PENTATHLON

Sommario

1 – Infrastruttura, tecnologie, linguaggi e requisiti

Introduzione

L'obiettivo di questo documento è descrivere le modalità di realizzazione del progetto pilota per l'azienda Pentathlon.

Il progetto prevede la realizzazione di una applicazione web e un'applicazione mobile ibrida fruibile dai clienti per potere accedere ai servizi offerti da Pentathlon. Sono previste inoltre due interfacce di backoffice, una dedicata agli impiegati del supermercato addetti alla pubblicazione e l'aggiornamento della lista dei prodotti prenotabili dai clienti ed una realizzata per gli addetti alle consegne per monitorare gli ordini da consegnare ai clienti.

In questo documento verranno dapprima definite le funzionalità previste e i requisiti funzionali e non funzionali. Verranno descritte le tecnologie ed i linguaggi di programmazione adottati, motivandone la scelta. Verranno definite le user stories e presentato un diagramma degli use case.
In seguito, verrà analizzata l'architettura del sistema e forniti alcuni esempi di metodi REST. Verrà fornita una panoramica sulla struttura del database, includendo il diagramma E-R e verranno fornite delle query di esempio. Successivamente, verranno fornite le indicazioni su user experience, accessibilità e presentati i wireframe. Infine, verranno individuate le figure professionali necessarie e la suddivisione del lavoro con un digramma Gantt.

Funzionalità e requisiti

Attraverso la web app e/o l'applicazione mobile gli utenti potranno:

- Registrarsi
- Richiedere una tessera fedeltà
- Visualizzare la propria tessera fedeltà
- Inserire le informazioni relative alle metodologie di pagamento che intendono usare
- Inserire e/o modificare i propri dati personali
- Visualizzare i prodotti
- Filtrare i prodotti per categoria merceologica
- Noleggiare uno o più prodotti
- Vedere i dettagli della propria prenotazione
- Vedere i dettagli del noleggio (es. data di inizio e data di riconsegna)

Attraverso l'interfaccia web di backoffice per la gestione del magazzino il commesso potrà:

- Pubblicare la lista dei prodotti
- Aggiungere un prodotto alla lista dei prodotti
- Modificare i dettagli e/o la disponibilità di un prodotto presente nella lista
- Differenziare il costo del noleggio per un prodotto in base al periodo di richiesta (ad esempio alta o bassa stagione)
- Controllare la disponibilità di un prodotto
- Ottenere dei report sui noleggi

Attraverso l'interfaccia web per il monitoraggio degli ordini, gli addetti alle consegne potranno:

- Visualizzare gli ordini in arrivo
- Monitorare le stime dell'orario di arrivo del cliente per ritirare l'ordine

Requisiti funzionali

In base alle funzionalità definite, andiamo ora a definire i requisiti funzionali di questo progetto:

- L'utente deve poter registrarsi al servizio
- L'utente deve poter richiedere la tessera fedeltà
- L'utente deve poter visualizzare la propria tessera fedeltà
- L'utente deve poter inserire i propri dati personali
- L'utente deve poter modificare i propri dati personali
- L'utente deve poter impostare il proprio metodo di pagamento
- L'utente deve essere in grado di visualizzare i prodotti
- L'utente deve poter filtrare i prodotti per categoria merceologica
- L'utente deve poter stilare una lista della spesa
- L'utente deve poter noleggiare un prodotto
- L'utente deve poter visualizzare i dettagli della sua prenotazione
- L'utente deve poter consultare i dettagli del suo noleggio
- I commessi devono poter pubblicare la lista dei prodotti
- I commessi devono poter aggiungere un prodotto alla lista dei prodotti
- I commessi devono poter modificare i dettagli di un prodotto
- I commessi devono poter aggiornare la disponibilità di un prodotto
- I commessi devono poter verificare la disponibilità di un prodotto
- I commessi devono poter differenziare il costo di noleggio in base alla stagionalità
- I commessi devono poter ottenere dei report sui noleggi
- Gli addetti alle consegne devono poter visualizzare gli ordini in arrivo
- Gli addetti alle consegne devono poter monitorare le stime di orario di arrivo dei clienti

