My Marketplace

Ingegneria del software - project work

Domenico Gaeni - 1065107
Fabio Palazzi - 1066365
Paolo Mazzoleni - 1057949

🎯 Obiettivo

Sviluppare di un'applicazione web implementando le varie metodologie viste a lezione o approfondite singolarmente.

- ➔ Estrapolazione requisiti e specifica dei requisiti;
- ➔ Modellazione del problema;
- ➔ Sviluppo e stesura documentazione (***GitHub***)
  - ◆ Front-end
  - ◆ Back-end
- ➔ Implementazione di test;
- ➔ Deploy dell'applicazione (***Heroku***)

 Testo

Sviluppo di un’applicazione web responsive in grado di implementare un servizio di compra-vendita di libri. Tra le funzioni più importanti ci sono:

- ➔ Autenticazione
- ➔ Registrazione
- ➔ Acquistare Libro
- ➔ Vendere Libro
- ➔ Recensire Libro
- ➔ Storico Libri (comprati e venduti)

♻️ Software Life Cycle

Il processo di sviluppo seguito è stato **AGILE**

- ➔ **Scrum** con un backlog (*issue su Github*)
- ➔ **Sprint** di 1 settimana
- ➔ **Meeting** ogni settimana per definire lavoro successivo
- ➔ Tutti i membri del team sullo **stesso piano**

 Configuration Management

- Utilizzo delle **issue** su Github
  - *To do*
  - *Progress*
  - *Pull Request*
  - *Testing*
  - *Done*
- Creazione di **branch** per ogni lavoro
- Creazione di **pull request**
- Utilizzo board **Kanban** per gestire lo stato delle issue

👷 People Management

- ➔ Modello **SWAT**
- ➔ **Adhocrazia**
- ➔ Suddivisione dei ruoli nel seguente ordine:

	Project Manager	Progettista Software	Progettista Database	Front-end	Back-end	Testing
Domenico	V	V	V		V	V
Fabio		V		V		V
Paolo	V		V	V	V	V

