predicting future risks accurately is difficult due to inherent uncertainties and organizations face intricate risks that can be hard to fully manage and extremely rare and unexpected events are challenging to predict and prepare for. Effective risk management relies on accurate and relevant data, which might be lacking. Data science is becoming more and more important, which could lead to an improvement in risk management too. Cognitive Biases can lead to underestimating or overlooking certain risks, that's why some companies prefer to outsource risk management and personal judgments can lead to inconsistent risk assessments. Adjourning employees on new risk management practices and standards is crucial. Organizations need to recognize these challenges and limitations while implementing risk management strategies. Mitigating these challenges requires a combination of proper training, data-driven approaches, adaptable frameworks, and a commitment to fostering a risk-aware culture.

3.2 Risk Management Elements

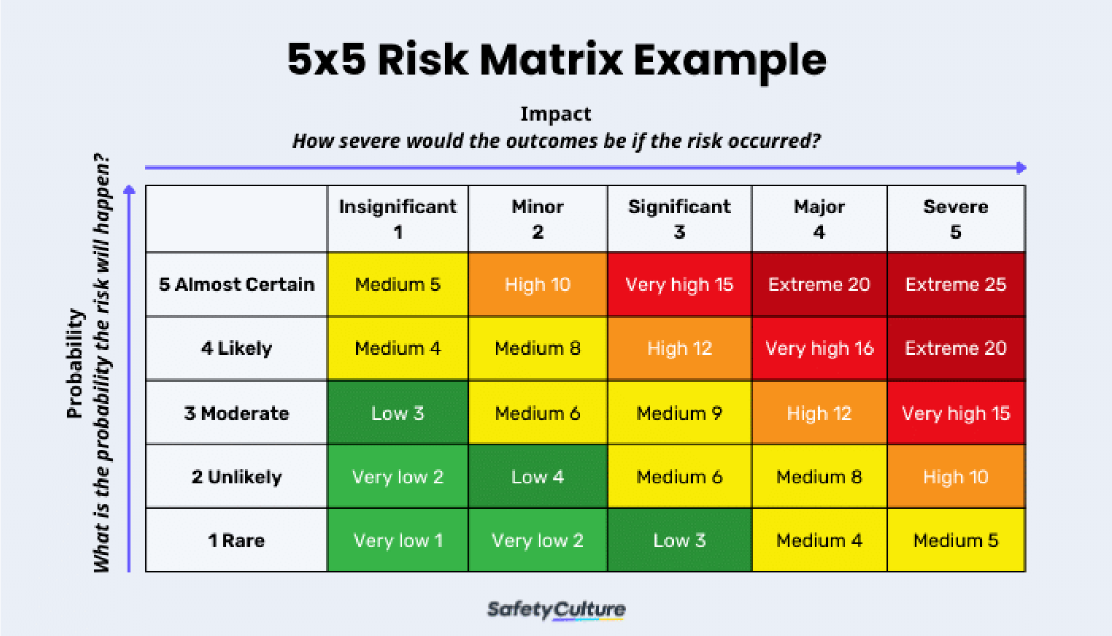
Effective risk management involves a systematic approach to identifying, assessing, mitigating, and monitoring potential risks that could impact a company's objectives, projects, or operations. Several key elements are crucial in establishing a robust risk management plan. Organizations that effectively integrate these elements into their risk management processes are more likely to succeed in reducing the risk's probability and consequences.

3.2.1 The Risk Register

The Risk Register is the collection that captures all identified risks. It contains detailed information about risks, such as description, causes, impacts, impact, likelihood, and final score. If a mitigation is in progress, also the status and the assigned owner can be specified. The Risk Register provides a comprehensive overview of the risks faced by the company, and it displays the updated assessment. An example is shown in table 3.1 which has been simplified for readability.

3.2.2 Impact and Likelihood: The Risk Matrix

The Risk Matrix is a structured graphical representation that catalogs risks based on their impact and likelihood. By categorizing risks into different levels of severity as shown in 3.2 identified using colors and risk levels, the Risk Matrix helps prioritize risks for mitigation. This visual tool enables stakeholders to quickly understand the importance of different risks to speed up the prioritization process.



5x5 Risk Matrix Example

Impact

How severe would the outcomes be if the risk occurred?

Probability

What is the probability the risk will happen?

	Insignificant 1	Minor 2	Significant 3	Major 4	Severe 5
5 Almost Certain	Medium 5	High 10	Very high 15	Extreme 20	Extreme 25
4 Likely	Medium 4	Medium 8	High 12	Very high 16	Extreme 20
3 Moderate	Low 3	Medium 6	Medium 9	High 12	Very high 15
2 Unlikely	Very low 2	Low 4	Medium 6	Medium 8	High 10
1 Rare	Very low 1	Very low 2	Low 3	Medium 4	Medium 5

SafetyCulture

Figure 3.2: The Risk Matrix
[8]

3.2.3 Metrics: Key Risk Indicators

Key Risk Indicators (KRIs) are metrics used to quantify risk exposure in operational risk analysis. They use numerical values and serve as early warning signals, alerting organizations to deviations from expected risk levels.

3.2.4 The Risk Breakdown Matrix

A Risk Breakdown Matrix (RBM) it's a tool for quantifying and categorizing risks. It is built starting from a Risk Breakdown Structure on the x-axis and a Work Breakdown Structure on the y-axis. In essence, the RBM helps to identify, track, and manage risks more efficiently and effectively.

ID	Risk Statement	Causes	Impacts	Likelihood	Impact	Score
1	IT governance and priorities not aligned with institutional priorities	IT failure to understand institutional strategy;...	Poor governance of enterprise IT;...	1	3	3
2	Failure to designate leadership (e.g., an individual or individuals) for institutional oversight and strategic direction for IT operations	Lack of institutional support for IT operations	Poor governance of enterprise IT;...	4	4	16
3	Failure to designate leadership (e.g., an individual or individuals) for institutional oversight and strategic direction for information security activities	Lack of institutional support for IT and information security operations	Poor governance of enterprise information security efforts;...	2	4	8
4	No succession plan for key institutional IT leaders (e.g., CIO, CISO, CTO, CPO, etc.)	Lack of institutional support for IT operations; human nature not to plan for succession activities;...	Leadership void in the event of separation of a key IT leader from the institution	2	5	10
5	Relevant stakeholders not included in important IT investment decisions (e.g., priorities, technologies, new applications)	Lack of senior management support;...	Uses of university IT systems that contravene good investment decision making;...	5	3	15

Table 3.1: Risk Register example.
[7]

- A **Risk Breakdown Structure (RBS)** is a tool used in risk management to divide risks into hierarchical categories. This hierarchical structure helps to identify the main causes of risks, allowing them to be managed more effectively. The RBS can also be used to define risk mitigation activities and to assign specific responsibilities for risk management. In essence, it is a structure that helps to identify, track, and manage risks more efficiently and effectively.

- A **Work Breakdown Structure (WBS)** is a hierarchical structure that breaks down the work required to complete a project into smaller, manageable tasks and activities. The WBS helps to organize and define the work required to complete the project and to assign specific responsibilities for each activity. In summary, the WBS helps to better manage the project, monitor its progress, and ensure that all activities are completed within the expected time frame.

The value of each entry in the matrix is given by the formula 4.1. By adding up the values in the columns and sorting the resulting values, it will be possible to identify the "worst" project risks.

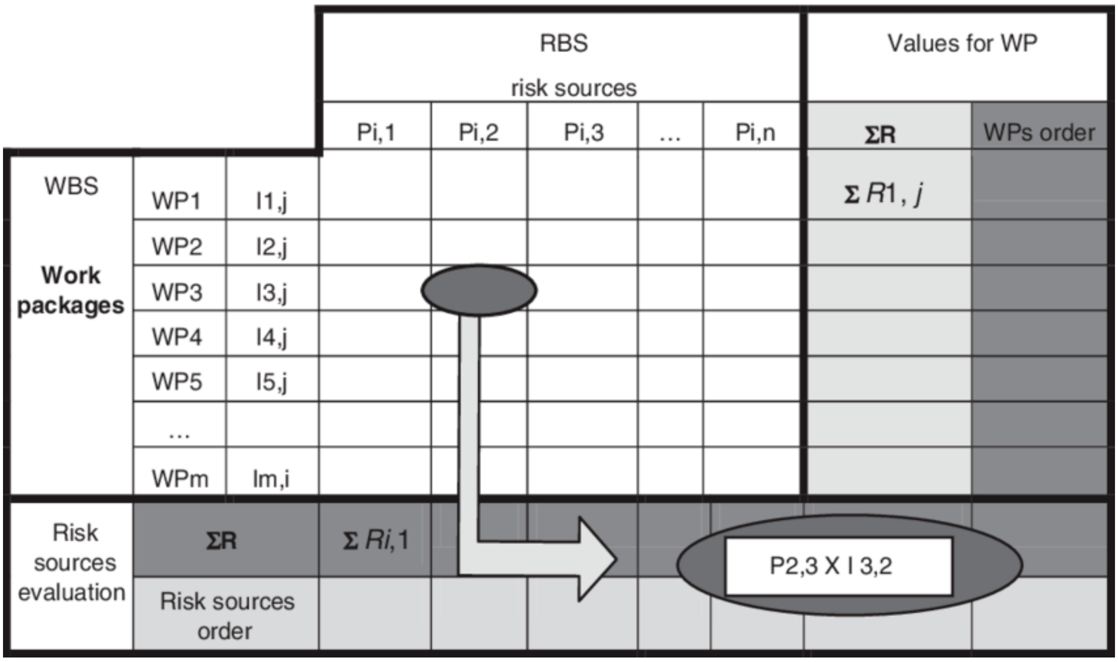


Figure 3.3: The Risk Breakdown Matrix (RBM)
[9]

3.2.5 Loss Events History

A Loss Events History records past risk events that have affected the company's projects. Analyzing historical data helps organizations identify patterns, assess the success of mitigation strategies, and adjust their risk management approach accordingly.

3.3 Risk Management Software

Risk management software is an essential tool for organizations that want to manage their risks effectively and efficiently. It provides a structured approach to risk management that helps organizations centralize their risk management activities, improve their visibility into their risk profile, and comply with regulatory requirements and industry standards.

3.4 Competitor Analysis

In the realm of risk management tools, it's essential to understand the competitive context. We've analyzed several key competitors, including Archer, Resolver, Project Risk Manager, and nTask, to gain insights into their strengths and weaknesses. However, our focus will be on Archer for the reasons described below.

3.4.1 Why Archer?

Archer [10] is an integrated risk management software developed by RSA Security LLC. that helps assess, monitor, and address risks. It also enables the users to capture an inventory of Loss Events and near misses and perform a loss event root cause analysis to understand why the loss occurred and take action to reduce the likelihood and impact of similar losses in the future. Archer helps establish and monitor metrics related to each business unit in the organization, and associates matrices with risks, controls, strategies, objectives, products, services, and business processes to monitor risk, quality assurance, and performance. The platform is based on collaboration, visibility, and flexibility and uses risk analytics, machine learning, and quantification tools.

Our choice to focus on Archer is driven by several key factors. Archer's extensive product portfolio and flexible platform align seamlessly with various organizations' risk management processes. As shown in 3.4 it is composed of different modules, which can be purchased according to current needs. The Enterprise and Operational Risk Management module establishes the basics for security, resiliency, regulatory, compliance, audit, and third-party governance. It has unmatched customization options, extensive regulatory compliance mapping, and deep integration capabilities.

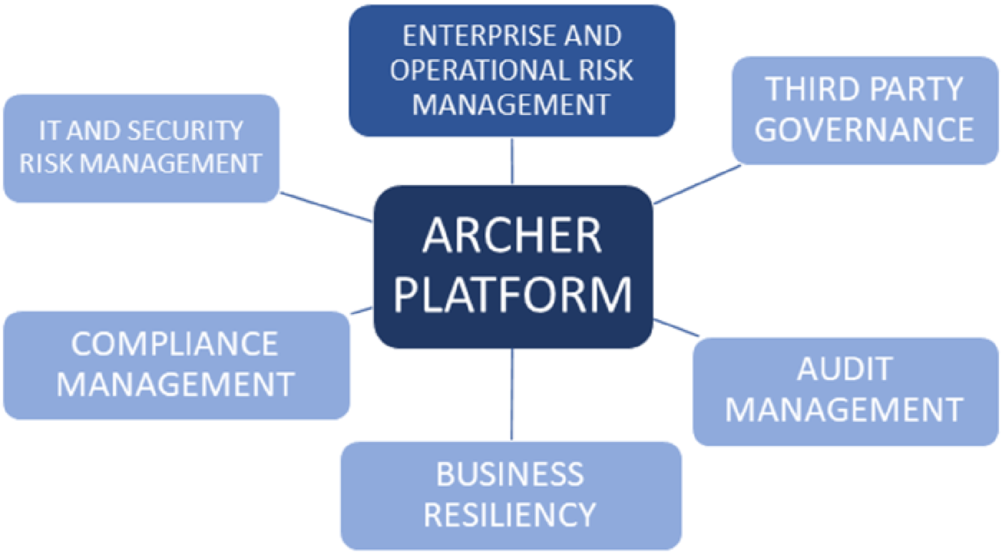


Figure 3.4: The Archer Platform
[10]

While competitors bring their own strengths to the table, Archer's combination of market presence, innovation, and flexibility positions them as the most suitable competitor for our organization's risk management tool. Our decision to focus on Archer reflects our commitment to achieving the highest standards of risk management and organizational resilience.

3.4.2 Key factors driving the need for a new tool

- **Holistic Risk Integration**: Current solutions often focus on specific risk domains, resulting in fragmented risk assessments. Our tool aims at providing a comprehensive framework that integrates risks across departments and processes.

- **Advanced Predictive Analytics**: Competing tools rely predominantly on historical data for risk analysis, potentially missing emerging risks. Our software uses predictive analytics, integrating real-time data and machine learning to identify nascent risks and trends before they escalate.

- **Customization and Scalability**: Many existing tools are rigid, making it difficult to adapt to unique organizational structures. Our tool is based on customization, allowing users to tailor risk assessments, workflows, and reporting to their specific needs, ensuring scalability.

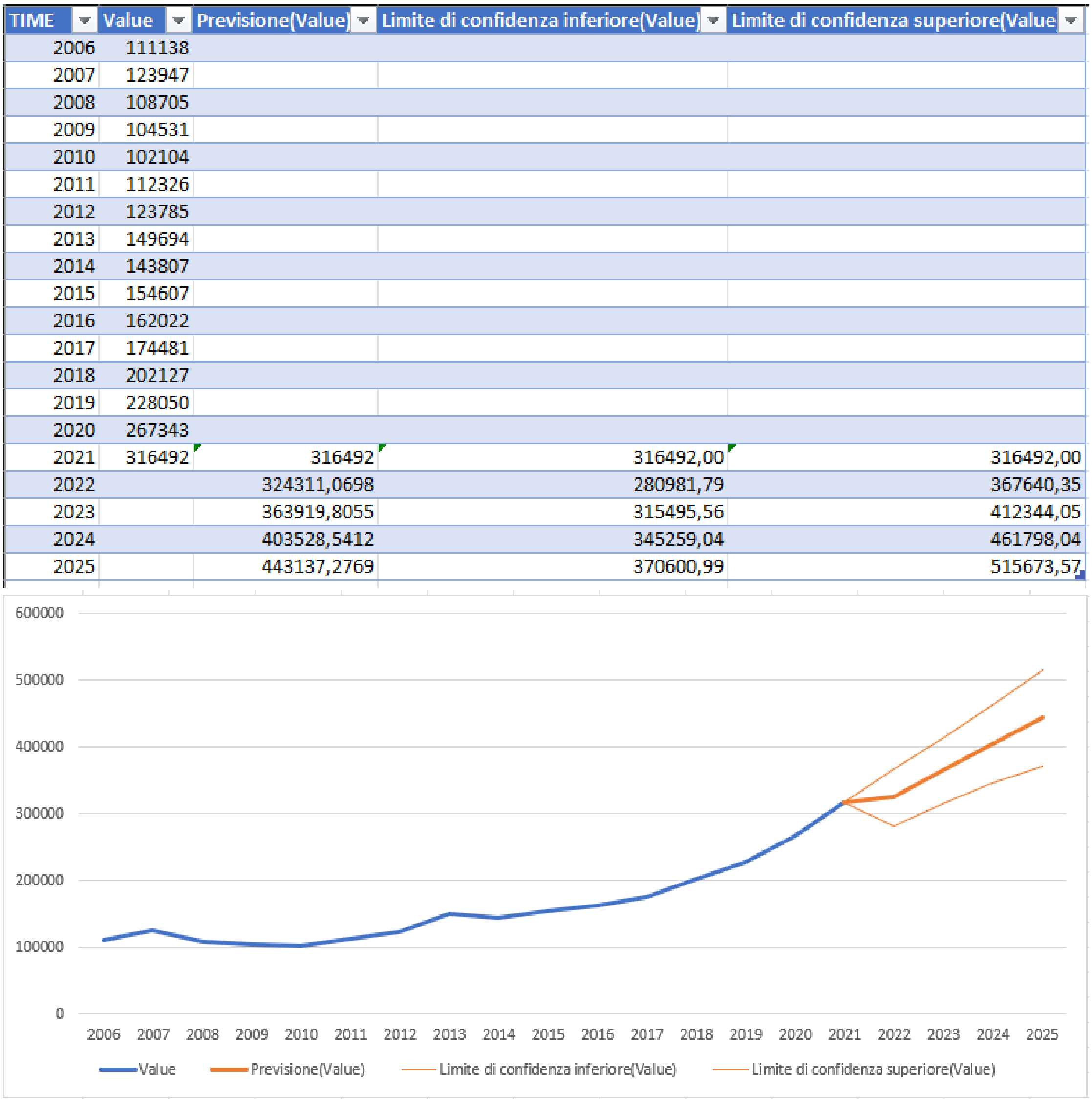
- **Intuitive User Experience**: Some competitors struggle with complex interfaces that get in the way of user adoption and efficiency. Our tool has an intuitive user interface designed with User Experience (UX) at its core.

3.5 Machine Learning For Risk Management

«Machine Learning (ML) refers to a system's ability to acquire, and integrate knowledge through large-scale observations, and to improve, and extend itself by learning new knowledge rather than by being programmed with that knowledge.» [11] Machine Learning (ML) is rapidly transforming risk management across various industries. By leveraging advanced data analytics and predictive modeling techniques, ML offers the potential to enhance risk assessment, mitigation, and decision-making processes. One of the primary applications of ML in risk management is predictive modeling. ML algorithms can analyze historical data to identify patterns and trends, enabling organizations to predict future risks more accurately. For instance, in financial risk management, ML models can forecast market fluctuations and assess the likelihood of credit defaults based on historical lending data. Different ML algorithms are used in risk management software. I will describe below the ones used during this study and their application. I will go deeper into used technologies and software integration details in the next chapters, which will focus on the project design and implementation.

3.5.1 Forecast Algorithms

Forecasting is the technique used for predicting future events based on the past and the present. Later, the resulting values are compared with the actual values to estimate the accuracy of the algorithm. Time series forecasting is useful to predict values based on past values observed at a specific time and at a given rate of observation. Forecasting techniques in machine learning are fundamental for predicting future risk-related events, which is essential for risk assessment and mitigation. For example, machine learning models can forecast financial risks such as stock price movements and currency exchange rates helping people and companies in making informed investment decisions and managing financial risks. Another application is supply chain predictions, such as delays in shipments or shortages of key materials. Accurate forecasts enable companies to plan for and mitigate these risks. In our study, time series forecasting has been used to analyze crime events in the past, in particular Cyber Crimes (an example in figure 3.5) and Assets Related Crimes, to analyze trends, and support Managers in loss events prevention and mitigation. Yearly data has been used, in order to predict future years' data and assess the risk.



TIME	Value	Previsione(Value)	Limite di confidenza inferiore(Value)	Limite di confidenza superiore(Value
2006	111138			
2007	123947			
2008	108705			
2009	104531			
2010	102104			
2011	112326			
2012	123785			
2013	149694			
2014	143807			
2015	154607			
2016	162022			
2017	174481			
2018	202127			
2019	228050			
2020	267343			
2021	316492	316492	316492,00	316492,00
2022		324311,0698	280981,79	367640,35
2023		363919,8055	315495,56	412344,05
2024		403528,5412	345259,04	461798,04
2025		443137,2769	370600,99	515673,57

Figure 3.5: Cyber crimes in Italy time series forecast. Based on ISTAT data from 2006 to 2021 and predicting values from 2022 up to 2025.
[12]

3.5.2 Classifier Algorithms

Classification models are used to categorize data into predefined classes or categories. The data provided is normalized and can be performed on both structured and unstructured data. Each data entry used to train the model has different attributes, one of each is the label, a category property that will later be predicted, that can have two or more possible values. In risk management classification algorithms can be employed to detect fraudulent activities in financial transactions. By analyzing transaction data, these models classify transactions by considering two classes: legitimate and potentially fraudulent. In this thesis work, classification algorithms have been used for different purposes:

- The two possible actions to be performed on risk are Acceptance and Mitigation, which can be considered as two classes and predicted using classifier algorithms. Suppose we have different previously assessed projects. It's possible to predict the correct strategy based on past values, such as assessed risk, likelihood, impact, and outcome of the project.
- Another application we explored, is related to loss events. It's possible to categorize the causes and consequences of a loss event and predict classes of possible future events. For example, if a company data breach caused a Consumer Lawsuit, other details on the cyber security event can be analyzed to predict if a future loss could have the same outcome.

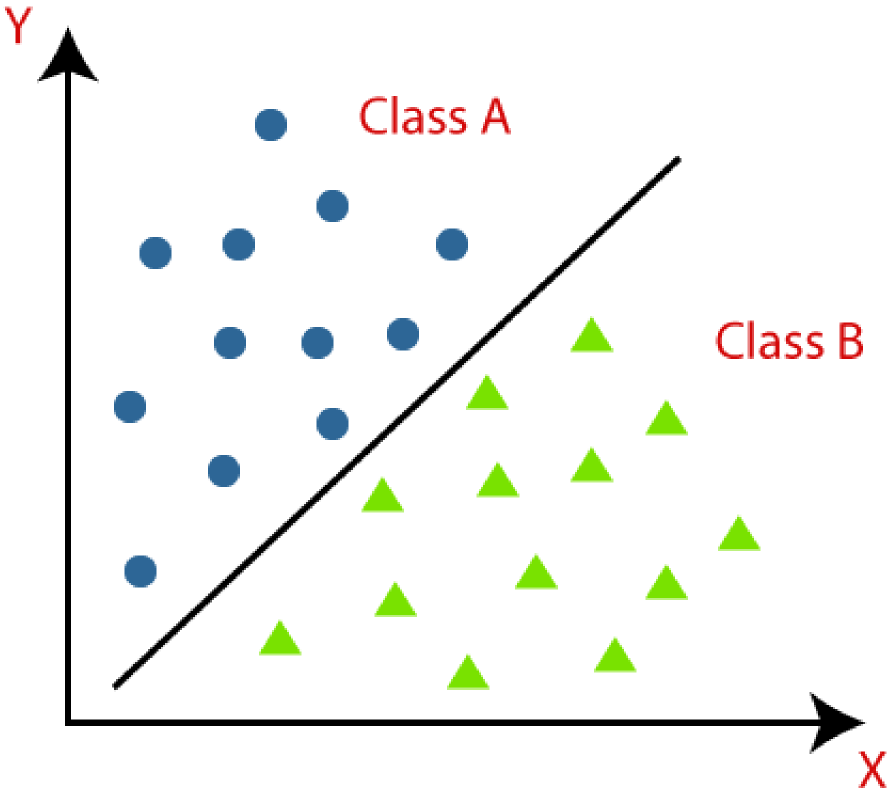


Figure 3.6: Classification Algorithm
[13]

It's important to note that machine learning models used for risk management should be trained on relevant historical data and regularly updated to adapt to changing factors. Additionally, The choice of the appropriate algorithm depends on the specific risk management problem and the nature of the data available. Model validation and ongoing monitoring are essential to ensure the models' accuracy and effectiveness in managing risks.

3.5.3 Challenges and Considerations

While ML is being used more and more in the risk management field, there are several challenges and considerations to keep in mind:

- **Data Quality and Quantity**: ML models rely on high-quality data. Inaccurate or biased data can lead to erroneous predictions. Also, a sufficient amount of data is necessary to obtain accurate values.
- **Interpretability**: Some ML algorithms, such as deep learning neural networks, can be challenging to interpret. Understanding why a model makes a particular prediction is crucial in risk management.
- **Model Validation**: ML models require rigorous validation to ensure their accuracy and reliability. Validation processes should align with regulatory requirements.
- **Ethical Concerns**: The use of ML in risk assessment raises ethical concerns, especially regarding bias and fairness. Careful consideration is necessary to avoid discrimination and bias in decision-making.

When used effectively and ethically, ML can enhance decision-making processes and contribute to improved risk management practices across various domains.

