

# Progettazione di Software

CON TECNICHE MODEL DRIVEN

Nome: Alessandro

Cognome: Sfriso

Matricola: 909769

Corso: Progettazione Model Driven del Software

Corso di laurea: Sicurezza dei Sistemi e delle Reti Informatiche (SSRI)



## Sommario

Introduzione .....	4
I vincoli assegnatemi in questo progetto .....	4
Le richieste di feature del livello di sviluppo .....	4
I requisiti della relazione .....	4
Requisiti del progetto del corso di Progettazione Model-Driven del Software .....	5
Use Case Diagram .....	6
Scenari .....	7
registrazione .....	7
Ricerca Auto .....	7
riconsegna auto .....	8
Class Diagram .....	8
Vicoli .....	12
OCL .....	12
JML .....	13
Sequence Diagram .....	13
login .....	13
registrazione .....	14
Noleggio Auto .....	14
Activity Diagram .....	15
noleggio .....	15
riconsegna auto .....	15
Macchine a Stati .....	16
Lettore carta di credito .....	16
Inserimento auto nella lista di auto .....	16
DataBase .....	17
Security .....	18
prepared Statement .....	18
Deployment Diagram .....	19
Pattern utilizzati .....	19
Design Pattern .....	19
Pattern GRASP .....	19

Interfaccia Grafica.....	20
utente non registrato .....	20
Impiegato.....	22
utente registrato.....	26
Funzioni principali.....	28
calcolo del prezzo di noleggio .....	28
Aggiornamenti Futuri .....	28
Per poter avviare il programma.....	29

## Introduzione

La progettazione model driven (MDE) è una metodologia di sviluppo software che si concentra sulla creazione e lo sfruttamento di modelli rappresentanti il funzionamento del sistema in ogni suo aspetto.

In questa relazione verrà esposta una progettazione di un software di medie dimensioni. In seguito, vengono descritti i vincoli che si sono seguiti durante tutta la progettazione del software

### I VINCOLI ASSEGNATEMI IN QUESTO PROGETTO

- Deve soddisfare i requisiti, informali, resi disponibili dal docente
- Il processo di sviluppo deve essere model-driven
- Il progetto deve essere implementato in java
- Si deve produrre una relazione

### LE RICHIESTE DI FEATURE DEL LIVELLO DI SVILUPPO

- Vincoli: vanno specificati, dove occorre, un certo numero di vincoli OCL
- Contratti: mostrare come almeno un piccolo sottoinsieme di vincoli OCL sia stato mappato in JML
- Pattern: va utilizzato almeno un pattern tra quelli di design e quelli architetturali
- GUI: l'interfaccia grafica sviluppata utilizzando Java Swing

### I REQUISITI DELLA RELAZIONE

La relazione deve descrivere l'intero processo di sviluppo del progetto, e deve contenere i seguenti modelli:

- Use case e descrizione scenari degli use case
- Diagrammi di sequenza
- Diagramma delle classi
- Diagrammi di attività, macchine di stato
- Diagramma delle componenti
- Diagramma di deployment

## REQUISITI DEL PROGETTO DEL CORSO DI PROGETTAZIONE MODEL-DRIVEN DEL SOFTWARE

Si progetti un sistema per la gestione di un'Agenzia di Autonoleggio di medie dimensioni.

L'applicazione dovrà servire sia il personale dell'Agenzia che gli utenti che desiderano noleggiare un'auto. Gli utenti possono essere sia utenti registrati che utenti non registrati.

L'utente non registrato deve poter:

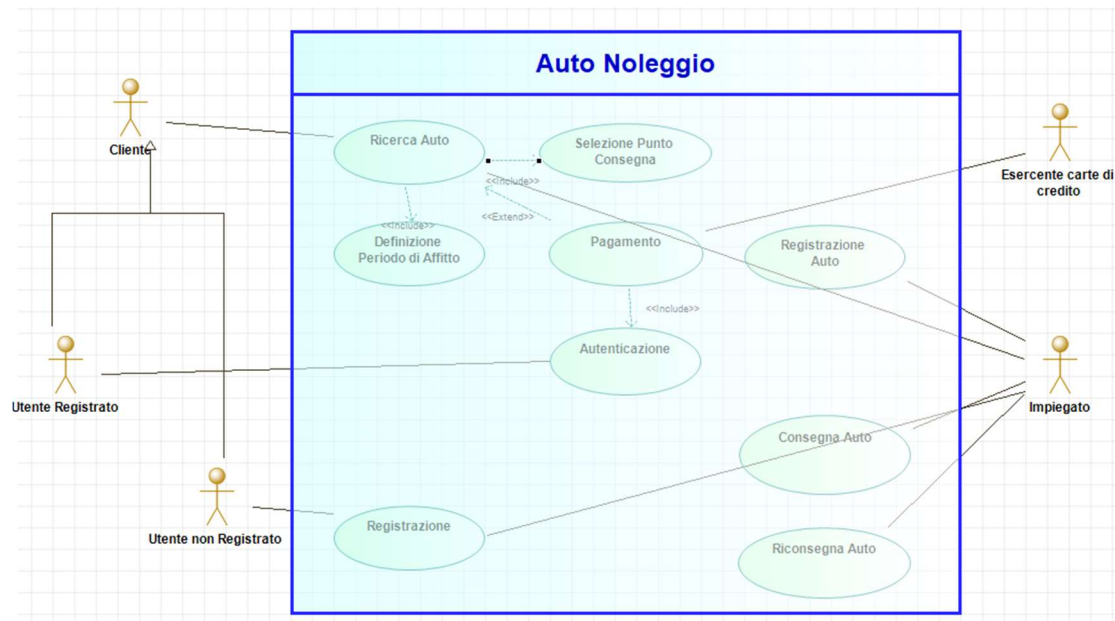
- Registrarsi al sistema inserendo alcuni dati personali (nome e cognome, data di nascita, dati della patente, ecc.) e scegliendo le credenziali per l'accesso registrato.
- Visualizzare il parco macchine disponibile e simulare una prenotazione per vedere le disponibilità e il prezzo, senza poter finalizzare la prenotazione (che richiedere l'accesso mediante le credenziali stabilite in fase di registrazione).

L'utente registrato deve poter:

- Prenotare un'automobile, scegliendo la macchina a seconda del numero di posti e/o della cilindrata dalla lista delle macchine disponibili e selezionando la durata del noleggio.
- Disdire una prenotazione.
- Modificare (ed eventualmente cancellare) i dati inseriti in fase di registrazione. Nel caso di deregistrazione, le credenziali precedentemente assegnate all'utente devono venire rimosse e l'utente non deve più essere in grado di accedere al sistema.
- [Opzionale e personalizzabile] Se l'utente non ritorna l'auto entro i termini prestabiliti deve pagare una multa (che varia a seconda del ritardo).
- L'impiegato dell'Agenzia deve poter:
- Visualizzare la lista di auto disponibili e di quelle a noleggio (con indicata la data della riconsegna) in un dato momento.
- Inserire (o togliere) un'automobile dal parco macchine a disposizione dell'Agenzia.
- Gestire l'effettiva consegna (e riconsegna) dell'automobile all'utente che l'aveva prenotata.
- [Opzionale e personalizzabile] Al momento della consegna e riconsegna, inserire i dettagli relativi allo stato dell'auto in termini di: danni alla carrozzeria, danni all'interno, pulizia interna/esterna.
- [Opzionale e personalizzabile] L'impiegato deve poter gestire tutto ciò che riguarda la manutenzione dell'automobile (revisione, tagliando, lavaggio interno ed esterno, ecc.).

L'applicazione va implementata in Java come applicazione stand-alone e non come applicazione Web (no Java Servlet). Deve utilizzare Java Swing per la gestione dell'interfaccia grafica e deve mantenere i dati persistenti in un database relazionale al quale si accede o via JDBC o Hibernate (ORM).

## Use Case Diagram



Esercente carte di credito è un attore terzo che ha il compito di verificare la carta di credito, di effettuare la transizione e verificare che la transazione sia andata a buon fine.

Nel programma finale non comunica con esercente di carte di credito perchè il metodo non è stato implementato.

## Scenari

Nella sezione scenari vengono descritte alcune delle possibili interazioni tra gli attori e il sistema. La descrizione è ad alto livello perchè non si ha ancora la certezza dei metodi che si andranno ad utilizzare.

### REGISTRAZIONE

Attore	Utente non registrato
precondizioni	Utente non deve essere registrato
Eventi	<ol style="list-style-type: none"><li>1) Utente selezione registrazione</li><li>2) Il sistema mostra una pagina dove mostra nome, cognome, username e altre informazioni</li><li>3) Utente inserisce le informazioni richieste dal sistema e clicca sul bottone salva</li><li>4) Sistema inserisce le informazioni nel sistema</li></ol>
Postcondizioni	L'utente viene registrato
Eccezioni	4a) Il sistema dà errore perchè lo username è già stato utilizzato da un altro utente

### RICERCA AUTO

Attore	Cliente e Impiegato
precondizioni	Nessuna
Eventi	<ol style="list-style-type: none"><li>1) Cliente/Impiegato seleziona il bottone della ricerca</li><li>2) Il sistema mostra la pagina dove sono richieste luogo di consegna, data di consegna e data di riconsegna e altre informazioni che servono per una ricerca più mirata</li><li>3) Cliente/Impiegato inseriscono le informazioni</li><li>4) Il sistema risponde con la lista di auto</li></ol>
Postcondizioni	Viene visualizzata la lista delle auto selezionate
Eccezioni	



## RICONSEGNA AUTO

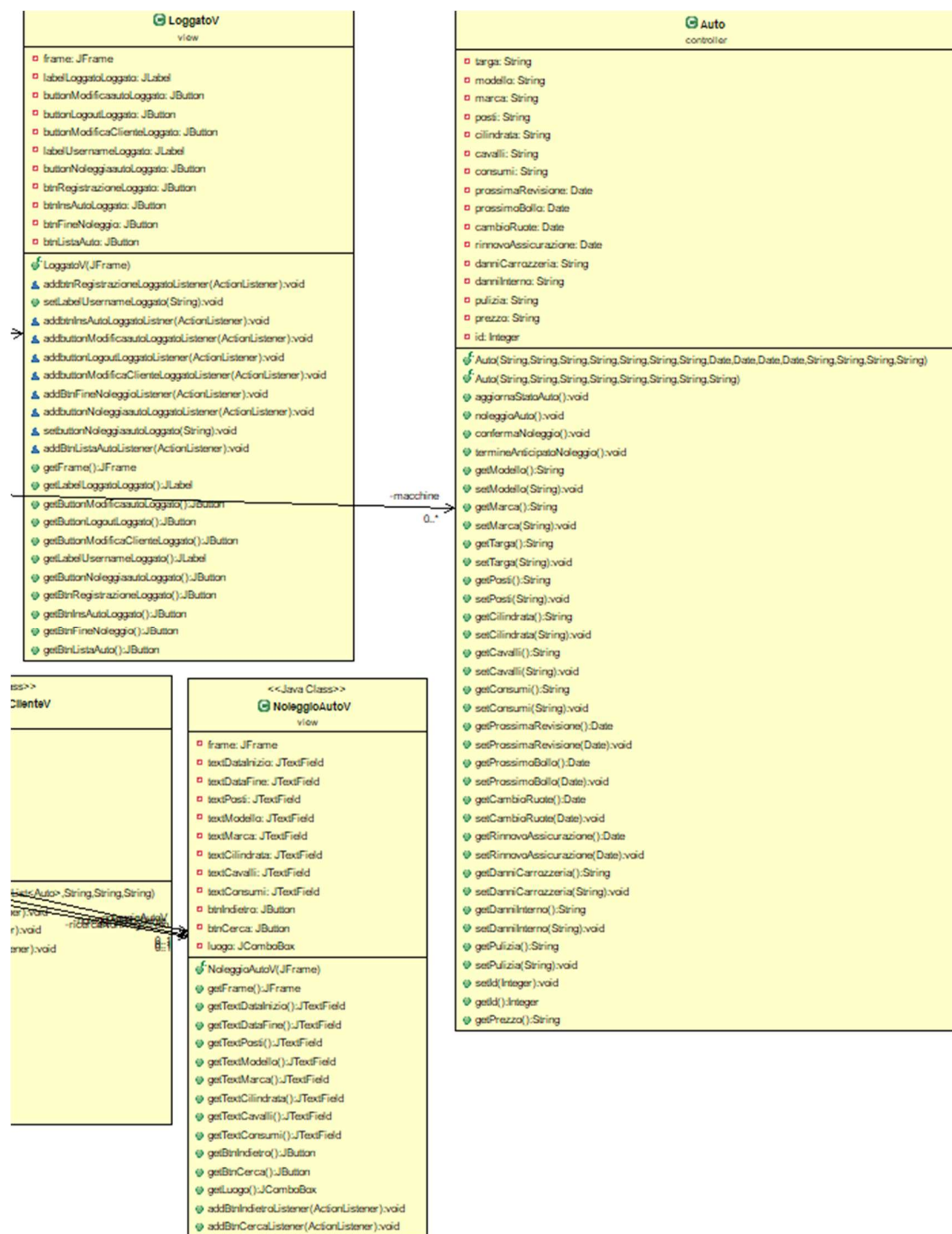
Attore precondizioni	Utente registrato e Impiegato  Utente registrato deve aver noleggiato un'auto ed essere munito di carta di credito carica
Eventi	1) l'utente arriva nel luogo di consegna 2) l'impiegato controlla l'auto 3) l'impiegato entra nella sezione apposita per consegnare l'auto 4) il sistema mostra la pagina 5) l'impiegato inserisce l'username dell'utente e la sua patente e, se necessario, modifica i vari parametri, infine seleziona il bottone di fine noleggio 6) il sistema mostra la pagina di pagamento con carta 7) il cliente inserisce la carta nel lettore di carte e inserisce il PIN 8) il sistema dà esito positivo
Postcondizioni Eccezioni	Vengono scalati i soldi dalla carta dell'utente 8a) il sistema dà esito negativo 9a) il cliente inserisce la carta nel lettore di carte e inserisce il PIN 10a) il sistema dà esito positivo  (in questa eccezione si effettuano tentativi fino a che non si raggiunge un esito positivo)