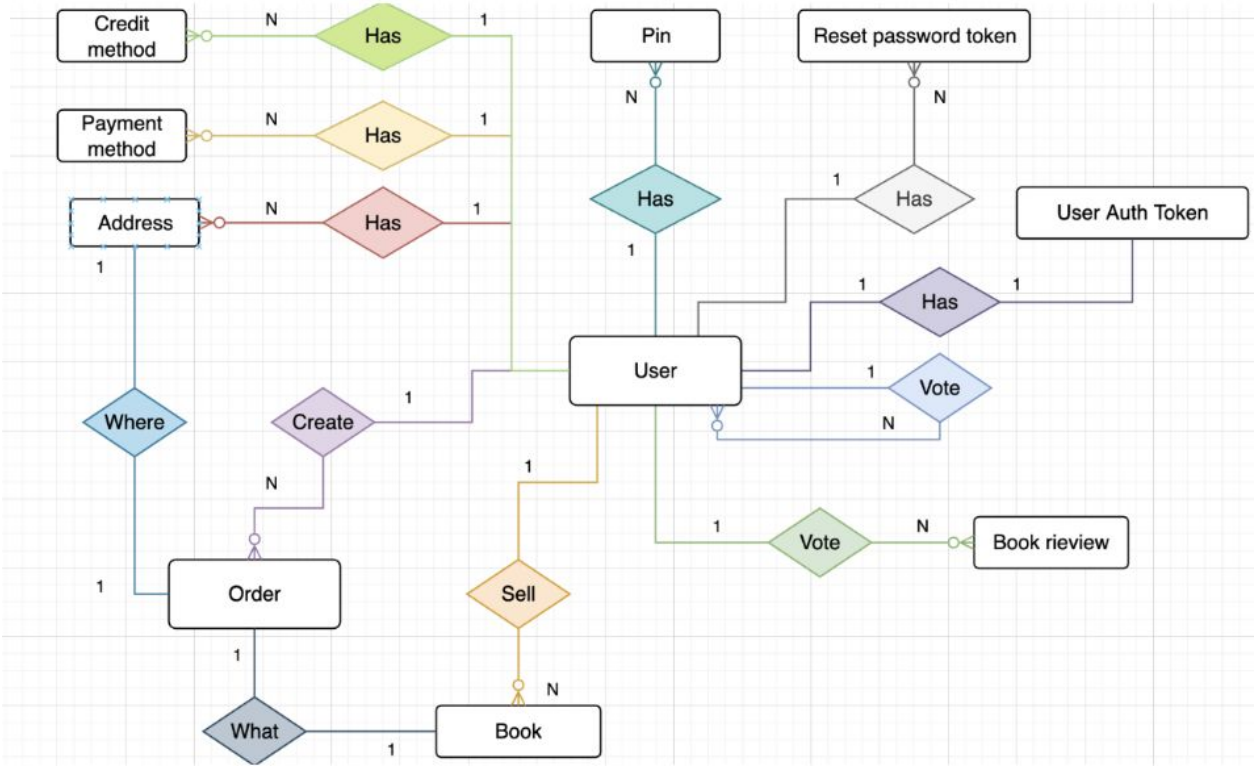
💯 Software Quality

- ➔ Assunzioni di **qualità**
  - ◆ Affidabilità
  - ◆ Efficienza
  - ◆ Correttezza
  - ◆ Usabilità
- ➔ Parametri **revisione** software
  - ◆ Testabilità
  - ◆ Flessibilità
  - ◆ Manutenibilità

