**Progetto:**

**Paninaro Next**

**Titolo del documento:**

**Sviluppo Applicazione**

INDICE

Scopo del documento

1. User Flows

2. Application Implementation and Documentation

2.1. Project Structure

2.2. Project Dependencies

2.3. Project Data or DB

2.4. Project APIs

2.4.1. Resources Extraction from the Class Diagram

2.4.2. Resources Models

2.5. Sviluppo API

2.5.1. Elenco Ingredienti

2.5.2. Creazione di un Ingrediente

2.5.3. Cancellazione di un Ingrediente

3. API documentation

4. FrontEnd Implementation

5. GitHub Repository and Deployment Info

6. Testing

Scopo del documento

Il presente documento riporta tutte le informazioni necessarie per lo sviluppo di una parte dell’applicazione Paninaro Next.

Partendo dalla descrizione degli user flow legate al ruolo del responsabile amministrativo dell’applicazione, il documento prosegue con la presentazione delle API necessarie (tramite l’API Model e il Modello delle risorse) per poter visualizzare, inserire e

modificare sia i clienti che gli ingredienti nel database.

Per ogni API realizzata, oltre ad una descrizione delle funzionalità fornite, il documento

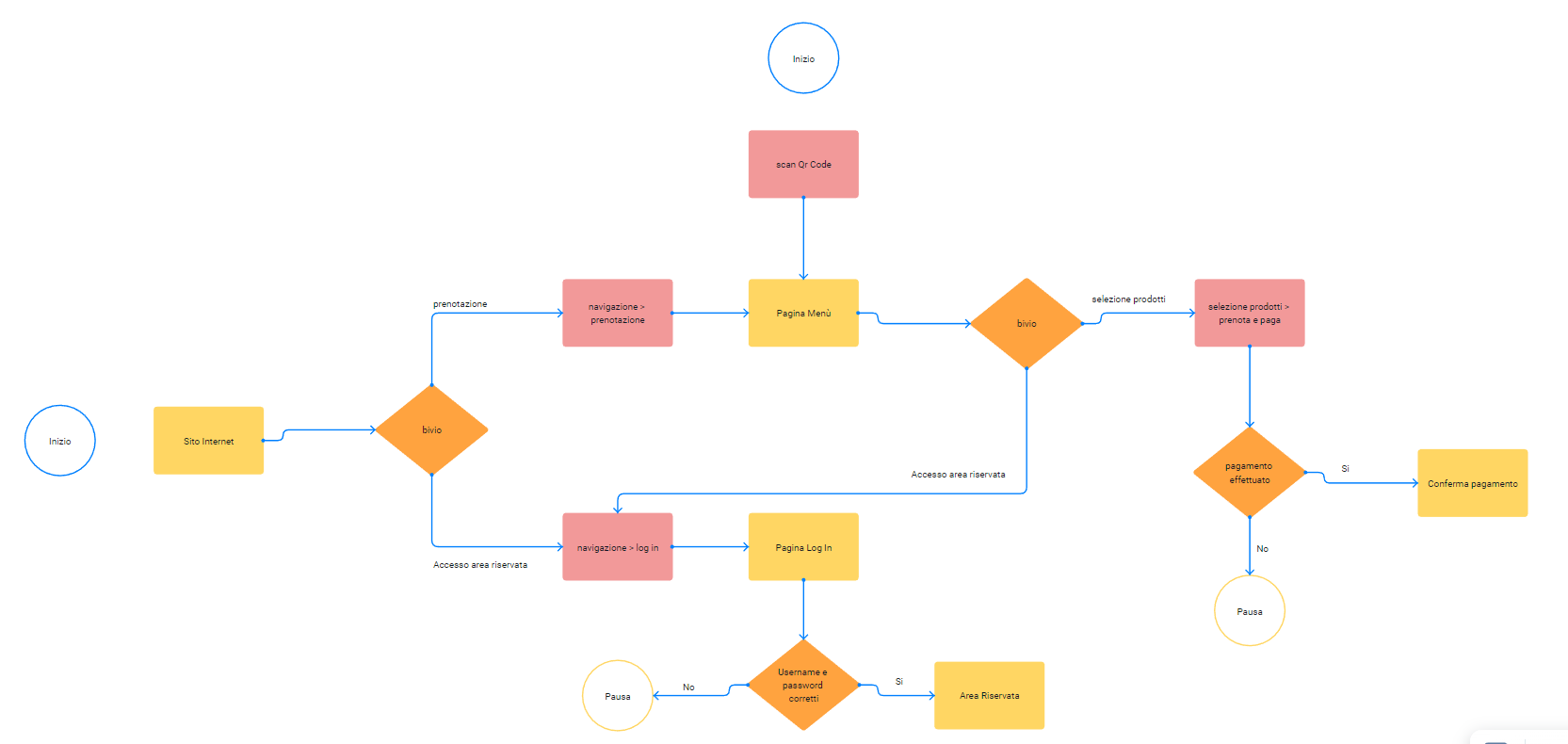
presenta la sua documentazione e i test effettuati. Infine una sezione e’ dedicata alle

informazioni del Git Repository e il deployment dell’applicazione stessa.