## Class Diagram

Si inizia a creare un primo elenco di classi che si andranno a inserire nel programma finale. Queste classi sono state aggiornate durante tutta la progettazione del software fino ad essere descritti come in figura.







Tutte le classi che nelle immagini non sono collegate ad una oclasse, sono collegate alla GestioneAutonoleggio.

Alcuni metodi come:

- addBtnPreventivo(ActionListener):Void nella classe RicercaNoleggioV,
- getTexfieldUsernameLogin():String e getTextfieldPasswordLogin():String nella classe LoginV,

non sono stati riportati a causa di mancanza di spazio

## Vicoli

Sono stati utilizzati dei vincoli OCL nella progettazione infine alcuni di questi vincoli sono stati implementati con il linguaggio JML.

### OCL

```
{context Noleggio
```

```
inv: self.dataInizioNoleggio.difference(self.dataFineNoleggio).equals(self.durata)
```

```
inv: dataInizioNoleggio < dataFineNoleggio}
```

il primo vincolo in linguaggio naturale permette di specificare che la variabile durata deve essere uguale alla differenza tra dataFineNoleggio e dataInizioNoleggio

il secondo vincolo in linguaggio naturale specifica che la dataInizioNoleggio deve essere minore di dataFineNoleggio

```
{context Utente
```

```
inv: self.allInstances () -> isUnique(patente)
```

```
inv: self.allInstances () -> isUnique(username)
```

```
}
```

I vincoli de Utente in linguaggio naturale specifica che in ciascuna istanza ci deve essere una diversa patente e username.

```
{context Impiegato
```

```
inv: self.allInstances () -> isUnique(username)
```

```
}
```

I vincoli dell'Impiegato in linguaggio naturale specifica che in ciascuna istanza ci deve essere un diverso username.



```

{context Auto

inv: self.allInstances () -> isUnique(targa)

}

```

I vincoli dell'Auto in linguaggio naturale specifica che in ciascuna istanza ci deve essere una diversa targa.

## JML

Rappresentazione dei vincoli OCL in JML

```

/*@ public invariant inizioNolo.compareTo(fineNolo) <= -1;

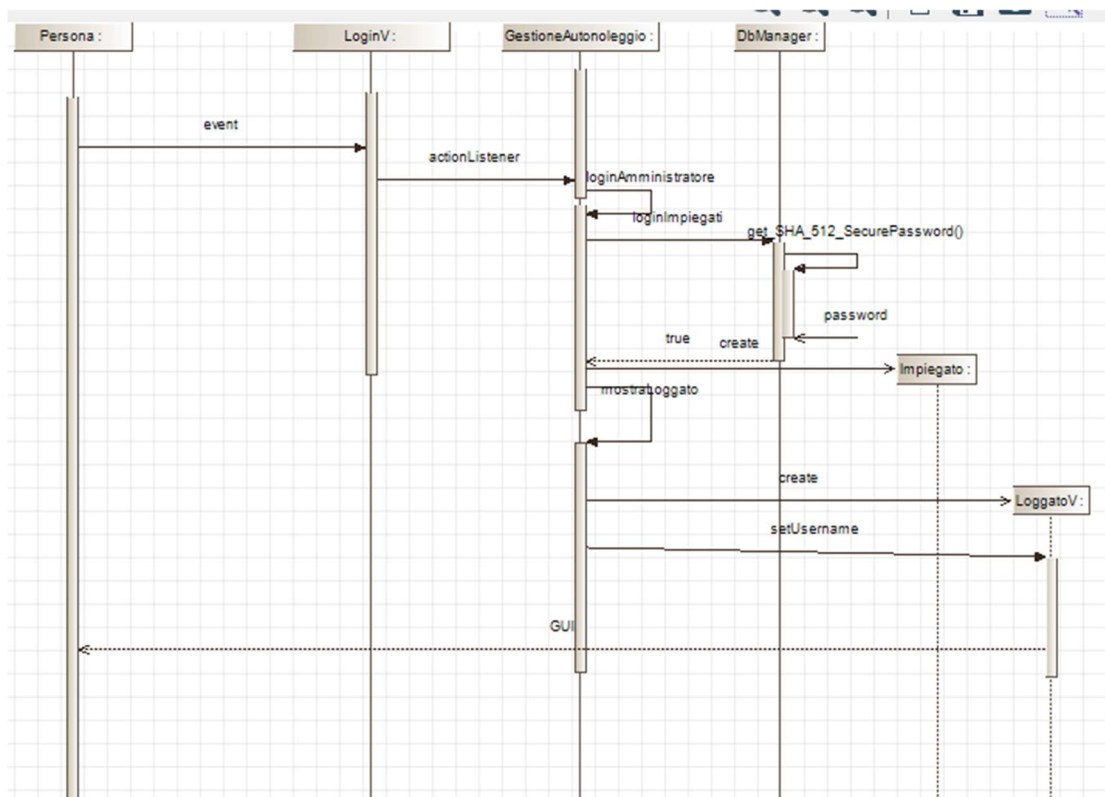
/*@ public invariant durata == TimeUnit.DAYS.convert((fineNolo.getTime()-inizioNolo.getTime()), TimeUnit.MILLISECONDS);

