POLITECNICO DI TORINO

Master degree in Computer Engineering

Master Degree Thesis

IBD Tool: development of a web application for the monitoring of patients with Inflammatory Bowel Diseases

Supervisors	**Candidate**
prof. Carla Fabiana Chiasserini	Flavio Martinez De Carnero Calzada
prof. Guido Pagana	

April 2022

Abstract

IBD tool is a project born from the collaboration of the Mauriziano Hospital in Turin and the Links Foundation, with the aim of developing a web application to support the monitoring of patients affected by IBD(inflammatory bowel disease).

IBDs are chronic diseases that affect the gastrointestinal tract causing, among other symptoms, abdominal pain, diarrhea, rectal bleeding and weight loss. The cause of IBDs is uncertain, and the only treatments that can be applied focus on trying to reduce the burden of the disease. People affected by IBD undergo periods of remission of the illness, with less symptoms detected, and periods of relapse, when the disease's symptoms are heavier. A very important part of the treatment is the follow up process, with continuous monitoring of the symptoms of the patient, that has to periodically describe to the doctors the effects of the disease and degree of impact on daily life. In this situation, the use of telemedicine is well suited as an integration to the follow up process, making it easier for the patient to better follow the indications of the doctors, with no need to worry going to the hospital when not necessary.

The goal of IBD tool is to provide an efficient monitoring system to connect patients and doctors reducing the need to go to the hospital. With the use of well known questionnaires, patients can describe to their doctors the symptoms of the illness in a fast and efficient manner. These questionnaires are sent periodically to the patients for compilation, but when in particular need, a patient can decide to compile a questionnaire at any moment. Doctors are always notified of the compilation of new questionnaires, and can evaluate the details anytime. To facilitate the communication between patients and doctors, a chat system is available on the platform, allowing for faster communication, for instant messaging and sending clinical files.

The work of this thesis focused on the optimization of the platform architecture, improving its overall efficiency and providing more scalability, and the development of new functionalities to further enrich the platform with

more features. To improve the platform functionalities, the already existing dashboard of the application has been extended to support new analysis: number of patients on remission or relapse from different points of view, such as gender, duration of illness, age groups, type of illness and others. More analysis were provided on the compiled questionnaires, such as questionnaires compiled by gender and total number of questionnaires by type of questionnaire. On the patient perspective, a new satisfactory questionnaire was developed to assess the degree of satisfaction with the platform and receive suggestions on the possible improvements. To further improve the usability of the system, a FAQ(Frequent Asked Questions) section was developed to provide answers to the most common questions of the patients.

Contents

List of Figures

Chapter 1

Introduction

1.1 Telemedicine

Telemedicine is the practice of medicine using electronic systems and software to provide clinical services to patients at a distance. Physicians and patients use electronic devices to communicate, to provide updates on the status of the illness and to prescribe treatments. Telemedicine services work in integration with the traditional healthcare services, providing a broader approach to deliver medical treatments. The use of telemedicine is encouraged in medical care areas where the in-person visits can be avoided, and the collection of patient information can be adapted to computing systems. The practice of telemedicine, where applicable, benefits both the patient and the healthcare system in multiple ways:

- No time and financial expenses for the patient to reach the hospital.
- Avoid the risks of exposure to contagious patients, by not coming to the hospital.
- Less interference with work and family responsibilities.
- Reduction of appointment cancellations, for example when the patient can't reach the hospital.
- Reduce number of patients inside of hospitals, allowing to prioritize the hospitalization of patients in urgent need of medical services.

For its characteristics, telemedicine is very suitable for follow-up visits, for the management of chronic illnesses and medication management.

1.2 Telemedicine in the Covid-19 pandemic

The trend of adoption of telemedicine practices among hospitals has seen a major increase during the Covid-19 pandemic. With the outbreak of the pandemic and its heavy impact on the hospitals and health care systems, the need for a strategy to produce early diagnosis and patient isolation empowered the idea that telemedicine was a key factor in the containment of contagious diseases. Particularly in Western countries, telemedicine has become a valid ally in the reduction of the damage dealt by Covid-19. In the U.S.A. four of the largest telehealth providers noticed a 50% increase in visits during the first quarter of 2020 as compared with the visits in the same quarter of 2019, with an increase of Covid-19-related encounters from 5.5% to 16.2% through the last 3 weeks of March. Furthermore, in surveillance week 13 of 2020 the number of visits increased by 154% as compared to the same period of 2019[1].

In Italy, despite being the second country with the largest number of confirmed Covid-19 cases in the early stages of the pandemic, telemedicine was not included among the essential care services provided by the National Health System. This impacted negatively on the response of the National Health System to provide means to reduce the hospitalization of patients. The first open call to telemedicine stakeholders for the research of digital solutions to handle the tracing of the virus came already 2 months after the beginning of the pandemic[2]. The result was the development of a contact tracing system via smartphone to notify the spread of the virus among people. Yet, no large-scale telemedicine services for monitoring acute and chronic patients' health status and allowing continuity of care were considered[3].

The acceleration of telemedicine practices caused by the COVID-19 pandemic has shown the importance of providing health care at a distance to reduce the workload of hospitals and to produce a response to the spread of contagious diseases. The adoption of telemedicine practices must be a key factor to be empowered in the future.

1.3 IBD

Inflammatory bowel diseases(IBD) are chronic inflammatory illnesses of the gastrointestinal tract. IBD is a complex disease of uncertain cause, which arises as a result of the interaction of genetic and environmental factors leading to inflammation in the intestine and immunological responses. A properly functioning immune system attacks foreign organisms, such as viruses, bacteria and other microorganisms, to protect the body. In patients with IBD, the immune system responds incorrectly and exaggerated to environmental triggers, leading to inflammation of the gastrointestinal system[4].

The most common IBD diseases are Crohn's disease and ulcerative colitis. While inflammation in the gastrointestinal tract occurs in both Crohn's disease and ulcerative colitis, there are important differences in the symptoms and the treatments of the two.

1.3.1 Crohn's disease

Crohn's disease can affect any part of the gastrointestinal tract, from the mouth to the anus. The most commonly affected tract is the end of the small intestine, the ileum, where it meets the beginning of the colon. Crohn's disease may appear in "patches," affecting only some areas of the gastrointestinal tract. In Crohn's disease, the inflammation may extend through the entire thickness of the bowel wall.

Among possible risk factors to develop Crohn's disease it has been found that there is a genetical factor, with more than 70 genes involved[5]. Another important risk to developing the disease is smoking, with smokers being twice as much more likely to develop Crohn's disease than non smokers[6].

Crohn's disease can cause[4]:

- abdominal pain
- diarrhea, may be mixed with blood
- abdominal distension
- fever
- weight loss
- skin rashes
- arthritis
- anemia

Bowel obstruction may occur as a complication of chronic inflammation, and those with the disease are at greater risk of colon cancer and small bowel cancer[6].

1.3.2 Ulcerative Colitis

Ulcerative colitis is a chronic illness mainly related with inflammations and ulcers of the colon and the rectum. The inflammation occurs only in the innermost layer of the lining of the intestine. It usually begins in the rectum and lower colon, but may also spread continuously to involve the entire colon.

Risk factors for UC are not certain, studies point out that genetics, changes in the normal gut bacteria or diet can affect the development of UC[7].

Ulcerative colitis can cause[4]:

- abdominal pain
- diarrhea mixed with blood
- weight loss
- fever
- anemia, with less occurrence

Often, symptoms show up slowly and can range from mild to severe. They typically occur intermittently, with periods of no symptoms and relapses. Among complications, may occur abnormal dilation of the colon, called megacolon, inflammation of the eye, joints, or liver, and colon cancer[7].

1.3.3 Medical treatment for IBD

Given the nature of IBDs, there is no cure for Crohn's disease and ulcerative colitis. Medical treatments focus on the remission of the symptoms. There is no standard treatment for patients with IBD, in most cases treatments are different from patient to patient. There are five main medications for IBD patients[4]:

- Antibiotics: antibiotics can be used for the treatment of Crohn's disease when abscesses occur. Antibiotics have not proven beneficial effects on the treatment of UC.
- Aminosalicylates: these are anti-inflammatory compounds (administered orally or rectally) used to decrease inflammation at the wall of the intestine. They are used primarily to treat UC, to reduce its symptoms. Not as effective in treating Crohn's disease.

- Biologic therapies: medications such as biopharmaceutical of biologics are used in the treatment of IBDs for patients with moderately to severely active disease.

- Corticosteroids: this type of treatment affects the body's ability to react to inflammations. Corticosteroids are most commonly used for the control of flare-ups. The treatment with corticosteroids is not usual for long-term administration because of its side effects, which can include infections, bone loss, cataracts, skin fragility and sleep disturbance.

- Immunomodulators: immunomodulators are used to modify the activity of the immune system to reduce the effects of inflammations.

When medications can't control the symptoms for IBD patients, surgical treatment may be necessary to remove damaged portions of the gastrointestinal tract. Surgical operations to treat IBD happen frequently for patients with Crohn's disease, about 70% of the cases, with 30% of patients that undergo surgery again within the first three years[8].

1.3.4 The growing trend of IBD

In the recent years, numerous studies on the incidence and diagnosis of IBD have shown that the number of patients with an IBD illness has increased in the Western countries, in particular in Europe and North America. In 2015 the 1.3% of adults in North America, a total of 3 milion, were diagnosed with either Crohn's disease or ulcerative colitis[9]. This data has shown a sensible increase from the number of adults diagnosed with IBD in 1999, approximately a 0.9% of the U.S. adult population, which was 2 million people[10].

The incidence of ulcerative colitis (UC) and Crohn's disease (CD) has also increased in Europe. In 1962 the incidence rate of UC was 6.0 per 100000 person/year and the incidence rate of CD was 1.0 per 100000 person/year. In 2010 the incidence rate of UC increased to 9.8 per 100000 person/year and for CD has increased to 6.3 per 100000 person/year[11].

These studies show that the increase of the incidence of IBD may be linked to the lifestyle of western countries population. This increasing trend must be carefully acknowledged and followed in order to provide a scalable solution to issue the necessary treatment in the incoming years.

Telemedicine must be an important factor to address this issue, for its possibilities of providing more efficient, cost-effective and scalable healthcare.

1.4 Telemedicine for IBD

Since IBD is a chronic disease, with a treatment that requires constant follow-up, there are a lot of activities which could be addressed by a telemedicine service. Patients could record their health status on a digital system, and request appointments with doctors in case of particular complications. Doctors could prescribe medicines to patients and address a calendar, step by step, program, that the patient could follow to complete the treatment.

Lots of studies have proven that a successful telemedicine service can be beneficial for the treatment of IBD, especially among patients with a shorter disease duration, and patients experiencing a flare[12].

For patients with a shorter duration of the disease, telemedicine systems can be very helpful by increasing the awareness of the disease, providing helpful information on how to live with IBD, and how to keep a healthy lifestyle.

In case of patients with a longer duration of the disease, telemedicine systems can be helpful by providing a better communication system, with a direct and more fluid communication channel between patients and doctors.

1.5 IBD tool project

IBD tool is an application that aims at assisting in the treatment of patients with IBD. It offers a direct communication channel between doctors and patients, through a specific chat or via the administrative email address of the platform, where they can discuss the required treatments and keep in touch in an efficient and fast manner.

The application aims at being user-friendly, one of the most important factors when interacting with patients and doctors, by means of an intuitive user interface and a simple design for the most common operations.

Patients can provide their health status by compiling questionnaires, which doctors can evaluate to understand the evolution of the illness and better decide the necessary treatments. This way, patients can communicate information about their current health whenever they want, with maximum convenience and better follow-up. With the assistance of notifications, the application responsively connects patients and doctors, who will be informed immediately of any updates, like when a new questionnaire is received or compiled.

IBD tool is in continuous evolution, trying to understand the necessary services that can improve the management of IBD, monitoring how the patients are using the application and looking for new ways to provide a better experience.

Chapter 2

IBD tool: a general overview

2.1 Introduction

IBD tool is a project developed with the collaboration between Fondazione LINKS, Politecnico di Torino and the Mauriziano Hospital in Turin, for the implementation of a web service to provide assistance in the healthcare of patients with IBD diseases. Patients of IBD diseases of the hospital were offered the possibility to use the web application as a support to the normal healthcare provided to them.

In the following, the different sections of the application will be presented, providing a brief description of how they are intended to work. All the presented sections precede the work of this thesis.

2.2 Registration

The registration of patients and doctors are different processes: a doctor has to invite a patient to the application in order to use it; instead a doctor can be registered to the application by applying for the registration process of the web page.

By navigating to the registration section of the startup page, a doctor inserts their personal data, an email and a password for the registration process. When the request is completed, an email is sent to the administrative email address of the platform where the request is evaluated. If the request is correct, an email is sent to the email address provided by the doctor in the registration process to confirm their account.

In order to register a patient, a doctor has to insert in the application the personal information of the patient to be registered. On completion, an email is sent to the email address of the patient, providing a randomly generated password for the account, that the patient will use together with the email to log in the application. The patient can change the password of their account at any moment if desired.

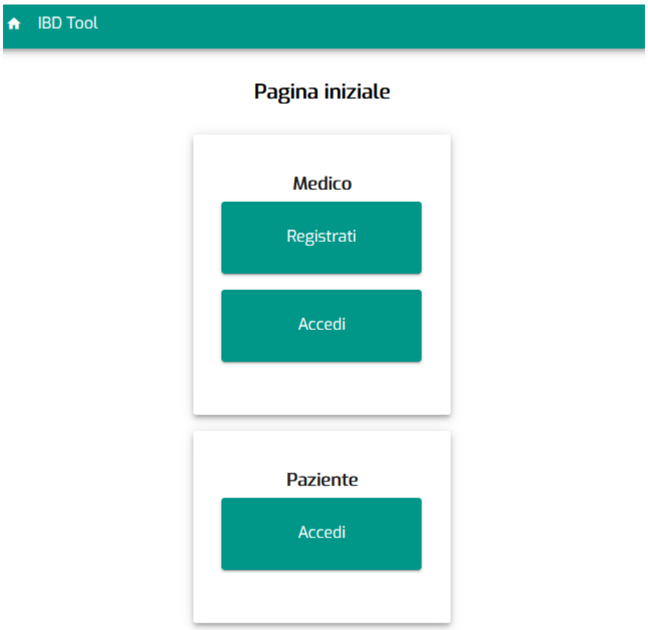


Figure 2.1. Startup page: doctors and patients registration/login

There are essentially two type of users of the application, with different authorizations and functionalities: doctors and patients. The application divides the functionalities in patient functionalities and doctor functionalities, with different web pages for patients and doctors.

