

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria del Cinema
e dei Mezzi di Comunicazione



Tesi di Laurea Magistrale

Progetto e implementazione di un cruscotto interattivo a supporto dell'analisi dinamica di una supply chain

Relatori

Prof. Giovanni Malnati

Prof. Daniele Apiletti

Candidato

Simone SEGALINI

ANNO ACCADEMICO 2019-2020

Sommario

La circolazione globale delle merci pone sfide operative complesse nella gestione delle filiere produttive: la necessità di operare just in time in modo flessibile ed elastico, contenendo i costi di produzione per restare competitivi sul mercato, impone la rivisitazione del modello di supply chain da semplice insieme di attori debolmente correlati a partner che cooperano per il perseguimento di un obiettivo comune. A tale scopo la condivisione in tempo reale dei dati di processo tra i diversi attori diventa fondamentale. Nell'ambito di questa tesi è stata sviluppata un'interfaccia utente web che consente l'esplorazione strutturata dei dati condivisi e la presentazione di indici di prestazioni che consentono il monitoraggio complessivo della catena e abilitano scelte operative in reazione ai cambiamenti dinamici che questa subisce. L'esigenza di gestire dati massivi, coniugata con la complessità del dominio che presenta molteplici sfaccettature, ha portato all'individuazione di un modello di rappresentazione degli eventi innovativo capace di veicolare in forma sintetica il comportamento del sistema osservato, permettendo al tempo stesso di scendere ad un livello di dettaglio sufficientemente basso per poter operare scelte concrete. In particolare, facendo leva sullo standard internazionale GS1 per la codifica degli eventi nei processi produttivi, è stato costruito un cruscotto interattivo per la piattaforma XTap che si propone come motore generalizzato a supporto delle supply chain. Particolare attenzione è stata data alla progettazione della user experience complessiva allo scopo di facilitare l'interazione dei soggetti con la piattaforma e permettere la massima efficacia ed efficienza nei processi di analisi e governo dei dati raccolti. Sul piano tecnologico, il lavoro è stato sviluppato utilizzando la tecnologia React ed il framework Material-UI allo scopo di veicolare una presentazione agile e moderna in linea con le aspettative del mercato. Il sistema è stato validato grazie all'utilizzo di un simulatore di eventi che ha consentito di popolare la piattaforma con dati simili a quelli provenienti dal campo, sia in termini semantici sia in termini quantitativi consentendo così di valutarne le prestazioni e la robustezza complessiva. I risultati ottenuti hanno mostrato come la piattaforma sia in grado di scalare sulle dimensioni richieste da questo tipo di applicazione e come la rappresentazione dei dati sia abilitante per l'attivazione di processi di ottimizzazione della supply chain.

Abstract

The global circulation of goods poses complex operational challenges in the management of production chains: the need to operate just in time in a flexible and elastic way, containing production costs in order to remain competitive on the market, it requires the review of the supply chain model from a simple set of loosely related actors to partners who cooperate for the pursuit of a common goal. For this purpose, real-time sharing of process data between the various actors becomes fundamental. As part of this thesis, a web user interface was developed which allows the structured exploration of shared data and the presentation of performance indices that allow the overall monitoring of the chain and enable operational choices in reaction to the dynamic changes it undergoes. The need to manage massive data, combined with the complexity of the domain that has multiple facets, has led to the identification of an event representation innovative model capable of conveying the behavior of the observed system in a synthetic form, while allowing it to descend at a sufficiently low level of detail to be able to make concrete choices. In particular, leveraging on the GS1 international standard for the coding of events in production processes, an interactive dashboard was built for the XTap platform which is proposed as a generalized engine to support supply chains. Particular attention was given to the design of the overall user experience in order to facilitate the interaction of subjects with the platform and allow maximum effectiveness and efficiency in the analysis and governance of the collected data. On a technological level, the work was developed using React technology and the Material-UI framework in order to convey an agile and modern presentation in line with market expectations. The system was validated thanks to the use of an event simulator that made it possible to populate the platform with data similar to those ones from the field, both in semantic and quantitative terms, thus allowing to evaluate its overall performance and robustness. The results obtained showed how the platform is able to scale to the size required by this type of application and how the representation of the data is enabling for the activation of supply chain optimization processes.

Indice

Elenco delle figure	6
1 Introduzione	9
2 GS1	11
2.1 Lo standard EPCIS	11
2.2 Processo di business e EPCIS	12
2.3 EPCIS e altre categorie di dati	13
2.4 Infrastruttura aziendale e EPCIS	13
2.5 Tipi di eventi e le quattro W	15
2.6 La forza di EPCIS e i Master Data	19
2.7 GTIN e GLN	20
3 Applicazioni Web	22
3.1 Storia delle Applicazioni Web	22
3.2 SPA vs MPA	24
3.3 React	24
3.3.1 La scelta di React	25
3.3.2 JavaScript	26
3.4 React concetti chiave	26
3.4.1 JSX - JavaScript Syntax Extension	28
3.4.2 Componenti	28
3.4.3 Il ciclo di vita di un componente	29
3.5 React Hooks	30
3.5.1 React Hooks lifecycle	31
3.5.2 Regole degli Hooks	31
3.5.3 useState e useReducer	32
4 User Experience Design	35
4.1 Usabilità	35
4.2 UCD - User Centered Design	37

4.3	User Experience	39
4.3.1	Il modello di Garrett	40
4.4	Material Design	42
4.4.1	Material UI	43
5	Implementazione di XTap	46
5.1	Architettura dell'applicazione	46
5.1.1	Autenticazione	47
5.1.2	Routing	51
5.1.3	Struttura generale	52
5.2	Analisi delle pagine	55
5.2.1	Dashboard	55
5.2.2	Product History	59
5.2.3	Production Flow	63
5.2.4	KPI Analytics	68
5.2.5	My Company	70
5.2.6	My Items	71
5.2.7	My Places	73
5.2.8	Companies e Others' Items	75
5.2.9	Event Models e Create New Event Template	75
5.2.10	Graph e Create Relationship	77
6	Conclusioni	82
6.1	Conoscenze Acquisite	83
6.2	Lo stato attuale	83
6.3	Sviluppi futuri	84
A	Icone Business Steps	86
	Bibliografia	88

Elenco delle figure

2.1	Semplice processo di business	12
2.2	EPCIS all'interno dell'infrastruttura di un'azienda	14
2.3	Formato GLN	21
3.1	Esempio sistema client-server	23
3.2	Logo React	24
3.3	Grafico opinioni degli sviluppatori	25
3.4	Esempio di struttura a componenti	27
3.5	Lifecycle di un componente React	30
3.6	React Hooks lifecycle	32
4.1	Le tre dimensioni dell'usabilità	36
4.2	Timer sui telefoni Apple	37
4.3	Processo User Centered Design per i sistemi interattivi	39
4.4	Diagramma UX Honeycomb di Peter Morville	40
4.5	Struttura del modello di Garrett	41
4.6	Icona Delete nei diversi temi	43
5.1	Schermata Login di XTap	46
5.2	Schermata Login dati errati	48
5.3	Layout di XTap	53
5.4	Menù di XTap	54
5.5	Secondo tema di Xtap	54
5.6	Dashboard di XTap	55
5.7	Layout della dashboard	56
5.8	Dettaglio di un evento	56
5.9	Grafico dimensione When	57
5.10	Filtro dimensione When	57
5.11	Dimensione Where	58
5.12	Tabella dimensione Why	58
5.13	Dashboard e filtri	59

5.14	Tabella Product History	60
5.15	Gruppi When e Why	60
5.16	Product History rappresentazione a grafo	61
5.17	Product History dettagli evento	62
5.18	Pag. Production Flow	63
5.19	Componente Timeline/Control	64
5.20	Espansione dei nodi	65
5.21	Simulazione caricamento	66
5.22	Filtro prodotti	66
5.23	Indicatore globale KPI	68
5.24	Indicatori KPI della supply chain	70
5.25	Pag. My Company	71
5.26	Pag. My Items	72
5.27	Dettaglio prodotto	72
5.28	Pag. Add Item	73
5.29	Pag. My Places	74
5.30	Pag. New Place	74
5.31	Pag. Event Models	76
5.32	Pag. Event Detail	76
5.33	Pag. Create New Event Template	78
5.34	Pag. Supply Chain	79
5.35	Pag. Nuova Relazione	79
5.36	Pag. Supply Chain con nuova relazione	80
6.1	Serious game XTap	85
6.2	Wireframe mobile app	85
A.1	Icone Business Steps	87

Capitolo 1

Introduzione

In un mercato globalizzato e condizionato da fattori economici importanti, la competizione tra aziende non riguarda solo la qualità e il costo del prodotto, ma anche l'offerta di una filiera trasparente e certificata. Occorre superare l'idea, purtroppo ancora diffusa, che la trasparenza dei processi e la tracciabilità dei prodotti siano esclusivamente obblighi derivanti da normative e/o da richieste dei clienti e che le nuove proposte offerte dalla tecnologia portino solo a costi aggiuntivi. Nessuna impresa, oggi, lavora come soggetto indipendente, ma agisce all'interno di un sistema, che è la supply chain di appartenenza, formato da soggetti diversi i quali hanno la possibilità di condizionare, a più livelli e in modo diverso, l'intero ambiente e di conseguenza anche i singoli attori. Una supply chain non è un semplice processo lineare che porta un generico oggetto da un'azienda all'altra fino a raggiungere il consumatore finale, ma è una rete complessa di attori che si influenzano a vicenda. Non sempre le aziende hanno piena consapevolezza di questo fenomeno e si affidano a sistemi che gestiscono e ottimizzano i soli processi interni. Una visione dall'alto a livello di supply chain consente di monitorare la totalità di fattori che influenzano il processo produttivo consentendo decisioni più mirate e consapevoli.

La realizzazione della piattaforma XTap ha l'obiettivo di offrire una visione completa della filiera produttiva fornendo strumenti adatti a semplificare la raccolta di dati e la comunicazione di informazioni tra le imprese. L'utilizzo degli standard GS1 ed in particolare dello standard EPCIS, Electronic Product Code information Service, permette scambi di dati in modo efficace ed efficiente. EPCIS, grazie ad un linguaggio riconosciuto a livello globale e ad una descrizione basata sulle 4W, What, Why, Where e When, rende possibile la rappresentazione degli eventi che si verificano in tutte le attività dell'intera supply chain.

In questo lavoro di tesi è illustrata la prima fase dello sviluppo dell'applicazione web della piattaforma XTap, un sistema che mette a disposizione delle aziende diversi strumenti per il monitoraggio della filiera produttiva, per l'esplorazione dei

dati condivisi e per la valutazione dei processi attraverso indici di prestazione. Per lo sviluppo sono stati utilizzati la tecnologia React e il framework Material-UI.

L'applicazione è stata testata mediante un simulatore di eventi che ha permesso di popolare la piattaforma con dati congruenti ai dati della realtà e ha consentito la valutazione della robustezza e delle prestazioni.

I primi capitoli di questo testo sono dedicati alle informazioni teoriche sugli standard e sulle tecnologie utilizzate, e all'usabilità e soddisfazione nell'esperienza dell'utente. La descrizione del lavoro inizia con l'analisi dell'architettura dell'applicazione e dedica importanti spazi di approfondimento alle pagine e agli strumenti messi a disposizione dell'utente. L'ultimo capitolo riguarda le conclusioni e le conoscenze acquisite, una verifica dello stato attuale e gli sviluppi e i miglioramenti futuri.

Capitolo 2

GS1

2.1 Lo standard EPCIS

La piattaforma XTap ha l'obiettivo di offrire una soluzione per la comunicazione dei dati tra le aziende. Per la sua realizzazione si è scelto di utilizzare gli standard GS1 (Global Standards).

GS1 è un'associazione senza scopo di lucro fondata nel 1977 e conosciuta principalmente per i codici a barre, la quale agisce per facilitare la collaborazione e il dialogo tra le aziende.

Uno degli standard GS1 è l' EPCIS (Electronic Product Code Information Services) progettato con la finalità di creare e condividere dati sugli eventi sia all'interno dell'azienda che tra aziende diverse e garantire la tracciabilità del prodotto in tempo reale.[1]

Oggi, rispetto al passato, si rivela sempre più importante offrire la visibilità dell'intera supply chain, anche se ciò non avviene ancora in tutti i settori. Tracciando un processo di spedizione si nota come le informazioni sono immediate e precise, e queste funzioni sono attese e, a volte, richieste da sempre più soggetti: consumatori, organizzazioni, governi.

Lo standard EPCIS permette la condivisione delle informazioni sull'intera vita di un prodotto cioè sul suo percorso lungo la filiera, dal produttore al consumatore finale.

EPCIS, rispondendo alle domande What, Where, When e Why, offre la possibilità di tracciare un prodotto o un insieme di prodotti registrando ogni evento, ad esempio il passaggio da un punto all'altro della filiera, e di immagazzinare le informazioni rendendole condivisibili a tutti i soggetti della supply chain.

È indispensabile utilizzare l' EPCIS insieme allo standard GS1 CBV (Core Business Vocabulary) il quale è un vocabolario standardizzato che indica l'elenco di valori

possibili al fine di compilare le strutture dello standard EPCIS.[2] Ciò è fondamentale per garantire tra aziende diverse l'interoperabilità tra sistemi differenti e l'uniformità di significato di termini e concetti. A tale scopo i termini proposti all'interno del CBV sono generici in modo tale da poter essere utilizzati nel processo di qualsiasi azienda a prescindere dall'attività produttiva svolta.

2.2 Processo di business e EPCIS

I dati EPCIS forniscono la posizione e i transiti pregressi del prodotto all'interno del processo di business.

Un articolo è prodotto nella fabbrica, poi inviato ad un centro di distribuzione e lì ricevuto e spedito ad un negozio, il quale prima lo riceve in magazzino e dopo lo posiziona nel settore di vendita. Nell'esempio sottostante (Figura 2.1) sono evidenziati i punti in cui vengono generati dati EPCIS i quali registrano in modo dettagliato ogni passaggio.

Il singolo dato EPCIS che descrive il completamento di una fase aziendale (business step) è denominato evento EPCIS. Ogni business step produce un evento EPCIS, il cui contenuto è coerente con il business step, anche se tutti gli eventi hanno in comune la struttura a quattro dimensioni (4W).

L'insieme di eventi EPCIS in progressione consente una visione dettagliata dei tempi e degli spazi utilizzati in un processo di business.

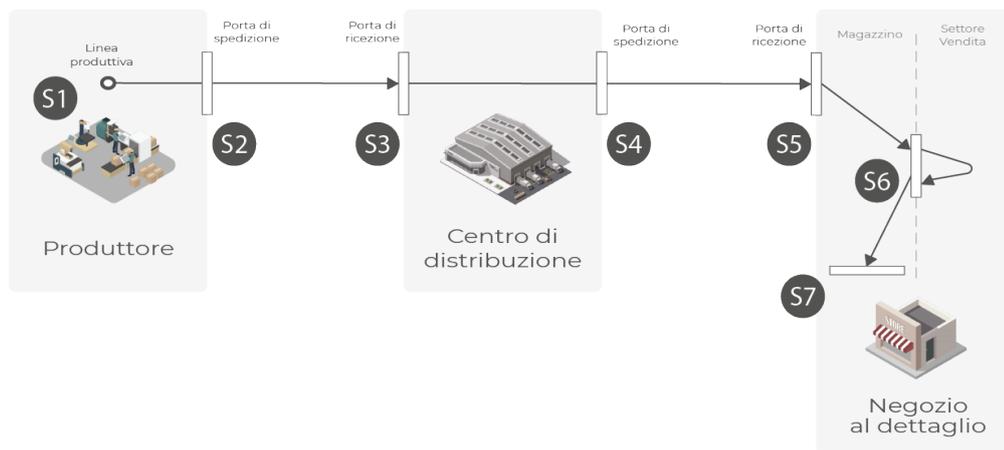


Figura 2.1: Semplice processo di business

2.3 EPCIS e altre categorie di dati

Lo standard EPCIS fa parte del gruppo “Share” degli standards GS1 che si riferiscono a tre categorie di dati:

- Master Data, dati condivisi tra un partner commerciale e altri partner che offrono attributi descrittivi di entità riconosciute attraverso chiavi identificative GS1.
- Transaction Data, dati che, utilizzando chiavi identificative GS1, descrivono operazioni inerenti la conferma dell'esecuzione di una mansione all'interno del processo di business come indicato dall'accordo commerciale.
- Visibility Data, dati che descrivono dettagli sull'attività di prodotti all'interno della supply chain o all'interno dell'azienda. Questi dati sono identificati da chiavi che specificano dove il prodotto si trova nel tempo e il motivo per cui ciò avviene, non solo all'interno dell'azienda, ma anche tra aziende diverse.

I visibility data sono generati in quantità maggiore rispetto alle altre due categorie in quanto anche un'unica transazione commerciale (produttore - venditore) può essere soggetta a diversi eventi di visibilità.

I visibility data sono complementari ai transaction data poichè, mentre alcune transazioni commerciali, ad esempio l'invio di un ordine, avvengono senza movimentazione di prodotti, al contrario, alcuni eventi di visibilità, ad esempio il trasferimento dal magazzino al negozio, hanno luogo senza alcuna transazione commerciale.

2.4 Infrastruttura aziendale e EPCIS

L'analisi e la descrizione delle diverse categorie di dati elaborati dallo standard EPCIS permette di comprendere come esso viene incluso in una consueta struttura aziendale di Information Technology.

I componenti che trattano master data e transaction data sono riuniti come “Applicazioni Back-End”, mentre per i visibility data, che sono spesso in quantità maggiori, è consuetudine utilizzare per l'elaborazione componenti dedicati (Figura 2.2). Tali componenti, partendo dal basso, sono:

- EPCIS Capture Application (applicazione di acquisizione EPCIS). Sono applicazioni software che generano eventi EPCIS ogniqualvolta viene completato un processo aziendale.
- EPCIS Repository. È un archivio permanente che contiene tutti gli eventi EPCIS generati dall'azienda e tutti gli eventi ricevuti dai partners.

- Accessing Application EPCIS (applicazioni di accesso EPCIS). Sono applicazioni diverse che eseguono attività a livello aziendale e che elaborano gli eventi EPCIS.

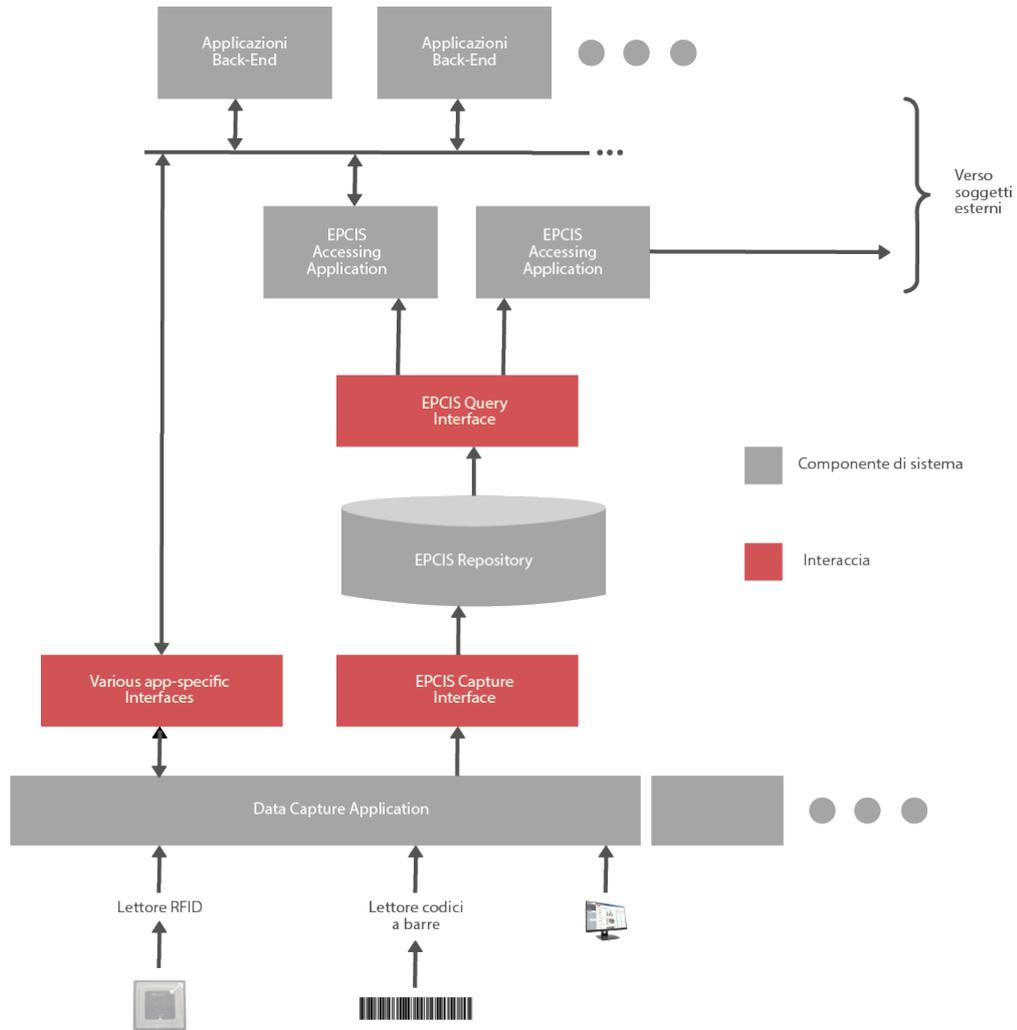


Figura 2.2: EPCIS all'interno dell'infrastruttura di un'azienda

Mediante EPCIS Capture Interface (interfaccia di cattura) le relative applicazioni inviano eventi EPCIS alle applicazioni di accesso oppure ad un repository, mentre attraverso EPCIS Query Interface le applicazioni di accesso recuperano i dati dal repository per l'utilizzo interno o per inviarli a partners commerciali.

2.5 Tipi di eventi e le quattro W

Una prima categorizzazione distingue quattro diverse tipologie di eventi EPCIS:

- **ObjectEvent.** Comprende qualunque evento avvenuto a uno o più oggetti;
- **AggregationEvent.** Rappresenta gli eventi nei quali uno o più oggetti sono fisicamente uniti o divisi l'uno dall'altro come ad esempio un certo numero di scatole legate su un pallet oppure rimosse da sopra un pallet;
- **TransformationEvent.** Descrive gli eventi in cui gli elementi in input vengono utilizzati in modo parziale o totale per la produzione di elementi output. A differenza degli Aggregation Event non hanno il percorso inverso, infatti non sono più ottenibili gli elementi di partenza;
- **TransactionEvent.** Rappresenta gli eventi in cui uno o più elementi sono uniti o divisi da uno o più identificativi di una transazione di business, come ad esempio il pallet e le scatole del prodotto e la fattura.

Le tipologie di eventi più utilizzate comprendono l' **ObjectEvent** e l'**AggregationEvent** i quali, uniti, rappresentano la maggioranza degli eventi che interessano un classico processo di business.

Gli **ObjectEvent**, gli **AggregationEvent** e i **TransactionEvent** condividono un attributo significativo chiamato "action", il quale indica che i loro eventi sono legati al ciclo di vita del prodotto. Tale attributo si riferisce a tre valori possibili: **ADD**, **DELETE** e **OBSERVE**.

Nel caso di **ObjectEvent** l'attributo prende il valore di **ADD** quando l'evento indica la creazione di un nuovo oggetto; di **DELETE** se l'evento indica la fine di ciclo di vita di un oggetto, considerando che nessun altro evento dovrebbe verificarsi successivamente e di **OBSERVE** per tutti gli altri casi.

Nel caso di **AggregationEvent** l'attributo assume il valore di **ADD** quando si tratta di aggregazione fisica, ad esempio l'imballaggio in un contenitore; di **DELETE** se l'evento indica la disaggregazione ad esempio l'operazione di disfacimento di un imballaggio al fine di prelevare i singoli colli e di **OBSERVE** nei casi in cui non avviene l'aggregazione e neppure la disaggregazione.

Nel caso di **TransactionEvent** l'attributo assume il valore di **ADD** se durante l'evento l'oggetto è associato alla transazione commerciale; di **DELETE** se durante l'evento l'oggetto viene svincolato dalla transazione commerciale e di **OBSERVE** se l'oggetto continua semplicemente ad essere associato alla transazione commerciale.

