

BIKE SHARING

1 Descrizione del problema

1.1 Analisi e specifica dei requisiti

Il programma gestisce un sistema di noleggio di biciclette. Per noleggiare una bicicletta è necessario abbonarsi. Sono previste varie tipologie di abbonamento, con relativi costi:

Abbonamento	Costo
Annuale	€ 36.00
Settimanale	€ 9.00
Giornaliero	€ 4.50

Ogni utente può avere solo un abbonamento attivo in un dato momento (si può avere un solo abbonamento per carta di credito, visto che è ragionevole pensare che ogni persona non ha più di una carta di credito).

Al momento della registrazione di un abbonamento, l'utente seleziona una password e gli viene restituito un codice univoco che sarà valido per tutta la durata dell'abbonamento.

Nel caso di abbonamenti occasionali (giornalieri e settimanali), l'abbonamento ha inizio a partire dal primo prelievo di bicicletta, altrimenti ha inizio immediatamente.

L'unica forma di pagamento prevista è tramite carta di credito (deve essere valida fino al termine dell'abbonamento desiderato).

Sono presenti diverse tipologie di biciclette, ciascuna con la relativa tariffa:

Frazione oraria	Bicicletta tradizionale	E-Bike	E-Bike con seggiolino
Prima mezz'ora	€ 0.00	€ 0.25	€ 0.25
Seconda mezz'ora	€ 0.50	€ 0.50	€ 0.50
Terza mezz'ora	€ 0.50	€ 1.00	€ 1.00
Quarta mezz'ora	€ 0.50	€ 2.00	€ 2.00
Ogni ora o frazione di ora successiva	€ 2.00	€ 4.00	€ 4.00

Sono inoltre previste tariffe agevolate per gli studenti:

Frazione oraria	Bicicletta tradizionale	E-Bike	E-Bike con seggiolino
Prima mezz'ora	€ 0.00	€ 0.25	€ 0.25
Seconda mezz'ora	€ 0.00	€ 0.50	€ 0.50
Terza mezz'ora	€ 0.00	€ 1.00	€ 1.00
Quarta mezz'ora	€ 0.00	€ 2.00	€ 2.00
Ogni ora o frazione di ora successiva	€ 2.00	€ 4.00	€ 4.00

Le biciclette sono disposte su apposite rastrelliere dotate di morse (differenti per ogni tipo di bicicletta).

Per ritirare una bicicletta, l'utente inserisce il proprio codice e la password sul totem vicino a una rastrelliera e sceglie il tipo di bicicletta desiderata. Il sistema indicherà il numero della morsa da cui prelevare la bicicletta e la sbloccherà.

Per restituire la bicicletta, l'utente deve inserirla in una morsa libera (e adatta al tipo di bicicletta). L'utente può verificare la corretta riconsegna digitando il codice e la password sul totem: esso confermerà con un messaggio l'avvenuta restituzione.

Il tempo massimo di uso di una bicicletta è di 2 ore.

Per ritirare una bicicletta devono essere passati almeno 5 minuti dall'ultimo utilizzo.

Superare per 3 volte la soglia di 2 ore o non riportare la bicicletta entro 24 ore comporta l'annullamento dell'abbonamento.

Se un utente non riporta la bicicletta noleggiata entro 24 ore dall'ultimo ritiro, verrà addebitata una penale di € 150.00, oltre ai supplementi.

In caso di danni alla bicicletta, l'utente dovrà comunicarlo al momento della riconsegna.

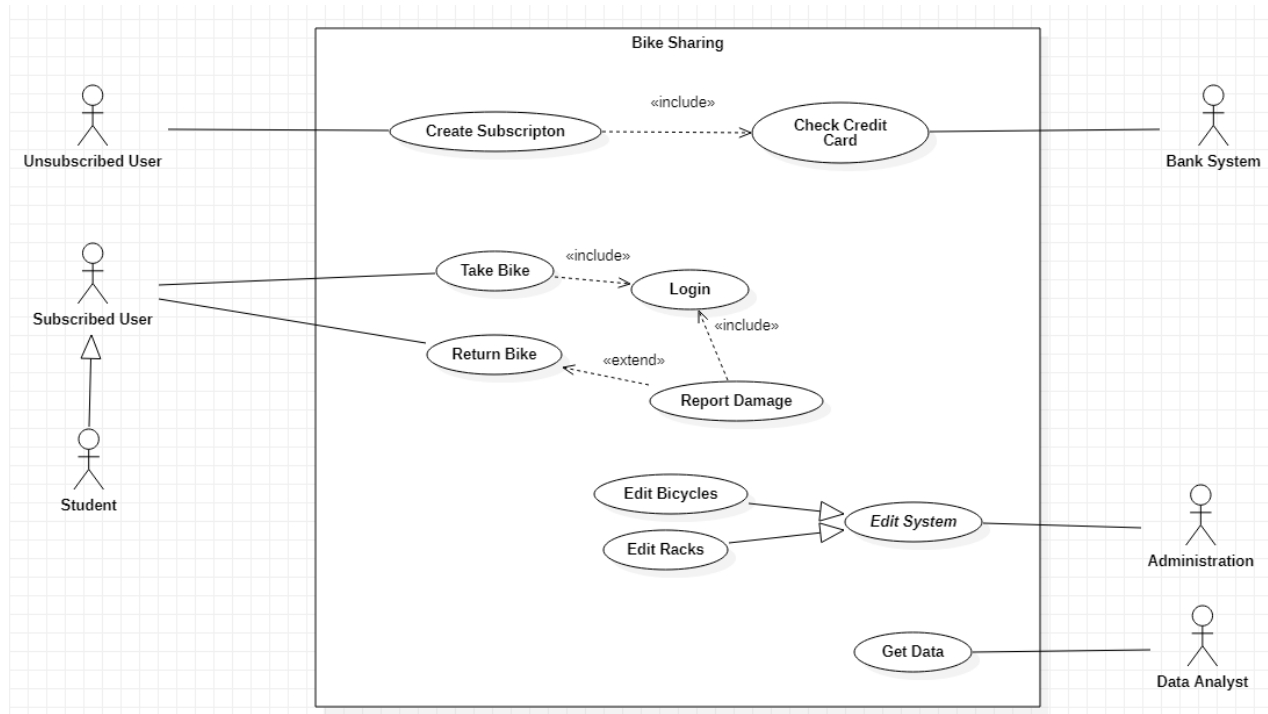
L'amministrazione del comune può aggiungere e rimuovere rastrelliere e biciclette.

Può inoltre, se necessario, spostare le biciclette da una rastrelliera ad un'altra con morse disponibili.

Infine, il sistema è in grado di fornire dati sull'utilizzo a personale autorizzato. In particolare verranno forniti i log degli abbonamenti (senza informazioni sulla carta di credito per ragioni di privacy) e degli affitti. Non verranno forniti dati aggregati per permettere ai data analyst la massima flessibilità.

2 Progettazione del Sistema

2.1 Diagramma dei casi d'uso



Caso d'uso	Create Subscription
Descrizione	Per abbonarsi l'utente deve interagire con l'applicazione selezionando il tipo di abbonamento e fornendo i dati della carta di credito, che verranno confermati dal caso d'uso "Check Credit Card"
Attore	Unsubscribed User
Goals	Abbonarsi
Precondizioni	La carta di credito non deve aver collegato un altro abbonamento

Caso d'uso	Check Credit Card
Descrizione	Controlla che la carta di credito inserita non abbia già un abbonamento collegato e che sia valida fino al termine dell'abbonamento selezionato
Attore	Unsubscribed User, Bank System
Goals	Controllare che la carta di credito sia valida e non collegata a un abbonamento esistente
Precondizioni	L'utente deve aver inserito le informazioni della carta di credito e selezionato il tipo di abbonamento

Caso d'uso	Take Bike
Descrizione	L'utente interagisce con il totem e, una volta eseguito il login (caso d'uso "Login") e verificato che è abilitato a ritirare una bicicletta (esempio: ha già noleggiato una bicicletta), il sistema sblocca una bicicletta e lo comunica all'utente, così che possa ritirarla
Attore	Subscribed User, Student
Goals	Noleggiare una bicicletta
Precondizioni	L'utente deve essere in possesso di un abbonamento e non aver noleggiato altre biciclette

Caso d'uso	Return Bike
Descrizione	L'utente inserisce la bicicletta in una morsa libera adatta per il tipo di bicicletta. Se lo desidera può interagire con il totem e autenticarsi per assicurarsi che la restituzione sia avvenuta correttamente
Attore	Subscribed User, Student
Goals	Restituire una bicicletta noleggiata
Precondizioni	L'utente deve aver noleggiato una bicicletta

Caso d'uso	Login
Descrizione	Controlla che i dati inseriti dall'utente corrispondano ad un abbonamento valido
Attore	Subscribed User, Student
Goals	Verificare la validità dei dati inseriti dall'utente
Precondizioni	L'utente deve avere un abbonamento e aver iniziato il processo per noleggiare, prenotare o dichiarare danni a una bicicletta

Caso d'uso	Report Damage
Descrizione	Dopo aver riconsegnato la bicicletta, l'utente può interagire con il totem e, dopo essersi autenticato e aver confermato la restituzione, può comunicare al sistema che la bicicletta ha subito danni
Attore	Subscribed User, Student
Goals	Comunicare al sistema che la bicicletta noleggiata ha subito dati
Precondizioni	L'utente deve aver noleggiato e restituito una bicicletta

Caso d'uso	Edit Bicycles
Descrizione	L'amministrazione del comune può aggiungere una bicicletta su una morsa libera, rimuoverne una parcheggiata, muoverne una verso una morsa libera e gestire la manutenzione delle biciclette danneggiate tramite l'applicazione di gestione
Attore	Administration
Goals	Gestire le biciclette presenti nel sistema
Precondizioni	Esiste almeno una rastrelliera (rack) su cui operare.

Caso d'uso	Edit Racks
Descrizione	L'amministrazione del comune può creare e rimuovere rastrelliere e modificare quelle esistenti aggiungendo e rimuovendo (purché libere) morse tramite l'applicazione di gestione
Attore	Administration
Goals	Gestire le rastrelliere
Precondizioni	//

Caso d'uso	Get Data
Descrizione	Gli analisti possono richiedere i log degli abbonamenti e degli affitti interagendo con l'applicazione di gestione
Attore	Data Analyst
Goals	Produrre dati statistici
Precondizioni	//

2.2 Descrizione degli scenari

Nome	Create Subscription								
Scopo	Abbonarsi								
Attore/i	Unsubscribed User								
Pre-condizioni	La carta di credito non deve aver collegato un altro abbonamento								
Trigger	L'utente inizia il processo di abbonamento tramite l'applicazione								
Descrizione sequenza eventi	<table border="1"> <thead> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> </thead> <tbody> <tr> <td>1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito</td><td></td></tr> <tr> <td></td><td>2. Il SISTEMA conferma la validità della carta di credito e registra l'abbonamento (con la relativa scadenza) generando un codice univoco</td></tr> <tr> <td></td><td>3. Il SISTEMA comunica all'utente la corretta riuscita dell'operazione e restituisce all'utente il codice univoco dell'abbonamento</td></tr> </tbody> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito			2. Il SISTEMA conferma la validità della carta di credito e registra l'abbonamento (con la relativa scadenza) generando un codice univoco		3. Il SISTEMA comunica all'utente la corretta riuscita dell'operazione e restituisce all'utente il codice univoco dell'abbonamento
Azioni attore	Risposte del sistema								
1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito									
	2. Il SISTEMA conferma la validità della carta di credito e registra l'abbonamento (con la relativa scadenza) generando un codice univoco								
	3. Il SISTEMA comunica all'utente la corretta riuscita dell'operazione e restituisce all'utente il codice univoco dell'abbonamento								
Alternativa/e	<p>Scenario alternativo 1: la carta di credito non è valida fino al termine dell'abbonamento selezionato o i dati inseriti non corrispondono a una carta di credito</p> <table border="1"> <thead> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> </thead> <tbody> <tr> <td>1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito</td><td></td></tr> <tr> <td></td><td>2. Il SISTEMA non conferma la validità della carta di credito e termina lo scenario</td></tr> </tbody> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito			2. Il SISTEMA non conferma la validità della carta di credito e termina lo scenario		
Azioni attore	Risposte del sistema								
1. L'UTENTE inizia il processo di abbonamento specificandone il tipo richiesto (giornaliero, etc.), la password e i dati della carta di credito									
	2. Il SISTEMA non conferma la validità della carta di credito e termina lo scenario								
Post-condizioni	Il sistema ha salvato in memoria l'abbonamento e l'utente ha la coppia "codice abbonamento" e password								

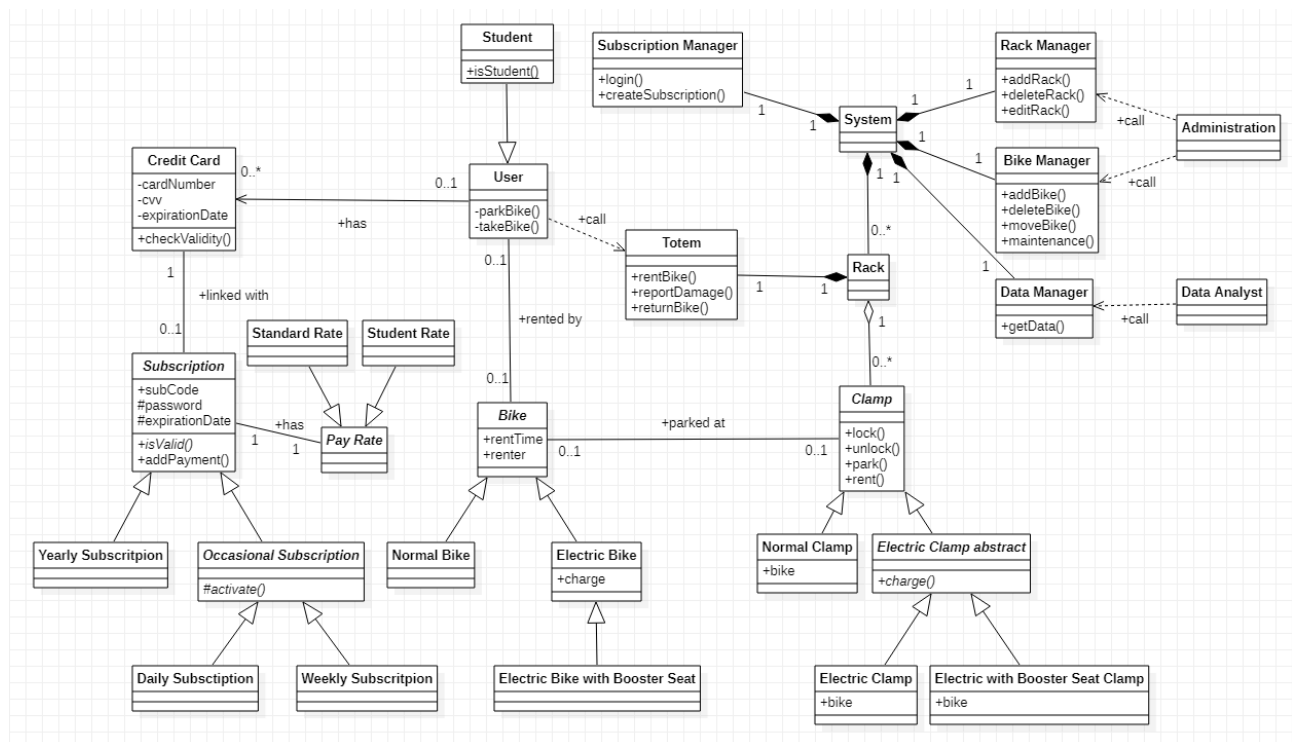
Nome	Take Bike										
Scopo	Noleggiare una bicicletta										
Attore/i	Subscribed User, Student										
Pre-condizioni	L'utente deve essere in possesso di un abbonamento e non aver noleggiato altre biciclette										
Trigger	L'utente inizia l'interazione con il totem e inizia il processo di noleggio										
Descrizione sequenza eventi	<table border="1"> <thead> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> </thead> <tbody> <tr> <td>1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata</td><td></td></tr> <tr> <td></td><td>2. Il SISTEMA conferma che l'abbonamento dell'UTENTE è valido e che non ha altre biciclette noleggiate</td></tr> <tr> <td></td><td>3. Il SISTEMA conferma che sulla rastrelliera è presente una bicicletta del tipo richiesto e la sblocca</td></tr> <tr> <td></td><td>4. Il SISTEMA restituisce all'utente il numero della morsa sbloccata per permettergli di ritirarla</td></tr> </tbody> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata			2. Il SISTEMA conferma che l'abbonamento dell'UTENTE è valido e che non ha altre biciclette noleggiate		3. Il SISTEMA conferma che sulla rastrelliera è presente una bicicletta del tipo richiesto e la sblocca		4. Il SISTEMA restituisce all'utente il numero della morsa sbloccata per permettergli di ritirarla
Azioni attore	Risposte del sistema										
1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata											
	2. Il SISTEMA conferma che l'abbonamento dell'UTENTE è valido e che non ha altre biciclette noleggiate										
	3. Il SISTEMA conferma che sulla rastrelliera è presente una bicicletta del tipo richiesto e la sblocca										
	4. Il SISTEMA restituisce all'utente il numero della morsa sbloccata per permettergli di ritirarla										
Alternativa/e	<p>Scenario alternativo 1: l'abbonamento fornito dall'utente non è valido o l'utente sta già noleggiando una bici</p> <table border="1"> <thead> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> </thead> <tbody> <tr> <td>1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata</td><td></td></tr> <tr> <td></td><td>2. Il SISTEMA non conferma che l'abbonamento è valido e termina lo scenario</td></tr> </tbody> </table> <p>Scenario alternativo 2: non è presente una bicicletta del tipo richiesto sulla rastrelliera</p> <table border="1"> <thead> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> </thead> <tbody> <tr> <td>1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento,</td><td></td></tr> </tbody> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata			2. Il SISTEMA non conferma che l'abbonamento è valido e termina lo scenario	Azioni attore	Risposte del sistema	1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento,	
Azioni attore	Risposte del sistema										
1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento, la password e il tipo di bicicletta desiderata											
	2. Il SISTEMA non conferma che l'abbonamento è valido e termina lo scenario										
Azioni attore	Risposte del sistema										
1. L'UTENTE inizia il processo di noleggio per una bicicletta fornendo il codice abbonamento,											

	la password e il tipo di bicicletta desiderata
	2. Il SISTEMA conferma che l'abbonamento dell'UTENTE è valido e che non ha altre biciclette noleggiate
	3. Il SISTEMA conferma che sulla rastrelliera non è presente una bicicletta del tipo richiesto, lo comunica all'UTENTE e termina lo scenario
Post-condizioni	Il sistema registra il noleggio e l'utente ritira la bicicletta parcheggiata presso la morsa sbloccata dal sistema

Nome	Return Bike										
Scopo	Restituire una bicicletta noleggiata										
Attore/i	Subscribed User, Student										
Pre-condizioni	L'utente deve aver noleggiato una bicicletta										
Trigger	L'utente arriva a destinazione										
Descrizione sequenza eventi	<table> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> <tr> <td>1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta</td><td></td></tr> <tr> <td>2. L'UTENTE inserisce la bicicletta nella morsa che si blocca</td><td></td></tr> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta		2. L'UTENTE inserisce la bicicletta nella morsa che si blocca					
Azioni attore	Risposte del sistema										
1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta											
2. L'UTENTE inserisce la bicicletta nella morsa che si blocca											
Alternativa/e	<p>Scenario alternativo 1: l'utente vuole assicurarsi che la consegna sia andata a buon fine</p> <table> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> <tr> <td>1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta</td><td></td></tr> <tr> <td>2. L'UTENTE inserisce la bicicletta nella morsa che si blocca</td><td></td></tr> <tr> <td>3. L'UTENTE inserisce il codice abbonamento e la password</td><td></td></tr> <tr> <td></td><td>4. Il SISTEMA conferma all'UTENTE che la restituzione è andata a buon fine</td></tr> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta		2. L'UTENTE inserisce la bicicletta nella morsa che si blocca		3. L'UTENTE inserisce il codice abbonamento e la password			4. Il SISTEMA conferma all'UTENTE che la restituzione è andata a buon fine
Azioni attore	Risposte del sistema										
1. L'UTENTE approccia la rastrelliera e individua una morsa adatta per parcheggiare la sua bicicletta											
2. L'UTENTE inserisce la bicicletta nella morsa che si blocca											
3. L'UTENTE inserisce il codice abbonamento e la password											
	4. Il SISTEMA conferma all'UTENTE che la restituzione è andata a buon fine										
Post-condizioni	Il sistema ha registrato la restituzione della bicicletta presa in prestito										

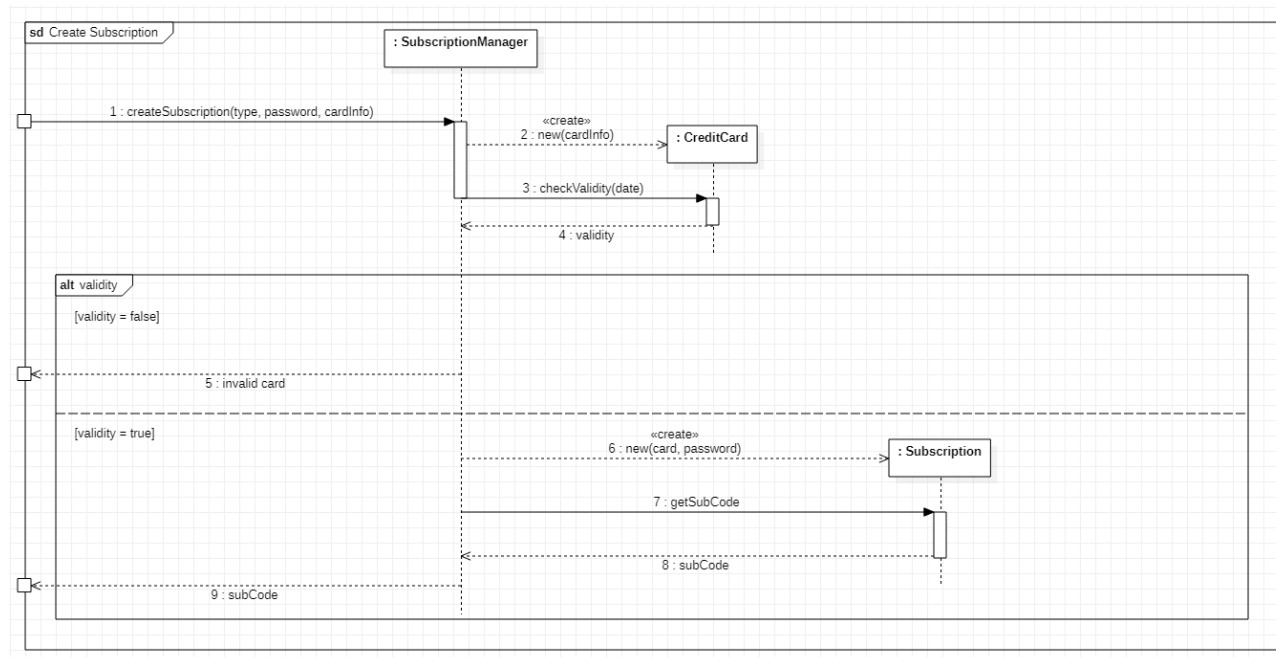
Nome	Report Damage						
Scopo	Comunicare al sistema che la bicicletta noleggiata ha subito dati						
Attore/i	Subscribed User, Student						
Pre-condizioni	L'utente deve aver noleggiato e restituito una bicicletta						
Trigger	L'utente inizia il processo di "report damage"						
Descrizione sequenza eventi	<table> <tr> <th>Azioni attore</th><th>Risposte del sistema</th></tr> <tr> <td>1. L'UTENTE riconsegna la bicicletta e interagisce con il totem per comunicare un danno alla bicicletta</td><td></td></tr> <tr> <td></td><td>2. Il SISTEMA registra il danno alla bicicletta, la mette in modalità di manutenzione e ringrazia l'UTENTE della segnalazione</td></tr> </table>	Azioni attore	Risposte del sistema	1. L'UTENTE riconsegna la bicicletta e interagisce con il totem per comunicare un danno alla bicicletta			2. Il SISTEMA registra il danno alla bicicletta, la mette in modalità di manutenzione e ringrazia l'UTENTE della segnalazione
Azioni attore	Risposte del sistema						
1. L'UTENTE riconsegna la bicicletta e interagisce con il totem per comunicare un danno alla bicicletta							
	2. Il SISTEMA registra il danno alla bicicletta, la mette in modalità di manutenzione e ringrazia l'UTENTE della segnalazione						
Alternativa/e	//						
Post-condizioni	Il sistema ha registrato il danno alla bicicletta e l'ha messa in modalità manutenzione						

2.3 Diagramma delle classi (modello di progetto)

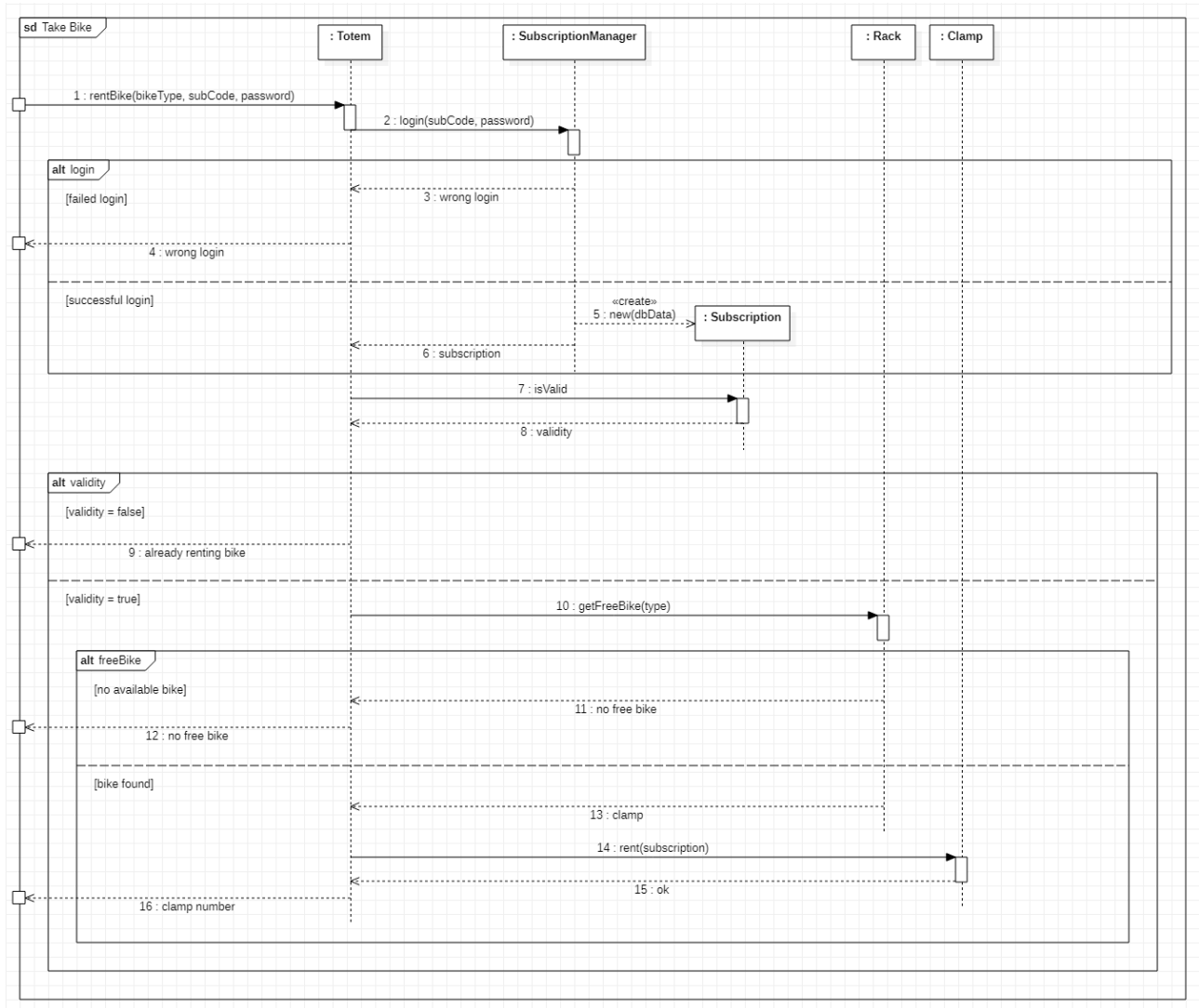


2.4 Diagrammi di sequenza

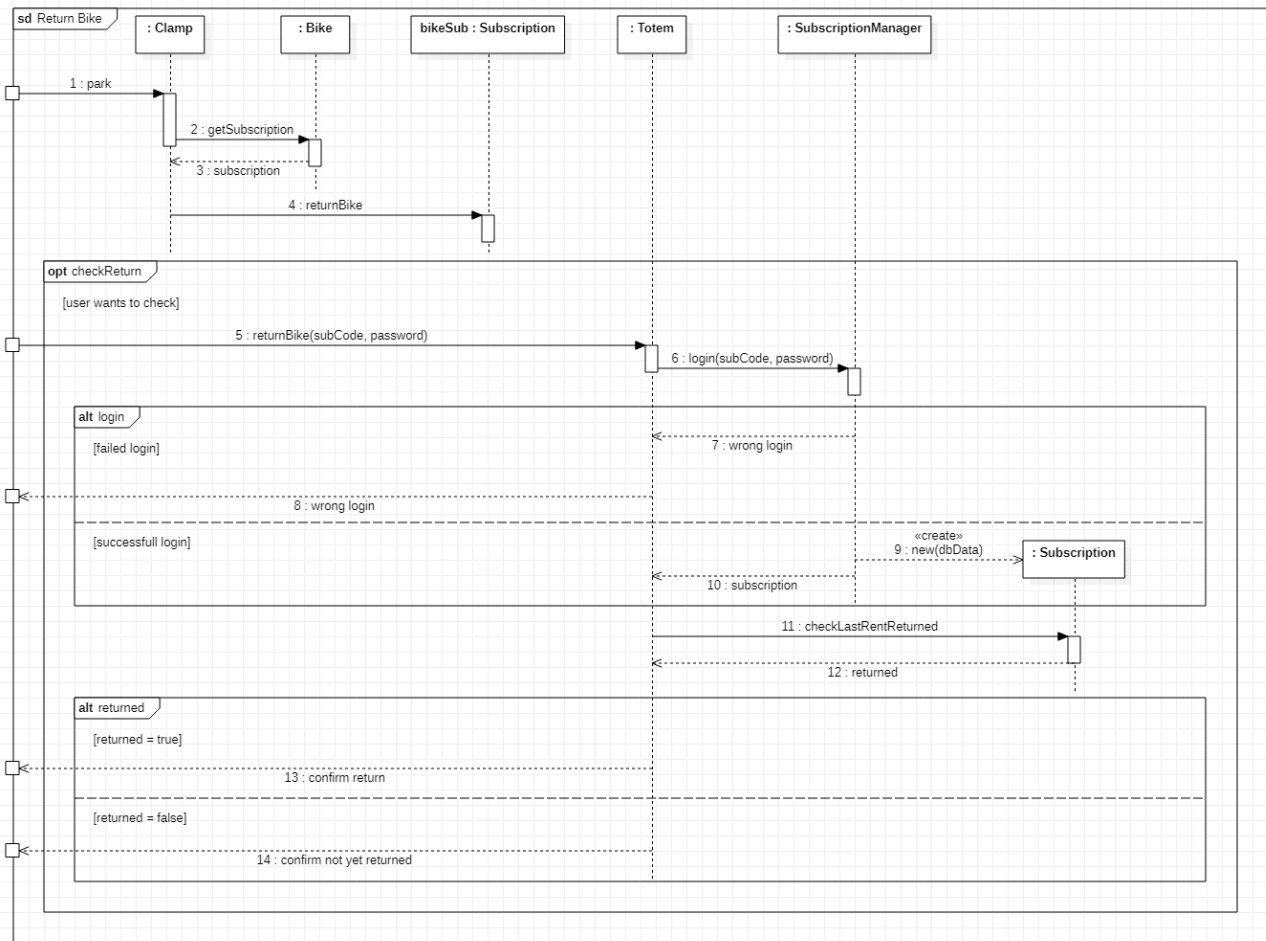
2.4.1 Create subscription



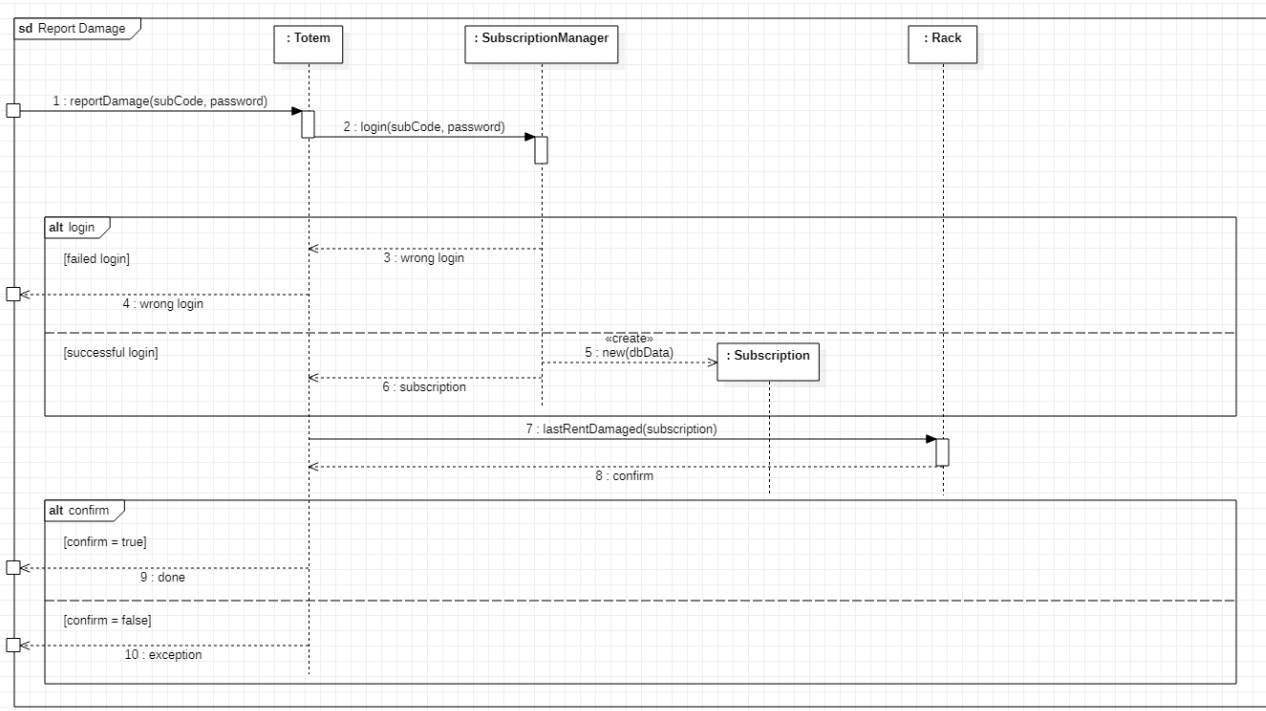
2.4.2 Take bike



2.4.3 Return bike

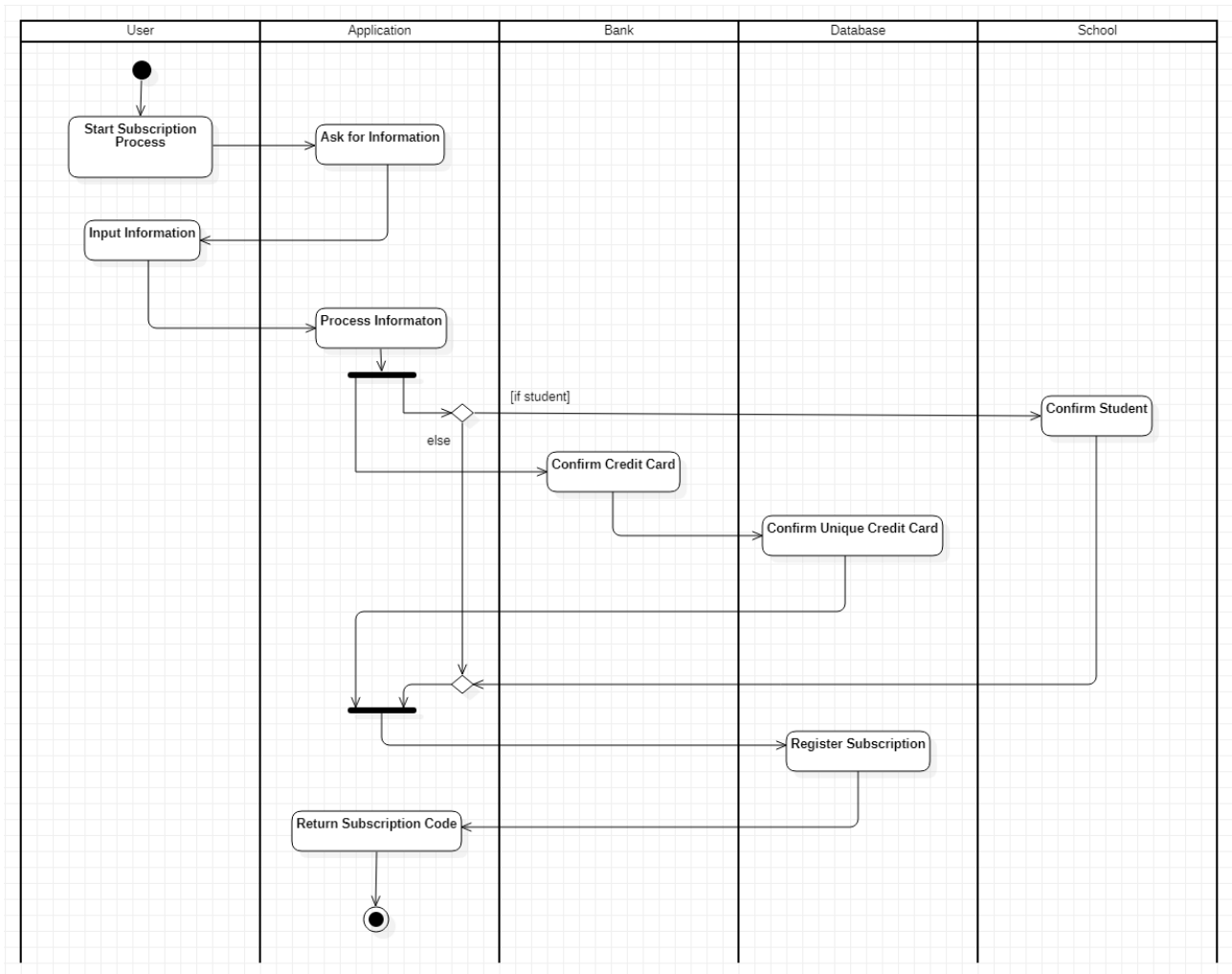


2.4.4 Report damage



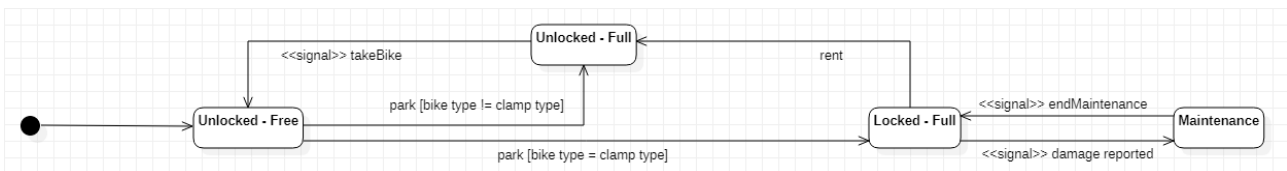
2.5 Diagrammi delle attività

2.5.1 Create subscription

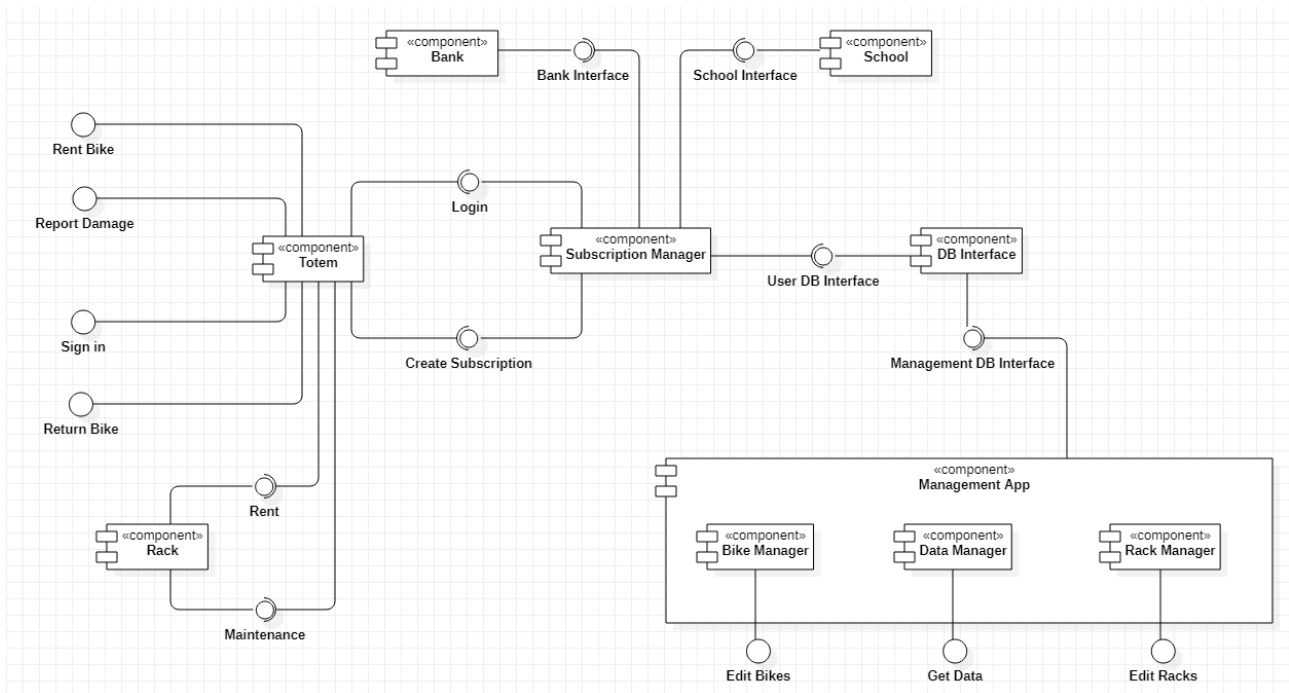


2.6 Macchine di stato

2.6.1 Clamp

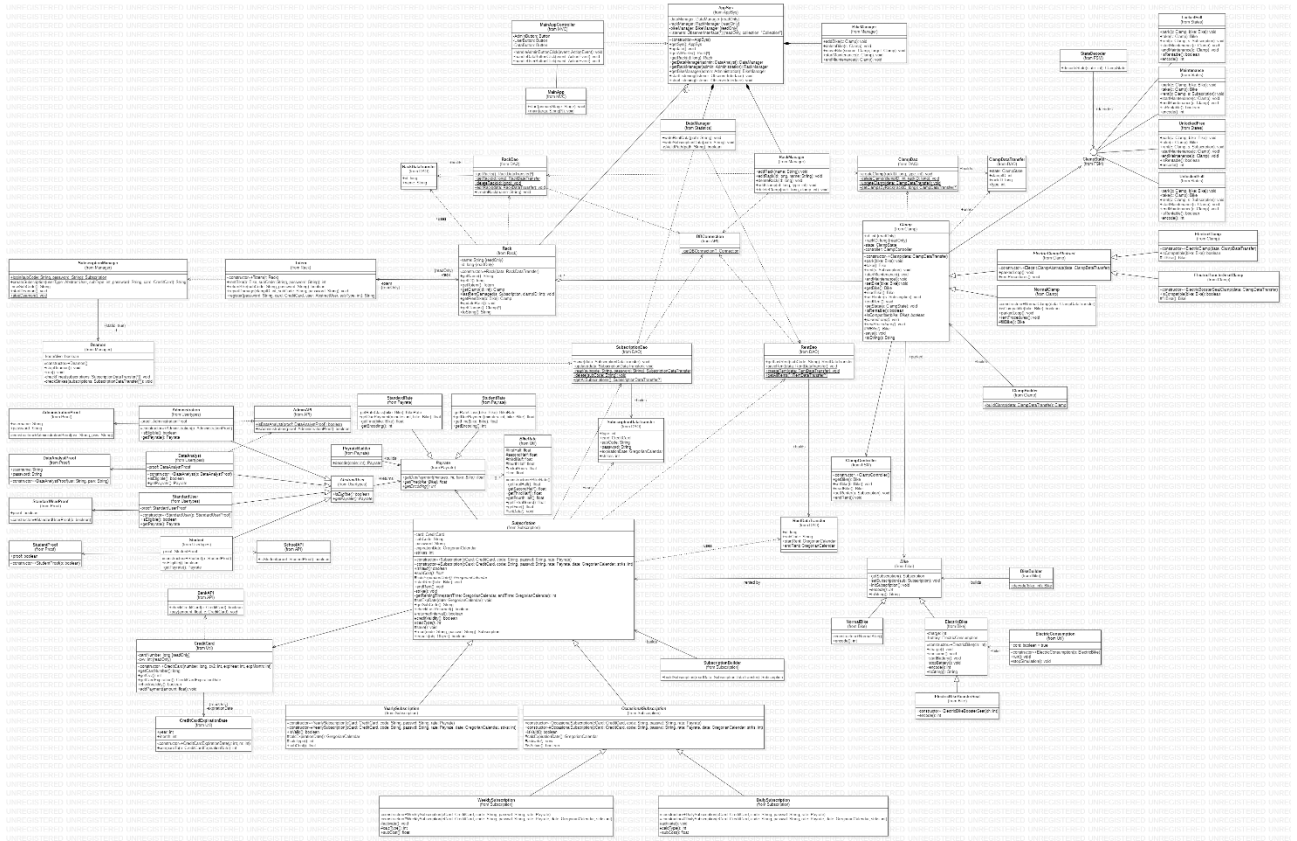


2.7 Diagramma dei componenti



3 Implementazione del sistema

3.1 Diagramma delle classi (modello di programma)



Note:

- Il ramo User del diagramma di progetto è stato spostato da componente attiva a componente passiva solo per validare i tariffari e utenti speciali (amministratori e data analyst). La funzionalità del ramo User vengono ora ricoperte dal controller dell'applicazione (seguendo il design pattern MVC).

3.1.1 Discussione dei Design Pattern utilizzati

3.1.1.1 Data Access Object Pattern

Tutte le interazioni con il database vengono implementate rispettando il pattern DAO, in modo da tenere separati modello e livello persistente dell'applicazione. Le informazioni vengono poi trasportate tra modello e livello persistente tramite degli oggetti ad hoc (i Data Transfer) che vengono inviati ai Builder.

3.1.1.2 Model View Controller Pattern

L'interfaccia utente è stata realizzata seguendo il pattern Model View Controller. Questo consente di tenere separati il modello e la parte di applicazione che gestisce l'interazione con l'utente. In particolare, in questo caso, la componente della vista e del controllore sono unite in un'unica componente di controllo.

3.1.1.3 *Observer Pattern*

Le diverse applicazioni fanno uso del pattern Observer per poter mantenere aggiornate le informazioni visibili all'utente. Quando un'applicazione viene aperta, questa si inserisce come listener all'interno del sistema, che si impegna a notificarla ogni volta che il modello subisce una modifica.

3.1.1.4 *Factory Pattern*

Per la costruzione di oggetti complessi come Subscripion, Rack, Clamp, ma anche per la ricostruzione degli oggetti del dominio a partire dalle informazioni contenute nel livello persistente si utilizza il pattern Factory per delegare ad una classe esterna la loro costruzione.

3.1.1.5 *Strategy Pattern*

Per gli stati della classe Clamp (interfaccia ClampState) si utilizza il pattern Strategy per permettere a ogni stato di definire la reazione ad una lista di eventi comuni (definita nell'interfaccia).

3.1.1.6 *Singleton Pattern*

Per garantire l'unicità di oggetti come le implementazioni concrete di ClampState o il sistema stesso (AppSys) viene utilizzato il Singleton Pattern. Questo garantisce che esiste un'unica istanza degli oggetti coinvolti e mette a disposizione un metodo statico per ottenere accesso all'istanza.

3.2 *Gestione dei dati persistenti*

Il Database memorizza informazioni sulle stazioni (Rack), sulle morse (Clamp), sugli abbonamenti (Subscription) e sui prestiti. Le biciclette vengono generate a Runtime a partire dagli stati delle morse (per design, visto che l'app non andrebbe mai spenta, le biciclette non ancora restituite vengono rimosse dal sistema se l'app viene spenta con biciclette non parcheggiate). I prestiti invece sono solo contenitori di informazioni e non è necessaria una loro implementazione nell'applicazione.

3.3 Descrizione dell'Interfaccia Grafica

La schermata principale permette di selezionare quale applicazione utilizzare. L'applicazione per gli amministratori permette poi di selezionare l'applicazione per la modifica delle stazioni o di quella per lo spostamento delle biciclette.

Rack Name Create Rack

Delete Rack

New Rack Name Change Rack Name

Create Clamp

Delete Clamp

Figura 1: Applicazione per modificare le stazioni

Rack Add Bike Remove Bike Start Maintenance End Maintenance

Clamp

Source Rack Target Rack

Source Clamp Target Clamp

Move Bike

Figura 2: Applicazione per spostare le biciclette

L'applicazione per i data analyst permette di scrivere su file i log degli abbonamenti e dei prestiti.

Rent Data Directory Write Rent Data

Subscription Data Directory Write Subscription Data

Figura 3: Applicazione per i data analyst

L'applicazione utente è una simulazione per un'utente reale (non verrebbe davvero usata in un contesto reale). L'utente può usarla per navigare tra le stazioni e interagire con esse o con le singole morse.

Current bike: Bike

Rack

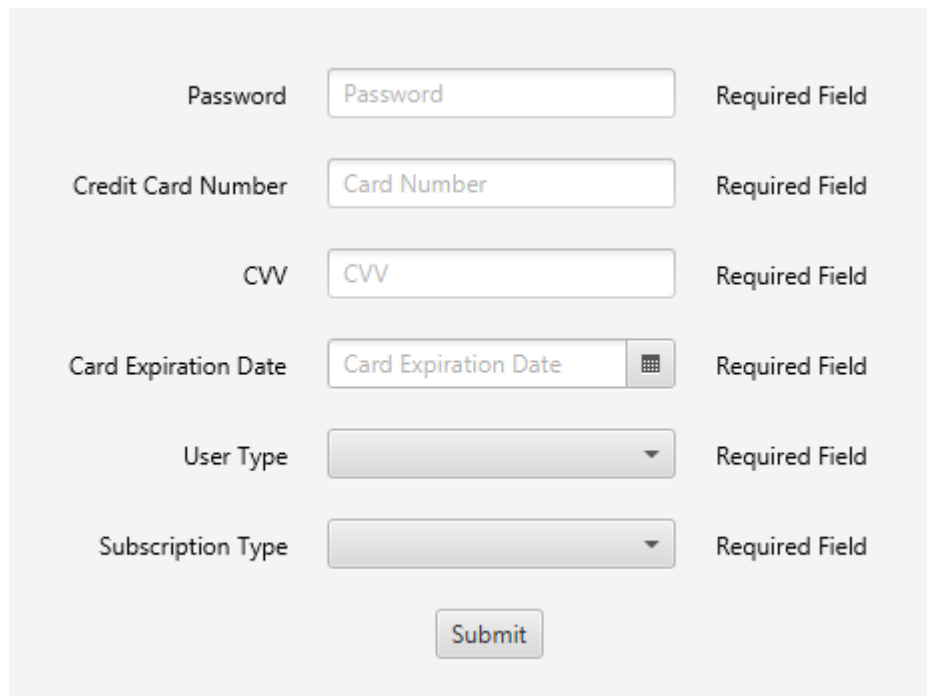
Clamp

Figura 4: Applicazione utente

Selezionando una stazione e cliccando su Open Totem si apre l'effettiva applicazione dei totem che verrebbe lasciata in esecuzione su ogni totem.

Figura 5: Applicazione totem

Selezionando l'opzione Register Subscription si apre la finestra di registrazione (per design la registrazione avviene direttamente da totem e non da un'applicazione esterna).



The image shows a registration form for a subscription. It consists of six rows of input fields, each with a label on the left and a 'Required Field' indicator on the right. The fields are: Password (text input), Credit Card Number (text input), CVV (text input), Card Expiration Date (text input with a calendar icon), User Type (dropdown menu), and Subscription Type (dropdown menu). A 'Submit' button is located at the bottom center of the form.


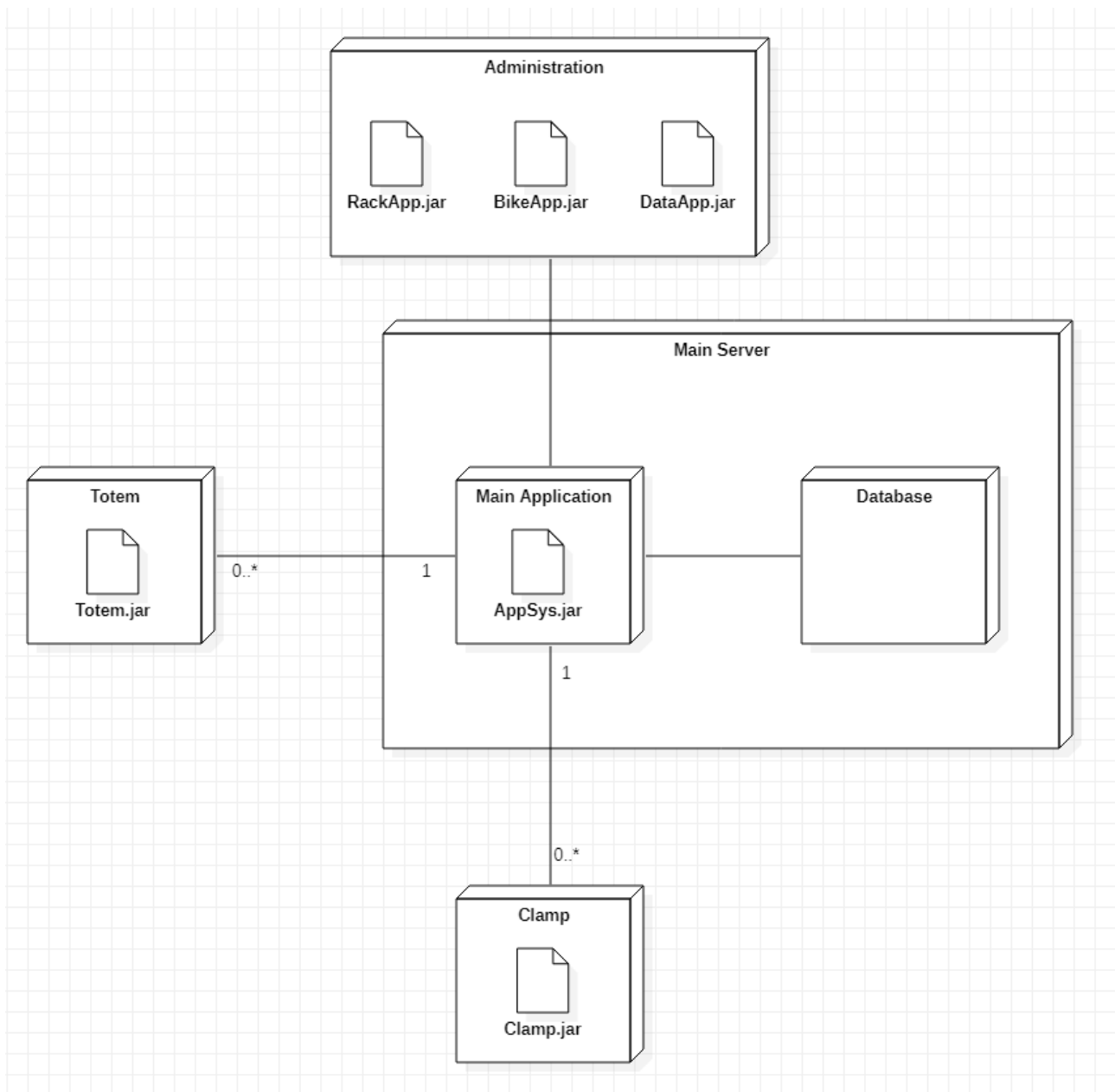
Password	<input type="text" value="Password"/>	Required Field
Credit Card Number	<input type="text" value="Card Number"/>	Required Field
CVV	<input type="text" value="CVV"/>	Required Field
Card Expiration Date	<input type="text" value="Card Expiration Date"/> 	Required Field
User Type	<input type="text" value=""/>	Required Field
Subscription Type	<input type="text" value=""/>	Required Field

Figura 6: Form di registrazione abbonamento

Tutte le applicazioni ad eccezione del percorso in cui salvare i log dei data analyst eseguono controlli dell'input per assicurarsi che ciò che l'utente inserisce in determinati campi corrisponda a ciò che il modello richiede.

3.4 Diagramma di deployment



3.5 Specifica e verifica dei vincoli

3.5.1 Subscription

3.5.1.1 OCL

```
{context Subscription
inv: Subscription.allInstances -> isUnique(subCode)
inv: self.password.length < 21 and self.password.length > 8
inv: self.isValid() implies self.card.chackValidity()
inv: self.card -> notEmpty()
inv: self.expirationDate != null -> self.expirationDate.after(now())
inv: self.strikes >= 0 and self.strikes <= 3
}
```

```
{context Subscription::startRent(b: Bike)
pre: self.isValid() and self.checkLastReturned()
post: not self.checkLastReturned()
}
```

```
{context Subscription::endRent()
pre: not self.checkLastReturned()
post: self.checkLastReturned()
}
```

```
{context Subscription::strike()
post: self.strikes = self.strikes@pre + 1
}
```

3.5.1.2 JML

```
//@requires this.isValid() && this.checkLastReturned();
//@ensures: !this.checkLastReturned();
public void startRent(Bike b) {...}
```

```
//@requires !this.checkLastReturned();
//@ensures this.checkLastReturned();
public void endRent() {...}
```

```
//@ensures this.strikes == \old(this.strikes) + 1;
private void strikes() {...}
```

3.5.2 CreditCard

3.5.2.1 OCL

```
{context CreditCard
inv: self.cardNumber.length = 13 || self.cardNumber.length = 16
inv: self.expirationDate -> notEmpty()
inv: self.cvv >= 100 && self.cvv <= 999
inv: self.checkValidity() implies self.expirationDate.after(now())
}
```

3.5.3 CreditCardExpirationDate

3.5.3.1 OCL

```
{context CreditCardExpirationDate  
inv: self.year > 0  
inv: self.month >= 0 and self.month <= 11  
}
```

3.6 Descrizione del testing

Il testing è stato realizzato con dei test unitari sulle classi del dominio. Le classi di connessione con il database o le classi di interazione con l'utente richiederebbero dei test di integrazione per essere testati adeguatamente, ma la bassa correttezza richiesta derivata dallo scopo per cui questa applicazione è stata realizzata non giustifica una spesa di realizzazione di test di integrazione.

Per i test unitari si è scelto di utilizzare il branch coverage per tutte le porzioni di codice ad eccezione dell'inserimento di dati, dove è stato utilizzato MCDC per assicurarsi che i parametri inseriti non siano nulli o invalidi. Essendo MCDC costoso e l'applicazione non sensibile come un software medico, si è scelto un criterio più semplice di coverage per coprire il codice restante.

3.7 Note per l'installazione e l'utilizzo

L'applicazione è stata sviluppata in Java 15.0.1

L'interfaccia grafica usa Javafx versione 0.0.9

Per la compilazione si appoggia al sistema di build automation Gradle versione 6.7.1

Per avviare l'applicazione dalla cartella principale eseguire tramite terminale il comando `./gradlew run`.

Il livello persistente si appoggia su un database mySQL versione 8.0.22

Per i test unitari si utilizza il framework Mockito versione 3.10.0

Per la connessione con il database si usa la seguente connessione:

- Username: bikeSharingSystem
- Password: bikesharing

Non sono presenti stazioni o abbonamenti di prova, il dump contiene solo la struttura del database.

Login per amministrazione:

- Username: admin
- Password: 123

Login per data analyst:

- Username: data
- Password: analyst