2.3 Doctor functionalities

In IBD tool doctors can revise the health status of their patients, look at their compiled questionnaires, produce evaluations of the questionnaires and communicate with them whenever needed. Doctors have also access to global application information, where they can check the overall number of patients, the number of patients with a certain illness and how many patients use actively the application.

2.3.1 Home page

After a doctor logs in, the application opens the home page, where all the major functionalities for doctors are displayed. In this page a doctor can chose to:

- manage personal patients
- manage global patients of the application
- visualize global data of the application
- export data of all patients to an excel file

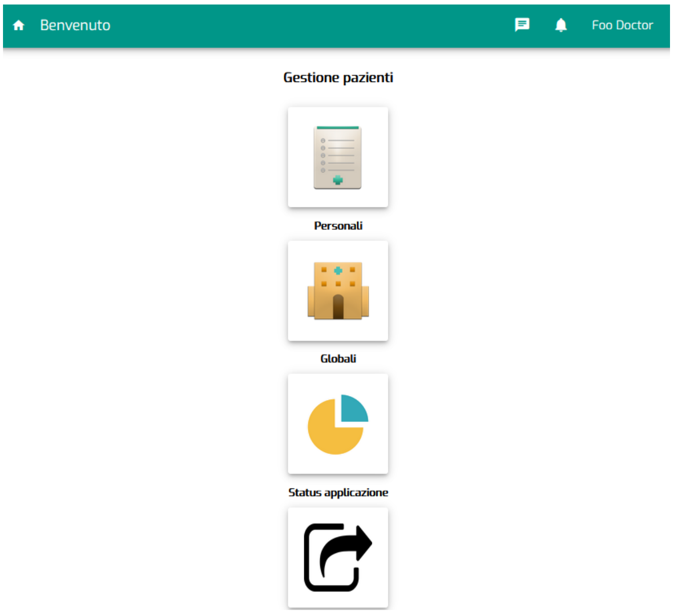
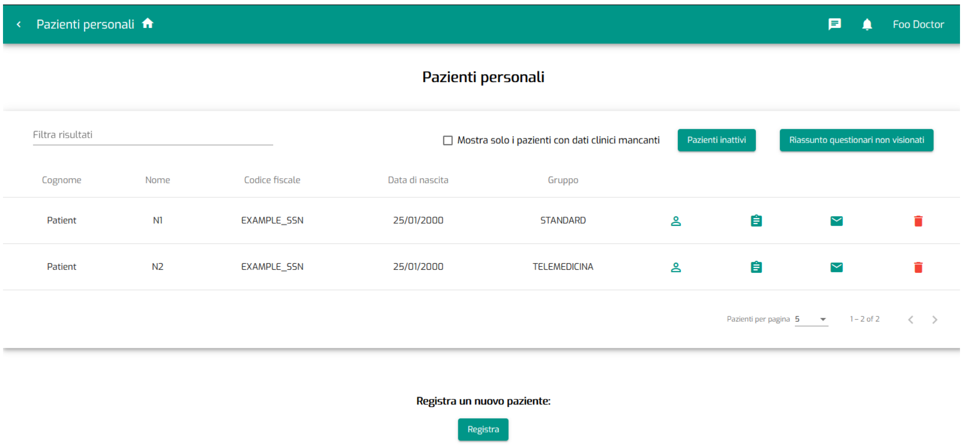


Figure 2.2. Home page for doctors

By accessing the toolbar, doctors can also use the chat and the notification center of the application.

2.3.2 Personal patients

This section provides the doctors with all the required functionalities for the management of their patients. The application shows the list of all the patients of the logged doctor, and for each patient it displays their name, surname, SSN, date of birth and type of assigned program, either telemedicine or standard of care.



Pazienti personali

□ Mostra solo i pazienti con dati clinici mancanti

Cognome	Nome	Codice fiscale	Data di nascita	Gruppo
Patient	N1	EXAMPLE_SSN	25/01/2000	STANDARD
Patient	N2	EXAMPLE_SSN	25/01/2000	TELEMEDICINA

Pazienti per pagina 5 1 – 2 of 2

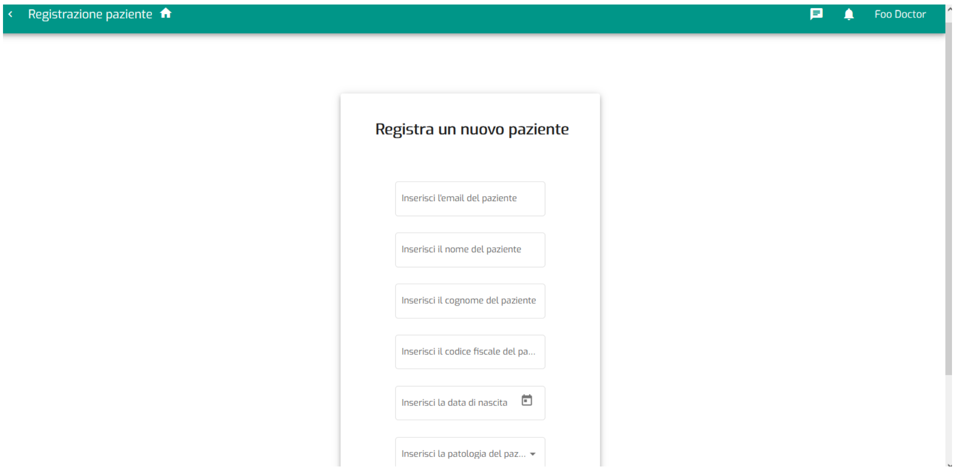
Registra un nuovo paziente:

Figure 2.3. Personal patients section.

In this section, a set of clickable icons are displayed, that let the doctor perform some actions for the management of each patient. The doctor can access the medical information of the patients by clicking on the provided icons. The features provided allow to:

- Access the medical information of the patient: this information includes data such as weight, height, sex, age and date of diagnosis of illness.
- Visualize questionnaires compiled by the patient: visualize list of questionnaires compiled and produce an evaluation.
- Contact patient via email.
- Remove patient from the platform: in case errors happened during the invitation process.

From this page it's also possible to invite new patients to the platform.



Registra un nuovo paziente

Inserisci l'email del paziente

Inserisci il nome del paziente

Inserisci il cognome del paziente

Inserisci il codice fiscale del pa...

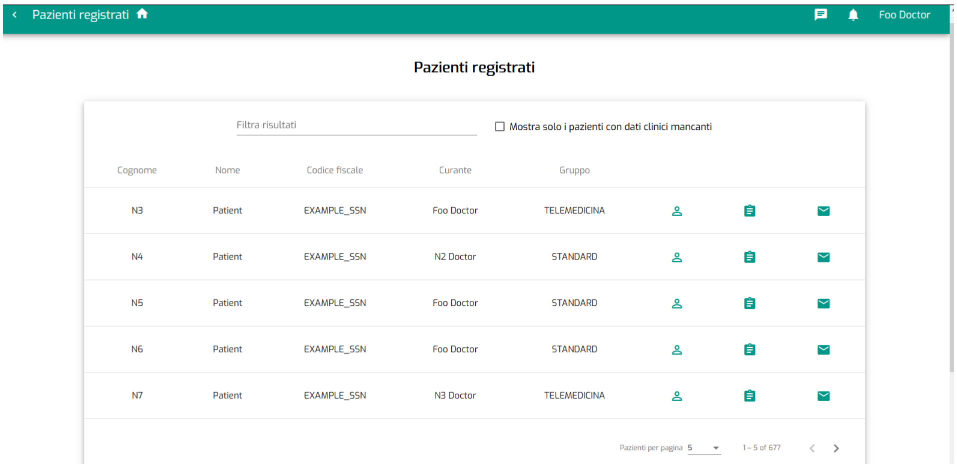
Inserisci la data di nascita

Inserisci la patologia del paz...

Figure 2.4. Form to invite new patients.

2.3.3 Global patients

In this section a doctor can see the list of all patients registered in the platform, their assigned doctors and manage their health status. All the operations available in the personal patient management page are also provided in this page, with the exception of removing patients from the platform, which is exclusively reserved for personal patients. This section is used to allow doctors to visualize the health status of all the patients of the platform, to let them cooperate and work together, being able to visualize the medical details of all the patients.



Pazienti registrati

Filtra risultati

□ Mostra solo i pazienti con dati clinici mancanti

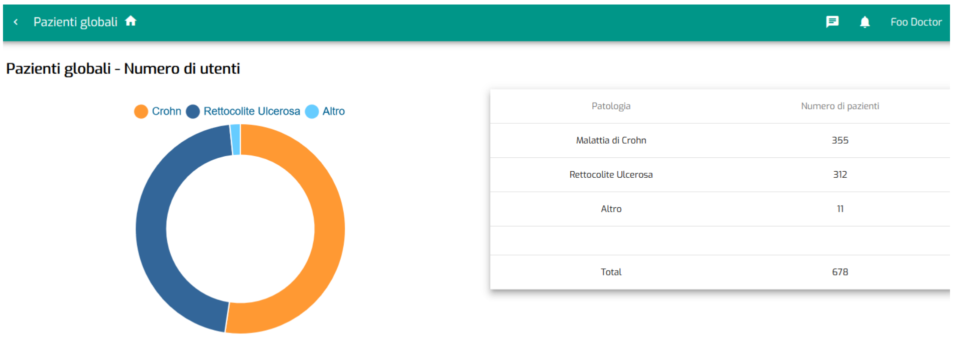
Cognome	Nome	Codice fiscale	Curante	Gruppo			
N3	Patient	EXAMPLE_SSN	Foo Doctor	TELEMEDICINA	👤	📋	✉
N4	Patient	EXAMPLE_SSN	N2 Doctor	STANDARD	👤	📋	✉
N5	Patient	EXAMPLE_SSN	Foo Doctor	STANDARD	👤	📋	✉
N6	Patient	EXAMPLE_SSN	Foo Doctor	STANDARD	👤	📋	✉
N7	Patient	EXAMPLE_SSN	N3 Doctor	TELEMEDICINA	👤	📋	✉

Pazienti per pagina 5 1 – 5 of 677

Figure 2.5. Global patients section

2.3.4 Status of application

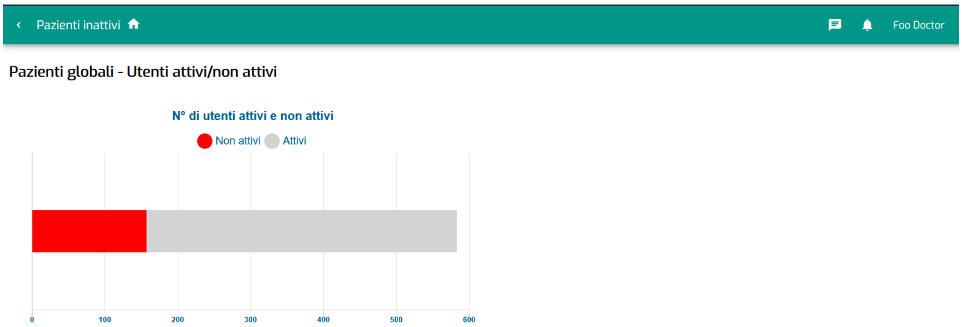
This section provides doctors with general data on the patients of the application and their use of the web page. In this page it's possible to visualize charts of the number of patients in the platform divided by type of disease, number of patients divided by status of the disease and charts of the number of active/inactive patients of the platform.



Pazienti globali - Numero di utenti

Patologia	Numero di pazienti
Malattia di Crohn	355
Rettocolite Ulcerosa	312
Altro	11
Total	678

Figure 2.6. Number of total patients per type of disease.

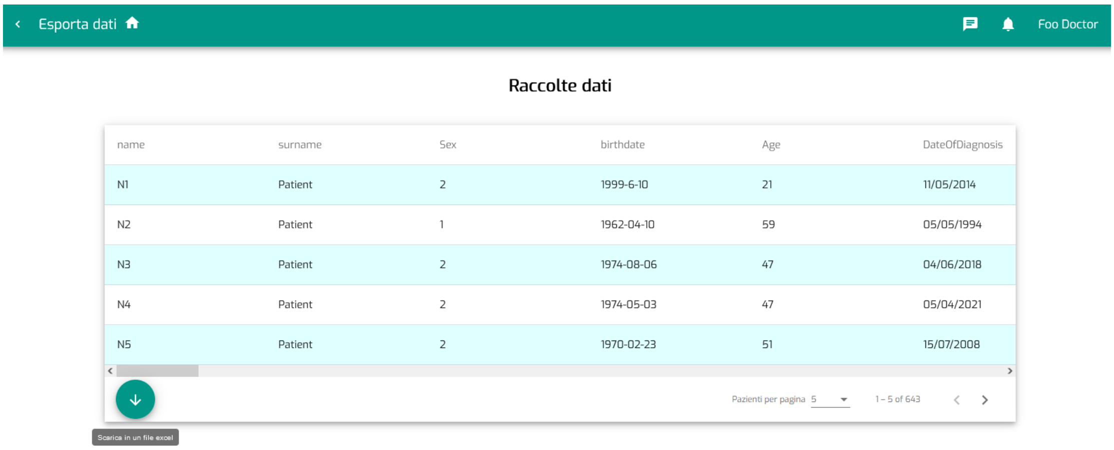


Pazienti globali - Utenti attivi/non attivi

Figure 2.7. Number of total active and inactive patients.

2.3.5 Export data

In this section doctors can process the medical data of all the patients and export it to an excel file. By simply clicking on the button, an excel file will be created on the computer of the doctor where the medical information of the users is stored.



Raccolte dati

name	surname	Sex	birthdate	Age	DateOfDiagnosis
N1	Patient	2	1999-6-10	21	11/05/2014
N2	Patient	1	1962-04-10	59	05/05/1994
N3	Patient	2	1974-08-06	47	04/06/2018
N4	Patient	2	1974-05-03	47	05/04/2021
N5	Patient	2	1970-02-23	51	15/07/2008

Pazienti per pagina 5 1 – 5 of 643

Scarica in un file excel

Figure 2.8. Export data section.

2.4 Patient functionalities

In IBD tool a patient can basically perform 3 actions:

- Compile a new questionnaire
- Visualize compiled questionnaires
- Contact the doctor

2.4.1 Home page

In the home page, a patient can easily navigate through all the services of the application, by simply clicking on the icons displayed. Previous to the work of this thesis, patients could navigate to the compile questionnaire section, to the contact doctor section and to the visualize compiled questionnaires section.