Requisiti non funzionali

Andiamo adesso a definire i requisiti non funzionali per questo progetto:

- Le tecnologie e i linguaggi utilizzati per realizzare le applicazioni mobile e web, che verranno descritti nella sezione apposita di questo documento.
- Le performance delle applicazioni.
- Gli accorgimenti sulla sicurezza anche per quanto concerne la sicurezza dei dati personali e dei pagamenti.
- La disponibilità e la resistenza del servizio sotto carico.
- Gli accorgimenti sul trattamento dei dati personali e sulla GDPR-compliance.
- Gli accorgimenti sull'usabilità delle applicazioni, che verranno approfonditi nell'apposita sezione.
- Gli strumenti di comunicazione e gestione del progetto utilizzati dal team di sviluppo.

Tecnologie adottate

Andiamo a definire ed analizzare le tecnologie che sono state decise di adottare per lo sviluppo di questo progetto.

- **Frontend –** Per il lato frontend dell'applicazione web usata dai clienti e delle interfacce web usate dai commessi di Pentathlon e dagli addetti alle consegne verranno utilizzati **HTML, CSS e JavaScript.** Verrà utilizzata la libreria di JavaScript, **React**, in modo da poter realizzare delle **SPA** (single page application) in modo da incrementare le prestazioni delle nostre web app e l'efficienza nell'utilizzo delle risorse. Verrà inoltre utilizzato il gestore di stato **Redux** per semplificare la gestione degli stati in React. Per migliorare gli aspetti di responsività dei siti e agevolare la creazione dell'interfaccia grafica verrà utilizzata la libreria CSS, **Boostrap**.
- **Backend –** Per il lato backend, sia delle applicazioni web che dell'applicazione mobile per i clienti, è stato deciso di utilizzare **Java** ed il framework **Springboot**. Tale scelta è motivata dall'affidabilità e solidità della piattaforma Java in ambito Enterprise, che garantisce un'ottima scalabilità, permettendo di ottenere ottime performance anche sotto elevato carico.
- **Mobile –** Per la realizzazione dell'app mobile è stato scelto di sviluppare un'applicazione ibrida-nativa utilizzando **React-native**. Questa scelta è stata realizzata per contenere i costi e i tempi di sviluppo, pur garantendo un risultato soddisfacente dal punto di vista delle performance. Sviluppare un'app ibrida ci permette di utilizzare lo stesso linguaggio di sviluppo sia per l'applicazione iOS che per quella Android. Sebbene in questo modo non è possibile sfruttare appieno le funzionalità più avanzate offerte da Swift e SwiftUI per iOS e dallo sviluppo nativo con Java/Kotlin su Android, per i requisiti definiti per la nostra applicazione tali funzionalità avanzate non sono necessarie. La scelta di React-native, rispetto ad altre alternative come Flutter, ricade sull'integrazione con l'ecosistema React, permettendoci di riutilizzare parte del codice React frontend nello sviluppo dell'app mobile.
- **Database –** Il DBMS scelto per questo progetto è **MySQL** e utilizzeremo quindi il suo dialetto di SQL per l'esecuzione di query.
- **Gestione dei pagamenti –** I pagamenti verranno processati esternamente con **Stripe,** per garantire la sicurezza dei pagamenti.
- **Gestione della sicurezza e integrità dei dati –** Per garantire la sicurezza del sistema e l'integrità dei dati verranno adottati i seguenti provvedimenti:
    - La connessione al sito e alle interfacce di backoffice avverrà mediante protocollo **HTTPS** (in questo modo i dati inseriti, come ad esempio password, sono criptati e non lasciati in chiaro).
    - Le chiamate API avverranno tramite il meccanismo di autenticazione **JWT** (JSON Web Token), in modo da garantire che solo gli utenti autenticati e ad autorizzati possano accedere ai dati richiesti.
    - Il login prevederà **l'autenticazione a due fattori**.
    - Le notifiche dell'applicazione mobile verranno gestite tramite i **servizi di gestione delle notifiche** forniti da Apple (APNs) e Google.
    - Le query sul database non verranno immesse direttamente, verranno utilizzate le funzionalità offerte da SpringBoot e JPA per effettuare le query.
    - La sicurezza dei pagamenti viene garantita da Stripe.

User Stories, Attori e Use Case

All'interno del nostro sistema possiamo individuare tre attori: i clienti, i commessi del negozio e gli addetti alle consegne.

Andiamo adesso a definire le user stories:

1. Come cliente voglio potermi registrare per poter accedere ai servizi offerti da Pentathlon.
2. Come cliente voglio poter visualizzare la mia tessera fedeltà così che il commesso possa scannerizzarla.
3. Come cliente voglio poter visualizzare, inserire e/o modificare i miei dati personali.
4. Come cliente voglio poter filtrare i prodotti per categoria così da poter trovare quello che mi serve.
5. Come cliente voglio poter realizzare una lista della spesa per tenere traccia dei prodotti che voglio noleggiare.
6. Come cliente voglio poter noleggiare un prodotto.
7. Come cliente voglio poter vedere i dettagli della prenotazione per sapere quando e dove ritirarlo.
8. Come cliente voglio poter vedere i dettagli del noleggio per sapere quando restituire il prodotto noleggiato.
9. Come commesso voglio poter aggiungere un prodotto alla lista dei prodotti.
10. Come commesso voglio poter aggiornare i dettagli e/o la disponibilità di un prodotto per tenere aggiornati i clienti.
11. Come commesso voglio poter consultare la disponibilità di un prodotto.
12. Come commesso voglio poter differenziare il prezzo di un prodotto in base al periodo del noleggio.
13. Come commesso voglio poter ottenere un report sui noleggi per monitorare l'andamento dei noleggi.
14. Come addetto alla consegna voglio poter vedere gli ordini in arrivo.
15. Come addetto alla consegna voglio poter monitorare la stima dell'orario di arrivo del cliente per potergli consegnare subito il suo ordine.

Diagramma Use Case

Date le user stories, ecco il diagramma degli Use Case.