Ogni evento EPCIS, attraverso le quattro dimensioni **What**, **Why**, **Where** e **When**, descrive le diverse fasi di un processo aziendale. Con riferimento all'esempio del

processo di business illustrato nella sezione 2.2 (Figura 2.1) e analizzando in modo dettagliato lo step S3 è possibile comprendere cosa rappresenta ognuna delle quattro dimensioni. L'oggetto singolo, identificato da un GTIN (Global Trade Item Number), chiave usata da GS1 per identificare i prodotti con aggiunto un numero di serie, è inviato dal produttore e ricevuto dal centro di distribuzione. L'evento EPCIS corrispondente allo step S3 si struttura nelle quattro dimensioni:

- What, identifica il prodotto ricevuto utilizzando il GTIN e il numero di serie. Rappresenta, infatti, l'oggetto o gli oggetti coinvolti nell'evento;
- Why, fornisce informazioni sul contesto aziendale, identifica la fase del processo aziendale, nel contesto "receiving";
- Where, indica il luogo in cui il prodotto è stato ricevuto, in questo caso il centro di distribuzione. Se si fruisce di un trasporto ne è indicata l'origine e la destinazione;
- When, offre informazioni temporali sull'evento, nell'esempio citato indica il momento in cui il centro di distribuzione riceve l'oggetto.

Nella Tabella 2.1 sono riassunte le informazioni riguardanti l'evento EPCIS per lo step S3 i cui attributi verranno in seguito analizzati in modo più approfondito.

La dimensione What identifica gli oggetti fisici o digitali trattati nell'evento. Nessun evento EPCIS può avere questa dimensione vuota. Nello standard EPCIS un articolo è identificato a livello di istanza oppure a livello di classe. Nel primo caso il GTIN può essere integrato da un codice seriale e la combinazione prende il nome di SGTIN; nel secondo caso, quando l'identificativo è comune ad una quantità di oggetti, il GTIN è accompagnato dal numero di lotto e la combinazione forma un codice LGTIN nel quale sono indicate la quantità e, facoltativamente, l'unità di misura.

La dimensione Why di un evento EPCIS include diversi attributi i quali descrivono il contesto aziendale in cui è stato generato l'evento e quali motivi ne hanno avviato la generazione. Senza queste informazioni sarebbe possibile solamente sapere il luogo e il tempo della presenza di un prodotto, ma non si potrebbe conoscere cosa è effettivamente accaduto. Ciò conferma l'importanza di questa dimensione per rendere significativi gli eventi EPCIS, i quali definiscono almeno il Business Step, attributo che, per evitare l'insorgere di ambiguità, contiene un verbo definito all'interno del CVB.

Attributi presenti in questa dimensione:

- Business Step (BizStep) indica quale step del processo di business è in atto. Nell'esempio citato sopra "receiving". Nella Tabella 2.2 sono indicati alcuni valori di Business Step;

Dimensione	Data Element	Contenuti	Commenti
	Event Type	Object Event	
	Action	OBSERVE	
What	EPC List	La lista contiene solo un elemento: GTIN 10614141123459 Serial 12345	Identifica il prodotto che è stato ricevuto
When	Event Time	Nov 5, 2012, at 10:12am UTC	Il momento in cui il prodotto è stato ricevuto
	Event Time Zone Offset	-05:00	
Where	Read Point	GLN 5012345678900 Extension D123	Il luogo dove il prodotto è stato ricevuto, in questo caso una specifica porta del centro di distribuzione
	Business Location	GLN 5012345678900	Il luogo in cui il prodotto dovrebbe trovarsi dopo l'evento, in questo caso il centro di distribuzione
Why	Business Step	Receiving (from CBV)	Identificativo definito nel CBV per indicare che è un business step di ricezione
	Disposition	In Progress (from CBV)	Identificativo definito nel CBV per indicare che il prodotto si sta muovendo normalmente lungo la supply chain
	Business Transaction List	Lista contenente due business transaction: Purchase Order: GLN 50123450000015 PO# ABC123 Invoice: GLN 06141450000015 INV# XYZ987	Ogni transazione di business è qualificata con un GLN per renderla globalmente identificata e inoltre per identificare il sistema che ha generato il numero. *Purchase Order* e *Invoice* sono parole identificate dallo standard definite nel CBV per identificare i tipi di transazione di business.
	Source List	La lista contiene una solo source: Owning Party: GLN 06141450000015	Receiving è un step all'interno di un trasferimento globale di proprietà dalla source alla destination. La parte proprietaria della source è identificata dal suo GLN.
	Destination List	La lista contiene un solo destinatario: Owning Party: GLN 50123450000015	Receiving è un passaggio all'interno di un trasferimento globale di proprietà dalla source alla destination. Qui, la parte proprietaria della destination è identificata dal suo GLN.

Tabella 2.1: Contenuto dell'evento EPCIS per lo Step S3.

- Disposition segnala la condizione commerciale successiva all'evento dell'oggetto indicato nella dimensione What. Talvolta corrisponde ad un aggettivo che specifica la condizione commerciale in funzione del processo di business complessivo. Il CVB include un elenco di valori standard di disposition e la funzione prioritaria di questo attributo consiste nell'evidenziare dove emergono delle eccezioni. Nell'esempio "**in_progress**" significa che l'oggetto si muove in modo normale lungo la supply chain;
- Business Transaction List identifica una o più transazioni commerciali particolarmente importanti per l'evento. Ogni transazione è identificata da una coppia di identificatori, dei quali il primo indica a quale tipologia di transazione commerciale si fa riferimento, mentre il secondo si riferisce alla specifica transazione commerciale di quel tipo. Nell'esempio ci sono due transazioni rilevanti: l'ordine di acquisto del rivenditore e la fattura del produttore;
- Source List and Destination List offre un contesto aziendale supplementare quando un evento EPCIS fa parte di un trasferimento di proprietà, di una

situazione di responsabilità e/o custodia. Source e Destination sono contrassegnati da una coppia di identificatori. Nell'esempio la parte proprietaria della source è il produttore, mentre la parte proprietaria della destination è il rivenditore.

Business Steps	
Valore	Descrizione
<i>accepting</i>	Indica un'attività specifica all'interno di un processo di business in cui un oggetto cambia possesso e/o proprietà.
<i>commissioning</i>	Procedura di abbinamento di un identificatore a livello di istanza (esempio un EPC) a un oggetto specifico, oppure procedura di abbinamento di un identificatore a livello di classe, non ancora utilizzato in precedenza, con uno o più oggetti.
<i>decommissioning</i>	Procedura di dissociazione di un identificatore a livello di istanza (esempio un EPC) con un oggetto. L'oggetto può essere ripristinato in futuro solo con un nuovo identificatore a livello di istanza.
<i>inspecting</i>	Procedura di controllo degli oggetti per affrontare potenziali difetti fisici o della documentazione.
<i>receiving</i>	Indica un'attività specifica all'interno di un processo di business, tale attività indica che un oggetto è stato ricevuto in una posizione e viene inserito all'interno dell'inventario del destinatario. Utilizzando receiving si esclude l'utilizzo di arriving e accepting.
<i>shipping</i>	Indica il processo complessivo di staging_outbound, caricamento e partenza. Può essere utilizzato quando le informazioni sulla fase del processo più granulari sono sconosciute o inaccessibili. Può indicare un evento finale da un punto di spedizione. L'uso della spedizione si esclude a vicenda dall'uso di staging_outbound, departing o loading.

Tabella 2.2: Valori e descrizioni di alcuni Business Step.

La dimensione Where di un evento individua il luogo in cui si è verificato. Sono consentiti due tipi di attributi negli eventi EPCIS entrambi non obbligatori.

Essi sono il ReadPoint e la Business Location (bizLocation), il primo serve per indicare la posizione in cui l'evento si è verificato, il secondo, invece, indica il luogo in cui si trova l'oggetto fino al verificarsi di un evento successivo. per indicare le posizioni è possibile utilizzare i Global Location Number (GLN), un GLN più un seriale (SGLN) oppure le coordinate geografiche.

Le posizioni possono essere individuate con un livello di precisione molto elevato. Se si usa come termine di paragone un edificio è sufficiente utilizzare GLN, se invece è necessaria una maggiore precisione è necessario usare la chiave SGLN. In riferimento al processo di business illustrato nella sezione 2.2, nello step S3 il Read Point indica il luogo in cui avviene il ricevimento dell'articolo, ipoteticamente la piattaforma di carico del centro di distribuzione identificata da un SGLN (GLN + estensione), mentre la Business Location definisce il luogo dove l'articolo si trova dopo essere stato ricevuto, luogo che è identificato da un GLN.

La dimensione When indica quando un evento EPCIS si è verificato, viene richiesta in ogni evento e ha al suo interno tre attributi:

- Event Time, attributo obbligatorio che indica data e ora in cui si è verificato l'evento;
- Event Time Zone Offset, attributo obbligatorio che indica il fuso orario in vigore nell'ora e nel luogo in cui si è verificato l'evento. Permette alle applicazioni di visualizzare l'ora dell'evento rispetto all'ora locale;

- Record Time, attributo non obbligatorio che serve a indicare la data e l'ora in cui l'evento è registrato in un repository EPCIS.

Nel processo di business visto nella sezione 2.2 per lo step S3 l'Event Time segnala la data e l'ora dell'evento EPCIS in cui il prodotto è stato ricevuto, mentre l'Event Time Zone Offset indicherà il fuso orario del luogo in cui è avvenuto l'evento.

2.6 La forza di EPCIS e i Master Data

Il principale punto di forza dello standard EPCIS consiste nella possibilità di unire i singoli eventi generati nel tempo e lungo un processo aziendale o lungo una supply chain e sfruttare le opportunità che ne derivano. Diventa così possibile:

- individuare per un oggetto l'evento più recente, per conoscere il suo stato attuale e il luogo in cui si trova. Ciò è definito "tracking";
- analizzare una raccolta di eventi prelevati nel tempo in un luogo specifico o in un particolare processo aziendale. Ciò è definito "analysis";
- confrontare lo stato di un articolo in base ad un evento EPCIS attuale e le aspettative in funzione di una transazione commerciale o un evento EPCIS precedente. Ciò è definito "checking";
- automatizzare processi aziendali in funzione di cosa rivela un evento EPCIS riguardo alla conclusione di un determinato Business Step. In questo caso si parla di "automation".

Le opportunità sopra elencate rappresentano solo in parte le potenzialità effettive dello standard, infatti un'applicazione aziendale può utilizzare i dati EPCIS anche in funzione del raggiungimento di obiettivi diversi.

I Visibility Data, rappresentati in EPCIS utilizzando identificatori globalmente univoci per descrivere informazioni ovvero termini inseriti nel CBV per garantire l'interoperabilità, non sono a volte sufficienti per illustrare in modo approfondito cosa accade in un determinato evento EPCIS. Talvolta i Visibility Data non sono sufficienti per capire in maniera dettagliata cosa sta succedendo in un determinato evento EPCIS, quindi per fornire un contesto più dettagliato vengono messi in gioco i master data. I master data offrono la possibilità di contestualizzare in maniera più efficace le informazioni contenute in un evento e questi possono essere anche comunicati tra partner al di fuori di EPCIS. Mancando meccanismi sicuri e consolidati per la trasmissione di questi dati, lo standard permette di allegare i dati dell'evento EPCIS attraverso il campo Instance Lot Master Data che attraverso l'utilizzo di coppie chiave/valore e un namespace contiene una lista di dati

di contesto e anche eventuali informazioni aggiuntive (es. la data di scadenza del prodotto).

2.7 GTIN e GLN

All'interno della piattaforma XTap che è stata sviluppata nell'ambito di questo progetto di tesi sono stati utilizzati diversi codici GS1. In questa sezione andremo ad analizzare due che sono tra i più importanti: il Global Trade Item Number (GTIN) e il Global Location Number (GLN).

Il GTIN (Global Trade Item Number) è la chiave GS1 usata in tutto il mondo per identificare i prodotti (unità consumatore) e i colli (unità imballo), per la vendita nella grande distribuzione e online. [3]

Non esiste un'unica tipologia di GTIN in quanto ognuna è destinata ad identificare prodotti diversi ed ha una lunghezza variabile:

- il GTIN-14 (14 cifre) identifica unità d'imballo (colli) a peso variabile;
- il GTIN-13 (13 cifre) identifica prodotti a peso fisso, prodotti a peso variabile e unità d'imballo (colli) a peso fisso;
- Il GTIN-8 (8 cifre) identifica prodotti a peso fisso e di piccole dimensioni.

Ogni GTIN termina con una cifra detta cifra di controllo o check digit che serve a verificare la correttezza dei codici GS1 ed è calcolata in base alle cifre che la precedono. [4]

Nello standard EPCIS non sono utilizzati i GTIN, ma i SGTIN che sono codici ricavati partendo dal GTIN e completati con un identificatore seriale che corrisponde al numero assegnato dall'azienda ad ogni singolo pezzo, così da permettere l'identificazione di ogni singolo prodotto (unità consumatore) o colle (unità imballo) protagonista dell'evento EPCIS.

La cifra indicatore corrisponde a 0 nel caso di un GTIN-13, a 9 nel caso di un GTIN-14 se si tratta di un imballo a peso variabile, invece in tutti gli altri casi è una cifra compresa tra 1 e 8. Il prefisso aziendale di 9 cifre è assegnato all'azienda proprietaria del marchio quando aderisce allo standard GS1. L'azienda assegna il codice prodotto ad ogni prodotto che deve essere identificato, la cifra di controllo (check digit) è calcolata rispetto alle cifre che la precedono, infine c'è il seriale che caratterizza in maniera univoca ogni singolo pezzo ed anch'esso è stabilito dall'azienda.

Il GLN (Global Location Number) è la chiave GS1 utilizzata per identificare fabbriche, rivenditori, enti pubblici, centri di distribuzione, sedi operative e legali, reparti interni, banche e magazzini e i diversi luoghi ad essi collegati. [5]

Il GLN è composto da 13 cifre: le prime nove sono il prefisso aziendale, le tre cifre successive sono assegnate dall'azienda per indicare un luogo, l'ultima, invece, è una cifra di controllo ed è calcolata, utilizzando un algoritmo, in base alle cifre che la precedono.

L' SGLN è la chiave GS1 usata per identificare con esattezza un luogo all'interno dell'azienda che è già identificata con un GLN.

La chiave SGLN inizia con un prefisso “urn:epc:id:sgln:” a cui si aggiungono il prefisso aziendale e il codice del luogo separati da un punto (9 cifre il primo e 3 il secondo), completano la chiave una cifra di controllo stabilita da un algoritmo e un'estensione assegnata dall'azienda che può essere compresa tra 1 e 17 cifre. La funzione dell'estensione riguarda la possibilità di localizzare un punto fisico specifico all'interno del luogo precedentemente individuato attraverso il GLN.

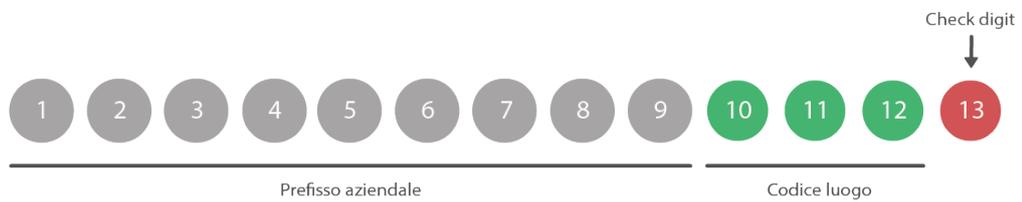


Figura 2.3: Formato GLN

Capitolo 3

Applicazioni Web

Le applicazioni web (web application) sono applicazioni client-server (Figura 3.1) alle quali si accede tramite un browser, con il supporto dei consueti protocolli di rete. Tali applicazioni non necessitano di installazione in quanto le loro funzionalità applicative sono caricate da un server web. A differenza delle applicazioni desktop sviluppate per una specifica piattaforma e quindi vincolate ad essa, un'applicazione web funziona su tutti i dispositivi e su tutti i sistemi operativi che dispongono di un browser. Ogni applicazione web presenta elementi interattivi che invitano e guidano l'utente al compimento di determinate azioni, i normali siti Internet, invece, sono spesso statici e sono caratterizzati da forme testuali di tipo informativo. Alla categoria delle applicazioni web fanno parte alcuni servizi di Google, ad esempio Gmail e Google Maps e anche servizi come Amazon Music e Amazon Video. Le applicazioni web offrono importanti vantaggi. I più importanti sono la facilità dell'aggiornamento, l'assenza di vincoli rispetto alle piattaforme e l'assenza di problemi riguardo alla compatibilità sulla versione poiché tutti gli utenti che accedono ad un'applicazione web utilizzano la stessa.

3.1 Storia delle Applicazioni Web

Le prime applicazioni client-server iniziarono ad essere utilizzate all'inizio degli anni 80 del XX secolo. L'utente doveva installare sul proprio computer un programma precompilato che aveva la funzione di interfaccia utente e il carico dell'applicazione era condiviso tra il codice sul server e il codice installato localmente su ciascun client. Ogni aggiornamento dell'applicazione lato server però, spesso richiedeva un aggiornamento del codice lato client, il quale trovandosi sul pc degli utenti obbligava all'installazione della nuova versione su tutte le macchine aumentando di parecchio i costi di supporto e riducendo nello stesso tempo la produttività. Un ulteriore impatto sui costi era dovuto al porting, che è un processo di traduzione, talvolta

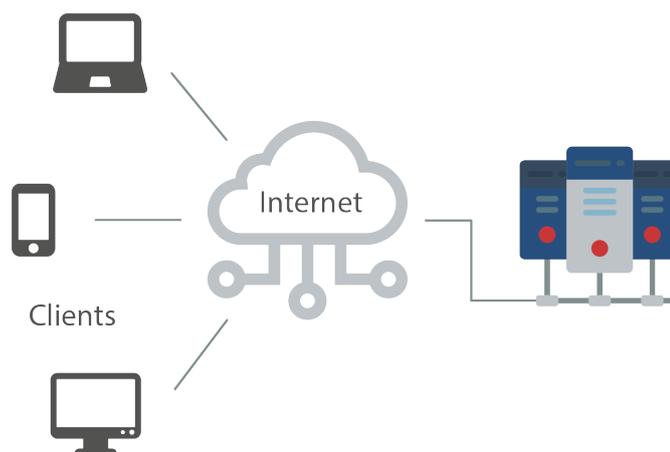


Figura 3.1: Esempio sistema client-server

con modifiche, di un componente software, realizzato per consentirne l'utilizzo in un ambiente diverso da quello originale. Il porting era necessario in quanto sia la parte server che la parte client delle applicazioni erano legate ad una particolare architettura e ad un determinato sistema operativo.

Le applicazioni web invece utilizzano documenti web scritti in formati standard (HTML e Javascript) in modo tale da garantire la capacità di interpretazione alla maggior parte dei browser degli utenti. Le applicazioni web possono essere definite una variante del software che veniva installato sul pc di ogni utente. Il software viene scaricato quando l'utente visita la pagina web dell'applicazione utilizzando protocolli standard come HTTP. Non sono più necessari aggiornamenti esterni poiché l'aggiornamento si esegue ogni volta che si accede alla pagina web, inoltre il browser è considerato come client universale per qualsiasi applicazione web.

Nella storia delle applicazioni web il 1995 risulta un anno fondamentale poiché proprio in quell'anno Netscape presentò Javascript (sezione 3.3.2) il quale consentiva ai programmatori di inserire elementi dinamici all'interno delle pagine web che giravano sui browser degli utenti. L'anno successivo Macromedia presentò Flash, un riproduttore di animazioni che ogni utente poteva aggiungere al proprio browser come plugin. Queste due nuove applicazioni permettevano agli utenti di ottenere pagine dinamiche senza l'obbligo di una comunicazione client-server. Altra tappa importante fu l'introduzione nel 2005 di Ajax (Asynchronous Javascript and XML) che consentiva l'aggiornamento dinamico di una pagina web senza l'intervento dell'utente. Ajax offriva alle applicazioni web la possibilità di ricevere e inviare dati dal server in modo asincrono in background. Nel 2011 venne pubblicato HTML5 il quale evitava l'obbligo di installare plugin come Flash per ottenere animazioni all'interno delle interfacce e rendeva disponibili agli utenti nuovi elementi per strutturare le pagine, ad esempio Header e Footer, ma non solo.

3.2 SPA vs MPA

Single-page application (SPA) e multi-page application (MPA) sono le due strade per lo sviluppo di una web application. Entrambe presentano aspetti positivi e aspetti negativi.

Le single-page application offrono un approccio più moderno. Sono applicazioni che funzionano all'interno di un browser e durante il loro utilizzo non richiedono il ricaricamento della pagina. Molti studi indicano come il livello e il tempo di attenzione delle persone si siano ridotti, quindi la velocità diventa un fattore determinante. Le SPA si caricano più velocemente rispetto alle MPA in quanto caricano la maggior parte dei dati una sola volta e non si ricaricano completamente ogni volta che l'utente interagisce. Le single-page application, basandosi sull'utilizzo di JavaScript ne richiedono l'attivazione sul dispositivo e ciò le rende più sensibili agli attacchi hacker. Le SPA inoltre presentano maggiori problemi nell'ottimizzazione SEO (Search Engine Optimization). Sono state utilizzate anche da Google e da Facebook.

Le multi-page application (MPA) sono considerate un approccio più classico nello sviluppo di una web application.

3.3 React

React è una libreria JavaScript open source utilizzata per realizzare interfacce utente (Figura 3.2). Permette di sviluppare applicazioni web utilizzando considerevoli quantità di dati in maniera semplice, veloce e graduale. Facebook l'ha sviluppata, ne mantiene la gestione del progetto e l'ha sfruttata per realizzare ad esempio la versione web di Instagram. React è anche utilizzata da aziende importanti come Netflix, Uber e AirBnb ed ha una community molto ampia e in costante crescita.



Figura 3.2: Logo React

3.3.1 La scelta di React

La scelta di React per la realizzazione del front end di XTap è stata determinata dal fatto che è open source, infatti una considerevole community di sviluppatori collabora con gli sviluppatori di Facebook. In tal modo Facebook gestisce la libreria principale ed affida alla community la gestione ed il mantenimento delle librerie correlate.

Altre caratteristiche hanno influito sulla scelta: la semplicità di utilizzo, le performance, il testing e lo sviluppo modulare.

L'utilizzo di Javascript e l'approccio basato su componenti rendono React semplice da imparare per chi possiede le conoscenze di base di Javascript, CSS e HTML.

React permette di realizzare piattaforme molto sofisticate garantendo le migliori prestazioni.

Il test delle applicazioni React è semplice in funzione della regola di responsabilità unica incorporata nei suoi componenti e al flusso unidirezionale dei dati.

La realizzazione di un'interfaccia attraverso l'utilizzo di componenti permette di rispondere con prontezza ai continui cambiamenti delle specifiche richiesti dal cliente e lo sviluppo modulare consente di modificare un componente senza incidere sul funzionamento degli altri.

Il grafico (Figura 3.3, dal sito *2019stateofjs.com*) evidenzia le opinioni dichiarate dagli sviluppatori riguardo ai diversi frameworks. I segmenti più chiari rappresentano le opinioni negative, mentre quelli più scuri le opinioni positive.



Figura 3.3: Grafico opinioni degli sviluppatori

3.3.2 JavaScript

JavaScript è un linguaggio di programmazione orientato agli oggetti. Venne realizzato nel 1996 da Brendan Eich durante la sua esperienza come ingegnere in Netscape. Inizialmente fu denominato Mochan, poi Livescript e successivamente JavaScript per una scelta di marketing che tentava di sfruttare la popolarità del linguaggio Java. Venne standardizzato per la prima volta nel 1997 da una società europea di standardizzazione: la ECMA e ciò portò alla nascita della prima edizione degli ECMAScript. La terza edizione, avvenuta nel 1999 promosse un significativo aggiornamento e da quel momento lo standard è rimasto relativamente uguale.

L'ingannevole semplicità di JavaScript talvolta nasconde importanti e potenti caratteristiche. La sintassi di JavaScript deriva dal linguaggio C e Java, infatti se ne ritrovano parecchie strutture. Molte applicazioni utilizzano questo linguaggio e ciò conferma come una sua conoscenza approfondita sia una fondamentale abilità per uno sviluppatore web. JavaScript non ha concetti di input e output. È stato progettato per essere eseguito come linguaggio di scripting in un ambiente ospite, diventa perciò responsabilità del suddetto ambiente l'offrire meccanismi utili per la comunicazione con l'esterno. Il più comune ambiente ospite è il browser, ma ci sono interpreti di questo linguaggio anche in altri ambienti come ad esempio Photoshop e immagini SVG.

3.4 React concetti chiave

React è una libreria JavaScript basata su componenti che, combinati tra loro, offrono la possibilità di costruire interfacce grafiche interattive anche complesse. Sfrutta i concetti di ES6 (EcmaScript6) che talvolta i browser non riescono ad interpretare. Si rivolgono allora agli strumenti Babel e Webpack che vengono sfruttati in maniera trasparente per lo sviluppatore.