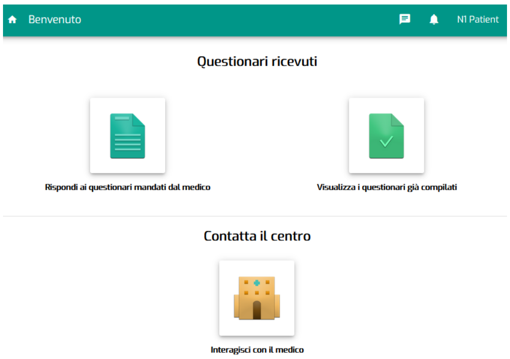
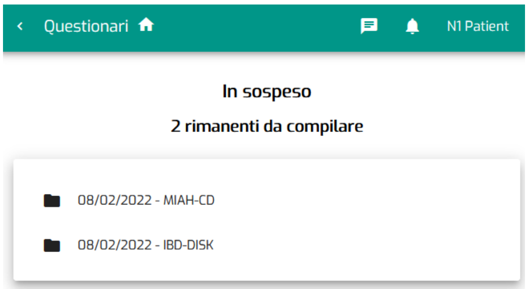

Benvenuto
N1 Patient
Questionari ricevuti
Rispondi ai questionari mandati dal medico
Visualizza i questionari già compilati
Contatta il centro
Interagisci con il medico

Figure 2.9. Home page of patients.

2.4.2 Compile questionnaires

In the compile questionnaires section, the patient can find the list of all questionnaires to be compiled. Depending on the type of their illness, patients will receive different questionnaires, scheduled by their doctors when the patient enters the platform and sent automatically by the platform on a periodical basis.



In sospeso

2 rimanenti da compilare

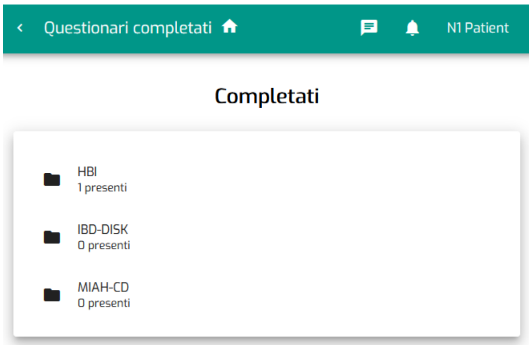
08/02/2022 - MIAH-CD

08/02/2022 - IBD-DISK

Figure 2.10. Questionnaires due to compile.

2.4.3 Visualize compiled questionnaires

In this section, patients can visualize the list of all the questionnaires that they have compiled, and look for the details of each questionnaire.



Completati

HBI
1 presenti

IBD-DISK
0 presenti

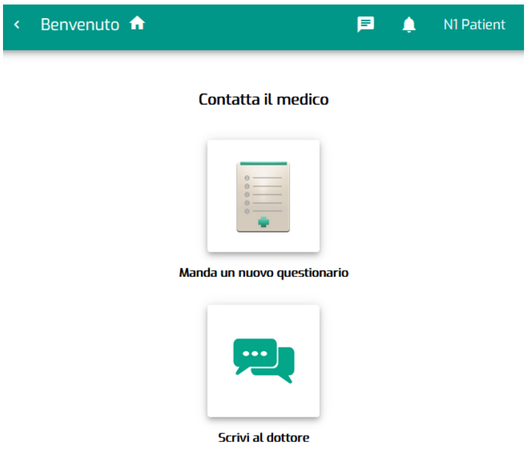
MIAH-CD
0 presenti

Figure 2.11. Compiled questionnaires section.

2.4.4 Interact with doctor

Patients can interact with their doctor by navigating to a specific section in the home page. The platform offers two ways to contact the doctor:

- questionnaire: patients can send questionnaires to doctors at any time, even if unscheduled, to notify their doctor a change on the illness status.
- chat: through the chat, patients can contact their doctor by sending messages. Each patient can contact their specific doctor through the chat, while a doctor has the list of patient to follow.



Contatta il medico

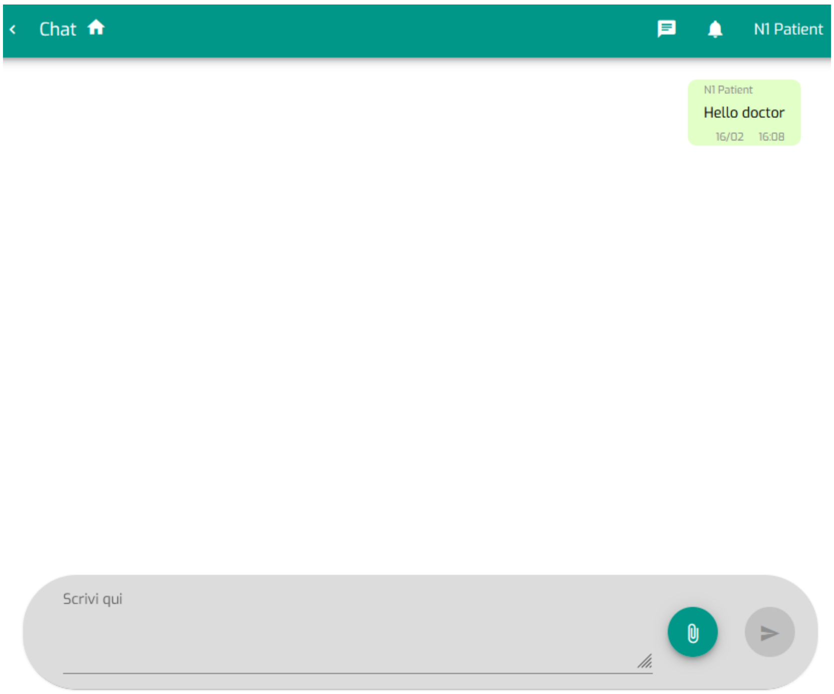
Manda un nuovo questionario

Scrivi al dottore

Figure 2.12. Interact with doctor section.

2.5 Chat

Patients and doctors can communicate with a platform specific chat. Through this chat, a patient can send direct messages to all the doctors of the platform, asking for feedback on their treatment, communicating doubts on the use of the platform or requesting medical appointments. Doctors sharing the view of the chat messages enables better collaboration from a medical point of view and also a faster reaction time to the request of each patient. The chat also allows to include files on the communication, making it easier to send images or pdf files regarding the medical procedures.



N1 Patient
Hello doctor
16/02 16:08

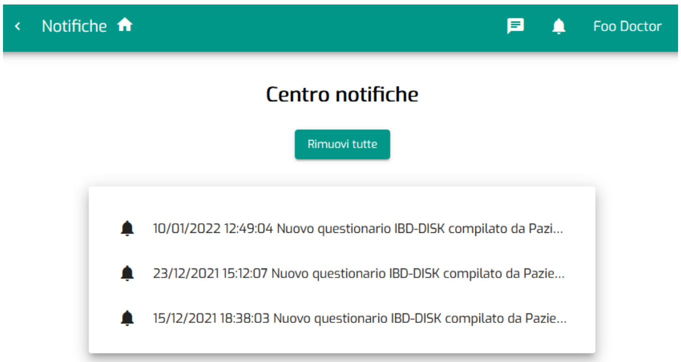
Figure 2.13. Chat page.

2.6 Notification center

In IBD tool users and doctors exchange a lot of information, for example they use chat messages to communicate or questionnaires to send medical information. In order to make sure that this information is visualized as soon as possible, notifications are sent to both doctors and patients.

By allowing to receive notifications from the web page, patients are notified of the presence of new questionnaires to be compiled or the removal of a given questionnaire. On the other hand, doctors will be informed when a new questionnaire has been compiled by a patient, or when a patient has notified an error on the compilation of a questionnaire.

In the home page of both doctors and patient, a red badge will appear on the bell icon of the toolbar, signaling the number of unread notifications of the user. Clicking on this red bell will redirect to the notification center page, where the list of unread notifications is displayed.



Centro notifiche

10/01/2022 12:49:04 Nuovo questionario IBD-DISK compilato da Pazi...

23/12/2021 15:12:07 Nuovo questionario IBD-DISK compilato da Pazie...

15/12/2021 18:38:03 Nuovo questionario IBD-DISK compilato da Pazie...

Figure 2.14. Notification center.

Chapter 3

Materials

3.1 Platform technologies

IBD tool is a web application based on a client-server architecture, where the users(doctors and patients) interact with the business logic of the application through a browser. The software architecture of IBD tool can be roughly divided in two main components:

- Front-end
- Back-end

3.1.1 Front-end

The front-end is the part of the application that manages the user interactions, defining a graphical user interface that guides the users through the services offered by the application. This part of the application is developed with the Angular framework.

Angular

Angular is an open source framework and development platform for developing sophisticated single page applications. This framework uses HTML and Typescript, integrated with a wide variety of libraries to provide support for the development and maintenance of the front-end. Some nice features of this framework are the possibility to use the TypeScript language, which allows to enforce stronger typing on objects, and the possibility to integrate the business logic and the view with a two-way data binding mechanism, avoiding a push and pull logic for connecting the two parts, a real concern for front-end JavaScript development.

The architecture of an Angular application is based on a set of fundamental concepts. The basic building blocks of an Angular project are components and services. Angular distinguishes components and services to increase modularity, by separating the view's strictly related functionalities from other processing tasks. Components define classes, which contain application and data logic, and the HTML template that associates that component to a specific view.

Every Angular application must have at least a root component, which is responsible for connecting the component hierarchy with the page document object model(DOM). A component class is identified by the @Component decorator and by a set of files:

- an HTML file that defines the HTML elements that compose the view
- a CSS file that defines the style of the HTML elements on the view
- a TypeScript file that defines the business logic of the component

A service is a specialized class that implements functionalities for the overall application. Ideally a component should only implement the functionalities to provide data and presentation logic for a specific view. On the other hand, services should implement functionalities that will be used by many components, such as fetching data from the server, validating user input or logging on files. In order to consume services, Angular provides a specific Dependency Injection mechanism, based on a global Injector defined at bootstrap, that creates the dependencies and injects the services where they are required. Defining a service with the @Injectable decorator allows to provide this service as a variable for the constructor of the components that will use this specific service.

```
@Injectable({providedIn: 'root'})
    export class Logger {
    log(logging_string: string) {
    console.log(logging_string);
    }
}
```

3.1.2 Back-end

The back-end is the part of the application that manages the server side tasks: processing user input, sending responses to clients, storing and retrieving data from the database. The back-end is also responsible of handling the security of the server, carefully checking the legitimacy of the communications from the clients and protecting personal data of its users. In the case of IBD tool, the back-end has to handle a wide variety of operations, some of which are very common to most web applications, such as creating an account, logging in, protecting communications, and also platform specific functionalities such as compiling new questionnaires, sending emails and communicating through chat.

Spring Framework

IBD tool is implemented using the Java programming language and the Spring Boot framework, a programming and configuration model for developing web applications. Spring provides over 100 starter packs which offer specific libraries for various scopes, from web programming to access to SQL/No SQL databases and cloud based support.

This framework is built on the concepts of fast development, reduction of boilerplate code and pluggable, ready-made configuration.

Some nice Spring features for web development are:

- Inversion of Control(IoC)
- Aspect Oriented Programming(AOP)
- Dependency Injection(DI)

The inversion of control principle works together with the concept of dependency injection. In object oriented applications, often classes need to interact with each other, to exchange data or use application logic. In order to communicate they keep instances of other classes or references to these classes.

When programs with complex component relationships are developed, it's often difficult to understand the connections between the classes and the overall workflow of the application. There is also the problem of circular dependencies: let's imagine the example of a class C1 that requires an instance of class C2, meanwhile the class C2 requires an instance of a class C1. In this case the program would have a run-time error and terminate the application.

```
public class C1{        public Class C2{
    int res;                int res;
    C2 var;                 C1 var;
}                       }
```

In Spring objects declare their dependencies as arguments of their constructor. But differently from other frameworks, the responsibility for injecting the dependencies is carried out by a specific component of the framework.

In Spring there is a specific object, the Application Context, that is responsible for instantiating, configuring and injecting the dependencies. The application context understands through the annotation decorators declared in the classes, which dependencies to inject for each class, and provides a mechanism to insert this dependencies when the application runs. In Spring, classes can declare these dependencies with the @Autowired decorator, followed by the object to be injected in the class.

```
public class C1{
    int val;
    @Autowired
    C2 var;
}
```

Aspect oriented programming is a programming paradigm that provides another way of thinking about program modularity. In AOP the base unit of modularity is the aspect. The aspect represents a concern, a functionality which is spread across different parts of the codebase, for example functionalities as logging, security, transaction management. Such concerns are named ”crosscutting” concerns. By using Spring annotations, programmers can develop crosscutting concerns in a single place and inject them everywhere they need to be issued.

```
@Aspect
@Component
public class Logger {
@Around("@annotation(Timer)")
    public log(){
    long start = System.nanoTime();
    doSomeTask();
    long end = System.nanoTime();
    System.out.println("Execution time: "+ end-start+" ns.");
    }
}
```

Multi-tier architecture

Most modern web applications use a similar approach to build the structure of the back-end. This approach divides the structure of the back-end in different, logically separated, layers or tiers. Each of these tiers performs a specific job and works independently from the others. Depending on the size and on the approach, web applications tend to create 3 or 4 tiers that define the architecture of the application. IBD tool follows a 4 tier approach, consisting of:

- Presentation layer: this layer is responsible of interfacing with the client requests, send this requests to the Service layer and receive back the processed data from the Service layer to create and send a response. It's also a duty of the presentation layer to communicate errors happened in the communication, for example an error on the request sent by the client or an internal server error(code 404 for resource not found or code 502 for internal server error). In Spring, a class of the presentation layer is called a Controller, and must be decorated with the @Controller annotation. Spring allows to easily configure the methods of a controller by using annotations: a method can be reached by using a mapping mechanism, where the method is either annotated with a @GetMapping or a @PostMapping decorator, passing the URI of the resource as a parameter to the annotation. This way programmers can easily create methods for GET and POST requests.

- Service layer: responsible for the implementation of the business logic of the application, building a bridge between client requests and database. The methods of this layer widely use transactions to manipulate data from the database, in order to preserve its integrity. Spring allows to implement a transactional behavior on a method by decorating it with the @Transactional annotation. A service method should always avoid to leak any personal information from the data layer, therefore it always exchanges a public representation of data with the presentation layer, DTOs, and operates with the personal data on the database level, the only place where it belongs.

