Progetto di ingegneria del Software

Università degli studi di Milano Bicocca



Oleg Stoianov 829519

Lorenzo Nosotti 830011



Sommario

2.Tecnologie utilizzate	4
3.Scelte di progettazione	6
4.Descrizione del sistema	7
5.Analisi dei requisiti	7
6.Diagrammi	14
7.Strumenti per l'analisi del software	21
8.Strumenti per la comunicazione	23
9.Conclusioni	24
10.Tutorial installazione App UnimiBike	24

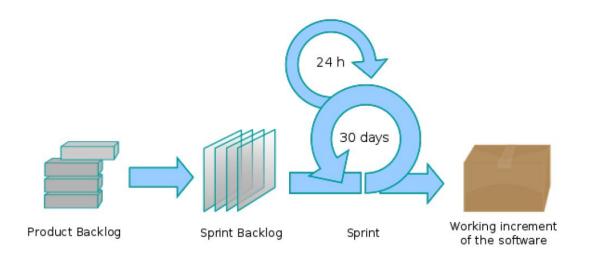
1.Pianificazione

1.1 Suddivisione dei ruoli

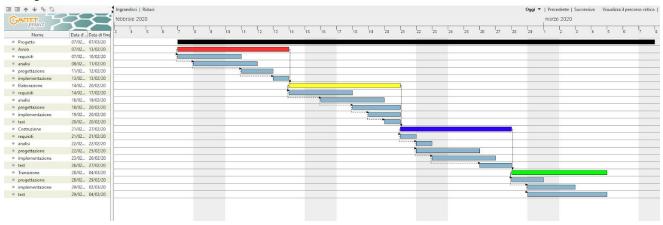
I ruoli sono stati suddivisi in maniera tale da permettere a ciascuna risorsa di potersi concentrare principalmente su una parte del sistema. La parte relativa al Frontend è stata assegnata a Lorenzo Nosotti 830011. La parte relativa al Backend è stata assegnata a Oleg Stoianov 829519.

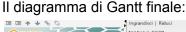
1.2 Piano di lavoro

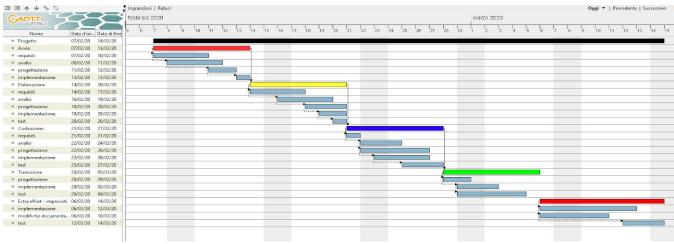
Per il piano di lavoro si è scelto di utilizzare un approccio al lavoro **agile** cercando di impostare i vari step progettuali rispettando il framework **Scrum**, studiato durante il corso di Analisi e progettazione del Software:



Il diagramma di Gantt iniziale:







2.Tecnologie utilizzate

2.1 Backend

Per il back end è stato scelto Spring, che è un framework molto utilizzato e supportato per le moderne applicazioni basate su Java. Il grosso del lavoro di Spring è l'aiuto che da agli sviluppatori cercando di semplificare l'unione ed il collegamento delle parti dell'applicazione, permettendo così ad essi di concentrarsi unicamente sulla logica di business. È basato su concetti chiave come l'inversion of control, la dependency injection ed il supporto alla persistenza dei dati.



Inversion of control: L'inversione di controllo è un design principle secondo cui un componente a livello applicativo riceve il controllo sull'esecuzione da un componente appartenente ad una libreria generica. Lo scopo di tale pattern è

di rendere le parti dell'architettura software quanto più indipendenti possibili, in modo da poter effettuare modifiche sulle singole componenti senza che si riflettano sul resto del sistema.

Dependency injection: L'iniezione delle dipendenze è un design pattern che permette di applicare l'inversion of control. È una tecnica in cui un oggetto acquisisce le dipendenze necessarie in fase di creazione, quando gli viene iniettato un oggetto, il quale viene poi utilizzato come servizio dall'oggetto principale.

Persistenza dati: per la persistenza dei dati sono stati utilizzate le librerie JPA, il fulcro di queste librerie è CrudRepositoy essa è un'interfaccia delegata alla comunicazione con la sorgente dati che implementa le principali operazioni CRUD.Il Database utilizzato per la memorizzazione dei dati è quello embedded **Apache Derby**.

Per la parte dell'autenticazione è stato utilizzato **Spring Security** che è un framework del progetto Spring che consente di gestire in modo semplice e trasparente l'autenticazione (ovvero chi sei) e la profilazione (ovvero cosa sei autorizzato a fare) degli utenti che accedono ad una applicazione web. Esso è stato implementato con l'utilizzo della tecnologia **JWT** (JSON Web Token) è uno standard open (RFC 7519) che definisce uno schema in formato JSON per lo scambio di informazioni tra vari servizi.