I principi generali sui quali si basa React sono la composizione e la dichiaratività. L'interfaccia è composta da un insieme di blocchi indipendenti detti anche componenti che possono essere suddivisi e riutilizzati. Nell'esempio in Figura 3.4 possiamo identificare cinque diversi componenti dove il più esterno evidenziato in arancione è la tabella. All'interno di questo componente troviamo evidenziata in blu la barra di ricerca dove vengono ricevuti gli input dell'utente, in verde la tabella dei prodotti filtrata in base agli input, in azzurro un titolo che identifica la categoria e in rosso le righe corrispondenti ad ogni prodotto. I componenti che si trovano all'interno di un altro componente saranno suoi figli nella gerarchia.

Lo sviluppatore deve fornire una descrizione dell'interfaccia in termini di componenti, loro proprietà e struttura. Non è necessario indicare a React come agire, ma bisogna indicare alla libreria l'obiettivo finale, quindi non come fare, ma cosa

fare. React trasformerà tale descrizione in un programma eseguibile che andrà a realizzare l'interfaccia e le sue interazioni.

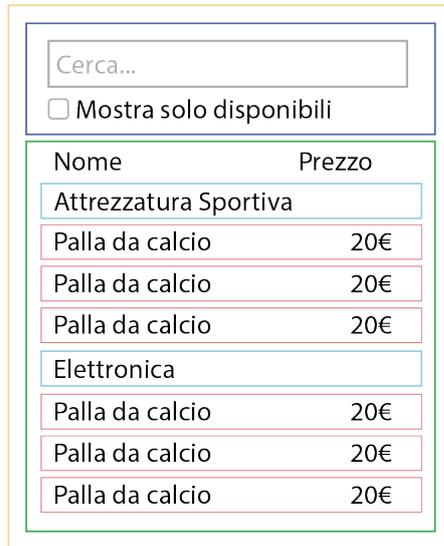


Figura 3.4: Esempio di struttura a componenti

```
const input = ['a', 'b', 'c']
let result = input.map(str => str + str)
console.log(result) // [ 'aa', 'bb', 'cc' ]
```

Sopra si può vedere un esempio di codice dichiarativo dove dichiariamo di voler mappare ogni elemento dell'array e operare sui suoi elementi, come si può notare non dobbiamo dire al computer esattamente le operazioni da fare passo dopo passo.

Hello React

Di seguito è riportato un primo esempio di componente React che renderizza la scritta "Hello React!".

```
const Hello = (props) => {
  return (
    <div className = "container">>
      <h1>{props.title}</h1>
    </div>
  );
}
```

```
const App = () => {
  return (
    <div>
      <Hello title="Hello React!"/>
    </div>
  );
}
```

3.4.1 JSX - JavaScript Syntax Extension

JSX è un'estensione della sintassi JavaScript. A prima vista pare un linguaggio di template, ma ha la prerogativa di sfruttare tutta la potenzialità di JavaScript. Ogni elemento JSX ha un nome, può avere zero o più props (attributo = valore) e può racchiudere zero o più figli. I figli sono elementi semplici o a loro volta componenti diversi. Il transpiler è un elemento della catena che ha il compito di trasformare il codice JSX in chiamate a funzioni le quali permettono di assemblare la struttura richiesta. L'output di un componente è sempre un albero JSX. Analizzando l'esempio del componente "Hello" visto nel paragrafo precedente possiamo evidenziare il lavoro del transpiler Babel dove invece del JSX ci sono le chiamate a funzioni della libreria React:

```
const Hello = (props) => {
  return (
    React.createElement("div", {className: "container"},
      React.createElement("h1", null, "Hello, React!")
    );
}
```

3.4.2 Componenti

I componenti permettono di pensare e dividere l'interfaccia utente in diverse parti indipendenti tra loro e pensare a ciascuna di esse in maniera isolata, inoltre i componenti possono talvolta essere riutilizzabili.

I componenti React possono essere scritti come funzioni oppure come classi, accettano in input dei dati arbitrari e ritornano elementi React contenenti la descrizione di quello che si dovrebbe vedere sullo schermo.

Ogni componente possiede un nome univoco, accetta in ingresso un insieme di informazioni immutabili "props" (da "properties" in inglese, proprietà) e può avere un suo stato interno che può cambiare nel tempo. Le proprietà che prendono il nome di props sono un oggetto Javascript che nel componente Hello dell'esempio

risulta essere `{title: "Hello React!"}`.

I componenti funzione sono funzioni JavaScript che hanno come argomento delle props e restituiscono un'interfaccia utente solitamente tramite JSX.

I componenti classe sono classi di Javascript che prevedono un metodo di render il quale ritorna un'interfaccia utente solitamente tramite JSX.

Tutti i componenti React devono comportarsi come funzioni pure, ovvero non possono provare a cambiare i dati in input. È possibile anche utilizzare la destrutturazione e in tal caso il componente Hello può essere riscritto:

```
const Hello = ({ title }) => {
  return (
    <div className = "container">
      <h1>{title}</h1>
    </div>
  );
}
```

Un componente React, nel suo output ha la possibilità di chiamare altri componenti, ciò permette ad ogni livello di ottenere la massima astrazione. In un'applicazione React il componente più alto della gerarchia è solitamente chiamato App. Tale componente racchiude in esso tutti gli altri componenti. I componenti React statici, la cui realizzazione è stata illustrata finora, ricevono in ingresso dei dati props e il loro output rimane invariato al momento della renderizzazione.

Le interfacce utente, però, sono dinamiche e cambiano nel tempo, diventa essenziale quindi la possibilità di costruire dei componenti in grado di modificare il proprio output in relazione al comportamento dell'utente o ad altre azioni. A tal proposito si parlerà nel paragrafo 3.5.3. del concetto di "stato".

3.4.3 Il ciclo di vita di un componente

Ogni componente dell'interfaccia è creato a seguito della richiesta iniziale alla funzione `ReactDOM.render(<Component>, root_node)`. Spesso al posto di `<Component>` troviamo `<App/>` e al posto di `root_node` troviamo `document.getElementById('root')`. All'interno del componente `<App/>` saranno presenti altri componenti che hanno la funzione di comporre l'interfaccia.

React crea questi componenti e controlla il loro ciclo di vita composto dalle seguenti tre fasi: Mounting, Updating e Unmounting.

Nella prima fase detta Mounting, è eseguita la funzione del componente con la restituzione di una struttura poi inserita nell'albero DOM. Nella seconda fase, detta Updating, se le props o lo stato di un componente variano, ad esempio attraverso l'interazione con l'utente o al risultato dalla rete, la funzione è rieseguita. Il risultato

di tale funzione sarà confrontato con la struttura già presente nell'albero DOM e verranno apportate le modifiche necessarie per ottenere il nuovo risultato. Nella terza fase, detta Unmounting, la struttura del componente è rimossa dall'albero DOM.

In Figura 3.5 è riportato il lifecycle di un componente React.

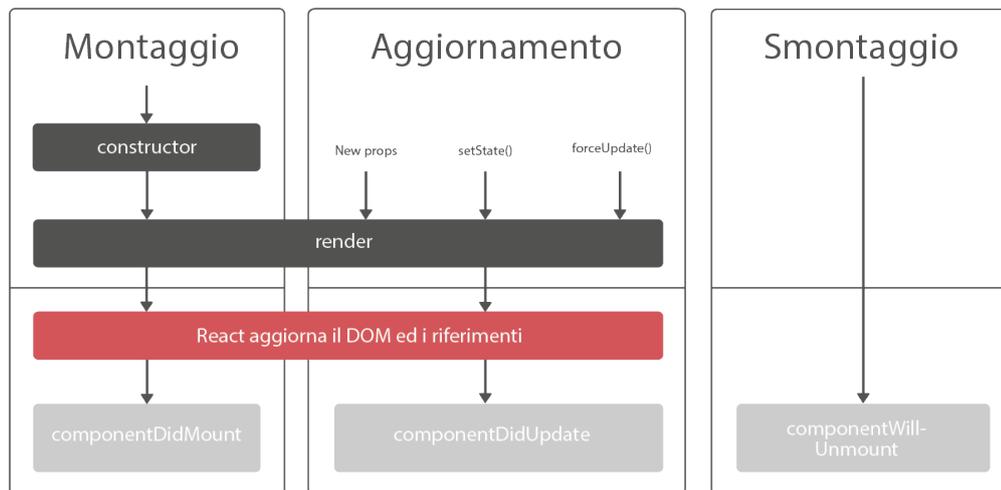


Figura 3.5: Lifecycle di un componente React

3.5 React Hooks

Gli Hooks, introdotti in React dalla versione 16.8.0, rendono possibile la soluzione di alcune difficoltà:

- Le classi creano confusione. L'uso della classi ma in particolare la conoscenza del `this` in JavaScript sono difficili da leggere e da comprendere sia per gli esseri umani che per le macchine. Gli Hooks offrono la possibilità di utilizzare più componenti funzione senza dover utilizzare le classi.
- Una logica che dipende dallo stato era difficoltosa da riutilizzare in componenti diversi. Prima degli Hooks per incapsulare la logica di gestione dello stato si usavano componenti di ordine superiore ed era necessario renderizzare le props. La struttura creata era difficile da leggere e da scrivere, e facilmente soggetta ad errori in caso di modifiche (wrapper hell).
- Si dimostrava arduo comprendere componenti complessi in quanto al loro interno a volte ci si trovava a dover attuare metodi del lifecycle che contenevano logica non correlata facilitando l'introduzione di bug. Gli Hooks invece di

utilizzare una divisione basata sui metodi del lifecycle offrono la possibilità di suddividere un componente in funzioni più piccole basate su pezzi tra loro correlati, ad esempio la richiesta di dati.

React ha l'obiettivo di permettere al programmatore di mantenere il codice il più dichiarativo possibile, anche se talvolta all'interno del programma servono pezzi con effetti collaterali. Per rendere gestibile il programma è necessario confinare questi pezzi in punti precisi e ben identificabili separandoli da tutto ciò che è puramente funzionale. Le funzioni Hooks, messe a disposizione della libreria React, al loro interno eseguono tutte le operazioni non funzionali come possono essere la gestione e la memorizzazione dello stato o l'esecuzione di effetti collaterali. La libreria identificando e tracciando questi pezzi è in grado di aggiornare un componente ogni volta che si verifica un cambiamento.

3.5.1 React Hooks lifecycle

I componenti hanno un ciclo di vita e per comprendere gli Hooks è necessario sapere che essi sono eseguiti in determinate fasi di questo ciclo. Nella Figura 3.6 si nota come l'hook `useMemo` è eseguito solo nella fase di mounting del componente, mentre l'hook `useEffect` è eseguito in tutte e tre le fasi del componente. L'hook `useEffect` inoltre ritorna una funzione di possibile esecuzione durante la fase unmounting del componente.

3.5.2 Regole degli Hooks

I componenti possono essere espressi come componenti funzione o componenti classe. React per ogni componente funzione mantiene al proprio interno un oggetto con corrispondenza 1:1 in cui è descritto tutto ciò che succede a quel componente. Se all'interno del componente funzione vengono usati gli hooks React essi sono aggiunti in una lista mantenuta all'interno dell'oggetto.

Gli hooks sono molto potenti e anche flessibili, ma è fondamentale rispettare alcune regole:

- Possono essere utilizzati solo in componenti funzione e non in componenti classe;
- Vanno invocati all'inizio dei componenti funzione ed è importante l'ordine in cui si invocano;
- Non si devono chiamare all'interno di un codice condizionale (`if/switch`) o all'interno di `loop`;
- Devono essere invocati lo stesso numero di hooks ogni volta che una funzione componente è eseguita.

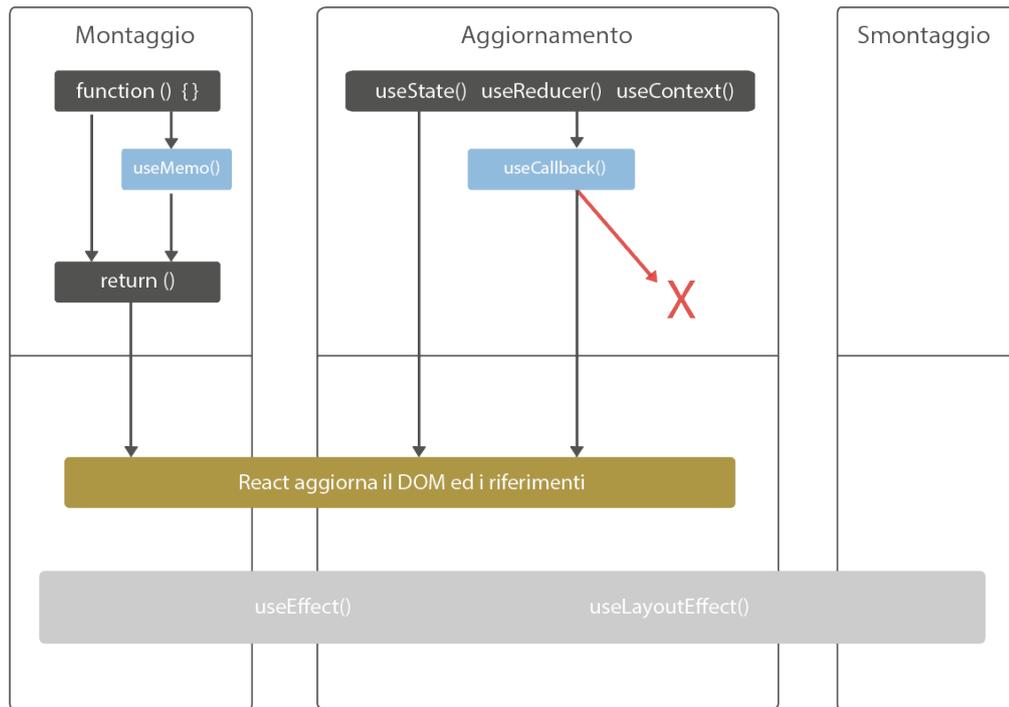


Figura 3.6: React Hooks lifecycle

3.5.3 useState e useReducer

Un'interfaccia utente è composta da diversi componenti, l'utente interagisce con essi e richiede le loro reazioni e la memorizzazione delle azioni precedenti. I componenti puramente funzionali, per definizione non hanno un proprio stato anche se ogni componente ha la necessità di un proprio stato che rappresenti l'effetto delle azioni che il sistema o l'utente hanno effettuato su di lui. La libreria React mette a disposizione l'hook `useState`, il più comune degli hook, il quale è in grado di svolgere questo compito. La funzione `useState` accetta come argomento un solo parametro, lo stato iniziale (un oggetto, una stringa, ...) e restituisce una coppia di valori. Lo stato corrente e una funzione che ha il compito di aggiornare lo stato. Un componente al proprio interno può mantenere più stati e ciò è possibile invocando più volte la funzione `useState()`.

Un esempio di semplice utilizzo dell'hook `useState` è la realizzazione di un componente Contatore:

```
const Contatore = (props) => {
  const [contatore, setContatore] = useState(0);
  return (
```

```

    <div >
      <p> Hai cliccato {contatore} volte </div>
      <button onClick = { () => setCount(contatore+1) }>
        +
      </button>
    </div>
  );
}

```

L'hook `useState` permette di mantenere al proprio interno più di uno stato, ma quando la complessità e il numero degli stati supera un certo limite il solo utilizzo di questo hook diventa problematico ed è preferibile affidarsi a `useReducer`.

`UseReducer` offre per la gestione dello stato un approccio più strutturato e solido, quando è presente una logica complessa e all'interno del componente ci sono più stati non completamente indipendenti l'uno dall'altro.

È fondamentale comprendere da subito il significato dei termini azione e riduzione. Se l'utente compie un'operazione (azione) lo stato cambia. Nell'esempio del contatore clicca sul tasto "+". Questa azione è un oggetto JavaScript contenente la chiave "type" che ha come valore una stringa la quale definisce la mia azione, oltre al campo "type" si possono passare argomenti dell'azione.

La riduzione è una funzione che prende in ingresso due parametri: lo stato precedente e un'azione, e restituisce il nuovo stato in base all'azione ricevuta. La riduzione inoltre lascia immutato nello stato tutto ciò che non deve essere modificato inserendo i nuovi valori e aggiornandolo. La struttura di un riduttore è basata su un costrutto di tipo `switch(action.type)` in cui all'interno vengono trattate le azioni che interessano e anche la clausola di default riguardante le azioni non conosciute. Considerando l'esempio di un contatore con due pulsanti "+" e "-", avremo due azioni: una di incremento ("incrementa") e una di decremento ("decrementa"). Definite le azioni il riduttore costruito avrà una struttura di questo tipo:

```

const initialState = {contatore: 0}

const reducer = (state, action) => {
  switch (action.type) {
    case "incrementa":
      return {count: state.contatore + 1};
    case "decrementa":
      return {count: state.contatore - 1};
    default: return state;
  }
}

```

La libreria React, per l'uso di un riduttore, mette a disposizione l'hook `useReducer`: una funzione che richiede due parametri cioè la funzione di riduzione e lo stato iniziale. Così come già evidenziato per `useState`, anche `UseReducer` restituisce due parametri: il primo è lo stato corrente, il secondo è la funzione `dispatch`. Quest'ultima funzione prende un'azione in ingresso e la passa internamente al reducer che provvede a calcolare il nuovo stato provocando l'aggiornamento degli opportuni componenti

Nell'esempio sottostante il componente `Contatore` è riscritto utilizzando il reducer visto sopra.

```
const [state, dispatch] = useReducer(reducer, initialState);

return (
  <div >
    <p>Contatore:{ state.contatore}</p>
    <button onClick = { () => dispatch({type:"decrementa"}) }>-
    </button>
    <button onClick = { () => dispatch({type:"incrementa"}) }>+
    </button>
  </div>
);
```

Capitolo 4

User Experience Design

La piattaforma XTap realizzata nell'ambito di questo progetto di tesi ha come obiettivo primario l'esposizione di molti dati proposti attraverso una lettura semplice e intuitiva. La rappresentazione di molti dati all'interno di un'interfaccia presenta alcuni problemi e una progettazione poco accurata e non facilmente fruibile può creare difficoltà al cliente, cioè all'utilizzatore finale il quale ne potrebbe decidere l'abbandono e il relativo insuccesso. La valutazione più o meno positiva di un prodotto è molto condizionata dall'esperienza che l'utente si costruisce durante l'uso, quindi rendere il prodotto funzionale e funzionante è essenziale.

L'utente non ha gli obiettivi, lo stesso background e i modelli mentali dello sviluppatore e di questi aspetti è fondamentale tenerne conto nell'ambito della progettazione. È importante perciò riuscire a creare un sistema adatto all'utilizzatore finale. Un metodo efficace coinvolge l'utente nelle varie fasi rendendolo protagonista dell'intero processo di sviluppo. La metodologia di progettazione User Centered Design assolve questa funzione considerando le esigenze, i limiti e i desideri dell'utente in ogni momento del percorso progettuale.

4.1 Usabilità

L'usabilità è definita dall' ISO 9241-210:2010 come *l'efficacia, l'efficienza e la soddisfazione con le quali determinati utenti raggiungono determinati obiettivi in determinati contesti*.

Il Nielsen Norman Group, gruppo leader nell'ambito della User Experience, stabilisce un ordine di importanza e sostiene che l'usabilità è seconda solamente all'utilità, e viene prima della desiderabilità e della brand experience, di conseguenza quando un prodotto risolve uno o più problemi all'utente, se ne deve valutare l'usabilità. La definizione data dall'ISO descrive l'usabilità secondo tre variabili tra loro indipendenti: l'efficacia, l'efficienza e la soddisfazione.

L'efficacia è determinata dall'accuratezza e dalla completezza attraverso le quali l'utente raggiunge gli obiettivi prefissi. La sua valutazione emerge dalla percentuale dei compiti completati e/o dal rapporto tra successi e insuccessi.

L'efficienza riguarda la quantità di risorse spese in relazione all'accuratezza e alla completezza con cui l'utente raggiunge i suoi obiettivi. La sua valutazione è legata al tempo impiegato per il completamento di un task o alla percentuale e/o al numero di errori.

La soddisfazione è il comfort, la piacevolezza con cui si utilizza un determinato prodotto, valutata, ad esempio, attraverso le recensioni.

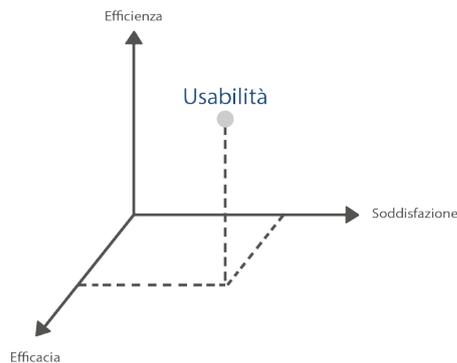


Figura 4.1: Le tre dimensioni dell'usabilità

Il timer sui telefoni Apple è una dimostrazione chiara di quanto le tre variabili descritte sono indipendenti, ma anche come ognuna può influenzare notevolmente l'usabilità del prodotto.

La funzione Timer dell'applicazione orologio degli Iphone permette di impostare un nuovo timer scorrendo sui valori di ore, minuti e secondi.

L'operazione è certamente gratificante in quanto i valori scorrono accelerando fermandosi poi lentamente sul valore scelto, anche se richiede più tempo dell'impostazione classica con l'uso della tastiera.

La scelta progettuale, in questo caso, ha favorito la soddisfazione rispetto all'efficacia e all'efficienza.

Alle tre variabili inerenti l'usabilità se ne aggiungono altre altrettanto importanti: la sicurezza e la gestione dell'errore, l'utilità, l'apprendibilità e la memorabilità.

La sicurezza e la gestione dell'errore hanno lo scopo di proteggere l'utente da situazioni sgradevoli e/o di pericolo, permettendogli la gestione autonoma di un problema. L'utilità è legata alla funzionalità rispetto al bisogno. L'apprendibilità consiste nella semplicità e nel tempo utile all'utente per imparare le nozioni di base indispensabili per l'utilizzo di un determinato sistema. A questa è strettamente legata la memorabilità, infatti ciò che è stato appreso deve poter essere ricordato

facilmente anche dopo un periodo relativamente lungo di non utilizzo del sistema. L'usabilità non è da considerarsi come una proprietà assoluta di un prodotto, ma deve essere valutata contestualmente rispetto al compito da svolgere, all'utente e alla modalità d'uso. L'usabilità inoltre non può essere valutata attraverso un'osservazione isolata.

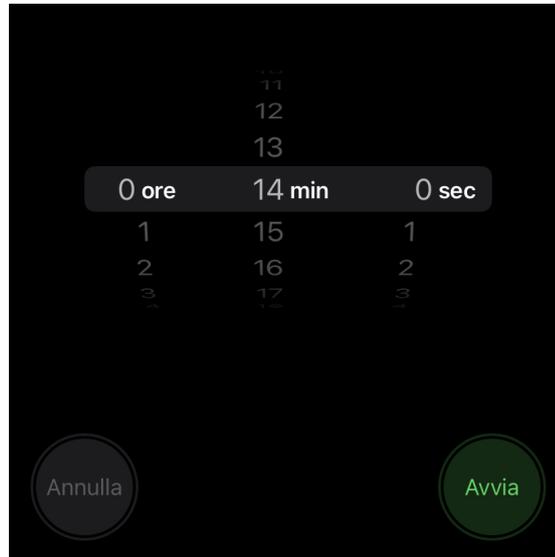


Figura 4.2: Timer sui telefoni Apple

4.2 UCD - User Centered Design

Lo psicologo e ingegnere statunitense Donald A. Norman nel libro *User-Centered System Design: New Perspectives on Human-Computer Interaction* pubblicato nel 1986, utilizza per la prima volta l'espressione "User Centered Design" e ne descrive la metodologia.

L'UCD, appunto l'User Centered Design, pone i desideri, i limiti e le esigenze dell'utente finale al centro dell'intero processo di progettazione in modo da offrirgli un prodotto adeguato alle sue aspettative.

I desideri (Wants) sono i beni, i servizi e le funzionalità la cui presenza migliora la qualità della vita di una persona;

I limiti hanno diverse tipologie: un limite fisico come l'altezza è una caratteristica fisica; la funzione del sonno è un limite fisiologico; la memoria come processo mentale è un limite cognitivo; l'emozione della rabbia è un limite affettivo.

Le esigenze (Needs) rappresentano beni, servizi e funzionalità la cui assenza impedisce di raggiungere un determinato obiettivo o mette a rischio l'esistenza in vita.

Questi punti chiave su cui si basa l'UCD vanno considerati non solo nella fase iniziale della progettazione, ma durante tutte le fasi. A differenza dell'approccio system-centered in cui l'efficienza del sistema è anteposta all'utente nell'approccio UCD il prodotto è sviluppato seguendo i bisogni dell'utente evitando forzature sul suo comportamento.

