



SDD

System Design

Document

Riferimento	
Versione	1.9
Data	03/12/2020
Destinatario	AutoErre S.r.l
Presentato da	Caprio Mattia, Pepe Sara, Torino Francesco Maria, Iodice Michele Attilio, Mori Mattia, Sarro Antonio
Approvato da	Dario Di Dario



Revision History

Data	Versione	Descrizione	Autori
21/11/2020	1.0	Prima Struttura SDD	Mori Mattia
27/11/2020	1.1	Introduzione	Iodice Michele Attilio Pepe Sara
27/11/2020	1.2	Architettura Sistema Corrente	Iodice Michele Attilio Pepe Sara
27/11/2020	1.3	Architettura Sistema Proposto #1	Iodice Michele Attilio Pepe Sara
28/11/2020	1.4	Architettura Sistema Proposto #2	Torino Francesco Maria Sarro Antonio
30/11/2020	1.5	Architettura Sistema Proposto #3	Mori Mattia Sarro Antonio
02/12/2020	1.6	Modifica Deployment Diagrams	Mori Mattia Sarro Antonio Torino Francesco Maria
02/12/2020	1.7	Modifica Diagramma MVC	Mori Mattia Sarro Antonio Torino Francesco Maria
03/12/2020	1.8	Condizioni limite	Sarro Antonio
03/12/2020	1.9	Servizi dei Sottosistemi	Caprio Mattia



Sommario

1.Introduzione	4
1.1 Scopo del Sistema.....	4
1.2 Design Goals	4
1.3 Definizioni, Acronimi e Abbreviazioni	7
1.4 Riferimenti	10
1.5 Panoramica del Documento.....	10
2.Architettura del Sistema Corrente	11
3.Architettura del Sistema Proposto	12
3.1 Panoramica	12
3.2 Decomposizione in sottosistemi	13
3.3 Mapping Hardware/Software.....	15
3.4 Gestione dati persistenti	15
3.5 Controllo degli accessi e sicurezza	19
3.6 Controllo flusso globale del sistema.....	21
3.7 Condizione limite	21
4 Servizi dei Sottosistemi.....	24

1. Introduzione

1.1 Scopo del Sistema

Lo scopo del sistema è quello di fornire supporto all'azienda AutoErre S.r.l. consentendo una gestione quanto più semplice possibile del processo di noleggio auto. Ad oggi un sistema simile non esiste, alcuni competitor sfruttano un sistema più semplificato basato sulla singola azione di visualizzazione delle auto.

La piattaforma EasyLease nasce dal bisogno da parte dell'azienda AutoErre di semplificare completamente il processo di noleggio di un'auto che va dalla scelta di quest'ultima da parte di un cliente, alla stipulazione finale di un contratto con il relativo pagamento. Inoltre, con EasyLease l'azienda vuole migliorare ed ampliare la visibilità del proprio business.

EasyLease da un lato si prefigge lo scopo di rendere il meccanismo di noleggio auto il più semplice e veloce possibile, ottimizzando tutte le azioni necessarie, sia per il cliente che per il consulente, in termini di efficienza e di praticità. Grazie alla piattaforma e al suo sistema online, il cliente non avrà più l'esigenza di recarsi fisicamente in concessionaria a seguito di una prenotazione, mentre i consulenti dell'azienda potranno svolgere i propri compiti di gestione nel modo più semplice e affidabile possibile senza alcuna perdita di tempo.

Il sistema progettato è una piattaforma web accessibile a tutti, nella quale sono presenti delle aree riservate accessibili solo a determinati tipi di utenti abilitati. Il sistema può essere suddiviso in quattro aree di interazione:

- **Area Amministratore:** visualizzazione di auto e consulenti, aggiunta auto e consulenti, eliminazione auto, consulenti e modifica auto.
- **Area Consulenti:** visualizzazione auto, gestione e visualizzazione richieste preventivi e ordini.
- **Area Cliente:** visualizzazione auto, gestione preventivi e ordini.
- **Area Utente:** ricerca auto, consultazione catalogo e visualizzazione auto.

Il sistema deve fornire un metodo di autenticazione sicuro in modo che un utente non possa aver effettuato più di un accesso con account differenti contemporaneamente e che un qualsiasi utente non possa accedere all'area riservata ad un tipo di account non associato a lui. Coerentemente con ciò che è indicato nei requisiti non funzionali, il sistema dovrà avere caratteristiche di facilità d'uso e fluidità, così da offrire ai suoi utenti l'esperienza di una piattaforma intuitiva senza l'obbligo di consultare la documentazione.

1.2 Design Goals

I Design Goals sono organizzati in cinque categorie in base ai criteri di: Performance, Affidabilità, Costo, Manutenibilità e Usabilità. I Design Goals identificati nel nostro sistema sono i seguenti:

Criteri di performance

- Tempo di risposta:

Il sistema deve essere in grado di permettere una navigazione veloce con tempi di risposta brevi indipendentemente dal tipo di utente o dal carico applicativo che si presenta.

- Memoria:



Il sistema dovrà avere a sua disposizione una quantità di memoria sufficiente a contenere i dati del proprio database ma dovrà anche essere estendibile.

Criteri di affidabilità

- **Robustezza:**

Il sistema tenterà di evitare errori dovuti ad una errata compilazione dei form da parte dell'utente effettuando diversi controlli sui campi da esso compilati. Nei casi di criticità il sistema provvederà ad informare l'utente con opportune notifiche.

- **Affidabilità:**

Il sistema deve garantire una buona affidabilità nei suoi servizi. I risultati visualizzati devono essere sempre attendibili. Per quanto riguarda le auto che sono visibili sul sistema esse rispecchieranno la reale disponibilità da parte dell'azienda AutoErre. Ad esempio, se l'azienda ha in magazzino un nuovo modello di auto esso verrà aggiunta al sistema, così come se un'auto presente sul sistema non è più disponibile dall'azienda essa non verrà visualizzata. Il processo di login da parte di tutti gli utenti sarà gestito in modo affidabile, assicurando il corretto funzionamento del sistema, e garantendo l'accesso alle relative aree private, e l'utilizzo di determinate funzioni solo al tipo di utente per cui quell'area o quella funzione è riservata.

- **Disponibilità:**

Una volta online, il sistema sarà disponibile a tutti gli utenti (sia essi registrati o meno), i consulenti abilitati da un amministratore e a tutti gli amministratori del sistema.

- **Tolleranza ai guasti:**

Il sistema potrebbe riscontrare fallimenti dovuti a varie cause, tra cui un sovraccarico di dati nel database. Periodicamente verrà effettuata un'eliminazione all'interno del database degli ordini scaduti, dei preventivi vecchi stipulati e non accettati dopo un determinato periodo e degli ordini e/o preventivi che sono stati rifiutati dall'utente e/o consulente.

- **Sicurezza:**

L'accesso al sistema sarà garantito mediante e-mail e password; mentre l'accesso all'area riservata ai clienti è concessa solo agli utenti che hanno eseguito il processo di registrazione sulla medesima piattaforma; l'accesso all'area riservata degli amministratori è consentita ai soli utenti identificati come amministratori, mentre l'accesso all'area riservata dei consulenti è consentita solo agli utenti che sono stati aggiunti da un amministratore come consulenti.

Criteri di costi

- **Costo di sviluppo:**

È stimato un costo complessivo di 350 ore per la progettazione e lo sviluppo del sistema (50 per ogni membro del progetto).

Criteri di manutenibilità

- **Estensibilità:**

È possibile aggiungere nuove funzionalità al sistema, in seguito di nuove esigenze da parte dell'azienda o di nuove tecnologie che migliorino il sistema.



- **Adattabilità:**

Il sistema è fatto su misura per l'azienda AutoErre e quindi funziona solo per essa e tutti gli utenti che intendono interagire con essa.

- **Portabilità:**

L'interazione con il sistema avviene attraverso un browser a prescindere da quale esso sia, quindi possiamo definire il sistema come portabile. Poiché il sistema viene sviluppato come una piattaforma web, esso è accessibile da qualunque dispositivo, che sia esso mobile o fisso, che abbia un sistema operativo windows o altro, purché abbia un browser installato. Questa caratteristica garantisce la portabilità dello stesso.

- **Tracciabilità dei requisiti:**

La tracciabilità dei requisiti è garantita da una matrice di tracciabilità, che informa il lettore sull'associazione tra un determinato requisito ed una specifica parte del progetto relativa a quel requisito.

Criteri di usabilità

- **Usabilità:**

Il sistema sarà di facile comprensione e utilizzo, permettendo di effettuare in modo semplice e immediato le varie operazioni grazie a un'interfaccia intuitiva, di facile comprensione e utilizzo. L'intuitività del sistema è garantita da un'ottima prevedibilità di quelle che sono le azioni dell'utente, cioè la risposta del sistema ad un'azione utente sarà corrispondente alle aspettative. Inoltre, tramite l'utilizzo di apposite immagini il sistema sarà ancora più chiaro e intuitivo.

- **Utilità:**

Il compito del sistema è quello di sostenere l'utente nello svolgimento delle azioni necessarie a stipulare un contratto di leasing, infatti il tutto avverrà in maniera digitale dal proprio dispositivo, senza alcun bisogno di recarsi in concessionaria. Inoltre, faciliterà e velocizzerà il lavoro dei consulenti e amplierà la visibilità sul mercato dell'azienda AutoErre S.r.l.

1.2.1 Design Trade-off

Performance vs Memoria:

Il sistema predilige la performance alla memoria, per cui preserva la rapidità delle risposte, anche a costo di memorizzare dati superflui. Pertanto, verranno introdotte ridondanze per evitare interrogazioni costose e l'eliminazione dei dati non sarà istantanea.

Tempo di risposta vs Affidabilità

Il sistema sarà implementato in modo tale da preferire l'affidabilità al tempo di risposta, in modo tale da garantire una risposta del sistema consistente e non errata, contenente le giuste informazioni a discapito dal tempo impiegato per produrla (soprattutto nel caso della compilazione dei preventivi).

Disponibilità vs Tolleranza ai guasti

Il sistema punta ad essere sempre disponibile, ciò vuol dire che continuerà a funzionare anche nel caso di un guasto ad una funzionalità non indispensabile (funzionalità a priorità media o bassa), notificando

l'utente che in quel momento tale funzionalità non è disponibile. Il sistema verrà messo in manutenzione fino al risolversi del guasto in caso si tratti di una funzionalità prioritaria.

Criteri di manutenzione vs Criteri di performance

Il sistema nella sua implementazione preferirà la manutenibilità alla performance consentendo un processo di aggiornamento del software più semplice ed efficace agli sviluppatori, anche se ciò influirà sulle performance del sistema.

Di seguito è riportata la tabella che mostra i design goal che si prediligono nei trade off. Il grassetto indica la preferenza.

Trade-off	
Performance	Memoria
Affidabilità	Tempo di risposta
Disponibilità	Tolleranza ai guasti
Criteri di manutenzione	Criteri di performance

1.3 Definizioni, acronimi, abbreviazioni

1.3.1 Definizioni

Cliente: rappresenta un generico cliente, che sia intenzionato ad usare la piattaforma, utilizzando le operazioni descritte sopra per finalizzare un ordine. Ogni cliente ha un proprio account salvato sulla piattaforma in cui saranno specificate le seguenti informazioni:

1. ID cliente;
2. Nome;
3. Cognome;
4. E-mail;
5. Password;
6. Genere;
7. Data di nascita;
8. Luogo di nascita;
9. Residenza.