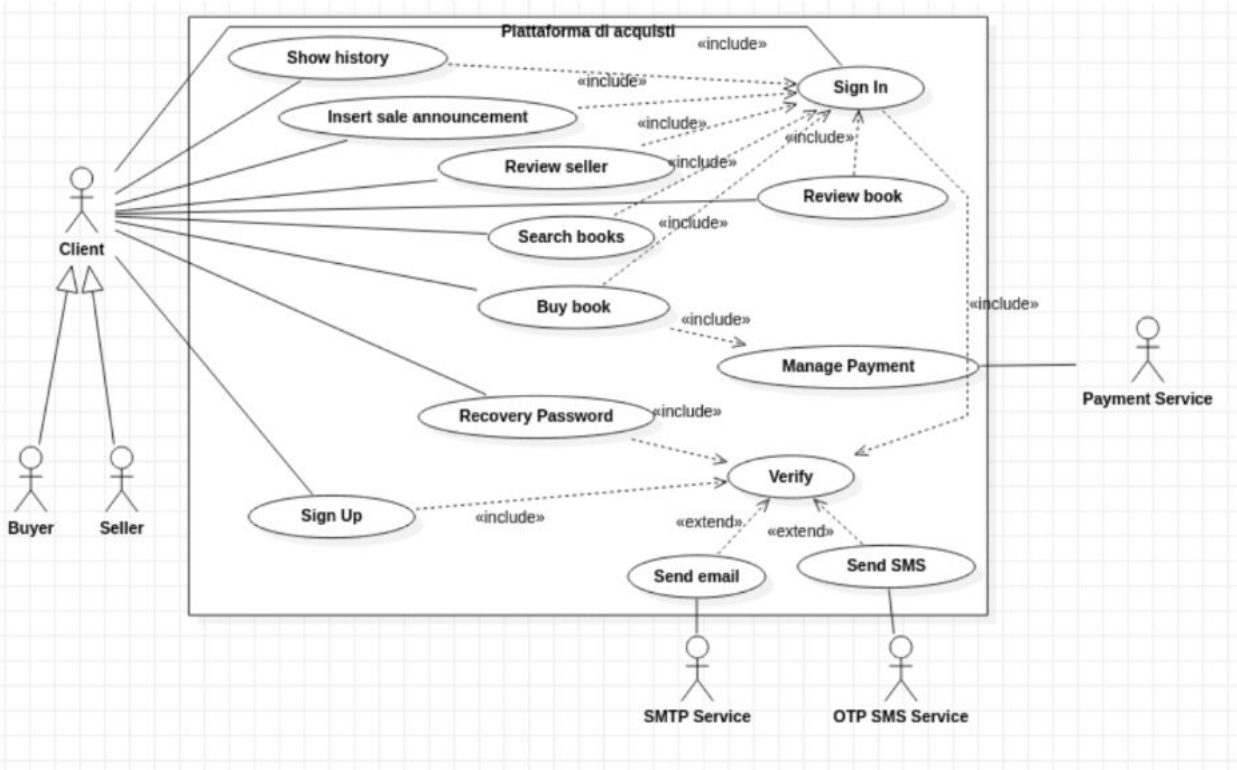
✏️ Requirements Engineering

- ➔ Classificazione dei requisiti secondo il modello **MoSCoW**
- ➔ In particolare:
  - ◆ ***Funzionali***: Autenticazione tramite mail e password
  - ◆ ***Non Funzionali***: la password deve essere composta da almeno 12 caratteri, contenente almeno un numero e un simbolo e memorizzata tramite cifratura hash a 256 bit