Il modello ISO 13407 illustrato nella Figura 4.3 mostra il processo User Centered Design per i sistemi interattivi. Il processo di sviluppo di tipo interattivo avviene solo dopo una prima fase di pianificazione (1) nella quale è importante identificare le necessità dell'utente attraverso le risposte a domande specifiche, ad esempio: Per chi si progetta? Perché si progetta? Quali sono gli obiettivi?.

Il processo si svolge attraverso altre quattro fasi:

Analisi del contesto (2)

Si analizza prima il contesto in cui l'utente utilizzerà il prodotto, poi la tipologia dei futuri utenti e lo scopo per cui utilizzeranno il prodotto. L'indagine conoscitiva può essere svolta attraverso osservazioni sul campo oppure con sondaggi a potenziali utenti, svolti mediante questionari, interviste, focus group, . . .

Definizione dei requisiti (3)

Si definiscono le caratteristiche specifiche del prodotto: la struttura concettuale, le funzionalità del sistema e i requisiti di usabilità, tenendo in considerazione le peculiarità dell'utente e dell'azienda. In questa fase le informazioni possono essere desunte dallo studio di un caso d'utilizzo o attraverso mappe cognitive o personas.

Sviluppo (4)

Nella fase dello sviluppo si inizia il processo di progettazione vero e proprio, con le proposte di progetto e la costruzione di simulazioni inizialmente in formato cartaceo, poi con wireframe digitali e prototipi.

Valutazione (5)

L'ultima fase parte da un piano di valutazione e dalla raccolta dei feedback dei potenziali utenti per verificare il raggiungimento degli obiettivi prefissati riguardanti l'efficacia e l'efficienza. La raccolta dei feedback permette inoltre di rientrare nel processo e di ottimizzare il prodotto. Gli strumenti utilizzabili nella fase di valutazione sono test economici come i test a bassa fedeltà oppure un test molto più costoso come può essere una simulazione.

Fino a quando gli utenti non offrono un feedback soddisfacente continua il processo iterativo, considerando nel contempo le esigenze aziendali in termini di tempi e di costi.

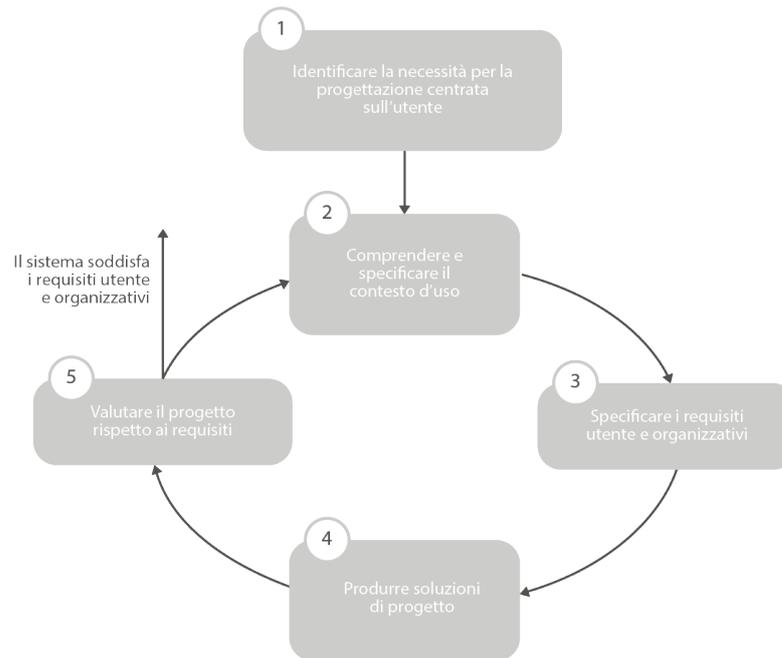


Figura 4.3: Processo User Centered Design per i sistemi interattivi

4.3 User Experience

Donald Norman, nel 1995, definisce la UX-User Experience come *“L’esperienza che la persona vive quando interagisce con un prodotto nel mondo reale”*. Norman intendeva colmare tutti gli aspetti dell’esperienza provata da una persona nell’interazione con un sistema, considerando riduttivi i due concetti di “human interface” e “usabilità”.

La User Experience non rappresenta l’aspetto grafico di un prodotto e neppure l’insieme delle sue funzionalità, ma diventa uno strumento talmente importante da fare la differenza tra il suo successo o il suo completo fallimento. La UX del prodotto è influenzata dalle modalità con cui la persona interagisce, dalle sensazioni che provoca nell’ utilizzatore e da quanto il prodotto ne soddisfa il bisogno. La UX, secondo Norman e Nielsen, comprende tutti gli aspetti dell’interazione come ad esempio le reazioni e le emozioni che la persona sperimenta durante l’uso, le quali influiscono sulla determinazione dell’esperienza che sarà piacevole, utile, ecc. Tutti i prodotti, siano essi oggetti o servizi hanno una propria UX e ciò stimola l’intenzione di progettare per la UX, ovvero creare un prodotto che offra caratteristiche tali da rendere il suo utilizzo piacevole per un maggior numero di utenti possibile. La progettazione di un sistema per la UX presuppone la centralità dell’utente, integrata da quegli aspetti di sostenibilità e di funzionalità che lo completano.

Peter Morville, nel diagramma UX Honeycomb (Figura 4.4), indica con le seguenti definizioni i sette aspetti della UX che considera essenziali: utile, desiderabile, accessibile, credibile, trovabile, usabile e valore.

Utile poichè deve rispondere ad un bisogno reale dell'utente; desiderabile in quanto deve fornire qualcosa che aggiunge valore; accessibile a tutte le persone, anche alle persone diversamente abili; credibile perchè deve creare un senso di fiducia; trovabile in quanto rende semplice reperire le informazioni; usabile siccome tutti gli utenti hanno la possibilità di raggiungere il loro obiettivo in maniera efficace, efficiente e con soddisfazione; valore ciò che fornisce il prodotto al cliente e all'azienda.

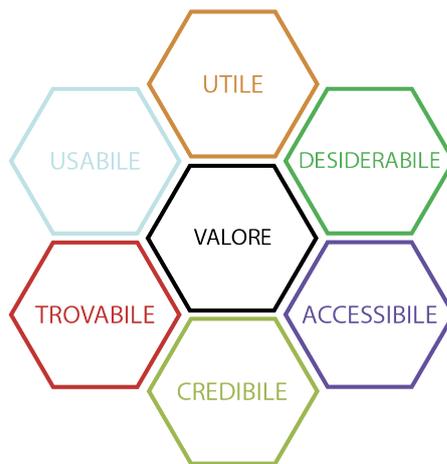


Figura 4.4: Diagramma UX Honeycomb di Peter Morville

4.3.1 Il modello di Garrett

Un progetto per la User Experience richiede che ogni aspetto sia il risultato di una specifica volontà del progettista.

Uno dei più noti User Experience Designer, Jesse.J. Garrett, ha analizzato il processo di progettazione della UX di un prodotto software e ha individuato cinque passaggi corrispondenti a cinque piani di lettura: Strategia, Obiettivo, Scheletro, Struttura e Superficie.

Garrett ha diviso verticalmente i cinque piani in modo che ogni piano fosse composto da due aree. In seguito a questa divisione verticale i singoli piani sono ulteriormente suddivisi in elementi del piano. Nella prima area di ogni piano, il prodotto è considerato dal punto di vista della funzionalità, come strumento per lo svolgimento di un compito; nella seconda area il prodotto diventa uno strumento informativo a due livelli: le informazioni che esso mette a disposizione e cosa significano le informazioni per l'utente.

Partendo dal basso verso l'alto i problemi che ognuno dei piani deve affrontare

sono sempre più concreti e ogni piano non risulta isolato, ma ha un'importante dipendenza dai piani sottostanti. Questa dipendenza fa sì che le scelte disponibili all'interno di un piano siano limitate dalle decisioni effettuate nei piani precedenti, quindi una scelta decisa al livello più basso, ovvero al piano Strategia, ha effetti lungo tutta la struttura. Garrett evidenzia quanto sia importante iniziare le attività su un piano prima del completamento delle attività sul piano sottostante, evitando di sovrapporre più di due piani.

La Figura 4.5 rappresenta la struttura descritta da Garrett.

Il piano della Strategia occupa il livello più basso, il più astratto. Ogni scelta compiuta a questo livello, come indicato in precedenza, influisce su tutta la struttura. All'interno di questo piano avviene la definizione dell'utente del sistema, degli obiettivi che si pone e dei suoi bisogni (User Needs), e anche quali sono gli obiettivi di business (Product Objectives). Il piano dell'Obiettivo, che segue in verticale il piano Strategia, descrive in modo dettagliato le funzionalità (Functional Specification) e quali sono i contenuti necessari (Content Requirements) che il sistema deve avere per soddisfare i bisogni dell'utente.

Il piano immediatamente successivo è quello della Struttura, utile per definire l'interazione tra utente e sistema (Interaction Design) e l'organizzazione dei contenuti in modo che l'utente ne possa avere un rapido ed efficace accesso (Information Architecture).

Nel penultimo piano, il piano dello Scheletro sono indicati il design dell'interfaccia con la quale l'utente dovrà interagire (Interface Design), gli elementi che gli offrono la possibilità di navigare attraverso i contenuti (Navigation Design) e le modalità attraverso cui le informazioni gli sono presentate in modo facilmente comprensibile (Information Design).

Il piano della Superficie, l'ultimo della struttura, è il più concreto, visibile all'utente con il quale interagisce. In questo piano è privilegiato l'aspetto grafico: icone, bottoni ecc, che insieme promuovono un'esperienza sensoriale positiva nel prodotto finale.

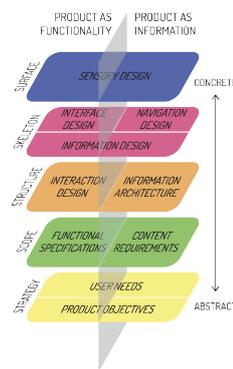


Figura 4.5: Struttura del modello di Garrett

4.4 Material Design

Durante lo sviluppo di un'applicazione è fondamentale mantenere un'identità grafica in tutte le sue parti, risulta quindi indispensabile affidarsi a delle linee guida. Il Material Design venne introdotto nel 2014 da Google per la progettazione di interfacce utente ed è un insieme di linee guida del linguaggio visuale, il quale integra i principi classici del buon design con le nuove tecnologie. L'obiettivo del Material Design è il perfezionamento dell'esperienza che l'utente vive durante l'utilizzo di un prodotto digitale, riducendo la distanza tra il mondo digitale e il mondo reale. Il termine "Material" (Materiale) indica l'ispirazione alla sostanza fisica del mondo e tale proposito conduce alla creazione di elementi dell'interfaccia utente sempre più simili agli oggetti veri con dimensioni, forme, prospettive, luminosità e capacità di adeguamento a situazioni diverse in tempo reale. Le animazioni naturali e reattive attirano e trattengono l'attenzione, creano continuità, evidenziano elementi o azioni e agiscono come feedback per l'utente durante le interazioni.

Material Design dedica anche una particolare attenzione alla scelta del colore che è molto importante per lo sviluppatore e per l'utente. I colori giusti accostati in modo corretto migliorano l'aspetto dell'applicazione e la sua accessibilità. Google, per la composizione di una palette, mette a disposizione Material Color Tool, uno strumento che, accostando un colore primario, e come primario s'intende il colore più utilizzato all'interno dell'applicazione, e un colore secondario, ne verifica l'accostamento sfruttando i diversi tipi di interfacce. Genera inoltre automaticamente le varianti chiare (Light) e scure (Dark) dei due colori e in un apposito pannello ne analizza l'accessibilità.

Le icone, parte integrante del linguaggio del web, ed aspetto significativo del Material Design, sostituiscono il testo scritto e rappresentano un linguaggio universalmente comprensibile, se utilizzate correttamente migliorano notevolmente l'usabilità e il design di un prodotto.

Il Material Design distingue due tipologie di icone: le icone "Product" che comunicano l'idea che sta alla base del prodotto con dimensione 48dp e le icone di sistema che individuano le azioni e i comandi all'interno delle applicazioni con dimensione 28dp.

Il Material Design offre linee guida anche per la gestione della scrittura nei diversi momenti dell'interfaccia, infatti un testo chiaro e conciso rende l'interfaccia utente più utilizzabile. La scelta del font non riguarda solamente l'estetica e lo stile dell'interfaccia, ma è determinante per l'immediatezza e la leggibilità del testo, il quale è essenziale nella search engine optimization (SEO) essendo riconosciuto dai motori di ricerca. Roboto e Noto sono i caratteri tipografici standard scelti da Google.

Nel Material Design il layout sfrutta griglie e modelli strutturali basati sui principi della stampa classica per rendere omogenea l'identità grafica dell'applicazione. Un

concetto chiave per adattare i layout allo schermo e ottenere un design dinamico in grado di adattarsi ai diversi device è quello dei “breakpoint”, punti cardine dove colonne, spazi, margini o elementi diversi si trasformano rendendo i layout responsivi, in grado quindi di adattarsi alle dimensioni diverse dei dispositivi utilizzati.

4.4.1 Material UI

Material-UI è un framework che fornisce un’ampia selezione di componenti React. Tali componenti sono integrabili e personalizzabili in maniera semplice e utili per la rapida costruzione di un’interfaccia che segua e integri i principi del Material Design. I componenti messi a disposizione da Material-UI sono indipendenti e funzionano senza configurazioni aggiuntive e, non basandosi su fogli di stile globali, permettono l’inserimento al loro interno solo lo stile ad essi destinato. Il framework Material-Ui utilizza Roboto come carattere predefinito.

L’installazione di Material-UI nella propria applicazione è semplice in quanto si può installare tramite npm eseguendo il comando `npm install@material-ui/core` oppure attraverso l’uso di CDN. Con l’installazione è permesso l’accesso a tutti i componenti principali del framework i quali sono utilizzabili come è riportato nell’esempio seguente:

```
import React from 'react';
import Typography from '@material-ui/core';

function App() {
  return <Typography variant="h2">Hello React</Typography>;
}
```

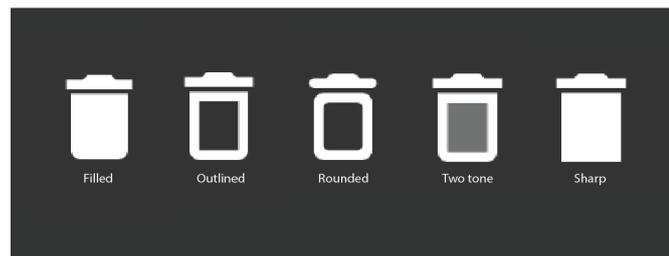


Figura 4.6: Icona Delete nei diversi temi

Le oltre 1100 icone ufficiali in formato SVG standardizzate da Material Design sono accessibili solo con l’installazione dell’apposito pacchetto tramite il comando `npm install@material-ui/icons` e l’elenco completo delle icone è disponibile su <https://material-ui.com/components/material-icon>.

Ogni icona è realizzata in cinque temi grafici diversi:

Filled è completamente piena; Outlined ha solo i bordi ed è vuota all'interno; Rounded ha i bordi arrotondati; Two tone ha due tonalità di colore; Sharp ha i bordi marcati.

Nella Figura 4.6 è possibile vedere l'icona "Delete" rappresentata nei diversi temi. Per ogni SVG si esporta il corrispondente componente React dal pacchetto @material-ui/icons il quale utilizza il componente material-UI SvgIcon per rendere l'apposito path SVG. L'uso delle icone è facile in quanto è sufficiente importarle all'interno dell'ambiente dove si devono utilizzare.

Il tema di default delle icone al momento dell'importazione è Filled, ma per cambiarlo basta modificare il percorso dell'importazione (@material-ui/icons/DeleteRounded). Quando è necessario usare un'icona SVG personalizzata, è possibile utilizzare il componente SvgIcon che estende l'elemento HTML5 nativo <svg>.

Il framework Material-UI mette a disposizione molti componenti in continua evoluzione divisi in nove categorie:

- la categoria Layout contiene tutti i componenti utili per la disposizione degli elementi nella pagina. Il componente Grid, ad esempio, permette la creazione di layout responsive che si adattano alle dimensioni dello schermo garantendo la proporzione tra le diverse dimensioni. Un altro esempio significativo è il componente Box il quale è in grado di wrappare l'elemento HTML <div> (di default) o altri componenti permettendo l'inserimento diretto di regole di stile;
- la categoria Inputs comprende tutti i componenti utilizzati dall'utente per interagire con il sistema. I componenti Button, Checkbox e Textfield, ad esempio, hanno una documentazione molto dettagliata;
- la categoria Navigation include tutti i componenti che servono all'utente per navigare nell'applicazione. Un esempio è Link, un altro esempio è Stepper il quale può essere utilizzato per creare form;
- la categoria Surfaces contiene componenti come Paper il quale imita sullo schermo le proprietà fisiche della carta, oppure Accordion che contiene diversi elementi e può espandersi o ridursi lasciando scegliere all'utente cosa visualizzare;
- la categoria Feedback comprende i componenti adatti ad offrire e a chiedere risposte all'utente. il componente Progress, ad esempio, esprime un tempo di attesa o un caricamento, mentre il componente Dialog chiede decisioni all'utente o ne focalizza l'attenzione su particolari attività;
- la categoria Data Display rappresenta diverse tipologie di componenti che hanno la funzione di mostrare i dati all'utente attraverso tipologie diverse

di visualizzazioni come ad esempio il componente List per gli elenchi e il componente Table per gli insiemi di dati. In questa categoria ci sono anche i componenti delle diverse icone.

- la categoria Data Grid contiene i componenti che consentono allo sviluppatore di gestire al meglio le tabelle in modo da organizzare al meglio l'esposizione delle informazioni.
- la categoria Utils include tutti i componenti che utilizza lo sviluppatore per migliorare il proprio sistema. Il componente Popover, per esempio, serve per mostrare all'utente il contenuto sovrapposto ad un componente. Le transizioni aiutano l'interfaccia utente ad essere più facile e piacevole da utilizzare. Nella sezione Transition sono elencati i componenti, ad esempio Fade, Collapse, Slide, che integrano in modo significativo l'utilizzo delle transizioni.
- la categoria Lab contiene tutti i componenti ancora in fase di sviluppo, i quali possono subire variazioni anche importanti, a differenza dei componenti delle altre categorie. Un componente molto utilizzato è Autocomplete il quale permette un text input che può essere arricchito mediante diverse opzioni.

Capitolo 5

Implementazione di XTap

All'esposizione teorica dei principali concetti utilizzati per la realizzazione della piattaforma XTap segue l'analisi riguardante l'effettiva implementazione del sistema. La piattaforma è stata progettata e validata attraverso un simulatore di eventi che ha permesso di testarla con dati simili a quelli provenienti dal mondo reale sia dal punto di vista quantitativo che nell'aspetto semantico. La filiera produttiva della carne bovina rappresenta il riferimento argomentativo.

5.1 Architettura dell'applicazione

La prima sezione analizza la struttura dell'applicazione. È necessario iniziare dalla schermata di login e dai processi di autenticazione utilizzati in quanto, per ottenere l'accesso, ogni utente deve essere autenticato.

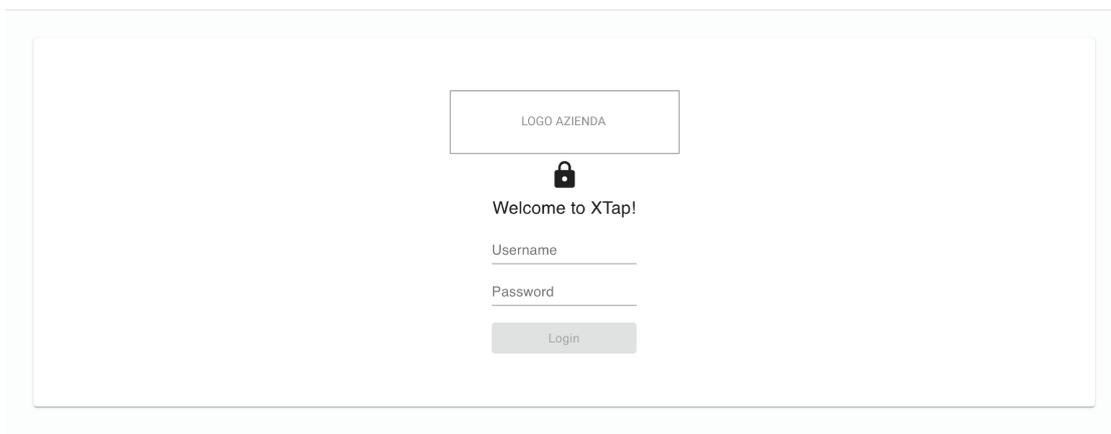


Figura 5.1: Schermata Login di XTap

5.1.1 Autenticazione

La piattaforma XTap ha come standard per l'autorizzazione OAuth (Open Authorization) il quale si basa sull'utilizzo di un access token. La schermata di Login (Figura 5.1) presenta un layout minimale, ma allo stesso tempo accurato, il quale comprende il logo dell'azienda committente, due campi di input (componente Text-Field) e un bottone attivato solo al riempimento dei due campi di input. L'utente accede compilando gli appositi campi del form e, nell'eventualità di un inserimento non corretto dei dati username o password, l'interfaccia mostra un messaggio di errore (Figura 5.2). L'autenticazione dell'utente avviene in seguito all'inserimento corretto dei campi del form, quando l'authorization server emette un access token e un refresh token i quali permettono all'applicazione di accedere alle risorse ospitate nel server. L'access token deve essere inserito nell'header per ogni richiesta effettuata ad una risorsa protetta per dimostrare di disporre dell'autorizzazione necessaria. L'access token ha una validità temporale, in XTap cinque minuti, mentre il refresh token, ha una durata sensibilmente maggiore, in XTap 24 ore. La maggiore durata del refresh token serve per il rinnovo dell'access token, infatti quando quest'ultimo scade, l'applicazione invia una richiesta e, attraverso il refresh token, ne viene generato uno nuovo. Allo scadere del refresh token l'utente accede all'applicazione solo tramite le credenziali. Tale meccanismo detto silent refresh, riducendo la frequenza dell'accesso manuale, migliora l'user experience.

JSON Web Token (JWT) è uno standard open (RFC 7519) di Token Authentication per la trasmissione sicura delle informazioni tra le parti come oggetto JSON.

I JWT sono compatti, leggibili e firmati mediante una chiave privata o una coppia di chiavi pubbliche dall'Identity Provider, in tal modo è possibile verificarne l'integrità. È un meccanismo di autenticazione stateless dove lo stato dell'utente non è memorizzato dal server.

La struttura di un JWT comprende tre parti: Header, Payload e Signature.

Header è un oggetto JSON in cui si enunciano la tipologia del token, in questo caso valorizzata a JWT, l'algoritmo utilizzato per la codifica e la firma. L'algoritmo può essere HMAC, RSA, HS256, SHA256 o RS256.

Payload è un oggetto JSON che contiene una serie di informazioni, claims, che è possibile categorizzare in tre blocchi. I claims standard che offrono informazioni sul token, ad esempio il tempo rimanente di validità, e i custom claims i quali, sfruttando la flessibilità della struttura JSON, permettono di inserire informazioni aggiuntive.

Signature è la parte più importante di un JSON Web Token in quanto è indispensabile per verificare la validità dei primi due campi.

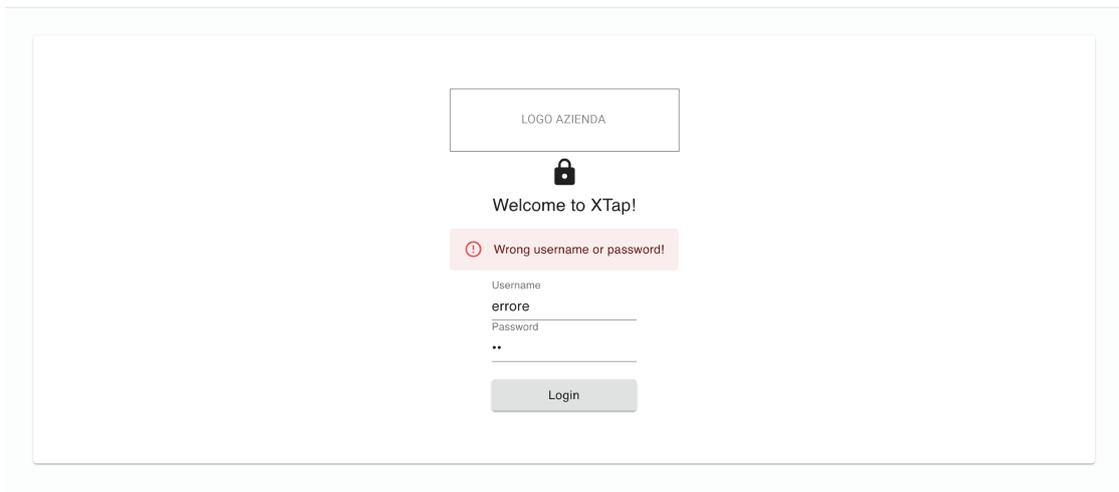


Figura 5.2: Schermata Login dati errati

Nel momento in cui l'utente inserisce username e password negli appositi campi e preme il bottone *Login* viene contattato l'endpoint */token* il quale, se la procedura e i dati precedenti risultano corretti, restituisce al client l'access token e il refresh token. Il client ha la possibilità di contattare le altre API della piattaforma utilizzando per autenticarsi l'access token ricevuto e memorizzato. Al fine di rendere il sistema più sicuro da attacchi esterni è sembrato opportuno mantenere l'access token all'interno del local storage in quanto è di breve durata, mentre il refresh token viene mantenuto su un cookie `HttpOnly`.