```

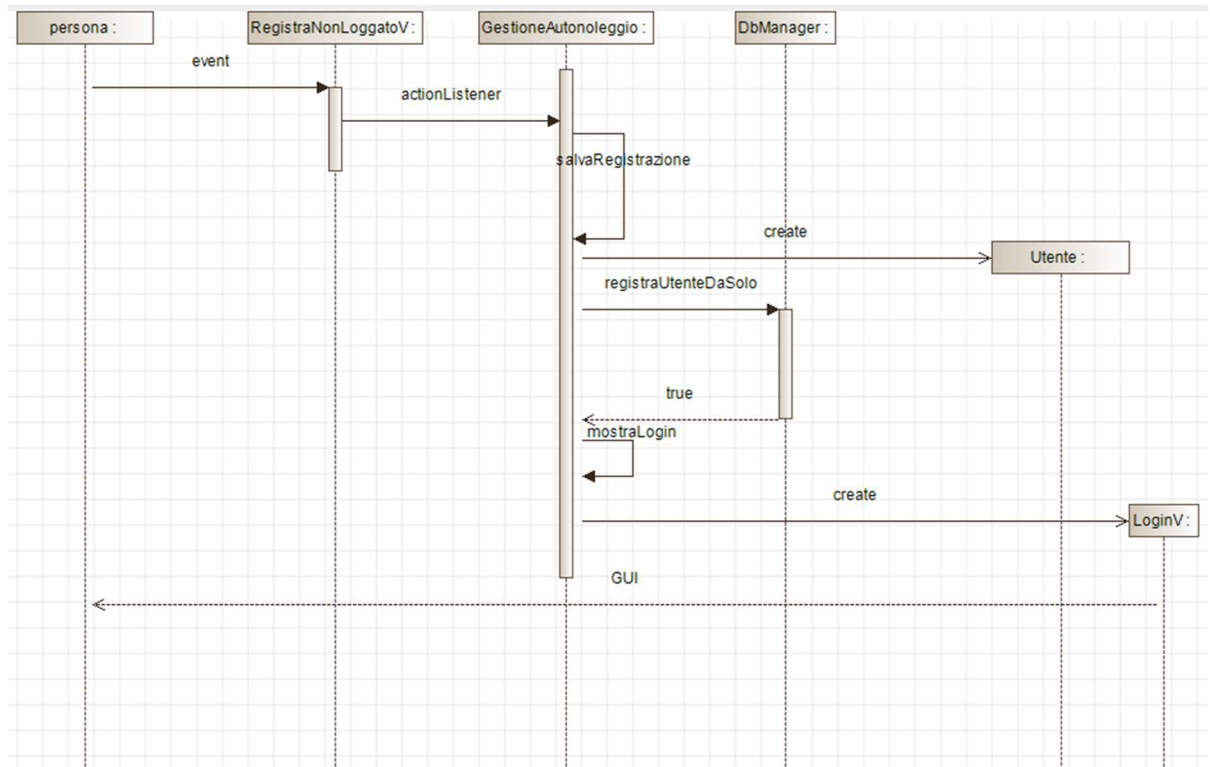
## Sequence Diagram

I Sequence Diagram sono dei diagrammi che permettono di comprendere come le varie classi interagiscono tra di loro.

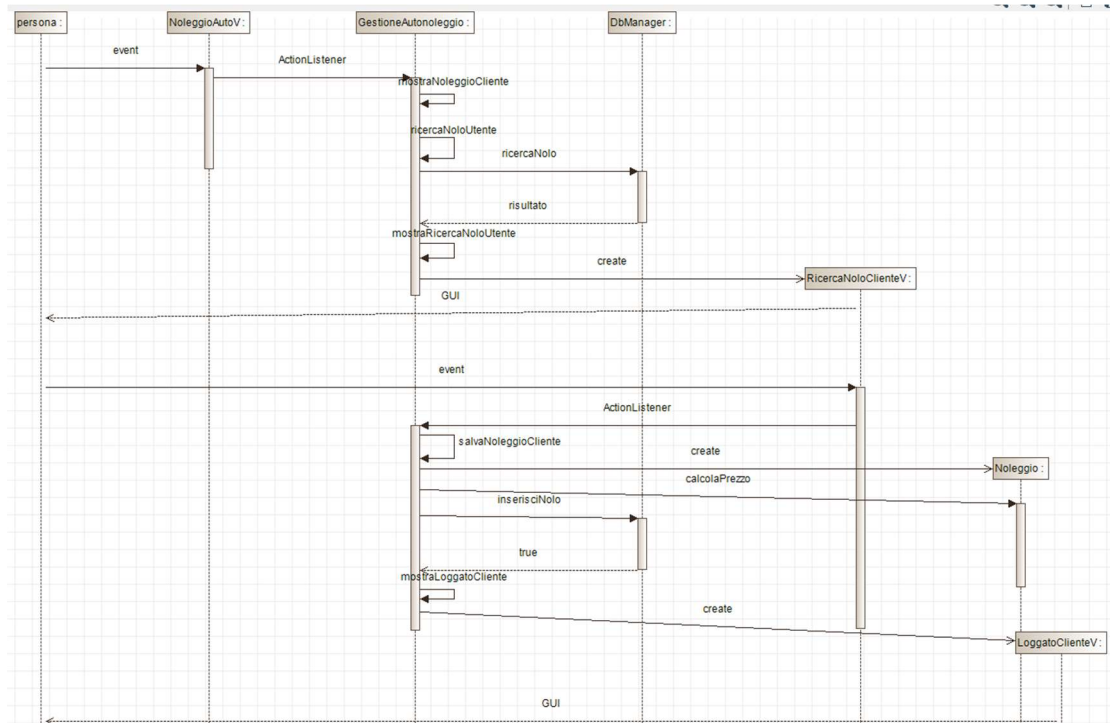
### LOGIN



## REGISTRAZIONE



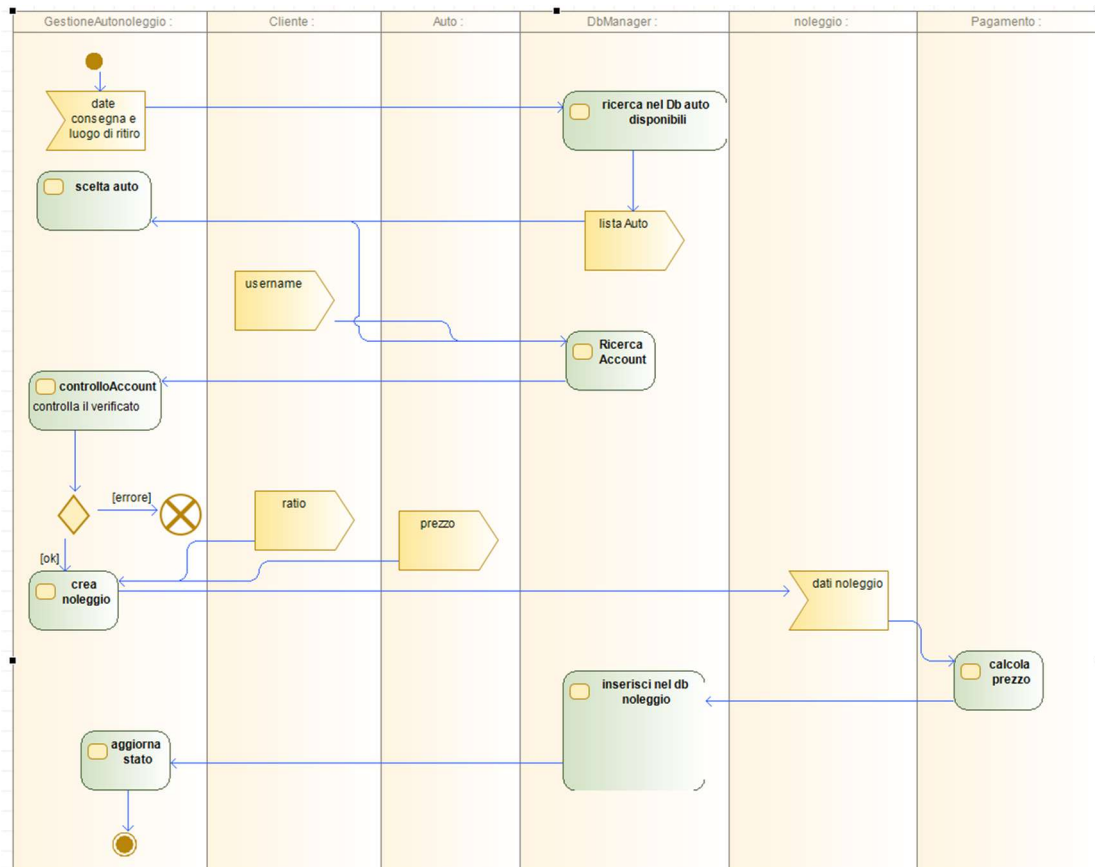
## NOLEGGIO AUTO



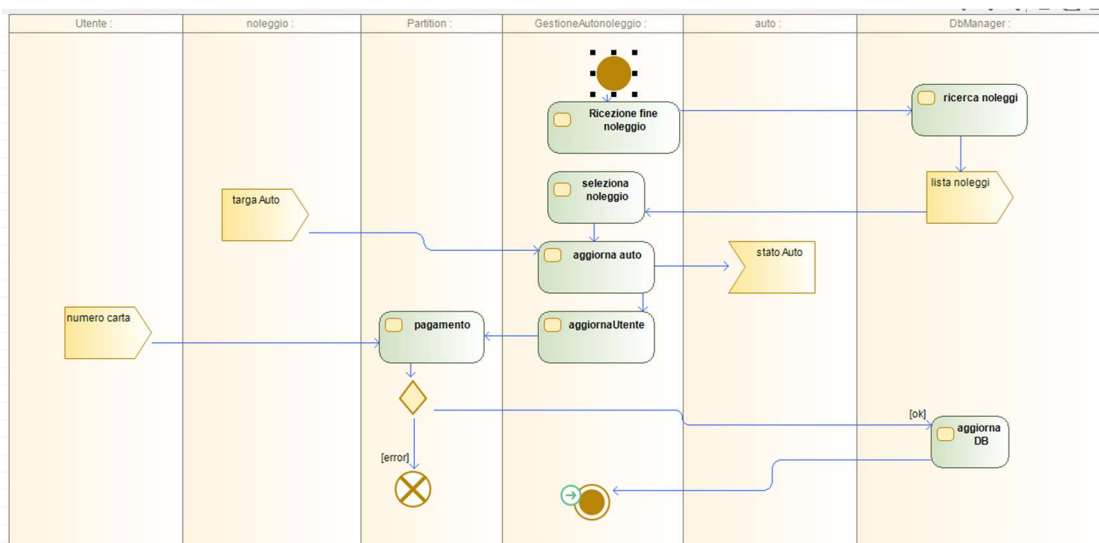
## Activity Diagram

I diagramma delle attività sono utilizzati per comprendere le varie attività che le classi all'interno del programma effettuano. Di seguito sono raffigurate le operazioni principali

### NOLEGGIO



### RICONSEGNA AUTO

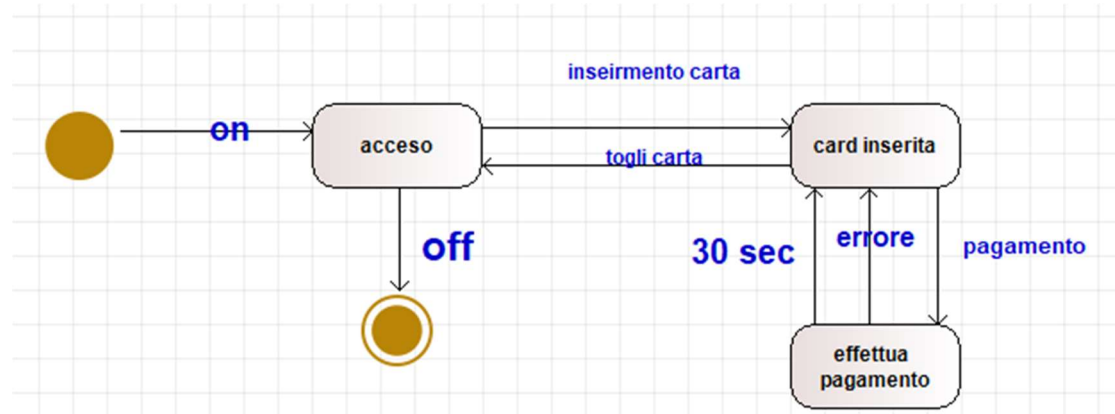




## Macchine a Stati

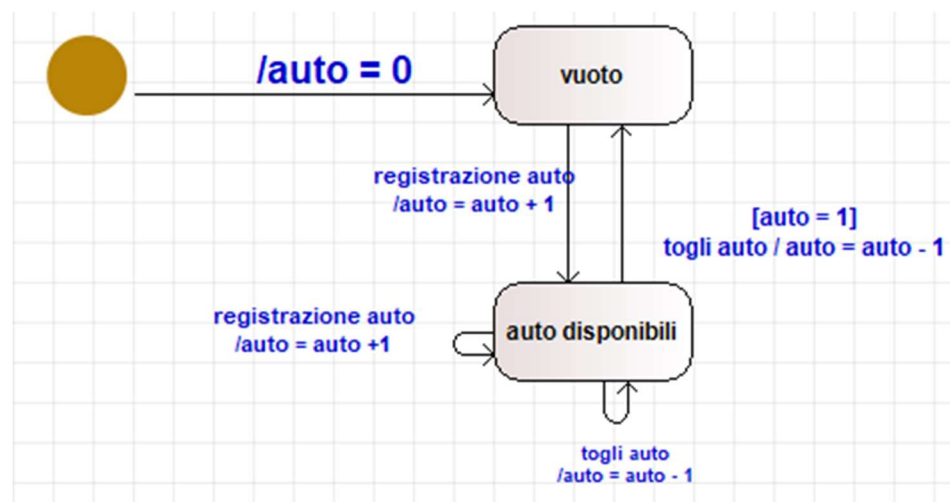
Le macchine a stati sono dei diagrammi che permettono di comprendere il funzionamento di alcuni dispositivi esterni, transazioni e mutatori di ruolo. Inoltre una macchina di stato rappresenta il ciclo di vita di un oggetto.

### LETTORE CARTA DI CREDITO



All' interno del progetto non è possibile utilizzare il lettore di carte di credito perchè al momento della progettazione non era presente il dispositivo. Si presuppone però che il suo funzionamento sia come quello in figura.

### INSERIMENTO AUTO NELLA LISTA DI AUTO



## DataBase

▼	CLIENTI	
	username	TEXT
	password	TEXT
	Salt	TEXT
▼	IMPIEGATI	
	Username	TEXT
	Password	TEXT
	Salt	TEXT
▼	INFOAUTO	
	targa	TEXT
	marca	TEXT
	modello	TEXT
	posti	TEXT
	cilindrata	TEXT
	cavalli	TEXT
	consumi	TEXT
	prossimaRevisione	TEXT
	prossimoBollo	TEXT
	cambioRuote	TEXT
	rinnovoAssicurazione	TEXT
	danniCarrozzeria	TEXT
	danniInterno	TEXT
	pulizia	TEXT
	prezzo	TEXT
▼	INFOIMPIEGATI	
	username	TEXT
	nome	TEXT
	cognome	INTEGER
▼	NOLEGGIO	
	targa	TEXT
	Username	TEXT
	dataInizio	TEXT
	dataFine	TEXT
	assicurazione	TEXT
	luogoConsegna	TEXT
	preventivo	TEXT

È stato utilizzato un database in SQLite, e le varie table sono quelle in figura.

## Security

Per quanto riguarda la security sono state utilizzate tecniche sia per lo store di password che per la gestione del database.

Le password all'interno del DB sono crittografate con SHA-512, il quale è molto utilizzato soprattutto perchè sono difficili da trovare delle collisioni. Per renderlo ancora più sicuro ho utilizzato un salt di 20 bit.

```
public String generateSalt() {
    SecureRandom random = new SecureRandom();
    byte bytes[] = new byte[20];
    random.nextBytes(bytes);
    String string = "";
    for (int i = 0; i < bytes.length; i++) {
        string += (char) bytes[i];
    }
    return string;
}

public String get_SHA_512_SecurePassword(String passwordToHash, String salt) {
    String generatedPassword = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-512");
        md.update(salt.getBytes(StandardCharsets.UTF_8));
        byte[] bytes = md.digest(passwordToHash.getBytes(StandardCharsets.UTF_8));
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < bytes.length; i++) {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).substring(1));
        }
        generatedPassword = sb.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return generatedPassword;
}
```

## PREPARED STATEMENT

Ho utilizzato prepared statement per negare la possibilità di modificare le query attraverso l'input.

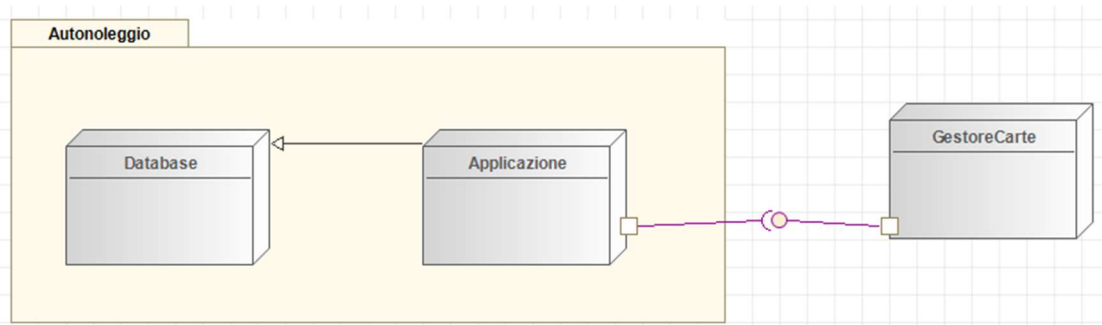
```
public void creaLoginUtenti(String username, String password) throws SQLException {
    PreparedStatement prst = connection.prepareStatement(dbu.INSERISCI_IMPIEGATI);
    String salt = generateSalt();
    prst.setString(1, username);
    prst.setString(2, get_SHA_512_SecurePassword(password, salt));
    prst.setString(3, salt);
    prst.execute();
    prst.close();
}
```

Nel DB la entry viene visualizzata in questo modo.

	username	password	Salt
	Filter	Filter	Filter
1	alessa	19cd187dd27...	/x□ΔuNt□ ...

## Deployment Diagram

Deployment Diagram permette di comprendere la struttura hardware del sistema che si sta progettando. In questo caso il sistema è formato da un GestoreCarte esterno all'Autonoleggio. Nell' Autonoleggi si evince che che è formato da una applicazione ed un database.



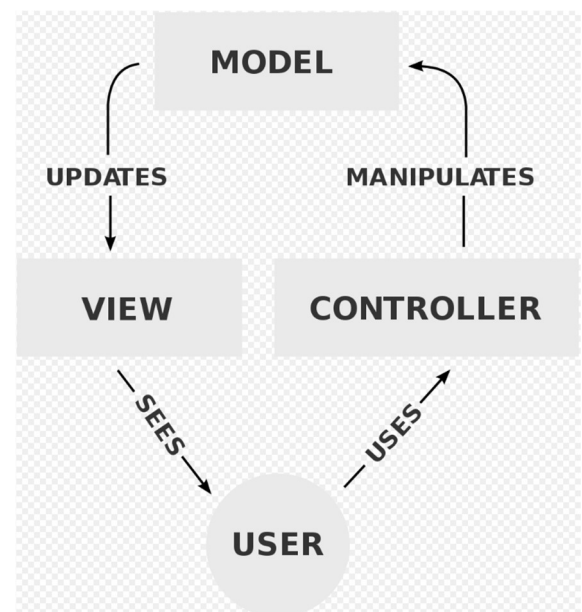
## Pattern utilizzati

### DESIGN PATTERN

È stato utilizzato il pattern Model View Controller.

Questo Design Pattern permette di separare i compiti all'interno della computazione.

- Controller: è la porzione di codice che è adibita alla gestione di tutto il programma
- Model: è la porzione di codice che si interfaccia con i dati statici, per esempio con il DB
- View: è la porzione di codice che permette di rappresentare i dati inviati dal Model



### PATTERN GRASP

È stato utilizzato il pattern Creator e Expert. Secondo questa serie di pattern inoltre si è cercato di tenere un basso numero di accoppiamento con un'alta coesione tra le classi.

In questa circostanza il Creator e Expert di Auto, Noleggio e Cliente è GestioneAutonoleggio.

## Interfaccia Grafica

### UTENTE NON REGISTRATO

The screenshot shows a web interface titled "Gestionale Autonoleggio". It features a login section with two input fields: "UserName" and "Password". Below these fields are four buttons: "Ricerca Auto", "Cliente", "Impiegato", and "Registrati". The "Registrati" button is positioned centrally below the other three.

**Gestionale Autonoleggio**

UserName

Password

Pagina di login

The screenshot shows a web interface titled "Cerca Auto Noleggio". It contains several search criteria with input fields: "Luogo" (a dropdown menu showing "Sede di R..."), "Seleziona Data Inizio e Fine Noleggio" (two date input fields), "Posti" (text input), "Modello" (text input), "Marca" (text input), "Cilindrata" (text input), "Cavalli" (text input), and "Consumi" (text input). At the bottom are two buttons: "Indietro" and "Cerca".

**Cerca Auto Noleggio**

Luogo  ▼

Seleziona Data Inizio e Fine Noleggio

Posti

Modello

Marca

Cilindrata

Cavalli

Consumi

Cerca Auto di un utente non registrato

## Noleggia Auto

Seleziona Auto

Targa	Data Inizio	Data Fine	Marca	Modello	Posti	Cilindrata	Cavalli	Consumi	Prezzo al ...	Luogo
sfdsg	12/12/20	13/12/20	BMW	lkj	343	4324	4324	34	300	Sede di R...

Assicurazione Casco ▼

Prezzo Preventivo

320.0

Indietro

Prevent...

Lista auto cercate utente non registrato

## Registrazione

**Nome**

**Cognome**

**Data di Nascita**

**Seriale Patente**

**Scadenza Patente**

**Email**

**Username**

**Carta Di Credito**

**Numero di telefono**

**Inserisci Passwo...**

**Rinserisci Paswo...**

Indietro

Salva

Registrazione

Dopo che un utente si è registrato deve essere verificato da un impiegato per far si che il suo profilo possa essere utilizzato per noleggiare le auto.

## IMPIEGATO

### Ti Sei Loggato

admin

Inserisci Auto	Modifica Cliente
Modifica Auto	Registrazione
Noleggia Auto	Fine Noleggio
	Logout

Login Impiegato

### Inserisci Auto

Targa	<input type="text"/>
Marca	<input type="text"/>
Modello	<input type="text"/>
Posti	<input type="text"/>
Cilindrata	<input type="text"/>
Cavalli	<input type="text"/>
Consumi	<input type="text"/>
Prossima Revisi...	<input type="text"/>
Prossimo Bollo	<input type="text"/>
Pros. Cambio Ru...	<input type="text"/>
Pros. Assicurazi...	<input type="text"/>
Danni Carrozzeria	<input type="text"/>
Danni Interno	<input type="text"/>
Pulizia	<input type="text"/>
Prezzo	<input type="text"/>

Inserisci Auto impiegato

### Modifica Auto

Targa	Marca	Modello	Posti	Cilindrata	Cavalli	Consumi	Prossima R.	Prossimo B.	Data Cambi.	Rinnovo As.	Danni Carro.	Danni Interni	Pulizia	Prezzo
sdfsdf	sda	lkj	343	4324	4324	34	02/02/20	02/02/20	02/02/20	02/02/20	02/02/20	02/02/20	02/02/20	300
dsa	sda	asd	sad	asd	sad	sad	12/12/20	12/12/20	12/12/20	12/12/20	ok	ok	ok	100

Indietro
Elimina
Salva

### Modifica Auto

In questa pagina possono essere modificati tutti i paramentri tranne la targa se è uguale ad un' altra targa di un'altra auto nella lista.

## Cerca Auto Noleggio

**Luogo**

Sede di R...
▼

**Seleziona Data Inizio e Fine Noleggio**

**Posti**

**Modello**

**Marca**

**Cilindrata**

**Cavalli**

**Consumi**

Indietro

Cerca

Pagina di ricerca noleggio



### Noleggia Auto

Seleziona Auto

Targa	Data Inizio	Data Fine	Marca	Modello	Posti	Cilindrata	Cavalli	Consumi	Prezzo al ...	Luogo
dsa	12/12/20	19/12/20	sda	asd	sad	asd	sad	sad	100	Sede di R...