Chapter 4

Methodology and System Architecture

4.1 Introduction to Methodology and Design Principles

The road to success in software development is highlighted by the processes and design principles that guide the process. This chapter acts as an overview, covering the methodologies, design choices, and priorities that form the foundation of our software project. As we begin this review we examine the basic principles underpinning the development process, further clarifying the deliberate choices and strategies that motivated the work.

Especially in risk management, developing robust software requires a comprehensive approach that incorporates best practices, agility, and flexibility. It requires a combination of innovative design and machine learning capabilities that are not den. Careful focus on requirements and design elements requires a consistent strategy that delivers a Software as a Service (SaaS) solution capable of meeting the multifaceted needs of modern businesses.

In this chapter, we'll examine the profound impact of the Software as a Service (SaaS) model on the design, its use, and its advantages and challenges. We navigate the Agile landscape, illuminating our Agile approach's impact on iterative development, collaboration, and project adaptability. We'll also go deeper into the testing and deployment methods that have been critical to ensuring the quality and reliability of our software.

Also, we go deeper into the Domain Driven Design (DDD) phase, a foundational approach that has shaped the design of our software by emphasizing the importance of domain modeling. In addition, we explain the seamless integration of machine learning, where data-driven insights and predictive analytics enhance the core functionality of our software.

As we traverse the sections of this chapter, we provide a detailed overview of the requirements documents and design materials that contributed to the design of our software. Finally, we'll draw conclusions on the design choices.

4.2 Software as a Service (SaaS) Model

Software as a Service (SaaS) is a cloud-based service where instead of downloading software on a desktop PC or business network to run and update, the application is accessed via an internet browser. The software application could be anything from office software to unified communications among a wide range of other business apps that are available. [14] SaaS operates via the cloud delivery model, where a software provider takes responsibility for hosting the application and its associated data. This hosting can occur on the provider's own infrastructure, including servers, databases, networking, and computing resources. Alternatively, it may involve an Independent Software Vendor (ISV) partnering with a cloud provider to host the application within the cloud provider's data center. The beauty of this approach lies in its accessibility; users can access the SaaS application from any device connected to the internet. Typically, SaaS applications are accessed conveniently through standard web browsers. As a consequence, organizations employing SaaS applications are relieved from the burden of setting up and configuring the software. Instead, users only need to pay a subscription fee to use the software. In the SaaS model, the service provider offers customers access to a network-hosted application that has been built for SaaS distribution. This application employs a single, standardized source code that remains consistent for all customers. Any updates, enhancements, or new features introduced by the provider are uniformly made available to all users. The storage of customer data depends on the specific Service Level Agreement (SLA) and can occur locally, in the cloud, or in a combination of both environments [15].

4.2.1 A multi-tenant architecture

SaaS applications and services commonly adopt a multi-tenant approach. This means that a single instance of the SaaS application runs on the hosting servers, serving each subscribing customer or cloud tenant. This single instance maintains a consistent version and configuration that applies to all customers or tenants. Despite multiple subscribing customers utilizing the same cloud instance with a shared infrastructure and platform, data from different customers remains separate.[15]

The multi-tenant architecture of SaaS applications provides several advantages. It enables cloud service providers to manage maintenance, updates, and bug fixes more efficiently and promptly. Instead of implementing changes across numerous instances, engineers can make necessary modifications for all customers by maintaining a unified, shared instance. Additionally, multi-tenancy ensures a broader pool of resources is available to a larger user base without compromising essential cloud functions like security, speed, and privacy.[15]

4.2.2 Advantages and disadvantages of SaaS

SaaS offers several benefits which lead companies to prefer it with respect to standard software. Organizations no longer need to install and manage applications on their own hardware, reducing expenses related to hardware acquisition, provisioning, and maintenance. Additionally, there's no need to worry about software licensing, installation, and support. Instead of purchasing software or additional hardware, customers subscribe to a SaaS service, shifting costs to predictable recurring operating expenses. Users have the flexibility to terminate SaaS subscriptions whenever they want, stopping recurring costs. Also, SaaS provides high vertical scalability, allowing customers to access different software components as needed. SaaS providers handle software updates management, relieving the IT staff from this burden and SaaS applications are accessible over the internet from any device and location, offering users flexibility and portability. Those applications are often customizable and can be seamlessly integrated with other business applications, particularly those from the same software provider.[15]

While SaaS offers numerous advantages, it also presents some potential challenges and risks. Businesses may face disruptions when SaaS providers experience service interruptions, make unwelcome changes to service offerings, or suffer security breaches. To proactively address these challenges, customers should understand their SaaS provider's Service Level Agreement (SLA) and ensure its enforcement. Customers have limited control over versioning. When a provider adopts a new application version, it is typically rolled out to all customers. This may require organizations to allocate additional time and resources for training. Switching SaaS vendors can be challenging, as it often involves migrating substantial amounts of data. Some vendors use proprietary technologies and data formats, complicating data transfer between different cloud providers. Vendor lock-in occurs when customers find it difficult to transition between service providers due to these factors. Security is frequently cited as a significant challenge for SaaS applications. Ensuring data security in a shared cloud environment is crucial for businesses.[15]

4.2.3 Why we choose the SaaS Model

Choosing a Software as a Service (SaaS) model for our risk management tool was a strategic decision driven by several key factors. First of all, the SaaS model aligns well with our goal of providing accessible and convenient risk management solutions to users. By hosting the application in the cloud, we ensure that our tool is easily accessible everywhere, allowing collaboration to work with team members regardless of geographic location. Additionally, the SaaS model enables us to deliver automated updates and enhancements, ensuring that our users always have access to the latest features and security enhancements without the hassle of manual installation. This approach not only simplifies the user experience but also provides cost savings as organizations can subscribe to our service without the need for significant hardware investment or complex infrastructure. Furthermore, the multi-tenant architecture is aligned with our idea of a modular Web Application: different users can select only the modules they need, and pay a fee according to chosen services. Overall, the SaaS model increases accessibility, scalability, and cost, while allowing us to focus on delivering a secure and reliable risk management solution in line with the evolving needs of users.

4.3 Agile Methodology

Agile methodologies focus more on code and implementation, and less on documents. The goal of Agile Methodologies is to satisfy the client (not only with respect to the contract) by giving early and continuous delivery of valuable software, and giving methodologies able to reduce the cost and the time of software development, increasing the quality. The Agile methodologies are based on the Agile Manifesto [16] whose main points are:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan.

From those principles, it is evident that the highest priority is given to the customer, welcoming changes in the requirements even late in development, and delivering working software frequently (from a couple of weeks to months, preferring shorter timescales). Furthermore, the Agile principles suggest building projects around motivated individuals, giving them the needed support and an environment with the needed characteristics. These individuals should communicate quite always face to face, incrementing the collaboration and communication inside the time. Agile suggests being as simple as possible, being clearer in code and documentation, projecting and modeling: the result of such simplicity is a project that is more readable and easier to modify when needed. The full agile principles are shown in table 4.1.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Business people and developers must work together daily throughout the project.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Working software is the primary measure of progress.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Continuous attention to technical excellence and good design enhances agility.
Simplicity–the art of maximizing the amount of work not done–is essential.
The best architectures, requirements, and designs emerge from self-organizing teams.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Table 4.1: The Agile Manifesto
[17]

During the thesis work, the Agile methodology has been used for the development of the web application. In particular, the Scrum framework has been adopted. Such framework is the topic of the next section.

4.3.1 The Scrum Framework

This subsection will follow the official Scrum Guide [18], mentioning the most important components and principles. Scrum is a lightweight framework for developing and sustaining complex products. It is based on empirical process control and as shown in figure 4.1 consists of defined roles, events, artifacts, and rules.

The Scrum Team includes three fundamental roles:

- **Product Owner**: Manages the Product Backlog and prioritizes the work.
- **Scrum Master**: Not a boss - but a leader, who helps the team understand and implement Scrum principles and practices following the Guide.

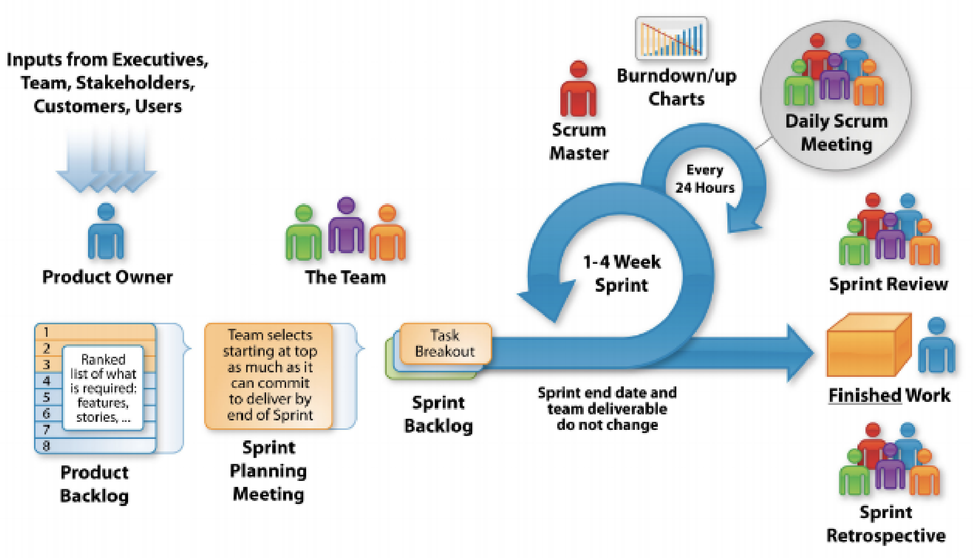

Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

Figure 4.1: Representation of the Scrum Process (Ref: "Analysis of User Stories and Effort Estimations in Agile Software Development", written by Rupert Dürre)

- **Development Team**: Self-organizing and cross-functional group responsible for delivering an increment of the product. Every member has the same value.

Other crucial components of the Scrum framework are the five Scrum Events:

- **Sprint**: A time-boxed iteration (typically 2-4 weeks) where the Development Team creates a potentially shippable product increment.
- **Sprint Planning**: A meeting at the start of the Sprint to plan the work that will be done during the Sprint.
- **Daily Scrum**: A short daily meeting (15 minutes) for the Development Team to synchronize their work and plan for the day.
- **Sprint Review**: A meeting at the end of the Sprint to inspect and adapt the product increment.
- **Sprint Retrospective**: A meeting at the end of the Sprint for the Scrum Team to reflect on their process and identify areas for improvement.

Finally, we have Scrum Artifacts:

- **Product Backlog**: An ordered list of all the work that could be done on the product. The product backlog items are composed using the user stories format. A user story is an essential description of a functional requirement. A typical example of its format is:

```
As a <actor type>
I want <to do something>
So that some value is created.
```

- **Sprint Backlog**: The subset of the Product Backlog selected for the current Sprint.
- **Increment**: The sum of all the completed work from previous Sprints, which must be in a potentially shippable state.

In order to decide when a product increment can be declared shippable, a common Definition of Done (DoD) must be met for a product increment to be considered complete and potentially releasable. Also, for every sprint a Sprint Goal must be defined: a short statement that describes the purpose of the Sprint and what the Development Team intends to achieve.

Scrum is based on three pillars of empiricism:

- **Transparency**: Everyone involved in the project should have a clear and shared understanding of the work and its progress.
- **Inspection**: It has to be done frequently and attentively on artifacts and events to detect issues that may bring a deviation to the product goal.
- **Adaptation**: The team should make continuous improvements based on the information gathered during the inspection phase.

Scrum emphasizes five values - commitment, courage, focus, openness, and respect - which help teams work effectively together.

4.3.2 Agile and Scrum Application during the Project

During my thesis experience in Orbyta Tech, I worked in a team where there was one product owner who also behaved as a scrum master and a team of developers, in which I was in charge of the main work, while other developers mostly had a supporting role. This was given to the fact my project was academic research with the goal of developing a Proof of Concept (PoC). The project was only later shown to possible customers. The product owner was of course in charge of the Product Backlog. However, the process was done together with myself, as a chance to experiment also that side of development. This means that, for academic purposes, my role varied from that of a Product Owner to that of a Full Stack Developer. The user stories format explained in the previous subsection was implemented. For the sprint planning, we used Azure's DevOps "Boards" and "Sprints" sections, as later described in the following section. I attended several meetings with the Product Owner/Scrum master, to understand requirements and user needs. The product backlog had some changes over the course of months. As previously explained, we mainly worked on two modules of the application, which will later be composed of many more. This is the reason why the product backlog only contains stories related to these. Most of the time was dedicated to theoretical and technical studies, to improve both the content and software quality. Every functionality and study was split into self-assigned tasks, which were completed following our Definition of Done that consisted of pushing on Git working and end-to-end and unit-tested code. Daily scrum meetings were set up every day in the morning, informally in the office where I had to update the other team members about the state of my work, listing the goals reached, eventual obstacles, and future implementing goals. After every sprint, a sprint retrospective was performed where the application with the new features was presented in a meeting and the team members and product owner participated. Then, a new sprint was planned adding issues to solve in the product backlog and choosing the other requirements to be delivered in the next sprint.

4.4 Testing and Deployment Strategies

In the software development process, testing and deployment are critical phases that ensure a product's performance and reliability. This section delves into the strategies employed during the development process, emphasizing the significance of thorough testing and the streamlined deployment processes.

4.4.1 Testing Strategies

Effective testing is a crucial phase in producing high-quality software. During the development of our risk management tool, we implemented a multifaceted testing strategy to validate every aspect of the application. This strategy encompassed various types of testing, including:

- **Unit Testing**: At the beginning of our testing process, unit tests were conducted to verify the correctness of individual code units or modules. This granular testing approach enabled us to isolate and fix issues early in the development cycle, ensuring code integrity.

- **Integration Testing**: To evaluate the interactions between different components and modules, integration tests were performed. This ensured that the integrated parts of the system functioned seamlessly, and data flowed correctly between them.

- **End-to-End Testing**: End-to-end testing was manually performed through the whole application development. It was used to check that the software met the functional criteria outlined in our design and requirement documents.

More details on testing implementation will follow in the chapter 5.

4.4.2 Agile Testing

For the project's development, an Agile testing strategy has been followed. Agile testing is a fundamental component of the Agile software development methodology. It's a dynamic and iterative approach that aligns with Agile principles, emphasizing collaboration, continuous improvement, and customer-centricity. Unlike traditional development methodologies, where testing is often a later phase, Agile integrates testing throughout the entire development lifecycle. One of the core principles of Agile testing is "early and continuous testing." This means that testing activities start at the beginning of a project and are maintained throughout its duration. Testing is not a separate phase but a parallel and integrated part of development. This approach enables teams to identify and rectify issues promptly, promoting higher software quality. At the end of each sprint, testing was performed to validate that the new functionality aligned with user expectations.

Collaboration is another fundamental aspect of Agile testing. It encourages close cooperation among team members, including developers, testers, product owners, and stakeholders. Testers work closely with developers to define acceptance criteria, create test cases, and validate that user stories meet the expected criteria. In the case of the Risk Management Tool, there wasn't a specific developer with the role of the tester. However, while tests were coded by me, the whole team decided the criteria for which the tests were considered useful to determine a successful sprint.

Frequent regression testing is a necessity within Agile. Since development involves continuous changes and updates, regression tests are rerun to verify that new changes haven't introduced defects into previously working functionality.

Automation is another crucial aspect of Agile testing. Automated testing with Azure DevOps was implemented to automate repetitive and time-consuming test cases. This enables testers to focus on exploratory and critical testing. Automated tests are executed in each iteration, offering rapid feedback on the health of the software.

User Acceptance Testing (UAT) is often a part of Agile projects, ensuring that the software aligns with user expectations. UAT is typically performed collaboratively with business stakeholders to validate that the software meets their specific needs. Agile testing nurtures a culture of continuous feedback. Testers provide feedback to developers regarding defects and issues, while business stakeholders offer feedback on whether the software aligns with their objectives. This feedback loop is vital for making necessary adjustments promptly and enhancing the overall quality of the software. During the project's development process, continuous feedback was provided by members of the team testing the application. However, towards the end of the six months, more people were asked to review the application, including experts. This helped locate defects and identify possible future developments.

In Agile, quality assurance is a shared responsibility across the team. It's not the sole domain of testers but a commitment from all members. Accurate testing uncovers and addresses software defects, enhancing the application's stability and user experience. It ensures that the tool functions as expected, resulting in satisfied users who can rely on its accuracy and performance.

4.4.3 Continuous Integration and Continuous Delivery (CI/CD)

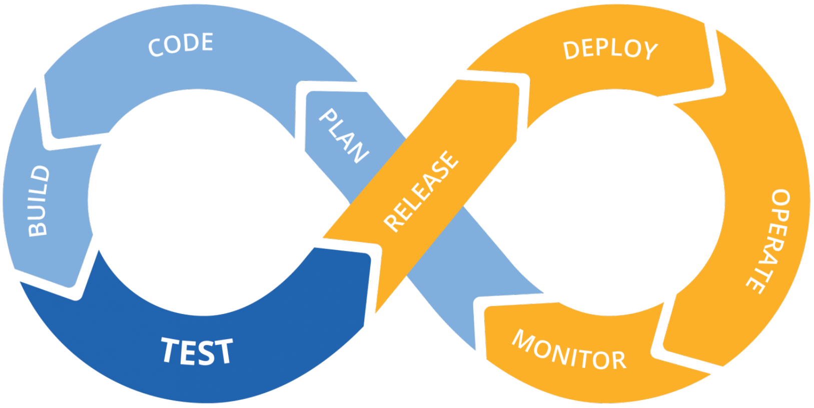


Figure 4.2: CI/CD
[19]

The development process was significantly enhanced through the implementation of Continuous Integration (CI) and Continuous Delivery (CD) DevOps pipelines. CI/CD pipelines are an integral part of modern software development practices, facilitating automation, collaboration, and the rapid delivery of updates. Our CI/CD setup allowed for:

- **Continuous Integration**: Continuous Integration is a development practice that involves frequently integrating all code changes into a shared source code repository's main branch. Each change is automatically tested upon committing or merging, triggering an automated build process. By adopting CI, developers can swiftly identify and address errors and security issues in the early stages of development. Continuous integration promotes the seamless integration of code changes, even when multiple developers are collaborating on the same application. It minimizes the potential for code conflicts, and quick feedback allows for the prompt resolution of bugs and security vulnerabilities. CI typically begins with a static code analysis to assess code quality. Once the code passes these static tests, automated CI processes package and compile the code for further automated testing. A version control system is essential within CI to track code changes and maintain version consistency.

- **Continuous Delivery**: Continuous Delivery is a complementary software development practice that aligns closely with CI. After the code has undergone testing and building during the CI process, CD takes over the final stages, ensuring that the code is packaged for deployment to any environment at any time. CD encompasses everything from provisioning infrastructure to deploying the application to testing or production environments. With CD, the software is constructed to be deployable to production at any moment. Deployment can be initiated manually, or organizations can transition to continuous deployment, where deployments are automated as well.

- **Continuous Deployment**: Continuous Deployment is an advanced stage that automates the entire deployment process, eliminating the need for human intervention. In continuous deployment, DevOps teams set predefined criteria for code releases, and when these criteria are met and validated, the code is deployed automatically to the production environment. This approach enables organizations to be agile and introduce new features to users swiftly. While continuous integration can exist independently of continuous delivery or deployment, achieving continuous deployment necessitates having a solid CI foundation. This is because deploying to production at any time requires fundamental CI practices, such as regularly integrating code into a shared repository, automating testing and builds, and performing these activities in small, regular batches.

- **Continuous and Automated Testing**: Continuous Testing is a fundamental software testing practice where tests are executed continuously to detect bugs as soon as they are introduced into the codebase. In a CI/CD pipeline, continuous testing is typically automated, with each code change triggering a series of tests to validate that the application behaves as expected. This approach helps identify issues early in the development process, preventing them from becoming more complex and costly to rectify later on. Continuous testing also provides valuable insights to developers about their code's quality, enabling them to identify and address potential problems before they reach production. Continuous testing encompasses various types of tests within the CI/CD pipeline, including unit testing to verify individual code units, integration testing to assess the interactions between different application modules or services, and regression testing to ensure that previously resolved bugs do not resurface. [20]

In conclusion, testing and CI/CD pipelines have been instrumental in shaping the development of our risk management tool, reinforcing our commitment to delivering a high-quality, dependable, and user-friendly solution. The rigorous testing processes and streamlined deployment strategies have positioned our tool for success in real-world risk management scenarios. These strategies were implemented through Azure DevOps pipelines, more technical details will be explained in chapter 5.

4.5 Domain Driven Design (DDD)

Domain Driven Design (DDD) is a software development methodology that places the focus on the core domain of a business, aiming to align software systems with the complexities of that domain. The DDD approach aligns application development with the SOLID principles, a set of five fundamental principles deeply rooted in Object Oriented Programming (OOP). These principles were initially introduced in the publication "Design Principles and Design Patterns" by renowned American software engineer Robert Martin [21]. The overarching goal of SOLID is to foster the creation of software that is not only comprehensible but also more maintainable and extensible.

- The **Single Responsibility Principle (S)** emphasizes that each module or class within an application should shoulder the responsibility for just one specific functionality provided by the application. This principle fosters clarity and simplicity in code design.

- The **Open/Closed Principle (O)** dictates that existing and operational software entities, such as classes, modules, and methods, should be open to extension while remaining closed to direct modifications. Essentially, alterations to established code should only occur when addressing internal bugs; otherwise, extension through mechanisms like inheritance is favored.

- The **Liskov Substitution Principle (L)** asserts that objects of child classes should be capable of seamlessly replacing objects of their parent classes. Adhering to this principle ensures that such substitutions won't compromise the application's intended behavior.

- The **Interface Segregation Principle (I)** advocates for the division of larger interfaces into smaller, more focused ones. This allows developers to concentrate solely on creating methods pertinent to a specific class, thus avoiding the implementation of unnecessary methods that larger interfaces might impose.

- The **Dependency Inversion Principle (D)** introduces the dependency injection pattern as a solution to a common OOP issue related to tightly coupled classes. It advocates for the replacement of concrete dependencies with abstractions, mitigating potential compilation errors that may arise when instantiating classes with strong dependencies. The dependency injection pattern provides a practical implementation of this principle.

Adhering to these SOLID principles while applying the DDD approach in software development leads to more structured, maintainable, and flexible codebases. DDD was popularized and refined by Eric Evans in his seminal book "Domain-Driven Design: Tackling Complexity in the Heart of Software" [22]. Halil İbrahim Kalkan's "Implementing Domain Driven Design" [23] serves as a practical guide for putting DDD principles into action and was used as a guide through the development process, since the ABP.io framework used for coding the application is based on those guidelines.

There are four fundamental layers of a Domain Driven Based Solution. **Business Logic** is divided into two distinct layers: the Domain Layer and the Application Layer, each serving different aspects of the business logic.

- The **Domain Layer** is responsible for embodying the core business logic that is independent of specific use cases within the domain or system.

- The **Application Layer**, on the other hand, focuses on implementing the use cases of the application, which can be thought of as user interactions within the User Interface (UI).

- The **Presentation Layer** encompasses all UI elements, such as pages and components, that compose the application's user interface.

- The **Infrastructure Layer** plays a supportive role by implementing abstractions and integrations with third-party libraries and systems, thereby providing essential support to the other layers.

DDD primarily places its emphasis on the Domain and Application Layers while treating the Presentation and Infrastructure Layers as secondary elements. These latter layers are considered to be implementation details, and the core principle is to ensure that the business layers remain independent of them, minimizing any dependencies.

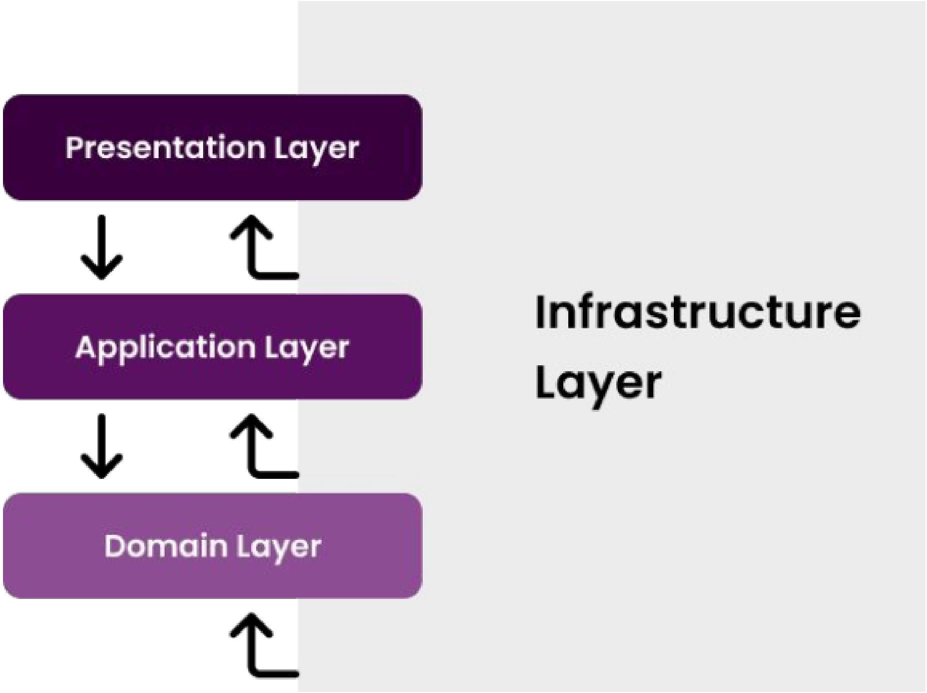


Figure 4.3: Domain Driven Design Architecture
[23]

4.5.1 Key Concepts of Domain Driven Design

DDD is based on some key concepts fundamental for implementing the methodology in a correct way:

- **Ubiquitous Language**: DDD emphasizes the importance of establishing a shared vocabulary between software developers and domain experts. This language, known as the "ubiquitous language," ensures that everyone involved in the project understands the domain-specific terms and concepts. It bridges the gap between business stakeholders and technical teams, facilitating effective communication.