Attraverso il sito <https://jwt.io> è possibile decodificare un token JWT ricevuto a seguito di un'autenticazione corretta avvenuta inserendo username e password all'interno di XTap e osservare la struttura di header e payload.

Header :

```
{
  "typ": "JWT",
  "alg": "HS256",
}
```

Payload :

```
{
  "sub": "login_username",
  "cpname": "company_name",
  "company_prefix": "urn:gs1:prefix:0000000",
  "exp": 1604397827
}
```

All'interno del payload si possono inserire informazioni utili al client. In questo sistema, all'interno dell'oggetto JSON sono inseriti i campi "sub" che contiene l'username inserito nel form di login; "cpname" dov'è inserito il nome della company; "company_prefix" nel quale c'è il company prefix assegnato da GS1; "exp" che indica in secondi il restante tempo di validità del token.

AXIOS e useAxios

La libreria Axios presenta una sintassi precisa e pulita ed è facilmente installabile eseguendo il comando `npm install-save axios`, ciò ne ha favorito la scelta per l'invio delle richieste HTTP e per la gestione delle risposte all'interno della piattaforma. Per l'utilizzo di Axios si chiama la funzione `axios` e in seguito uno dei metodi forniti dalla libreria per rappresentare i diversi verbi HTTP. Axios lavora con le Promises: ogni promise è un oggetto contenente il risultato di un'operazione asincrona che può riuscire o fallire. Il successo dell'operazione permette l'ingresso nel `then` e l'esecuzione di cosa c'è all'interno; se l'operazione non ha successo, invece, l'ingresso è nel `catch` seguito dall'esecuzione di cosa si trova dentro.

L'esempio seguente di GET utilizza la libreria Axios:

```
let url = "my_url"
axios.get(url)
  .then( response => {
    gestione risposta
  })
  .catch( error => {
    gestione errore
  })
```

Ricorrere all'uso delle Promises, che gestiscono le operazioni asincrone, può rivelarsi, a volte, molto complesso. `Axios-hooks` risolve le difficoltà che emergono con una sintassi più semplice sfruttando tutta la potenza degli Hooks di React.

L'Hook `useAxios` restituisce tre elementi:

- `data`: oggetto che contiene la risposta del server;
- `loading`: booleano che indica se la richiesta è in sospeso;
- `error` se la risposta contiene un codice di errore con i relativi messaggi.

Sono questi tre elementi che permettono all'interfaccia di adattarsi alle varie fasi in maniera semplice e di mostrare in modo esaustivo ciò che sta accadendo. A seguito di una GET, ad esempio, la fase `loading` in cui si è in attesa di risposta è `true`, l'utente vede uno spinner che gli indica che è in corso il caricamento, mentre quando `loading` passa a `false` ci saranno o i dati di una risposta avvenuta con successo

nell'elemento data o un errore nell'elemento error.

Nell'applicazione XTap gli access token non sono rinnovati automaticamente, ma in funzione delle richieste fatte dal client, il quale contattando l'endpoint */token* si memorizza l'access token e il refresh token. Ogni richiesta fatta al server deve contenere nell'intestazione un token valido e il sistema, per inserire in ogni richiesta il token suddetto, usa il metodo interceptor di axios.

Gli interceptor sono metodi chiamati dopo il metodo principale, ne esistono di due tipologie: il request interceptor, chiamato prima dell'effettiva chiamata all'endpoint e il response interceptor, chiamato prima del completamento della promise e della ricezione dei dati dalla callback.

Prelevando dal local storage il token salvato, attraverso l'utilizzo di un request interceptor si inserisce il token nell'intestazione prima che venga effettuata ogni chiamata all'endpoint.

```
axios.interceptors.request.use(
  config => {
    const token = localStorage.getItem('token');
    if(token) {
      config.headers.authorization = `Bearer ${token}`;
    }
    return config;
  },
  error => {
    return Promise.reject(error)
  }
);
```

Nel caso in cui la richiesta fallisce perchè manca l'autorizzazione per accedere ai dati (401 Unauthorized) è possibile sfruttare il response interceptor. Il refresh token viene prelevato e si inserisce nell'header di una nuova richiesta la quale contatta l'endpoint */refresh_token_from_header*.

Se il refresh token è valido, un nuovo access token è restituito e salvato nel local storage, e dopo si ripete la chiamata HTTP precedentemente fallita. Tale meccanismo, completamente trasparente all'utente, permette di avere un token sempre valido.

Se invece anche la seconda richiesta fallisce significa che l'access token e il refresh token sono scaduti, allora l'utente viene reindirizzato alla schermata di login.

5.1.2 Routing

Nel paragrafo precedente sono stati illustrati i processi di autenticazione in XTap, e il routing, che è la capacità di muoversi tra le diverse schermate di un'applicazione, è strettamente collegato all'autorizzazione all'accesso a determinate risorse.

Un utente può navigare all'interno di XTap digitando direttamente l'url della pagina, utilizzando il drawer posto sul lato sinistro dell'interfaccia oppure cliccando su un elemento interno dell'applicazione.

In XTap il routing è gestito attraverso l'uso della variante react-router-dom della libreria React-router, variante specifica per le applicazioni web. Il pacchetto include una serie di router da scegliere in base alla piattaforma che si sta sviluppando. XTap è un'applicazione browser-based per cui la scelta è caduta su BrowserRouter che ha come figli tutti i componenti dell'applicazione.

Il componente `<Route/>` è uno degli elementi essenziali del pacchetto React-Router perchè esegue il rendering dell'interfaccia utente appropriata quando il path corrente corrisponde con il path definito nel componente `<Route/>`. Path è la proprietà del componente `<Route/>` che indica il percorso che si sceglie di assegnare al componente e che permette al componente di essere trovato e renderizzato. Inoltre mette a disposizione tre props utili per determinare quale componente renderizzare. Le tre props sono `component`, `render` e `children`.

`Component` è la props che definisce il componente React che verrà restituito dalla `Route`; `render` offre la possibilità di scegliere ciò che si vuole renderizzare e di passare ulteriori props all'elemento, si aspetta una funzione che ritorni un componente React; `children` si aspetta una funzione che restituisca un componente React. A differenza della prop `render` il componente viene restituito per tutti i percorsi indipendentemente dal fatto che il path corrente corrisponda con il path definito nel componente `<Route/>`.

La realizzazione del componente `PrivateRoute` permette di gestire il routing insieme al fatto che l'utente deve essere autenticato per accedere alle varie pagine dell'applicazione in XTap.

```
const PrivateRoute = ({component: Component, isAuthenticated, ...
  props}) => {
  return (
    <Route {...props}
      render={props =>
        isAuthenticated ? (<Component {...props} />
          ) : ( <Redirect to={{ pathname: "/Login",
            state: {referrer: props.location}
          }}
        ) />
    )}
  />
}
```

```
    );  
  }  
  
<BrowserRouter>  
  <PrivateRoute path="/" isAuthenticated={isAuthenticated}  
    component={Dashboard}/>  
  Altri PrivateRoute dell'applicazione  
</BrowserRouter>
```

Analizzando l'estratto di codice sopra è possibile osservare come il componente `PrivateRoute` riceva tre props: `path`, `isAuthenticated` e `component`. `Path` è il percorso al quale la route risponde, `isAuthenticated` contiene un valore booleano che indica se l'utente è autenticato o meno e `component` accoglie il componente `Dashboard` renderizzato nell'interfaccia utente. `PrivateRoute` restituisce un componente `Route` che si comporta in modo diverso a seconda se l'utente è autenticato o no, cioè se `isAuthenticated` è `true` o `false`. Se è `true`, è renderizzato il componente ricevuto come prop, in questo caso `Dashboard`, se è `false`, attraverso il componente `Redirect`, l'utente è reindirizzato alla schermata di login in quanto non possiede le autorizzazioni necessarie per accedere al percorso.

Ogni router crea un oggetto `history` utilizzato per tracciare la posizione corrente e rieseguire il rendering dell'applicazione tutte le volte che la posizione cambia. `BrowserRouter` utilizza l'Api della cronologia HTML5 per mantenere l'interfaccia utente sincronizzata con la URL nella barra degli indirizzi del browser. L'oggetto `history` racchiude al suo interno una serie di proprietà tra le quali c'è `location` che, a sua volta, contiene `pathname`, una proprietà che in seguito dimostrerò il suo valore.

5.1.3 Struttura generale

L'applicazione XTap, come mostra la Figura 5.3, si presenta con una struttura generale suddivisa in tre parti. La sezione superiore evidenzia l'utilizzo di un componente `AppBar` sempre presente, nella parte sinistra c'è un componente `Drawer` anch'esso sempre presente e nello spazio centrale rimanente, all'interno di un tag HTML `<main>`, le diverse pagine dell'applicazione.

All'interno dell'`AppBar`, a sinistra è posizionato il titolo della schermata corrente, `Dashboard` nella figura, e a destra c'è un componente `Chip` che ha come label il company name dell'azienda loggata. Il company name dell'azienda si ricava decodificando il token JWT. Cliccando, il `Chip` apre un menù (Figura 5.4) che offre al proprio interno la possibilità di navigare alla pagina della propria company oppure di effettuare il logout. Il titolo della schermata viene settato usando l'hook `useLocation` di React Router, il quale fornisce l'accesso all'oggetto `location`. Il titolo scelto per la schermata è settato ogni volta che l'oggetto cambia attraverso

l'utilizzo di un costrutto switch che ne verifica la proprietà pathname.

All'interno del drawer, nella parte alta, c'è un'area dedicata al logo dell'azienda committente, seguito da una lista che contiene sei componenti Accordion. Ogni componente Accordion offre all'utente la possibilità di espandere o ridimensionare il suo contenuto e consente di inserire al suo interno un AccordionSummary e un AccordionDetails.

In ogni AccordionSummary sono inseriti un'icona e un titolo relativo ad ognuna delle seguenti sezioni: Dashboard, Registry, Partners, Event, Models, Supply Chain e Setting. In ogni AccordionDetails, invece, si trovano i link che collegano alle pagine dell'applicazione. Quando l'AccordionDetails corrisponde alla pagina corrente è evidenziato in blu. Nel momento in cui l'utente effettua la login con successo la prima pagina che l'applicazione presenta è la schermata di Dashboard e di conseguenza, evidenziata in modo opportuno, l'espansione del corrispondente componente Accordion.

All'interno di un'applicazione realizzata con React è indispensabile utilizzare un reducer che, come si è visto nella sezione 3.5.3, permette una gestione dello stato più solida e più strutturata. Il meccanismo del reducer è usato nel componente <App>, che è il componente principale dell'applicazione XTap, perchè è in questo punto della gerarchia che è contenuta tutta la logica del sistema. Il componente può vedere state e dispatch e passarli ai propri figli tramite props, ma quando un'applicazione ha una struttura complessa com'è quella di XTap, diventa effettivo il rischio di creare una catena lunghissima di passaggio di props (prop drilling). Per eludere l'ingresso nel meccanismo di property drilling, la libreria React mette a disposizione un hook useContext il quale accetta un oggetto di tipo contesto e restituisce il valore di contesto corrente.



Figura 5.3: Layout di XTap

Un componente genitore registra un contesto, un oggetto React incapsula un dato con contenuti arbitrari, e ogni componente figlio sia diretto che indiretto lo può utilizzare senza passaggi attraverso componenti intermedi.

All'interno di XTap, alla radice, a disposizione della gerarchia, utilizzando il Provider, ci sono tre contesti: AuthContext, StateContext e ThemeContext.

AuthContext fornisce a tutta la gerarchia i dati ottenuti mediante la decodifica del token JWT: username, company name e company prefix, e l'accesso alla funzione SetAuthentication che permette di settare se l'utente è o non è autenticato (true o false).

StateContext mette a disposizione di tutti i componenti il reducer, mentre ThemeContext il tema scelto dall'utente.

I suddetti tre contesti, messi a disposizione alla radice, sono impiegati dai diversi componenti attraverso l'uso dell'hook useContext, il quale offre la possibilità di accedere in modo semplice ai contesti dichiarati in precedenza. L'hook ha un solo parametro: il contesto che si intende consumare, e restituisce il valore corrente contenuto al suo interno.

L'utente, mediante il contesto ThemeContext, seleziona il tema corrente e i suoi colori, ai quali accede ogni componente della gerarchia all'interno di XTap.

Utilizzando la destrutturizzazione si accede ai dati di contesto come segue:

```
const {primaryColor, secondaryColor} = useContext(ThemeContext)
```

Nella prima fase di sviluppo della piattaforma è stato studiato un solo tema principale che si avvale del colore primario #74d7b e del colore secondario #f5f5f5 e, come esempio, mette a disposizione altri due temi. La struttura dell'applicazione (Figura 5.5) è in grado di adattarsi al tema scelto selezionabile all'interno della pagina Settings.



Figura 5.4: Menù di XTap

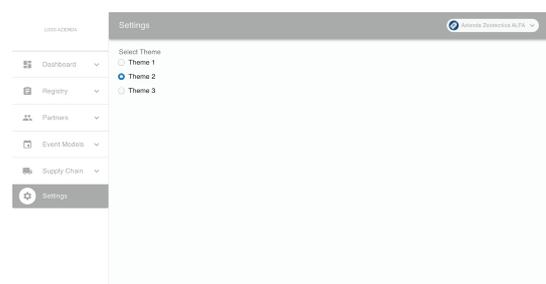


Figura 5.5: Secondo tema di Xtap

5.2 Analisi delle pagine

All'analisi della struttura generale dell'applicazione, dei processi di autenticazione e del routing segue l'esposizione delle schermate più significative seguendo l'organizzazione all'interno del drawer.

5.2.1 Dashboard

L'applicazione XTap pone le sue basi sugli standard GS1 e in particolare sullo standard EPCIS il quale descrive ogni evento attraverso quattro dimensioni What, Why, Where e When. Questa tipologia di descrizione ha ispirato la scelta nel processo di creazione del layout della pagina.

Nelle pagine in cui sono visualizzati molti dati, l'organizzazione è fondamentale per evitare confusione nell'utente. La schermata di Dashboard (Figura 5.6), che è la prima che si presenta dopo il login, ne è un chiaro esempio.

Il layout della pagina di dashboard (Figura 5.7) si divide in tre aree: a sinistra sono segnalati gli eventi più recenti; la parte destra, a sua volta divisa in quattro settori, mostra gli eventi aggregati nelle quattro dimensioni dello standard; in alto sono indicati i filtri selezionati dall'utente.

I problemi riguardanti le prestazioni emergono quando i dati ricevuti dal server e da trattare lato client sono molti, anche durante lo sviluppo della pagina Dashboard si sono manifestati, ma il superamento di tali difficoltà è stato possibile permettendo alla schermata di trattare migliaia di eventi con caricamenti nell'ordine di pochi secondi.

Name	Date	Time
Sale of boxes of slices of meat	2019-12-26	19:00:00
Slaughter	2019-12-26	09:00:00
Death of cattle	2019-12-26	09:00:00
Packing of meat slices	2019-12-26	04:00:00

Name - GTIN	#Items	#Events	Last Event
Vitello Piemontese 07876863582242	76	76	Healthcare treatment 2019-11-27 01:00:00
Vitello Frisona 03584606069984	90	90	Receiving of returned cattle 2019-12-11 15:00:00
Vitello Grigio Alpino 01672857805290	53	53	Birth of cattle 2019-11-01:00

Figura 5.6: Dashboard di XTap

L'utente, durante il caricamento e il trattamento dei dati, visualizza uno spinner con un messaggio "Preparing Data..." che lo rende a conoscenza delle operazioni che l'applicazione sta effettuando.

Gli eventi più recenti (Most Recent Events) sono evidenziati sul lato sinistro della pagina all'interno di una tabella, componente MaterialTable. La tabella permette all'utente di settare il numero di righe da visualizzare, 30 di default, di scorrere le diverse pagine di dati e di ordinare gli eventi per data in ordine crescente o decrescente.

Contattando l'API `epcis_events/`, inserendo alcuni parametri, si ottengono i dati da inserire nella tabella. I parametri relativi all'impaginazione sono `skip` e `limit`; per l'ordine degli eventi `direction` e altri ancora sono attinenti ai filtri. L'API restituisce un oggetto con due proprietà: `count` che indica il numero totale degli eventi e `data`, un array di oggetti di lunghezza pari al numero delle righe della tabella visualizzabili. Ogni riga della tabella mostra un singolo evento ed è cliccabile. All'estrema sinistra ci sono un'icona e il business step oppure, se disponibile, il nome e la data. L'icona è ricavata da un set di icone realizzato esclusivamente per XTap, ognuna delle quali rappresenta un business step. Cliccando su una riga della tabella si apre una dialog che mostra all'utente i dati dell'evento organizzati con la divisione nelle quattro W.

La Figura 5.8 mostra il dettaglio di un evento nella versione "Parsed View", ma è possibile anche selezionare "Raw View" che è un altro modello di visualizzazione che sfrutta la libreria `react-json-view`.

Le quattro diverse aggregazioni relative alle quattro dimensioni W che si vedono sul lato destro della pagina, sono gestite lato client e ciò ha richiesto una cura particolare per la creazione di un'interfaccia adeguata a livello di prestazioni.

Un esempio funzionale è il raggruppamento per la dimensione What, dove per ottenere tutti gli oggetti contenuti in tutti gli eventi bisogna possedere tra i propri dati tutti gli eventi relativi all'utente, che possono essere decine di migliaia. Un oggetto evento contiene al suo interno molti dati e, nella prima fase di sviluppo dell'applicazione, l'interfaccia risultava troppo lenta e di conseguenza diventava



Figura 5.7: Layout della dashboard

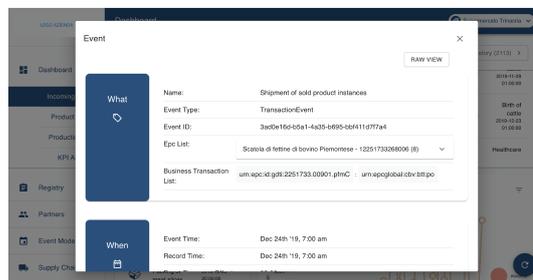


Figura 5.8: Dettaglio di un evento

evidente un problema di utilizzabilità. La soluzione migliore in termini di prestazione si è ricavata usando degli eventi “appiattiti”, dove invece di rappresentare un evento come un oggetto con tutte le sue proprietà, lo si rappresenta come un array contenente solo le informazioni essenziali per eseguire le aggregazioni. Il client ottiene questi eventi contattando l’API `epcis_events/flattened_items` che restituisce un oggetto con due proprietà: `items`, un array che ha all’interno tutti gli eventi nel nuovo formato, e `items_indexes` che contiene i riferimenti per mappare le posizioni dei dati all’interno dell’array. È inoltre possibile filtrare questi eventi per le proprietà `gtin`, `gln`, `why`, `start_date` e `end_date`. Mediante i dati ottenuti si eseguono le aggregazioni per le quattro dimensioni e si definiscono i dati necessari per le opportune visualizzazioni.

L’aggregazione per la dimensione What (Figura 5.6) è visibile in una tabella nella quale ogni riga mostra informazioni diverse.

A sinistra si trova il nome del prodotto e il suo GTIN, il numero dei prodotti (`#Items`), il numero di eventi in cui è stato rilevato (`#Events`) e, nell’ultima colonna, il nome e la data dell’ultimo evento rilevato. Ogni riga ha un’icona (`PinnedIcon`) che permette di selezionare un determinato What e di filtrare gli eventi della tabella di sinistra (`Most Recent Events`) e gli eventi “appiattiti” mostrati nella parte destra della pagina. Passando con il mouse sull’icona di ogni riga compare un tooltip (componente `Tooltip`) che segnala “Filter by GTIN” se la riga non è selezionata e “Clear filter” se il filtro sulla riga è attivo.

Il grafico dell’aggregazione per la dimensione When (Figura 5.9) rappresenta sull’asse delle ascisse un asse temporale e sull’asse delle ordinate un asse numerico: l’asse temporale suddivide le informazioni mediante l’anno e le settimane dell’anno. La curva arancione scuro illustra il numero di eventi, la curva più chiara il numero di oggetti.

Per la realizzazione del grafico è stata sfruttata la libreria `nivo` (<https://nivo.rocks/line/>) e in particolare il componente `ResponsiveLine` il quale, dato un array di serie di dati con un id e un array nidificato di punti con proprietà `x,y`, calcola la

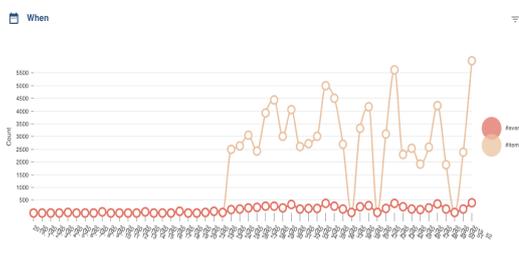


Figura 5.9: Grafico dimensione When

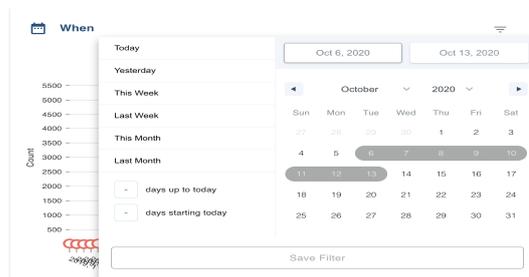


Figura 5.10: Filtro dimensione When

linea per ogni serie di dati.

Cliccando l'icona in alto a destra (FilterListIcon) si apre il data range picker (Figura 5.10) che offre all'utente la possibilità di selezionare o un singolo giorno o un range di date utilizzando modalità diverse: digitando sulla tastiera, selezionando un range sul calendario o usando i bottoni posti sulla sinistra. Cliccando sul tasto "Save Filter", dopo aver selezionato le date, si ottiene il filtro desiderato relativo a tutti gli eventi mostrati nella dashboard.

Una mappa, sulla quale con i pin si indicano i vari punti in cui si sono verificati gli eventi, permette di visualizzare l'aggregazione per la dimensione Where (Figura 5.11). La libreria *React Leaflet*, che adotta come React un approccio basato sui componenti, è stata scelta per la realizzazione del componente CustomMap, il quale è utilizzato per tutte le mappe che ci sono all'interno dell'applicazione in quanto è in grado di adattarsi alle diverse esigenze. Cliccando su un pin della mappa si rivelano i dati relativi alla posizione (Figura 5.11), il GLN dell'azienda, l'indirizzo, il numero di eventi che si sono verificati nel luogo (#Events) e il numero di oggetti (#Items). È anche possibile filtrare per luogo, cliccando sul bottone "Filter by GLN".

Una tabella permette la visualizzazione dell'aggregazione per la dimensione Why (Figura 5.12). Ogni riga della tabella, da sinistra verso destra, mostra il business step oppure il nome dell'evento più il business step con la rispettiva icona, il numero di prodotti (#Items) e il numero di eventi di una specifica tipologia (#Events). Anche in questa dimensione è possibile filtrare per un why cliccando sull'icona della riga (PinnedIcon) oppure rimuovere un filtro cliccando sull'icona apposita (UnPinnedIcon).

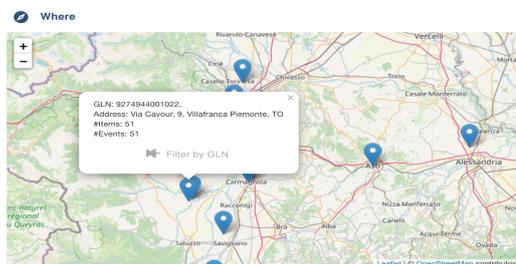


Figura 5.11: Dimensione Where

The screenshot shows a table with the following data:

Business Step	#Items	#Events
Receiving of purchased cattle (Receiving)	10	10
Death of cattle (Decommissioning)	7	7
Slaughter (Commissioning)	14	14
Meat maturation (Inspecting)	7	7
Shipment of carcasses (Shipping)	7	7
	-	-

Figura 5.12: Tabella dimensione Why

La schermata dashboard, quando l'utente filtra per le dimensioni What e Where, è visibile nella figura 5.13.