2.2 Frontend

Per il front end è stato scelto di utilizzare Angular 8, Bootstrap 4, JQuery e alcune librerie Javascript di supporto, quali Canvas JS e Stripe Checkout JS.

Bootstrap e JQuery hanno permesso di migliorare la parte grafica della nostra web application, migliorando componenti HTML come barra di navigazione, tabelle, card, form e aggiungendo semplici animazioni. Le librerie di Canvas JS le abbiamo sfruttate per l'implementazione dei grafici per le statistiche, mentre Stripe Checkout JS ha permesso di rendere più realistico il metodo di pagamento.



Ma la scelta più importante per quel che riguarda il front end è stata di utilizzare Angular 8. Angular 8 è un framework molto utilizzato per le web application. Permette di utilizzare TypeScript e fornisce compatibilità con librerie e moduli esterni basati anche su JavaScript. I costrutti principali di questo framework sono component e service.

Component: sono dei blocchi formati da tre files differenti contenenti TypeScript, HTML e CSS. Typescript si occupa di ricevere le risposte lato server e gestire le interazione con l'utente.

Service: sono utilizzati per gestire le risposte del server e inviarle in maniera corrette al component.

Angular consente di utilizzare il pattern della **Dependency Injection**. Infatti i service vengono spesso iniettati nei component, che li utilizzano senza dover risolvere dipendenze e senza dover essere modificati al cambiamento di un servizio.

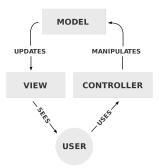
3. Scelte di progettazione

3.1 Model-View-Controller

Spring MVC è il framework che si basa sul pattern Model-view-controller (modello-vista-controllo) è stato scelto come pattern architetturale poiché molto diffuso nello sviluppo di sistemi software. Esso si basa sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

Il model fornisce i metodi per accedere ai dati utili all'applicazione;

Il **view** visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti; Il **controller** riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.



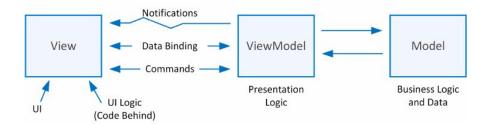
3.2 Model-View-ViewModel

Angular permette l'utilizzo del pattern Model-View-ViewModel: pattern architetturale, simile a MVC, che permette anche una parte di gestione della logica di business.

Il **model** è composto da tutto ciò che è definito nelle classi TypeScript e definisce il contenuto dell'applicazione.

La **view** è tutto ciò che l'utente può osservare e con il quale interagisce, in questo caso il codice HTML e relativi CSS. Il template è collegato al viewModel tramite il cosiddetto data-binding.

Il **viewModel** è la parte astratta della vista ed è responsabile anche di parte della logica. Infatti il viewModel interagisce con il Model, invocando i metodi delle sue classi.



4. Descrizione del sistema

Il sistema consiste in un servizio di Bike sharing universitario, estendibile anche a persone esterne all'università. Questo servizio offre la possibilità di prenotare le bici nell'arco della giornata che sono collocate in apposite rastrelliere nell'area universitaria Bicocca. Per prenotare bisogna scegliere una rastrelliera di inizio, una rastrelliera di fine, un orario di partenza e un orario di termine. Per evitare di tenere occupate le bici per troppo tempo durante tutto l'arco della giornata, l'orario di inizio corsa può essere scelto al massimo entro un'ora. Il personale universitario avrà la possibilità di prenotare le bici gratuitamente, mentre gli utenti esterni dovranno pagare 0,1€/min o in alternativa possono scegliere di acquistare un abbonamento. Una volta prelevata la bici, sarà possibile iniziare la corsa sull'applicazione. Al momento della consegna è necessario terminare la corsa. Se l'orario di consegna non viene rispettato sarà applicata una sanzione di 0,1€/min. Per la gestione del sistema è possibile censire anche utenze di admin. L'utente admin ha la possibilità di visualizzare le statistiche, tramite appositi grafici, riquardanti il numero di bici utilizzate di bici per rastrelliera e l'utilizzo di bici per fascia oraria. L'admin si potrà occupare anche della manutenzione, infatti ha la possibilità di visualizzare le bici in riparazione che, una volta riparate, potrà rendere nuovamente disponibili. Inoltre potrà visualizzare il numero di bici per rastrelliera e durante la fascia oraria 00:00-01:00 potrà riallocare le bici in numero equo presso le apposite rastrelliere.

5. Analisi dei requisiti

5.1 Casi d'uso formato breve

CUB1-Registrazione: Ogni utente per utilizzare il servizio dovrà prima registrarsi, indicando, nel form tutti campi necessari ed inoltre il suo ruolo, che può essere uno tra: ADMIN, PERSONALE, GENERICO. L'utente non potrà utilizzare email già registrate nè tantomeno username già scelti da altri utenti.

