Progetto:

**iSpesa**

Titolo del documento:

**Documento di architettura**

Informazioni Documento

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Documento | *iSpesa\_Sviluppo\_Applicazione\_D4* | Numero  Documento | D4 |
| Descrizione | Il documento parla dello sviluppo dell’applicazione | | |

Indice

[Scopo del documento 3](#_Toc153401410)

[1. Diagramma delle classi 4](#_Toc153401411)

[1.1 Autenticazione e Registrazione 4](#_Toc153401412)

[1.2 Pagina Utente 6](#_Toc153401413)

[1.3 Visualizzazione dei Prodotti 7](#_Toc153401414)

[1.3 Visualizzazione Negozi 8](#_Toc153401415)

[1.4 Visualizzazione Volantini 9](#_Toc153401416)

[1.5 Visualizzazione Sconti 10](#_Toc153401417)

[1.7 Aggiornamento Sito 10](#_Toc153401418)

[1.8 Recensioni 11](#_Toc153401419)

[1.9 Moderazione Utenti 12](#_Toc153401420)

[1.10 Interfaccia Gmail 12](#_Toc153401421)

[1.11 Classi di supporto 13](#_Toc153401422)

[1.12 Classe di Lettura/ScritturaDBUtenti 13](#_Toc153401423)

[1.13 Classe di Lettura/ScritturaDBDati 14](#_Toc153401424)

[1.14 Diagramma delle classi completo 15](#_Toc153401425)

# Scopo del documento

Il seguente documento riporta tutte le informazioni necessarie per descrivere lo sviluppo di una parte, abbastanza completa, dell’applicazione web iSpesa.

Nel primo capitolo viene riportato lo user flow, ovvero una descrizione tramite diagramma di tutte le azioni che si possono eseguire sulla parte implementata di iSpesa, descrivendo le varie richieste effettuabili a front-end in ogni pagina e le varie risposte possibili.

Successivamente rappresentiamo una struttura del codice realizzato, descrivendo le dipendenze installate, i modelli realizzati e le API implementate. Una attenta descrizione delle API implementate viene fatta con il diagramma delle risorse e il diagramma di estrazione delle risorse, in cui si individuano le risorse estratte a partire dal diagramma delle classi del documento D3.

Nel capitolo quattro si spiega ciò che si è fatto con Swagger per la documentazione delle API.

Successivamente viene fornita una breve descrizione per le pagine implementate e una descrizione del repository di GitHub con le istruzioni per effettuare il deployment.

Per finire mostriamo i vari casi di test realizzati per verificare il corretto funzionamento delle API.

# 1. User-Flow

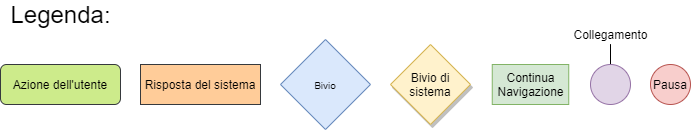
Riportiamo in seguito lo user-flow dell’applicazione, il quale descrive ciò che è possibile fare nell’implementazione descritta nel dettaglio nel seguente documento.