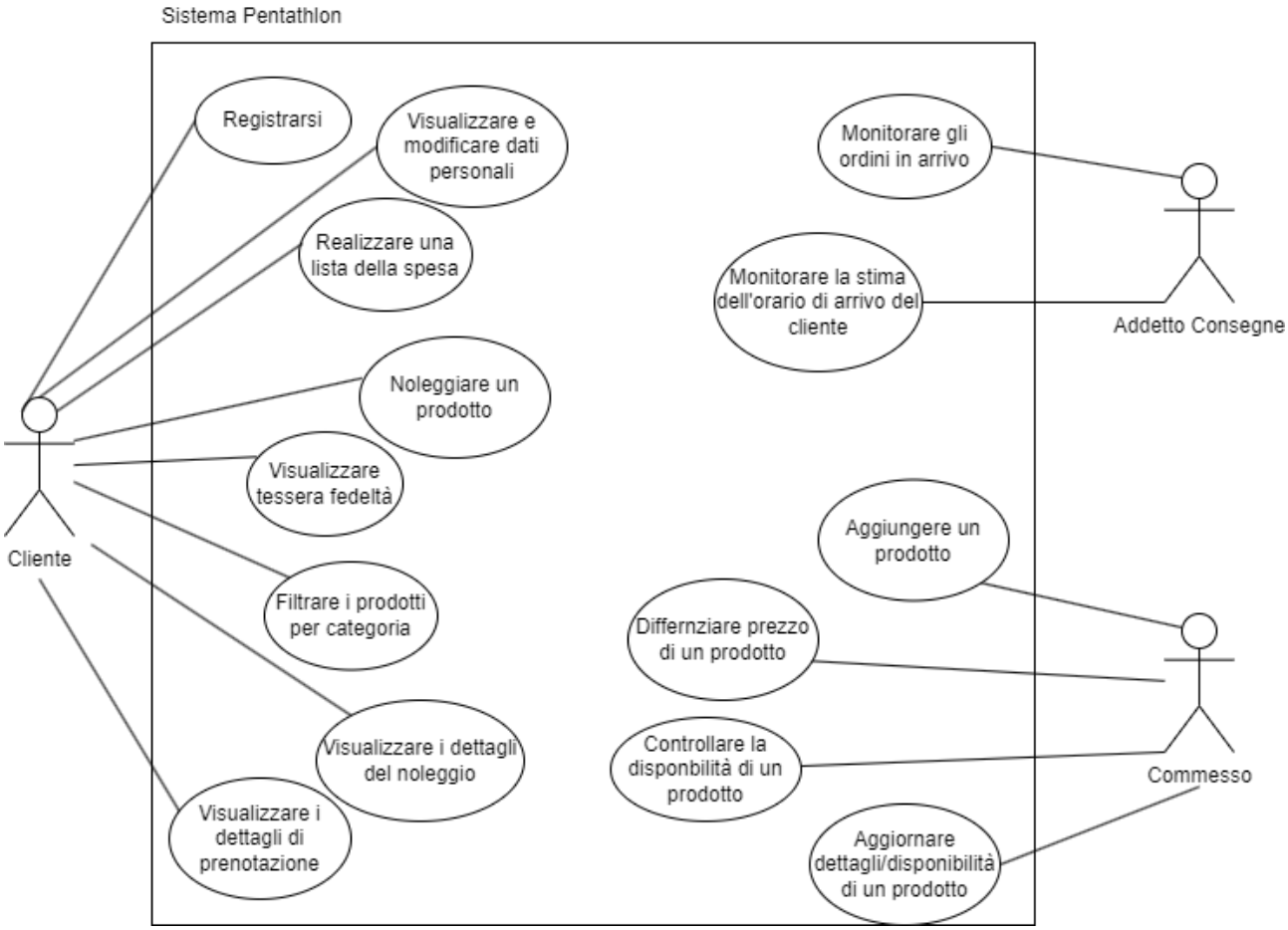
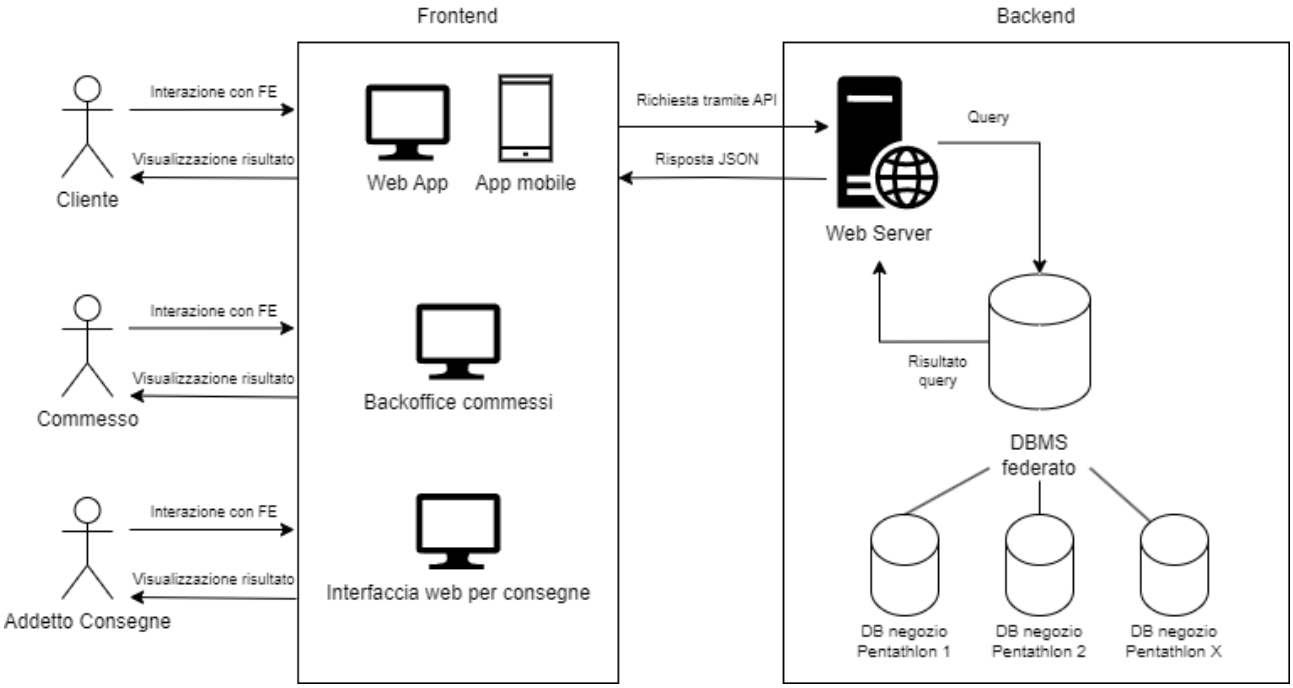