CUB2-Login: Ogni utente registrato avrà la possibilità di effettuare il login utilizzando le sue credenziali ovvero : username e password, indicate in fase di registrazione. Potrà accedere così alla sua area personale e quindi utilizzare l'applicazione.

CUB3-Gestione Conto: Gli utenti personale e generico avranno la possibilità di consultare il saldo del proprio conto. Se necessario avranno la possibilità di effettuare un pagamento per ricaricare il conto, inserendo i dettagli della carta.

CUB4-Gestione Abbonamento: Gli utenti con il ruolo generico hanno la possibilità di consultare la scadenza del proprio abbonamento, se ne posseggono uno, oppure di acquistarlo scegliendo tra i diversi abbonamenti disponibili: MENSILE, SEMESTRALE, ANNUALE.

CUB5-Effettua Prenotazione: Gli utenti personale e generico avranno la possibilità di visualizzare le bici disponibili per prenotare una corsa, selezionando l'ora di inizio

prenotazione, l'ora di fine prenotazione, la rastrelliera di partenza e la rastrelliera di arrivo. L'utente personale potrà quindi prenotare gratuitamente la bici scelta, mentre l'utente generico potrà prenotare solo se avrà un saldo sufficiente o un abbonamento attivo.

CUB6-Effettua Corsa: Gli utenti una volta prenotata la bici si dovranno recare presso la rastrelliera indicata durante la prenotazione dalla quale dovranno ritirare la bici con l'identificativo corretto, quindi far partire la corsa. Al termine della corsa dovranno consegnare la bici alla rastrelliera indicata. La consegna dovrà avvenire nell'ora prestabilita, nel caso contrario si incorrerà in una sanzione di 0.1€/min.

CUB7-Visualizza Statistiche: L'admin può consultare due tipologie di statistiche differenti. Il primo, un grafico a torta nel quale vengono visualizzate le percentuali di utilizzo delle bici per fascia oraria nell'arco della giornata. Il secondo, un istogramma nel quale è possibile visualizzare il numero di bici utilizzate per rastrelliera.

CUB8-Ripara bici: L'admin visualizza tutte le bici che sono in riparazione e quindi non disponibili. Può decidere di farle riparare e quindi di renderle nuovamente disponibili.

CUB9-Rialloca bici: L'admin può vedere il numero delle bici per ogni rastrelliera. Nella fascia oraria dedicata, cioè 00:00-01:00, potrà riallocare le bici in modo equo in tutte le rastrelliere.

5.2 Casi d'uso formato dettagliato

Caso d'uso CUD1: Prenota (CUB5-CUB6)

Portata: Applicazione UnimiBike Sharing

Livello: Obiettivo utente

Attore primario: Cliente (Personale, Generico)

Attore secondario: Sistema

Attore finale: Cliente (Personale, Generico)

Parti interessate e interessi

-Cliente: Vuole prenotare ed utilizzare una bicicletta da una determinata rastrelliera ad un'altra e fruire di un servizio rapido, nel modo più semplice possibile. Vuole una visualizzazione chiara delle biciclette disponibili e del costo della corsa. Vuole avere una chiara visione delle prenotazioni pregresse e una gestione facile del suo conto virtuale.

- Azienda: Vuole poter aver a disposizione più biciclette possibili.

Pre-condizioni: Il cliente ha effettuato correttamente il login. Siamo nella fascia oraria 01:00-24:00. Cliente(Generico) ha saldo sufficiente o abbonamento attivo.

Garanzia di successo: La prenotazione viene effettuata con successo. Il cliente preleva la bici nell'ora stabilita e conclude la corsa con successo riponendo la bici nella rastrelliera definita in fase di prenotazione nell'ora stabilita. Il pagamento viene correttamente effettuato. La prenotazione viene salvata correttamente nello storico del cliente.

Scenario Principale di successo:

- 1.Il Cliente effettua il login e viene autenticato.
- 2.Il Cliente si assicura di avere un abbonamento attivo o un saldo sufficiente.
- 3.Il Cliente effettua una ricerca indicando la rastrelliera d'inizio di fine e l'ora di inizio e fine

prenotazione.

- 4.Il Sistema mostra a video le biciclette disponibili indicando l'id, lo stato e la condizione di esse
- 5.Il Cliente seleziona la bici ed effettua la prenotazione
- 6.Il Sistema salva la prenotazione e rende indisponibile la bici selezionata.
- 7.Il Cliente si reca alla rastrelliera all'ora indicata e preleva la bici facendo partire la corsa
- 8.Il Cliente ripone la bici e termina la corsa.
- 9.Il Sistema salva la corsa e rende disponibile la bici nella rastrelliera d'arrivo.