- Data Access layer: this layer interacts with the data tier, providing access to the data stored in the database. Provides support for CRUD operations, which stand for Create, Retrieve, Update and Delete. In Spring terminology, the classes that implement a data access behavior are Repositories, and are annotated with the @Repository decorator. With repositories it's possible to define CRUD operations on a given database, along with other custom operations. Custom queries can be easily defined with the @Query annotation, where programmers can put custom queries or even dynamically resolved queries that are defined at run-time.

```
@Query("{'title': ?#{[0]} }")
Book findByQueryWithExp(String book);
```

- Data Domain layer: this layer creates the classes that will conform the database structure. When the service layer retrieves data from the database, this data will be returned in objects modeled by this classes. Spring calls these classes Entities, every class annotated with @Document creates a new collection on the database with the attributes of the entity.

```
@Document(collection = "books")
@Data
public class Book {
@Id
private String id;
private String title;
private String author;
}
```

Spring Security

On every web application that manages users and personal data, the protection and preservation of such information is one of the most important features to be implemented.

In particular, much effort is required to handle access control, that is the process of managing user accounts, their access to the system and the protection of the account information. All systems that manage user accounts are storing personal information, that can be stolen by malicious people, and create a lot of damage.

In order to secure account information, user passwords must never be stored in the database in plain text, but encoded in such a way that even if attackers can see the password, they can't obtain the original password or access the system. A simple hash function applied to the password is not good enough, because attackers can use a dictionary of pre-hashed passwords to try to decode user information.

A good practice is to increase the security of the password by applying a transformation before hashing it. For every new user, create a sequence of characters, that should be long enough and random, called "salt", to be concatenated to the users' password before applying a hash function to it. This salt has to be different for each user, to avoid the risk of a dictionary attack with a single salt attached. The resulting password can be safely stored in the database.

Another part of the access control process is the management of user privileges and authorization, that is the process of controlling the actions that a certain user is allowed to make on the system.

Spring security offers a specific mechanism to handle the authentication and authorization of users.

In the Spring architecture, when a user has proven their identity, for example via a username-password mechanism, all the information related to the user can be configured in a class called Authentication. This class manages information such as:

- principal, identifies the user
- credentials, a password, often cleared when user is logged to avoid any leaking
- authorities, the list of authorities associated to the logged user

Through this object, which can be accessed by everywhere on the application, it's possible to enforce controls on the authorization of the current user to access certain parts of the application. This is applied by surrounding methods, of the Service layer or even of the Presentation layer, with special annotations. For example using the @PreAuthorize decorator, an API can enforce control over the authorizations of a given user before executing a processing task. Also the opposite behavior can be achieved with the @PostAuthorize decorator, that allows to enforce controls over the returned value of the API.

IBD tool uses hashing with salt to protect user account information, and uses the authorization decorators to provide control over actions that can only be executed by doctors, and the actions that can be performed by patients.

In order to provide authorization to the users of the platform, IBD tool uses an approach based on Json Web Tokens(JWT). A JWT is a shared secret between the user and the platform that allows IBD tool to recognize an authenticated user. When the system verifies that the credentials sent in the request are correct, it creates a JWT string and sends it back in the response. This JWT is a proof of authentication, it contains information that the server has crafted specifically for an authenticated user. In particular a JWT contains:

- Header: contains metadata about the token.
- Payload: contains information of the user, with the associated granted authorities.
- Signature: contains the parts above, signed with a secure secret key, only known by the platform.

In the next requests, the user will embed this token on the authentication header of the HTTP packets to provide proof of authentication, and the system will assess the validity of the token.

3.1.3 REST

REST, which stands for REpresentational State Transfer, is an architectural style for distributed hypermedia systems. The main principles of a RESTful service are:

- Uniform interface: an interface must uniquely identify each resource. Each resource should have a uniform representation in all APIs of the server.
- Client-Server: a RESTful architecture should be client-server, strictly separating the functionalities of the client from the business logic of the server. This means that if the client application needs to be changed, any changes implemented should have no impact on the server. At the same time, a change on the database on the server side should not impact the client application.
- Stateless: each request must contain all the necessary information for the server to produce a response. This means that there is no need to perform series of requests to execute operations, reducing the possible errors occurring over time.

- Cache: a REST API should encourage the storage of cacheable data. Whenever possible, the responses from the server should say that data can be stored by a client for an expiration time, allowing to reduce the number of interactions with the API.

- Layered system: components of the system are divided in hierarchical layers, only working with the closest layers of the current component.

- Code on demand: an optional feature, RESTful APIs should try, whenever possible, to send scripts or applets to be executed on the client, reducing the need of pre-implemented features.

3.1.4 MongoDB

As a database to store the business data, IBD tool uses MongoDB, an open-source, non relational, database management system. MongoDB is originally developed following the NoSQL philosophy, that is a way of thinking about databases completely different from the old and reliable SQL relational databases, with a very strict structure and a very specific way to express queries. Relational SQL databases relied heavily upon the concepts of tables, joins and SQL query language. These systems were able to store data efficiently and perform complex queries on the dataset.

Non relational databases started becoming more relevant when applications required more flexible systems to store their data, capable to adapt faster to changes in the definition of the datasets.

Non relational database are based on a set of concepts:

- Schema-less structure: schema can change dynamically, the data previously stored can change its composition at any moment with no side-effects for the datasets.

- Horizontal scalability: when there are problems with resources, NoSQL databases scale by adding more servers.

- Complex data: non relational databases are more suitable for complex data, such as JSON or XML.

- No joins: non relational databases try to avoid join operations, which are usually complex and slow.

MongoDB is a document oriented NoSQL database. A document is the base storage unit of the system, corresponds to the concept of row in a relational database. Whenever a document is stored in the database, it is associated to a key, that uniquely identifies the document on the database. A document can store any kind of data, allowing also to store nested data. This allows to store lists, maps or arrays of objects inside of a document. Documents are organized in collections. Collections correspond to tables in relational databases, gathering together documents logically similar, for example the documents that belong to the collection users.

```
users:{
{
name: "Mario";
surname: "Rossi";
job: "Sales and Finance";
sales: List<Sale>;
}
{
name: "Mario";
surname: "Verdi";
job:"Software Engineer";
bugFixes: Map<Bug,Fix>;
}
}
```

MongoDB provides high availability through sharding. When databases work with huge quantities of data and high computing load, the scalability of the system is very important. Shards allow to distribute data in multiple servers, allocating subsets of the overall dataset to each server. When in need of more computing resources, the system can be scaled by adding more servers and distributing data in more parts of the system. This overall is more efficient, convenient and cheaper than upgrading servers.

IBD tool works with servers on the cloud, with multiple replica sets that allow to address big workloads and provide high availability.

3.1.5 HTTP protocol

The HTTP protocol is an application level protocol for exchanging resources on the web. This protocol works with a client-server philosophy, where the initiation for the communication is issued by a client, which is called a request, and the messages sent by the server are called responses.

The HTTP protocol is stateless, which means that each packet of the communication is independent from the previous ones. This could be problematic when users try to execute operations based on previous actions performed in a specific interaction. Though natively HTTP doesn't support a state, the capability to extend its header allows to add information like cookies to keep sessions over communications.

HTTP is used to fetch resources on the web. Each resource is identified by a unique URL(Uniform Resource Locator). When trying to use a resource, the HTTP packet will refer to this resource through its URL.

The HTTP protocol is used to perform a set of actions, the most common are:

- GET: this action is used to request a resource from a server. Such resources could vary from raw data to HTML pages. The resource to be requested is identified by the URL of the request.

- POST: this action is used to send data to the server, for example forms with personal information.

- DELETE: this action is used to issue the destruction of the resource identified by the URL.

- OPTIONS: this action requests for the supported options of the resource identified by the URL.

For every HTTP communication, the server has to provide a status code to the response to communicate the result of the operations. Each status code is composed by three numbers, the first one corresponds to the class type of result, the other two identify specifically an outcome:

- 1xx-Informational: this class means that the request was correctly received and is being processed.

- 2xx-Successful: this class indicates that the request was correctly issued. A common status code is 200, which indicates that the request of the client was successfully executed.

- 3xx-Redirection: this class is returned to indicate that further actions are necessary to complete the request. A very common code to issue URL redirection.

- 4xx-Client error: this class indicates a communication error on the client side. For example the code 404 indicates that the client was trying to fetch a resource that doesn't exist on the server(NOT FOUND).
- 5xx-Server error: this class indicates that the server was not able to execute the operation required due to an internal error.

3.1.6 Github

Github is a hosting service for the development of software products. Using this hosting service, developers can work on a distributed platform that encourages code sharing and cooperation. Github is based on Git, a distributed version control tool that provides a way to keep track of software program versions. Software projects are organized in repositories, storage units that contain the code and the dependencies of the projects. Each collaborator of the project has access to the repository and can collaborate with the main project by opening branches. A repository has many branches, each branch contains a part of the system developed independently by collaborators.

The main operations provided by Git are:

- Fork: creates a copy of the forked repository. It's normally used for proposing possible bug fixes or as starting point of a new project.
- Fetch: downloads refs from repositories or branches. Allows to align local branches to remote branches that have been updated.
- Pull: fetches changes from remote branches and integrates them into the current branch.
- Commit: creates a new commit containing the changes on the current branch and the given log message and points the HEAD of the branch to the new commit.
- Push: updates remote branches with the content of the local branch, allowing to publish the latest commits to the remote repository.
- Merge: integrates changes between two branches.

IBD tool is hosted on Github, on two private repositories, one that manages the front-end and one for the back-end. Each repository has a production branch, that is the where the production code is stored for deployment.

Developers work on upgrades of the platform on personal branches, where code is tested locally. Once a new version of the system is ready, changes are merged in the ”dev” branch, a branch for aligning developers. Then changes are pushed in the ”main” branch, where the code is ready for deployment on remote servers.

3.1.7 Heroku

Heroku is a platform as a Service(PaaS) that supports the build and deployment of applications and multiple programming languages. Heroku hosts the back-end part of the web service. Linking the Github account that hosts the repositories with heroku allows to perform automatic building and deployment of the application when the main branch is updated with a new version.

3.1.8 Firebase

Firebase is a serverless development and deployment platform for a wide variety of applications. This platform can host applications and offer services to implement new features. IBD tool uses Firebase for hosting the front-end part of the application, deployable using the Command Line Interface(CLI), that provides a fast and easy to use way to build and deploy applications through command line commands.

To provide user notifications on the browser, Firebase Cloud Messaging offers a way to create a messaging queue to address real-time notifications. Firebase creates queues based on ”tokens”, which are keys that match a specific queue. By identifying user devices with a unique token, it’s possible to send notifications from the server to a specific device.

Whenever a user allows the reception of notifications, a token is sent to the server, which will assign a specific queue for the device. When a notification must be sent, the server sends a message to Firebase, which will send the notification to the device through HTTP.

3.2 Questionnaires

Questionnaires are the basic tool used by the application to receive and record the medical information of the patients. Questionnaires are compiled periodically by the patients and sent to their doctors to evaluate their current health status. The type of questionnaire to be compiled depends on the type of pathology of the patient and when a doctor registers a new patient on the platform, the specific set of questionnaires related to the patient's pathology are prepared to be sent automatically, on a periodical basis.

In order to monitor the health status of patients, the application uses the following questionnaires:

- **HBI**(Harvey-Bradshaw Index): this questionnaire monitors the activity of Crohn's disease by asking a set of questions based on clinical information:

  - General health status of patient, from a scale of 0 to 4.
  - Abdominal pain, from a scale of 0 to 3.
  - Number of liquid evacuations.
  - Abdominal mass, from a scale of 0 to 3.
  - Existence of complications.

  The score of the questionnaire is computed summoning the numerical values of the answers provided by the patient. The value of the score defines the status of the activity of Crohn's disease on the patient: with a score lower than 5 the activity is considered remitting, between 5 and 8 it's considered mild, between 8 and 16 it's considered moderate and over 16 it's considered severe[15].

- **SCCAI**(Simple Clinical Colitis Activity Index): this questionnaire monitors the activity of ulcerative colitis by asking a set of questions regarding:

  - Bowel frequency.
  - Urgency of defecation.
  - Blood in stool.
  - General health.

  Similarly to HBI, the SCCAI defines a score based on the sum of the numerical values of the answers provided: a score lower than 5 is indicative of a remission of the disease's activity, a score greater than 5 indicates a relapse. In IBD tool patients compile the PATIENT-SCCAI, the specific version of the SCCAI for patients, meanwhile doctors compile the CLINICAL-SCCAI.

- **MIAH-UC**(Monitor IBD At Home): this questionnaire monitors the health status of the patient with ulcerative colitis by asking questions regarding general health, rectal bleeding, number of bowel movements, existence of urgent bowel movements, mucus loss, asthenia. A score lower than 3.6 indicates remission, a score greater than 3.6 indicates relapse.

- **MIAH-CD**: this questionnaire monitors the health status of the patient with Crohn's disease by asking questions regarding general health, rectal bleeding, number of bowel movements, existence of urgent bowel movements, abdominal pain. A score lower than 3.6 indicates remission, a score greater than 3.6 indicates relapse.

- **PRISM**: this questionnaire asks the patients to define the illness' impact in their normal lifestyle. The questionnaire shows two circles, one represents the patient and the other the illness. By overlapping the two circles the patients can describe the impact of the illness in their lives. The score is computed in terms of distance between the two circles, the more overlap between the circles, the heavier the burden of the disease. Also for this questionnaire there's a version for patients and a version for doctors in IBD tool.

- **IBD-DISK**: this questionnaire evaluates the impact of the illness in the patient's life based on a set of points of view: education and work, energy, emotions, sleep, abdominal pain, joint pain, regulating defecation, body image and sexual functions. The patient answers with numerical values from 0 to 10 for each topic and a score is computed based on these.

- **WPAI**(Work Productivity and Activity Impairment): this questionnaire assesses the impact of the illness on the patient's working life.

- **TSQM**(Treatment Satisfaction Questionnaire for Medication): this questionnaire evaluates the patient's satisfaction with the drugs taken for the treatment, considering its effectiveness, its possible side effects and its convenience.

- **IBDQ**: this questionnaire evaluates the impact of the illness on the quality of life of the patient. The questionnaire is composed of 32 questions regarding topics such as social life and emotional life.

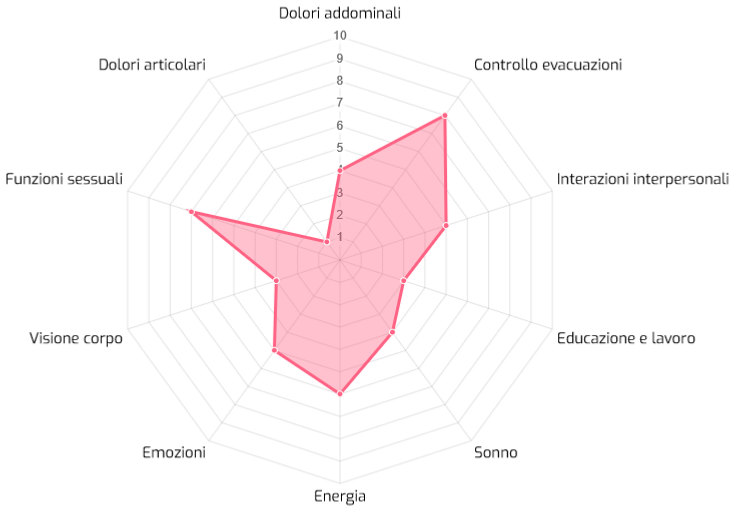


Figure 3.1. Example of IBD-DISK questionnaire, graphical representation

- **MMAS8**: this questionnaire assesses the patient's consistency in following the treatment, with questions regarding the frequency taking the necessary drugs and the times those drugs were not taken.