- **Bounded Contexts**: DDD recognizes that large software systems often comprise multiple subdomains, each with its own distinct rules and constraints. To manage this complexity, DDD introduces the concept of "bounded contexts." Bounded contexts define explicit boundaries within which a specific domain model is valid. This allows for the creation of distinct models for different parts of the system, preventing conflicts and confusion.

- **Aggregate Roots**: In DDD, aggregates are clusters of domain objects that are treated as a single unit. Each aggregate has an "aggregate root" that serves as the entry point for interacting with the objects within the aggregate. This concept enforces consistency and encapsulation within the domain model.
- **Repositories**: Repositories provide a mechanism for accessing and storing aggregates. They abstract away the details of data storage and retrieval, allowing developers to work with domain objects without concerning themselves with the underlying data infrastructure.
- **Value Objects and Entities**: DDD distinguishes between value objects and entities. Value objects are immutable objects that derive their identity from their attributes. Entities, on the other hand, have a distinct identity that persists over time. Understanding this distinction is crucial for modeling domain objects effectively.

4.5.2 Benefits of Domain Driven Design

DDD ensures that software systems closely align with the business domain, leading to solutions that genuinely address business needs. By breaking down complex domains into bounded contexts and well-defined aggregates, DDD promotes modular and maintainable code. Moreover, the ubiquitous language helps collaboration between domain experts and developers, reducing misunderstandings and enhancing the development process. DDD's focus on bounded contexts allows for the independent development and scaling of different parts of a system. DDD encourages the creation of a rich domain model, leading to higher code quality and fewer defects.

By embracing DDD principles, developers can build software that stands the test of time and delivers genuine value to organizations and end-users alike.

4.6 Data Research and Analysis

4.6.1 Data Research

The data research phase of this project played a crucial role in shaping the application's functionalities. Collecting historical data related to the module's domains it's the first step to provide useful support to an organization's Risk Management. The data research focused on the two modules of the application which were part of the thesis project's studies: Data Loss Prevention (Cyber Security) and Asset Loss and Logistics Risk Management. Extensive efforts were dedicated to sourcing and adapting external datasets to provide users with valuable risk assessment information. Cyber security is a developing field, so multiple academic datasets were found online. On the other hand, the approach we decided to have for the

Asset Loss and Logistics Risk Management tool, consisted of creating a connection between reported crimes that affected material goods and geographical locations.

To meet the specific needs of the Data Loss Prevention module, a variety of datasets were analyzed, as online open datasets didn't have enough entries to consider a single data source. Key data sources included:

- ISTAT «Delitti denunciati dalle forze di polizia all'autorità giudiziaria»: we used ISTAT's REST APIs to extract Italy's data, by selecting the categories «Delitti Informatici» and «Truffe e frodi informatiche» from the Italian police's records. [12]
- Double Extortion Dataset: the double extortion platform's dataset was used to analyse data regarding extortion attacks, in particular social engineering and ransomware. [24]
- The Data Loss Attrition Dataset [25]
- Kaggle Data Breaches Dataset [26]

For the Asset Loss & Logistics Risk Management, the risk assessment drew extensively from ISTAT REST APIs, with a focus on Italian police reports. In particular, the nData [27] guide was used to dig deeply into the data from 2006 to 2021. The analysis encompassed complaint categories linked to product, cargo, and shop damage, including "Arson", "Attack", "Damage", "Arson Damage", "Shop Robbery", "Shop Theft", and "Truck Theft." [12] Data extraction was conducted across the entirety of Italy, with API queries tailored to each Italian province. This was given to the fact that considering other countries would have required more careful and time-consuming research which wouldn't have had the same detailed results (province precision) as the ones we obtained by focusing just on the Italian territory. The datasets were meticulously adapted and normalized for seamless integration into the software.

4.6.2 Data Analysis

The Data Analysis phase was used to extract information useful for the implementation phase, explained in chapter 5, by analyzing the previously listed datasets. Following the dataset order of the previous subsection, I'll explain what information was determined and by using which methodology.

For the Data Loss Prevention module, ISTAT's data [12] on cyber crimes and cyber frauds was used to perform a Forecast (as explained in 3.5.1) to assess the trend and perform predictions, with a horizon of 4 years. Figure 3.5 shows that cyber crimes in Italy are increasing each year. The consequence of these increasing numbers should inevitably lead to an improvement in risk prevention measures.

For example, to prevent cyber crimes, companies should adopt more effective security measures. At the same time, a solution for cyber fraud could consist of starting a mitigation plan for instructing employees on social engineering attacks and sensitive information disclosure. Regarding social engineering attacks, another dataset that was extremely useful was the double extortion dataset [24]. It is the most updated dataset found. Every time a new extortion event leading to a data breach happens, it's recorded in the database, and statistics are computed. The study performed on this dataset consisted of observing statistics shown in the dashboard and observing results that could be useful in assessing cyber risk. In figure 4.4 we can see two graphics showing the most threatened industries on the left and the most threatening actors on the right. The first information is useful in planning mitigations according to the business type, for example, the chart shows that Business Support Services and Heavy Construction need a better Cyber Security Protection plan. This could also help in tailoring risk plans according to third-party and customer relationships. The second chart is useful in knowing where the risk comes from, this could lead to a more accurate employee instruction program and a better antivirus and firewall configuration. This analysis can be enriched by a time chart in figure 4.5, which helps instead in understanding how these threat actors are evolving over time. Country distribution (figure 4.6) shows that America and Europe have a higher rate of cyber extortion crime. This information is also useful in protecting organizations from cyber risk, by protecting external communications. The next dataset to be analyzed is the Attrition Data Loss Dataset [25]. The importance of this dataset was given by the more specific and useful attributes contained in each entry. However, the reliability of the extracted information must be taken with a grain of salt, since the last entry goes back to 2008. The value of this dataset is given by the boolean values related to data loss and the company's legal repercussions of these types of events. The three attributes that were evaluated are Data Recovered, Consumer Lawsuit, and Arrest Prosecution, and the values are shown on the charts in figure 4.7. These properties were also used as classes to perform predictions on user-inserted values (see 3.5.2). The last dataset used, the Data Breaches Dataset [26] was used to go deeper into the most popular breach methods and organization types attacked. From the charts in figure 4.8 we can see that Web, Healthcare, Financial, and Government are the most affected organizations, and Hacking, Physical, and Poor Security are the most diffused breach methods. This last piece of information, once again, suggests a necessary improvement in companies' security.

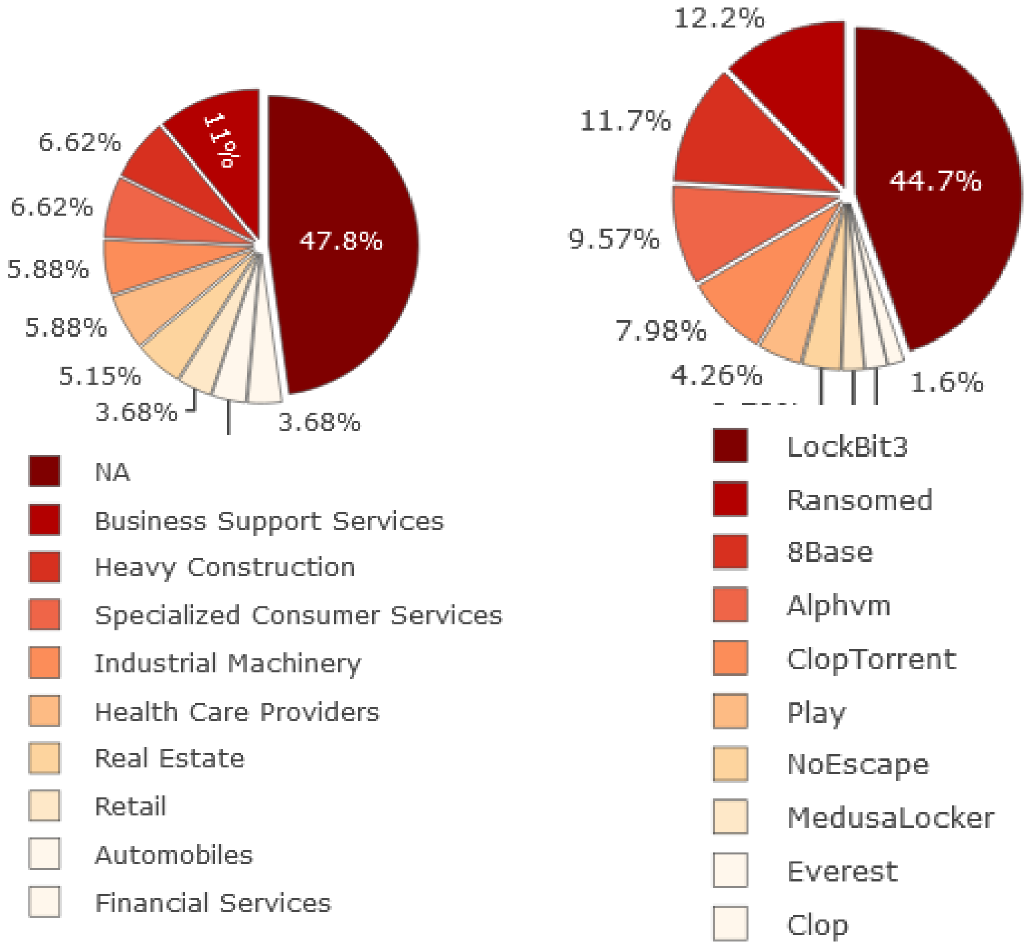


Figure 4.4: On the left the Industry Vertical plot shows the distribution of events among the top 10 impacted business sectors. On the right, the Actor Statistics plot shows the distribution of the top 10 threat actors or affiliation programs causing most of the observed attacks.

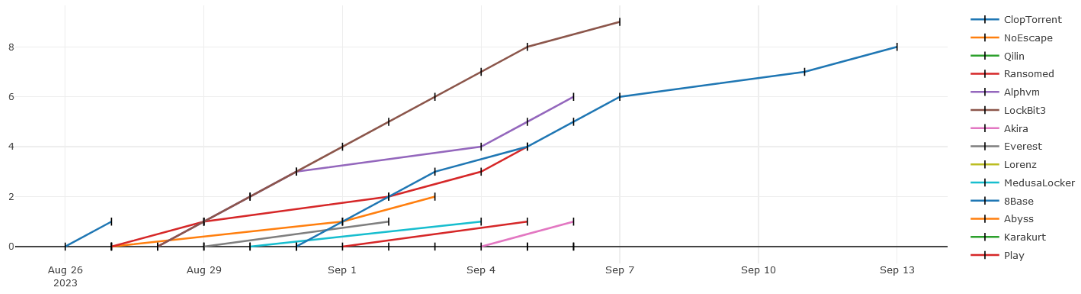


Figure 4.5: The Actor Activities over time plot reports the observed double extortion attempts grouped by threat actor or affiliation program.

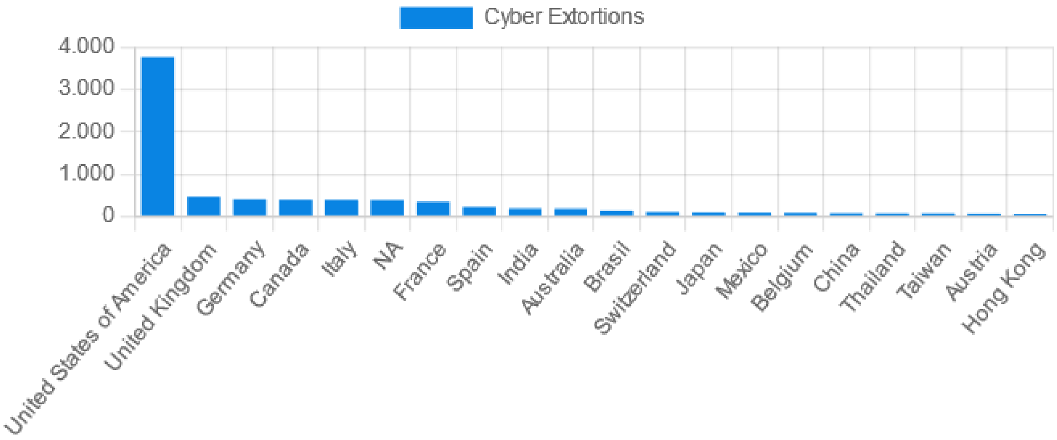
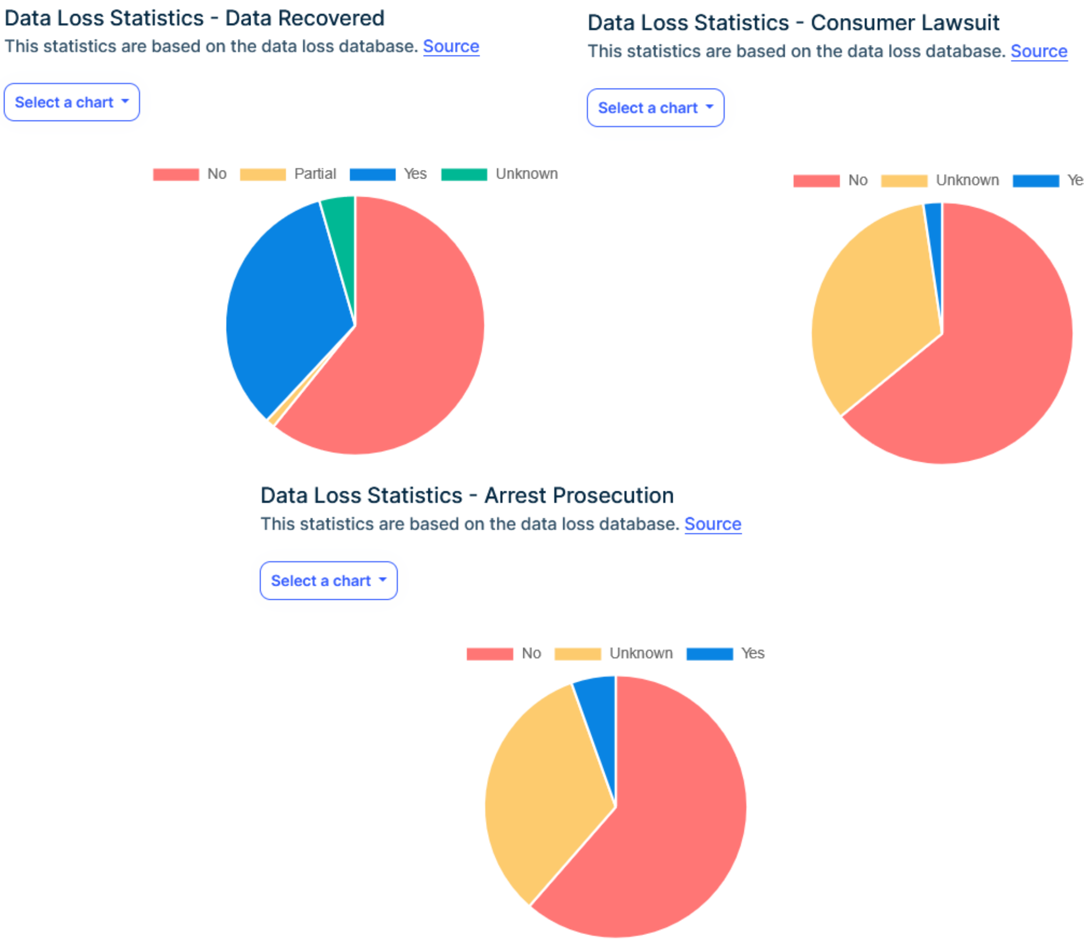


Figure 4.6: The last update was performed on the 15/06/2023 (observed 9119 events across 176 countries). This chart was taken from the Risk Management Tool.



Data Loss Statistics - Data Recovered

This statistics are based on the data loss database. Source

Data Loss Statistics - Consumer Lawsuit

This statistics are based on the data loss database. Source

Data Loss Statistics - Arrest Prosecution

This statistics are based on the data loss database. Source

Figure 4.7: Data Recovered, Consumer Lawsuit, Arrest Prosecution. These charts are taken from the Risk Management Tool.

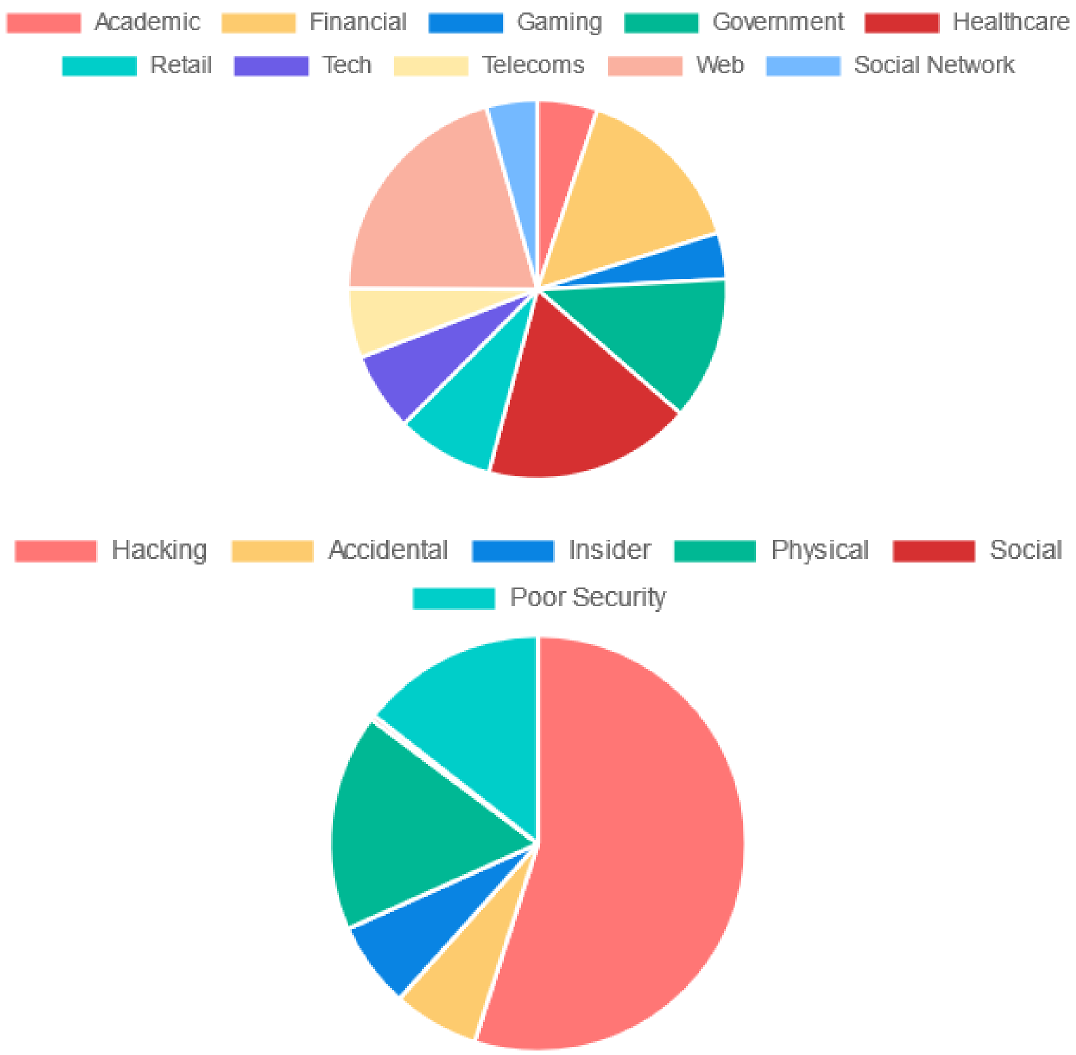


Figure 4.8: On the top Top 10 Organization Types affected by Data Breaches. On the bottom top 10 most popular Breach Methods. These pictures were taken from the Risk Management Tool.

I'll now discuss the data analysis performed on ISTAT's Asset Crimes Data. The columns extracted from the APIs were the City, the Year, the type of Crime ("Arson", "Attack", "Damage", "Arson Damage", "Shop Robbery", "Shop Theft", or "Truck Theft.") and the number of reported crimes during that year in that place. Unlike Cyber data, Asset data wasn't analyzed on the whole Italian territory. However, statistics were performed for each province. As an example, in figure 4.9 the forecast for the city of Turin, and in figure 4.10 the chart shows the different types of attacks on a pie chart. Computing these statistics on every Italian province had the goal of comparing them, in order to obtain an "Asset Crime Rate" for each and obtain the most and least dangerous provinces in which companies can transport goods. To compare different values, we needed to have comparable numbers. To achieve these results we used the following normalization formula:

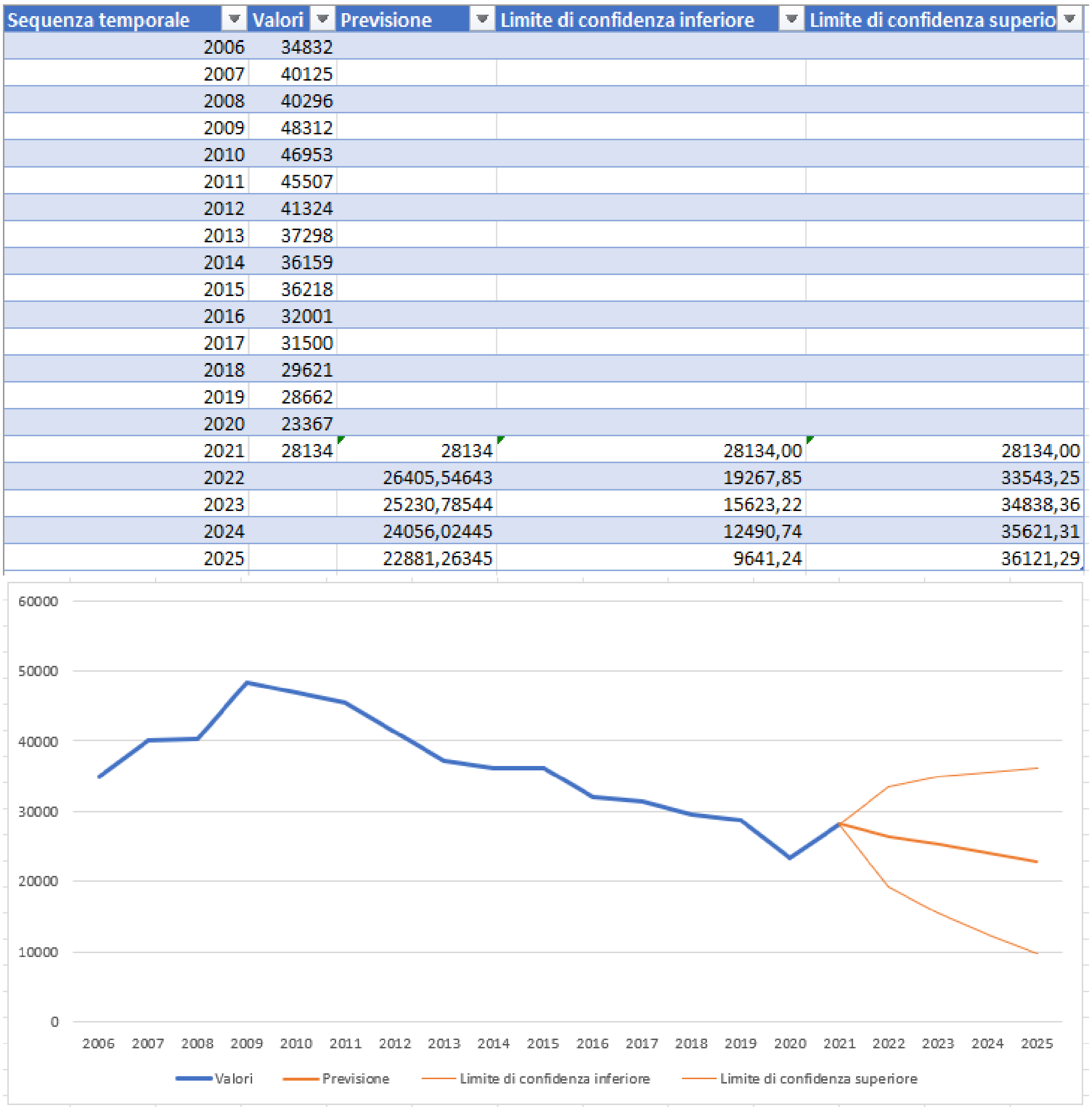
$$X_{\text{normalized}} = \frac{X - X_{\text{minimum}}}{X_{\text{maximum}} - X_{\text{minimum}}} \tag{4.1}$$

where:

- X is the number of asset crimes in the current year.
- $X_{minimum}$ is the lower asset crimes number in all considered years.
- $X_{maximum}$ is the higher asset crimes number in all considered years.