Estensioni(o flussi alternativi):

- 1a. Autenticazione non effettuata correttamente
 - 1.II Sistema segnala l'errore e lo notifica al Cliente
- 3a.L'orario indicato per prelevare la bici è superiore di 1h rispetto all'ora corrente
 - 1.II Sistema segnala che i filtri non sono stati inseriti correttamente.
 - 2.Il Cliente deve rimpostare l'ora di inzio rispettando i vincoli.
- 3b.L'orario indicato per prelevare la bici non rientra negli orari disponibile per le prenotazioni
 - 1.II Sistema segnala l'errore e lo notifica al Cliente
- 4a.La rastrelliera indicata nei filtri della ricerca non ha bici disponibili
 - 1.Il Sistema non mostra bici e da la possibilità di ripetere la ricerca
- 5a.Il saldo del Cliente(Generico) non è sufficiente per effettuare la prenotazione
- 1.Il Sistema segnala che il saldo è da ricaricare prima di poter effettuare la prenotazione
- 8a.L'orario di consegna supera l'orario prestabilito
 - 1.Il Sistema segnala del ritardo e applica una sanzione al Cliente.
- 9a.La bici non è più in condizioni per essere disponibile
 - 1.II Sistema non rende disponibile la bici
 - 2.Il Sistema segnala all'admin che la bici è da riparare

Caso d'uso CUD2: Statistiche (CUB7)
Portata: Applicazione UnimiBike Sharing

Livello: Obiettivo utente Attore primario: Admin Attore secondario: Sistema

Attore finale: Admin

Parti interessate e interessi

- -Amministratore di sistema: Vuole visualizzare le statistiche relative al servizio di Bike Sharing offerto. In particolare vuole visualizzare graficamente le statistiche relative all'utilizzo di bici per fascia oraria e per rastrelliere, per avere un'idea più chiara riguardo l'utilizzo del sistema da parte dei clienti.
- -Azienda: Vuole sfruttare le statistiche fornite al fine di rendere il sistema adatto alle richieste dei clienti.

Pre-condizioni: L'amministratore ha effettuato l'accesso con utenza ADMIN.

Garanzia di successo: L'amministratore accede alla pagina in cui vengono mostrate le statistiche. Il sistema mostrerà un grafico a torta nel quale vengono visualizzate le

percentuali di utilizzo delle bici per fascia oraria nell'arco della giornata ed un istogramma nel quale è possibile visualizzare il numero di bici utilizzate per rastrelliera.

Scenario Principale di successo:

- 1.L'amministratore effettua il login e viene autenticato.
- 2.L'amministratore accede alla pagina in cui vengono mostrate le statistiche.
- 3.Il Sistema effettua il calcolo delle statistiche.
- 4.Il Sistema mostra i grafici richiesti sfruttando i dati calcolati nelle statiche.

Estensioni(o flussi alternativi):

1a. Autenticazione non effettuata correttamente.

1.II Sistema segnala l'errore e lo notifica.

Caso d'uso CUD3: Ripara(CUB8)

Portata: Applicazione UnimiBike Sharing

Livello: Obiettivo utente Attore primario: Admin Attore secondario: Sistema

Attore finale: Admin

Parti interessate e interessi

- -Amministratore di sistema: Vuole monitorare le condizioni delle bici, avendo la possibilità di ripararle.
- -Azienda: Vuole poter aver a disposizione più biciclette possibili.

Pre-condizioni: Deve essere presente almeno una bici da riparare

Garanzia di successo: La bici viene riparata con successo e resa nuovamente disponibile nella apposita rastrelliera.

Scenario Principale di successo:

- 1.L'amministratore effettua il login e viene autenticato.
- 2.Il Sistema mostra la tabella delle bici da riparare.
- 3.L'amministratore cliccando sull'apposito bottone in corrispondenza della bici, la ripara e la rende nuovamente disponibile.

Estensioni(o flussi alternativi):

1a. Autenticazione non effettuata correttamente

1.Il Sistema segnala l'errore e lo notifica

Caso d'uso CUD4: Rialloca(CUB9)

Portata: Applicazione UnimiBike Sharing

Livello: Obiettivo utente Attore primario: Admin Attore secondario: Sistema

Attore finale: Admin

Parti interessate e interessi

-Amministratore di sistema: Vuole monitorare il numero di bici per rastrelliera, avendo la

possibilità di riallocare le bici (in un servizio notturno).

-Azienda: Vuole poter garantire una distribuzione equa delle bici.

Pre-condizioni: L'orario per poter riallocare le bici è 00:00 - 01:00.

Garanzia di successo: Le rastrelliere dopo la riallocazione contengono un numero equo di bici.

Scenario Principale di successo:

- 1.L'amministratore effettua il login e viene autenticato.
- 2.Il sistema mostra il nuemero di bici per rastrelliera.
- 3.L'amministratore cliccando sulla riallocazione distribuisce in numero equo le bici per rastrelliera.

Estensioni(o flussi alternativi):

1a. Autenticazione non effettuata correttamente

1.Il Sistema segnala l'errore e lo notifica

5.3 Definizione requisiti

Per tracciare i requisiti in rapporto agli use case che li soddisfano, è stata utilizzata la seguente classificazione:

R[Tipo][Priorità]-[Codice]

dove il significato di Tipo Priorità e Codice è esposto di seguito.

Tipo: Funzionale, di qualità, di vincolo

- **F**: Requisito funzionale. Questi requisiti descrivono funzionalità del sistema software, in termini di servizi che esso deve fornire, di come reagisce a specifici tipi di input e come si comporta in situazioni particolari;
- **Q**: Requisito di qualità. Includono requisiti di efficacia, efficienza e i requisiti per garantire la qualità nel prodotto;
 - V: Requisito di vincolo. Rappresenta una limitazione tecnologica o strategica.

Priorità: Obbligatorio, Desiderabile, Opzionale

- N: obbligatorio, indica un requisito irrinunciabile per il committente;
- **D**: requisito desiderabile, ma non strettamente necessario;
- **Z**: opzionale, facoltativo. Relativamente utile, ma verrà soddisfatto solo al completamento di tutti gli altri obiettivi.

Codice: codice numerico univoco identificativo del requisito. È costituito da uno o più numeri interi positivi separati da punti, opportunamente ordinati in maniera gerarchica, nel formato X.Y.Z.

5.3.1.Requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'utente non registrato può registrarsi, indicando un ruolo tra admin, personale universitario e utente generico.	CU1
RFN-2	L'utente registrato può effettuare il login per accedere alla sua area personale.	CU2
RFN-3	Gli utenti generico e personale possono visualizzare il proprio conto.	CU3
RFN-4	L'utente generico può vedere se ha un abbonamento attivo e la relativa data di scadenza.	CU4
RFN-5	Gli utenti generico e personale hanno la possibilità di ricaricare il proprio conto inserendo i dettagli per il pagamento online.	CU3
RFZ-6	Gli utenti generico e personale hanno la possibilità di far memorizzare i dettagli per il pagamento	CU3
RFN-7	L'utente generico può acquistare un abbonamento mensile, semestrale o annuale	CU4
RFN-8	Gli utenti generico e personale possono visualizzare le bici disponibili	CU5
RFN-9	Gli utenti generico e personale possono prenotare le bici disponibili.	CU5
RFN-10	Gli utenti generico e personale possono visualizzare le proprie prenotazioni in corso e passate.	CU5 CU6
RFN-11	Gli utenti generico e personale possono iniziare la corsa.	CU6
RFN-12	Gli utenti generico e personale possono terminare la corsa.	CU6
RFN-13	Gli utenti generico e personale saranno soggetti a sanzioni se non rispetteranno l'orario di consegna della bici.	CU6
RFN-14	L'utente admin può visualizzare statistiche	CU7
RFN-15	L'utente admin può visualizzare tutte le bici da riparare.	CU8
RFN-16	L'utente admin può rendere disponibile una bici dopo che è stata riparata.	CU8
RFN-17	L'utente admin può vedere il numero delle bici per rastrelliera.	CU9
RFN-18	L'utente admin può gestire la riallocazione delle bici.	CU9
RFZ-19	Gli utenti generico e personale possono annullare la prenotazione.	CU5

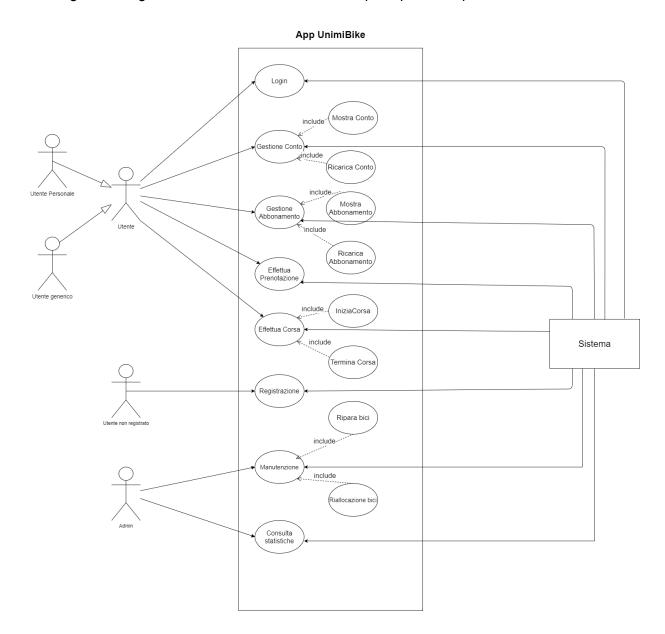
5.3.2.Requisiti non funzionali

Requisito	Descrizione	Use Case
RQZ-1	L'utente avrà dei feedback da parte del sistema durante la fruizione di tutti i servizi.	
RQD-2	L'utente ha la possibilità di scegliere tra tipologie di bici differenti.	
RQD-3	Il sistema gestirà la sessione dell'utente tramite la tecnologia JWT.	
RQD-4	Il sistema mette a disposizione un'interfaccia utente gradevole.	
RVZ-5	Il sistema ha una capacità di memorizzazione limitata dovuta all'utilizzo database embedded.	
RQZ-6	L'utente, registrato e non, può visualizzare la pagina principale in cui è fornita una descrizione sul funzionamento del servizio.	

6.Diagrammi

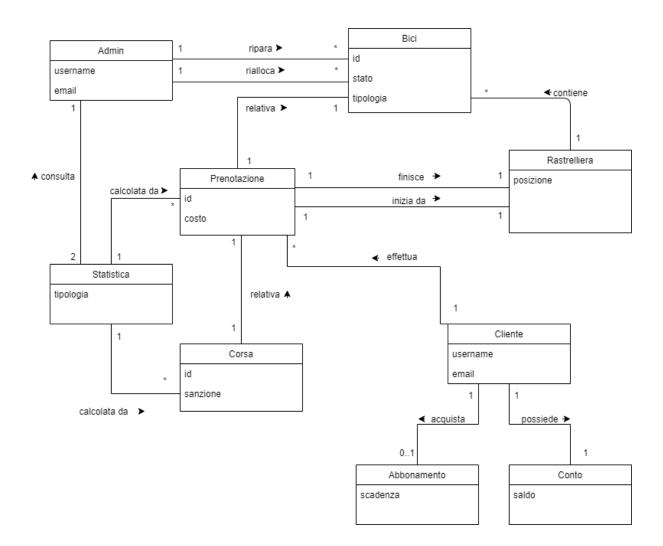
6.1 Diagramma dei casi d'uso

Nel seguente diagramma sono mostrati i casi d'uso principali e i rispettivi attori.



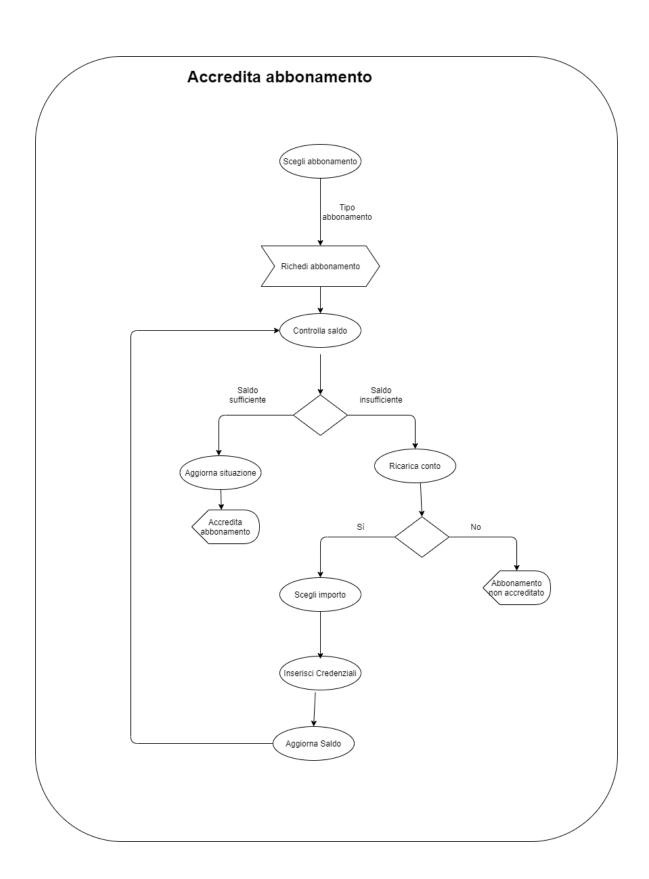
6.2 Diagramma di dominio

Nel seguente diagramma sono mostrate le classi di dominio e le rispettive relazioni.



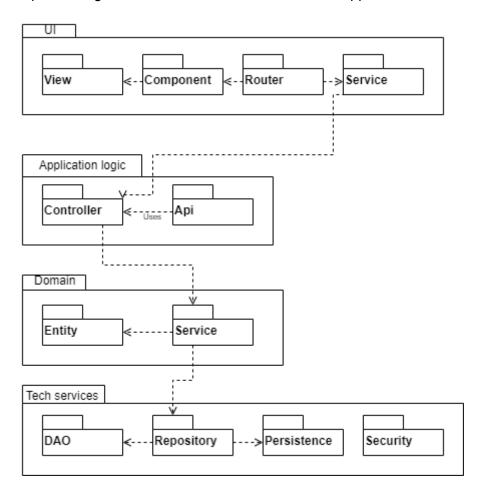
6.3 Diagramma delle attività

In questo diagramma delle attività viene mostrato il processo di acquisto di un abbonamento.



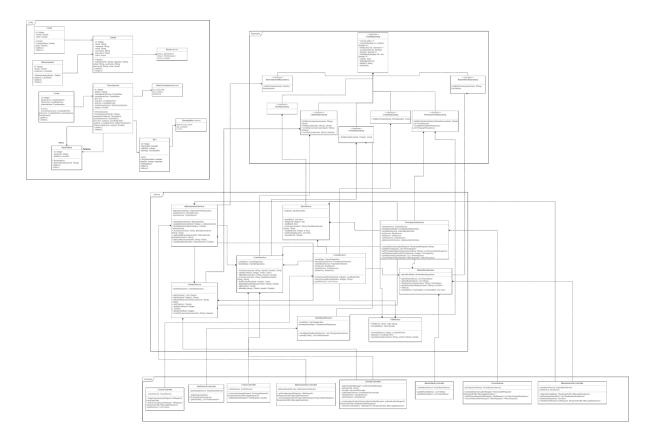
6.4 Diagramma architetturale

In questo diagramma è evidenziata la struttura dell'applicazione UnimiBike.



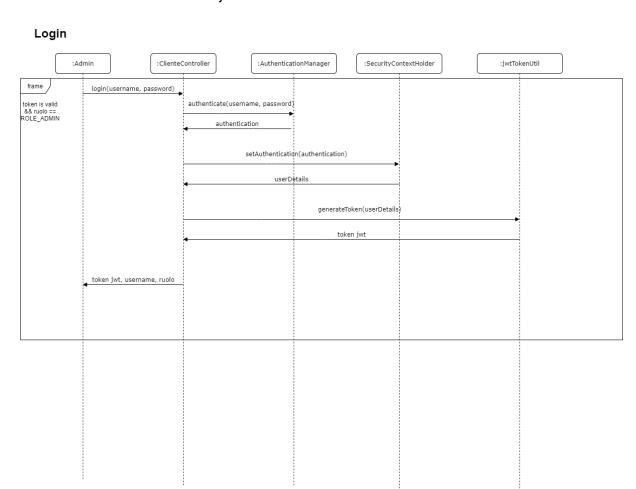
6.5 Diagramma di progettazione

Di seguito è riportato il diagramma di progettazione finale raggruppati per package.



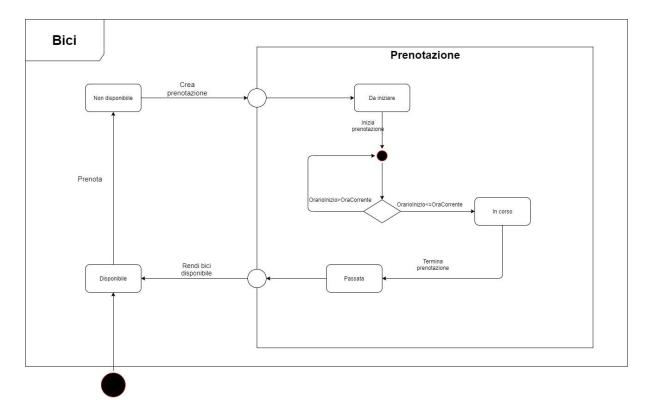
6.6 Diagramma di sequenza

Questo diagramma di sequenza mostra il login nell'applicazione, in particolar modo come avviene la creazione del token jwt.



6.6 Diagramma degli stati

In questo diagramma viene evidenziato il passaggio di stato della bici da disponibile a non disponibile.

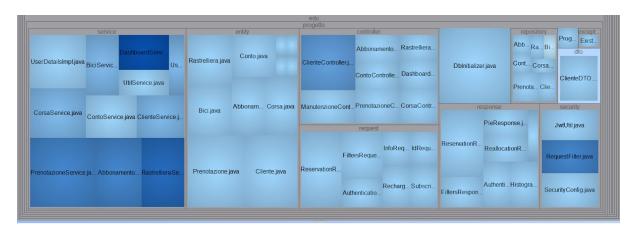


7. Strumenti per l'analisi del software

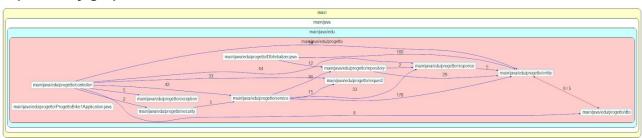
7.1 Understand

Il tool Understand è stato utilizzato per la ricerca di antipatterns strutturali legati a problemi di dipendenza e ad altri tipi di problemi. Nel corso dello sviluppo è stato fondamentale per effettuare del refactoring poiché risultavano dei "Local Breakable" e delle "Local Butterfly".