Diagramma di architettura del sistema

Andiamo ora a definire il diagramma di architettura del sistema.



Per semplificare la lettura del diagramma l'interazione tra frontend e backend è stata riportata solamente con le frecce in alto. Chiaramente quel tipo di interazione è valido anche per il frontend del backoffice commessi e per l'interfaccia web per le consegne.

Sistema delle notifiche mobile

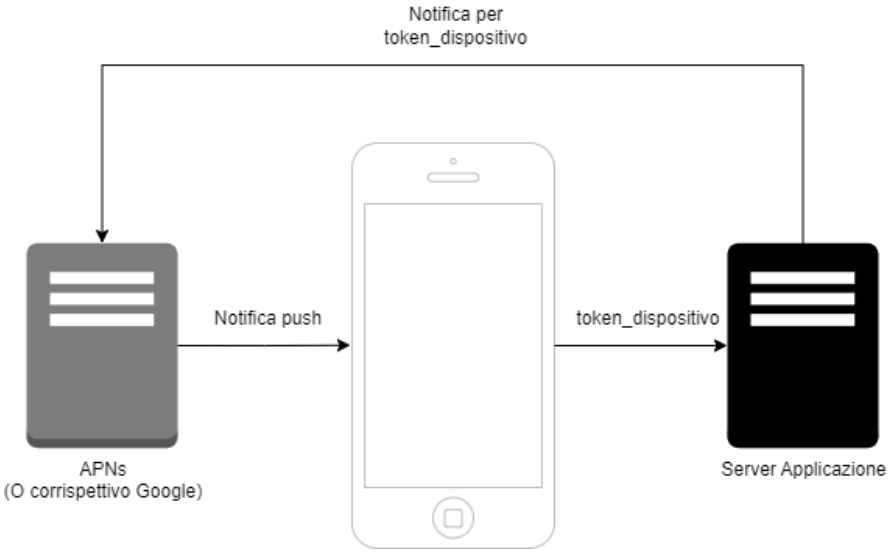


Tabella Metodi Rest

Andiamo adesso ad analizzare alcuni metodi REST previsti per questo progetto. Per ciascun metodo descriveremo l'endpoint, il metodo http, i parametri se necessari, un esempio di status code, output e JSON di esempio.