After calculating all the values, it was possible to put them on a colored map (figure 4.11), which indicates the likelihood value for asset crimes to happen in different Italian provinces.

This careful data analysis was performed with the aim of being integrated into the software, both through automatic machine learning mechanisms and statistical charts supporting the Risk Managers using the application in their decisions.



Sequenza temporale	Valori	Previsione	Limite di confidenza inferiore	Limite di confidenza superio
2006	34832			
2007	40125			
2008	40296			
2009	48312			
2010	46953			
2011	45507			
2012	41324			
2013	37298			
2014	36159			
2015	36218			
2016	32001			
2017	31500			
2018	29621			
2019	28662			
2020	23367			
2021	28134	28134	28134,00	28134,00
2022		26405,54643	19267,85	33543,25
2023		25230,78544	15623,22	34838,36
2024		24056,02445	12490,74	35621,31
2025		22881,26345	9641,24	36121,29

Figure 4.9: Asset crimes in Turin time series forecast. Based on ISTAT data from 2006 to 2021 and predicting values from 2022 up to 2025.

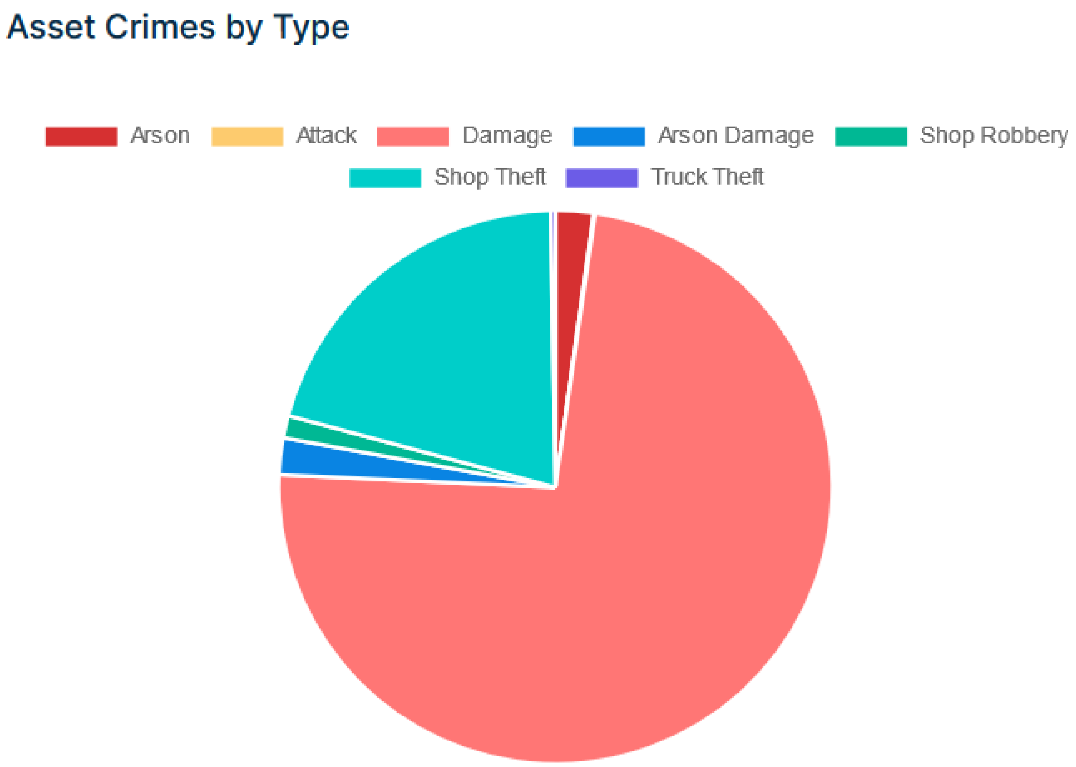


Figure 4.10: Asset crimes by type in Turin. Based on ISTAT data from 2006 to 2021. This chart was taken from the Risk Management Tool.

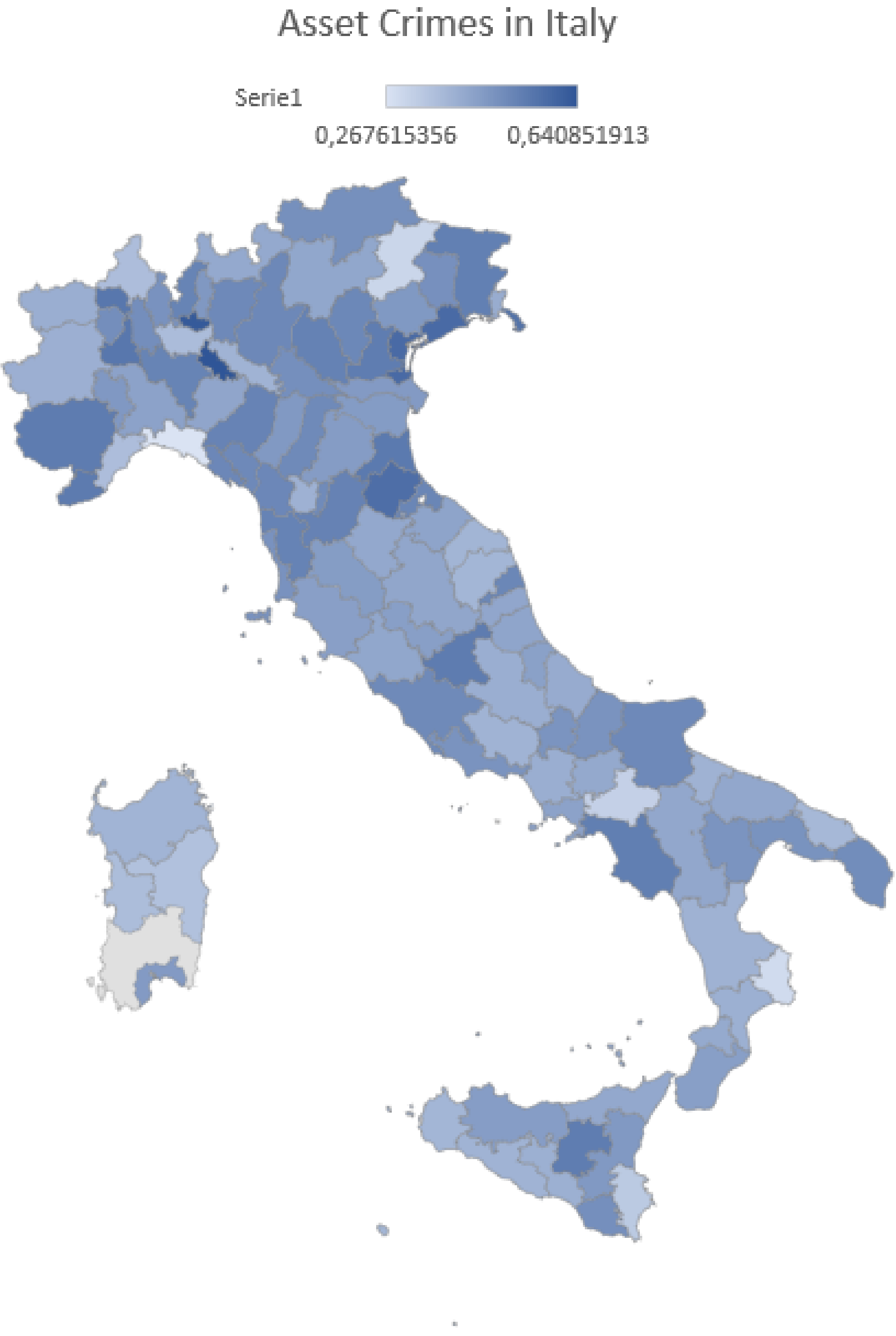


Figure 4.11: Asset crimes in Italy Map by crime rate.

4.7 Machine Learning Integration

Machine Learning (ML) is integrated into our software development process with the goal of augmenting the capabilities of our SaaS application in the realm of risk management. We adopted an Agile methodology, emphasizing flexibility and adaptability, which aligns well with the iterative nature of ML model development. We designed, developed, and trained ML models tailored to specific risk management scenarios discussed together with the team. These models were trained using the historical data in addition to user-inserted data (if available) previously listed to recognize patterns, trends, and potential risk factors. For instance, in asset risk management, ML models can forecast crime trends, aiding in transport decisions. Risk Managers can rely on these insights to formulate strategies that are informed and effective.

The design principles followed in the whole ML process were:

- **User-Centric Approach**: Machine Learning enhances the user experience by offering data-driven insights in a user-friendly format through charts, maps, and alerts. Risk Managers can easily grasp and act upon the information provided, aligning with our user-centric design philosophy.

- **Modularity**: We've designed our application to be highly modular, by providing a module for Data Loss Prevention and a module for Asset Loss and Logistics Risk Management and ML fits into this framework. Different ML models were integrated into different application modules, allowing for flexible customization based on client requirements and SaaS subscription.

- **Continuous Improvement**: The adaptive nature of ML aligns with our commitment to continuous improvement. ML models evolve and adapt by getting new data from APIs, ensuring our application remains effective in managing risks over time.

4.8 Requirement Documents and Design Artifacts

The risk management software development process was guided by detailed requirement documents and design artifacts. These included Entity-Relationship (ER) diagrams and design documents that outlined the software's functionalities and interfaces. The ER diagrams provided a visual representation of the data entities and their relationships, which helped to identify the key data elements and their attributes. The design documents described the software's architecture, modules, and interfaces, and provided guidelines for the implementation and testing of the software.

These documents played a crucial role in shaping the design and development process of the software. They helped to ensure that the software met the functional and non-functional requirements of the stakeholders, including the system administrators and end-users. The requirement documents provided a clear understanding of the software's features and functionalities, which helped to prioritize the development tasks and allocate resources effectively. The design artifacts provided a blueprint for the software's architecture and modules, which helped to ensure that the software was scalable, maintainable, and extensible.

4.8.1 Stakeholders and Interfaces

In our risk management application, various stakeholders play a role in ensuring effective risk assessment, mitigation, and overall application management. Table 4.2 provides an overview of these key stakeholders and their respective responsibilities within the application ecosystem. From Risk Managers who are responsible for identifying and assessing risks to IT and Logistics Units managing specific types of loss events, this table elucidates the diverse range of stakeholders contributing to the success of our risk management solution. Understanding their roles and responsibilities is fundamental to comprehending the holistic functionality of the application and its seamless integration into organizational processes.

Stakeholder Name	**Description**
Risk Managers	People who identify and assess the risks, measure the KRIs, start Mitigation Plans, and associate risks with projects.
Project Managers (From every Unit)	People who add the Projects and the Work Breakdown Structure and want to know the project risks.
IT Unit	People who manage Data Loss Events.
Logistics Unit	People who manage Asset Loss Events and Transits.
System Administrator	Person who manages the app.

Table 4.2: Stakeholder Roles

Understanding the interfaces between various actors and the system is vital for comprehending the application's functionality. Table 4.3 presents an overview of the interfaces utilized by actors within the system. Actors, including Users and System Administrators, interact with the application through logical interfaces, such as Graphical User Interfaces (GUIs) of the application. These logical interfaces facilitate the interaction and data input within the application. Concurrently, the physical interfaces, typically personal devices, through which these actors access the logical interfaces, are outlined. This understanding of interfaces is crucial in grasping how users and administrators engage with our risk management solution.

Actor	**Logical Interface**	**Physical Interface**
User	GUIs of the Application	Personal Device
System Administrator	Enhanced GUIs	Personal Device

Table 4.3: Actor Interfaces

4.8.2 Class Diagram

After defining Actors and Interfaces, the next step was to define a class diagram for data modeling of the structure of the application. In the diagram reported in figure 4.12, the box representing the class includes the name and attributes of the class, and lines connecting different classes represent relationships. Also, multiplicities of relationships are specified in the diagram. Since we followed DDD design principles, some of these classes were implemented as Aggregate Roots while others as Entities. In addition to the attributes shown in the picture, some entities used in the present work contain some audit information like the creation, the modification, and the deletion times, the user who created/deleted/modified an instance of the class, and the tenant identifier used for the multi-tenancy feature. Those fields have been omitted for a more understandable diagram and for readability. Let us now describe the classes used in the application, while their actual implementation will instead be covered in chapter 5.

- **Risk**: aggregate root representing a possible risk in the organization. Each Risk is identified by an Id and some specific information like Name, Description, and an optional Image. This class also contains attributes related to the risk assessment: Likelihood, Impact, Velocity, and the final Score given by the product of the three. Since the Risk is an aggregate root, it is bound to other classes such as **Cause** (causes of the risk, a risk can have many causes), **Impact** (impacts of the risk, a risk can have many impacts), **Field** (indicates possible company fields affected by the risk), **RiskCategory** (more can be selected between Strategic, Operational...), and **LossType** that are identified by an Id, Name, and Description. The LossType associated with the Risk also contains a Category which is an Enum and contains categories related to modules (Data and Asset for now).

- **KRI**: This class indicates a metric for the risk. It has an Id and other fields such as MeasurableKRI (code), Description, FailedCounter (current KRI value), LowThreshold, MediumThreshold, and HighThreshold. These last three fields are used to identify when a metric indicates a Low, Medium, or High Risk. A KRI is associated with a single risk, but a risk can be measured through many KRIs.

- **Mitigation**: A class for describing the intention of mitigating a risk. It has an Id, Name, Description, and a Goal. A Mitigation can be only for a Risk, but a Risk can have many mitigations.

- **Project, Section, WorkItem**: Those classes are the ones used for Project Management. A Project has an Id, Name, and Description. It also specifies the Business Unit in which the Project takes place and the Expected and actual Start and End Dates. A Project contains Sections that, together with standard fields, include a OrderNumber for the execution. Sections Include WorkItems (also called tasks) described by standard fields and Expected and actual Dates.

- **RBMEntry**: This class does exactly what a Risk Breakdown Matrix does, it connects a WorkItem to a Risk by performing a risk assessment specific to a single task. It is composed of an Id and a Code, together with assessment fields (Likelihood, Impact and Score) and an Enum field for deciding if the risk for that task can be Accepted or should be Mitigated.

- **RiskMitigationPlan**: This class has the goal of applying a Mitigation to a RBMEntry. It has an Id, Name, Description, Expected and Actual Dates and Expected and Actual Residual Risk after the mitigation project. It also contains a Status field (Created, Ready, In Progress, Success, Fail) and Notes.

- **Transit**: This class is part of the Asset Risk Management module describing a shipment. It's identified by an Id and some details like StartPoint, EndPoint, Expected and Actual Dates, Driver, Vehicle, and Notes. It's an aggregate root because it includes details through the **Stop** class and the **ProductType** class. The Stop class has Id, Name, Location, and Description attributes. A transit can have from zero to many stops. The same applies to ProductType containing Id, Name, Code, Description, and a TypeValue used to assess the risk based on the value of the product.

- **LossEvent**: A LossEvent is used to describe a damaging consequence of a risk. It has an Id, a Name, a Date, TotalAffected (number of lost data/products), LossAmount (economic loss), InsideOutside, Recovered (boolean), ConsumerLawsuit, ArrestProsecution, ThirdParty, ThirdPartyName, and Details. This class includes details on the **LossType** and the **Country** in which the event happened. If the LossEvent was of Assets it can also include a **Warehouse** and **ProductType** (from 0 to many). A Warehouse has an Id, Name, Address, City, PostalCode and Telephone.

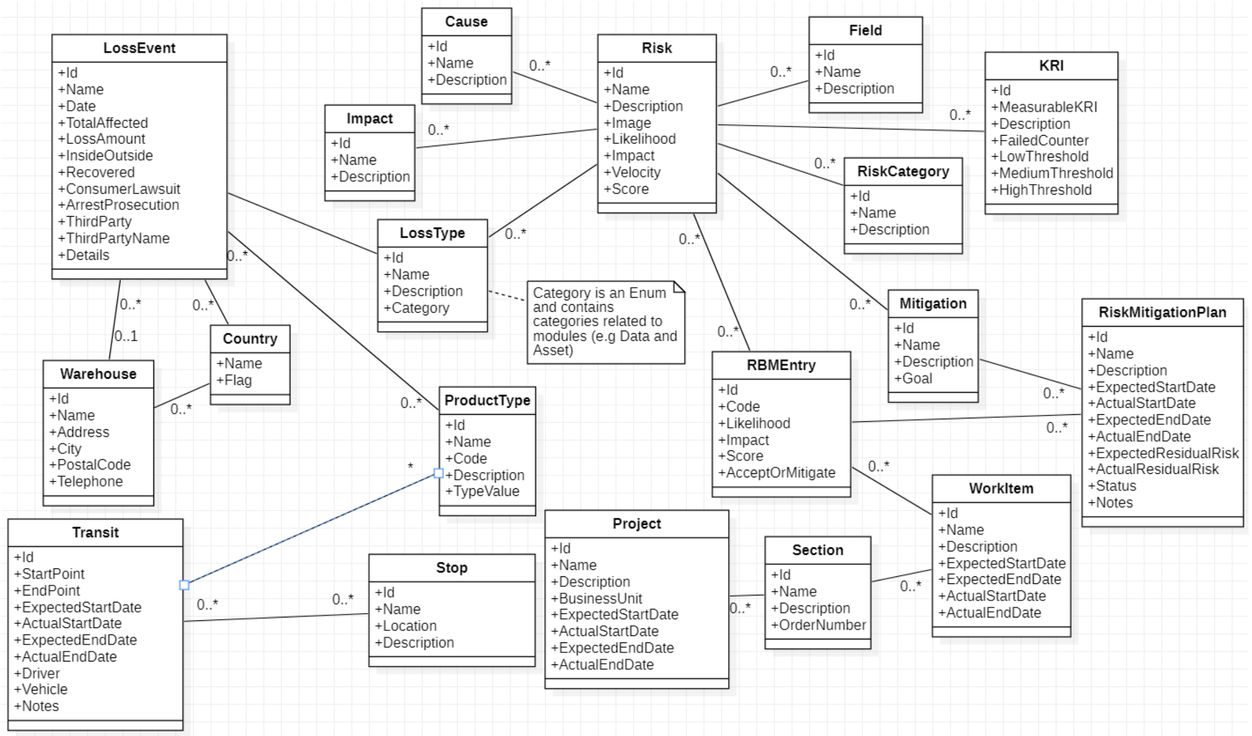


Figure 4.12: Class Diagram

4.8.3 Context and Use Case Diagrams

The Context Diagram in figure 4.13 and the Use Case Diagram in figure 4.14 respectively represent a High-Level and a Low-Level diagram of system interactions. The Users interacting with the system are Management Users (Risk and Project Managers), Logistics Unit Users, and IT Unit Users. Also, the System Administrator interacts directly with the system for configuration and development. The Use Case diagram goes into more detail on the way roles and permissions are implemented in the application. The Risk Manager has access to all those actions related to managing Risk, Mitigations, and Metrics. On the other hand, the Project Manager handles all data related to Projects, including Mitigation Plans which can be managed both by the Project and Risk Manager. All Management users can execute operations on Fields and Product Types. The Risk Manager can list

Projects but can't modify them. Loss Events are uploaded on the system by the IT and Logistics Unit, depending on the LossType, but the risk manager can list them. The Logistics Unit is also responsible for Managing Warehouses, Countries, Transits, and Stops. The System Administrator is allowed to perform all operations listed above, plus administrative and configuration operations, omitted from the diagrams. It is fundamental to consider that users can have more than one role. For example a Manager can be both a Risk Manager and a Project Manager.

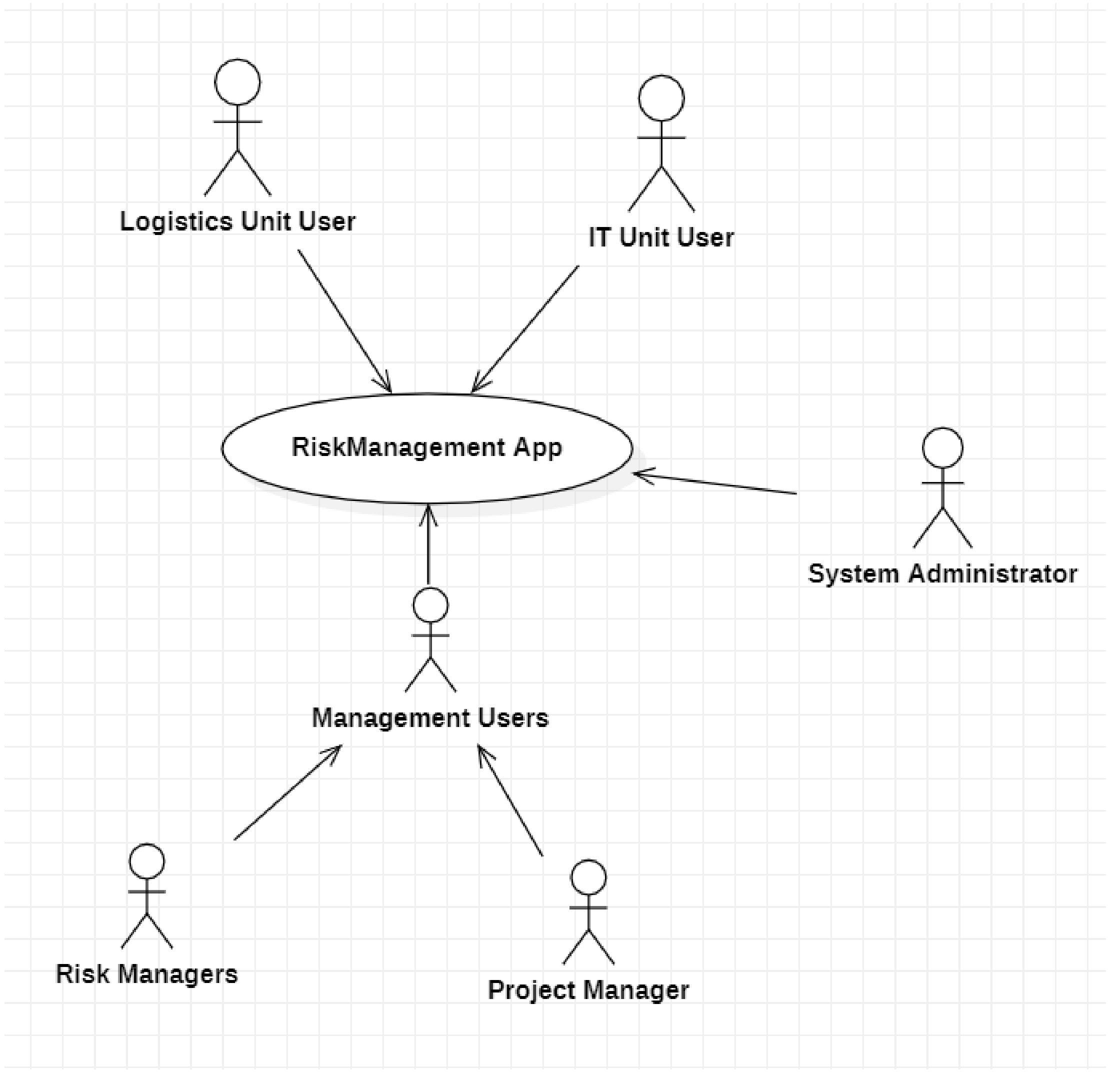


Figure 4.13: Context Diagram

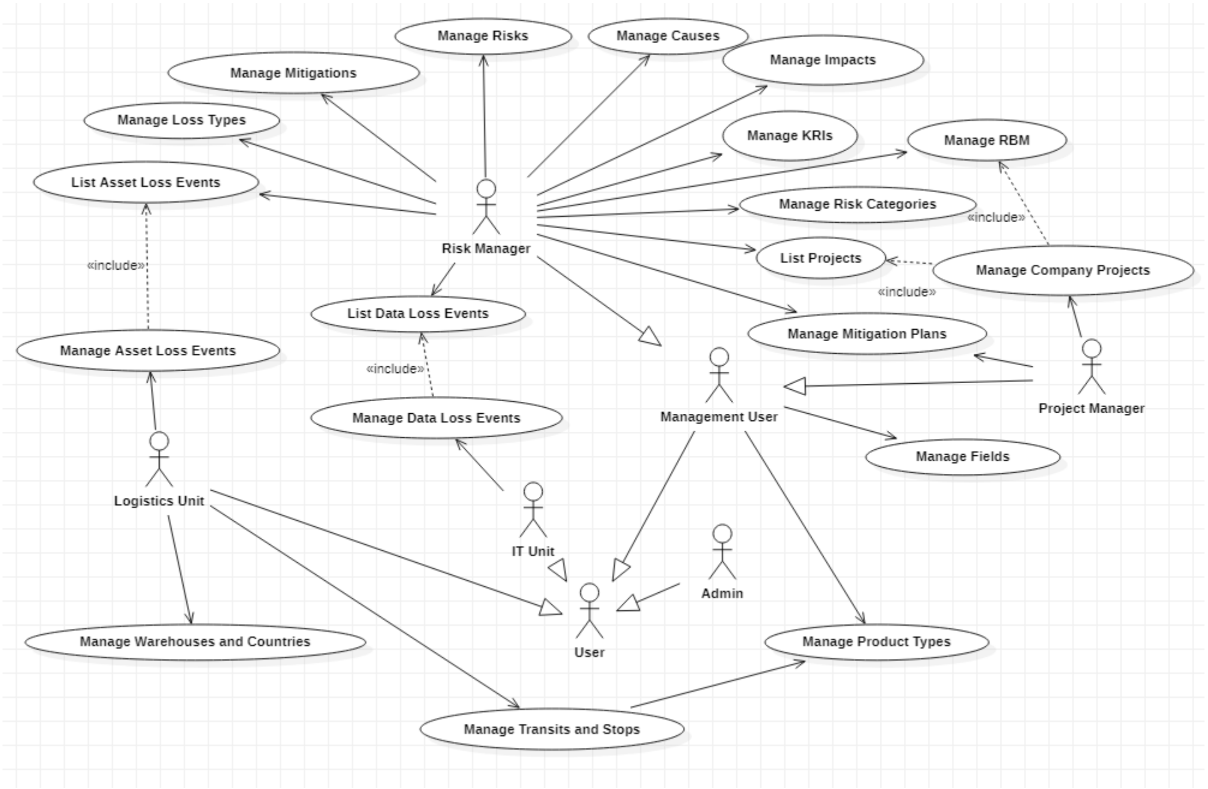


Figure 4.14: Use Case Diagram

4.8.4 Functional Requirements

In this section an extensive list of functional requirements for the Risk Management Tool will be presented. Each requirement is identified by an ID and described in detail. These requirements cover various aspects of risk management and a distinct table for each section and module was created for a better understanding and readability. As previously said, the tool is composed of a series of sections common to every customer and some additional modules depending on the subscription fee. The requirements for the sections are summarized in tables 4.4,4.5, 4.6, and 4.7 while requirements on currently implemented modules are 4.8 and 4.9. They enclose basic CRUD operations, searching, and filtering together with other operations such as statistics, predictions, chart generation, and visualization. Transit functionalities also require maps and alerts. Finally, every CRUD entity allows an Excel export to compose a formal risk report. Account and role-related functionalities were omitted, together with SaaS configuration and administration since the development of those modules was done with the help of the Commercial ABP Licence owned by the company, which provides pre-made modules and automatic code integration (more details on technologies and implementation will follow on chapter 5). The requirement document was written at the beginning of the thesis work together with the team. However, we adopted an Agile approach which lead to modifications and changes in implementation thoughout the whole development process.

ID	Description
FR1 Show Statistics on Dashboard	FR1.1 Show Risk Management Charts FR1.2 Show Project Management Charts FR1.3 Show Data Loss Charts FR 1.4 Filter statistics by Date

Table 4.4: Dashboard Functional Requirements

ID	Description
FR1 Manage Risks	FR1.1 Create, Read, Update and Delete Risks FR1.2 Search and Filter Risks FR1.3 Export Risk Register to Excel
FR2 Manage Causes	FR2.1 Create, Read, Update and Delete Causes FR2.2 Search and Filter Causes FR2.3 Export Causes to Excel
FR3 Manage Impacts	FR3.1 Create, Read, Update and Delete Impacts FR3.2 Search and Filter Impacts FR3.3 Export Impacts to Excel
FR4 Manage Risk Categories	FR4.1 Create, Read, Update and Delete Risk Categories FR 4.2 Search and Filter Risk Categories FR4.3 Export Risk Categories to Excel
FR5 Manage KRIs	FR5.1 Create, Read, Update and Delete KRIs FR5.2 Search and Filter KRIs FR5.3 Export KRIs to Excel FR5.4 Show KRIs by Risk Level FR5.5 Show KRIs on Speedometer
FR6 Manage RBM Entries	FR6.1 Create, Read, Update and Delete RBM Entries FR6.2 Search and Filter RBM Entries FR6.3 Export RBM Entries to Excel FR6.4 Predict Acceptance/Mitigation
FR7 Manage Mitigations	FR7.1 Create, Read, Update and Delete Mitigations FR7.2 Search and Filter Mitigations FR7.3 Export Mitigations to Excel

Table 4.5: Risk Functional Requirements

ID	**Description**
FR1 Manage Projects	FR1.1 Create, Read, Update and Delete Projects FR1.2 Add and Remove Sections and Work Items From Project FR1.3 Generate Gantt Diagram FR1.4 Search and filter Projects FR1.5 Export Projects to Excel
FR2 Manage Sections	FR2.1 Create, Read, Update and Delete Sections FR2.2 Search and filter Sections FR2.3 Export Sections to Excel
FR3 Manage Work Items	FR3.1 Create, Read, Update and Delete Work Items FR3.2 Search and filter Work Items FR3.3 Export Work Items to Excel
FR4 Manage Risk Mitigation Plans	FR4.1 Create, Read, Update and Delete Risk Mitigation Plans FR4.2 Search and filter Risk Mitigation Plans FR4.3 Export Risk Mitigation Plans to Excel

Table 4.6: Project Functional Requirements

ID	**Description**
FR1 Manage Loss Events	FR1.1 Create, Read, Update and Delete Loss Event FR1.2 Search and filter Loss Events FR1.3 Predict Recovered, Consumer Lawsuit and Arrest Prosecution FR1.4 Export Loss Events to Excel
FR2 Manage Loss Types	FR2.1 Create, Read, Update and Delete Loss Types FR2.2 Search and filter Loss Types FR2.3 Export to Excel

Table 4.7: Loss Events Functional Requirements

ID	**Description**
FR1 Show Data Loss Prevention Statistics	FR1.1 Show charts, trends, predictions and statistics on Data Loss Prevention FR1.2 Suggest possible mitigations
FR2 Manage Fields	FR2.1 Create, Read, Update and Delete Fields FR2.2 Search and filter Fields FR2.3 Export Fields to Excel

Table 4.8: Data Loss Prevention Functional Requirements

ID	**Description**
FR1 Show Asset Loss Prevention Statistics	FR1.1 Show charts, trends, predictions and statistics on Asset Loss Prevention FR1.2 Suggest possible mitigations
FR2 Manage Transits	FR2.1 Create, Read, Update and Delete Transits FR2.2 Search and filter Transits FR2.3 Export Transits on Excel FR2.4 Show path on a map and alert the user if the path crosses a “Dangerous Zone” FR2.5 Propose a path without “Dangerous Zones”
FR3 Manage Warehouses	FR3.1 Create, Read, Update and Delete Warehouses FR3.2 Search and filter Warehouses FR3.3 Export Transits to Excel
FR4 Manage Countries	FR4.1 Create, Read, Update and Delete Countries FR4.2 Search and filter Countries FR4.3 Export Countries to Excel
FR5 Manage Product Types	FR5.1 Create, Read, Update and Delete Product Types FR5.2 Search and filter Product Types FR5.3 Export Product Types to Excel
FR6 Manage Stops	FR6.1 Create, Read, Update and Delete Stops FR6.2 Search and filter Stops FR6.3 Export Stops to Excel

Table 4.9: Asset Loss Prevention Functional Requirements

4.8.5 Non-Functional Requirements

I will now consider Non-Functional Requirements implemented in the application.

The main requirement considered was **Usability**, the user should be able to navigate in a fast and easy way through our software. The application must adopt a clear and intuitive user interface always showing progress, errors, and success of actions performed by users. These requirements are implemented through buttons, colors, menus, popups, and other graphical instruments. The specific User Interface (UI) and User Experience (UX) implementation will be explained in chapter 5. Users should always know what is happening when using the application. In this way, the user is able to see whether the application is responding to his requests or not.

Another Non-Functional requirement we considered in the development process was **Localization**. For this purpose, localization strings for the Italian and English languages have been configured and in the future, more will be added.

Another fundamental requirement is **Portability**: our web application is responsive and works on Computers, Tablets, and Mobile devices without any problem. This requirement is fundamental in ensuring that the user can access and use the application from everywhere, without having to use a specific instrument.

Also requirements concerning **Scalability** were crucial in the development process. The Azure platform, used to deploy the application, gives the possibility to configure resources based on the user and data load.

The **Security** Non-Functional Requirement was obtained thanks to the ABP commercial modules included in the application, which will be better explained in the next chapter.

4.8.6 Deployment Diagram and System Design

Finally, the last step in Designing our application's structure was creating a Deployment diagram to visualize the hardware and the software involved in the execution and a System Design diagram to visually represent the system's architecture (figure 4.15).

The Risk Management application is deployed on a Server using Azure App Service. This means that the Web Application runs on a server without having to manually configure or maintain it. Everything is done by Azure and the developer is only responsible for the code. The User accesses the application on his/her device using an internet connection and runs the tool on a browser. This structure is a standard SaaS.

The System, on the other hand, has a simple design. It is composed of a **Backend**, implemented in C#, and a **Frontend**, implemented in Angular. In the next chapter we'll go deeper on the different modules composing the Backend structure, following Domain Driven Design (DDD) with the ABP Framework.

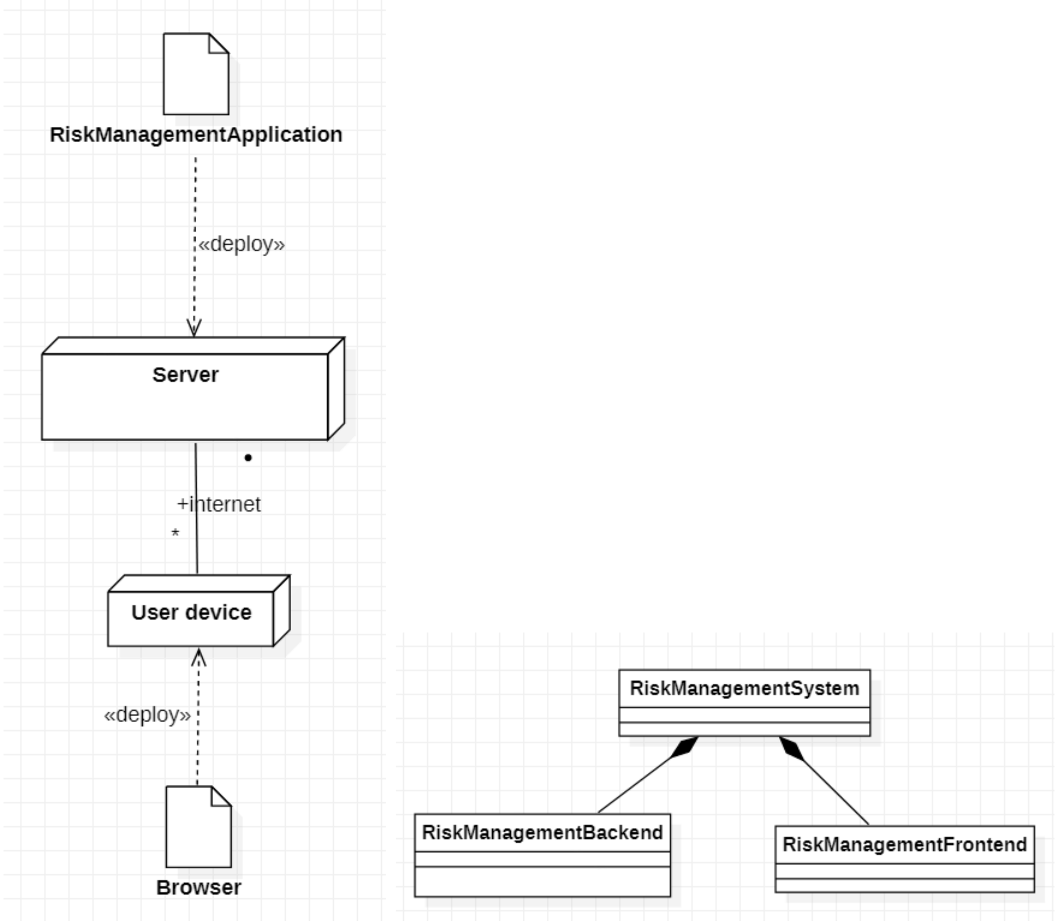


Figure 4.15: Deployment Diagram on the left and System Design on the right.

Chapter 5

Implementation Details and Technology Stack

5.1 Technology Stack

The development of the Risk Management Tool involved the integration of various technologies to create a comprehensive and user-friendly solution. This section outlines the primary technologies utilized throughout the project.

5.1.1 ABP.io Framework

The ABP.io Framework is the cornerstone of the entire Risk Management Tool, providing a comprehensive and adaptable architecture for the application's development.

The ABP Framework is characterized as an opinionated framework. It holds the belief that certain methodologies in software development inherently offer superior approaches, thereby steering developers towards those specific paths. It presents a set of preferences regarding the structure, design patterns, tools, and libraries to be employed within your solution. While the ABP Framework does maintain a degree of flexibility, allowing for the integration of different tools and libraries as well as changes to architectural choices, it is most advantageous when adhering to its established opinions. [28]

This framework is characterized by its modular approach, allowing the application to be divided into discrete modules, each with its own specific set of responsibilities and functions.

One of the primary advantages of ABP.io's modular design is its inherent scalability and maintainability. It enables developers to work on individual modules without affecting the rest of the application, making it easier to manage and update as the project evolves. This modularity is particularly beneficial for the Risk Management Tool, which encompasses various risk-related domains and functionalities.

ABP offers a wide array of application modules that can seamlessly integrate into any software application. For our software we used the Identity module, facilitating user, role, and permission management, as well as the Account module, streamlining the implementation of login and registration functionalities within the application. [28]

Moreover, the ABP.io Framework strongly embraces the principles of Domain Driven Design (DDD) for the development of SaaS applications. ABP.io's DDD-centric approach allows for the creation of a more effective and efficient system. By closely modeling the software after the problem domain, developers can ensure that the application is not only functionally robust but also conceptually aligned with the risk management challenges it tackles. This alignment makes it easier for users, such as Risk Managers, to interact with the application in a way that makes sense within the context of their work.

ABP's application startup template comes with multiple options for the UI Framework and the Database Provider. You can start with Angular, Blazor, or MVC (Razor Pages) options as the UI framework, and use Entity Framework Core (with any database management system) or MongoDB as the database provider. For the development of the Risk Management Tool, we decided to use Angular for the UI and Entity framework Core with SQL Server as a database provider.

5.1.2 C# Back-End

The back-end of the application was developed using C#, a versatile and widely adopted programming language. Entity Framework Core was employed as the database provider, enabling the application to accommodate complex business logic with a code-first approach. This approach allowed for the efficient management of data structures and relationships within the application. As shown in 5.1, the backend is composed of different application modules, each serving distinct purposes. These modules are:

- **Application**: The Application module is a critical part of an ABP.io application. It contains application-specific logic, including application services and application layer functionality. This module is responsible for orchestrating and coordinating interactions between the presentation layer (the Angular front-end) and the core business logic found in the Domain module.

- **Application.Contracts**: The Application.Contracts module complements the Application module by defining service interfaces and DTOs. These interfaces and DTOs serve as standardized communication channels between different application layers, ensuring a clean separation of concerns. It's crucial for maintaining a well-structured and maintainable codebase.

- **DbMigrator**: The DbMigrator module plays a pivotal role in database management. It is responsible for handling database migrations and ensuring that the database schema aligns with the application's current state. When changes are made to your application's data model or schema, the DbMigrator module helps you update the database seamlessly.

- **Domain**: The Domain module represents the heart of your application. It encapsulates the core business logic, data entities, and domain-specific functionalities. Here, you define your domain objects, application services, and the rules that govern your application. Changes made within this module have a direct impact on your application's behavior.

- **Domain.Shared**: The Domain.Shared module serves as a repository for shared resources that need to be accessible across various application layers. It houses common data structures, enums, and constants, ensuring uniformity and coherence throughout your application.

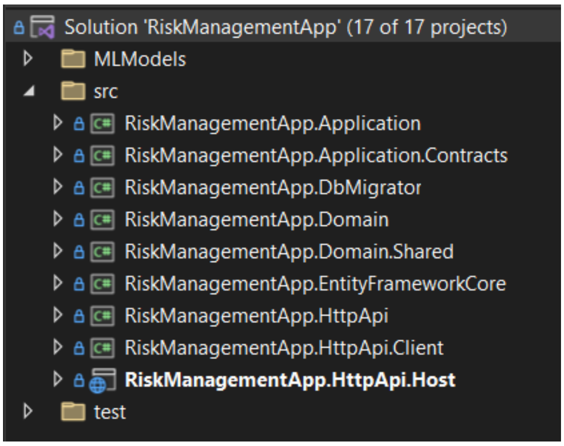
- **EntityFrameworkCore**: When using Entity Framework Core as the database provider, the EntityFrameworkCore module assists in integrating your domain entities with the underlying database. It handles tasks related to data persistence, retrieval, and the communication between your application and the database.

- **HttpApi**: The HttpApi module is responsible for generating HTTP APIs that expose your application services. These APIs are designed to be accessible to external clients, making it easier to integrate your application with different front-end technologies or other services.

- **HttpApi.Client**: On the client-side, the HttpApi.Client module provides strongly-typed HTTP clients. These clients simplify the consumption of your application's HTTP APIs. It ensures that communication between the front-end and back-end is type-safe and efficient.

- **Host**: The Host module serves as the entry point for your ABP.io application. It configures and initializes the application, including setting up the dependency injection container, configuring middleware, and starting the application. This module is responsible for hosting your application and handling its lifecycle.

The diagram in figure 5.2 shows the essential dependencies between the projects in the solution [23]. However, in our case, the Web doesn't exist in the solution. Instead, an HttpApi.Host application will be in the solution to serve the HTTP



Solution 'RiskManagementApp' (17 of 17 projects)

- MLModels
- src
  - RiskManagementApp.Application
  - RiskManagementApp.Application.Contracts
  - RiskManagementApp.DbMigrator
  - RiskManagementApp.Domain
  - RiskManagementApp.Domain.Shared
  - RiskManagementApp.EntityFrameworkCore
  - RiskManagementApp.HttpApi
  - RiskManagementApp.HttpApi.Client
  - **RiskManagementApp.HttpApi.Host**
- test

Figure 5.1: RiskManagementApp Backend Solution

APIs as a standalone endpoint to be consumed by the UI applications via HTTP API calls. The projects have been explained before. Now, we can explain the reasons for the dependencies;

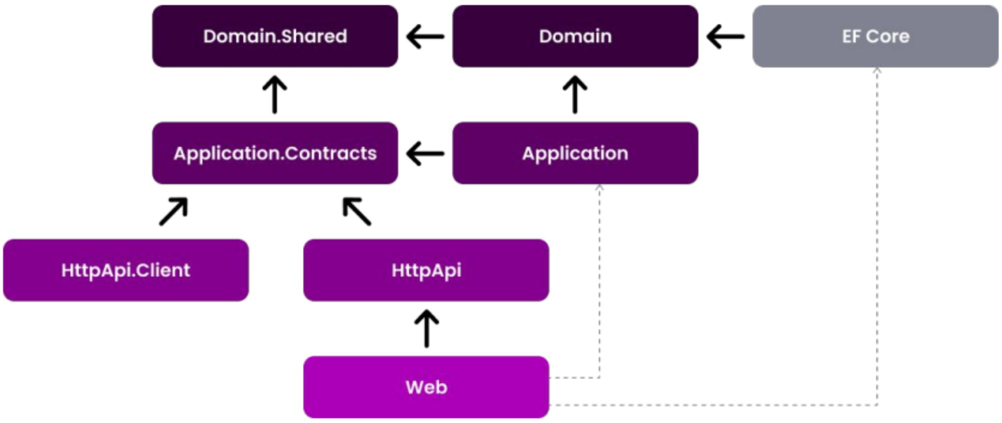


Figure 5.2: Project Dependencies

- Domain.Shared is the project that all other projects directly or indirectly depend on. So, all the types in this project are available to all projects.
- Domain only depends on the Domain.Shared because it is already a (shared) part of the domain. For example, an enum in the Domain.Shared can be used by an entity in the Domain project.

- Application.Contracts depends on the Domain.Shared. In this way, you can reuse these types in the DTOs.

- Application depends on the Application.Contracts since it implements the Application Service interfaces and uses the DTOs inside it. It also depends on the Domain since the Application Services are implemented using the Domain Objects defined inside it.

- EntityFrameworkCore depends on the Domain since it maps the Domain Objects (entities and value types) to database tables (as it is an ORM) and implements the repository interfaces defined in the Domain.

- HttpApi depends on the Application.Contracts since the Controllers inside it inject and use the Application Service interfaces as explained before.

- HttpApi.Client depends on the Application.Contracts since it can consume the Application Services as explained before.

- Host depends on the HttpApi since it serves the HTTP APIs defined inside it. Also, in this way, it indirectly depends on the Application.Contracts project to consume the Application Services in the Pages/Components.

5.1.3 Entity Framework Core

Entity Framework Core is a central component in the backend architecture of ABP.io applications, serving as the bridge between the application's object-oriented domain model and the underlying relational database. This Object-Relational Mapping (ORM) framework simplifies and streamlines database interactions, allowing developers to work with databases using familiar C# code.

One of the fundamental functions of EF Core in ABP.io is data modeling. Developers define their data models as C# classes, specifying the structure of their data entities and the relationships between them. EF Core then takes these class definitions and translates them into the corresponding database schema, adhering to the code-first approach. This approach is not only efficient but also promotes database schema consistency with the application's domain model.

A significant advantage of using EF Core in ABP.io applications is its database provider abstraction. This abstraction layer enables developers to work with various database systems without the need for extensive changes to their code. This flexibility empowers developers to select the database engine that best aligns with their project's requirements, SQL Server in our case. ABP.io leverages this feature to ensure that developers have the freedom to choose the most suitable database solution for their specific needs.

Querying data is another area where EF Core shines in ABP.io applications. It provides a powerful querying mechanism using LINQ (Language Integrated Query), which simplifies data retrieval and manipulation. This LINQ support results in code that is not only expressive but also highly readable, making it easier for developers to work with data.

Schema management and migration are crucial aspects of database development. EF Core simplifies the process by automating database schema updates as the application's data model evolves. Developers can create migrations to describe changes to the schema, and EF Core generates the corresponding SQL scripts to apply those changes as shown in 5.3. This feature is particularly useful in ABP.io applications, where the architecture encourages iterative development and schema modifications. Optimistic concurrency control is another valuable feature of EF Core used in ABP.io applications. It enables applications to handle concurrent data modifications gracefully. When multiple users attempt to modify the same data simultaneously, EF Core can detect conflicting changes and prevent data inconsistency issues. This ensures data integrity and reliability.

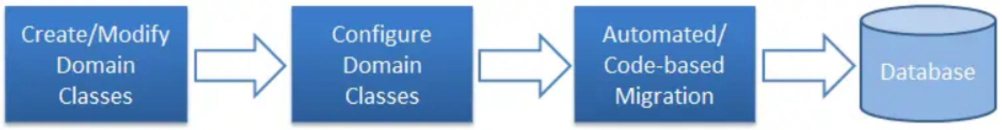


Figure 5.3: EFCore Code-First approach
[29]

Performance optimization is essential for any application, especially those dealing with large datasets. EF Core provides features like lazy loading and eager loading, allowing developers to fine-tune how data is retrieved from the database. ABP.io applications benefit from these optimization options to ensure efficient data retrieval and processing.

EF Core seamlessly integrates with the dependency injection container used in ABP.io applications. This integration simplifies the management of database contexts and promotes code maintainability and testability. It enables developers to inject database contexts into services and controllers effortlessly.

Logging and debugging are vital aspects of application development. ABP.io's integration with EF Core includes support for comprehensive logging and debugging of database operations. Developers can monitor database queries, track errors, and assess performance metrics, facilitating troubleshooting and optimization efforts.

In essence, Entity Framework Core plays a pivotal role in the backend of ABP.io applications, empowering developers to interact with databases in a cohesive and efficient manner. Its support for data modeling, querying, schema management, performance optimization, and more aligns perfectly with ABP.io's mission to simplify application development while upholding best practices in software architecture and data management.

5.1.4 Angular Front-End

The front-end of our application was constructed using Angular, a JavaScript framework known for its versatility and ability to create dynamic, responsive web applications. Angular played a pivotal role in shaping the User Interface (UI) of our application.

Angular embraces a component-based architecture, meaning the application is divided into self-contained units called components. Each component encapsulates a specific part of the UI along with its corresponding functionality. These components consist of HTML templates, TypeScript code, and CSS styles, making it easier to manage and maintain even the most intricate user interfaces.

Angular templates, the building blocks of our UI, define how the application's layout and structure should appear. These templates are crafted using HTML, but they're enriched with Angular directives, which imbue them with dynamic capabilities. This dynamism allows our templates to render data, respond to user interactions, and deliver content with grace and efficiency.

To handle business logic and data access, Angular employs services. These services act as containers for reusable functions, enabling the sharing of data and functionality across various components. They also facilitate communication with our back-end, allowing us to retrieve and send data via HTTP requests seamlessly.

Angular's dependency injection system is another invaluable asset. It facilitates modularity and testability by allowing components and services to declare their dependencies explicitly. This practice fosters loose coupling and ensures that components can be extended or replaced with ease.

Routing is a core feature in Angular. It enables the creation of single-page applications by managing navigation between different components and views while preserving the application's state. This ensures users can move smoothly between sections of our application without the need for full-page reloads.

Angular offers a suite of directives that enhance the dynamic behavior of our UI. These directives allow us to manipulate the Document Object Model (DOM) dynamically. For instance, we can use directives to conditionally display elements, iterate over lists, or toggle content visibility based on user actions.

Forms are a fundamental aspect of web applications, and Angular excels in this regard. It provides robust support for creating and validating forms, offering both Reactive Forms and Template-Driven Forms approaches. These forms empower us to collect and validate user data efficiently, guaranteeing data integrity and a seamless user experience.

While Angular doesn't prescribe a specific state management library, it accommodates various state management approaches. This includes the use of NgRx for reactive state management using RxJS, as well as services for simpler applications. Effective state management is crucial for maintaining consistent application behavior.

Lastly, Angular emphasizes responsive web design principles. This enables us to create applications that gracefully adapt to different screen sizes and devices. Leveraging CSS frameworks like Angular Material and Bootstrap, we ensure our UI components respond effectively to diverse user environments.

5.1.5 Lepton UI Theme

The User Interface (UI) plays an important role in shaping user experiences. This significance has led to the emergence of various UI themes and templates that developers can integrate into their applications. In the context of ABP.io, a robust application framework known for its flexibility and modularity, the choice of a suitable UI theme becomes a critical decision for developers aiming to create visually appealing and user-friendly applications.

One such UI theme that has garnered attention within the ABP.io community is the Lepton Theme. Unlike generic themes, the Lepton Theme is specifically designed for integration with ABP.io applications. It offers a sleek and modern appearance characterized by clean layouts, carefully chosen color schemes, and elegant typography.

The Lepton Theme prioritizes aesthetics by adhering to modern design principles. Its visually pleasing design contributes significantly to the overall look and feel of an application.

In today's digital landscape, where users access applications across various devices, responsiveness is fundamental. The Lepton Theme ensures that ABP.io applications adapt seamlessly to diverse screen sizes, from large desktop monitors to smaller smartphones.

Developers have the freedom to customize it according to their application's branding requirements. This flexibility allows for the creation of unique and visually cohesive user interfaces.

As seen in figure 5.4 the Lepton Theme includes a range of pre-designed components and widgets. These components simplify the implementation of common UI features, such as data tables, charts, and forms.

What distinguishes the Lepton Theme is its seamless integration with the ABP.io framework. Developers can incorporate the theme into their ABP.io-based applications with minimal effort, aligning it perfectly with ABP.io's modular architecture and design principles.

Beyond aesthetics, the Lepton Theme contributes to an improved User Interface. It offers a consistent and intuitive interface, enabling users to navigate through applications effortlessly. This coherence leads to higher user satisfaction and engagement.

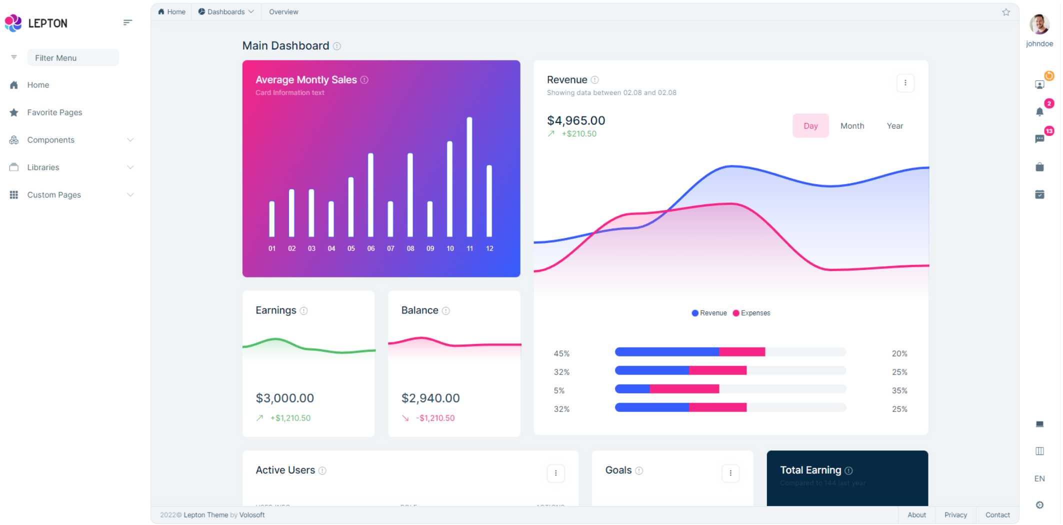


Figure 5.4: Lepton Theme

5.1.6 Chart.js Library

To enhance the user interface and provide users with visual insights, the Javascript Chart.js library was employed.

One of the primary use cases for Chart.js in ABP.io applications is the presentation of data-driven insights. These insights can include various forms of information, such as statistical data, trends, comparisons, and more. Chart.js offers a wide range of chart types, including line charts, bar charts, pie charts, and scatter plots, among others. ABP.io developers can select the most suitable chart type based on the nature of the data they wish to convey. Integration of Chart.js into ABP.io applications is accomplished in the frontend part of the application.

In the Risk Management Tool, Chart.js was used in the Dashboard and in the modules' statistics sections. An example of a chart can be seen in 5.5.

5.1.7 Leaflet for Maps

For the logistics aspect of risk management, interactive maps were crucial. Leaflet, a lightweight and versatile JavaScript library, was used to implement maps within the application. These maps offered real-time visualization of geographical data,

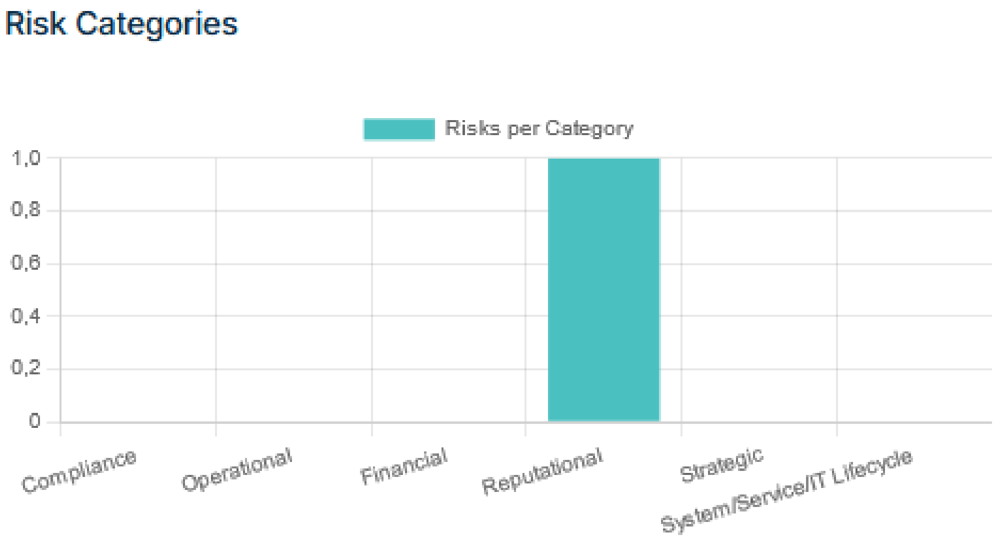
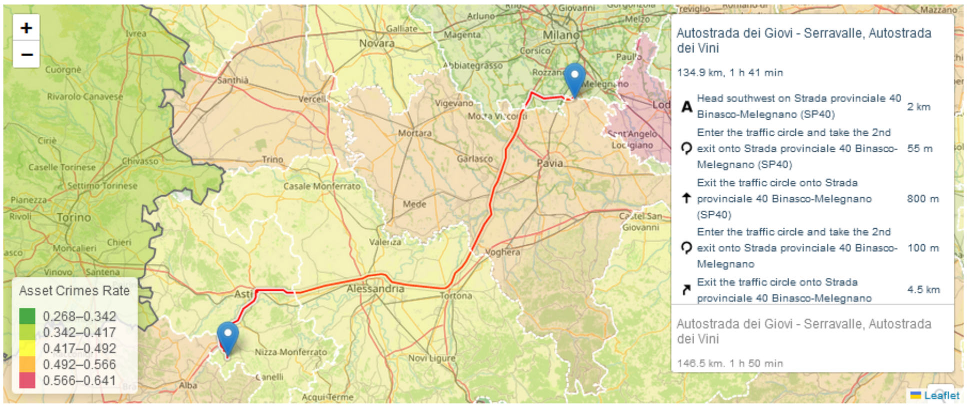


Figure 5.5: Bar Chart used to display risks by category

including transportation routes (see figure 5.6), "Dangerous Zones," and warehouses locations, enhancing logistics risk management.



Autostrada dei Giovi - Serravalle, Autostrada dei Vini

134.9 km, 1 h 41 min

A	Head southwest on Strada provinciale 40 Binasco-Melegnano (SP40)	2 km
↻	Enter the traffic circle and take the 2nd exit onto Strada provinciale 40 Binasco-Melegnano (SP40)	55 m
↑	Exit the traffic circle onto Strada provinciale 40 Binasco-Melegnano (SP40)	800 m
↻	Enter the traffic circle and take the 2nd exit onto Strada provinciale 40 Binasco-Melegnano	100 m
↗	Exit the traffic circle onto Strada provinciale 40 Binasco-Melegnano	4.5 km

Autostrada dei Giovi - Serravalle, Autostrada dei Vini

146.5 km, 1 h 50 min

Figure 5.6: Transit with risk areas shown on Leaflet Map

5.1.8 Mermaid APIs for Gantt Diagrams

Gantt diagrams play a vital role in project management and risk assessment. To enable users to create Gantt diagrams effortlessly, Mermaid APIs were leveraged. An algorithm was devised to translate project-related information, input by the user via forms, into Markdown Language. This Markdown data was then passed to the Mermaid APIs, allowing users to construct Gantt diagrams seamlessly, eliminating the need for additional manual steps. An example of a Mermaid Gantt diagram can be seen in figure 5.7 and the code to generate the Gantt starting from a ProjectDto can be seen in code snippet 5.1.8.

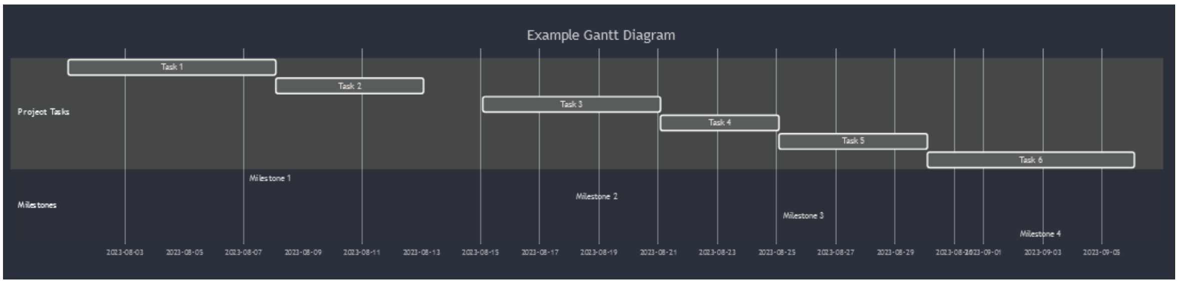


Figure 5.7: Gantt Diagram generated with Mermaid

```
generateMermaidGanttChartString(project: ProjectDto){
    var mermaidString = "gantt\n";
    mermaidString += "\ttitle " + project.name + "\n";
    mermaidString += "\tdateFormat YYYY-MM-DD\n";
    var sections = project.sections.sort((a,b)=> a.orderNumber - b
   .orderNumber);
    sections.forEach(s => {
      mermaidString += "\tsection " + s.name + "\n";
      s.workItems.sort((a,b)=> +new Date(a.expectedStartDate) - +
   new Date(b.expectedEndDate)).forEach(w => {
        mermaidString += "\t" + w.name + " : " + w.
   expectedStartDate.split('T')[0] + ", " + w.expectedEndDate.
   split('T')[0] + "\n";
      })
    });
    return mermaidString;
  }
```

Listing 5.1: Code to translate a ProjectDto to a Markdowm language for Mermaid

5.1.9 Machine Learning with ML.NET

Machine Learning (ML) capabilities were seamlessly integrated into the application using ML.NET, a powerful and versatile framework specifically designed for developing, training, and testing ML models within the C# programming language. ML.NET proved to be an invaluable addition to our software stack, simplifying the implementation of complex ML models without demanding an extensive background in Machine Learning.

This integration empowered the application to harness the full potential of ML, providing predictive and analytical insights that are instrumental in supporting critical risk management decisions.

ML.NET's compatibility with various data sources and formats allowed us to easily incorporate real-time data into our risk assessments, ensuring that our application remained responsive to dynamic, evolving risks. In section 5.3 I will go into more detail on the algorithms chosen for the software and the code to implement them.

5.2 Actual App Functionalities with Screenshots

This section provides an overview of the key functionalities within the Risk Management Tool, along with accompanying screenshots to offer a visual representation.

5.2.1 Dashboard

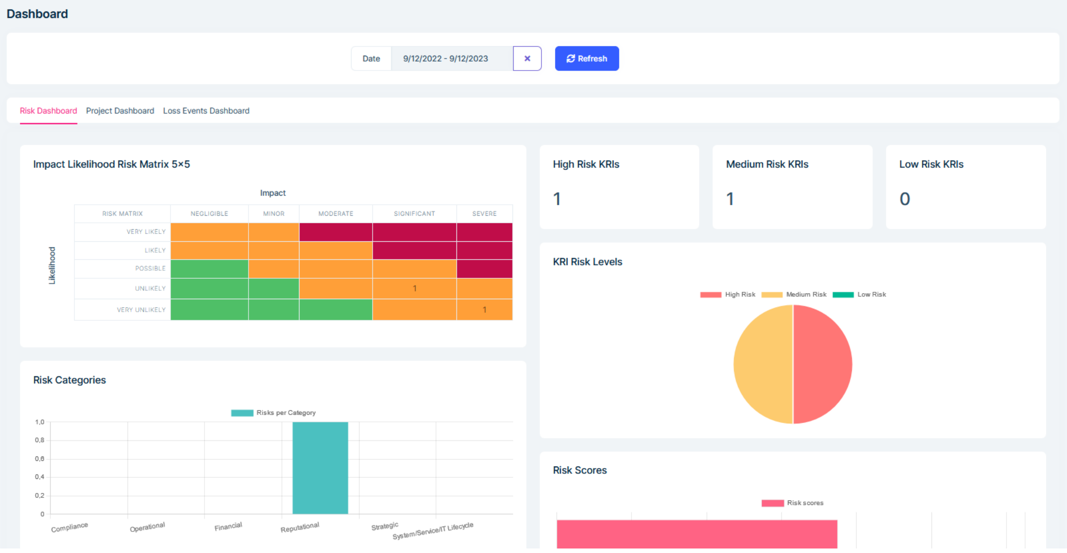
The Dashboard serves as the central hub for accessing key insights and data visualizations related to risk management. It's possible to filter insights by date.

It is divided into three tabs:

- Risk Dashboard: This tab, as shown in 5.8, presents a comprehensive view of risks registered in the system and their assessment. The cornerstone of modern risk management, the Impact-Likelihood matrix, is prominently featured here.

- Project Dashboard: In this tab, users can access charts and graphs pertaining to ongoing or upcoming projects that require assessment and risk management.

- Loss Events Dashboard: This section provides valuable information regarding the historical loss events of the company, allowing for an in-depth analysis of past incidents.



Date 9/12/2022 - 9/12/2023

Impact Likelihood Risk Matrix 5×5

Impact

RISK MATRIX	NEGLIGIBLE	MINOR	MODERATE	SIGNIFICANT	SEVERE
VERY LIKELY	🟧	🟧	🟥	🟥	🟥
LIKELY	🟧	🟧	🟧	🟥	🟥
POSSIBLE	🟩	🟧	🟧	🟧	🟥
UNLIKELY	🟩	🟩	🟧	🟧 1	🟧
VERY UNLIKELY	🟩	🟩	🟩	🟧	🟧 1

High Risk KRIs

1

Medium Risk KRIs

1

Low Risk KRIs

0

KRI Risk Levels

Risk Categories

Risk Scores

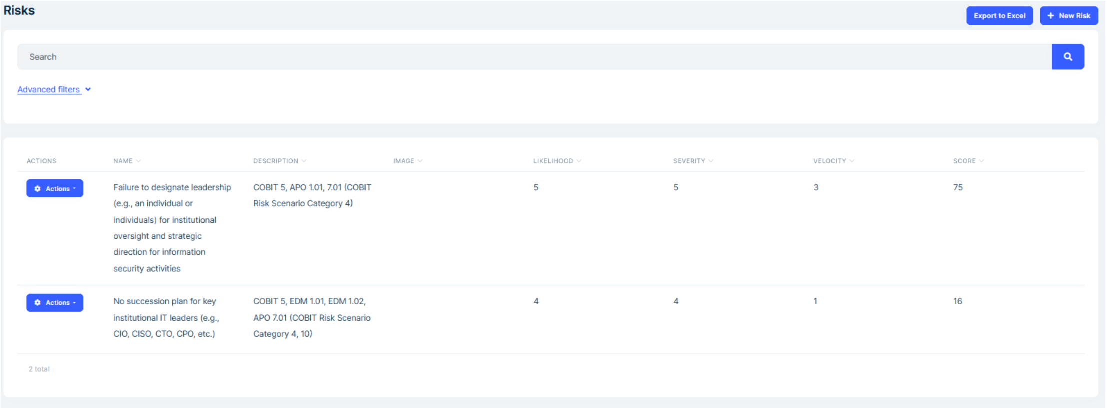
Figure 5.8: Risk Dashboard

5.2.2 Risk Management

The Risk Management section is the core of the application, enabling users to efficiently handle and evaluate risks.

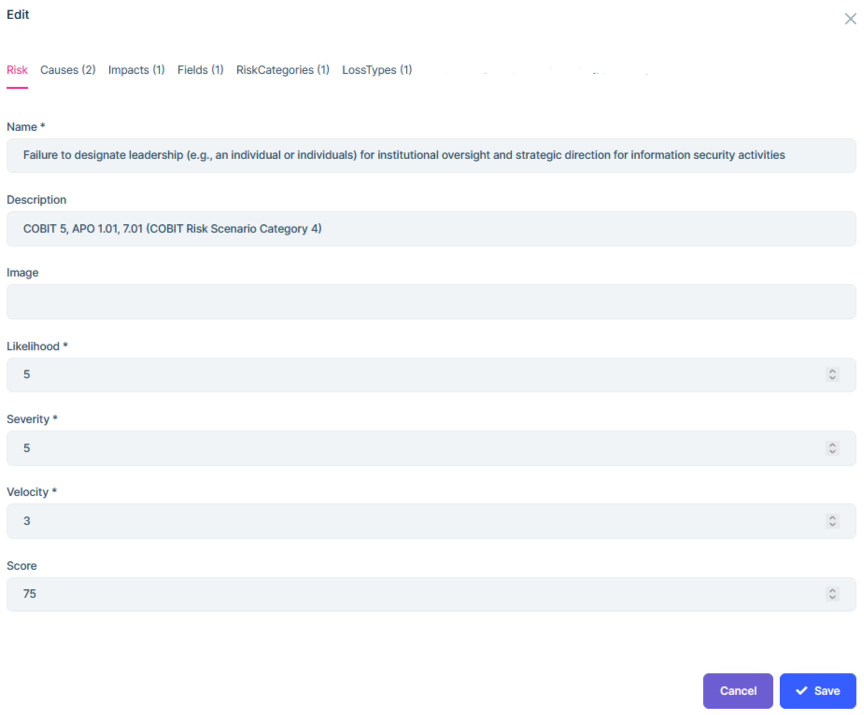
- Risk Register: This feature is where risks are recorded and managed through the assessment of Impact, Likelihood, and Velocity values (see figure 5.9). Users can also link risks to other attributes that have been added in various subsections, such as Causes, Impacts, Risk Categories, and LossTypes associated with the risk (see figure 5.10).
- Causes: Here the user can manage possible factors causing the risks.
- Impacts: Functionality to manage possible consequences of the risks.
- Risk Categories: In this subsection possible categories of risks in the Register are managed (e.g., Reputational, Financial, Operational).
- Key Risk Indicators (KRIs): This subsection allows users to manage and display Key Risk Indicators on a Speedometer (see figure 5.11), providing an intuitive visualization of risk levels.
- Mitigations: Users can add new Mitigations, which form the basis for Mitigation plans.
- Risk Breakdown Matrix (RBM) Entries: The Risk Breakdown Matrix is a powerful tool for determining the most critical risks in a project. In this

subsection, as shown in 5.13 it's possible to connect a Risk to a specific Work Item and decide whether to Accept or Mitigate it. It's also possible to modify assessment values for the specific case, instead of using general risk assessment values.



ACTIONS	NAME	DESCRIPTION	IMAGE	LIKELIHOOD	SEVERITY	VELOCITY	SCORE
Actions	Failure to designate leadership (e.g., an individual or individuals) for institutional oversight and strategic direction for information security activities	COBIT 5, APO 1.01, 7.01 (COBIT Risk Scenario Category 4)		5	5	3	75
Actions	No succession plan for key institutional IT leaders (e.g., CIO, CISO, CTO, CPO, etc.)	COBIT 5, EDM 1.01, EDM 1.02, APO 7.01 (COBIT Risk Scenario Category 4, 10)		4	4	1	16

Figure 5.9: The Risk Register



Name *
Failure to designate leadership (e.g., an individual or individuals) for institutional oversight and strategic direction for information security activities

Description
COBIT 5, APO 1.01, 7.01 (COBIT Risk Scenario Category 4)

Image

Likelihood *
5

Severity *
5

Velocity *
3

Score
75

Figure 5.10: Form for adding/editing a Risk

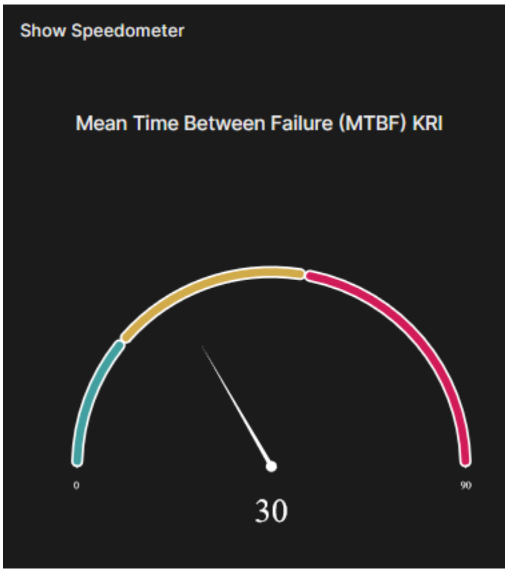
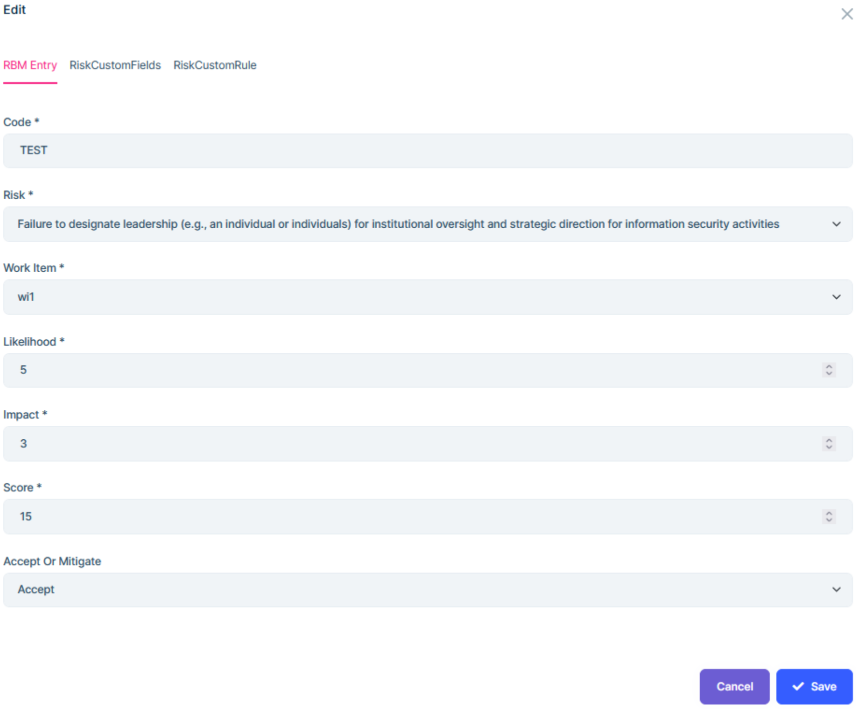


Figure 5.11: Speedometer to show a KRI


Edit

Code *
TEST

Risk *
Failure to designate leadership (e.g., an individual or individuals) for institutional oversight and strategic direction for information security activities

Work Item *
wi1

Likelihood *
5

Impact *
3

Score *
15

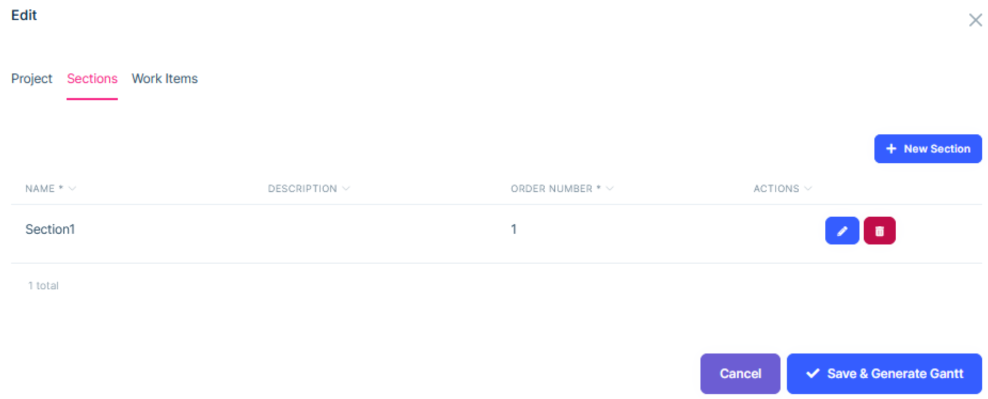
Accept Or Mitigate
Accept

Figure 5.12: Form for adding/editing a RBM Entry

5.2.3 Project Management

This section facilitates the registration and management of projects within the application, allowing for organized and efficient project oversight.