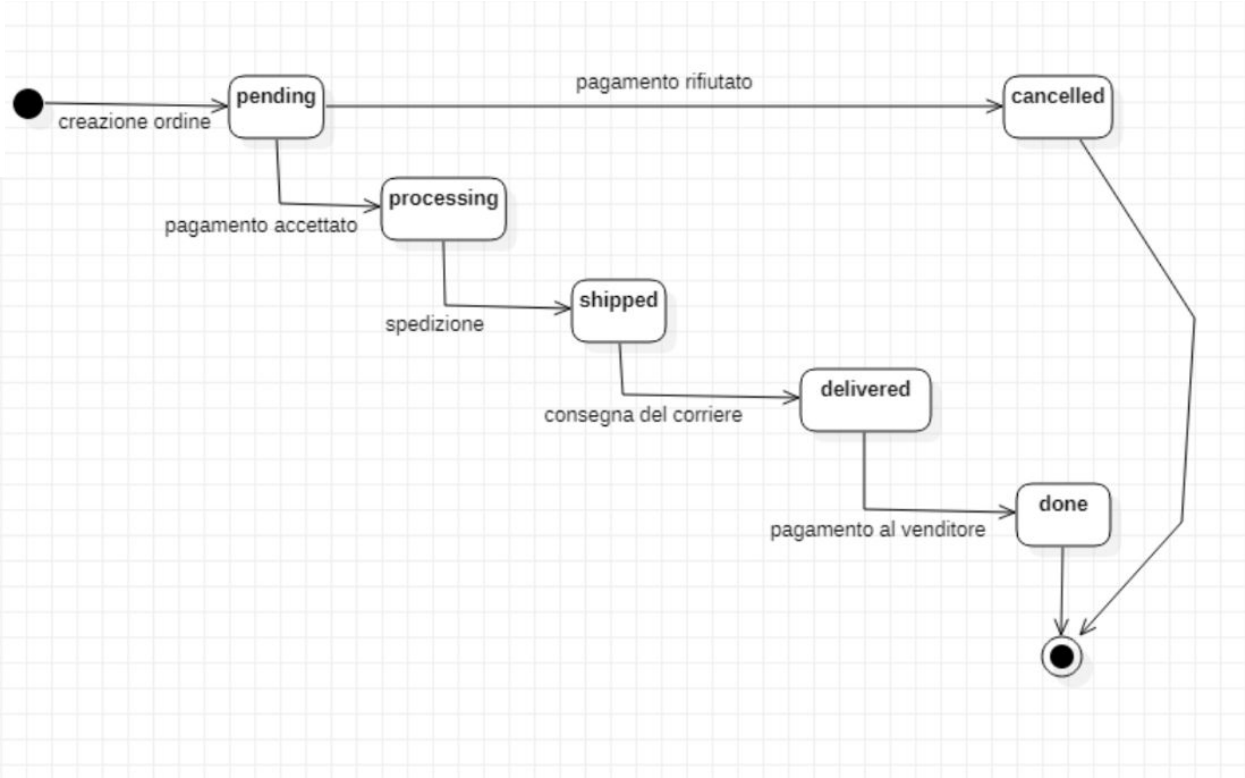
Schema ER


Credit method
Has
Payment method
Has
Address
Has
Pin
Reset password token
Has
Has
User Auth Token
Has
User
Vote
Where
Create
Vote
Book rieview
Order
Sell
What
Book
N
1

👦 Use Case Diagram



⚡ Macchina a Stati



🏛️ Software architecture

L'architettura della nostra applicazione è client-server basata sul **pattern MVC**.

Possiamo suddividerla in due componenti principali:

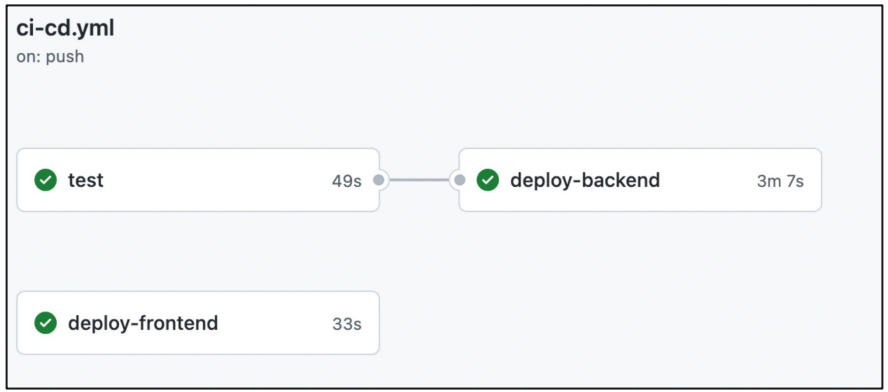
- **Front-end**: View
- **Back-end**: Controller-Model

Architettura a servizi: il front-end invia richieste http/**REST** al backend il quale ritornerà dei risultati.

- Minimizziamo accoppiamento e massimizziamo coesione
- Maggior Manutenibilità del codice
- Maggior Flessibilità

🧪 Software testing e manutenibilità

- ➔ Test statici con **Psalm**
- ➔ Test di unità con **PHPUnit**
  - ◆ Generazione di dati fake con Faker (pattern ***Factory***)
- ➔ **CI-CD**
  - ◆ **Deploy del frontend**
  - ◆ Esecuzione dei **test** sul backend
    - Se test ok => **deploy backend**



🌝 Front-end

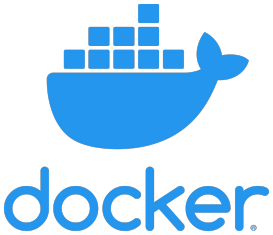
A livello di front-end l'applicazione è stata sviluppata utilizzando:

- ➔ **HTML** 5 / CSS
- ➔ Libreria **Bootstrap** 4
- ➔ Framework **jQuery** 3.6

Back-end

Il back-end è stato sviluppato utilizzando:

- ➔ **PHP** 8.0
- ➔ Framework **Lumen** (*based on Laravel*)
- ➔ **Docker**
- ➔ Pattern **MVC**
- ➔ **PHPUnit**
- ➔ **PostgreSQL**



🚀 Cosa abbiamo realizzato?

- Front-end:
  - https://front-end-ingegneria-software.herokuapp.com
- Back-end:
  - https://ingegneria-software.herokuapp.com/
- Github:
  - https://github.com/domenicogaeni/progetto-ingegneria-software