Il Base URL è: https://www.pentathlon.com/api/

Endpoint	Metodo	Parametri	Status Code	Output	Esempio
login	Post	Email, password (stringhe)	200 (Ok)	Struttura di tipo Cliente (id, nome, cognome, email)	{id: 100, nome: "Mario", cognome: "Rossi, email: "mario.rossi@esempio.it"}
login	Post	Email, password (stringhe)	400	Messaggio di errore	{message: "Errore! Email o password incorretti"}
products	GET	//	200	Lista di Prodotti (id, nome, categoria, prezzo)	[{id: 1, nome: "esempio", categoria: "Abbigliamento", prezzo: 10}, {id: 50, nome: "prova", categoria: "Accessori", prezzo: 2}]

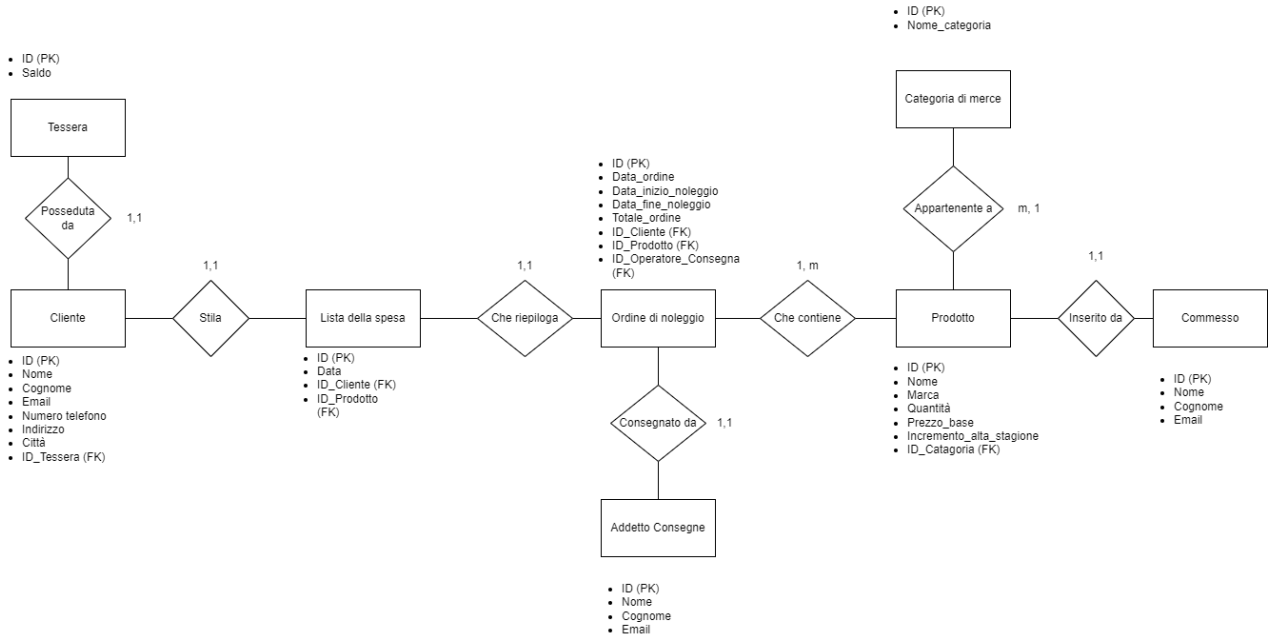
register	POST	Nome, cognome, email, telefono, password, indirizzo, città	200	Esito della registrazione con id utente	{id: 101, message: “Registrazione avvenuta con successo”}
register	POST	Nome, cognome, email, telefono, password, indirizzo, città	400	Messaggio di errore	{message: “Errore! Email già registrata”}
addProduct	POST	Nome, marca, quantità, prezzo base, incremento alta stagione, id categoria	400	Esito dell’inserimento con id prodotto	{id: 51, message: “Prodotto aggiunto al magazzino con successo”}

2 – Progettazione del database

Impostazione database

Come stabilito nella prima sezione, utilizzeremo un database di tipo relazionale usando il DBMS MySQL. Considerata la natura dell’applicazione è stato deciso di utilizzare un database di tipo federato (FDBS). Un FDBS viene visto dall’applicazione come un unico database ma contiene al suo interno la mappatura ai dati di database esterni. In questo modo possiamo avere un unico database che contiene i riferimenti ai database dei singoli negozi Pentathlon. In questo modo ciascun negozio Pentathlon può gestire i dati del proprio database, riguardo ad esempio gli articoli e la disponibilità di questi ultimi nel negozio. In questo modo è più facile garantire la segregazione dei dati.