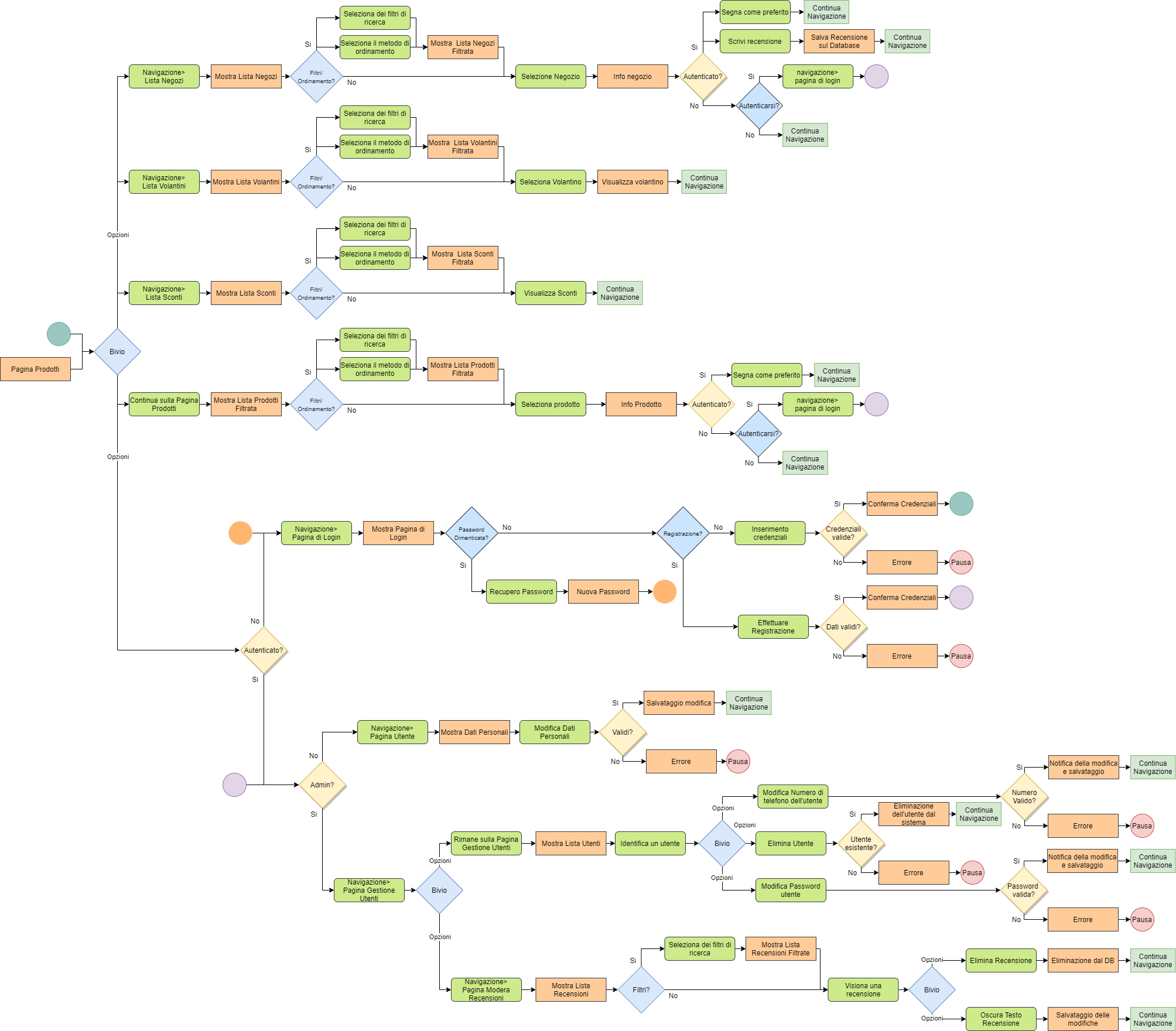
È stata riportata anche una didascalia dei vari componenti utilizzati nello user-flow.

Si può notare la presenza di due differenti tipi di bivi, quello semplice, che rappresenta una scelta dell’utente tra molteplici azioni fattibili in un determinato momento, e il bivio di sistema, che rappresenta differenti stati dell’applicazione (per esempio essere autenticato o meno) che fanno sì che si possano o non possano fare determinate azioni.

È presente, inoltre, un componente collegamento che ha lo scopo di collegare due zone molto distanti del diagramma. Infine vi è anche il componente “Continua Navigazione”, che indica un punto in cui l’utente può ritornare allo stato iniziale o andare in una qualunque delle pagine apribili nello stadio iniziale del sito.

Figura 1. User-flow Diagram





# 2. Implementazione dell’Applicazione e Documentazione

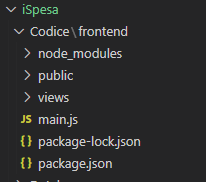
L’applicazione iSpesa è stata sviluppata utilizzando NodeJS per la parte di frontend. A livello di back-end per la memorizzazione dei vari dati abbiamo utilizzato MySQL.

Come si può notare dallo user-flow abbiamo sviluppato ogni aspetto dell’applicazione, tranne le parti relative ad Auth0. Purtroppo l’API esterna richiedeva un pagamento per utilizzare la funzione di 2FA quindi abbiamo implementato la parte di login e registrazione soltanto sul DB locale dell’applicazione. Abbiamo anche implementato una simulazione dell’acquisizione di dati da parte di un negozio.

Infine abbiamo anche utilizzato le API del sistema esterno GMail per gestire l’invio di email

agli utenti.

## 2.1 Struttura del progetto

La seguente immagine riporta la struttura principale dell’applicazione. Come si può notare la cartella principale è la cartella Codice\_iSpesa ed è così costituita:

* File package.json che rappresenta il file di configurazione generale del progetto.
* File swagger.json, dove sono descritte approfonditamente e con opportuni esempi le API implementate nel sistema. In esso è definita l’intera documentazione delle API da noi sviluppate.
* File main.js, che rappresenta il file principale del sistema, ovvero il file da eseguire per attivare la connessione verso il database MySQL e attivare il server all’indirizzo localhost:8080
* Cartella /node\_modules, dove sono scaricate le dipendenze utilizzate nel sistema
* Cartella /public, che contiene tutti quei file considerati “pubblici” (per esempio alcune immagini)
* Cartella /views, ovvero la cartella che contiene tutta la business logic del sistema dal lato front-end.

Nella cartella /public troviamo:

* La cartella /css, che contiene eventuali file .css utili per il front-end
* La cartella /img, che contiene tutte le immagini di dominio pubblico utili al sistema. Per esempio in essa sono contenute loghi, immagini del sito ed altro.