- Projects: Users can create and edit projects, which serve as containers for related Sections and Work Items. Sections and Work Items belonging to the Project can be added directly from the Project's form, as shown in figure, or separately. A Gantt diagram is automatically generated based on the project hierarchy.
- Sections: Sections represent distinct parts of a project and can be organized hierarchically.
- Work Items: These items make up the sections and are essential for project planning and execution.
- Mitigation Plans: This feature enables users to create mitigation plans, which are projects specifically designed to mitigate risks.



NAME *	DESCRIPTION	ORDER NUMBER *	ACTIONS
Section1		1	

1 total

Figure 5.13: Form for adding sections to a Project

5.2.4 Loss Events History

In this section, users can manage the company's historical loss events and categorize them by specific types.

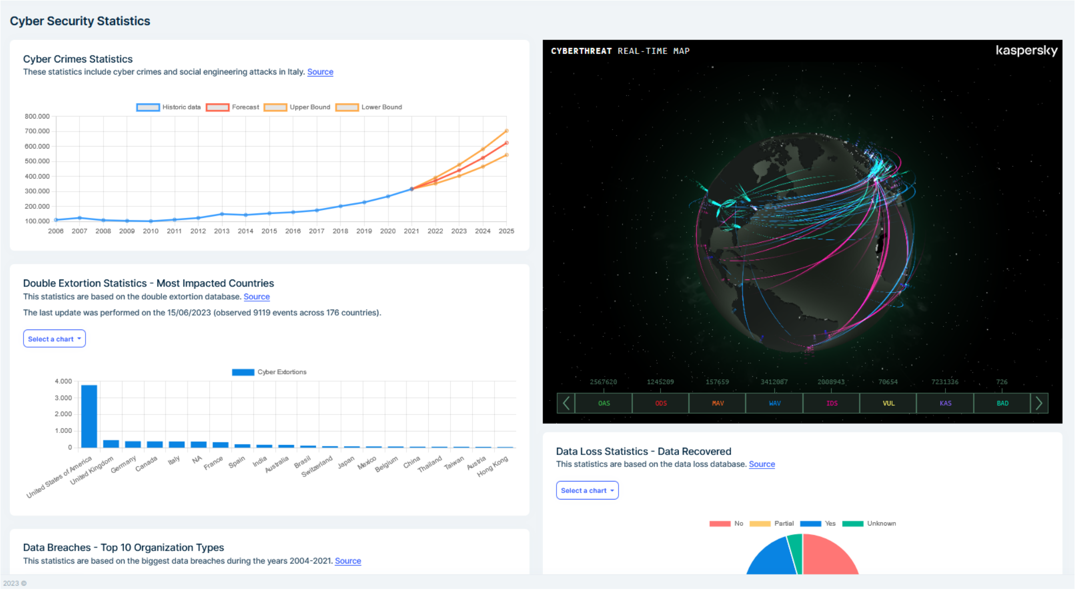
- Loss Events: Users can access and manage records of previous damaging events within the company.
- Loss Types: This subsection allows users to categorize loss events based on their specific types, such as Hacking or Theft.

5.2.5 Data Loss Prevention

The Data Loss Prevention module focuses on safeguarding sensitive information from loss, theft, or exposure to unauthorized parties.

- Cyber Security Statistics: Users can gain real-time insights into the current security landscape and make informed decisions. See figure 5.14

- Fields: This subsection enables users to manage various organizational sectors susceptible to cyber-attacks.



Cyber Security Statistics

Cyber Crimes Statistics

These statistics include cyber crimes and social engineering attacks in Italy. Source

Double Extortion Statistics - Most Impacted Countries

This statistics are based on the double extortion database. Source

The last update was performed on the 15/06/2023 (observed 9119 events across 176 countries).

Data Breaches - Top 10 Organization Types

This statistics are based on the biggest data breaches during the years 2004-2021. Source

Data Loss Statistics - Data Recovered

This statistics are based on the data loss database. Source

Figure 5.14: Cyber Security Statistics

5.2.6 Asset Loss & Logistics Risk Management

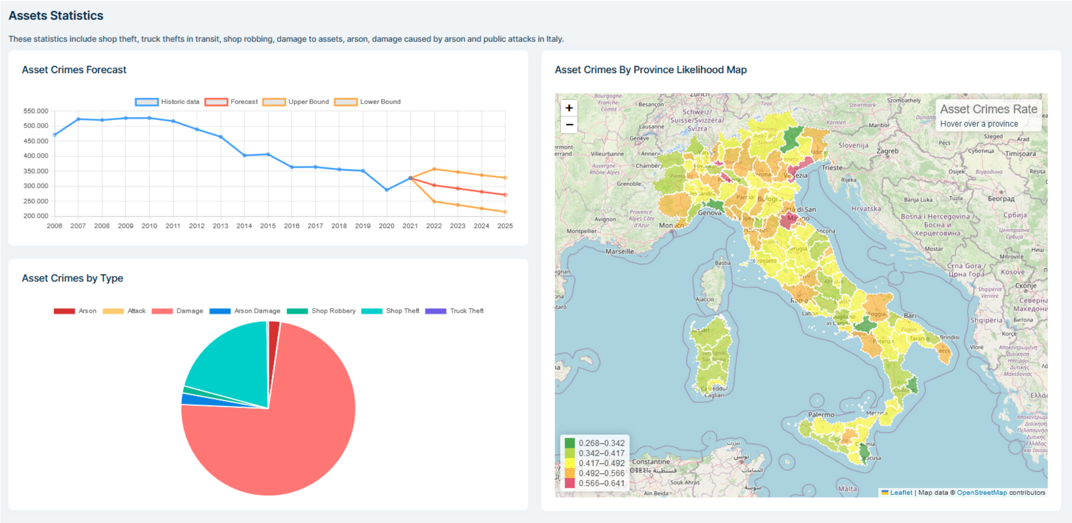
This section is dedicated to identifying, assessing, and controlling risks that may impact the transportation or storage of goods.

- Asset Statistics: Users can access up-to-date risk assessments based on thorough data analysis for each Italian province by clicking on the map (see screenshot 5.15).

- Transits and Stops: This subsection allows the management of shipments and monitoring when a transit crosses a "Dangerous Zone".

- Warehouses and Countries: Users can manage risks associated with items that are not in transit and adapt security plans based on the value of transported goods.

- Product Types: This feature facilitates the adjustment of security plans and risk management based on the type and value of transported goods.

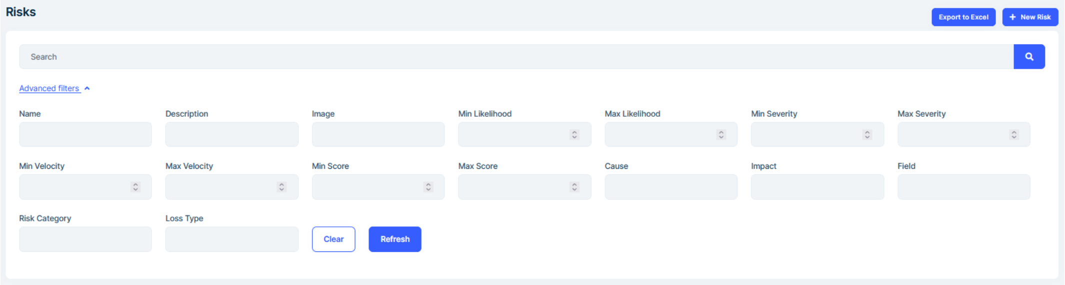


Assets Statistics

These statistics include shop theft, truck thefts in transit, shop robbing, damage to assets, arson, damage caused by arson and public attacks in Italy.

Figure 5.15: Asset Statistics

Throughout the application, as seen in 5.16, an intuitive and consistent user interface ensures ease of use, offering functionalities such as searching, filtering, and exporting data to Excel, enabling users to efficiently collect information for formal Risk Reports.



Risks

Search

Advanced filters

Name | Description | Image | Min Likelihood | Max Likelihood | Min Severity | Max Severity

Min Velocity | Max Velocity | Min Score | Max Score | Cause | Impact | Field

Risk Category | Loss Type

Figure 5.16: Additional Functionalities (Export to Excel button on the top right, searching and filtering)

5.3 Machine Learning Model Implementation

In this section, I will go into detail on how Machine Learning (ML) models have been implemented using ML.NET, by including and describing pieces of code.

5.3.1 Cyber Crimes Forecast

```
var filePath = "path were to save the model";

// Initialize the MLContext
var context = new MLContext();

// Load data from the provided source
var data = context.Data.LoadFromEnumerable(
   CyberCrimesMLForecast.GetModelInput());

// Define the forecasting pipeline
var pipeline = context.Forecasting.ForecastBySsa(
    nameof(ModelOutput.Value),
    nameof(ModelInput.OBS_VALUE),
    windowSize: 5,
    seriesLength: 10,
    trainSize: 16,
    horizon: 4,
    confidenceLowerBoundColumn: nameof(ModelOutput.Value_LB)
   ,
    confidenceUpperBoundColumn: nameof(ModelOutput.Value_UB)
   );
```

This code serves as the entry point for the program. It initializes the MLContext for ML.NET and sets the file path for the ML model. It also loads data from a source using the LoadFromEnumerable method, which calles a method that gets data from ISTAT's APIs. The algorithm selected was ForecastBySsa. The forecasting pipeline is defined, specifying input and output columns, window size, series length, train size, horizon, and confidence bounds.

- filePath: This variable holds the file path where the ML model will be saved.
- context: The MLContext object is created to manage the ML.NET operations.
- data: Data is loaded from an enumerable source using LoadFromEnumerable.
- pipeline: The forecasting pipeline is defined with various configuration parameters.

```
        // Fit the model with the data
    var model = pipeline.Fit(data);

    // Save the model to the specified file path
    context.Model.Save(model, data.Schema, filePath);
```

This section fits the ML model using the defined pipeline and the loaded data. It then saves the trained model to the specified file path.

- model: The trained ML model.
- context.Model.Save: Saves the model to the specified file path using ML.NET.

```
public static ModelOutput GetModelOutput(int horizon, string wwwRootPath)
{
    var modelOutput = Predict(wwwRootPath, horizon: horizon);
    return modelOutput;
}

public static ModelOutput Predict(string wwwRootPath, ModelInput? input = null, int? horizon = null)
{
    var PredictEngine = CreatePredictEngine(wwwRootPath);
    return PredictEngine.Predict(input, horizon);
}

private static TimeSeriesPredictionEngine<ModelInput, ModelOutput> CreatePredictEngine(string wwwRootPath)
{
    var mlContext = new MLContext();
    ITransformer mlModel = mlContext.Model.Load($"{wwwRootPath}\\wwwroot\\Models\\model-asset.zip", out var schema);
    return mlModel.CreateTimeSeriesEngine<ModelInput, ModelOutput>(mlContext);
}
```

Here, a time series prediction engine is created based on the trained model. This engine is used to make predictions on new data.

- PredictEngine: The time series prediction engine created from the trained model.
- modelOutput: Predictions are made using the Predict method of the engine.

The forecasted values, upper bounds, and lower bounds provide the predictions.

- Value: The forecasted values.
- Value_UB: Upper bounds of the forecasts.
- Value_LB: Lower bounds of the forecasts.

Accuracy

This model's Evaluation Metrics are:

```
Mean Absolute Error: 6789,203
Root Mean Squared Error: 34974,648
```

- Mean Absolute Error (MAE): MAE measures the average absolute difference between predicted and actual values. Lower MAE indicates better accuracy. An MAE of 6789.203 suggests that my model's predictions have an absolute error of approximately 6,789.203 units. In the context of this data, where the values range from 102,104 to 316,492, this error represents a relatively small percentage of the data's range.
- Root Mean Squared Error: RMSE is similar to MAE but penalizes larger errors more heavily. An RMSE of 34,974.648 means that my model's predictions have a root mean squared error of approximately 34,974.648 units. As with MAE, in the context of my data, this RMSE value appears reasonable given the data's scale.

To provide a more interpretable assessment in terms of a percentage of the data's range, we can calculate the relative errors:

- Relative MAE (%): (MAE / (316,492 - 102,104)) * 100 = (6789.203 / 214,388) * 100 ≈ 3.17%
- Relative RMSE (%): (RMSE / (316,492 - 102,104)) * 100 = (34,974.648 / 214,388) * 100 ≈ 16.31%

Interpreting these relative errors:

- The relative MAE of approximately 3.17% suggests that, on average, my model's predictions have an error of about 3.17% of the data's range. This is generally considered quite good, especially for time series forecasting.

- The relative RMSE of approximately 16.31% indicates that my model's predictions have a root mean squared error of about 16.31% of the data's range. While slightly higher, this is still within an acceptable range for many applications.

5.3.2 Asset Crimes Forecast

This prediction model is the same as the previous one, except for the input. This model offers the possibility to select different provinces as input and return the right prediction based on the model associated with that province. Models are generated una tantum for every province, and selected based on their code.

```
private static TimeSeriesPredictionEngine<ModelInput, ModelOutput> CreatePredictEngine(string provinceCode, string wwwRootPath)
{
    var mlContext = new MLContext();
    /*insert correct model name here*/
    ITransformer mlModel = mlContext.Model.Load($"{wwwRootPath}\\wwwroot\\Models\\model_" + provinceCode + ".zip", out var schema);
    return mlModel.CreateTimeSeriesEngine<ModelInput, ModelOutput>(mlContext);
}
```

As shown in the code, the CreatePredictionEngine module receives the provinceCode as an argument and selects the corresponding Model file.

Accuracy

This model's Evaluation Metrics are:

```
Mean Absolute Error: 30468,852
Root Mean Squared Error: 36542,474
```

Let's interpret the MAE and RMSE values in the context of this data:

- Mean Absolute Error: An MAE of 30,468.852 means that, on average, the model's predictions have an absolute error of approximately 30,468.852 units. In the context of this data, where the values range from 290,787 to 531,168, this error represents a moderate percentage of the data's range.

- Root Mean Squared Error: An RMSE of 36,542.474 indicates that my model's predictions have a root mean squared error of approximately 36,542.474 units.

As with MAE, in the context of this data, this RMSE value represents a moderate percentage of the data's range.

Now, let's calculate the relative errors:

- Relative Mean Absolute Error (%): (Mean Absolute Error / (531,168 - 290,787)) * 100 = (30,468.852 / 240,381) * 100 ≈ 12.69%
- Relative RMSE (%): (RMSE / (531,168 - 290,787)) * 100 = (36,542.474 / 240,381) * 100 ≈ 15.21%

Interpreting these relative errors:

- The relative Mean Absolute Error of approximately 12.69% suggests that, on average, the model's predictions have an error of about 12.% of the data's range. This error rate, while moderate, may still be acceptable for some applications.
- The relative RMSE of approximately 15.21% indicates that the model's predictions have a root mean squared error of about 15.21% of the data's range. Similar to Mean Absolute Error, this is a moderate error rate, which can be considered acceptable for many scenarios.

5.3.3 DataRecovered, ConsumerLawsuit and ArrestProsecution Classification

As previously said, classification algorithms were exploited to analyze boolean values associated with Loss Events. The classification was performed for each of these classes, but to avoid repetitions, I will use DataRecovered as an example.

```
using Microsoft.ML;
using DataRecoveredClassification;
using static Microsoft.ML.DataOperationsCatalog;
using System.Data;
using System.IO;

// Define file paths
string _appPath = Path.GetDirectoryName(Environment.GetCommandLineArgs()[0]);
string _dataPath = Path.Combine(_appPath, "..", "..", "..", "Data", "DataLossNew.CSV");
string _modelPath = Path.Combine(_appPath, "..", "..", "..", "Models", "model.zip");

// Initialize MLContext
```

```
MLContext _mlContext;
PredictionEngine<DataLoss, DataRecoveredPrediction>
   _predEngine;
ITransformer _trainedModel;
IDataView _dataView;

// Create an instance of MLContext
_mlContext = new MLContext(seed: 0);

// Load data from a text file
_dataView = _mlContext.Data.LoadFromTextFile<DataLoss>(
   _dataPath, hasHeader: false, separatorChar: ';');
```

In this section, the code initializes paths for data files and model files. It also sets up the ML.NET environment, including creating an MLContext, a prediction engine, and loading data from a text file.

- _appPath: The path to the application directory.
- _dataPath: The path to the data file.
- _modelPath: The path to the ML model file.
- _mlContext: The MLContext object is created to manage the ML.NET operations.
- _dataView: Data is loaded from a text file using LoadFromTextFile.

```
// Split data into training and testing sets
TrainTestData splitDataView = _mlContext.Data.TrainTestSplit
   (_dataView, testFraction: 0.2);

// Build the data processing pipeline
var pipeline = ProcessData();
var trainingPipeline = BuildAndTrainModel(splitDataView.
   TrainSet, pipeline);
```

This section splits the loaded data into training and testing sets. It also defines the data processing pipeline and builds a training pipeline for the ML model.

- splitDataView: Data is split into training and testing sets.
- pipeline: Data processing pipeline is created.

- trainingPipeline: Training pipeline for the ML model is constructed.

```
IEstimator<ITransformer> ProcessData()
{
    // Define the data processing pipeline
    var pipeline = _mlContext.Transforms.Conversion.MapValueToKey(inputColumnName: "DataRecovered", outputColumnName: "Label")
        // ... More transformations ...
        .AppendCacheCheckpoint(_mlContext);

    return pipeline;
}
```

This section defines the data processing pipeline, which includes transformations on the input data.

- pipeline: The data processing pipeline is constructed, featuring various transformations.

```
IEstimator<ITransformer> BuildAndTrainModel(IDataView trainingDataView, IEstimator<ITransformer> pipeline)
{
    // Build and train the ML model
    var trainingPipeline = pipeline.Append(_mlContext.MulticlassClassification.Trainers.SdcaMaximumEntropy("Label", "Features"))
        .Append(_mlContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));
    _trainedModel = trainingPipeline.Fit(trainingDataView);
    _predEngine = _mlContext.Model.CreatePredictionEngine<DataLoss, DataRecoveredPrediction>(_trainedModel);

    // Create a sample DataLoss object for prediction
    DataLoss dataLoss = new DataLoss()
    {
        // ... Sample data values ...
    };

    // Make a prediction using the trained model
    var prediction = _predEngine.Predict(dataLoss);

    // Display the prediction result
    Console.WriteLine($"=============== Single Prediction just-trained-model - Result: {prediction.DataRecovered} ===============");
    return trainingPipeline;
}
```

This section builds and trains the ML model. It also showcases how to use the trained model to make predictions.

- trainingPipeline: The training pipeline for the model is built, including the choice of the training algorithm.
- _trainedModel: The ML model is trained using the training data.
- _predEngine: A prediction engine is created for making predictions.
- dataLoss: A sample data instance is created for prediction.
- prediction: A prediction is made using the trained model, and the result is displayed.

```
void Evaluate(DataViewSchema trainingDataViewSchema)
{
    // Load test data
    var testDataView = splitDataView.TestSet;

    // Evaluate the model using test data
    var testMetrics = _mlContext.MulticlassClassification.Evaluate(_trainedModel.Transform(testDataView));

    // Display evaluation metrics
    Console.WriteLine($"*******************************");
    Console.WriteLine($"*       Metrics for Multi-class Classification model - Test Data     ");
    Console.WriteLine($"*------------------------------");
    Console.WriteLine($"*       MicroAccuracy:    {testMetrics.MicroAccuracy:0.###}");
    Console.WriteLine($"*       MacroAccuracy:    {testMetrics.MacroAccuracy:0.###}");
    Console.WriteLine($"*       LogLoss:          {testMetrics.LogLoss:#.###}");
    Console.WriteLine($"*       LogLossReduction: {testMetrics.LogLossReduction:#.###}");
    Console.WriteLine($"*******************************");

    // Save the trained model
    SaveModelAsFile(_mlContext, trainingDataViewSchema,
   _trainedModel);
}
```

This section evaluates the trained model using test data and displays evaluation metrics. It also saves the trained model to a file.

- testDataView: Test data is loaded for evaluation.
- testMetrics: Metrics for evaluating the model's performance on test data are calculated and displayed.

Accuracy

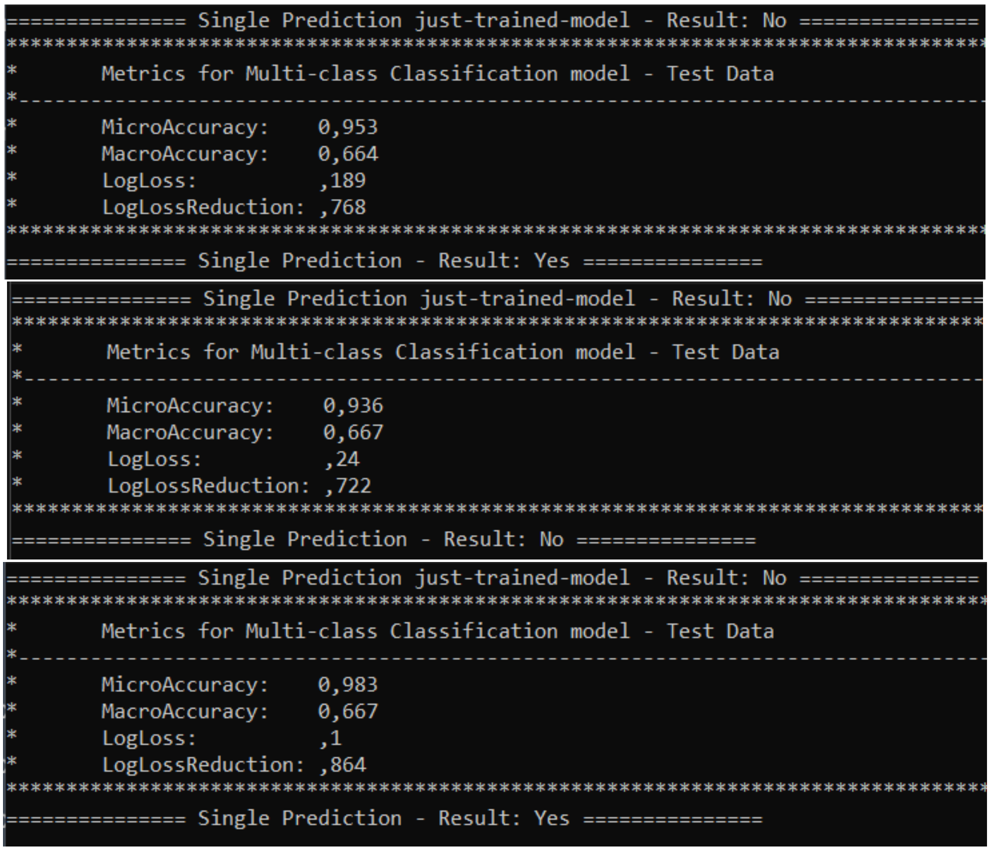
Classification models use metrics defined in table 5.1 to determine accuracies and the results obtained are shown in picture 5.17. Comparing the values with the metrics, we'll see that the accuracy obtained by classification models can be considered a good starting result.

5.3.4 Acceptance and Mitigation Classification

The Acceptance and Mitigation Classification followed the same approach as the previous subsection. The Features considered were the ones related to the Project (the WorkItem at risk) and the Risk and its assessment. Also, the specific RBM assessment was considered. However, this algorithm didn't produce accurate results, since we didn't have enough data from previously assessed Projects. This means that this algorithm will be used in practice after a number of users upload their data and perform their assessments. An alternative could be using synthetic data as an input, but this wouldn't produce results as accurate as the results obtained using actual organization's data.