Nella parte alta della pagina è sempre visibile una sezione che mostra il riassunto dei filtri selezionati e permette di rimuoverli. Un componente Chip descrive all'utente l'icona relativa alla dimensione e il contenuto del filtro (es. Carcassa di vitello piemontese - 09274944029217).

In questa sezione, usando il bottone apposito, si possono rimuovere tutti i filtri, oppure mediante il bottone "Show History" visualizzare la storia del prodotto. Nella scritta "Show History(7)", la cifra 7 indica il numero di SGTIN dei quali cliccando sul bottone si può vedere la storia.

Ancora nella pagina dashboard di XTap, in basso a destra, attraverso il bottone di refresh è possibile ricaricare la pagina ottenendo dati aggiornati.

5.2.2 Product History

La pagina Product History rappresenta la storia di uno o più prodotti attraverso la visualizzazione delle trasformazioni avvenute, consultabili nella modalità tabellare (Table Mode) o a grafo (Graph Mode), le quali usano strutture dati diverse. Il passaggio da una visualizzazione all'altra è possibile con l'uso del corrispondente switch. Due differenti punti dell'applicazione permettono l'accesso alla pagina Product History: il bottone "Show History" e il drawer laterale. Il componente Autocomplete, che si incontra dopo l'accesso dal drawer, offre la possibilità di selezionare uno o più prodotti digitando l'SGTIN/LGTIN o il nome del prodotto stesso. Se invece si accede attraverso la Dashboard, già con gli SGTIN, non è possibile ricercare nuovi prodotti, ma solo consultare la storia dei prodotti preselezionati.

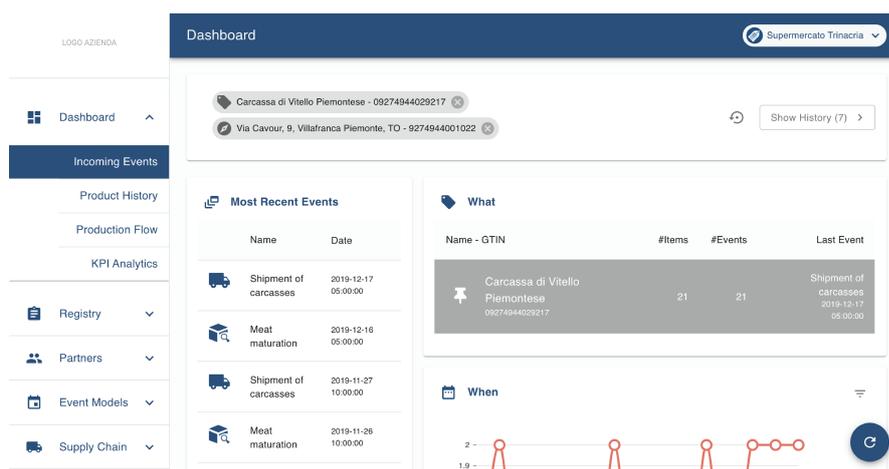


Figura 5.13: Dashboard e filtri

Lo use case solito di questa pagina è la verifica e l’origine dell’anomalia di un prodotto, questo controllo è fattibile da parte dell’azienda digitando un SGTIN o il nome del prodotto stesso.

L’acquisizione dei dati avviene contattando l’API `epcis_events/_product_history`, la quale restituisce al client un oggetto con all’interno quattro proprietà: `events` dove si trovano tutti gli eventi; `items` contenente un array di tuple che rappresentano le trasformazioni; i due dizionari `locations` e `names`: il primo offre informazioni sui luoghi partendo dal GLN; il secondo ottiene il nome del prodotto dal GTIN. Il client riceve questi dati e, attraverso un apposito algoritmo, crea un nuovo oggetto che comprende le proprietà `children` e `parent`. Ad ogni SGTIN/LGTIN sono assegnati tutti i suoi figli e i suoi padri e l’elenco degli eventi in cui esiste la relazione. La struttura dati appena creata serve per la visualizzazione in tabella della storia. Nella figura 5.14 è rappresentato uno use case con l’autenticazione a nome di “Supermercato Trinacria” che è l’ultimo elemento della supply chain. Lo use case riguarda la storia delle fettine di bovino adulto piemontese (`urn:epc:class:lgtin:2251733.0100191017`). L’interfaccia filtra tutti gli eventi nel tempo riguardanti gli oggetti che hanno portato alla realizzazione di quelle fette di bovino adulto piemontese. Il primo evento è un evento di nascita, un solo evento dimostra che tutte le fettine derivano da un solo bovino. Nel caso in cui, durante il processo, fossero state effettuate aggregazioni di lotti, potenzialmente si potrebbero vedere più nascite che avrebbero condotto alla realizzazione delle fettine di carne. L’interfaccia presenta una tabella divisa in quattro colonne. Ognuna di queste colonne rappresenta una dimensione `What`, `Why`, `Where` e `When` e, come evidenzia la figura 5.15, è possibile raggruppare per una o più di queste dimensioni. Per effettuare un raggruppamento, l’utente deve selezionare il titolo della colonna (es. `What`) e trascinarlo all’interno della sezione “`Drag headers here to group by`”. Successivamente è possibile ordinare la visualizzazione in ordine crescente o decrescente cliccando sulla freccia posta accanto al nome della dimensione. All’interno della visualizzazione, l’utente, utilizzando il comando `Search`, ha la facoltà di ricercare un dato specifico sia esso un `What`, un `Why`, un `When` o un `Where`.

What	Why	Where	When
Wolone Frisone - 0010702460050	Birth of cattle	Conso Farnocci, 105, Torino, TO - 0107824009009	Feb 0th '19, 1:00 am
Wolone Frisone - 0010702460050	Feeding of cattle	Conso Farnocci, 105, Torino, TO - 0107824009009	Mar 2nd '19, 1:00 am
Wolone Frisone - 0010702460050	Healthcare treatment	Conso Farnocci, 105, Torino, TO - 0107824009009	Mar 2nd '19, 1:00 am

Figura 5.14: Tabella Product History

Grouped By: When	Why	What	Where
When: Apr 1st '19, 1:00 am			
When: Feb 0th '19, 1:00 am			
When: Jun 11th '19, 10:00 am			
When: Jun 12th '19, 10:00 am			
When: Jun 12th '19, 11:00 pm			
Why: Disposal of carcasses			

Figura 5.15: Gruppi When e Why

La creazione della struttura dati per la realizzazione a grafo della storia dei prodotti è stata effettuata attraverso un algoritmo apposito partendo dai dati ricevuti dall'API `epcis_events/_product_history`. Per la realizzazione del grafo si è scelto di utilizzare la libreria `d3-dag` (Directed Acyclic Graph) facilmente installabile attraverso il comando `npm i d3-dag`. L'accesso alla visualizzazione è possibile cliccando sul bottone Graph Mode.

Nella rappresentazione a grafo (Figura 5.16) il prodotto selezionato è evidenziato con un pallino arancione, da lì sono osservabili tutti gli eventi pregressi e successivi relativi a quell'oggetto. Analizzando l'immagine, in progressione, si nota che le fettine di bovino adulto piemontese, all'interno della supply chain, sono inserite in un nuovo prodotto "Scatola di fettine di bovino adulto piemontese". Il grafo mostra tutte le fettine di bovino adulto piemontese ricavate dalla medesima sezione di bovino dalla quale deriva l'oggetto precedentemente selezionato. Osservando la storia pregressa invece, c'è la sezione di bovino dal quale giunge il prodotto selezionato, da quale carcassa (Carcassa di vitello piemontese) e da quale specifico animale (Vitellone Frisona).

È possibile contrarre o espandere ogni nodo. Il nodo si contrae, cioè si racchiudono al suo interno tutti i figli, cliccandoci sopra. Per espandere un nodo e per visualizzare in tal modo i dettagli, è necessario, invece, cliccare sulla sua lente d'ingrandimento. Tale procedura nel grafo della storia in XTap è equivalente alla ricerca per l'SGTIN/LGTIN dell'oggetto rappresentato nel nodo. Accanto alla precedente icona è posta l'icona di un occhio che, se cliccata, apre una dialog la quale mostra nella parte superiore tutti gli eventi ordinati in successione temporale e con il nome specifico in cui l'SGTIN/LGTIN è stato rilevato. La definizione di tutti gli eventi della storia, dalla nascita alla macellazione, del prodotto Vitellone

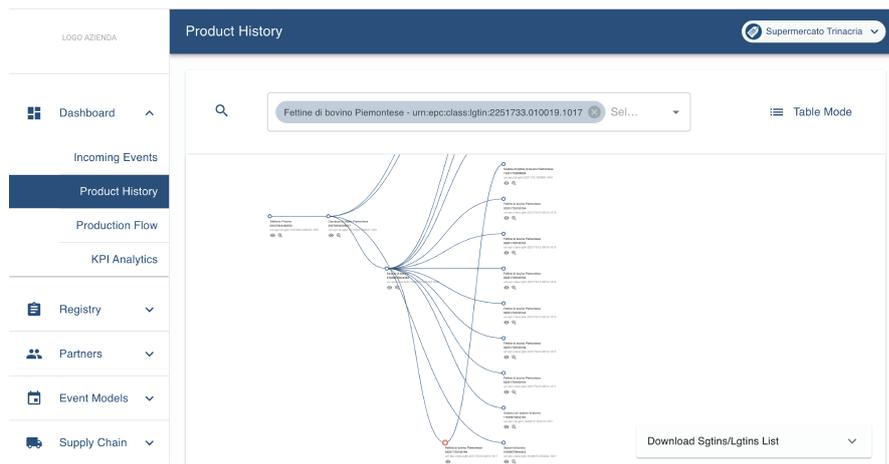


Figura 5.16: Product History rappresentazione a grafo

Frisona è visibile nella Figura 5.17. All'interno della dialog sono evidenti i dettagli di tutti gli eventi con la classica divisione delle 4W di EPCIS, cliccando sull'evento che si vuole osservare.

In basso a destra nella pagina che ha come summary "Download Sgtins/Lgtins List" si nota la presenza di un componente Accordion (Figura 5.16). Cliccando sul componente si selezionano tutti gli oggetti di una determinata tipologia coinvolti nel grafico e si scarica un file CSV che ha la seguente struttura:

```
"Product Name", "Sgtin / Lgtin"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1020"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1019"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1018"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1017"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1016"
"Fettine di bovino Piemontese", "urn:epc:class:lgtin:2251733.010019.1015"
```

Se all'interno del grafo si evidenzia un'anomalia, ciò può permettere di individuare l'oggetto difettoso e di richiederne la restituzione.

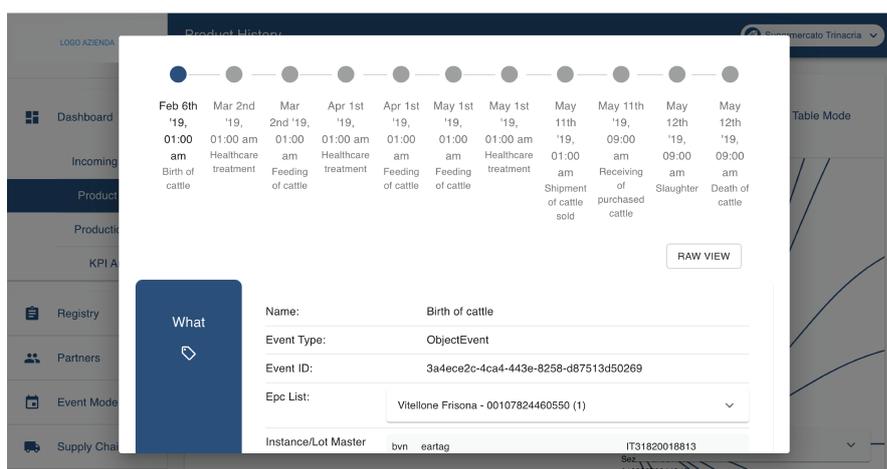


Figura 5.17: Product History dettagli evento

5.2.3 Production Flow

Il flusso di produzione della supply chain si trova all'interno di questa pagina ed è mostrato attraverso la visualizzazione a grafo molto potente ed apprezzata dagli utenti/clienti in quanto permette una visione efficace e chiara. Al momento della stesura di questo documento, lo sviluppo di questa parte dell'applicazione è ancora in corso siccome, essendo una parte fondamentale della piattaforma, richiede continui aggiornamenti e miglioramenti. La visualizzazione della pagina permette all'utente una visione dall'alto chiara e immediata, quindi semplice a livello globale, di ciò che si sta verificando nella catena. La progettazione di questa pagina e della sua visualizzazione ha come obiettivo aggiuntivo importante la possibilità di servirsi di uno strumento capace di entrare nei dettagli andando ad esplorare i singoli pezzi fino a mostrare anche i dettagli di ogni evento specifico. L'interfaccia della pagina (Figura 5.18) mostra nella parte centrale il grafo, in alto a sinistra un'area dedicata ai filtri e nella parte inferiore un pannello di controllo utile per gestire la visualizzazione all'interno del grafo.

Il grafo è stato realizzato utilizzando la libreria *dagre-d3* ed è basato su una struttura a nodi e a link. Quando l'utente accede alla pagina, ogni nodo, segnalato da un rettangolo blu (primary color), indica un luogo della supply chain. Dai nodi del grafo entrano ed escono diverse frecce che indicano il legame tra due nodi, l'esistenza di eventi che legano i due luoghi. Ogni nodo che mostra un luogo all'interno della catena offre all'utente tre informazioni: il nome del luogo (Macello Spinnato), il numero di eventi calcolati in base al raggruppamento selezionato nel pannello di controllo (un giorno di default) e una legenda, realizzata attraverso l'utilizzo della libreria *chroma*, che distingue con i colori i prodotti gestiti all'interno di un luogo. Passando sopra ai segmenti colorati, l'utente accede, attraverso un

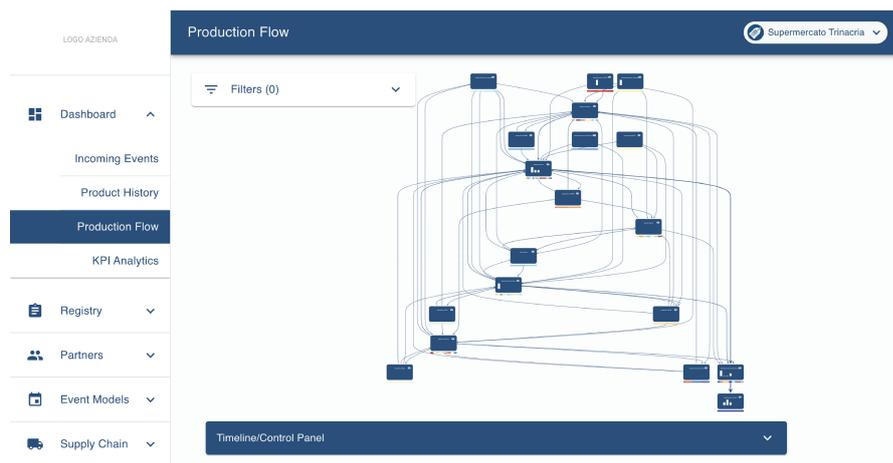


Figura 5.18: Pag. Production Flow

tooltip, alle informazioni sul nome e sul GTIN-14 dell'oggetto.

Espandendo il componente Accordion della Timeline/Control Panel (Figura 5.19), l'utente distingue come gli eventi sono raggruppati nella visualizzazione: 1 Hour, 4 Hour, 1 Day, 1 Week, 1 Month, 3 Months e 1 Year.

Sotto ai bottoni di selezione del raggruppamento, mediante una timeline, appare la divisione dello spazio di visualizzazione all'interno dei nodi. Analizzando la figura 5.19 è visibile il raggruppamento selezionato che è 1 Month e la timeline divisa in sette segmenti (luglio-agosto,...).

Un rettangolo/barretta bianco rappresenta ognuno dei segmenti all'interno del nodo e può avere altezze diverse. L'altezza definisce il numero di eventi in relazione al numero massimo di eventi nel raggruppamento temporale selezionato.

All'interno del Time/Control Panel l'utente mediante uno slider, che ha come estremo inferiore la data meno recente in cui si è verificato un evento e come estremo superiore la data più recente, può muoversi in senso cronologico e vedere in tempo reale i rettangoli/barrette muoversi all'interno dei nodi. Dentro ogni nodo che rappresenta un luogo c'è l'icona di un occhio. L'utente può espandere il nodo cliccando sopra questa icona oppure cliccando sul nodo stesso. L'espansione è utile per controllare nel dettaglio cosa si verifica in un luogo particolare, cioè i diversi business step. L'utente ha anche la possibilità di espandere tutti i nodi del grafo ottenendo una visualizzazione più complessa, ma più dettagliata di ciò che accade in tutta la supply chain.

L'espansione di un nodo è contrassegnata dal colore blu del raggruppamento, il colore rosso, invece, indica l'ultimo nodo espanso. Nel grafo (Figura 5.20) sono visibili due nodi espansi. Cliccando sull'area colorata che rappresenta il raggruppamento, è possibile contrarre il nodo e tornare alla visualizzazione precedente.

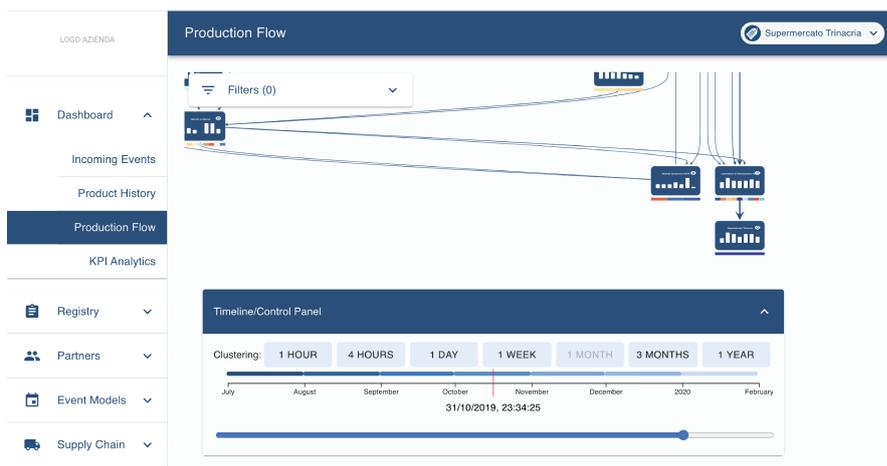


Figura 5.19: Componente Timeline/Control

Durante lo sviluppo dell'espansione e della contrazione dei nodi sono emerse importanti difficoltà di user experience dell'utente, in quanto ogni nodo veniva espanso immediatamente e la libreria utilizzata, per ottimizzare la visualizzazione, modificava la disposizione degli elementi nel grafo creando confusione. Il problema è stato risolto creando un ritardo intenzionale. In tal modo, cliccando su un nodo, si apre un componente appositamente costruito (Figura 5.21) che simula un caricamento in tre step: Updating flow, Finding Optimal View e Done. Al termine del caricamento l'utente assiste ad un cambio di distribuzione degli elementi del grafo invece che a una ridistribuzione immediata.

Quando un nodo è contratto e rappresenta un luogo fisico, i rettangoli/barrette che rappresentano il numero di eventi in un raggruppamento temporale indicano anche l'unione dei raggruppamenti di tutti i sottonodi. All'atto dell'espansione i nuovi nodi visualizzati, attraverso i rettangoli/barrette, in questo caso cliccabili, rappresentano il numero di eventi nel raggruppamento temporale scelto per il business step che rappresentano.

L'utente, cliccando su un rettangolo/barretta, attiva l'apertura di una sezione, visibile sulla destra dello schermo, che in una tabella espone gli eventi rappresentati dal rettangolo/barretta stesso. I suddetti eventi sono descritti con il nome, l'icona che definisce il business step e la data dell'evento per il quale la tabella è ordinata. Un semplice click su una riga apre una dialog a tutto schermo sulla quale l'utente vede i dettagli dell'evento selezionato suddivisi secondo l'ordine delle quattro dimensioni: What, When, Where e Why dello standard EPCIS. Ogni dettaglio è visualizzabile in due modalità: Parsed View, di default, e Raw View la quale, con l'uso della libreria *react-json-view*, permette la visualizzazione del JSON dell'evento da parte dell'utente.

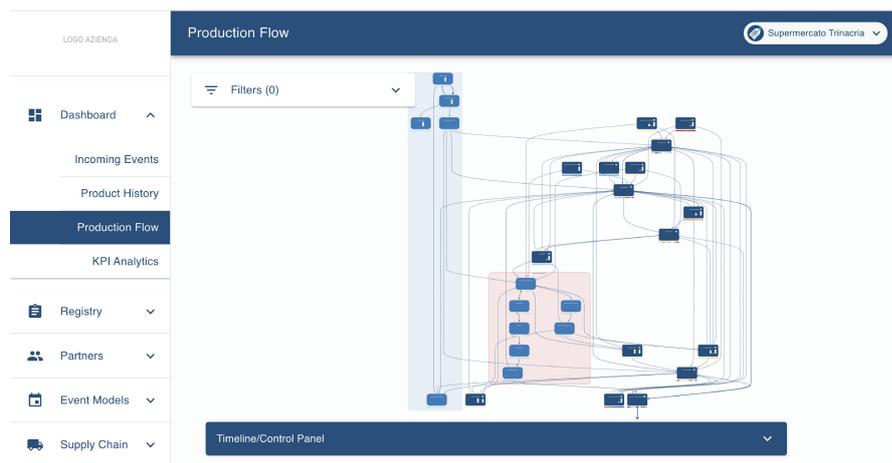


Figura 5.20: Espansione dei nodi

La sezione filtri (Figura 5.22) utilizza un componente Accordion posizionabile all'interno della pagina scelta mediante trascinamento.

In questa fase dell'applicazione è possibile applicare al grafo solo un filtro per prodotti della supply chain: lo sviluppo di un filtro per i luoghi è in fase di sviluppo. Digitando nell'apposito campo di input il nome o il GTIN-14 di un prodotto oppure scegliendo dall'elenco mostrato, l'utente filtra il grafo per i prodotti che desidera. In seguito alla selezione del filtro l'intero grafico viene ricalcolato dall'applicazione con eccellenti prestazioni funzionali. Un colore, che compare anche come background nel chip di sommario, identifica ogni prodotto (Figura 5.22).

Un buon livello di usabilità è raggiungibile solo se la notevole quantità di dati rappresentati all'interno della pagina è strutturata e gestita in modo efficiente. Contattando l'API `epcis_events/marble/v2` si ottiene un oggetto con tre proprietà: `nodes`, `older_datetime` e `most_recent_datetime`. All'interno della proprietà `nodes` c'è un array di oggetti nodo, mentre le altre due proprietà sono utili per costruire lo slider, presentato in precedenza, che permette all'utente di scorrere nel tempo tra l'evento più lontano nel passato e l'evento più recente. Attraverso l'API `marble/clustered_events/`, settando la proprietà "group" in base al raggruppamento scelto nel pannello Timeline/Control Panel (`one_day` di default), si ottiene per ogni node id un oggetto simile all'esempio che segue:

```
"5588522098454276": {
  "one_day": {
    "17897": {
      "event_count": 1,
      "items_count": 1
    },
    "18004": {
      "event_count": 1,
      "items_count": 1
    }
  }
}
```

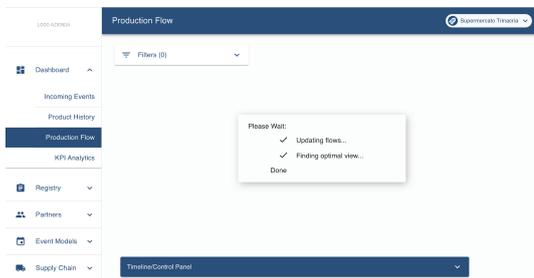


Figura 5.21: Simulazione caricamento

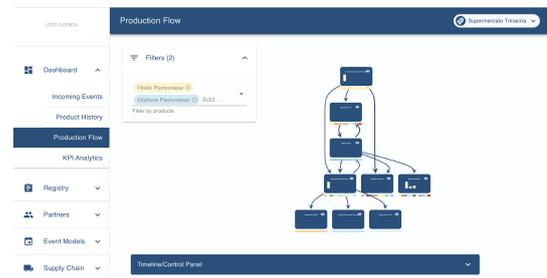


Figura 5.22: Filtro prodotti

La complessità della pagina richiede un reducer dedicato che permetta la gestione dell'intero stato del grafo in un unico punto il quale ha come stato iniziale l'unione di tutti i nodi del grafo con gli eventi clusterizzati.

La struttura dei dati richiesta dalla libreria usata per renderizzare il grafo è composta attraverso algoritmi ottimizzati all'interno del reducer. Tali algoritmi raggruppano i nodi in modo appropriato e gestiscono i collegamenti tra i nodi in base agli stati in cui essi si trovano, espanso o non espanso.