- **PHQ9**(Patient Health Questionnaire-9): this questionnaire is used to evaluate a possible depression of the patient. The score of the questionnaire analyses the status of depression: minimal depression with a score lower than 4, mild depression with scores from 5 to 9, moderate depression with scores from 10 to 14, moderately severe depression with scores from 15 to 19 and severe depression with scores greater than 20.

- **IPAQ-SF**(International Physical Activity Questionnaire Short Form): this questionnaire evaluates the physical activity of the patient in the last seven days. Only physical activities that lasted at least 10 consecutive minutes are counted for this questionnaire. The amount of physical activity is calculated in MET minutes(metabolic equivalent task). Each activity is assigned a specific weight depending on the physical effort of the activity, for example walking has a 3.3 weight, moderate activities have 4 weight and vigorous activities have 8 weight. Walking for 10 minutes a day for 4 days computes 3.3x10x4=132 MET. Scores lower than 700 indicate that the patient is inactive, scores greater than 700 and lower than 2500 indicate usually active and scores greater than 2500 indicate very active patients.

- **PSQI**(Pittsburg Sleep Quality Index): this questionnaire evaluates the quality of sleep of the patient in the last month. Patients are asked to describe the quality of their sleep in terms of number hours slept, the time taken to fall asleep and the awakening hours. A score lower than 5 indicates remission, a score greater than 5 indicates relapse.
- **EQ5D5L**: this questionnaire evaluates the general quality of life of the patient, considering work life, physical and psychological health and personal care.

The questionnaires are sent periodically to the patients, depending on the category assigned by the application. When a patient is registered in the application, the system assigns them randomly to one of two categories: standard or telemedicine. A patient in the telemedicine category will receive different questionnaires and more frequently than patients assigned to standard.

Patients on standard category are sent questionnaires every three months:

- HBI, IBD-DISK and MIAH-CD for Crohn's disease.
- PATIENT-SCCAI, IBD-DISK and MIAH-UC for ulcerative colitis.
- IBD-DISK for other IBDs.

Patients on telemedicine receive questionnaires more frequently and compile also other questionnaires:

- Monthly:
  - HBI, IBD-DISK and MIAH-CD for Crohn's disease.
  - PATIENT-SCCAI, IBD-DISK and MIAH-UC for ulcerative colitis.
  - IBD-DISK for other IBDs.
- Every three months:
  - PHQ9, PRISM, IPAQ-SF, WPAI, IBDQ, MMAS8, TSQM for all patients.
- Every six months:
  - PSQI, EQ5D5L for all patients

Chapter 4

Results

The main work of this thesis focused on the improvement of the previous infrastructure of the platform, by optimizing the existing features and fixing incorrect behaviors, and the development of new functionalities to enrich the overall service.

4.1 Improvement of the previous infrastructure

At the start of this thesis, the platform required some optimizations on the data layer architecture of the application, where the need of a more scalable service was not met by the previous architecture. Some other improvements were required for the adjustment of previous functionalities that were not meeting the new needs of the service.

In the following, the most important changes are described.

4.1.1 Optimizing the data layer architecture

In the starting point of this work, the application showed stability problems on the doctor side of the service. Some doctors of the platform could not consistently use the most common features, such as access to the list of personal patients, of the global patients, or to the export data section. Other features, for example the dashboard page, were too slow to be used with comfort.

This situation happened mostly due to the increase of the number of patients of the platform in the previous year, which also increased the data associated with the questionnaires, with the chat or with the notifications.

Studying the codebase of the platform, no issues were found on the processing of requests by the APIs, suggesting that the business logic of the application was designed correctly and therefore the issue was somewhere else. Analysing the composition of the data structures on the database, some possible bottlenecks were found.

The data layer architecture of the back-end was not adequate anymore to the new data load on the platform, which caused inconsistency problems on the responses of the server, due to the incapability to address certain requests.

With the previous data layer architecture, the data was roughly divided in 3 different categories:

- users: a collection to store all the information regarding the users of the platform, like email, name, surname and hashed password.
- questionnaires: a collection to store all the questionnaires compiled by patients of the platform.
- chat messages: a collection to store the messages exchanged between patients and doctors of the platform.

In particular, users of the platform are divided in two types: doctors and patients. A doctor in the system must be associated to a list of patients, the patients assigned to the doctor, with the possibility to modify their personal information and medial data. For each user of the platform, both doctors and patients, a set of notifications is associated, related to the compilation of questionnaires, the removal or the reception of a new questionnaire, the incorrect compilation of a questionnaire or the modification of personal information of a patient. All this information was previously stored inside the users collection.

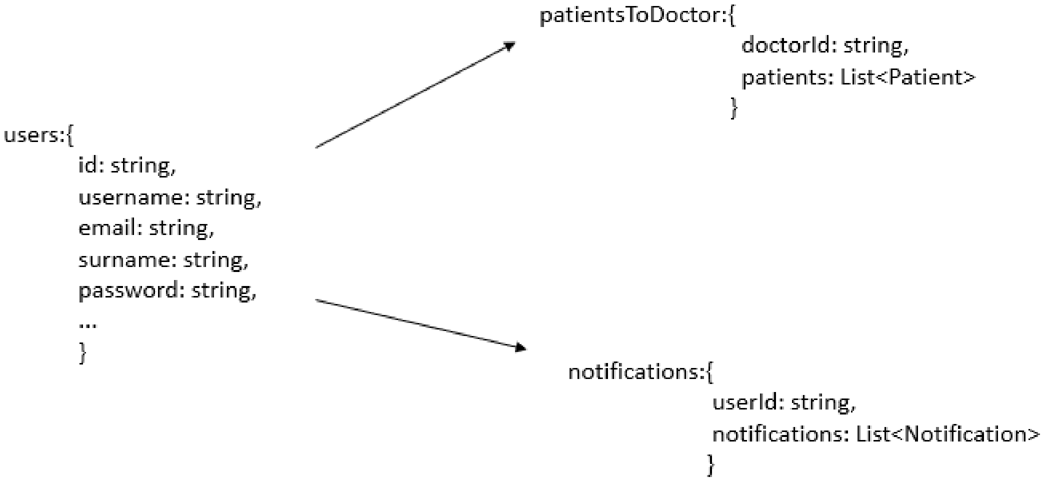
```
users:{
    {
    id: string,
    email: string,
    name: string,
    surname: string,
    notifications: List<Notification>,
    patients: List<Patient>
    ...
    }
}
```

The structure of the users collection presents scalability issues. When the number of patients on the platform increases, and therefore the list of notifications increases, each user document grows too much on dimension, stressing the database, but also the memory of the server when it fetches the data.

In particular, the bottleneck of this architecture was that doctor documents where too big in dimension, having all the list of patients with related data and the list of notifications of the doctors, making it very hard for the server to fetch all the data from the database in a reasonable time and also very consuming from a memory perspective.

The solution adopted required to divide the data in the users collection in multiple new collections, to reduce on average the data fetched by the server and the size of the user documents:

- doctor patients: one specific collection to store the list of patients for each doctor.
- notifications: one specific collection to store the notifications for each user.



```
users:{
        id: string,
        username: string,
        email: string,
        surname: string,
        password: string,
        ...
        }
```

```
patientsToDoctor:{
                doctorId: string,
                patients: List<Patient>
                }
```

```
notifications:{
                userId: string,
                notifications: List<Notification>
                }
```

Figure 4.1. Division of data to new collections

With this division of data, the system scales better on the increase of patients on the system, because the average size of the user document doesn't increase linearly with the increase of patients on the platform. The data on the ”doctor patients” collection only keeps track of the patients of a specific doctor, requiring less amount of data to store this information.

The new structure of the database required to create new classes on the back-end of the system to support the previous APIs, adapting all the codebase to the new database structure. First of all, the previous data regarding the patients associated to the doctor and the notifications of the user were removed from the user Entity and attached to the new Entities. For each new collection, a new MongoRepository instance was created, to build a communication gateway with the database. And for each new Repository instance, a new Service has been developed to address all necessary interactions and offer utility methods for all the other classes. As a result, all methods of the existing codebase were adapted to use this services, removing the old functions offered by the user collection service with new methods of the newly created services.

With the new architecture, some operations were slowed down due to the fact that in the previous architecture the number of interactions with the database was lower. Dividing the storage of the data in multiple collections, the number of interactions with the database increases, because what used to be performed in one single place now has to be spread in several places. For example, registering a new user in the application previously required to update the doctor associated to that user with a new Patient associated, and create a new user document on the database. This changed to several transactions with the database, to create a new user document, to register the association doctor-patient and to create an association for the notifications of that user.

The slow down of some operations was eventually negligible, too little to impact the performance of the application. On the other hand, the decrease of the dimension of the user documents in the database and the specialization of the operations on different services, greatly decreased the memory load on the server, allowing the application to support again the normal operations of the doctors of the platform. In more specific details, the number of critical error on the application dropped significantly, most of them being related to the impossibility to send a response to the client before timing out. The response time and the memory consumption on the server stabilized, allowing for a more comfortable use of the application.

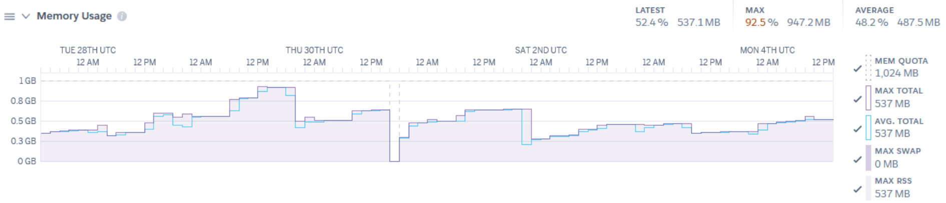


Figure 4.2. Heroku server metrics: memory consumption

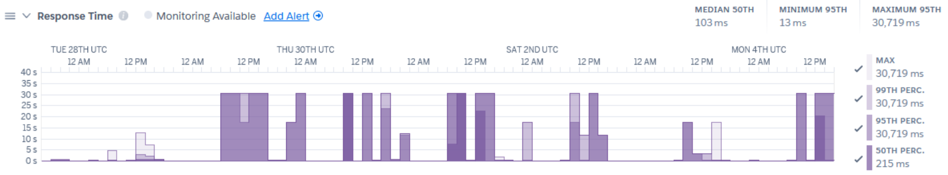


Figure 4.3. Heroku server metrics: response time

4.1.2 Addition of junior and senior doctors

The doctors of the platform requested a way to create a distinction on the system between junior doctors and senior doctors. Previously the platform used the id of doctors in the system to differentiate between senior and junior doctors, which required to know the exact ids of the doctors and hardcode these values to the APIs that required this distinction.

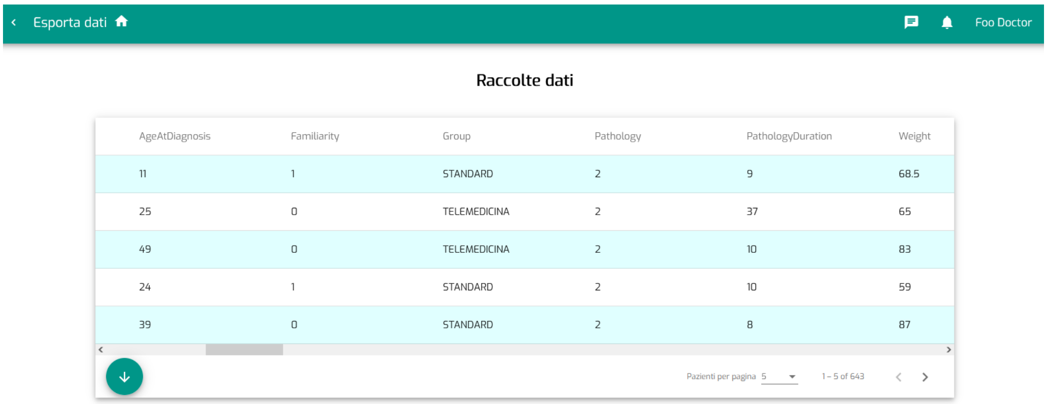
In order to provide a better distinction, a new attribute in the users collection was created for doctors, the field ”doctorRole”, that can only accept the values ”JUNIOR” and ”SENIOR”. This way, all the APIs that require the role of a given doctor were modified to obtain the role from the user document of the specific doctor. Furthermore, the registration function for doctors has been modified, requiring also to provide the role of the new doctor in the registration process, so that every new doctor that registers on the system already has an assigned role.

4.1.3 Export data

Previously the export data section was composed by two tables, one for the patients with Crohn’s disease, and one for the patients with ulcerative colitis. Each table included information such as name, surname, birth date of the patient, date of compilation and result of questionnaire, either CLINICAL-HBI or CLINICAL-PRISM, and the evaluations of the doctors.

Consulting the doctors of the platform, some requests for change were issued regarding the data to be exported for the concordance study. Contrarily to the previous implementation, the concordance study now required to merge the two tables and to include the medical data of each patient. Also the requirements on the evaluation of doctors changed, with no need to fetch the evaluation of all the junior doctors, but only the last evaluation of a junior doctor on a questionnaire.

To achieve this, the back-end function was modified to provide the new data. To keep the business logic of the previous code, the two functions providing the data of patients of Crohn's disease and ulcerative colitis were not integrated in one single method. This decision was also taken to reduce the response time of the server for this specific method, which is heavy on the data exchanged and therefore slow. Keeping the two endpoints, it's possible to start populating the table faster, meanwhile the second request is processed. For each type of illness and for each patient fetched from the database, to their personal information it was also attached the specific medical data related to the patient. All this information constitutes the response from the server. On the front-end, a new MatTableDataSource was created to visualize the concordance study data, and the list of headers was updated to properly hold the medical data.