1. User Flows

In questa sezione del documento di sviluppo riportiamo gli “user flows”, descritti dalla figura sottostante, con una legenda.





Si nota come un utente cliente possa accedere all’applicazione sia cercandola con un motore di ricerca, sia scannerizzando un apposito QR code. D’altro canto l’amministratore può accedere alla piattaforma tramite le sue credenziali per entrare nell’area riservata.

2. Application Implementation and Documentation

Nelle sezioni precedenti abbiamo identificato le varie features che devono essere implementate per la nostra applicazione con un’idea di come il nostro utente finale puo’ utilizzarle nel suo flusso applicativo. L’applicazione è stata sviluppata utilizzando NodeJS e VueJS. Per la gestione dei dati abbiamo utilizzato MongoDB.

2.1. Project Structure

La struttura del progetto è presentata in figura:

…immagine struttura progetto…

2.2. Project Dependencies

I seguenti moduli Node sono stati utilizzati e aggiunti al file Package.Json

• Express:

• Express-file upload

• MongoDB

• Cors

• Body-parser

2.3. Project Data or DB

Per la gestione dei dati utili all’applicazione abbiamo definito le strutture dati

come illustrato in figura.

Immagine che contiene tavolo

Descrizione generata automaticamente

Per rappresentare i vari tipi di dati:

Per la tabella Clienti

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Per la tabella Ingredienti

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

2.4. Project APIs

2.4.1. Resources Extraction from the Class Diagram

Immagine che contiene testo, diagramma

Descrizione generata automaticamente

2.4.2. Resources Models

Ecco i Resources models:

Immagine che contiene testo, diagramma, linea, Piano

Descrizione generata automaticamente

2.5. Sviluppo API

Ecco le varie API sviluppate:

2.5.1 Elenco Ingredienti/Clienti

Mostra un elenco di tutti gli ingredienti/clienti presenti nel database

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

2.5.1 Trova Ingrediente/Cliente

Trova un ingrediente/cliente in base al nome/nr\_conto

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

2.5.1 Modifica Ingrediente/Cliente

Modifica il prezzo di un ingrediente dato il nome, e il conto di un cliente dato il nr\_conto

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

2.5.2 Creazione di un Ingrediente/Cliente

Crea un ingrediente/cliente

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

2.5.3 Cancellazione di un Ingrediente/Cliente

Cancella un ingrediente/cliente dato il nome/nr\_conto

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

3. API documentation

Le API Locali fornite dall’applicazione Paninaro Next e descritte nella sezione precedente sono state documentate utilizzando il modulo NodeJS chiamato Swagger UI Express. In questo modo la documentazione relativa alle API è direttamente disponibile a chiunque veda il codice sorgente.

Per poter generare l’endpoint dedicato alla presentazione delle API abbiamo utilizzato Swagger UI in quanto crea una pagina web dalle definizioni delle specifiche OpenAPI.

In particolare, di seguito mostriamo la pagina web relativa alla documentazione che presenta le 4 API (GET, POST, DELETE e PATCH) per la gestione dei dati della nostra applicazione.

La GET viene utilizzata per visualizzare i dati in una pagina HTML. La POST per inserire un nuovo dato nel nostro sistema. La DELETE per cancellare un dato dal nostro sistema. E la PATCH per modificare dati.

L’endpoint da invocare per raggiungere la seguente documentazione e’:

**http://localhost:3000/api-docs**

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente

4. FrontEnd Implementation

Il FrontEnd fornisce le funzionalità di visualizzazione, inserimento e modifica degli ingredienti. Un pulsante serve per mostrare gli ingredienti, mentre il form per modificare il prezzo e aggiungere gli ingredienti, usando gli appositi bottoni.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

5. GitHub Repository and Deployment Info

<< Descrizione di come e’ strutturato il Repo Git >>

<< Info sul deployment e sul link da utilizzare per eseguire l’applicazione su Heroku

6. Testing

Ecco i vari casi di test fatti:

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente