



A.A. 2020/2021

# Relazione Progetto Bike Sharing

Corso di "Ingegneria del Software"

Giuseppe Del Campo, 935114

Mese di consegna: Settembre 2021



# 1 Descrizione del problema

**Premessa iniziale:** ogni immagine e grafico visualizzabile nella relazione sono disponibili nella folder chiamata “images” e nel file BikeSharing.mdj presenti all’interno dell’archivio, nel caso non fossero sufficientemente visibili.

## 1.1 Analisi e specifica dei requisiti

Il progetto consiste nella realizzazione di un sistema di bike sharing.

Gli asset fisici di questo sistema sono diversi: le **biciclette** sono di tre tipi, ovvero normali, elettriche, ed elettriche con seggiolino; le **morse** sono invece degli agganci in cui vanno inserite le biciclette, e possono essere normali oppure elettriche. I **totem** sono dei dispositivi presenti vicino le postazioni contenenti le morse, e sono dotati di schermo touchscreen: questi permettono ad un utente di iscriversi, effettuare il login, noleggiare una bici, e comunicare eventuali danni che possono essere avvenuti durante l’utilizzo della bicicletta.

Gli attori principali del sistema sono gli *utenti*: per registrarsi è necessario effettuare un abbonamento che può essere giornaliero, settimanale, annuale oppure riferito ad un utente che fa parte del personale amministrativo; nel caso l’utente sia uno studente avrà diritto a delle agevolazioni. È inoltre necessario presentare i dati della carta di credito per poter pagare l’abbonamento, e anche una password che servirà poi ad effettuare il login. Se i dati inseriti sono validi, verrà restituito un codice che l’utente potrà poi utilizzare insieme alla password per potersi autenticare per tutta la durata dell’abbonamento. Nel caso di abbonamento giornaliero oppure settimanale la durata avrà inizio a partire dal primo prelievo di una bicicletta; nel caso di abbonamento annuale invece ha inizio appena terminata la registrazione.

Per poter registrare un nuovo abbonamento, la carta di credito deve essere valida per almeno un anno nel caso di quello annuale, e per almeno 3 mesi nel caso di abbonamento giornaliero o settimanale. In caso contrario non sarà possibile creare l’abbonamento. Nel caso di abbonamento giornaliero o settimanale l’utente avrà quindi a disposizione 3 mesi di tempo per usufruire dei servizi e far partire l’abbonamento (tramite il primo noleggio effettuato).

Un *utente* può noleggiare una bicicletta solo nel caso siano passati almeno cinque minuti dall’ultimo noleggio; inoltre, il noleggio non può durare più di due ore consecutive: nel caso il noleggio superi le due ore per tre volte durante la durata dell’abbonamento, quest’ultimo viene annullato. Nel caso venga

restituita una bicicletta dopo più di 24 ore dal noleggio, l'utente dovrà pagare una penale di 150€.

Un *utente* può essere anche un *responsabile del comune*: in tal caso avrà a disposizione ulteriori strumenti per la gestione del sistema, ad esempio per aggiungere o rimuovere biciclette dalla rastrelliera, aggiungere o rimuovere morse da una rastrelliera, aggiungere e rimuovere rastrelliere, riparare biciclette che sono state segnalate come danneggiate, e visualizzare dei dati statistici riguardo l'utilizzo complessivo del sistema.

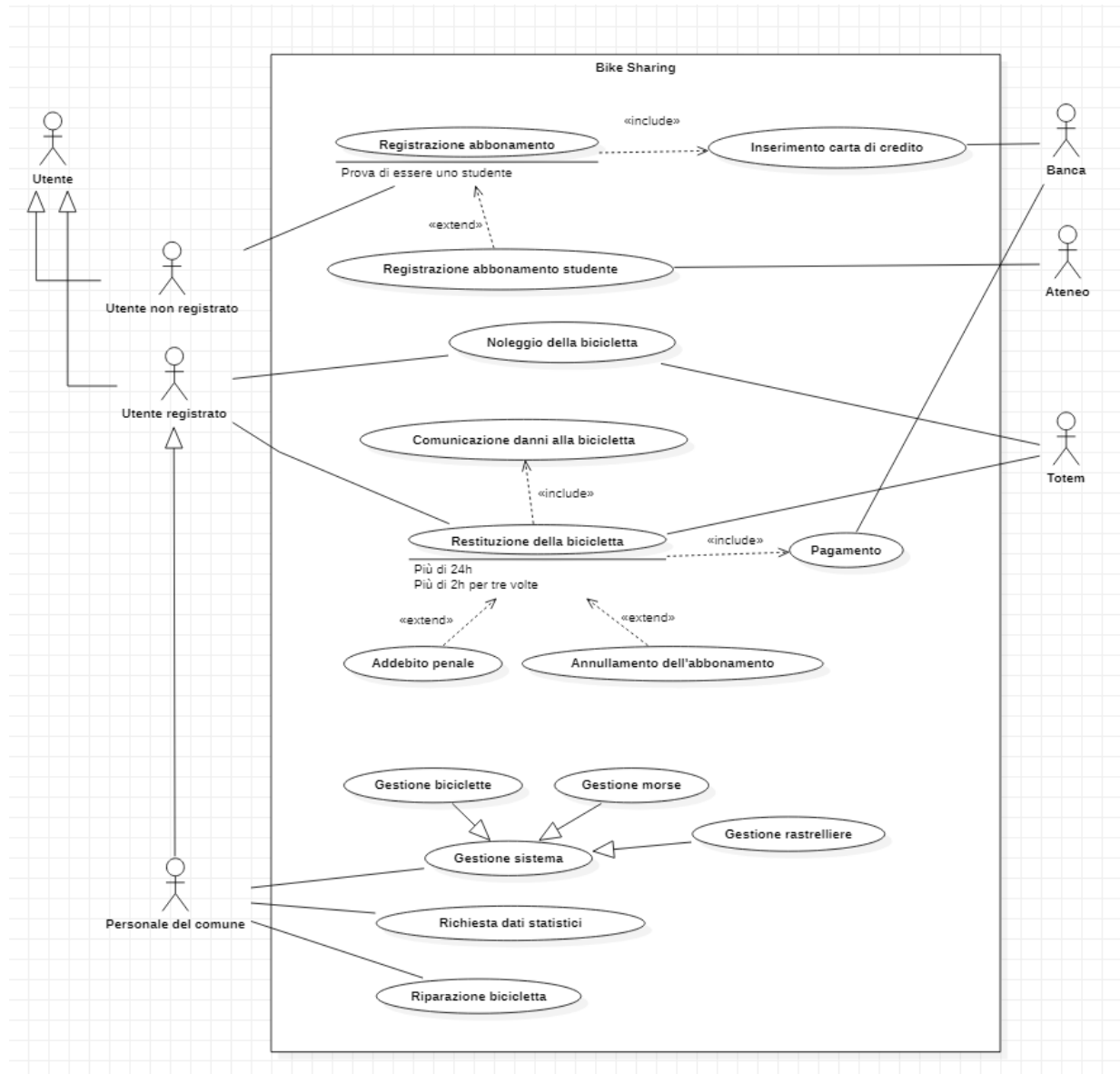
Le tariffe sono visualizzate nella seguente tabella:

Tipo di bicicletta	Prima mezz'ora	Seconda mezz'ora	Terza mezz'ora	Quarta mezz'ora	Ore successive
Normale	Gratis	0,50€	0,50€	0,50€	2,00€
Elettrica	0,25€	0,50€	1,00€	2,00€	4,00€
Elettrica con seggiolino	0,25€	0,50€	1,00€	2,00€	4,00€

Nel caso l'utente sia uno studente, l'utilizzo della bicicletta di tipo "normale" sarà sempre gratuito.

## 2 Progettazione del sistema

### 2.1 Diagramma dei casi d'uso



- **Registrazione abbonamento:** l'utente inserisce la durata dell'abbonamento che vuole effettuare, e indica una password che servirà poi per l'autenticazione; viene assegnato all'utente un codice univoco.
- **Registrazione abbonamento studente:** l'utente, nel caso dimostri di essere uno studente, potrà effettuare un abbonamento seguendo le tariffe da studente;
- **Inserimento carta di credito:** l'utente inserisce i dati riguardanti la carta di credito; in questo modo il sistema verificherà se è valida;
- **Ritiro della bicicletta:** l'utente inserisce il proprio codice e password sul totem apposito per effettuare l'autenticazione, e seleziona il tipo di bicicletta che vuole



noleggiare dalla rastrelliera. Il sistema visualizza sul display del totem il numero della posizione nella quale si trova la bicicletta che potrà poi essere utilizzata;

- **Restituzione della bicicletta:** l'utente restituisce la bici depositandola nella morsa del tipo corretto in base alla bicicletta noleggiata. Il sistema visualizzerà sul display del totem un messaggio per avvisare l'utente della corretta avvenuta restituzione;
- **Pagamento:** quando l'utente restituisce la bicicletta, il sistema effettua un addebito sulla carta di credito in base alle caratteristiche del noleggio effettuato;
- **Comunicazione danni alla bicicletta:** il cliente può comunicare tramite l'apposito totem se la bicicletta ha subito dei danni oppure ha dei difetti, in modo tale che possa essere portata in manutenzione dal personale;
- **Addebito penale:** nel caso l'utente non rispetti le condizioni specificate inizialmente riguardo la restituzione dopo 24 ore, verrà addebitata una penale;
- **Annullamento dell'abbonamento:** nel caso l'utente violi ripetutamente le condizioni specificate inizialmente riguardo la durata del noleggio, il suo abbonamento verrà annullato;
- **Gestione sistema:** il personale del comune può gestire il sistema manipolando le biciclette, le morse, e le rastrelliere;
- **Gestione biciclette:** il personale del comune può aggiungere oppure rimuovere una bicicletta da una rastrelliera;
- **Gestione morse:** il personale del comune può aggiungere oppure rimuovere una morsa da una rastrelliera;
- **Gestione rastrelliere:** il personale del comune può aggiungere oppure rimuovere rastrelliere dal sistema;
- **Riparazione bicicletta:** il personale del comune può eventualmente riparare una bicicletta che è stata segnalata essere danneggiata;
- **Richiesta dati statistici:** il personale del comune può richiedere dei dati statistici riguardo l'utilizzo complessivo del sistema, come ad esempio l'utilizzo medio giornaliero delle biciclette o la rastrelliera con maggiori e minori richieste, quante bici sono state danneggiate ecc.

## 2.2 Descrizione degli scenari

Nome	<i>Registrazione abbonamento</i>
Scopo	Permettere ad un utente di registrarsi per poi poter effettuare il login e utilizzare i servizi offerti dal sistema
Attore/i	Utente non registrato, Banca
Pre-condizioni	L'utente non è registrato
Trigger	L'utente si trova ad un totem e preme il pulsante per la registrazione
Descrizione sequenza eventi	<ol style="list-style-type: none"> <li>1. L'utente sceglie il tipo di abbonamento e la password;</li> <li>2. L'utente dichiara di essere uno studente;</li> <li>3. L'utente inserisce i dati riguardanti la carta di credito;</li> </ol>
Alternativa/e	2a. L'utente dichiara di non essere uno studente

<b>Post-condizioni</b>	L'utente è dotato del suo codice univoco e l'abbonamento risulta presente nel sistema
------------------------	---

<b>Nome</b>	<i>Noleggio della bicicletta</i>
<b>Scopo</b>	Permettere ad un utente registrato e quindi dotato di abbonamento di noleggiare una bicicletta
<b>Attore/i</b>	Utente registrato, Totem
<b>Pre-condizioni</b>	
<b>Trigger</b>	L'utente preme sul pulsante per il noleggio di una bicicletta
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente effettua il login nel sistema inserendo il proprio codice utente e password</li> <li>2. L'utente sceglie quale tipo di bicicletta vorrebbe noleggiare</li> <li>3. La bicicletta del tipo selezionato viene sbloccata e può essere prelevata dall'utente</li> </ol>
<b>Alternativa/e</b>	2a. Nel caso non sia disponibile nella rastrelliera una bicicletta del tipo richiesto dall'utente, viene visualizzato un messaggio apposito e il noleggio non viene effettuato
<b>Post-condizioni</b>	L'utente noleggia una bicicletta e inizia il noleggio

<b>Nome</b>	<i>Restituzione della bicicletta</i>
<b>Scopo</b>	Permettere ad un utente di restituire una bicicletta precedentemente noleggiata e concludere il noleggio
<b>Attore/i</b>	Utente registrato, Totem
<b>Pre-condizioni</b>	L'utente è in possesso di una bicicletta noleggiata da restituire
<b>Trigger</b>	L'utente reinserisce la bicicletta noleggiata in una morsa apposita
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente inserisce la bicicletta noleggiata nella morsa del tipo corretto in base al tipo di bicicletta (quindi normale oppure elettrica)</li> <li>2. Il noleggio termina e viene addebitato sulla carta di credito dell'utente il costo complessivo del noleggio</li> </ol>
<b>Alternativa/e</b>	<p>2a. L'utente effettua il login tramite il totem</p> <p>2b. Sul totem viene visualizzata l'avvenuta restituzione della bicicletta</p> <p>2c. Il sistema chiede all'utente se sono stati causati dei danni alla bicicletta</p> <p>3a. Nel caso siano passate più di 24 ore dal noleggio della bicicletta viene addebitata una penale sulla carta di credito dell'utente</p> <p>3b. Nel caso sia la terza volta che una bicicletta noleggiata viene restituita dopo due ore dall'inizio del noleggio, l'abbonamento dell'utente viene annullato.</p>
<b>Post-condizioni</b>	La bicicletta è restituita e il noleggio di conseguenza terminato; l'abbonamento dell'utente è eventualmente annullato

<b>Nome</b>	<i>Gestione biciclette</i>
<b>Scopo</b>	Permettere ad un utente che fa parte del personale del comune di aggiungere o rimuovere biciclette da una rastrelliera
<b>Attore/i</b>	Personale del comune
<b>Pre-condizioni</b>	L'utente che ha effettuato il login è del tipo corretto
<b>Trigger</b>	L'utente preme sul pulsante per la gestione del sistema
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie una bicicletta da aggiungere</li> <li>2. L'utente sceglie la postazione con totem</li> <li>3. Viene visualizzata una conferma sul display del totem</li> </ol>
<b>Alternativa/e</b>	1a. L'utente sceglie una bicicletta da rimuovere
<b>Post-condizioni</b>	La bicicletta viene aggiunta nella rastrelliera in caso ci sia una morsa disponibile del tipo corretto, oppure la bicicletta viene rimossa dalla rastrelliera

<b>Nome</b>	<i>Gestione morse</i>
<b>Scopo</b>	Permettere ad un utente che fa parte del personale del comune di aggiungere o rimuovere morse da una rastrelliera
<b>Attore/i</b>	Personale del comune
<b>Pre-condizioni</b>	L'utente che ha effettuato il login è del tipo corretto
<b>Trigger</b>	L'utente preme sul pulsante per la gestione del sistema
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie un tipo di morsa da aggiungere</li> <li>2. L'utente sceglie la postazione con totem</li> <li>3. Viene visualizzata una conferma sul display del totem</li> </ol>
<b>Alternativa/e</b>	1a. L'utente sceglie una morsa da rimuovere
<b>Post-condizioni</b>	La morsa viene aggiunta oppure rimossa dalla rastrelliera

<b>Nome</b>	<i>Gestione rastrelliere</i>
<b>Scopo</b>	Permettere ad un utente che fa parte del personale del comune di aggiungere o rimuovere rastrelliere dal sistema
<b>Attore/i</b>	Personale del comune
<b>Pre-condizioni</b>	L'utente che ha effettuato il login è del tipo corretto
<b>Trigger</b>	L'utente preme sul pulsante per la gestione del sistema
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie l'indirizzo dove creare la nuova postazione con totem</li> <li>2. Viene visualizzata una conferma sul display del totem</li> </ol>
<b>Alternativa/e</b>	1a. L'utente sceglie l'indirizzo della postazione con totem da rimuovere
<b>Post-condizioni</b>	La postazione con totem viene aggiunta oppure rimossa dal sistema

<b>Nome</b>	<i>Riparazione bicicletta</i>
<b>Scopo</b>	Permettere ad un utente che fa parte del personale del comune di riparare una bicicletta che è stata segnalata essere danneggiata
<b>Attore/i</b>	Personale del comune
<b>Pre-condizioni</b>	L'utente che ha effettuato il login è del tipo corretto

<b>Trigger</b>	L'utente preme sul pulsante per la gestione del sistema
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente sceglie una bicicletta danneggiata da riparare</li> <li>2. Viene visualizzata una conferma sul display del totem</li> </ol>
<b>Alternativa/e</b>	
<b>Post-condizioni</b>	La bicicletta risulta essere non più danneggiata

<b>Nome</b>	<i>Richiesta dati statistici</i>
<b>Scopo</b>	Permettere ad un utente che fa parte del personale del comune di visualizzare le statistiche inerenti al sistema
<b>Attore/i</b>	Personale del comune
<b>Pre-condizioni</b>	L'utente che ha effettuato il login è del tipo corretto
<b>Trigger</b>	L'utente preme sul pulsante per la gestione del sistema
<b>Descrizione sequenza eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente preme sul pulsante per la gestione del sistema</li> <li>2. L'utente visualizza i dati statistici sull'utilizzo del sistema</li> </ol>
<b>Alternativa/e</b>	
<b>Post-condizioni</b>	L'utente ha visualizzato le statistiche riguardanti il sistema



## 2.3 Diagramma delle classi (modello di programma)

Di seguito è possibile osservare il diagramma delle classi (di programma). Nella parte del dominio non è stato fatto un particolare uso dell'ereditarietà in quanto sin dall'inizio si è tenuto conto della futura interazione con il database, e di conseguenza è stato deciso di utilizzare appositi tipi per avere una compatibilità maggiore con la relazionalità del database.

Nel caso l'immagine non sia particolarmente visibile all'interno della relazione, è possibile osservare il diagramma nel file **BikeSharing.mdj**.

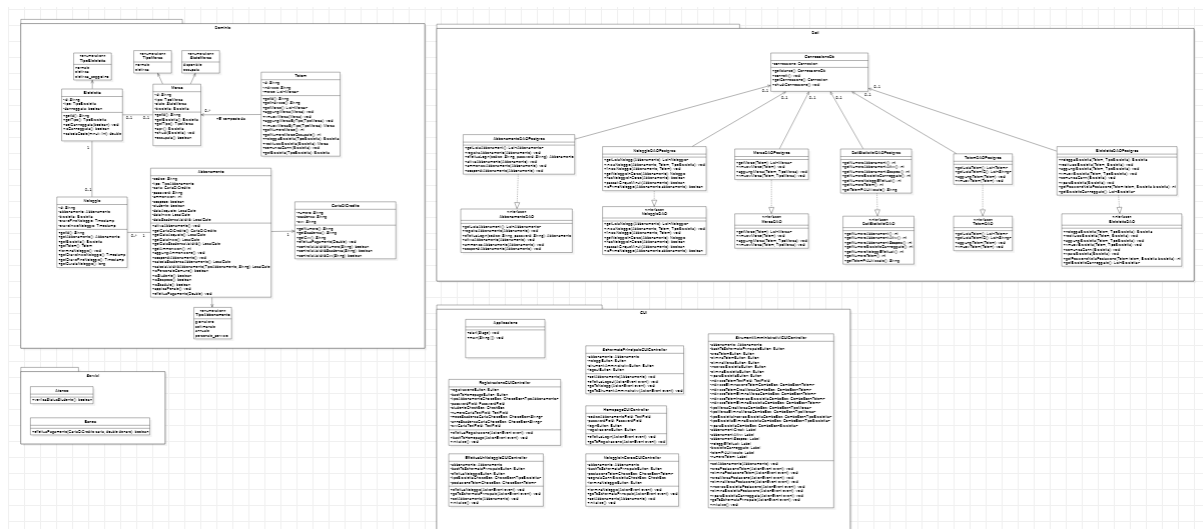
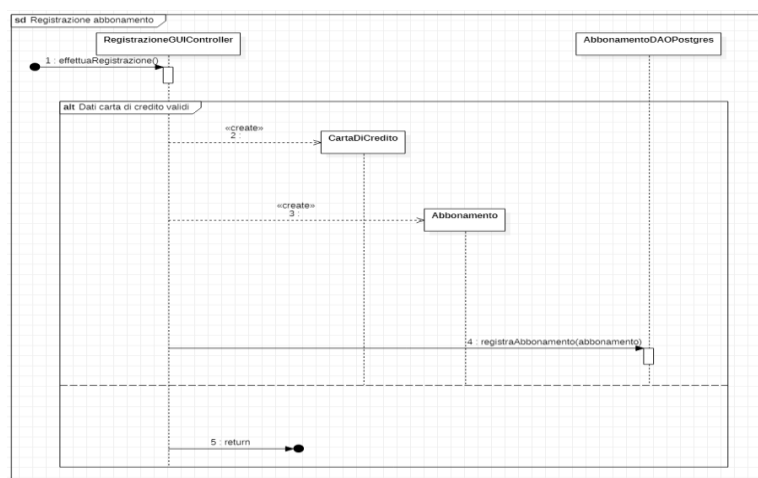


Figura 1: Diagramma delle classi (modello di programma)

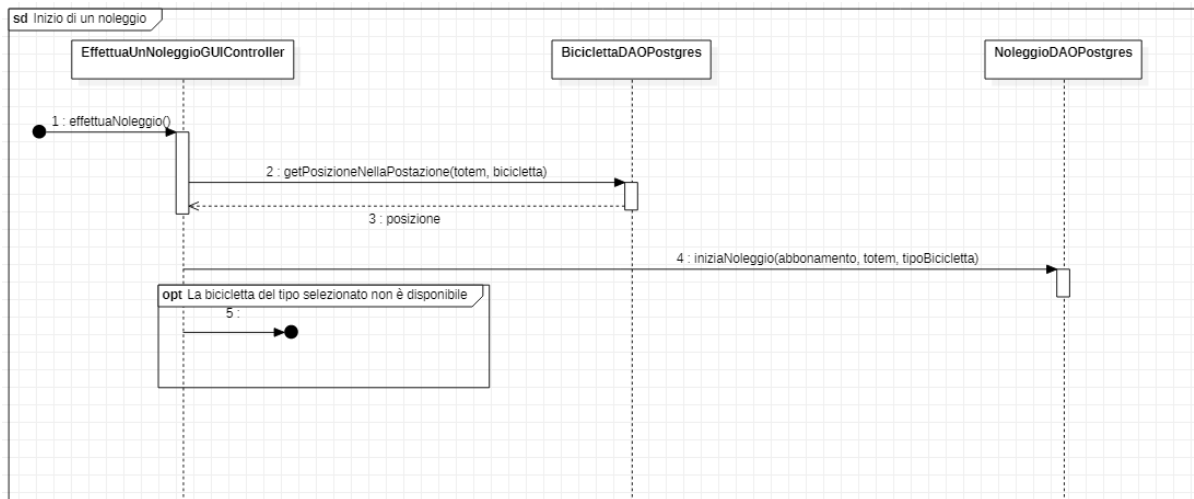
## 2.4 Diagrammi di sequenza

Di seguito è possibile osservare i diagrammi di sequenza realizzati:

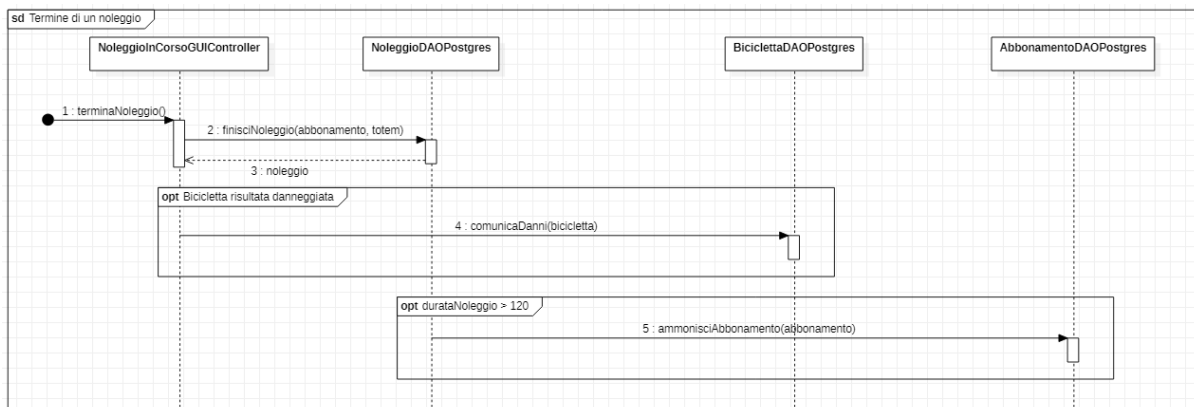
### 2.4.1 Registrazione di un abbonamento



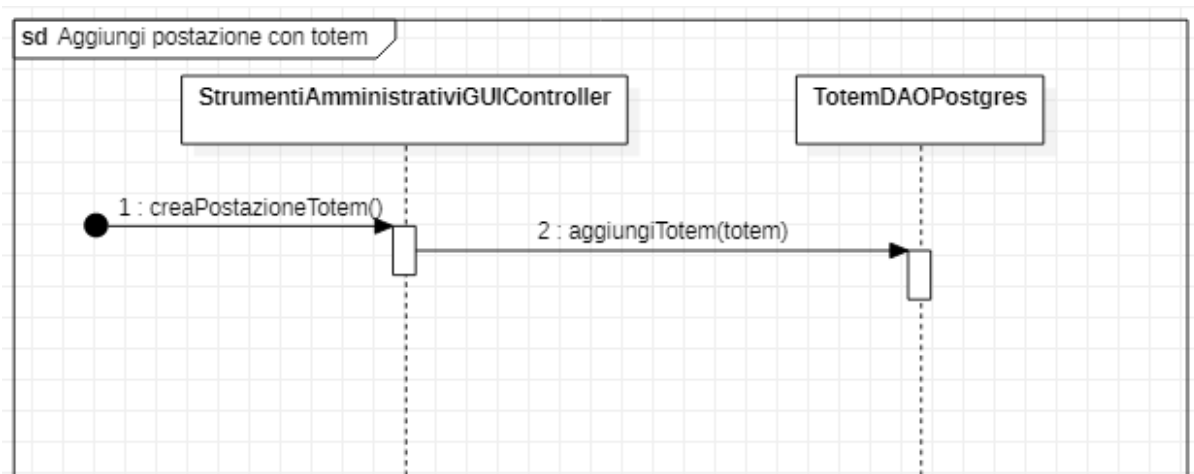
## 2.4.2 Inizio di un noleggio



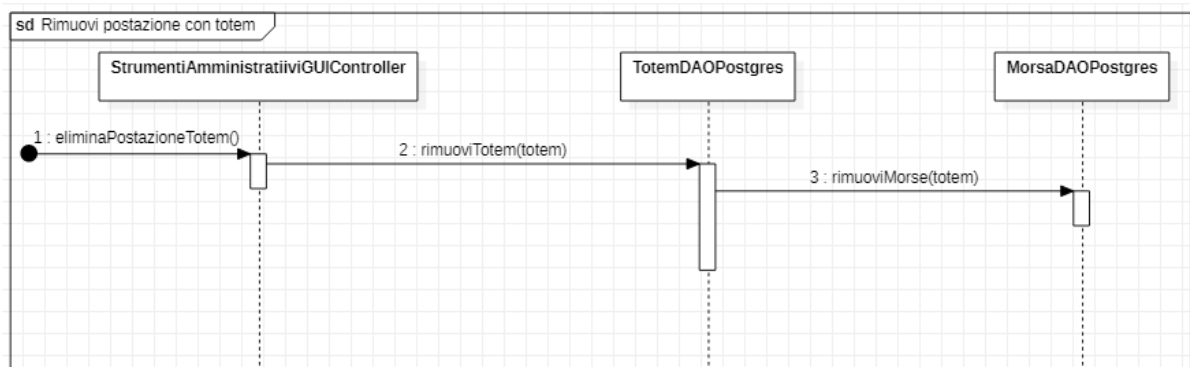
## 2.4.3 Termine di un noleggio



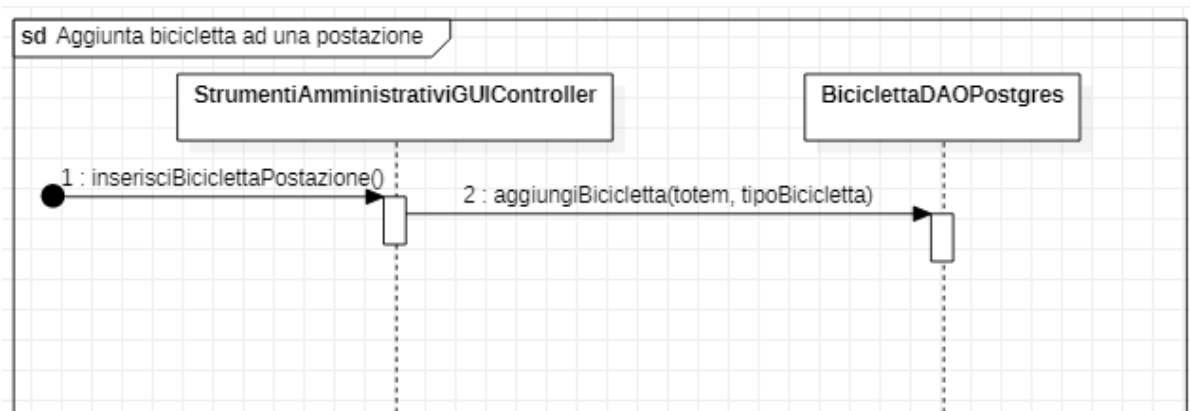
## 2.4.4 Creazione di una postazione con totem



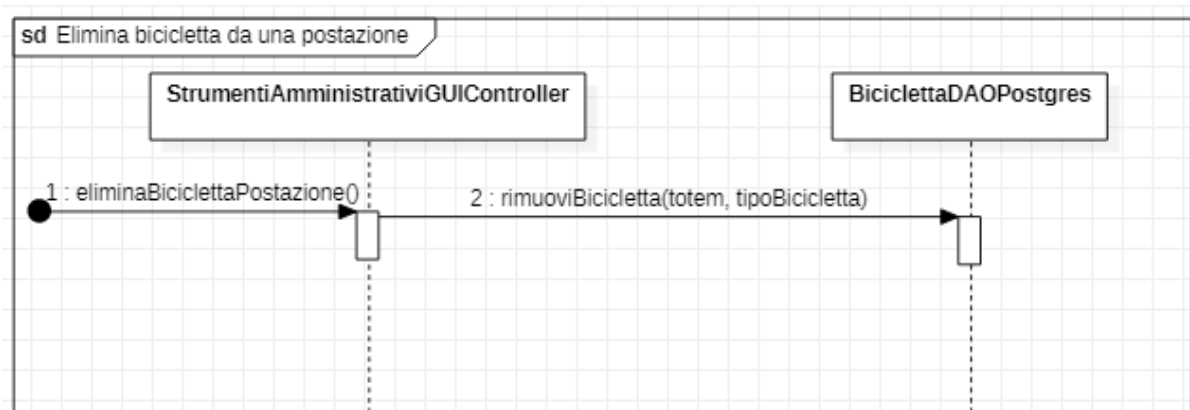
### 2.4.5 Eliminazione di una postazione con totem



### 2.4.6 Aggiunta di una bicicletta ad una postazione con totem

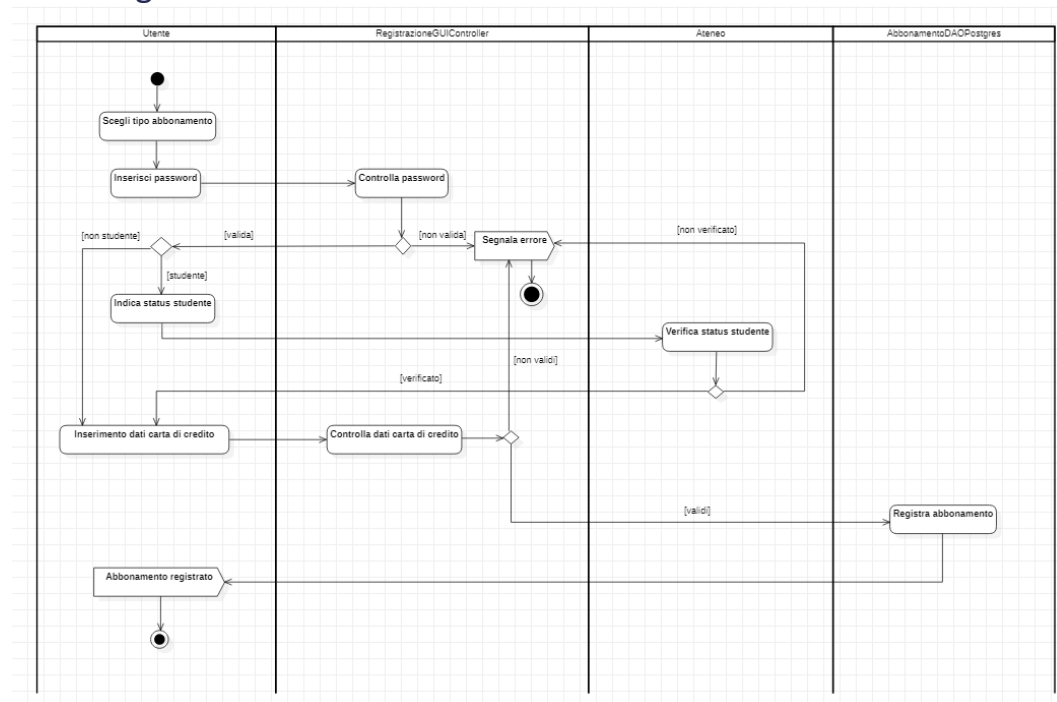


### 2.4.7 Eliminazione di una bicicletta da una postazione con totem

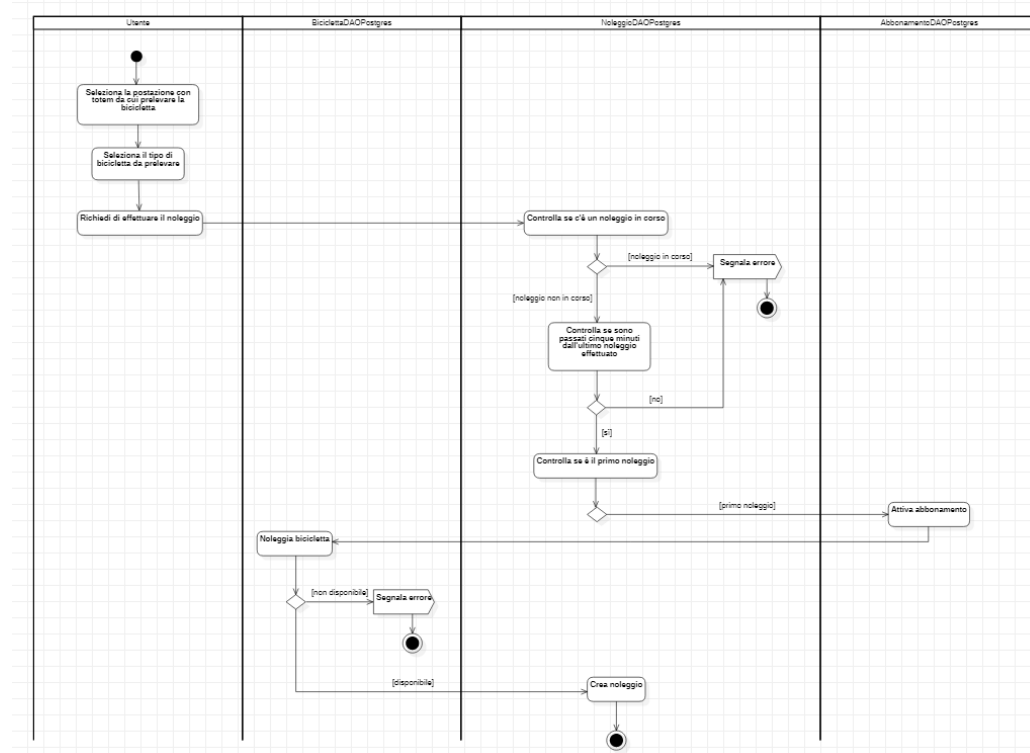


## 2.5 Diagrammi delle attività

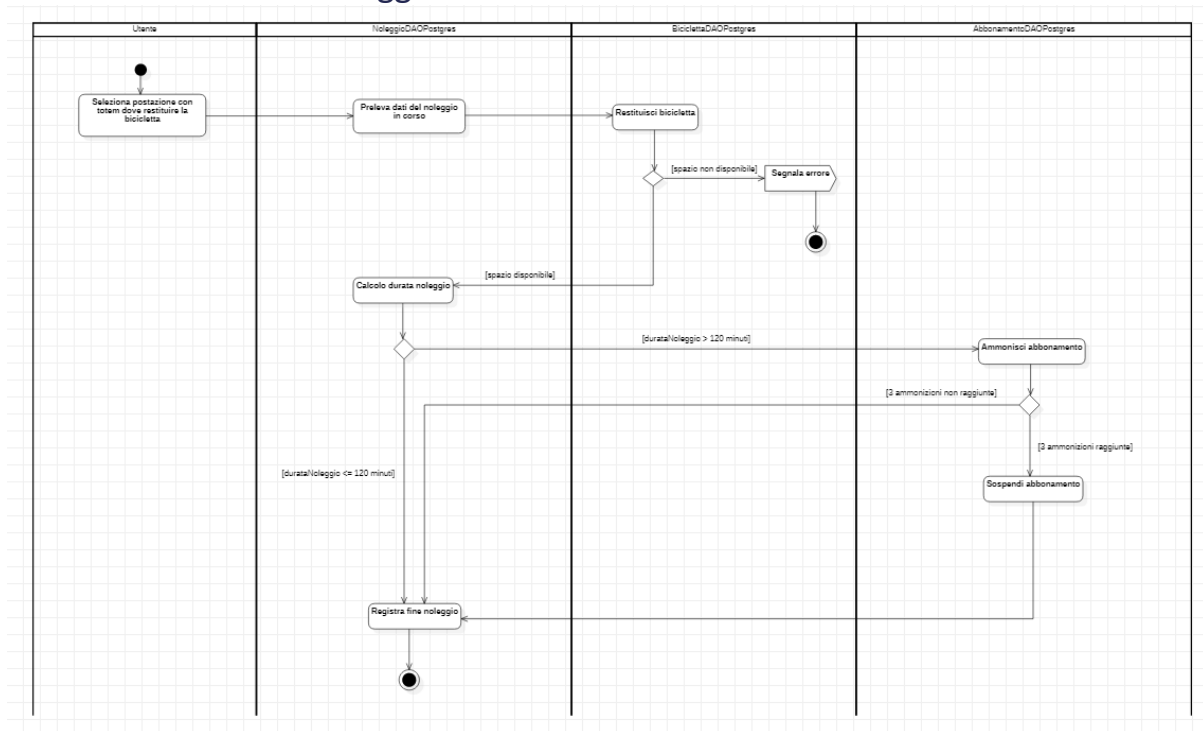
### 2.5.1 Registrazione di un abbonamento