Diagramma E-R



Analizziamo le entità del diagramma E-R.

- L'entità **Tessera** contiene le informazioni sulla tessera fedeltà di un cliente
  - ID (Primary Key)
  - Saldo
- L'entità **Cliente** descrive le informazioni del cliente
  - ID (Primary Key)
  - Nome
  - Cognome
  - Email
  - Numero di telefono
  - Indirizzo
  - Città
  - ID_Tessera (Foreign Key)
- L'entità **Lista della spesa** contiene i prodotti inseriti e viene utilizzata per effettuare un ordine di noleggio
  - ID (Primary Key)
  - Data
  - ID_Cliente (Foreign Key)
  - ID_Prodotto (Foreign Key)
- L'entità **Ordine di noleggio** contiene le informazioni sull'ordine di un cliente.
  - ID (Primary key)
  - Data_ordine
  - Data_inizio_noleggio
  - Data_fine_noleggio
  - ID_Cliente (Foreign Key)
  - ID_Prodotto (Foreign Key)
  - ID_Operatore_Consegna (Foreign Key)

- L'entità **Addetto Consegne** descrive le informazioni dell'addetto alla consegna
    - ID (Primary Key)
    - Nome
    - Cognome
    - Email
- L'entità **Prodotto** descrive le informazioni di un prodotto
    - ID (Primary Key)
    - Nome
    - Marca
    - Quantità
    - Prezzo_base
    - Incremento_alta_stagione
    - ID_Categoria (Foreign Key)
- L'entità **Categoria di merce** contiene le informazioni sulle categorie
    - ID (Primary key)
    - Nome_categoria
- L'entità **Commesso** descrive le informazioni del commesso
    - ID (Primary key)
    - Nome
    - Cognome
    - Email

Nel diagramma E-R vengono anche descritte le cardinalità delle relazioni tra le entità.

Query di esempio

Andiamo adesso a definire alcune query di esempio.

1) Prodotti noleggiati che devono essere restituiti entro una certa data (DATA_RICHIESTA).

SELECT ID.Prodotto as ID_Prod, Nome.Prodotto, ID.Cliente as ID_Cliente, Email.Cliente

FROM Prodotti

INNER JOIN Ordini_Noleggio ON Ordini_Noleggio.ID_Prodotto = Prodotto.ID_PROD

INNER JOIN Clienti ON Clienti.ID = Ordini_Noleggio.ID_Cliente

WHERE Ordini_Noleggio.Data_fine_noleggio = *DATA_RICHIESTA*

2) Saldo tessera cliente dato il numero di telefono (TELEFONO_RICHIESTO)

SELECT Saldo.Tessere, Nome.Clienti, Cognome.Clienti, Email.Clienti, Numero_telefono.Clienti

FROM Clienti

INNER JOIN Tessere ON Tessere.ID = Clienti.ID_Tessera

WHERE Numero_telefono.Clienti = *TELEFONO_RICHIESTO*

3) Ordini consegnati da uno specifico addetto alle consegne (ID_RICHIESTO) in un range di date specifico (RANGE_DATA_RICHIESTA_INIZIO e RANGE_DATA_RICHIESTA_FINE).

SELECT ID.Ordini, Data_inizio_noleggio.Ordini, Cognome.AddettiConsegne

FROM Ordini

INNER JOIN AddettiConsegne ON Ordini.ID_Operatore_Consegna = AddettiConsegne.ID

WHERE Data_inizio_noleggio.Ordini BETWEEN ‘RANGE_DATA_RICHIESTA_INIZIO’ AND ‘RANGE_DATA_RICHIESTA_FINE’ AND AddettiConsegne.ID = *ID_RICHIESTO*

3 – Mockup grafico, usabilità e accessibilità

Indicazioni su user experience e accessibilità

Il sito web e l'applicazione mobile verranno realizzati seguendo le linee guida di UX e accessibilità.