Raccolte dati

AgeAtDiagnosis	Familiarity	Group	Pathology	PathologyDuration	Weight
11	1	STANDARD	2	9	68.5
25	0	TELEMEDICINA	2	37	65
49	0	TELEMEDICINA	2	10	83
24	1	STANDARD	2	10	59
39	0	STANDARD	2	8	87

Pazienti per pagina 5 1 – 5 of 643

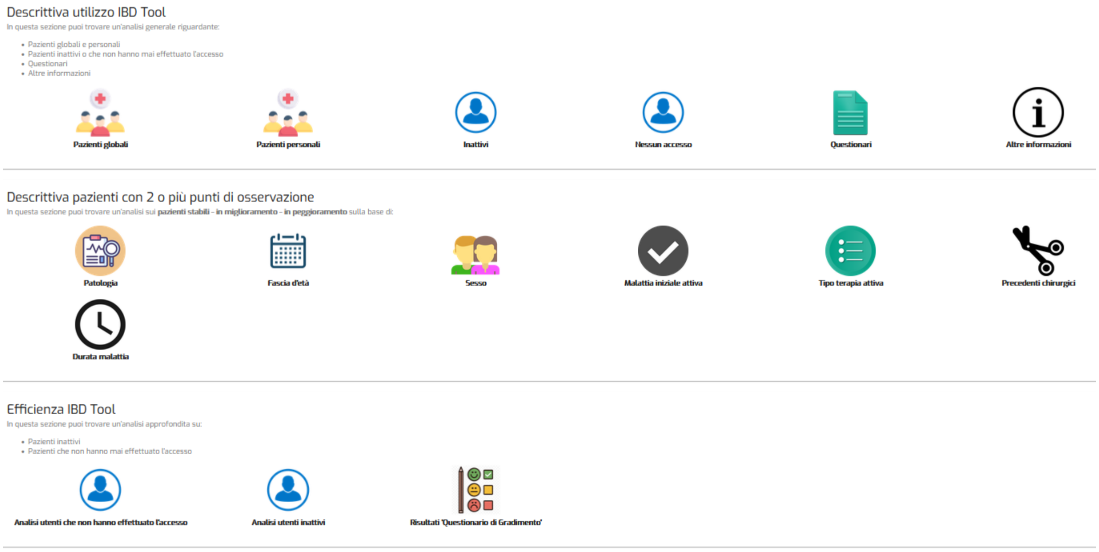
Figure 4.4. Export data section with medical data

4.2 Implementation of new functionalities

Talking with the doctors of the department of gastroenterology of the Mauriziano Hospital in Turin, some requests were made to implement new functionalities to enrich the usefulness of the application. From the discussion, it emerged that new features were required to support the studies of the doctors on the evolution of the health status of the patients of the application, and the need to better reach the patients of the application, asking for their advice, doubts on the use of the application and their experience on the use of the service so far.

4.2.1 Doctor side dashboard

The first request of new functionalities regarded the use of the application by doctors. In particular, new functionalities to extract meaningful information of the health status of the patients, of the compilation of questionnaires and other data to support their studies on the effectiveness of the service. To locate these new services, a new dashboard was created on the application status section of the doctors web page.


Descrittiva utilizzo IBD Tool

In questa sezione puoi trovare un'analisi generale riguardante:

- Pazienti globali e personali
- Pazienti inattivi o che non hanno mai effettuato l'accesso
- Questionari
- Altre informazioni

Descrittiva pazienti con 2 o più punti di osservazione

In questa sezione puoi trovare un'analisi sui **pazienti stabili - in miglioramento - in peggioramento** sulla base di:

Efficienza IBD Tool

In questa sezione puoi trovare un'analisi approfondita su:

- Pazienti inattivi
- Pazienti che non hanno mai effettuato l'accesso

Figure 4.5. New graphic interface for the doctor dashboard

This new dashboard is divided in three main sections:

- Analysis on the utilization of IBD tool: this section contains general analysis data on the use of the application by the patients. The different sections display information such as the total number of patients on the platform, the total number of patients by type of disease or the number of active patients on the platform, among other data.

- Analysis on the health status of patients: this section contains general information about the evolution of the health status of the patients.

- Analysis on the efficiency of IBD tool: this section contains general information about the efficiency of the platform at involving the patients on its usage. It contains a detailed analysis on how the patients prefer to use the application, for example a comparison of how patients of a certain age are more active users of the application then other age groups.

In the following, some of these sections will be presented with more detail.

4.2.2 Questionnaires

The doctors of the application requested the possibility to access to detailed information on the compilation of questionnaires by the patients, to better track the overall use of the application and the rate of compilation of questionnaires.

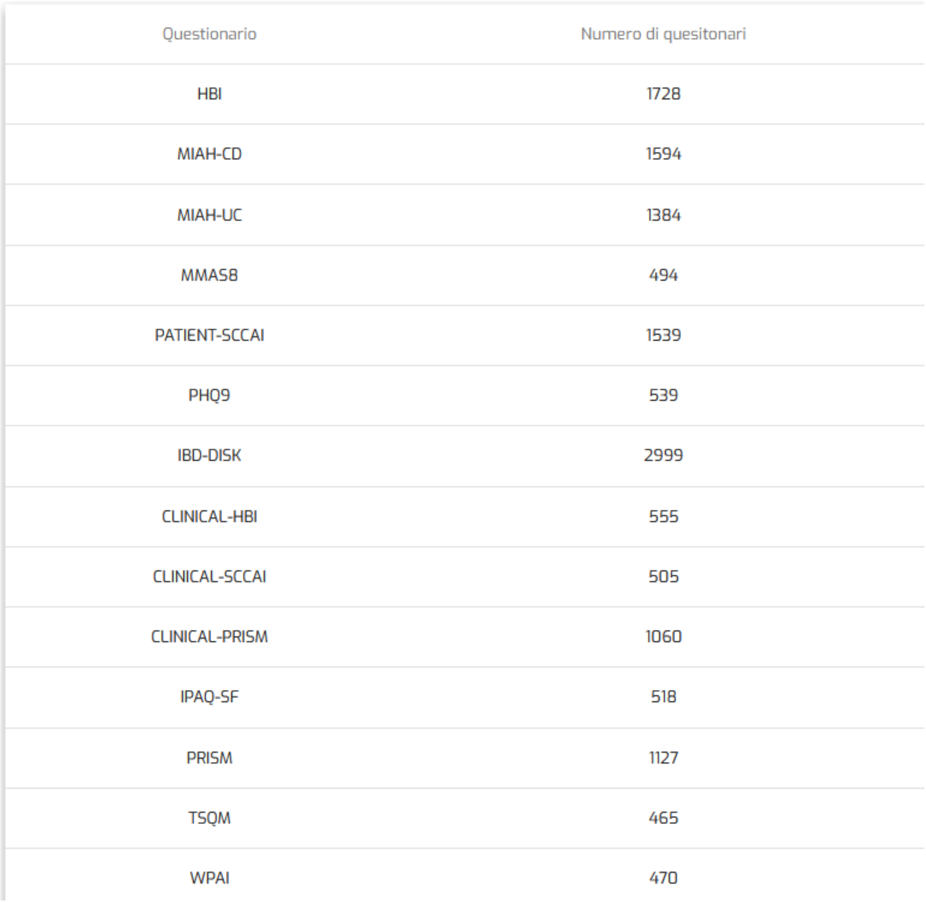
In order to achieve this, a set of different analysis were conducted:

- Number of questionnaires per type: the total number of questionnaires compiled by the patients were collected and divided by type of questionnaire. The results were displayed on a table for better visualization.

- Number of questionnaires per sex: in this analysis, the total number of compiled questionnaires were divided by the sex of the patient that compiled them. To visualize this data a doughnut chart was displayed, where doctors can immediately see how generally the data is distributed. To detail the exact number of questionnaires, a table was attached to the chart.

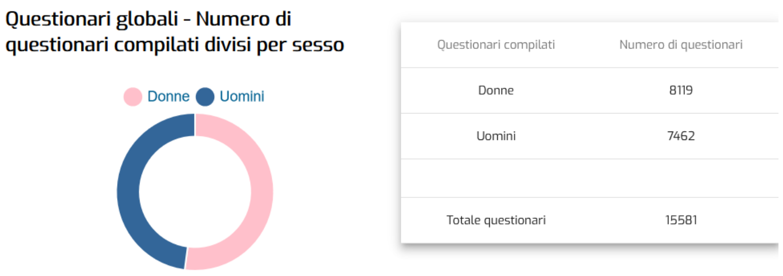
- Average number of questionnaires per patient: to collect this information, the total number of questionnaires of a given type was divided by the total number of patients on the platform. A subset of the type of questionnaires was chosen to be displayed on the chart.

- Average score of questionnaires: in this analysis, only a subset of the questionnaires can be considered, due to the fact that most questionnaires don't have a score attached to them. For each type of questionnaire with attached score, the average score was computed, and included in a chart.



Questionario	Numero di quesitonari
HBI	1728
MIAH-CD	1594
MIAH-UC	1384
MMAS8	494
PATIENT-SCCAI	1539
PHQ9	539
IBD-DISK	2999
CLINICAL-HBI	555
CLINICAL-SCCAI	505
CLINICAL-PRISM	1060
IPAQ-SF	518
PRISM	1127
TSQM	465
WPAI	470

Figure 4.6. Number of compiled questionnaires per type


Questionari globali - Numero di questionari compilati divisi per sesso
● Donne ● Uomini

Questionari compilati	Numero di questionari
Donne	8119
Uomini	7462
Totale questionari	15581

Figure 4.7. Number of questionnaires compiled by male/female patients

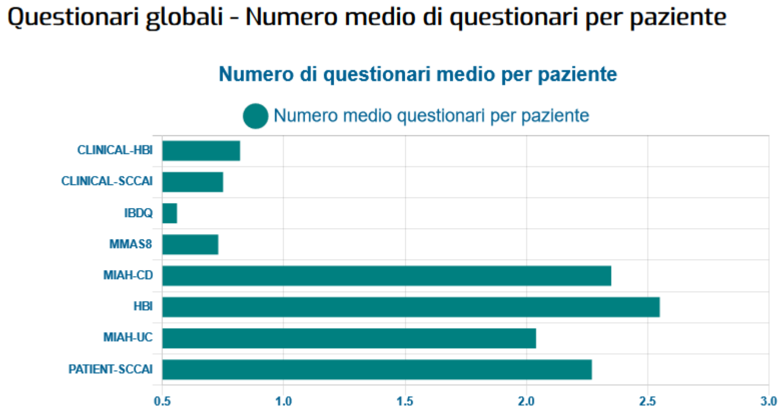


Figure 4.8. Average number of questionnaires compiled by patients

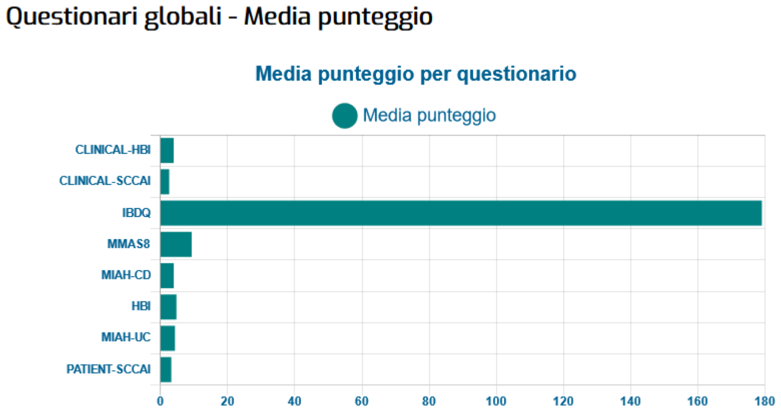


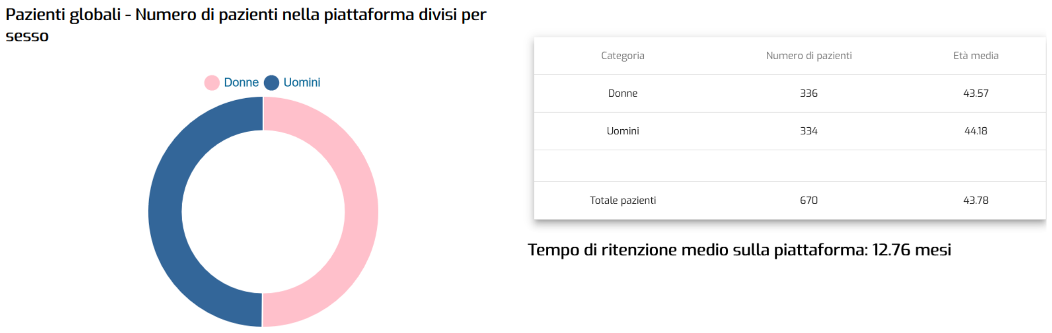
Figure 4.9. Average score of questionnaires

4.2.3 Patients

Another analysis was necessary to provide overall information on the patients of the platform. In particular the requested information was:

- Number of male patients and average age.
- Number of female patients and average age.
- Average retention time on the platform, to be considered as the average number of months that patients have been on the platform.

All this information was provided on a single page, using a doughnut chart to display the number of male/female patients and a table to provide the necessary data.



Pazienti globali - Numero di pazienti nella piattaforma divisi per sesso

Categoria	Numero di pazienti	Età media
Donne	336	43.57
Uomini	334	44.18
Totale pazienti	670	43.78

Tempo di ritenzione medio sulla piattaforma: 12.76 mesi

Figure 4.10. Number of male/female patients, average age of male/female patients and average retention time on the platform

4.2.4 Descriptive analysis on health status of patients

An important part of the new functionalities was to provide doctors with an analysis of the evolution of patients' health status. In order to assess the health status of a patient, scores of questionnaires are considered. In patients affected by Crohn's disease, the reference questionnaire considered is the HBI, for the patients affected by ulcerative colitis, the reference questionnaire considered is the SCCAI.

In this analysis, patients belong to one of three categories:

- improving: the health status of the patient is improving in the last period of time.

- deteriorating: the health status of the patient is deteriorating compared to the previous period.
- stabilizing: no significant changes in the health status of the patient have been found.

In order to understand the category where patients belong, the scores of the last two questionnaires, HBI for Crohn's disease and SCCAI for ulcerative colitis, are compared. The health status of a patient is considered improving when the score of the last questionnaire compiled is 2 points lower than the score of the previous questionnaire. A deteriorating status is defined when the score of the last questionnaire compiled is 2 points greater than the previous one. A patient's health status is considered stabilizing if the last questionnaire compiled is neither two points greater nor two points lower than the previous one. These definitions apply both for HBI and SCCAI questionnaires.

Furthermore, other general definitions were applied to classify patients in other subgroups. Each of the three categories described before can be further divided in the different subgroups.

Patients affected by ulcerative colitis were further divided in two subgroups:

- relapse: patient is considered in relapse if the score of the last questionnaire is greater or equal to 5.
- remission: patient is considered in remission if the score of the last questionnaire is lower than 5.

Patients affected by Crohn's disease were further divided in 4 subgroups:

- mild activity: the status of the illness is considered mild if the score of the last questionnaire compiled by the patient is greater or equal to 5 and lower than 8.
- moderate activity: the status of the illness is considered moderate if the score of the last questionnaire compiled by the patient is greater or equal to 8 and lower than 16.
- severe activity: the status of the illness is considered sever if the score of the last questionnaire compiled by the patient is greater or equal to 16.

- remission: patient is considered in remission if the score of the last compiled questionnaire is lower than 5.

Given these considerations, the main focus of the analysis was to consider the evolution of the health status of patients from a set of different perspectives:

- stable and unstable: a general overview of the total number of patients divided in each category.
- age group: patients were divided in different age groups, considering four groups of people with ages ranging from 0 to 20 years, from 20 to 40 years, from 40 to 60 years and from 60 years onwards.
- sex: patients were divided based on sex.
- initial status of pathology: patients were divided based on the initial status of the pathology. For both diseases, the initial status of the illness is considered active if the score of the first questionnaire is bigger or equal to 5. Otherwise the initial status is considered inactive.
- type of therapy: patients were divided considering ongoing therapy. Patients of the platform can be either treated with 5-ASA-OS therapy, topic therapy, the oral steroid therapy GCS-OS, with immunosupressants or with biological drugs.
- Existence of previous surgeries: patients were divided considering the previous existence of surgeries.
- Duration of illness: patients were divided based on different duration of illness, considering four groups with ranges from 0 to 10 years, from 10 to 20 years, from 20 to 30 years and from 30 years onwards.

All this different perspectives were displayed on stacked bar charts, using the green colour to identify patients on remission, the red colour for relapse, light orange for mild activity, orange for moderate activity and red for severe activity. Hovering the mouse on the different bars shows the actual number of patients belonging to a specific bar.

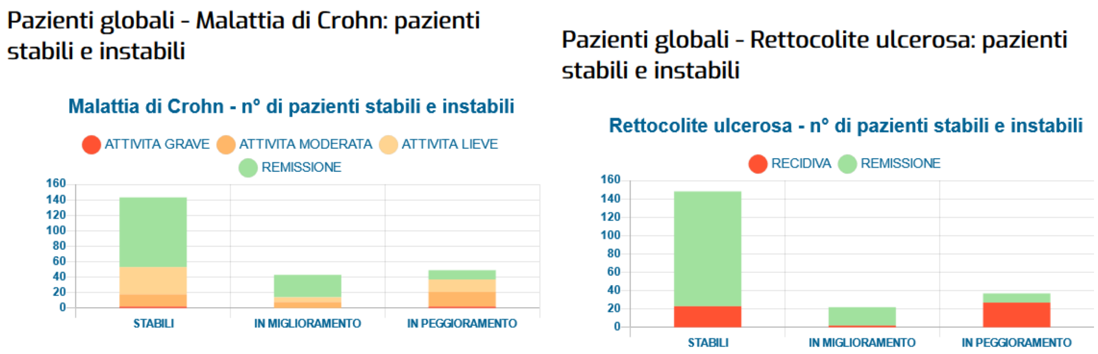


Figure 4.11. Stable and unstable patients.

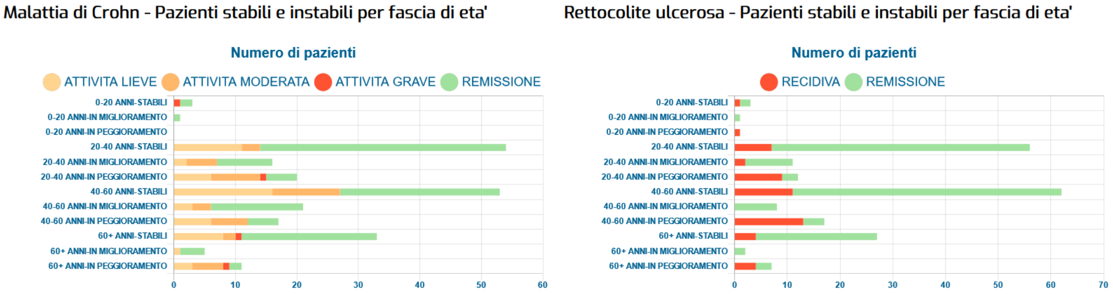
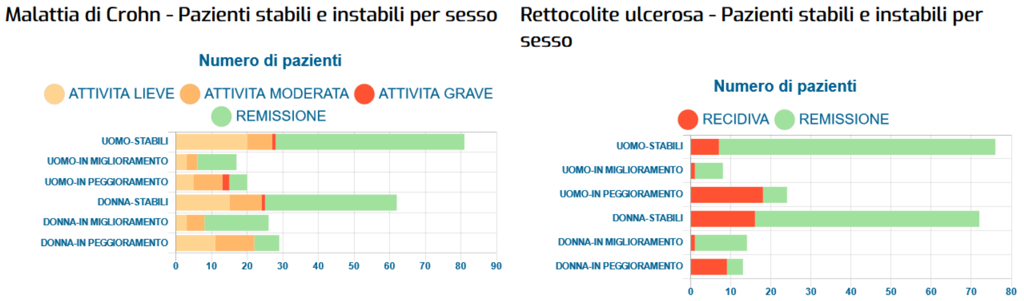


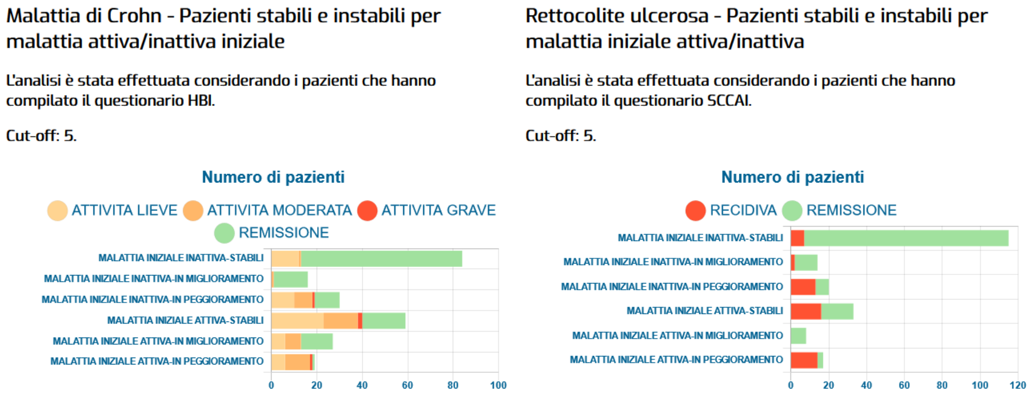
Figure 4.12. Stable and unstable patients per age group.



Malattia di Crohn - Pazienti stabili e instabili per sesso

Rettocolite ulcerosa - Pazienti stabili e instabili per sesso

Figure 4.13. Stable and unstable patients per sex.



Malattia di Crohn - Pazienti stabili e instabili per malattia attiva/inattiva iniziale

L'analisi è stata effettuata considerando i pazienti che hanno compilato il questionario HBI.

Cut-off: 5.

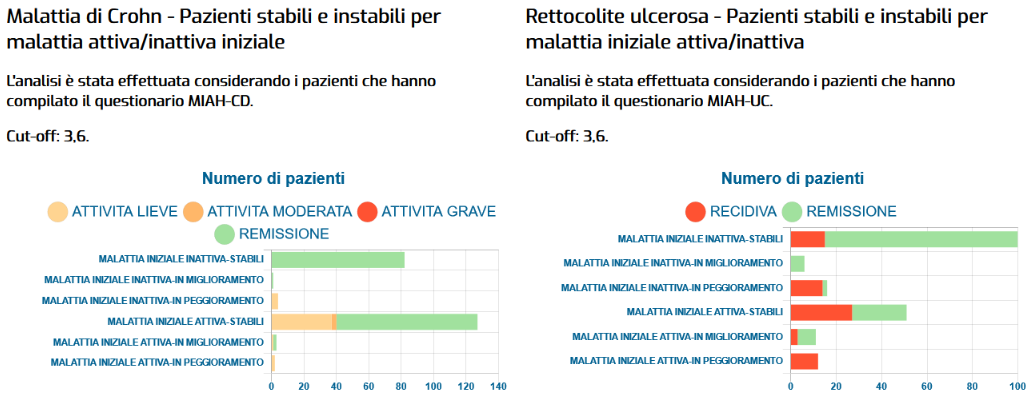
Rettocolite ulcerosa - Pazienti stabili e instabili per malattia iniziale attiva/inattiva

L'analisi è stata effettuata considerando i pazienti che hanno compilato il questionario SCCAI.

Cut-off: 5.

Numero di pazienti
ATTIVITA LIEVE
ATTIVITA MODERATA
ATTIVITA GRAVE
REMISSIONE
MALATTIA INIZIALE INATTIVA-STABILI
MALATTIA INIZIALE INATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE INATTIVA-IN PEGGIORAMENTO
MALATTIA INIZIALE ATTIVA-STABILI
MALATTIA INIZIALE ATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE ATTIVA-IN PEGGIORAMENTO
0
20
40
60
80
100
Numero di pazienti
RECIDIVA
REMISSIONE
MALATTIA INIZIALE INATTIVA-STABILI
MALATTIA INIZIALE INATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE INATTIVA-IN PEGGIORAMENTO
MALATTIA INIZIALE ATTIVA-STABILI
MALATTIA INIZIALE ATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE ATTIVA-IN PEGGIORAMENTO
0
20
40
60
80
100
120

Figure 4.14. Stable and unstable patients per initial status of pathology, questionnaires HBI and SCCAI



Malattia di Crohn - Pazienti stabili e instabili per malattia attiva/inattiva iniziale

L'analisi è stata effettuata considerando i pazienti che hanno compilato il questionario MIAH-CD.

Cut-off: 3,6.

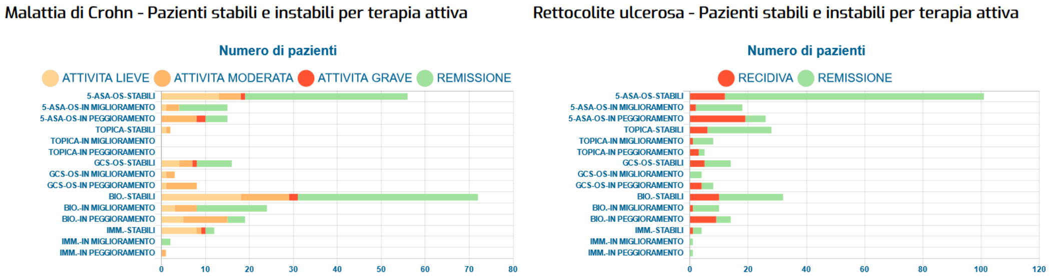
Rettocolite ulcerosa - Pazienti stabili e instabili per malattia iniziale attiva/inattiva

L'analisi è stata effettuata considerando i pazienti che hanno compilato il questionario MIAH-UC.

Cut-off: 3,6.

Numero di pazienti
ATTIVITA LIEVE
ATTIVITA MODERATA
ATTIVITA GRAVE
REMISSIONE
MALATTIA INIZIALE INATTIVA-STABILI
MALATTIA INIZIALE INATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE INATTIVA-IN PEGGIORAMENTO
MALATTIA INIZIALE ATTIVA-STABILI
MALATTIA INIZIALE ATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE ATTIVA-IN PEGGIORAMENTO
0
20
40
60
80
100
120
140
Numero di pazienti
RECIDIVA
REMISSIONE
MALATTIA INIZIALE INATTIVA-STABILI
MALATTIA INIZIALE INATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE INATTIVA-IN PEGGIORAMENTO
MALATTIA INIZIALE ATTIVA-STABILI
MALATTIA INIZIALE ATTIVA-IN MIGLIORAMENTO
MALATTIA INIZIALE ATTIVA-IN PEGGIORAMENTO
0
20
40
60
80
100

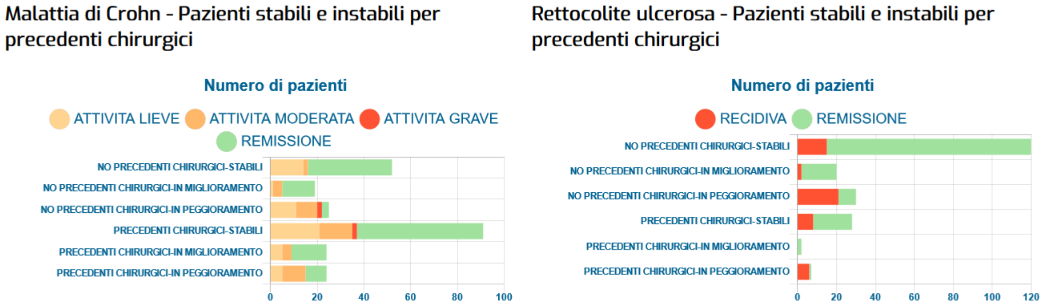
Figure 4.15. Stable and unstable patients per initial status of pathology, questionnaires MIAH-CD and MIAH-UC



Malattia di Crohn - Pazienti stabili e instabili per terapia attiva

Rettocolite ulcerosa - Pazienti stabili e instabili per terapia attiva

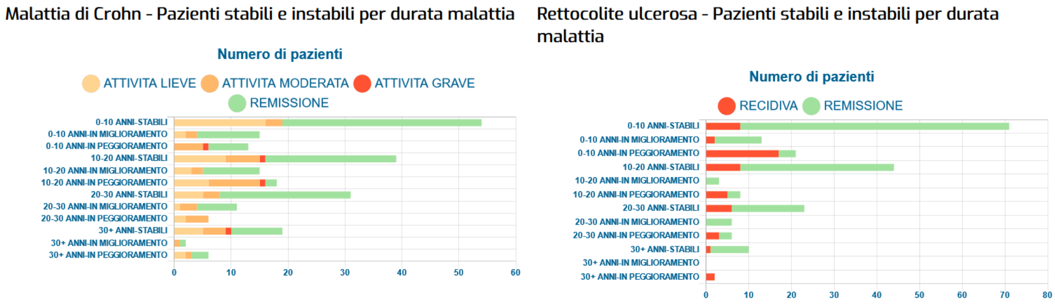
Figure 4.16. Stable and unstable patients per ongoing therapy



Malattia di Crohn - Pazienti stabili e instabili per precedenti chirurgici

Rettocolite ulcerosa - Pazienti stabili e instabili per precedenti chirurgici

Figure 4.17. Stable and unstable patients per existence of previous surgeries



Malattia di Crohn - Pazienti stabili e instabili per durata malattia

Rettocolite ulcerosa - Pazienti stabili e instabili per durata malattia

Figure 4.18. Stable and unstable patients per duration of illness

4.2.5 Satisfaction questionnaire

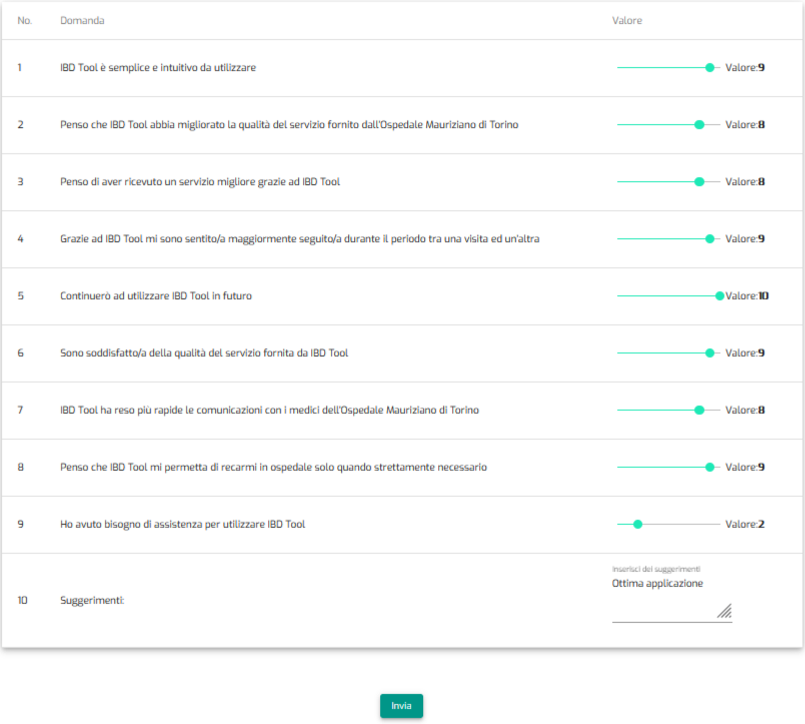
In order to study the system's capability to reach the patients' needs, a new questionnaire was created to evaluate the general degree of satisfaction on the use of the platform. This new questionnaire was developed based on the TUQ[13] and the TSQ[14] questionnaires for the evaluation of usability and satisfaction on telehealth and telemedicine systems. The satisfaction questionnaire used is composed of ten questions, nine of which accept a numerical value ranging from 1 to 10 as an answer, and an empty box where users can describe suggestions for the improvement of the platform.

The questions were chosen trying to address some specific functionalities:

- Ease of use:
  - I need assistance to use IBD Tool.
  - IBD Tool is easy and intuitive to use.
- Usefulness:
  - I think IBD Tool lets me go to the hospital only when strictly necessary.
- Quality of care provided:
  - I think that IBD Tool improved the quality of service offered by Mauriziano Hospital of Turin.
  - I think the service I obtained was better thanks to IBD Tool.
  - Thanks to IBD Tool, I felt more followed between a visit and another.
- Interaction quality:
  - IBD Tool made the communication with Mauriziano Hospital's clinicians faster.
- Satisfaction and future use:
  - I will continue to use IBD Tool in the future.
  - I am satisfied of the quality of service offered by IBD Tool.

The satisfactory questionnaire was added to the patient's home page as a functionality to let patients express their satisfaction on the usage of the platform. This section can be accessed anytime the patient wants to give opinions and servers as a good feedback to understand what can be improved on the existing platform.

On the server side, the satisfaction questionnaire was added to the collection containing the overall questionnaires, modeled with a list of integer values to store the answers of the questionnaire and a string for the provided suggestion.



No.	Domanda	Valore
1	IBD Tool è semplice e intuitivo da utilizzare	Valore:9
2	Penso che IBD Tool abbia migliorato la qualità del servizio fornito dall'Ospedale Mauriziano di Torino	Valore:8
3	Penso di aver ricevuto un servizio migliore grazie ad IBD Tool	Valore:8
4	Grazie ad IBD Tool mi sono sentito/a maggiormente seguito/a durante il periodo tra una visita ed un'altra	Valore:9
5	Continuerò ad utilizzare IBD Tool in futuro	Valore:10
6	Sono soddisfatto/a della qualità del servizio fornita da IBD Tool	Valore:9
7	IBD Tool ha reso più rapide le comunicazioni con i medici dell'Ospedale Mauriziano di Torino	Valore:8
8	Penso che IBD Tool mi permetta di recarmi in ospedale solo quando strettamente necessario	Valore:9
9	Ho avuto bisogno di assistenza per utilizzare IBD Tool	Valore:2
10	Suggerimenti:	Inserisci dei suggerimenti Ottima applicazione

Figure 4.19. Satisfaction questionnaire

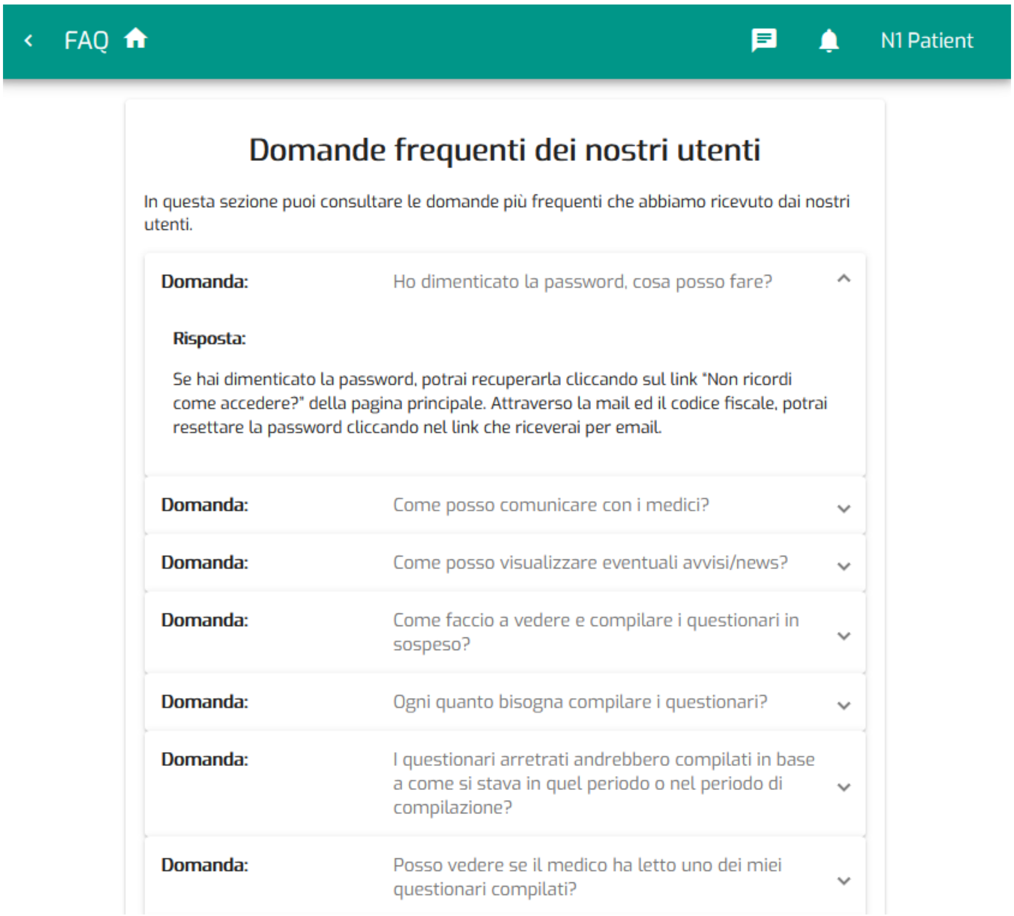
4.2.6 Frequent Asked Question section

Analysing the results of the satisfaction questionnaire, in particular the feedback provided by the suggestions, the patients of the platform showed a general appreciation for the service provided but also some questions about how certain functionalities of the application should be used.

These questions involved for the most part some common features of the application, for example what to do when the password is forgotten, how to compile a questionnaire or how to contact the doctors.

This suggestions point out that, even though the application does a good work at providing the general functionalities, some improvements can be made to help the patients understand how to generally use the application. Particularly for patients with less digital skills, another way to describe the use of the functionalities of the application could be very important to reach all the patients of the platform.

Discussing with the doctors of the platform, it was decided the creation of a Frequent Asked Questions section, where to describe through text the use of the general functionalities of the application, and in particular, to answer the most common questions provided by the patients in the satisfaction questionnaire and in the institutional email.



Domande frequenti dei nostri utenti

In questa sezione puoi consultare le domande più frequenti che abbiamo ricevuto dai nostri utenti.

Domanda: Ho dimenticato la password, cosa posso fare?

Risposta:

Se hai dimenticato la password, potrai recuperarla cliccando sul link "Non ricordi come accedere?" della pagina principale. Attraverso la mail ed il codice fiscale, potrai resettare la password cliccando nel link che riceverai per email.

Domanda: Come posso comunicare con i medici?

Domanda: Come posso visualizzare eventuali avvisi/news?

Domanda: Come faccio a vedere e compilare i questionari in sospeso?

Domanda: Ogni quanto bisogna compilare i questionari?

Domanda: I questionari arretrati andrebbero compilati in base a come si stava in quel periodo o nel periodo di compilazione?

Domanda: Posso vedere se il medico ha letto uno dei miei questionari compilati?

Figure 4.20. F.A.Q. section

The eight most common questions asked by the patients were collected and presented in this page. The section is modeled in a Question and Answer format, where patients can search for the question they have and look up the provided answer.

The logic of the section is built on the front-end, with static questions and answers. The general idea is that there is no need to create a logic on the server side when the questions can be updated anytime some new Q&A needs to be added by simply updating the front-end of the page.

The section was built using the ”expansion panel”, an Angular structure that allows to create a list of panels that can be expanded by clicking on them. The question was inserted in the description of the panel in order to visualize it regardless of the expansion; meanwhile the answer was inserted in the header of the panel to be visualized only upon expansion.

Chapter 5

Conclusions and future improvements

With the work of this thesis, the platform has gained more stability and scalability, creating a more consistent and reliable web service and a more efficient system that allows to keep adding more functionalities and more patients in the future. The new dashboard provides a wide variety of analysis of the data on the overall platform, showing different points of view of the health status of patients and allowing doctors to monitor the overall efficiency of the system. The development of a F.A.Q. section was a first step to improve the usability of the application and to meet the most common issues of the users with the usage of its features.

The IBD tool project is still growing, and some more improvements have already been defined for the future of the platform:

- addition of video tutorials to explain the use of the application, to help patients understand the most common operations and how they work.
- patient side dashboard, with the possibility for each patient to visualize the evolution of the illness.
- informative section, where the patient can learn some information on the illness.
- implementation of machine learning techniques to study the data of the platform.

IBD tool is already a very complete platform, which is adding more and more functionalities to improve the potentiality of the system for providing better quality of care. The involvement of patients and doctors with the platform already shows the effectiveness of the system and a general satisfaction with its features. This trend shows the potentiality of telemedicine systems, such as IBD tool, to provide an integration to the treatment process of patients with chronic diseases, for its ease of use and capacity to remotely connect doctors and patients.

Bibliography

[1] Koonin LM, Hoots B, Tsang CA, et al. Trends in the Use of Telehealth During the Emergence of the COVID-19 Pandemic — United States, January–March 2020. MMWR Morb Mortal Wkly Rep 2020;69:1595–1599. DOI: http://dx.doi.org/10.15585/mmwr.mm6943a3external icon

[2] Innova per l'Italia. [2020-03-26]. La tecnologia e l'innovazione per la lotta al Coronavirus: Le fast call https://innovaperlitalia.agid.gov.it/call2action/index.html.

[3] Omboni S. Telemedicine During the COVID-19 in Italy: A Missed Opportunity?. Telemed J E Health. 2020;26(8):973-975. doi:10.1089/tmj.2020.0106

[4] https://www.crohnscolitisfoundation.org/sites/default/files/2019-02/Updated IBD Factbook.pdf visited on 15/1/2022

[5] Baumgart DC, Sandborn WJ. Crohn's disease. Lancet. 2012 Nov 3;380(9853):1590-605. doi: 10.1016/S0140-6736(12)60026-9. Epub 2012 Aug 20. Erratum in: Lancet. 2013 Jan 19;381(9862):204. PMID: 22914295.

[6] Cosnes J. Tobacco and IBD: relevance in the understanding of disease mechanisms and clinical practice. Best Pract Res Clin Gastroenterol. 2004 Jun;18(3):481-96. doi: 10.1016/j.bpg.2003.12.003. PMID: 15157822.

[7] https://www.niddk.nih.gov/health-information/digestive-diseases/ulcerative-colitis visited on 19/1/2022.

[8] Sachar DB. The problem of postoperative recurrence of Crohn's disease. Med Clin North Am. 1990 Jan;74(1):183-8. doi: 10.1016/s0025-7125(16)30594-6. PMID: 2404175.

[9] Dahlhamer JM, Zammitti EP, Ward BW, Wheaton AG, Croft JB. Prevalence of Inflammatory Bowel Disease Among Adults Aged >=18 Years — United States, 2015. MMWR Morb Mortal Wkly Rep 2016;65:1166–1169. DOI: http://dx.doi.org/10.15585/mmwr.mm6542a3

[10] Nguyen GC, Chong CA, Chong RY. National estimates of the burden of inflammatory bowel disease among racial and ethnic groups in the United States. J Crohns Colitis. 2014 Apr;8(4):288-95. doi: 10.1016/j.crohns.2013.09.001. Epub 2013 Sep 24. PMID: 24074875.

[11] Johan Burisch & Pia Munkholm (2015) The epidemiology of inflammatory bowel disease, Scandinavian Journal of Gastroenterology, 50:8, 942-951, DOI: 10.3109/00365521.2015.1014407

[12] Roth, Erin PhD1; Chard, Sarah PhD1; Eckert, Kevin J. PhD2; Cross, Ray Jr. MD, MS3 TELEmedicine for Patients With Inflammatory Bowel Disease (TELE-IBD): A Qualitative Assessment, American Journal of Gastroenterology: October 2018 - Volume 113 - Issue - p S376

[13] Development of the Telehealth Usability Questionnaire (TUQ) Bambang Parmanto, Allen Nelson Lewus, Jr., Kristin M. Graham, MS, and Marnie H. Bertolet

[14] Yip MP, Chang AM, Chan J, MacKenzie AE. Development of the Telemedicine Satisfaction Questionnaire to evaluate patient satisfaction with telemedicine: a preliminary study. J Telemed Telecare. 2003;

[15] https://www.igibdscores.it/it/info-hbi.html visited on 19/1/2022