### 2.5.2 Inizio di un noleggio

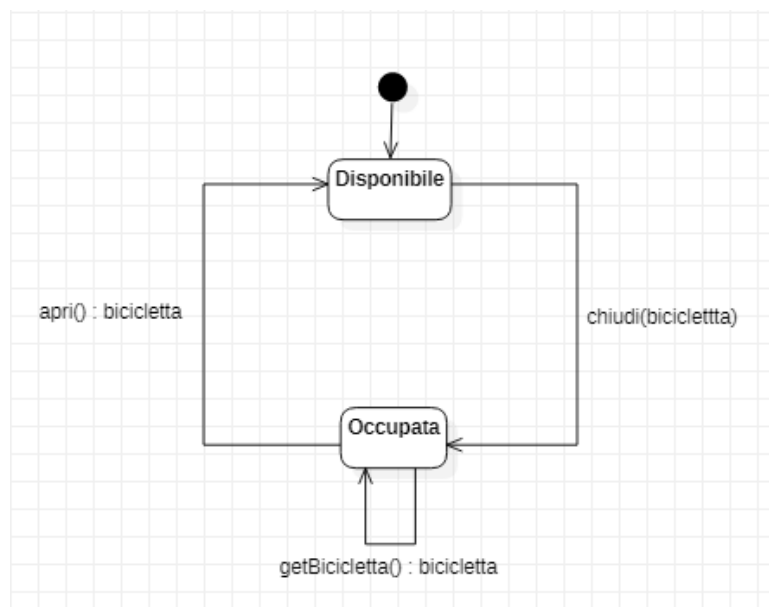


## 2.5.3 Termine di un noleggio

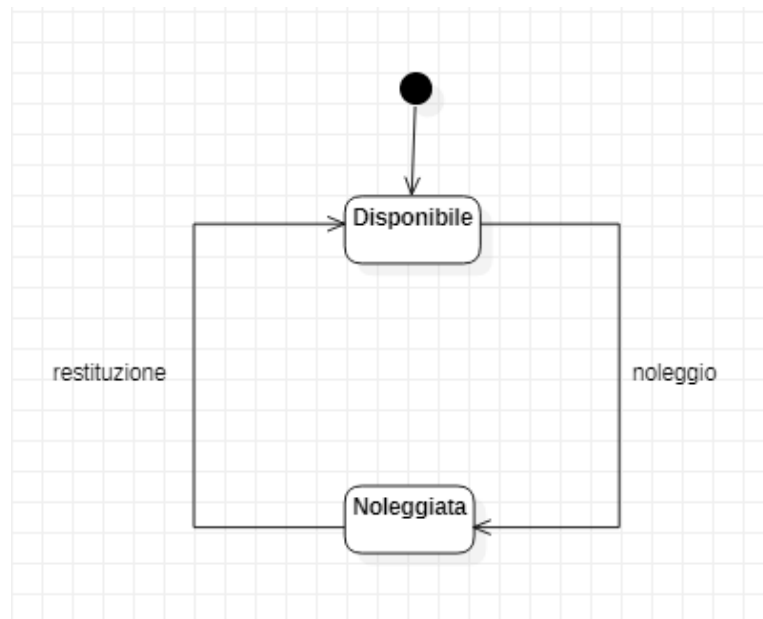


## 2.6 Macchine di stato

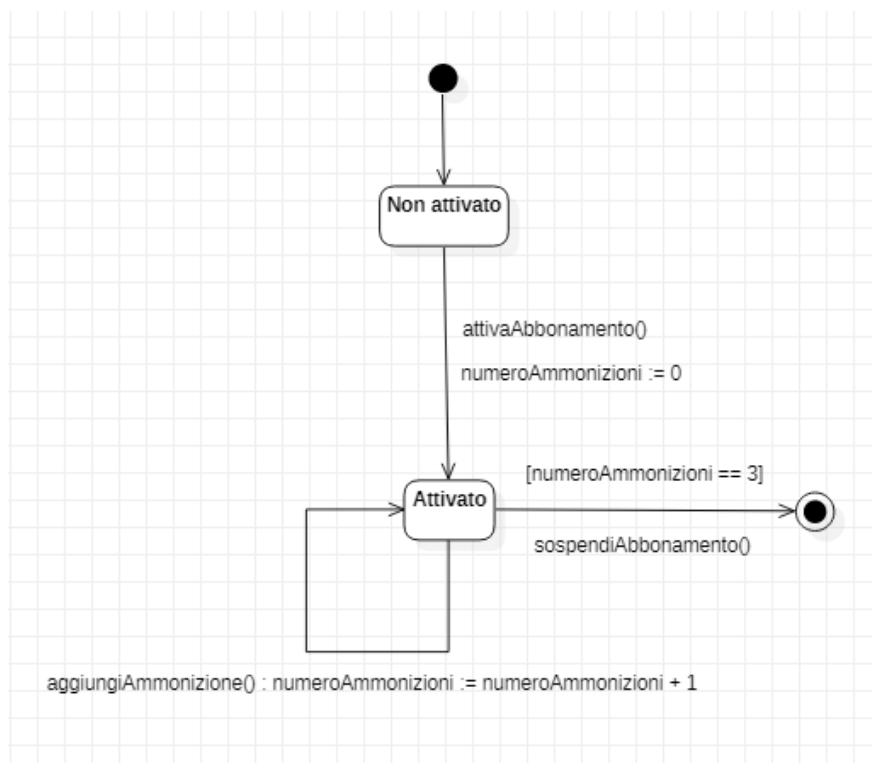
### 2.6.1 Macchina di stato della morsa



### 2.6.2 Macchina di stato della bicicletta

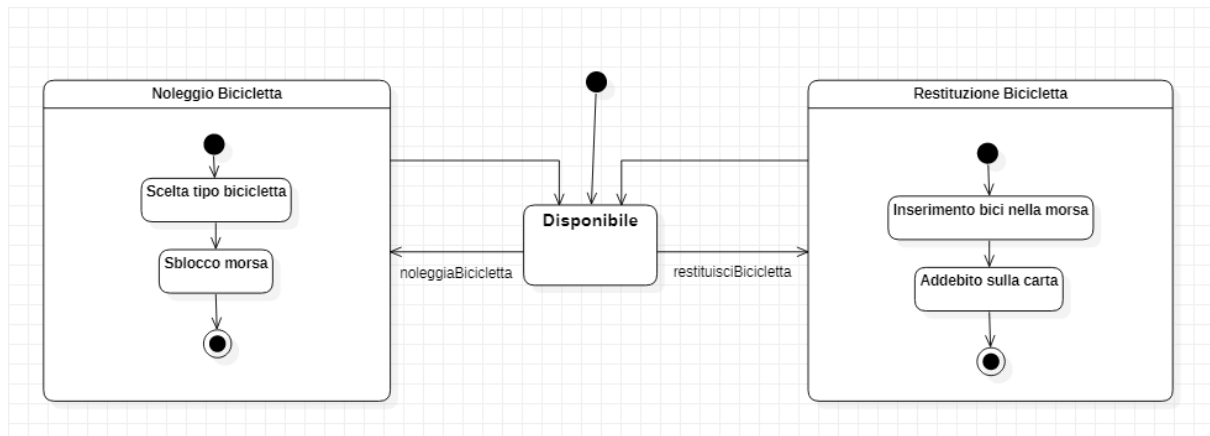


### 2.6.3 Macchina di stato dell'abbonamento



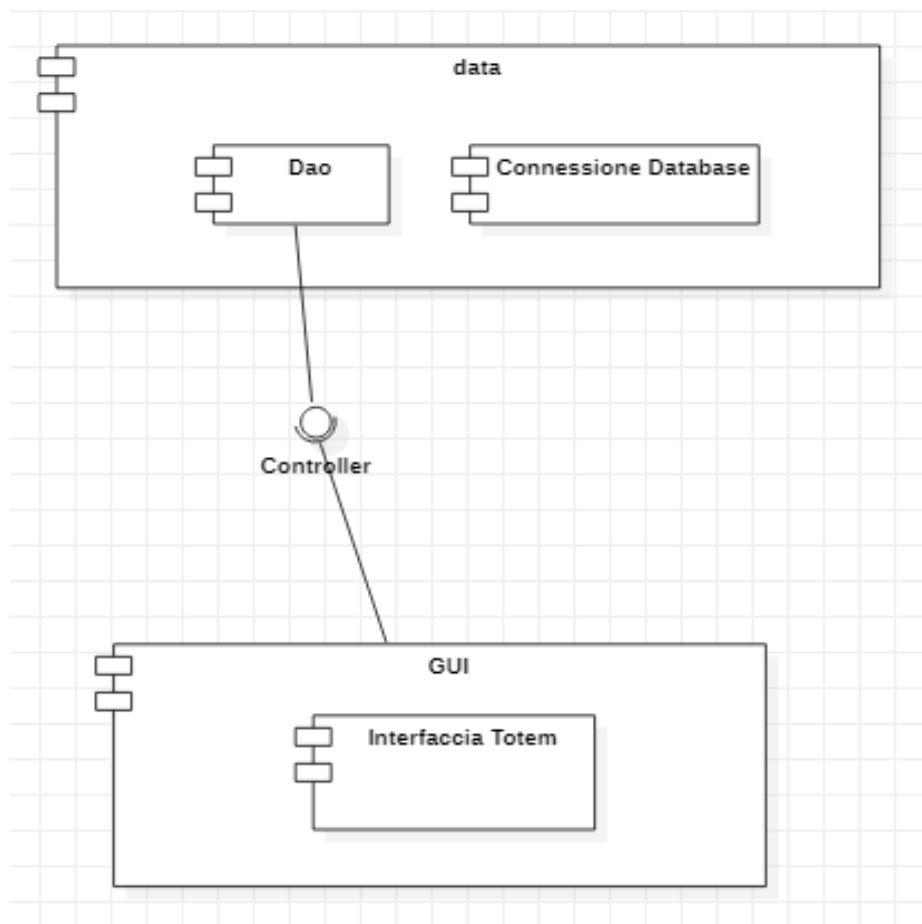


## 2.6.4 Macchina di stato del totem



## 2.7 Diagramma dei componenti

Di seguito è possibile osservare il diagramma dei componenti:



## 3 Implementazione del sistema

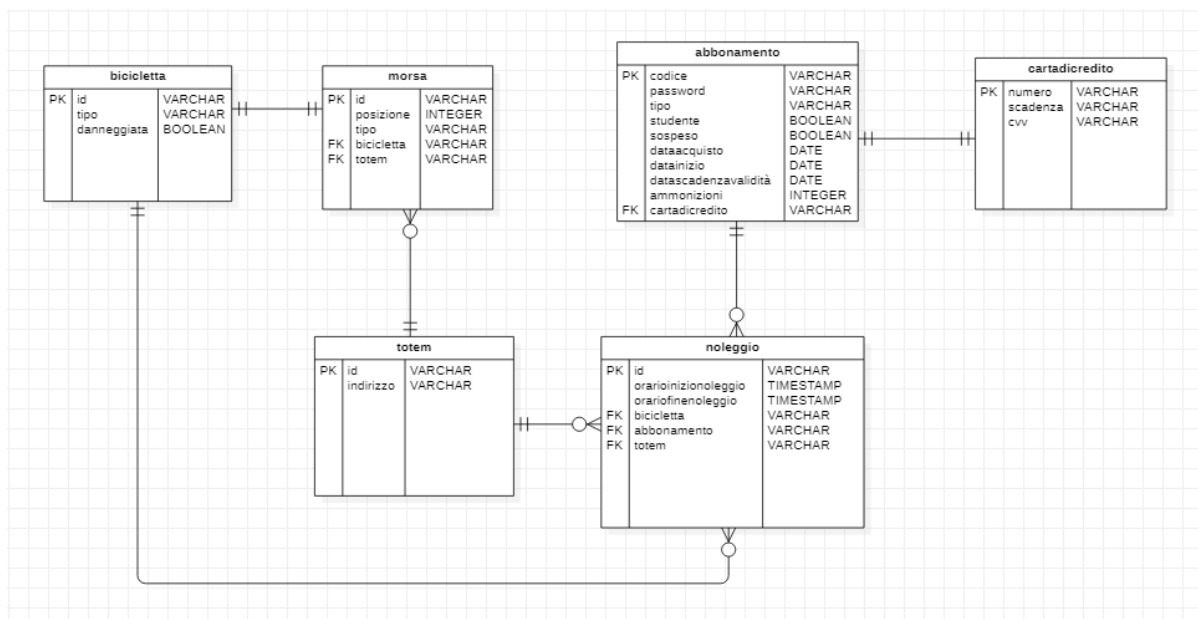
### 3.1 Discussione dei Design Pattern utilizzati

- **Singleton:** questo design pattern viene usato nel caso ci siano classi per cui deve esistere una sola istanza; in questo caso è stato utilizzato per la classe che gestisce la connessione al database.
- **Design Pattern Object (DAO):** questo design pattern è stato utilizzato per le classi che hanno dovuto interagire con il database. Permette di separare i dati presenti in memoria dall'effettiva gestione e accesso dei dati presenti nel database, e tramite l'utilizzo di apposite interfacce si può cambiare in modo piuttosto semplice il sistema che è stato utilizzato.
- **Model View Controller (MVC):** questo design pattern è stato usato per suddividere gli strati dell'applicazione in modello, vista e controller. Il modello è costituito dalle classi che compongono il dominio e da quelle che gestiscono i dati persistenti; la vista è gestita dalla GUI tramite l'utilizzo di SceneBuilder e dei vari documenti FXML; infine il controller gestisce l'interazione tra GUI e modello. Utilizzando questo pattern è stato possibile suddividere la parte funzionale dell'applicazione dalla sua interfaccia: in questo modo è possibile ad esempio riconfigurare completamente l'interfaccia grafica senza dover modificare le parti funzionali del sistema.
- **Observer:** questo design pattern è stato utilizzato per gestire l'interazione tra l'interfaccia grafica e le azioni eseguite dall'utente. Questo è stato possibile tramite l'utilizzo di controller appositi che permettono di catturare le azioni che l'utente compie ed eseguire procedure specifiche associate a quest'ultime.

## 3.2 Gestione dei dati persistenti

Per la gestione dei dati persistenti è stato utilizzato PostgreSQL versione 13; durante i vari test effettuati per lo sviluppo dell'applicativo è stato utilizzato anche lo strumento PGAdmin per effettuare velocemente query sul database.

Di seguito è possibile vedere le tabelle di cui è composto il diagramma ER:



- **Bicicletta**: rappresenta una bicicletta tramite un campo id e tipo. Si è utilizzato un campo “danneggiata” per permettere agli utenti di segnalare eventuali danni e capire se ha bisogno di manutenzione. E’ in relazione uno a molti con la tabella noleggio, e uno a uno con la tabella morsa.
- **Morsa**: rappresenta una morsa contenuta in una rastrelliera tramite l’id, la posizione nella morsa e il suo tipo (normale o elettrica). E’ in relazione uno a uno con la tabella bicicletta, e molti a uno con la tabella totem.
- **Totem**: rappresenta una postazione con totem che contiene l’insieme di morse (e quindi la rastrelliera) tramite un id e il relativo indirizzo. E’ in relazione uno a molti con la tabella noleggio per poter poi prelevare alcune statistiche sull’utilizzo di un totem in particolare, e in relazione uno a molti con la tabella morsa.
- **Abbonamento**: rappresenta un abbonamento creato da un utente tramite principalmente un campo codice utente, password, e tipo dell’abbonamento. La password viene memorizzata tramite funzione di hashing in modo tale che eventuali amministratori del sistema non possano vedere dati sensibili

riguardanti l'utente e preservare quindi la privacy dei relativi dati. E' in relazione uno a molti con la tabella noleggio, e uno a uno con la tabella carta di credito.

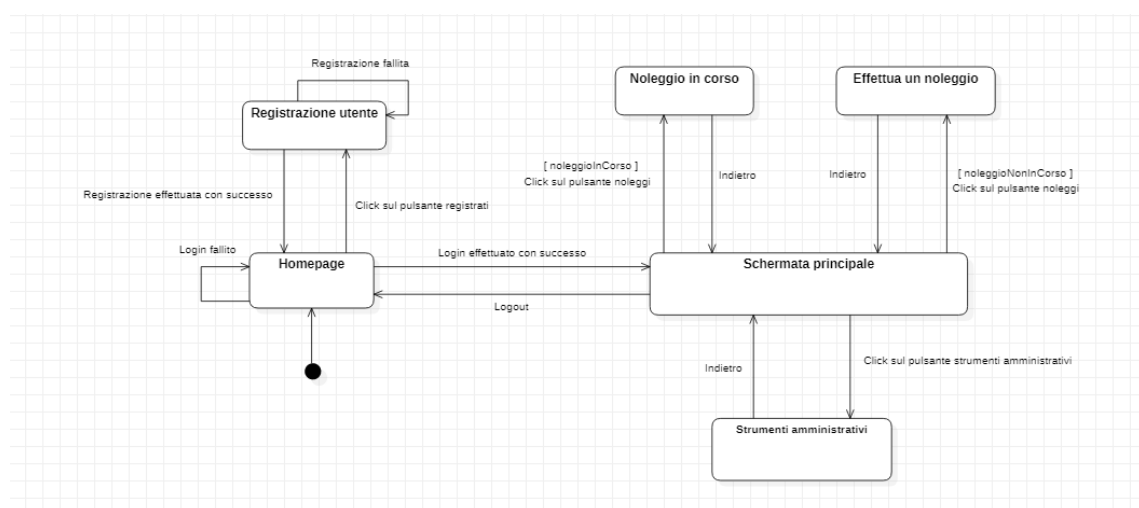
- **Carta di credito:** rappresenta una carta di credito di un utente tramite il numero, la data di scadenza, e il cvv. E' in relazione uno a uno con la tabella abbonamento.

- **Noleggio:** rappresenta le informazioni riguardanti il noleggio di una bicicletta tramite un id principale e l'orario di inizio e fine noleggio. E' in relazione molti a uno con la tabella abbonamento e molti a uno con la tabella bicicletta.

### 3.3 Descrizione dell'interfaccia grafica

L'interfaccia grafica è stata realizzata pensando all'interazione il più amichevole possibile con l'utente che utilizza il sistema. Grazie all'apposito tipo di abbonamento inoltre, il personale di servizio che ha bisogno di accedere agli strumenti di amministrazione potrà farlo come si suppone venga fatto nella maggior parte dei sistemi reali, quindi tramite un codice utente e password pre-impostati che permetteranno di accedere alla schermata per gli strumenti amministrativi. L'unica simulazione che è stata effettuata è legata all'accesso delle postazioni con totem fisicamente distinte.

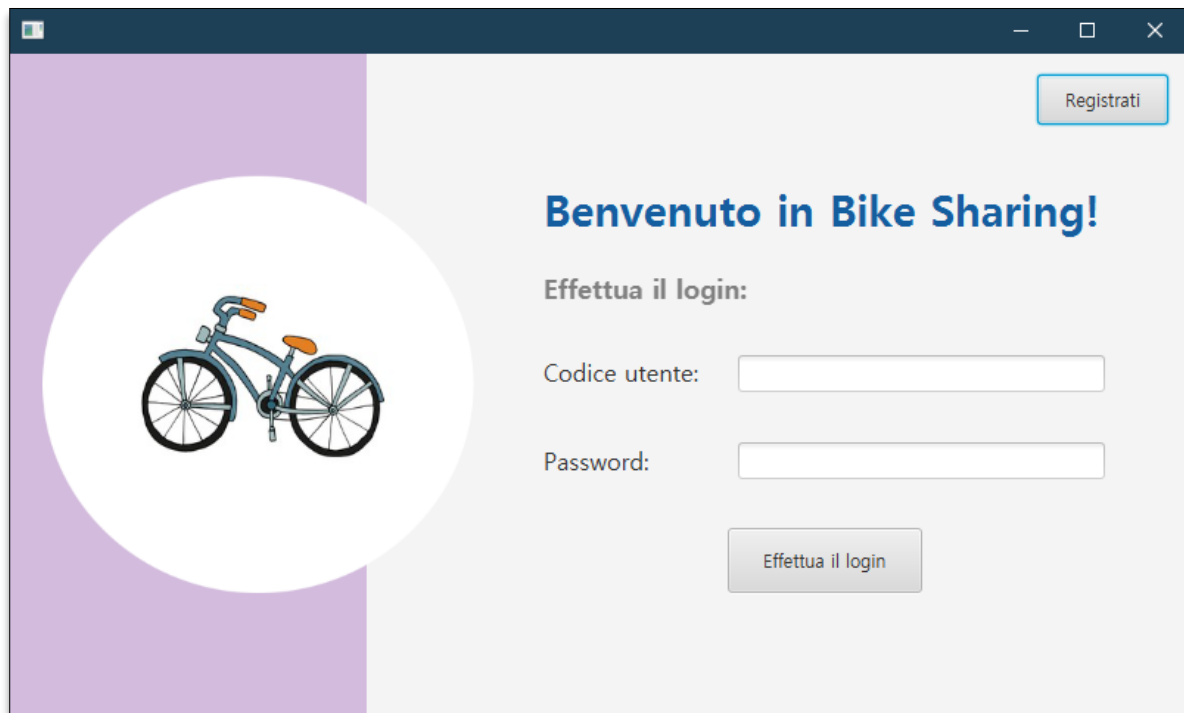
#### 3.3.1 Mappa di navigazione dell'interfaccia grafica



#### 3.3.2 Homepage

La homepage è la schermata iniziale del sistema di Bike Sharing: permette all'utente che si trova alla postazione con totem di effettuare il login se già registrato, oppure di effettuare la registrazione per un nuovo abbonamento.

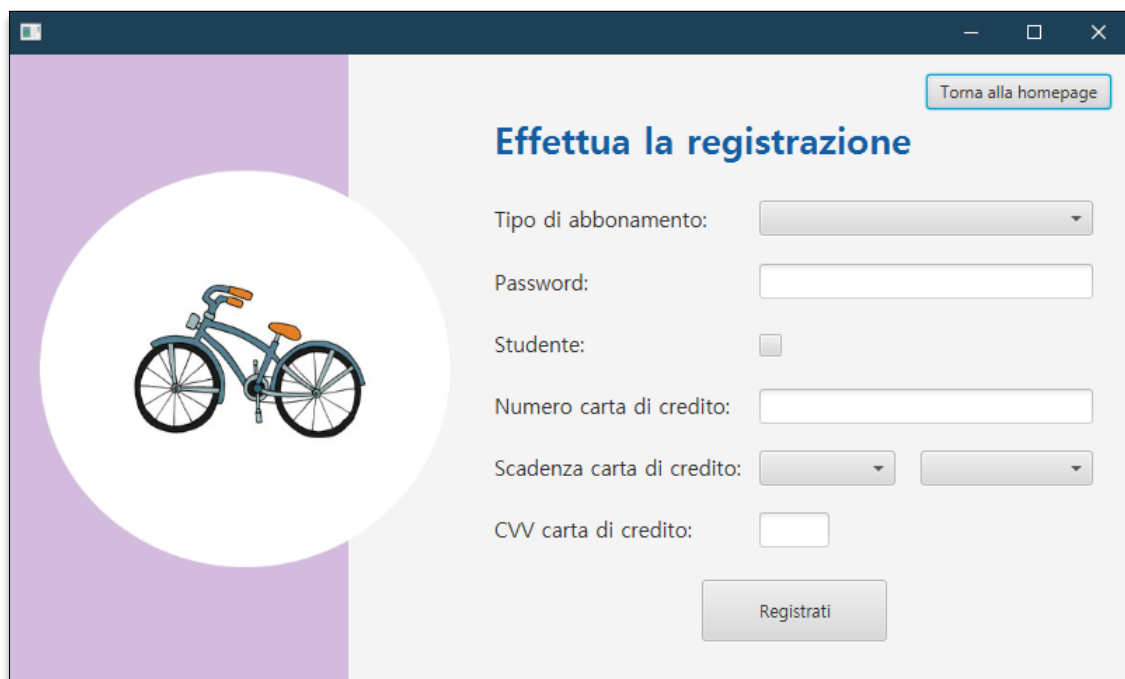
Nel caso di login con dati non corretti viene visualizzato un alert con i dettagli per permettere all'utente di correggere i dati inseriti.



The screenshot shows a web browser window with a login page for 'Bike Sharing'. On the left, there is a purple vertical bar and a white circle containing a blue bicycle icon. The main content area is light gray. At the top right, there is a 'Registrati' button. The heading 'Benvenuto in Bike Sharing!' is in blue. Below it, the text 'Effettua il login:' is in gray. There are two input fields: 'Codice utente:' and 'Password:'. Below these is a 'Effettua il login' button.

### 3.3.2 Registrazione di un nuovo abbonamento

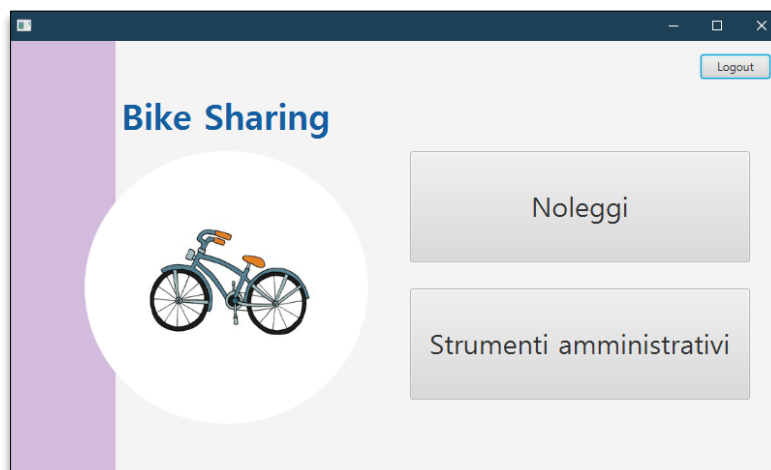
Questa schermata permette ad un utente di registrare un nuovo abbonamento. Nel caso di registrazione con dati non corretti viene visualizzato un alert con i dettagli per permettere all'utente di correggere i dati inseriti.



The screenshot shows a web browser window with a registration page for 'Bike Sharing'. On the left, there is a purple vertical bar and a white circle containing a blue bicycle icon. The main content area is light gray. At the top right, there is a 'Torna alla homepage' button. The heading 'Effettua la registrazione' is in blue. Below it, there are several input fields: 'Tipo di abbonamento:' (a dropdown menu), 'Password:', 'Studente:' (a checkbox), 'Numero carta di credito:', 'Scadenza carta di credito:' (two dropdown menus), and 'CVV carta di credito:'. At the bottom right is a 'Registrati' button.

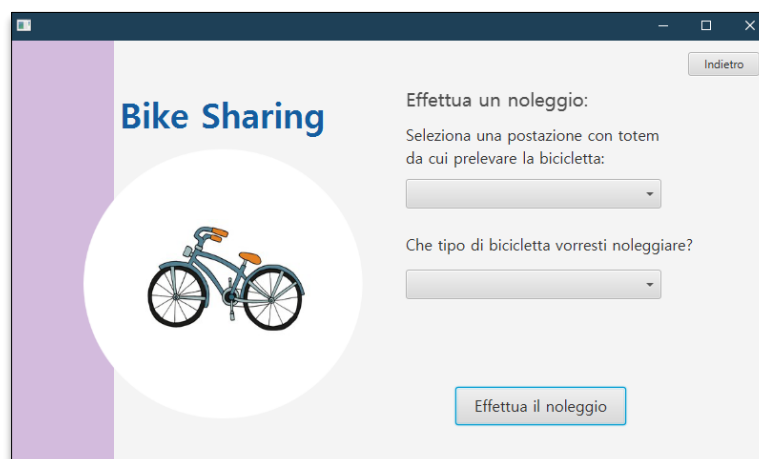
### 3.3.3 Schermata principale

Una volta effettuato il login, l'utente verrà indirizzato su questa schermata che è quella principale. Da qui è possibile accedere alla schermata riguardo i noleggi oppure agli strumenti amministrativi (nel caso il tipo di utente loggato sia del personale di servizio). Il bottone per raggiungere la schermata dei noleggi porterà alla schermata per effettuare un noleggio nel caso non ce ne sia già uno in corso, e alla schermata per terminare un noleggio nel caso ce ne sia uno già in corso.



### 3.3.4 Effettua un noleggio

Da qui l'utente può effettuare un noleggio selezionando la postazione con totem da cui prelevare la bicicletta, e il tipo di bicicletta che si intende noleggiare. Nel caso l'utente non selezioni le opzioni richieste e tenti di inizializzare il noleggio verrà visualizzato un apposito alert chiedendo di selezionare le varie opzioni richieste.



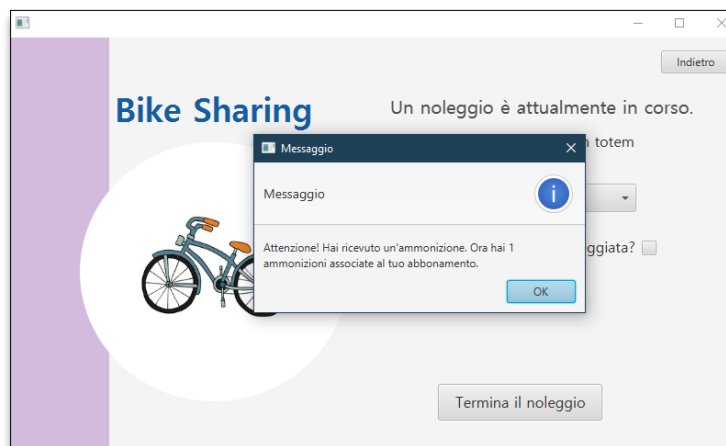


### 3.3.5 Termina un noleggio (Noleggio in corso)

Da qui l'utente può terminare un noleggio attualmente in corso, selezionando la postazione con totem dove restituire la bicicletta, ed eventualmente segnalando se la bicicletta è risultata danneggiata. Nel caso la bicicletta sia stata già segnalata come danneggiata in passato, l'opzione risulterà bloccata in quanto sarebbe ridondante segnalare nuovamente il danno. Anche in questo caso, se l'utente non seleziona la postazione con totem dove restituire la bicicletta e tenta di concludere il noleggio, verrà visualizzato un alert chiedendo di selezionare la postazione con totem.



Nel caso in cui l'utente restituisca la bicicletta dopo più di 2 ore verrà avvisato dell'avvenuta ammonizione e anche del numero complessivo di quest'ultime che sono associate al suo abbonamento. In caso della terza ammonizione, gli sarà permesso restituire la bicicletta ma verrà subito sospeso l'abbonamento, e di conseguenza l'utente verrà riportato alla schermata di homepage; se tenterà di rifeffettuare il login non gli verrà permesso e verrà visualizzato un alert apposito.



### 3.3.6 Strumenti amministrativi

Questa schermata serve appositamente agli utenti che fanno parte del personale del servizio per accedere ai vari strumenti di amministrazione del sistema, e visualizzare alcune statistiche riguardo quest'ultimo.

Le operazioni disponibili sono:

- Creazione di una nuova postazione con totem;
- Eliminazione di una postazione con totem;
- Creazione di una morsa in una postazione;
- Eliminazione di una morsa da una postazione;
- Inserimento di una bicicletta in una postazione;
- Eliminazione di una bicicletta da una postazione;
- Riparazione di una bicicletta risultata danneggiata.

I dati statistici disponibili sono:

- Abbonamenti (creati, attivi e sospesi);
- Noleggi effettuati e biciclette danneggiate;
- Totem più utilizzato, e numero di postazioni con totem.

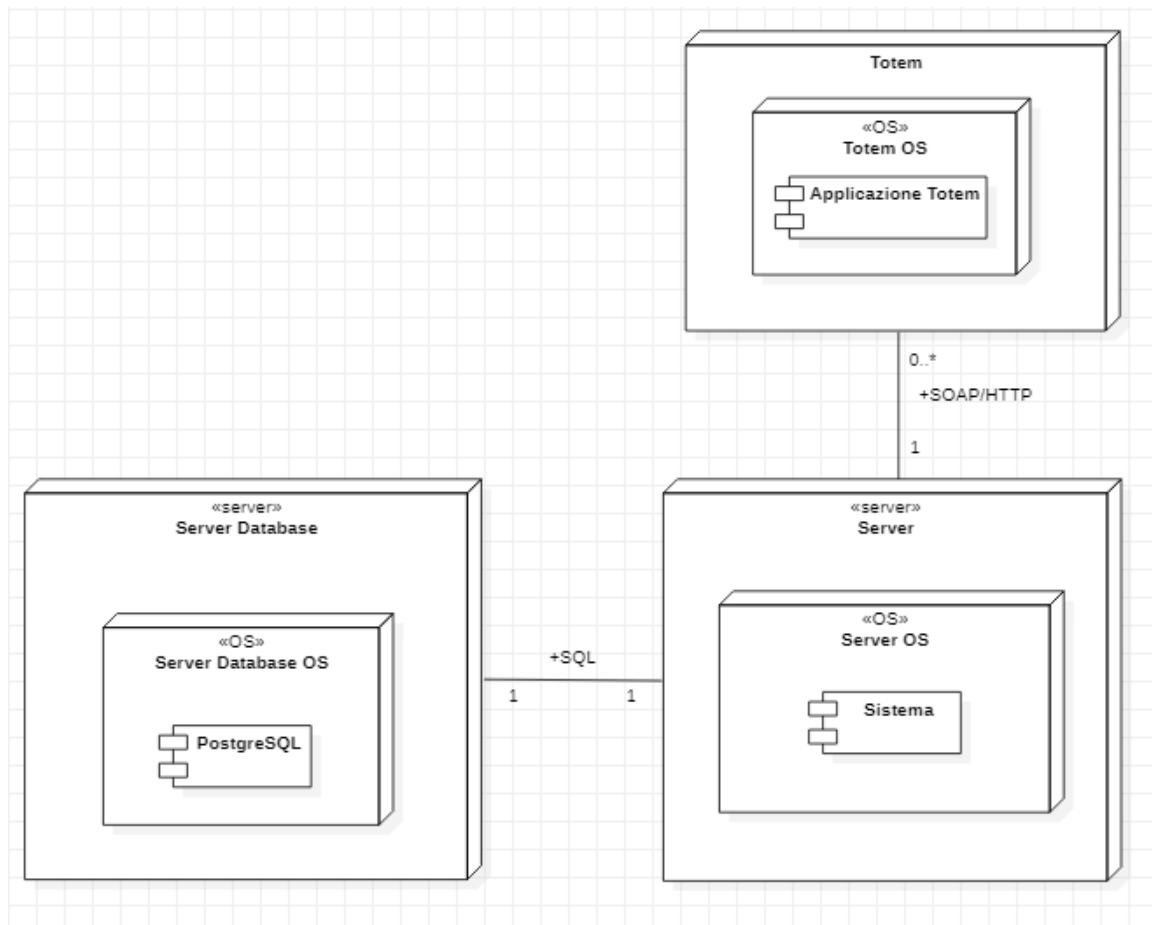
The screenshot displays a web application window titled "Bike Sharing". On the left is a purple sidebar with a white circle containing a bicycle icon. The main content area is titled "Strumenti amministrativi" and includes a list of actions with corresponding input fields and buttons. Below this is a section titled "Dati statistici" showing various metrics.

Strumenti amministrativi			
Crea una postazione con totem:	<input type="text" value="Inserisci un indirizzo"/>	<input type="button" value="Crea"/>	
Elimina una postazione con totem:	<input type="button" value="Seleziona un indirizzo"/>	<input type="button" value="Elimina"/>	
Crea una morsa in una postazione:	<input type="button" value="Indirizzo"/>	<input type="button" value="Tipo"/>	<input type="button" value="Crea"/>
Elimina una morsa da una postazione:	<input type="button" value="Indirizzo"/>	<input type="button" value="Tipo"/>	<input type="button" value="Elimina"/>
Inserisci una bicicletta in una postazione:	<input type="button" value="Indirizzo"/>	<input type="button" value="Tipo"/>	<input type="button" value="Crea"/>
Elimina una bicicletta da una postazione:	<input type="button" value="Indirizzo"/>	<input type="button" value="Tipo"/>	<input type="button" value="Elimina"/>
Ripara una bicicletta danneggiata:	<input type="button" value="Seleziona una bicicletta"/>	<input type="button" value="Ripara"/>	

Dati statistici			
Abbonamenti:	Creati: 8	Attivi: 7	Sospesi: 1
Noleggi effettuati:	13	Biciclette danneggiate:	3
Totem più utilizzato:	Via Denissonis 66		
Numero totem:	4		

### 3.4 Diagramma di deployment



### 3.5 Specifica e verifica dei vincoli

Di seguito è possibile osservare i vincoli OCL che sono stati individuati e la relativa mappatura in JML.

#### 3.6.1 Abbonamento

```

1) { context Abbonamento
inv: self.allInstances -> isUnique(codice)
inv: tipo = #GIORNALIERO or tipo = #SETTIMANALE or tipo = #ANNUALE
or tipo = #PERSONALE_SERVIZIO}
inv: ammonizioni >=0
}

```

In JML:

```

8 // *
9 * @invariant (\forallall Abbonamento abb1, abb2; \created(abb1) && \created(abb2); abb1 != abb2 ==> !(abb1.codice.equals(abb2.codice)));
10 * @invariant tipo == #ANNUALE || tipo == #SETTIMANALE || tipo == #GIORNALIERO || tipo == #PERSONALE_SERVIZIO;
11 * @invariant ammonizioni >=0
12 @*/

```

```
2) { context Abbonamento::aggiungiAmmonizione()
post: ammonizioni = ammonizioni@pre + 1
}
```

*In JML:*

```
95  /*@
96   * @ensures ammonizioni = \old(ammonizioni) + 1
97   @*/
```

```
3) { context Abbonamento::attiva()
pre: dataInizio = null
post: dataInizio = Now()
}
```

*In JML:*

```
89  /*@
90   * @requires dataInizio == null
91   * @ensures dataInizio = LocalDate.now();
92   @*/
```

### 3.6.2 Carta di credito

```
{ context CartaDiCredito
inv: numero.length = 16
inv: cvv.length = 3
}
```

*In JML:*

```
7  /*@
8   * @invariant numero.length == 16;
9   * @invariant cvv.length == 3;
10  @*/
```

### 3.6.3 Bicicletta

```
{ context Bicicletta
inv: self.allInstances -> isUnique(id)
}
```

*In JML:*

```
4  /*@
5   * @invariant (\forallall Bicicletta bici1, bici2; \created(bici1) && \created(bici2); bici1 != bici2 ==> !(bici1.id.equals(bici2.id)));
6   @*/
```

### 3.6.4 Totem

```
{ context Totem
inv: self.allInstances -> isUnique(id)
}
```

In JML:

```
7- /*@
8  * @invariant (\forallall Totem t1, t2; \created(t1) && \created(t2); t1 != t2 ==> !(t1.id.equals(t2.id)));
9  @*/
```

## 3.6 Descrizione del testing

Per il testing è stato scelto di utilizzare il branch-coverage come criterio di copertura, e sono state testate le classi principali del dominio, ovvero Abbonamento, Bicicletta, Morsa, Totem, Noleggio e CartaDiCredito.

Di seguito è possibile osservare alcuni esempi di test effettuati sulle classi precedentemente specificate:

### 3.6.1 Abbonamento

```
65  /* Questo test verifica che la data di scadenza venga calcolata in modo corretto per ogni tipo di abbonamento. */
66- @Test
67  public void calcolaScadenzaTest() {
68      CartaDiCredito c = new CartaDiCredito("4079029256061430", "9/2024", "242");
69
70      /* Abbonamento giornaliero */
71      Abbonamento abbGiornaliero = new Abbonamento("Test123", TipoAbbonamento.GIORNALIERO, c, true);
72      abbGiornaliero.attivaAbbonamento();
73      Assertions.assertEquals(abbGiornaliero.calcolaScadenzaAbbonamento(), LocalDate.now());
74
75      /* Abbonamento settimanale */
76      Abbonamento abbSettimanale = new Abbonamento("Test123", TipoAbbonamento.SETTIMANALE, c, true);
77      abbSettimanale.attivaAbbonamento();
78      Assertions.assertEquals(abbSettimanale.calcolaScadenzaAbbonamento(), LocalDate.now().plusDays(7));
79
80      /* Abbonamento annuale */
81      Abbonamento abbAnnuale = new Abbonamento("Test123", TipoAbbonamento.ANNUALE, c, true);
82      Assertions.assertEquals(abbAnnuale.calcolaScadenzaAbbonamento(), LocalDate.now().plusYears(1));
83
84      /* Abbonamento per il personale di servizio */
85      Abbonamento abbPersonaleServizio = new Abbonamento("Test123", TipoAbbonamento.PERSONALE_SERVIZIO, c, true);
86      Assertions.assertEquals(abbPersonaleServizio.calcolaScadenzaAbbonamento(), LocalDate.now().plusYears(3));
87
88  }
```

### 3.6.2 Bicicletta

```
10  /* Questo test verifica che una bicicletta segnalata come danneggiata lo diventi effettivamente */
11- @Test
12  public void danneggiaBicicletta() {
13      Bicicletta bici = new Bicicletta("TestID", TipoBicicletta.NORMALE, false);
14      bici.setDanneggiata();
15      Assertions.assertEquals(bici.isDanneggiata(), true);
16  }
```

### 3.6.3 Morsa

```
19  /* Questo test verifica che non venga sollevata un'eccezione quando si tenta di aprire una morsa con una bicicletta disponibile */
20- @Test
21  public void apriMorsaConBiciclettaTest() {
22      Morsa morsa = new Morsa(TipoMorsa.NORMALE);
23      Bicicletta bicicletta = new Bicicletta(TipoBicicletta.NORMALE);
24      morsa.chiudi(bicicletta);
25      Assertions.assertDoesNotThrow(() -> { morsa.apri(); });
26  }
```

### 3.6.4 Totem

```

15  /* Questo test verifica che non vengano sollevate eccezioni nel caso si noleggi una bicicletta che si sa essere disponibile */
16  @Test
17  public void noleggioBiciclettaTest() {
18      Totem totem = new Totem("Via Denissonis 66");
19      Bicicletta bici = new Bicicletta(TipoBicicletta.ELETTRICA);
20      Morsa morsa = new Morsa(TipoMorsa.ELETTRICA);
21      morsa.chiudi(bici);
22      totem.aggiungiMorsa(morsa);
23      Assertions.assertDoesNotThrow(() -> { totem.noleggiaBicicletta(TipoBicicletta.ELETTRICA); });
24  }

```

### 3.6.5 Noleggio

```

15  /* Questo test verifica che venga sollevata un'eccezione IllegalStateException nel caso si tenti di calcolare la durata di un noleggio non ancora terminato */
16  @Test
17  public void getDurataNoleggioTest() {
18      Bicicletta bici = new Bicicletta("TestID", TipoBicicletta.NORMALE, false);
19      CartaDiCredito c = new CartaDiCredito("4079029256061430", "9/2024", "242");
20      Abbonamento abb = new Abbonamento("Test123", TipoAbbonamento.SETTIMANALE, c, true);
21      Totem totem = new Totem("Via Denissonis 66");
22      Noleggio noleggio = new Noleggio(abb, bici, totem);
23
24      Assertions.assertThrows(IllegalStateException.class, () -> { noleggio.getDurataNoleggio(); });
25  }

```

### 3.6.6 Carta di credito

```

33  /* Questo test verifica che venga sollevata un'eccezione IllegalArgumentException nel caso si tenti di creare una carta di credito con cvv non valido. */
34  @Test
35  public void creaCartaCvvNonValidoTest() {
36      Assertions.assertThrows(IllegalArgumentException.class, () -> { new CartaDiCredito("3241251253254", "9.2023", "1"); });
37  }

```

## 3.7 Note per l'installazione e l'utilizzo

Per lo sviluppo del sistema di Bike Sharing è stato utilizzato Eclipse con la versione 2021-06 (4.20.0); è stato inoltre usato JavaFX con la versione 16.

Si raccomanda di esportare l'archivio in una folder a parte ma non cambiare la posizione della folder "images" e del progetto contenuto nella folder "Bike Sharing", altrimenti le immagini utilizzate nell'interfaccia grafica non verranno visualizzate correttamente. Per utilizzare il database nell'archivio è stato allegato un dump di esso, e i dati per effettuare il login sono i seguenti:

*Nome utente: "postgres", password: "postgres".*

Il dump è stato effettuato con il seguente comando:

```
pg_dump -U postgres bike_sharing > dump.sql
```

Nel caso si vogliano effettuare delle query sul database, si ricorda che è necessario impostare lo schema attuale tramite il comando:

```
set search_path = bike_sharing;
```



I dati degli abbonamenti per testare le varie funzionalità del sistema sono i seguenti:

Codice utente	Password	Tipo abbonamento	Sospeso	Studente
18597456-7524-4318-b4d5-5603612dfbab	Test123456	GIORNALIERO	F	T
18e0afb4-554e-4c20-b575-d17752c412d9	Test12345	SETTIMANALE	F	T
482e875d-8814-4e86-9fff-1c07ea4e33cc	Dennis123	ANNUALE	F	F
4bcd3928-f425-40db-bd31-c3bd2186be5d	Test123	ANNUALE	T	F
53dd1104-d045-4d4f-a7fa-946b4714e1bc	Dennis12345	PERSONALE_SERVIZIO	F	F

Nel caso di abbonamento di tipo “personale\_servizio” sarà possibile accedere agli strumenti amministrativi del sistema; mentre nel caso di abbonamento sospeso non sarà possibile effettuare il login.

Nel caso vengano testati abbonamenti giornalieri oppure settimanali si tenga conto che non sono ancora stati attivati (ovvero il primo noleggio non è stato ancora effettuato), quindi potranno essere utilizzati senza problemi.

Dato che si è utilizzato un algoritmo per il controllo della validità del numero della carta di credito, di seguito sono presenti alcuni codici che potranno essere utilizzati nel caso si voglia registrare un nuovo abbonamento:

4312269870457344 , 4297140149469163 , 4239775665067312

### 3.7.1 Possibili miglioramenti

Per generare tutti gli ID nel sistema è stata utilizzata una libreria di Java chiamata “UUID” (pacchetto java.util), pensando ad un sistema piuttosto grande che avesse bisogno di generare sempre id univoci: per l'utilizzo dei codici degli utenti però è leggermente impraticabile in un sistema reale dover utilizzare dei codici così lunghi per effettuare il login su delle postazioni con totem. Per questo, un possibile miglioramento potrebbe essere fatto implementando un generatore custom di id per generare dei codici più brevi in modo tale da essere ricordati più facilmente ed essere immessi più velocemente; sarebbe piuttosto importante però accertarsi che rimanga sempre l'univocità degli id generati.

Un altro miglioramento potrebbe essere quello di utilizzare una funzione di hash per tutti i dati sensibili che vengono memorizzati sul database, come ad esempio le password oppure i dati della carta di credito, in modo tale che gli amministratori del sistema non siano in grado di vedere i dati effettivi riguardanti gli utenti.