Questo significa che:

- **Tipografia**: verranno impiegati font leggibili e di dimensioni adeguate
- **Colori e contrasto**: la palette colori verrà studiata per garantire un contrasto che soddisfi almeno il livello AA secondo le linee guida WCAG.
- **Web:** verranno utilizzati i tag HTML opportuni per gli elementi del sito e implementate le best practice per l'accessibilità del sito (ad esempio inserire la descrizione delle immagini con l'attributo *alt*, in modo da favorire la fruizione del sito per coloro che utilizzano sistemi di lettura dello schermo)
- **Mobile:** verranno seguite le linee guida per la Human Interface fornite da Apple per la realizzazione dell'interfaccia dell'applicazione mobile.

Nel corso dello sviluppo, saranno previsti momenti di valutazione delle interfacce grafiche e della user experience attraverso valutazioni empiriche, come test di usabilità e questionari.

Wireframe

All'interno della cartella di consegna è possibile trovare il file pdf contenente i wireframe. Sono stati realizzati i wireframe per l'applicazione mobile e per le due interfacce web destinate ai dipendenti, rispettivamente l'interfaccia di backoffice per i commessi e l'interfaccia per il monitoraggio degli ordini per gli addetti alla consegna.

Il file *"wireframe.pdf"* è così strutturato:

1) **App –** Schermata di login/registrazione
2) **App –** Home
3) **App –** schermata di ricerca prodotti
4) **App –** Schermata visualizzazione tessera fedeltà
5) **App –** Schermata storico ordini
6) **App –** Impostazioni utente
7) **App –** Schermata di Dettaglio prodotto
8) **App –** Schermata lista spesa da cui è possibile inviare l'ordine
9) **Interfaccia addetti consegne –** Schermata ordini in arrivo
10) **Interfaccia addetti consegne –** Dettaglio ordine e arrivo cliente
11) **Interfaccia backoffice commessi –** Ricerca prodotti in magazzino
12) **Interfaccia backoffice commessi –** Inserimento nuovo prodotto
13) **Interfaccia backoffice commessi –** Schermata per scaricare report noleggi

Il numero della lista corrisponde al numero di pagina del file PDF.

4 – Suddivisione e gestione del Lavoro

Ruoli necessari

Per questo progetto sono stati individuati i seguenti ruoli, necessari per il suo completamento:

- Un project manager.
- Tre sviluppatori frontend con esperienza in React, per la realizzazione del sito web e delle interfacce di backoffice per i commessi e per il monitoraggio ordini per gli addetti alle consegne.
- Uno UI/UX Designer, per il design delle interfacce grafiche e dell'esperienza utente.
- Quattro sviluppatori backend con esperienza in Springboot, che si occuperanno di realizzare il comparto backend di tutti i servizi.
- Due sviluppatori mobile con esperienza in React Native, che si occuperanno di sviluppare l'applicazione mobile.
- Un Database Administrator, che si occuperà del design e della struttura del database e delle tabelle.

Saranno inoltre previste delle consulenze esterne in ambito legale per garantire che le applicazioni ed i servizi offerti saranno GDPR-compliant e che vengano rispettate tutte le norme sulla protezione dei dati dei prodotti rispetto alla concorrenza.

Strumenti per la collaborazione

Per tenere traccia dei progressi del progetto e per comunicare all'interno del team di sviluppo verranno utilizzati:

- **Jira**, per organizzare il lavoro e tenere traccia dell'andamento dei task e user stories assegnate agli sviluppatori.
- **Slack,** per la comunicazione all'interno del team.
- **Gmail,** verrà utilizzato per lo scambio mail tra membri del gruppo e il committente e persone esterne (ad esempio i consulenti legali).

Diagramma Gantt e stime sui tempi

Nella cartella di consegna è presente un file Excel contenente il diagramma Gantt.

La consegna di un MVP è prevista entro **28 settimane** a partire dalla consegna del seguente documento di specifiche.