Loadash

La libreria Loadash permette di mantenere un codice minimale, ordinato e di facile comprensione perciò è stata impiegata per lo sviluppo dell'applicazione XTap. Un esempio significativo sono i cicli, importanti per i programmatori, e osservare come sono scritti nella modalità classica:

```
var arr = ['banana', 'kiwi', 'mela', 'fragola', 'arancia']
for (var i = 0; i < arr.length; i++) {
  console.log('fruit: ' + arr[i]);
}
```

e con l'ausilio della libreria Loadash:

```
var arr = ['banana', 'kiwi', 'mela', 'fragola', 'arancia']
_.each(arr, function(fruit, index, fruits) {
  console.log('fruit: ' + fruit);
});
```

La funzione `each` adotta come primo parametro l'array sul quale iterare e come secondo parametro una funzione che esegue ad ogni iterazione.

Altre funzioni della libreria utilizzate nell'applicazione sono `_.find`, `_.pullAt` e `_.uniq`.

La funzione `_.find` permette la ricerca con l'uso di più proprietà sostituendo l'iterazione attraverso un array mediante un ciclo di ricerca per ogni oggetto specifico.

```
var user = _.find(users, { lastName: "Segalini",
  gender: "maschio" })
```

La funzione `_.pullAt` rimuove dall'array gli indici e restituisce un array di elementi rimossi.

```
var array = ['a', 'b', 'c', 'd'];  
var pulled = _.pullAt(array, [1, 3]); // ['b', 'd']
```

La funzione `_.uniq` crea una versione dell'array priva di duplicati nella quale rimane la prima occorrenza di ogni elemento.

```
_.uniq([2, 1, 2]); // => [2, 1]
```

5.2.4 KPI Analytics

XTap, nel suo motore, ospita molti dati riguardanti le attività della supply chain e i prodotti. Le aziende che compongono la filiera possono usufruire di diversi servizi come il controllo in tempo reale del processo produttivo o la verifica di eventuali problemi. L'offerta dei servizi deve corrispondere alle effettive esigenze di ogni azienda in merito all'intera filiera e non può avere un approccio generico. XTap, attraverso un sistema di plugins privati realizzati in specifico per ogni azienda cliente, propone strumenti di analisi capaci di rappresentare i KPI richiesti.

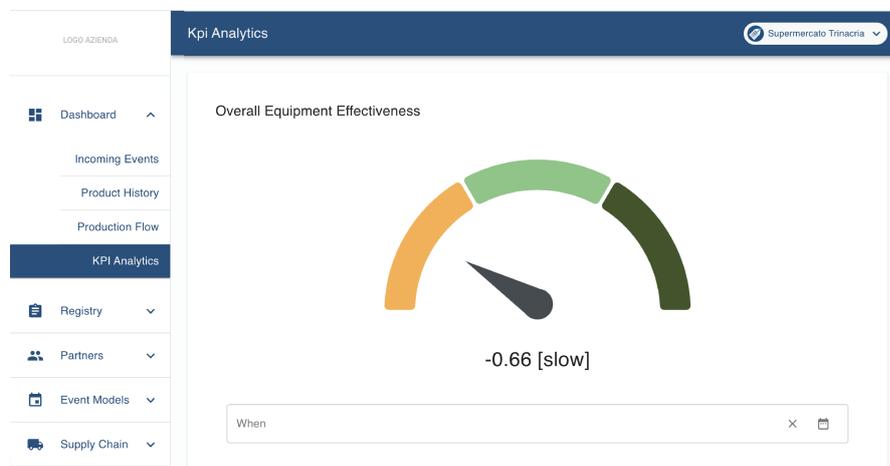


Figura 5.23: Indicatore globale KPI

Key Performance Indicators (KPI) identifica tutti gli indicatori utili per monitorare e misurare le performance e i risultati di un'attività, di un'azienda e/o di un processo.

Nella prima fase dello sviluppo della piattaforma XTap dove il test utilizzato è la supply chain della carne bovina è dimostrato l'effettivo funzionamento del KPI OEE (Overall Equipment Effectiveness) implementato all'interno. L'utilizzo di questo KPI è considerato la best practice per misurare e migliorare l'efficacia di un processo produttivo, è anche semplice da misurare e permette di eliminare gli sprechi.

L'OEE divide le prestazioni produttive in tre componenti separate: Disponibilità, Prestazione e Qualità, tutte migliorabili e misurabili.

La componente Disponibilità misura le perdite dovute a tempi di fermo pianificati oppure non programmati.

La componente Prestazione quantifica le perdite causate dal funzionamento nel processo dovute a tempistiche non adeguate e/o a interruzioni minori.

La componente Qualità misura le perdite motivate dalla produzione di un oggetto fabbricato non in conformità alle specifiche richieste.

$$\text{OEE} = \text{Disponibilità} \times \text{Prestazione} \times \text{Qualità}$$

Il livello medio accettato dell'OEE si trova tra il 45% e il 60%, anche se nei processi produttivi di notevole efficienza il livello può raggiungere l'85%.

Nella versione attuale della piattaforma XTap è stato sviluppato un plugin apposito per l'azienda "Supermercato Trinacria", in quanto è l'unica a raccogliere tutti gli eventi della filiera della carne bovina usata come test ed è l'unica in grado di monitorare lo stato della catena.

Il plugin ricava i tempi delle diverse lavorazioni intermedie partendo da quando il bovino è stato inviato alla macellazione. L'obiettivo è l'osservazione dei tempi nelle diverse fasi della filiera al fine di controllare quali punti funzionano nei tempi adeguati e quali, invece, hanno un funzionamento superiore o inferiore alle aspettative. Il confronto è dato da tempi standard relativi ad ogni passaggio all'interno della supply chain, tempi che rappresentano il valore di riferimento.

L'indicatore globale, posto nella parte superiore della pagina (Figura 5.23) indica se la catena ha un flusso più o meno adeguato. Nella simulazione di verifica sono stati inseriti dei ritardi, e l'indicatore, correttamente, indica la presenza di un flusso lento. Scorrendo nella pagina, per ogni operazione svolta all'interno della catena, ci sono altri indicatori (Selling received boxes of slices, Shipping maturated carcass, ...). Gli indicatori evidenziati dalla Figura 5.24 indicano quali operazioni, anche non direttamente gestite dal "Supermercato Trinacria", incidono in maniera significativa sulla lentezza della catena e quali operazioni si svolgono correttamente.

Colori diversi offrono una chiave di lettura del tempo: se la lancetta si trova nell'area verde chiaro l'operazione si svolge nei tempi previsti, se si trova nell'area



Figura 5.24: Indicatori KPI della supply chain

verde scuro, l'operazione è più veloce del previsto, se invece la lancetta è sulla parte arancione, l'operazione è lenta. Come mostra la Figura 5.23, valore indicato - 0,66, i valori sono normalizzati tra -1 e 1: estremamente lento corrisponde a -1; estremamente veloce corrisponde a 1; la prossimità dello 0 denota che l'operazione segue i tempi previsti.

La piattaforma XTap, mediante l'utilizzo di un apposito filtro, rende possibile la selezione di un range di date o di un singolo giorno e il controllo dei dati relativi al momento richiesto.

5.2.5 My Company

L'azienda loggata può vedere i propri dati all'interno della pagina My Company contattando l'API `companies/_by_prefix/` che restituisce un oggetto contenente tutti i dati necessari per popolare la pagina.

La pagina è divisa in due aree (Figura 5.25). Nell'area a sinistra compaiono il nome dell'azienda, una descrizione, il Company Prefix, il Fiscal Code, il Vat Number, la Supply Chain Label e un elenco di Places. L'elenco rappresenta i luoghi dell'azienda che sono descritti dal GLN, dalla città, dall'indirizzo e dallo Stato.

Nella parte destra della schermata c'è una mappa sulla quale dei pin indicano la posizione dei diversi luoghi dell'azienda.

“Supermercato Trinacria”, azienda loggata nell'applicazione, ha un solo luogo (Figura 5.21).

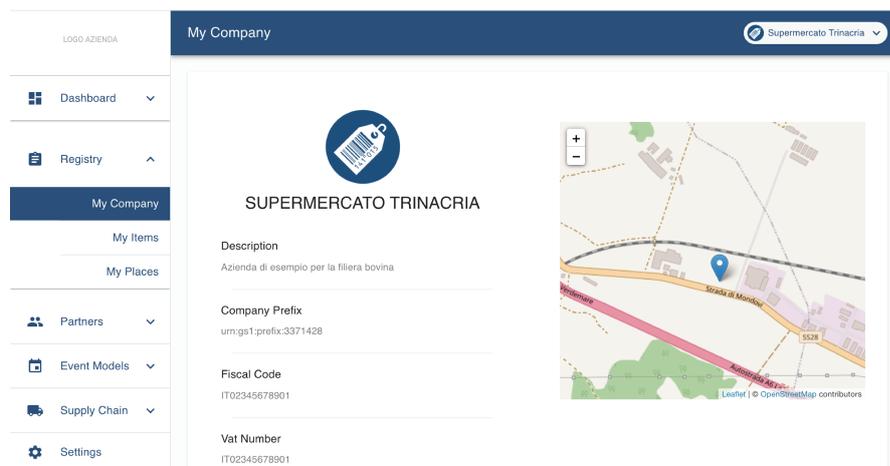


Figura 5.25: Pag. My Company

5.2.6 My Items

Un'azienda può visionare tutti i suoi prodotti attraverso la schermata My Items (Figura 5.26). Attraverso l'API `/items/` contattata mediante il settaggio della proprietà `producer` a `true` e `not-producer` a `false` (impostazioni di default), si ottiene un oggetto con due proprietà: `count` e `data`.

La proprietà `count` definisce il numero totale dei prodotti dell'azienda; la proprietà `data` contiene un array di oggetti prodotto.

Una tabella permette la visualizzazione dei dati dei prodotti: a sinistra è posta la colonna `name` all'interno della quale ci sono il nome e l'immagine del prodotto, `GTIN-14`, `Description` e `Tag`. All'interno della pagina, tramite la funzione `Search`, è possibile ricercare il prodotto desiderato utilizzando i valori di ciascuna delle colonne che lo rappresentano. L'utente può settare il numero di righe della tabella che vuole visualizzare e può scorrere tra le varie pagine, operando nell'apposito menù in basso a destra. Cliccando sul prodotto compare la pagina che lo descrive in dettaglio, la quale riceve direttamente tutti i dati del prodotto sfruttando la funzionalità dell'hook `useHistory`, il quale consente il passaggio dei dati tra pagine diverse.

L'esposizione dei dettagli di un prodotto (Figura 5.27) è gestita dividendo lo spazio in due settori: a destra l'immagine del prodotto (sostituita nell'esempio da un'immagine di default); a sinistra i dati descrittivi: nome, `GTIN-14`, descrizione, elenco di codici prodotto, `tags`, elenco di `custom features` e `attivo/non attivo`. Come non attivo s'intende che è terminato l'utilizzo del prodotto usato precedentemente.

All'interno della pagina My Items, cliccando sul bottone in alto "Add New", è possibile aggiungere un nuovo prodotto alla propria azienda. La schermata Add

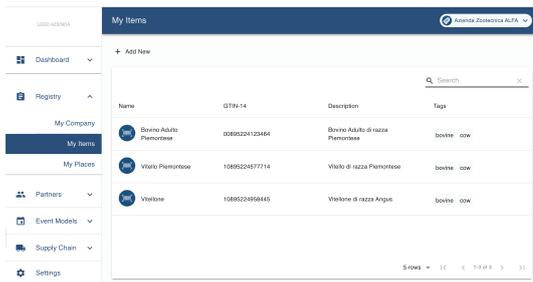


Figura 5.26: Pag. My Items

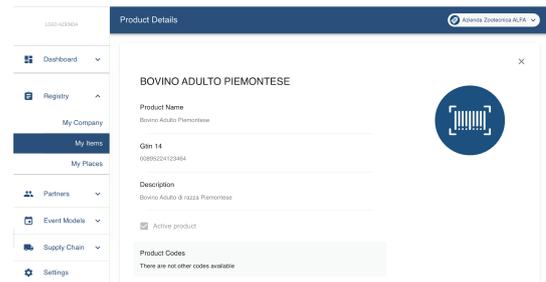


Figura 5.27: Dettaglio prodotto

Item (Figura 5.28), attraverso la compilazione di un form, ne permette l’inserimento indicando il nome, la descrizione e il GTIN-14, con l’eventuale aggiunta dei tags, delle custom features, dell’attivo/non attivo e di un’immagine. Al termine dell’operazione, cliccando sul bottone “Continue”, compare una pagina di sommario sulla quale l’utente può controllare le informazioni inserite. L’oggetto è aggiunto premendo il bottone “Complete” solo se sono stati immessi i dati minimi richiesti. Utilizzando solamente React, la realizzazione dei form e dei loro processi diventa difficoltosa, quindi conviene adoperare la libreria *React Hook Form* la quale permette di scrivere meno linee di codice, ha un’implementazione semplice, infatti mediante una singola chiamata hook rende possibile gestire una notevole complessità, ed è configurabile. La libreria react-hook-form inoltre ha la convalida integrata che può essere amministrata passando uno schema di convalida, rende possibile la modifica dell’output all’interno dello stesso form e infine, sfruttando la potenza degli hook, ha prestazioni molto elevate che riducono in modo significativo il numero di re-render della pagina.

Al termine della sezione è interessante fermare l’attenzione sul componente CodeProduct (Figura 5.29, all’interno del riquadro ‘Enter at least one code’) il quale, utilizzato all’interno del form della pagina Add Item, consente l’inserimento di uno o più codici prodotto garantendo che il primo codice inserito sia di tipo GTIN-14, requisito indispensabile per l’aggiunta di un oggetto, e assicura una corretta ed efficiente gestione dei cambi di stato in quanto al suo interno utilizza un hook useForm. Il componente CodeProduct usa un Autocomplete dove è l’utente che definisce la tipologia del codice. Se precedentemente nessun codice è stato inserito, il componente accetta solo la tipologia GTIN-14; se invece esiste un codice precedente, l’utente può scegliere la nuova tipologia.

Il componente TextField governa l’inserimento del codice del prodotto e il bottone IconButton che si attiva quando entrambi i campi sono valorizzati e che permette l’aggiunta di un codice. All’interno del componente CodeProduct sono

visibili i codici inseriti precedentemente che l'utente ha la facoltà di rimuovere. Gli elementi basilari di questo componente sono stati impiegati in altri componenti utilizzati all'interno del medesimo form, un esempio importante è il componente CustomFeature.

5.2.7 My Places

I luoghi dell'azienda sono mostrati all'interno della pagina My Places e si ottengono attraverso il contatto con due diverse API.

L'API `companies/_by_prefix` restituisce un oggetto che ha all'interno tutte le proprietà dell'azienda con cui si è autenticati. Tra queste proprietà c'è un array di place (GTIN), dove per ogni GTIN si invia una richiesta All'API `places/_by_gln/?gln=` che ritorna un oggetto place per ottenere i dettagli. Tutti i luoghi dell'azienda sono salvati dall'hook `useState` in uno stato all'interno della pagina.

My Place è una pagina che offre all'utente due diverse tipologie di visualizzazione: una tabella e una mappa.

Mediante la funzione `Search` nella `List View` viene avviata la ricerca di ognuno dei luoghi elencati nella tabella dove ogni luogo è descritto da Stato, città, indirizzo e GLN. Nella specifica sezione posta in basso a destra, come indicato anche per le altre tabelle dell'applicazione, è possibile scorrere tra le pagine e selezionare il numero delle righe da visionare. Cliccando su una riga della tabella, l'utente accede al dettaglio del luogo scelto. Nella parte destra della pagina di dettagli troviamo la posizione sulla mappa, nella parte sinistra le informazioni sul luogo: GLN, Country, City, Address. Il tasto `Map View` consente l'accesso alla mappa (Figura 5.29) sulla quale ogni luogo è indicato da un pin. Cliccando sul pin prescelto, si apre un tooltip che indica il GLN e l'indirizzo.

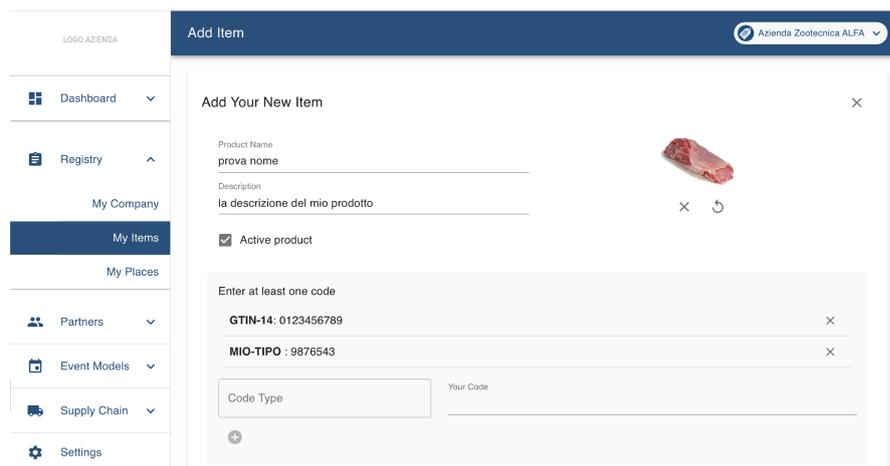


Figura 5.28: Pag. Add Item

Nella parte in alto della pagina My Places, attraverso il bottone Add New si accede al form di creazione di una nuova place (Figura 5.30).

Per la realizzazione del form, così come per la creazione di un nuovo oggetto, sono state sfruttate le potenzialità della libreria *react-hook-form*.

Un'azienda, per aggiungere un luogo all'interno del proprio database deve definire lo Stato di appartenenza, la città e il GLN. Se l'azienda non ha il proprio GLN, cliccando sul bottone posto accanto all'input, può richiederlo direttamente alla piattaforma XTap.

L'utente inoltre può aggiungere l'indirizzo, la latitudine, la longitudine e altre informazioni all'interno del GLN dichiarato (SGLN).

Per la selezione dell'indirizzo e delle relative coordinate di latitudine e longitudine ci sono a disposizione due modalità e attraverso uno switch è possibile passare da una all'altra.

La prima modalità (di default) sfrutta le potenzialità del geosearch di Leaflet e consente all'utente di digitare l'indirizzo scegliendolo in una lista così i dati (indirizzo, latitudine e longitudine) sono precisi.

La seconda possibilità permette la navigazione all'interno della mappa e definire l'indirizzo e i dati conseguenti cliccando direttamente sul luogo. I campi del form saranno popolati automaticamente.

Per aggiungere gli SGLN è stato progettato un componente che si avvale degli stessi principi del componente Code Product analizzato nella sezione 5.2.6. Il bottone Continue offre l'accesso alla pagina di sommario nella quale l'utente può controllare le informazioni inserite. Se non sono stati compilati i campi richiesti verrà segnalato un errore.

Cliccando sul bottone Complete il nuovo luogo farà parte dei luoghi dell'azienda e sarà visualizzabile.

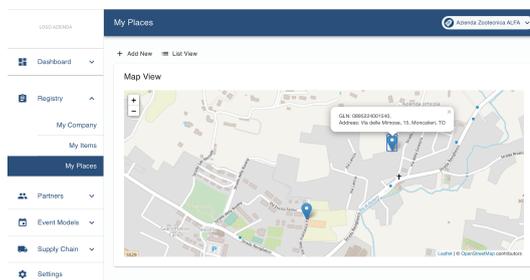


Figura 5.29: Pag. My Places

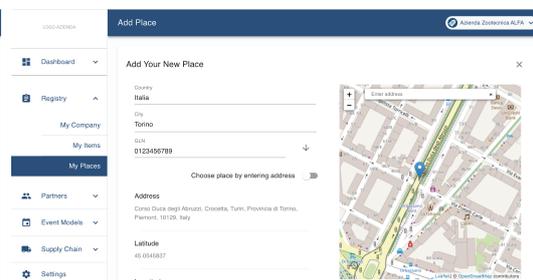


Figura 5.30: Pag. New Place

5.2.8 Companies e Others' Items

Le due pagine Companies e Others' Items si trovano nella sezione del drawer Partners.

Nella pagina Companies l'utente, in una tabella, visualizza le aziende che fanno parte di XTap. La descrizione di ogni azienda avviene attraverso il Supply Chain Label, il Name e l'immagine, il Company Prefix, il Fiscal Number e la Description. Utilizzando il menù in basso a destra, l'utente sceglie di visualizzare un determinato numero di righe della tabella e di scorrere tra le varie pagine, e attraverso l'apposita funzione ricerca un'azienda in particolare attraverso uno dei campi che la descrivono. Cliccando su una riga della tabella l'utente è reindirizzato alla pagina dei dettagli dell'azienda. I dati di dettaglio di ogni azienda sono gli stessi visti nella pagina My Company nella sezione 5.2.5 (Figura 5.25).

Nella pagina Others' Items l'utente visualizza tutti gli oggetti all'interno della piattaforma XTap che non appartengono all'azienda con cui è autenticato.

L'API `/items/?not_producer=true&producer=false` fornisce i dati da visualizzare. La proprietà `not_producer` è settata a `true`, al contrario avviene per la proprietà `producer` a `false`, rispettivamente settate a `false` e `true` di default.

Come nella pagina precedente i dati sono visibili in tabella, l'utente sceglie il numero di righe da visualizzare e può scorrere tra le varie pagine. Altra caratteristica comune con le altre tabelle della piattaforma è la ricerca di un determinato oggetto attraverso una sua proprietà. La descrizione di ogni oggetto è completa di Name e immagine, GTIN-14, Description e Tags. Cliccando su uno specifico oggetto si apre l'accesso alla pagina di dettagli dell'oggetto, pagina analizzata nella sezione 5.2.6 (Figura 5.27).

5.2.9 Event Models e Create New Event Template

I modelli degli eventi illustrati all'interno di XTap sono visualizzabili all'interno della pagina Event Models (Figura 5.31). Ogni azienda utente può creare dei propri modelli di evento attraverso l'apposito form nella pagina Create New Event Template oppure può utilizzare modelli preesistenti propri del sistema. Prima di procedere all'analisi della pagina e dei suoi aspetti di interfaccia è opportuno chiarire cosa sono i modelli di eventi.

Lo standard EPCIS, descritto nel capitolo 2, fornisce un dizionario di eventi per elencare le diverse operazioni che possono accadere all'interno di un processo produttivo. Lo standard però fornisce una definizione generica che ricopre le casistiche più comuni poichè, essendo notevole e vario il numero dei casi, non è in grado di entrare nel dettaglio di ogni singolo evento. I modelli di eventi predisposti o che è possibile costruire all'interno della piattaforma XTap hanno la funzione di evolvere e spiegare in dettaglio gli eventi che lo standard EPCIS definisce in modo generico i quali rischiano di essere troppo sintetici per un'analisi accurata della supply chain.

Lo standard EPCIS indica la nascita di un vitello, evento che si verifica all'interno dei dati del test, come la creazione di un oggetto cioè un evento di commissioning. Nella realtà, l'evento preso d'esempio, non è la semplice creazione di un oggetto, ma è la creazione di un oggetto con una tipologia specifica che può essere descritto da diverse caratteristiche come la data di nascita oppure da un suo preciso dato come l'ear tag.

Ogni modello di evento, un esempio è "Birth of Cattle" è composto dallo standard partendo dalle dimensioni 4W e indica in modo dettagliato come deve essere realizzato l'evento specifico cioè quali sono i campi obbligatori per la generazione di un evento con quel nome.

Per un evento "Birth of Cattle" all'interno dell'event model viene imposto che il business step sia un commissioning e questo a sua volta porta a forzare un'altra serie di parametri da inserire durante la creazione degli eventi.

La piattaforma XTap mostra allo stato attuale 45 modelli di eventi che corrispondono a tutte le operazioni eseguite all'interno della supply chain utilizzata come test. Oltre ai modelli già realizzati esiste la possibilità di creare e visualizzare modelli di evento dedicati a specificare e descrivere ulteriormente ogni particolare operazione avvenuta all'interno della catena.

All'interno della pagina Event Model, la piattaforma XTap, per ottenere i dati da visualizzare, contatta l'API `capture/event_templates` che restituisce un array di oggetti modello di evento visibili in una tabella nella quale ogni modello è descritto con il Name, l'icona che illustra il business step e l'Event Type. Il percorso di ricerca attinente ad un particolare modello di evento segue la stessa procedura delle precedenti tabelle. Cliccando su un modello si ottiene l'accesso alla pagina di dettaglio dell'evento (Figura 5.32) e l'utente può scegliere tra due tipi di visualizzazione: Parsed View (di default) o Raw View.

La visualizzazione offerta dalla Parsed View rimanda alle linee guida dello standard EPCIS dove le varie proprietà sono divise secondo le quattro dimensioni What, When, Where e Why.