Metric	Description
Micro-Accuracy	Micro-average Accuracy aggregates the contributions of all classes to compute the average metric. It is the fraction of instances predicted correctly. The micro-average does not take class membership into account, treating every sample-class pair equally. Closer to 1.00 is better, suitable for class-imbalanced datasets.
Macro-Accuracy	Macro-average Accuracy is the average accuracy at the class level. It computes accuracy for each class and then takes the average, treating every class equally. Minority classes are given equal weight as larger classes. Closer to 1.00 is better, treats all classes equally.
Log-loss	Logarithmic loss measures the performance of a classification model when predictions are probability values between 0.00 and 1.00. Log-loss increases as predicted probabilities diverge from actual labels. Closer to 0.00 is better; 0.00 represents a perfect model.
Log-Loss Reduction	Logarithmic loss reduction quantifies the advantage of the classifier over random guessing. It ranges from -inf to 1.00, with 1.00 indicating perfect predictions and 0.00 indicating mean predictions. For example, a value of 0.20 implies a 20

Table 5.1: ML.NET Classification Metrics
[30]



```
=============== Single Prediction just-trained-model - Result: No ===============
*********************************************************************************
*       Metrics for Multi-class Classification model - Test Data
*--------------------------------------------------------------------------------
*       MicroAccuracy:    0,953
*       MacroAccuracy:    0,664
*       LogLoss:          ,189
*       LogLossReduction: ,768
*********************************************************************************
=============== Single Prediction - Result: Yes ===============
=============== Single Prediction just-trained-model - Result: No ===============
*********************************************************************************
*       Metrics for Multi-class Classification model - Test Data
*--------------------------------------------------------------------------------
*       MicroAccuracy:    0,936
*       MacroAccuracy:    0,667
*       LogLoss:          ,24
*       LogLossReduction: ,722
*********************************************************************************
=============== Single Prediction - Result: No ===============
=============== Single Prediction just-trained-model - Result: No ===============
*********************************************************************************
*       Metrics for Multi-class Classification model - Test Data
*--------------------------------------------------------------------------------
*       MicroAccuracy:    0,983
*       MacroAccuracy:    0,667
*       LogLoss:          ,1
*       LogLossReduction: ,864
*********************************************************************************
=============== Single Prediction - Result: Yes ===============
```

Figure 5.17: Respectively Data Recovered, Arrest Prosecution and Consumer Lawsuit Accuracies

5.4 Implementation of Agile Testing

In this section, I'll delve into the practical implementation of Agile testing in the context of the Risk Management application.

To maintain agility and deliver software updates efficiently, we've implemented a CI/CD pipeline. The testing Implementations described below were executed at every push on the common repository. In this way, modifications to the code were followed by automated feedback. Tests were implemented at each Entity and throughout the whole development process. Following this approach, we ensured that bugs were discovered as soon as they were created and not only at the end.

5.4.1 C# Unit Testing Example

Below are practical examples of C# unit tests for the Risk Entity. These tests use the Shouldly library and Xunit framework to verify the application's functionality:

```
using System;
using System.Linq;
using Shouldly;
using System.Threading.Tasks;
using Volo.Abp.Domain.Repositories;
using Xunit;

namespace RiskManagementApp.Risks
{
    public class RisksAppServiceTests : RiskManagementAppApplicationTestBase
    {
        private readonly IRisksAppService _risksAppService;
        private readonly IRepository<Risk, Guid> _riskRepository;

        public RisksAppServiceTests()
        {
            _risksAppService = GetRequiredService<IRisksAppService>();
            _riskRepository = GetRequiredService<IRepository<Risk, Guid>>();
        }

        [Fact]
        public async Task GetListAsync()
        {
            // Act
            var result = await _risksAppService.GetListAsync(new GetRisksInput());

            // Assert
            result.TotalCount.ShouldBe(2);
            result.Items.Count.ShouldBe(2);
            result.Items.Any(x => x.Risk.Id == Guid.Parse("bdf95243-814c-47e3-8bb6-ca6e4350a13d")).ShouldBe(true);
            result.Items.Any(x => x.Risk.Id == Guid.Parse("24ce4eff-0d1e-44be-b83c-c9bd216cd915")).ShouldBe(true);
        }
        // ... (Other test methods)
    }
}
```

These tests cover critical aspects of the application, including data retrieval, record creation, updating, and deletion. By writing unit tests, we ensure that each component functions correctly and maintains data integrity.

5.4.2 Building the Test Base with Data Seeders

To ensure that the Risk Management application is thoroughly tested, a robust test base with data seeders is essential. Data seeders help populate the application's database with predefined data, allowing to execute tests against a consistent and known dataset.

Data seeder classes like the following were implemented for each application entity. Below the data seeder for the Risk Entity:

```
using System;
using System.Threading.Tasks;
using Volo.Abp.Data;
using Volo.Abp.DependencyInjection;
using Volo.Abp.Uow;
using RiskManagementApp.Risks;

namespace RiskManagementApp.Risks
{
    public class RisksDataSeedContributor : IDataSeedContributor, ISingletonDependency
    {
        private bool IsSeeded = false;
        private readonly IRiskRepository _riskRepository;
        private readonly IUnitOfWorkManager _unitOfWorkManager;

        public RisksDataSeedContributor(IRiskRepository riskRepository, IUnitOfWorkManager unitOfWorkManager)
        {
            _riskRepository = riskRepository;
            _unitOfWorkManager = unitOfWorkManager;
        }

        public async Task SeedAsync(DataSeedContext context)
        {
            if (IsSeeded)
            {
                return;
            }

            // Insert predefined risk data
            await _riskRepository.InsertAsync(new Risk
            (
                id: Guid.Parse("bdf95243-814c-47e3-8bb6-ca6e4350a13d"),
                name: "Example Risk 1",
                description: "Description for Risk 1",
                // ... (other properties)
            ));
            ...

            await _unitOfWorkManager.Current.SaveChangesAsync();

            IsSeeded = true;
        }
    }
}
```

By incorporating data seeders into the testing, it's possible to maintain a consistent and controlled environment for testing the application's functionality and data integrity.

5.4.3 Angular End-to-End (e2e) Testing Example

Angular applications also undergo rigorous testing. Here's an example of an Angular End-to-End (e2e) test using Protractor:

```
// Example Angular e2e test for the Risk Management application
import { AppPage } from "./app.po";
import { browser, logging } from "protractor";

describe("workspace-project App", () => {
  let page: AppPage;

  beforeEach(() => {
    page = new AppPage();
  });

  it("should display welcome message", () => {
    page.navigateTo();
    expect(page.getTitleText()).toEqual("RiskManagementApp app is running!");
  });

  afterEach(async () => {
    // Assert that there are no errors emitted from the browser
    const logs = await browser
      .manage()
      .logs()
      .get(logging.Type.BROWSER);
    expect(logs).not.toContain(
      jasmine.objectContaining({
        level: logging.Level.SEVERE
      } as logging.Entry)
    );
  });
});
```

This e2e test simulates user interactions with the application, validating that the user interface is responsive and that the welcome message is displayed. It also checks for any browser console errors, ensuring a smooth user experience.

Also, manual e2e tests were performed by different team members, who provided fast and detailed feedback.

The combination of C# unit tests and Angular e2e tests ensures that the application meets user expectations and maintains a high level of quality throughout its development lifecycle.

5.5 Deployment Implementation

In this section, we'll dive into how we took the Risk Management application from development to production. This is where the application becomes accessible to users.

5.5.1 Azure DevOps: Streamlining the Process

Azure DevOps became our command center for orchestrating the deployment process. Azure DevOps provided a virtual workspace for our development team. We used Git repositories to manage our source code. This helped us track changes, collaborate efficiently, and maintain code integrity.

Our Continuous Integration and Continuous Delivery (CI/CD) pipelines were the heart of our deployment process. Whenever code changes were pushed to the Git repository, these pipelines automatically kicked into action. They compiled the code, ran unit tests, and produced deployment-ready artifacts.

Azure DevOps simplified the deployment process with its release pipelines. These pipelines were like a set of instructions for deploying our application. They took care of provisioning infrastructure, deploying code, and configuring environment-specific settings.

We created distinct environments, such as development and production, within Azure DevOps. This separation allowed us to test changes in isolated environments before deploying them to production, reducing the risk of unexpected issues.

Azure DevOps wasn't just about deploying; it was also about monitoring. We integrated application monitoring tools, which collected data on how the application was performing in real-time. This data was invaluable for identifying and addressing issues quickly.

5.5.2 Terraform: Building Infrastructure as Code

Terraform played a crucial role in ensuring that our infrastructure was provisioned consistently and predictably. With Terraform, we defined our infrastructure requirements as code. This included virtual machines, databases, networking components, and security policies. These configurations became Terraform scripts.

Our Terraform scripts were version-controlled, just like our application code. This allowed us to track changes to the infrastructure over time. Before applying any changes, we could review and test them thoroughly.

Terraform made it easy to create and manage environments consistently. We could spin up development, staging, and production environments with identical configurations.

When we needed to scale our infrastructure up or down, Terraform made it straightforward. It also ensured that our infrastructure configurations were reproducible, minimizing configuration drift.

5.5.3 Deployment Workflow in Action

Here's how the deployment process played out in practice:

1. **Source Code Integration**: Developers pushed code changes to our Git repository hosted in Azure DevOps. This triggered our CI pipeline, which automatically built the application and generated deployment-ready artifacts.

2. **Infrastructure Provisioning**: Our Terraform scripts, kept in a separate repository, defined what infrastructure we needed. Whenever we made code changes or infrastructure updates, Terraform handled the provisioning or updating of infrastructure components.

3. **Deployment Automation**: Azure DevOps release pipelines took those built artifacts and deployed them to our chosen environments. This included deploying application code, configuring database connections, and setting environment-specific variables.

4. **Testing and Validation**: Within the deployment pipeline, we executed automated tests, including unit tests and end-to-end tests, to validate the application's functionality. Any test failures triggered alerts for immediate attention.

5. **Monitoring and Feedback**: After deployment, we kept a close eye on the application's performance. We collected user feedback and monitored error rates and user interactions. This continuous feedback loop helped us identify issues and prioritize improvements.

By combining Azure DevOps for streamlined orchestration and Terraform for consistent infrastructure provisioning, we ensured that changes to the Risk Management application could be rapidly deployed to production while maintaining the application's reliability and stability.

Chapter 6

Evaluation and Testing

In this chapter, we comprehensively evaluate the Risk Management application's performance, usability, and adherence to the defined requirements. The evaluation process involved developers and a Risk Management expert, who collectively assessed the application. This chapter presents the methodology used for evaluation, the real-world testing scenarios, the results and findings, and a comparison of the application's performance against the specified requirements.

6.1 Evaluation Methodology

The evaluation of the Risk Management application followed a structured methodology to ensure a thorough assessment. Before starting the evaluation, the team identified key components to be tested and evaluated. This methodology included the following key components:

1. **Usability Assessment**: Users interacted with the application to evaluate its user-friendliness, intuitiveness, and overall user experience. Feedback was collected to identify areas of improvement.

2. **Scalability Testing**: The application underwent scalability testing to assess its ability to handle increasing workloads. This involved simulating scenarios with growing data and user loads to measure performance under stress.

3. **Customization Evaluation**: We examined the application's customization capabilities to determine its adaptability to specific business needs. Feedback from stakeholders was collected to identify areas for further customization.

4. **Requirements Compliance**: A thorough review of the defined requirements was conducted to ensure that the application met the specified criteria. Any deviations or gaps were noted for future enhancement.

5. **Machine Learning Model Accessibility**: The accessibility of machine learning models through the user interface was verified. Any models not integrated were identified for inclusion.
6. **Risk Report Generation**: The generation of risk reports was tested to assess the application's ability to provide comprehensive risk insights. Feedback was gathered to enhance report generation capabilities.
7. **Multi-Tenancy Assessment**: The multi-tenancy feature was assessed. As it was initially implemented as a Proof of Concept (PoC), its readiness for production and customer use was examined.

The evaluation process combined both quantitative and qualitative data, providing valuable insights into the application's performance and areas for improvement.

6.2 Testing in Real-World Scenarios

Real-world testing scenarios were designed to simulate actual usage conditions and challenges. These scenarios included:

1. **Data Volume Increase**: The application was tested with an increasing volume of historical and evaluation data to assess its response time and resource utilization as data scaled.
2. **User Load Testing**: Simulated user loads were applied to determine how the application performed under various levels of concurrent users. This helped identify potential bottlenecks. This also helped us to configure a correct Azure configuration for the deployed software.
3. **Customization Requests**: Customization requests from potential customers were evaluated to understand their unique needs and requirements. These scenarios provided insights into the application's adaptability.
4. **Machine Learning Model Integration**: Pending machine learning models were evaluated for integration into the application to verify their accessibility and functionality via the user interface. The theoretical analysis of the Acceptance/Mitigation classification was performed. Challenges and requirements for its practical implementation were identified. The integration of loss events classification into the application was evaluated. Any gaps or missing features were documented.
5. **Risk Reporting**: Risk reporting features were used to generate comprehensive risk reports for real-world data. Feedback from users helped enhance report generation.

6. **Multi-Tenancy Verification**: The multi-tenancy feature was verified for its functionality and reliability as part of a potential customer engagement.

These real-world scenarios allowed us to identify strengths and weaknesses, ensuring that the application was well-prepared for practical use.

6.3 Results, Findings, and Comparison with Requirements

The evaluation process yielded several key results and findings. Users found the application to be user-friendly and intuitive, providing a seamless experience for risk management tasks. The application demonstrated impressive scalability with the right deployment configuration, handling increased data volumes and user loads without significant performance degradation. However, stakeholders expressed the need for further customization to align the application with specific business requirements.

While many Machine Learning models were accessible via the user interface, some were yet to be integrated, and this integration was identified as a priority. The theoretical analysis of the Acceptance/Mitigation classification identified the need for additional data and refinement before practical implementation. Integration of loss events classification was deemed essential and marked for immediate implementation.

The application successfully generated comprehensive risk reports, but user feedback suggested enhancements to report formatting and customization.

The multi-tenancy feature, initially a Proof of Concept (PoC), requires further development to fulfill requirements for production and customer use.

A potential customer expressed interest in the product but with specific customization requirements, opening opportunities for collaboration.

A critical aspect of the evaluation was comparing the application's performance with the defined functional and non-functional requirements. Additional customization is required to fully align the application with specific business requirements. Despite these gaps, the application's usability, scalability, portability, and overall performance aligned with or exceeded the defined non-functional requirements. Furthermore, the expression of interest from a potential customer signifies the application's potential value in the market.

Chapter 7

Conclusions and Future Directions

The journey through this thesis had the goal of reaching a comprehensive understanding of risk management, leading to the development of a Software as a Service (SaaS) application that encapsulates the essential elements of modern risk management systems. In this concluding chapter, I will describe the significant achievements, and the challenges faced, and explore possible future developments.

Our study commenced with a deep theoretical study of risk management, laying the foundation for the creation of a dynamic SaaS platform. This platform includes all core components and features crucial for effective risk management, offering a robust and versatile tool that provides a modern solution without taking lightly the basis of the subject.

The adoption of Agile methodology and rigorous testing practices played a great role in our success. Agile principles empowered our development process, ensuring compliance with defined requirements. The software quality was supported by unit and integration tests, together with end-to-end testing for the front end.

In the realm of software architecture, Domain Driven Design (DDD) principles guided our approach, resulting in a well-structured and maintainable codebase. DDD allowed a development process founded on the Core subject of this thesis: Risk Management. Our choice of technology stack, ABP.io with C# for the backend and EFCore, alongside Angular for the front-end, enhanced the application's reliability and scalability and gave us the possibility to obtain a high level of security through authorization and authentication pre-made modules.

The User Interface (UI) is a highlight of the tool, embellished with interactive charts, maps, and an intuitive dashboard, also thanks to the Lepton Theme. These features not only enhance User Experience (UX) but also provide valuable insights for risk assessment.

Machine Learning, powered by ML.NET, brought predictive capabilities to the application. Our ML.NET models demonstrated commendable accuracy, except for one notable challenge: the algorithm to predict Acceptance or Mitigation for Risk Breakdown Matrix (RBM) Entries. This challenge is rooted in the scarcity of historical project assessment data. However, we intend to overcome this problem in the future by harnessing actual usage data or synthetic data.

Loss events classification models, while implemented, await intuitive integration into the UI to maximize accessibility.

The deployment was performed through Azure DevOps, orchestrated by Continuous Integration and Continuous Delivery (CI/CD) pipelines, and facilitated by Terraform for infrastructure provisioning. This orchestrated approach streamlined the process, ensuring consistent and reliable deployment across different environments.

The application's robustness was subjected to evaluation by a Risk Management Expert. Valuable feedback was received, highlighting the need for improved reporting. Presently, the application allows for exporting data in Excel tables for individual entities, but there is a clear demand for a full PDF risk report generation and customization to fulfill diverse customer needs.

The potential of our SaaS application was further recognized when it attracted the interest of a possible buyer, affirming its relevance and power on the market.

While we celebrate our accomplishments, it's essential to acknowledge our limitations. Acquiring historical data was a notable challenge, and although open-source APIs and databases were employed, the key to enhanced results requires us to use real software usage data.

Our study does not end here, it was just the beginning of this software lifecycle. Future directions include integrating network monitoring tools into the data loss prevention module with the aim of fortifying our application against potential threats and attacks. Moreover, we intend to expand the application's capabilities to diverse company structures and requirements, enhancing its versatility and customization.

Additionally, this thesis will be joined with a preceding student's work in Orbyta that consisted of web scraping through Google searches. This blending holds the promise of an integrated Credit Risk Management module to be used during the Due Diligence of potential partners or customers. A structured result based on many Google entries could significantly speed up the investigation phase performed before collaborations occur, reducing the risk of failures.

In closing, this thesis has been a journey of discovery, innovation, and challenges.

Bibliography

[1] Stanford University. *Risk*. Stanford - Office of the Chief Risk Officer, 2019. URL: https://ocro.stanford.edu/enterprise-risk-management-erm/key-definitions/definition-risk (cit. on p. 11).

[2] Antonino Trapani. *Project risk management frameworks analysis and contingency evaluation criteria*. 2019-2020 (cit. on p. 11).

[3] *ISO 31000:2018 - Risk management – Guidelines*. International Organization for Standardization, 2018. URL: https://www.iso.org/standard/65694.html (cit. on p. 12).

[4] Committee of Sponsoring Organizations of the Treadway Commission (COSO). *Enterprise Risk Management - Integrated Framework*. COSO. 2004. URL: https://www.coso.org/Documents/990025P_ERM_ExecutiveSummary.pdf (cit. on p. 12).

[5] Linda Tucci. «What is risk management and why is it important?» In: *Tech Target* (2021). URL: https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important (cit. on pp. 12, 13).

[6] Kirk Patrick Price. «The 5 Components of Risk Management». In: *Kirkpatrick-Price* (2021). URL: https://kirkpatrickprice.com/blog/5-components-risk-management/ (cit. on p. 14).

[7] EDUCAUSE. «IT Risk Register». In: (2015). URL: https://library.educause.edu/resources/2015/10/it-risk-register (cit. on p. 17).

[8] Patricia Guevara. «A Guide to Understanding 5x5 Risk Matrix». In: (2023). URL: https://safetyculture.com/topics/risk-assessment/5x5-risk-matrix/ (cit. on p. 16).

[9] David Hillson, Sabrina Grimaldi, and C. Rafele. «Managing Project Risks Using a Cross Risk Breakdown Matrix». In: *Risk Management* 8 (Mar. 2006), pp. 61–76. DOI: 10.1057/palgrave.rm.8250004 (cit. on p. 18).

[10] *Archer*. https://www.archerirm.com/. Year: 2023 (cit. on pp. 19, 20).

[11] Beverly Park Woolf. «Chapter 7 - Machine Learning». In: *Building Intelligent Interactive Tutors*. Ed. by Beverly Park Woolf. San Francisco: Morgan Kaufmann, 2009, pp. 221–297. ISBN: 978-0-12-373594-2. DOI: https://doi.org/10.1016/B978-0-12-373594-2.00007-1. URL: https://www.sciencedirect.com/science/article/pii/B9780123735942000071 (cit. on p. 21).

[12] Istituto Nazionale di Statistica. *Delitti denunciati dalle forze di polizia all'autorità giudiziaria*. http://dati.istat.it/Index.aspx?DataSetCode=dccv_delittips. 2021 (cit. on pp. 22, 40).

[13] «Classification Algorithm in Machine Learning». In: *javaTpoint* (2021). URL: https://www.javatpoint.com/classification-algorithm-in-machine-learning (cit. on p. 23).

[14] Brian Turner. «What is SaaS? Everything you need to know about Software as a Service». In: *TechRadar* (2020). URL: https://www.techradar.com/news/what-is-saas (cit. on p. 26).

[15] Kathleen Casey Wesley Wesley. «What is SaaS (Software as a Service)? Everything You Need to Know». In: *TechTarget* (2022). URL: https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service (cit. on pp. 26, 27).

[16] Martin Fowler Kent Beck Robert C. Martin. «The Agile Manifesto». In: *agilemanifesto.org* (2001). URL: http://agilemanifesto.org/iso/en/manifesto.html (cit. on p. 28).

[17] André Janus. «Towards a Common Agile Software Development Model (ASDM)». In: *SIGSOFT Softw. Eng. Notes* 37.4 (July 2012), pp. 1–8. ISSN: 0163-5948. DOI: 10.1145/2237796.2237803. URL: https://doi.org/10.1145/2237796.2237803 (cit. on p. 29).

[18] Jeff Sutherland Ken Schwaber. «The Scrum Guide». In: *Scrum.org* (2020). URL: https://www.scrum.org/resources/scrum-guide (cit. on p. 29).

[19] Gianluca Tramontana. «Quali differenze tra metodologia Agile, CI/CD e DevOps». In: *gianlucaTramontana* (2020). URL: https://www.gianlucatramontana.it/2020/01/14/quali-differenze-tra-metodologia-agile-ci-cd-e-devops/ (cit. on p. 34).

[20] *What is CI/CD?* https://about.gitlab.com/topics/ci-cd/ (cit. on p. 36).

[21] Robert Martin. *Design Principles and Design Patterns*. Prentice Hall, 2000 (cit. on p. 36).

[22] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software.* Addison-Wesley Professional, 2003. ISBN: 978-0321125217 (cit. on p. 37).

[23] Halil İbrahim Kalkan. *Implementing Domain Driven Design.* Packt Publishing, 2013. ISBN: 978-1782160038 (cit. on pp. 37, 38, 64).

[24] M.Eng. Luca Mella. *Double Extortion: Cyber Extortion Attack Breach Tracker.* [Online; Thu Sep 07 2023]. 2023. URL: `https://doubleextortion.com/` (cit. on pp. 40, 41).

[25] *Data Loss Archive and Database (DLDOS).* 2007. URL: `https://attrition.org/dataloss/` (cit. on pp. 40, 41).

[26] Hishaam Ahmed. *List of Top Data Breaches (2004 - 2021).* 2021. URL: `https://www.kaggle.com/datasets/hishaamarmghan/list-of-top-data-breaches-2004-2021` (cit. on pp. 40, 41).

[27] nData. *Guida all'uso delle API REST di ISTAT.* 2022. URL: `https://ondata.github.io/guida-api-istat/` (cit. on p. 40).

[28] Halil Ibrahim Kalkan. *Mastering ABP Framework.* Packt Publishing, 2022. ISBN: 978-1801079242 (cit. on pp. 62, 63).

[29] Entity Framework Tutorial. *What is Code-First?* `https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx` (cit. on p. 67).

[30] Microsoft ML.NET Documentation. *Evaluate your ML.NET model with metrics.* URL: `https://learn.microsoft.com/en-us/dotnet/machine-learning/resources/metrics` (cit. on p. 89).

Acknowledgements

I would like to express my deepest gratitude to my family for their support and encouragement.

To my sister Orsola, your presence has been a constant source of motivation.

To Antonio, your wisdom and precious advice have been my greatest source of strength.

I extend my heartfelt thanks to my friends, both near and far, for moments of laughter that provided much-needed breaks from my academic pursuits.

To my supervisor, Professor Maurizio Morisio, for giving me the opportunity to conclude my university path with challenging and engaging work.

To my company tutor Angelo Nestola and my mentor Xhoi Kerbizi, I am grateful for your support and for providing me with the opportunity to apply my academic knowledge to real-world challenges. Your guidance has been instrumental in shaping my professional growth.

I would also like to thank all my colleagues at the company where I conducted my thesis research. Your insights, collaboration, and shared dedication to our projects enriched my learning experience.

To my study group of close friends, "I Duchi Degli Abruzzi", a small community always changing and growing. Thank you for the study sessions and fun times. Your friendship has been a light that guided these five years.

This thesis would not have been possible without the contributions and support of each of you.

Thank you for being a part of this journey.