Username

Patente

Assicurazione Normale ▼

Prezzo Preventivo

00.00 €

Auto Non Disponibili

Targa	Data Inizio	Data Fine	Marca	Modello	Posti	Cilindrata	Cavalli	Consumi	Prezzo al giorno
sdfsdf	12/12/20	19/12/20	sda	lkj	343	4324	4324	34	300

Pagina di ricerca noleggio dopo aver selezionato il bottone cerca

Nel riquadro sopra sono inserite le auto disponibili e sotto quelle non disponibili, che ripetano la ricerca precedente.

Per noleggiare un auto bisogna selezionare l'auto e poi inserire i dati del cliente, solo i clienti con un account verificato possono noleggiare le auto e fare un preventivo.

### Modifica Clienti

Nome	Cognome	Patente	Ratio	Email	accountVerifi...	Username	dataScadenz...	dataDiNascita	numeroDiCe...	cartaDiCred
alessandro	sfriso	alessa	0.0	asfdasf@g...	<input type="checkbox"/>	alessa	02/02/20	02/02/20	1341234	42142
luca	luca	luca	10.0	hkjds	<input checked="" type="checkbox"/>	luca	12/12/20	12/12/20	3423	32432

Modifica Clienti impiegato

In questa pagina l'impiegato può modificare tutti i clienti, non può essere modificato solo lo username

## Registrazione

Nome	<input type="text"/>
Cognome	<input type="text"/>
Data di Nascita	<input type="text"/>
Seriale Patente	<input type="text"/>
Scadenza Patente	<input type="text"/>
Email	<input type="text"/>
Username	<input type="text"/>
Carta Di Credito	<input type="text"/>
Numero di telefono	<input type="text"/>

Registrazione amministratore

L'impiegato inoltre può registrare un utente, ovviamente non può inserire la password. La nuova password sarà uguale allo username dell'utente, una volta entrato nel suo profilo il cliente potrà modificare la sua password.

### Fine Noleggio

Noleggio

Targa	Username	Data Inizio	Data Fine	Luogo Consegna	Preventivo
sdfsdf	luca	12/12/20	12/12/21	Sede di Roma	116800.0

Stato Auto

Targa	Danni Carrozzeria	Danni Interno	Pulizia	Prezzo
sdfsdf	02/02/20	02/02/20	02/02/20	300

Utente

Cellulare	Ratio	Email	Username
3423	10.0	hijds	luca

Fine noleggio

Una volta che è tornata in casa l'auto del cliente si effettua l'ultima pratica di consegna dell'auto. Una volta selezionato l'auto che si vuole consegnare si seleziona sul tasto cerca per far comparire le informazioni dell'auto e dell'utente. Tutte le varie informazioni dell'utente, dell'auto e del noleggio sono modificabili, tranne: targa, username e luogo consegna.

Una volta che è arrivata in casa l'auto, l'impiegato ha il dovere di controllare tutta l'auto e se necessario modificare i parametri, se vi sono gravi danni, l'impiegato ha il dovere di modificare la variabile ratio.

In questa schermata non è stato implementato il tasto Paga noleggio.

## UTENTE REGISTRATO

# Loggato

alessa

Modifica Profilo

Noleggia Auto

Logout

Cancella Nolo

Loggato cliente

## Modifica Profilo

Nome	Cognome	Patente	Ratio	email	AccountVerific...	Username	Data Scadenz...	Data Di Nascita	Numero di Cell	Carta Di Cred...
alessandro	sfriso	alessa	0.0	asfdasf@gm...	false	alessa	02/02/20	02/02/20	1341234	42142

Salva

### Modifica Password

Inserisci vecchia password

Nuova Password

Ripeti Password

Cambia Pass...

Indietro

Cancell...

Modifica profilo cliente

In questa pagina si possono modificare tutte le informazioni del proprio profilo, tranne: AccountVerificato e Username. Inoltre è possibile cancellare definitivamente il proprio profilo, una volta cancellato però non sarà possibile utilizzare lo stesso.

## Cerca Auto Noleggio

**Luogo**

Sede di R...
▼

**Seleziona Data Inizio e Fine Noleggio**

**Posti**

**Modello**

**Marca**

**Cilindrata**

**Cavalli**

**Consumi**

Indietro

Cerca

Cerca auto cliente

## Noleggia Auto

Seleziona Auto

Targa	Data Inizio	Data Fine	Marca	Modello	Posti	Cilindrata	Cavalli	Consumi	Prezzo al ...	Luogo
dsa	12/12/20	12/12/21	sda	asd	sad	asd	sad	sad	100	Sede di R...

Assicurazione Normale ▼

Prezzo Preventivo  
 00.00 €

Noleggia

Indietro

Prevent...

Noleggio Auto

Il cliente ha la possibilità dopo aver selezionato l'auto da noleggiare di fare un preventivo dell'auto e/o noleggiare l'auto scegliendo l'assicurazione che preferisce.

## I miei Noleggi

Targa	Username	Inizio Nolo	Fine Nolo	Assicurazione	Luogo Consegna	prezzo
dsa	alessa	12/12/20	12/12/21	Assicurazione Casco	Sede di Roma	43800.0

[Indietro](#)

[Cancella Prenotazione](#)

### I miei noleggi

Nella sezione i miei noleggi sono visibili tutti i noleggi attivi, inoltre è possibile cancellare la prenotazione da questa schermata.

## Funzioni principali

### CALCOLO DEL PREZZO DI NOLEGGIO

Il prezzo del noleggio viene calcolato secondo la seguente funzione:

$$((\text{prezzo} + \text{prezzo} * \text{ratio}) + \text{assicurazione}) * \text{numeroGiorni}$$

Prezzo è il prezzo dell'auto selezionato per ciascun giorno.

Ratio è una variabile che tiene conto il background dell'utente. Più alto è il valore più inaffidabile è l'utente, la sua inaffidabilità intaccherà il prezzo dei suoi futuri acquisti.

Assicurazione è il prezzo dell'assicurazione: per ora è presente l'assicurazione classica dell'auto a titolo gratuito oppure è presente l'assicurazione casco che costa 20€ al giorno.

numeroGiorni è la durata dell'affitto dell'auto.

## Aggiornamenti Futuri

La progettazione è stata pensata per essere utilizzata con sistemi che ad ora non sono supportati come il lettore di carte per i pagamenti. Il progettista si impegna nella prossima versione di aggiungere tale funzionalità.

## Per poter avviare il programma

Per poter avviare correttamente il programma bisogna modificare la variabile path nella classe GestioneAutonoleggio, con il path di dove è situato il file AutoNoleggio.db

La entry principale per l'impiegato è:

username: admin

password: admin