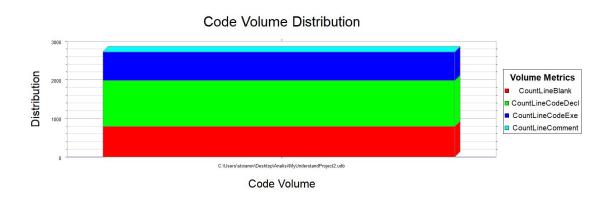
Metrics Treemap:



Dependency graph:

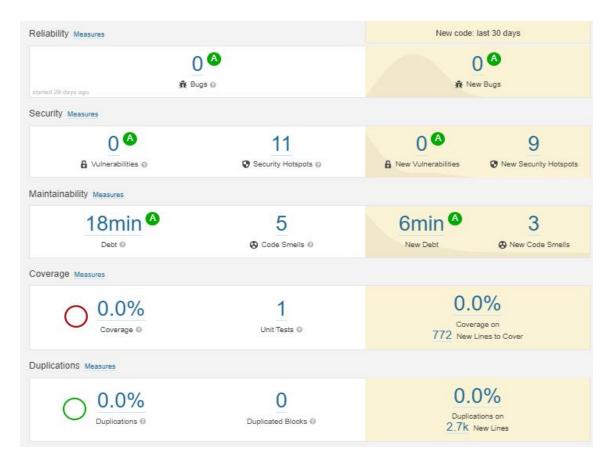


Code Volume Distribution:

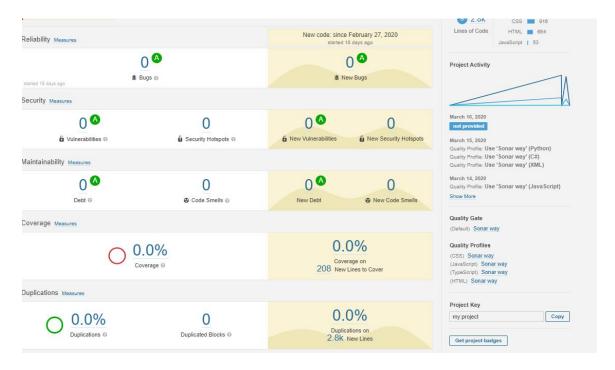


7.2 Sonarqube (per il BackEnd)

Analisi con Sonarqube per la parte di backend.



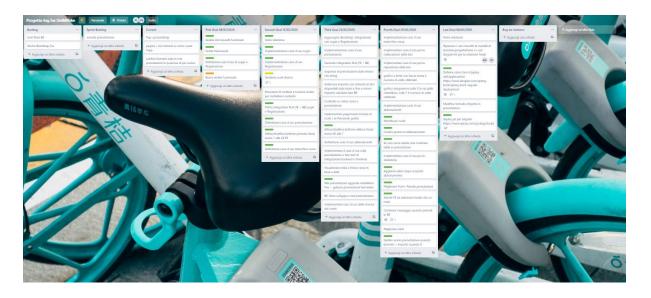
Analisi con Sonarqube per la parte di frontend.



8. Strumenti per la comunicazione

8.1 Trello

Tramite Trello siamo riusciti a coordinare le attività seguendo l'approccio Scrum.

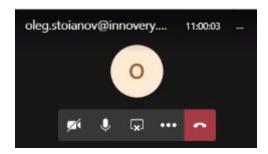


8.2 Teams

Tramite Teams siamo siamo riusciti a restare sempre in contatto, riuscendo a lavorare, quando necessario, anche in condivisione di schermo.



a volte anche andando oltre le 11 ore di conversazione



9.Conclusioni

Il progetto ha portato a noi studenti grandi soddisfazioni facendoci sperimentare tecniche di progettazione e sviluppo del tutto nuove. Lo studio dei framework Spring e Angular nella fase preliminare ha occupato gran parte del tempo a nostra disposizione, ma crediamo di aver ottenuto dei buoni risultati. Inoltre abbiamo compreso l'importanza degli strumenti che il corso ha messo a disposizione, in particolar modo Github, Sonarqube e Understand. In conclusione tutta l'esperienza che abbiamo ricavato dal corso e dal progetto ha ripagato lo sforzo ed inoltre essendo studenti-lavoratori abbiamo già iniziato a mettere in pratica anche in ambito lavorativo, le nozioni apprese, ottenendo già dei buoni risultati.

10. Tutorial installazione App Unimi Bike

Una volta scaricato il jar assicurati di aver installato nel pc la versione java 11, quindi assicurati di avere le variabili d'ambiente impostato correttamente JAVA_HOME e PATH: Lancia da terminale il comando:

java -jar Progetto-Bike1-0.0.1-SNAPSHOT.jar

java -jar Progetto-Bike1-0.0.1-SNAPSHOT.jar

L'applicazione sarà disponibile in locale all'url: http:localhost:8080 sul tuo browser.