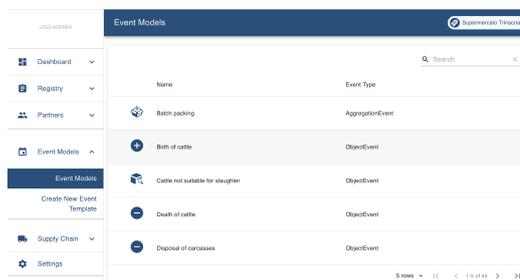


Figura 5.31: Pag. Event Models



Figura 5.32: Pag. Event Detail

La pagina Create New Event Template (Figura 5.33) permette di creare al suo interno nuovi modelli di evento. L'applicazione, all'interno della schermata, richiede di essere in possesso di diverse informazioni: dati sugli oggetti e sulle aziende, e i dati del CBV (Core Business Vocabulary). Tali dati si possono ottenere contattando varie API.

Il componente Material UI Stepper ha permesso la realizzazione del form e la visualizzazione del suo completamento progressivo mediante dei passaggi numerati. La creazione è stata sviluppata attraverso sei step: Event Typology, What, Why, Where, When e Instance Lot Master Data.

Nello step Event Typology, l'utente deve definire il nome dell'evento e indicare, mediante il componente RadioGroup, l'Event Type scegliendo tra ObjectEvent, AggregationEvent, TransactionEvent e TransformationEvent. La scelta dell'Event Type modifica la struttura del form e, come in tutti gli altri step, è possibile proseguire con il tasto NEXT solo se i campi sono corretti e validati dall'applicazione. Nello step What, l'utente deve indicare il genere di oggetto che intende tracciare, selezionare l'oggetto o gli oggetti usando un componente Autocomplete che permette una scelta multipla.

Nello step Why, l'utente deve scegliere il business step da un elenco predefinito, la tipologia di Action, il Disposition Input Type e il Disposition Value che indica in che stato si trova l'oggetto.

Lo step Where, uguale per tutti gli Event Type, permette di selezionare il Business Location Input Type e il Read Point Input.

La dimensione When rende possibile la selezione dell'Event Time Input settato di default a computed, e dei valori di Record Time Input Type e UTC Event Timezone Offset i quali possono essere settati a due valori: computed ed external. Nel valore computed, l'utente ricava il valore al momento in cui crea l'evento; in external, l'utente deve specificare il valore quando crea l'evento.

L'ultimo step, relativo agli Instance Lot Master Data consente all'utente di inserire una lista di attributi.

Il componente Stepper riduce le difficoltà insite nella compilazione del form che rimane comunque complesso poichè richiede all'utente la conoscenza di tante informazioni e un certo livello di competenza per il loro corretto inserimento. Le conseguenze più probabili sono lo scoraggiamento e la rinuncia e, per ovviare a ciò, sotto ogni passaggio una semplice comunicazione funge da guida informando sulle conseguenze delle scelte compiute.

5.2.10 Graph e Create Relationship

Nella piattaforma XTap, all'interno della pagina Graph, il titolo Supply Chain presenta la visualizzazione della filiera produttiva e illustra le regole di condivisione

degli eventi tra i vari attori.

Il grafo è stato realizzato utilizzando la libreria *nivo* e in particolare il componente *ResponsiveChord*. Il modello di grafico a corda è il più adeguato per la riproduzione grafica di flussi o di relazioni tra le diverse entità chiamate nodi. Ogni entità è rappresentata da un frammento sulla parte esterna della circonferenza e gli archi collegano le diverse entità.

Nel grafo della pagina, sulla circonferenza sono posti tutti gli attori della supply chain e sono visibili gli scambi di dati tra gli attori.

Il “Supermercato Trinacria” (Figura 5.34) con il quale si è autenticati riceve informazioni da tutti gli attori della filiera, ma non condivide i propri dati con gli altri, è dunque, all’interno della catena, solo un consumatore di eventi.

Nella parte in alto della pagina si vanno a scegliere la filiera produttiva, in questo caso “Bovine meat” che corrisponde alla filiera della carne bovina del test dell’applicazione, quali tipologie di regole visualizzare, ad esempio 4W Events o Master Data, e un filtro che indirizza alla selezione dell’azienda. Al momento il filtro non è ancora disponibile, ma è una feature che verrà inserita negli sviluppi futuri.

L’utente, scorrendo il mouse sopra ognuno dei collegamenti del grafo, vede attraverso un tooltip quali sono gli attori protagonisti della relazione e cliccando sul tooltip specifico apre un dialog che all’interno contiene i dettagli della relazione. Ogni relazione è definita da un nome, da una source, da una destination, dal nome della filiera produttiva, dalla tipologia di regole da visualizzare e dalla condizione attivo/non attivo.

L’applicazione XTap ottiene i dati visualizzati all’interno della pagina contattando l’ API `/supply_chains` la quale restituisce un oggetto con la proprietà `count`, che è il numero di supply chain, e un array di oggetti contenenti il nome e la descrizione

Figura 5.33: Pag. Create New Event Template

delle supply chains disponibili. L'API `/rules/_filtered`, contattata, ritorna invece un array di regole le quali, dopo l'intervento del client, rimangono visualizzabili all'interno del grafico. Utilizzando l'apposito form nella pagina Create Relationship è possibile creare le regole esposte nella pagina Graph.

La creazione delle relazioni di smistamento degli eventi richiede un'interfaccia apposita (Figura 5.35) dove si è autenticati alla piattaforma come "Supermercato Trinacria", il quale, essendo solo un'entità che riceve, non ha relazioni con altri partner e l'unica operazione possibile è la creazione di una nuova relazione.

Il grafico a corde (Figura 5.34) evidenzia le regole definite che hanno tutti gli altri attori della catena e che conducono al "Supermercato Trinacria". Agli altri attori basta autenticarsi alla piattaforma, ad esempio come Macello Spinnato, per trovare tra le relazioni esistenti le due prime relazioni che convergono verso il "Supermercato Trinacria": una 4W Events e una Master Data. Se esistono già altre relazioni e l'utente ne intende modificare una, può selezionare il nome della regola nel componente Autocomplete dedicato in modo da avviare in automatico il popolamento del form con i dati corretti. Successivamente l'utente ha la facoltà di modificare i campi desiderati e di salvare le modifiche.

La creazione di una nuova relazione di smistamento inizia con la selezione di "New Relationship" e con un nome identificativo.

Nella compilazione del form, l'utente deve indicare un destinatario cioè un'azienda esistente all'interno della piattaforma XTap. Esiste anche l'opzione dove ogni relazione è taggata con il nome di una supply chain per comunicare agli altri attori della filiera che la relazione è esistente e che due entità si stanno scambiando informazioni. In assenza di un tag la relazione rimane privata per l'utente autenticato. La creazione si completa con la definizione di una o più regole.

L'utente può scegliere e selezionare due tipologie di regole: una riguarda i Master Data e permette di condividere con gli altri attori le informazioni sul prodotto che ha precedentemente scelto. Nella realizzazione di una regola di tipo Master Data è anche possibile scegliere quali luoghi, Location e quali prodotti, Product, inserire al suo interno.



Figura 5.34: Pag. Supply Chain

Figura 5.35: Pag. Nuova Relazione

L'altra tipologia riguarda gli eventi prodotti, selezione 4W events di default. Una regola 4WEvents offre all'utente l'opportunità di indicare delle condizioni sulle 4W dello standard EPCIS, su tutte e quattro le dimensioni o anche su una sola di esse. L'utente procede al salvataggio della regola premendo il bottone "Save Rule". Nella parte alta del componente è visibile l'insieme delle regole create e salvate. L'utente, usando gli specifici bottoni, può modificare ogni regola o eliminarla.

La dimensione What può essere indicata dall'utente in quattro forme diverse, ovvero mediante il Product Name, il Prefix, un range di SGTIN oppure tutti i prodotti. Selezionando Product Name, attraverso un componente Autocomplete l'utente può scegliere un prodotto in un elenco di prodotti conosciuti dalla piattaforma XTap, definiti da nome prodotto-azienda detentrica del prodotto; selezionando invece Prefix o Sgtns range , l'utente deve inserire i codici corretti. L'aggiunta di una condizione di Why richiede all'utente l'indicazione del Business step, con le regole EPCIS, oppure dell'Event Name il quale specifica meglio l'operazione. L'utente può decidere di selezionare l'invio degli eventi relativi ad una particolare location, condizione Where possibile attraverso il GLN o attraverso il Read point. In ultimo l'utente, condizione di When, ha la facoltà di decidere delle regole con delle tempistiche determinate, come ad esempio la selezione di un arco temporale nel quale dati inviare ad un partner.

La creazione di una relazione tra il "Supermercato Trinacria", con la quali si è autenticati, e Marcello Spinnato è dedicata ad illustrare gli effetti di una relazione. In primo luogo l'utente determina il nome della relazione, nell'esempio "prova_relazione" e il destinatario "Marcello Spinnato", poi definisce un'unica regola che contiene le condizioni sulla dimensione What e sulla dimensione When.

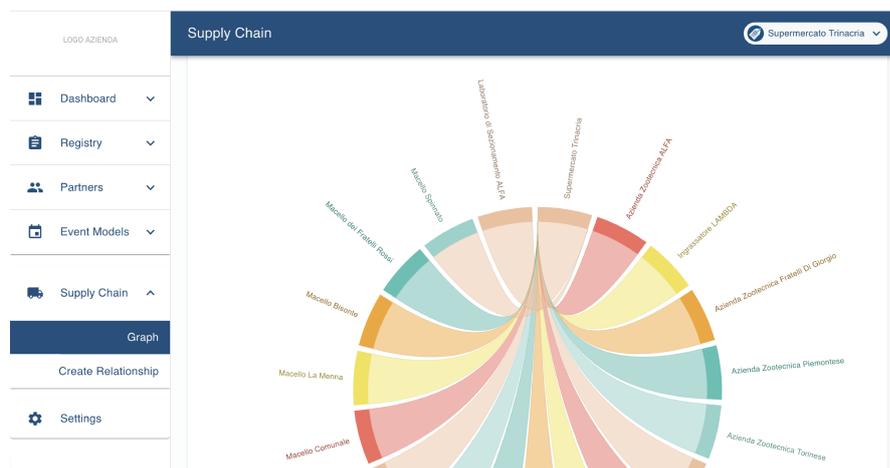


Figura 5.36: Pag. Supply Chain con nuova relazione

Quando l'utente preme sul bottone "Create Relationship", l'oggetto inviato al server nella POST dell'API */rules* ha la seguente struttura:

```
{
  active: true
  chain: "bovine_meat"
  destination: "xxxx"
  id: null
  name: "prova_relazione"
  partner: "urn:gs1:prefix:1639070"
  rules: Array(1)
    0:
      biz_location: null
      biz_step: null
      event_name: null
      read_point: null
      what:
        name_types: "product_name"
        prefixes: ["urn:epc:id:sgtin:1639070.044296.",
                  "urn:epc:class:lgtin:1639070.044296."]
      when:
        end: "2020-11-30T11:59:59"
        start: "2020-11-01T12:00:00"
      source: "xxxx"
      type: "4w_events"
}
```

Oltre alle informazioni base inserite nella proprietà dell'oggetto: nome, partner e chain è possibile osservare che la proprietà rules contiene un'unica regola nella quale ci sono le dimensioni What : Carcassa di vitello piemontese e When: tra il 01-11- 2020 e il 30-11-2020.

In seguito all'inserimento di questa relazione, il grafico a corde è cambiato (Figura 5.36) e, a differenza del grafico mostrato in Figura 5.34, si nota come anche il "Supermercato Trinacria" invia delle informazioni. La relazione ha un flusso che comprende entrambi gli attori. Cliccando sul flusso, l'utente può accedere ai dettagli delle relazioni.

Capitolo 6

Conclusioni

Le conclusioni più significative riguardano il percorso di crescita personale, culturale e professionale che mi ha permesso di imparare attraverso lo studio teorico, la relazione con gli altri e l'applicazione nel lavoro. Un traguardo importante è la consapevolezza che lo studio teorico non è terminato, che lo scambio di conoscenze e competenze con gli altri è fondamentale e che nessun obiettivo è raggiungibile se non lo si affronta con passione, volontà e umiltà.

Le conclusioni relative al progetto sono la necessità di migliorare la piattaforma XTap che è già in grado di offrire servizi e strumenti di provata qualità e di facile usabilità che la rendono ben accolta dall'utenza e competitiva rispetto ai prodotti della stessa tipologia. Il lavoro da svolgere è ancora tanto sia rispetto alle verifiche, ai miglioramenti e ai possibili potenziamenti della parte del progetto già sviluppata, sia riguardo agli sviluppi futuri.

Nessuna impresa, oggi, lavora come soggetto indipendente, ma agisce all'interno di un sistema, che è la supply chain di appartenenza, formato da soggetti diversi i quali hanno la possibilità di condizionare, a più livelli e in modo diverso, l'operato degli altri. La piattaforma XTap offre la possibilità di gestire fin nello specifico le informazioni relative al percorso produttivo della propria azienda, l'acquisizione di informazioni esterne, il controllo dei propri dati e la loro trasmissione ai partners. XTap fornisce strumenti che semplificano la raccolta dei dati e la comunicazione tra imprese e indirizza alla condivisione e alla visione sia globale che specifica dei processi. La condivisione e il controllo assumono quindi un grande valore che potenziano notevolmente la crescita e la competitività dell'azienda.

6.1 Conoscenze Acquisite

Il presente lavoro di tesi mi ha consentito di affrontare l'ideazione e lo sviluppo di un progetto interessante e ambizioso, che mi ha permesso di partecipare alle dinamiche interattive del gruppo di lavoro e all'approccio con i clienti committenti del progetto. Tali circostanze mi hanno insegnato tanto e mi hanno offerto forti stimoli in tutto il percorso condotto.

La fase iniziale è stata caratterizzata dallo studio teorico, sia riguardo agli standard GS1 utilizzati che sono una componente fondamentale del progetto, sia per la tecnologia React usata per il successivo sviluppo dell'applicazione.

La metodologia *learn by doing*, che promuove l'imparare facendo che ho seguito nello sviluppo di questo progetto, mi ha portato fin dall'inizio a confrontarmi con la tecnologia più avanzata e a scoprirne le potenzialità e le criticità. L'approccio *learn by doing* si è rivelato positivo in quanto ogni risultato raggiunto ha incrementato in me l'interesse e la volontà di imparare. Un progetto articolato e complesso come XTap ha richiesto la gestione e la coordinazione di un team di lavoro formato da specialisti con competenze diverse e, a volte, le relazioni sono risultate complesse. XTap però è la dimostrazione oggettiva che l'agire in team porta a risultati eccellenti anche quando il tempo è limitato.

6.2 Lo stato attuale

L'applicazione web di XTap è al momento in fase di sviluppo e noi del team continuiamo a lavorare per migliorarne l'usabilità e le prestazioni, anche attraverso il frequente confronto con il cliente. La relazione con il cliente permette di individuare quali sono le aree critiche dell'applicazione e quali sono le aree di maggior effetto per investire al meglio ulteriori risorse nello sviluppo.

In questo testo ho illustrato lo stato attuale della piattaforma la quale rende già operativi strumenti che permettono di usufruire di diversi servizi:

- la possibilità di osservare e filtrare la totalità di eventi della filiera per prodotto (What), per luogo (Where), per data (When) e per tipologia di operazione (Why);
- la facoltà di ricercare le informazioni desiderate;
- l'opportunità di osservare la storia di un prodotto, gli eventi del suo ciclo di vita, e di identificare tutti i prodotti dai quali deriva e quelli da lui derivati per facilitare operazioni come il *recall*;

- la facoltà di accedere alla visione dall'alto della supply chain e dei suoi eventi con l'opportunità di filtrare la visualizzazione per prodotto fino ad osservare un singolo evento e i suoi dettagli;
- la possibilità di realizzare dei plugins custom per le esigenze della singola azienda integrabili direttamente all'interno dell'applicazione;
- l'opportunità di creare nuovi template di eventi partendo dalle strutture base dello standard EPCIS, le quali permettono di identificare in modo più dettagliato le diverse operazioni del processo produttivo;
- la possibilità di controllare e gestire in maniera semplice ed efficace la condivisione delle proprie informazioni verso gli altri attori della filiera.

6.3 Sviluppi futuri

La piattaforma XTap, allo stato attuale, registra un rimando positivo da parte dei clienti a cui è stata proposta, in quanto offre strumenti migliori che la distinguono dagli altri progetti concorrenti e i potenziali clienti chiedono sempre più frequentemente spazi dimostrativi. Questi risultati indicano che è giusto continuare a credere e a lavorare con il massimo impegno nel completamento del progetto e nei suoi eventuali miglioramenti con l'intento di far diventare XTap un effettivo prodotto da proporre sul mercato. Il principale obiettivo per il futuro è il completamento e il consolidamento della piattaforma attuale. I colloqui con i potenziali clienti e l'attenzione alle sempre nuove esigenze di mercato saranno gli indicatori per lo sviluppo di nuove funzionalità in modo da proporre maggiori e adeguati servizi.

In futuro sarà indispensabile porre l'attenzione sul miglioramento delle prestazioni per offrire al cliente una buona esperienza di usabilità e sull'aggiunta di nuove feature, ad esempio il filtro per luoghi. Successivamente sarà necessario definire e gestire molteplici tipologie di utenti all'interno dell'applicazione, con autorizzazioni specifiche che permetteranno l'accesso solo ad alcune parti dell'applicazione.

Allo stato attuale l'applicazione ha un'unica interfaccia di amministrazione all'interno della quale si svolgono tutte le funzioni e si osservano tutti i dati. Risulterà importante riuscire ad adattare l'applicazione in base alle esigenze del cliente soprattutto per ciò che concerne i plugins privati i quali possono essere considerati come un'estensione personalizzata della piattaforma.

Per quanto riguarda i programmi futuri a livello di sistema XTAP, oltre ai continui sviluppi e miglioramenti sul motore XTap, sono già in fase di studio due nuovi progetti: una mobile application per l'utilizzatore finale e un gioco interattivo (serious game).

Nella fase iniziale la mobile application sarebbe utile per mostrare le informazioni sulla tracciabilità a livello dettagliato sfruttando la scansione di un QR Code;

in seguito potrebbe diventare un vero e proprio strumento di comunicazione tra aziende e cliente finale.

Il gioco interattivo ha una finalità educativo/didattica in quanto potrebbe illustrare e far vivere anche ai meno esperti la struttura complessa di una supply chain e far comprendere i vantaggi che offrono la condivisione dei dati e la collaborazione. Il gioco, un simulatore interattivo dove l'utente gestisce un attore della filiera, dovrà riuscire ad essere un generatore di eventi. Tali eventi, inviati in modo adeguato alla piattaforma XTap, verranno utilizzati dai diversi strumenti realizzati per mostrare le potenzialità della piattaforma e i vantaggi che essa offre nella gestione di un'azienda.



Figura 6.1: Serious game XTap



Figura 6.2: Wireframe mobile app

Appendice A

Icone Business Steps



Accepting



Arriving



Assembling



Collecting



Commissioning



Consigning



Creating class instance



Cycle counting



Decommissioning



Departing



Destroying



Dispensing



Entering exit



Holding



Inspecting



Installing



Killing



Loading



Other



Packing

Icone Business Steps

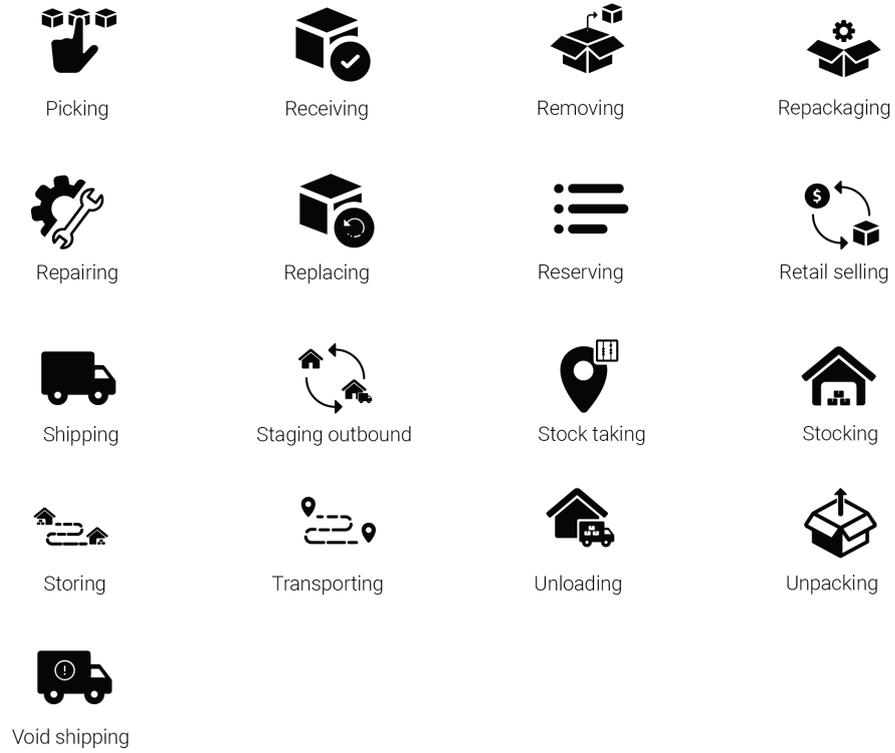


Figura A.1: Icone Business Steps

Bibliografia

- [1] GS1. *EPCIS and CBV Implementation Guideline. Using EPCIS and CBV standards to gain visibility of business processes*. URL: https://www.gs1.org/docs/epc/EPCIS_Guideline.pdf.
- [2] GS1. *Core Business Vocabulary Standard*. URL: <https://www.gs1.org/sites/default/files/docs/epc/CBV-Standard-1-2-r-2016-09-29.pdf>.
- [3] GS1. *GTIN - Global Trade Item Number*. URL: <https://gs1it.org/assistenza/standard-specifiche/gtin/>.
- [4] GS1. *Calcola la cifra di controllo*. URL: <https://gs1it.org/assistenza/calcolo-cifra-di-controllo/>.
- [5] GS1. *GLN - Global Location Number*. URL: <https://gs1it.org/assistenza/standard-specifiche/gln/>.
- [6] *Web Application*. URL: https://en.wikipedia.org/wiki/Web_application.
- [7] *single-page application vs multi-page application*. URL: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>.
- [8] *State of JavaScript*. URL: <https://2019.stateofjs.com/front-end-frameworks>.
- [9] *JavaScript*. URL: <https://en.wikipedia.org/wiki/JavaScript>.
- [10] *React concetti chiave*. URL: <https://it.reactjs.org/docs/getting-started.html>.
- [11] *Perchè gli Hooks*. URL: <https://it.reactjs.org/docs/hooks-intro.html#motivation>.
- [12] *Hooks*. URL: <https://it.reactjs.org/docs/hooks-intro.html>.
- [13] Daniel Bugl. *Learn React Hooks: Build and refactor modern React.js applications using Hooks*. 2019.
- [14] *Usabilità*. URL: <https://www.interaction-design.org/literature/topics/usability>.

- [15] *User Centered Design*. URL: <https://medium.com/redcatstudio/user-centered-design-method-28e3aafc8c8a>.
- [16] *User Experience*. URL: <https://medium.com/nyc-design/what-is-user-experience-d3f6833c2ffd>.
- [17] *Peter Morville, UX Honeycomb*. URL: <https://medium.com/mytake/the-ux-honeycomb-seven-essential-considerations-for-developers-acc372a398c>.
- [18] Jesse James Garrett. *The Elements of User Experience: User-centered Design for the Web*. 2002.
- [19] *Material Design*. URL: <https://medium.com/beginners-guide-to-mobile-web-development/a-guide-to-google-material-design-977315149ea5>.
- [20] *Material UI*. URL: <https://material-ui.com>.
- [21] *JWT Authentication*. URL: <https://medium.com/@sureshdsk/how-json-web-token-jwt-authentication-works-585c4f076033>.
- [22] *Axios*. URL: <https://medium.com/javascript-in-plain-english/what-is-axios-and-how-to-use-it-with-react-1470d19e1b83>.
- [23] *Axios Interceptors*. URL: <https://medium.com/datadriveninvestor/axios-instance-interceptors-682868f0de2d>.
- [24] *Routing in React*. URL: <https://medium.com/the-andela-way/understanding-the-fundamentals-of-routing-in-react-b29f806b157e>.
- [25] *Loadash*. URL: <https://lodash.com/docs/4.17.15>.
- [26] *OEE - Overall Equipment Effectiveness*. URL: https://medium.com/@info_82312/an-overview-of-overall-equipment-effectiveness-925108773a15.
- [27] *react-hook-form*. URL: https://medium.com/@angular_evan/yup-you-should-use-react-hook-form-9864d8bc80ef.