Consulente: rappresenta un lavoratore dipendente dell'azienda, il cui compito è quello di introdurre al cliente la stipulazione di un contratto di leasing per l'auto desiderata, quindi di sottoporre al suddetto un



preventivo che gli permetta di comprendere il costo a cui andrà in contro finalizzando l'ordine al momento del pagamento. Ogni consulente ha un proprio account salvato sulla piattaforma in cui saranno specificate le seguenti informazioni:

1. ID consulente;
2. Nome;
3. Cognome;
4. E-mail;
5. Password.

Amministratore: rappresenta un titolare o una figura che ne fa le veci, incaricato di gestire l'intera piattaforma, quindi di effettuare la gestione delle entità di consulenti e di auto. Ogni amministratore ha un proprio account salvato sulla piattaforma in cui saranno specificate le seguenti informazioni:

1. ID amministratore
2. Nome;
3. Cognome;
4. E-mail;
5. Password.

Utente registrato: rappresenta un generico utente registrato sulla piattaforma. Ogni utente registrato ha un proprio account salvato sulla piattaforma in cui saranno specificate le seguenti informazioni:

1. ID utente;
2. Nome;
3. Cognome;
4. E-mail;
5. Password.

Autovettura: rappresenta un'automobile e, come tale avrà le seguenti informazioni:

1. Id automobile;
2. Tipo di auto;
3. Marca;
4. Modello;
5. Caratteristiche tecniche;
6. Descrizione.

Preventivo: rappresenta un form contenente le informazioni associate ad una determinata auto di cui è stato richiesto il suddetto preventivo da parte di un cliente, che viene poi successivamente stipulato dal consulente. Tale preventivo è composto dalle seguenti informazioni:



1. ID auto;
2. Optional;
3. Prezzo;
4. Consulente;
5. Cliente.

Ordine: rappresenta un preventivo di cui è avvenuta la conferma da parte del cliente, che può essere confermato o meno dal cliente e poi convalidato o meno dal consulente:

1. ID ordine;
2. Data inizio;
3. Data fine;
4. Preventivo a cui fa riferimento;
5. Contratto;
6. Data minima di ritiro.

Utente: rappresenta un generico utente che fa uso della piattaforma, rappresentando quindi anche le entità di Cliente, Consulente, Amministratore e un qualsiasi utente non registrato.

Utente non registrato: rappresenta un generico utente che non è registrato sulla piattaforma.

Richiesta preventivo: rappresenta l'azione che effettua il cliente per richiedere il preventivo di una determinata auto.

Conferma preventivo: rappresenta l'azione di conferma di un preventivo da parte di un cliente.

Stipulazione preventivo: rappresenta l'azione da parte del consulente di compilare con i relativi costi un preventivo.

Conferma ordine: rappresenta l'azione da parte di un cliente di confermare un ordine.

Approvazione ordine: rappresenta l'azione da parte di un consulente di convalidare un ordine e di inserire la data minima di ritiro dell'auto in questione a seguito della conferma dello stesso ordine da parte del cliente stipulante.

Preventivo preso in carico: rappresenta il fatto che un consulente ha preso in carico il preventivo ed eventualmente il successivo ordine di un determinato cliente da lui scelto.

1.3.2 Acronimi ed Abbreviazioni

SDD = System Design Document (Documento di Progettazione del Sistema)

HW = Hardware

SW = Software

DBMS = Database Management System

A = Amministratore



AUT = Autenticazione

ACC = Account

CL = cliente

C = consulente

NA = Nessuna

CD = Class Diagram

GUI = Graphical User Interface (Interfaccia utente)

DBA = Data Base Administrator (Amministratore del Database)

1.4 Riferimenti

- RAD_V2.docx – Requirements Analysis Document
- <http://www.easylease.it>
- Bernd Bruegge & Allen H.Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (2nd edition), Prentice-Hall, 2004.
- Ian Sommerville, Ingegneria del software, (10a edizione), Pearson, 2017.

1.5 Panoramica del Documento

Il documento è diviso in quattro sezioni, a ciascuna delle quali è assegnato un compito ben preciso:

1. Introduzione: in questa parte viene fornita una descrizione del sistema, in modo da fornire una visione generale a chiunque legga questo documento. In particolare, viene illustrato lo scopo del sistema ed i suoi design goals. Inoltre, per evitare ambiguità e facilitare la comprensione, vengono illustrati gli acronimi, le definizioni e le abbreviazioni che sono state utilizzate nella stesura del documento. Infine, nell'introduzione è anche presente una sezione dedicata ai riferimenti utili a capire il suddetto documento.
2. Architettura del Sistema Corrente: questa sezione contiene la descrizione del sistema corrente. Tuttavia, come illustrato in seguito, ad oggi non esiste un vero e proprio sistema che fa ciò che si propone di fare EasyLease, ma esiste un sistema simile di riferimento.
3. Architettura del Sistema Proposto: in questa sezione viene illustrata l'architettura del Sistema Proposto. Questa sezione si apre con una breve panoramica che va ad illustrare il sistema proposto. Successivamente, viene effettuata la decomposizione del sistema generale in sottosistemi ed il mapping. Al punto successivo verrà esposto il mapping Hardware/Software. Poi verrà illustrata la gestione dei dati persistenti, il controllo degli accessi e la sicurezza, il controllo del flusso globale del sistema e le condizioni limite.
4. Servizi dei Sottosistemi: contiene la rappresentazione dei servizi dei sottosistemi.



2. Architettura Sistema Corrente

Attualmente non esiste un sistema che vuole realizzare gli stessi obiettivi di EasyLease: siamo quindi nel campo della Green-field Engineering, cioè stiamo esplorando funzionalità ancora mai scrutate. Un sistema simile di riferimento è Arval: i punti in comune con il sistema proposto sono la possibilità di ricercare e visualizzare un'auto e le sue caratteristiche e la possibilità di reperire dei contatti telefonici o telematici (e-mail); tuttavia, i due sistemi si diversificano dalla funzionalità che permette al cliente di compilare ed inviare una richiesta di preventivo e, per tanto, non esiste una vera e propria gestione del leasing online da parte del sistema Arval, ma solo una visualizzazione delle auto disponibili alla stipulazione di un contratto di leasing che andrà formalizzato fisicamente dal consulente e dal cliente.

3. Architettura Sistema Proposto

3.1 Panoramica

Il sistema proposto rientra nella Green-field Engineering, in quanto stiamo sondando per primi questo campo. EasyLease è una piattaforma web che per sua natura ha un ciclo di vita molto lungo (in quanto l'azienda AutoErre vuole mettere a disposizione la piattaforma a tempo indeterminato) e, per tanto, nel tempo il sistema andrà incontro ad un processo di reengineering al fine di aggiungere nuove funzionalità e migliorare quelle già presenti.

Il sistema è rivolto all'azienda AutoErre S.r.l. (quindi a chiunque sia in essa coinvolto, sia esso amministratore o consulente) e a chiunque sia interessato ad effettuare un leasing. La piattaforma mette a disposizione funzionalità diverse, in base a chi ne fa richiesta.

Un utente non registrato che si reca sul sito può registrarsi alla piattaforma, diventando così cliente. L'utente non registrato si differenzia dal cliente poiché può effettuare solo una parte delle attività che può effettuare il cliente. Infatti, l'utente non registrato può ricercare e visualizzare le auto, ma non richiedere un preventivo o, di conseguenza, sottoscrivere un contratto.

Il cliente, invece, oltre a ricercare e visualizzare le auto, può anche richiedere un preventivo. Una volta che la sua richiesta di preventivo riceve risposta, egli ha a disposizione un tempo prestabilito per accettare o rifiutare. Se egli decide di accettare, si procede con la sottoscrizione del contratto. Il cliente ha, infine, la possibilità di recarsi nella propria area utente e modificare le proprie informazioni personali.

Il consulente non ha la possibilità di registrarsi, poiché ciò viene fatto dall'amministratore. Egli ha la possibilità di fare l'accesso al sistema, visualizzare tutti gli ordini ad esso associati, visualizzare e rispondere alle richieste di preventivo ancora non associate ad alcun consulente, inoltre ha la possibilità di visualizzare tutti i clienti iscritti alla piattaforma oltre alle funzionalità di ricerca e visualizzazione di un'auto. Così come il cliente, il consulente ha la possibilità di andare nella propria area utente e modificare le proprie informazioni personali.

L'amministratore non si registra, poiché le sue credenziali sono già generate alla nascita della piattaforma. Egli ha la possibilità di fare l'accesso al sistema, effettuare l'aggiunta o la rimozione di un consulente e analogamente, l'amministratore può aggiungere, modificare o rimuovere un'auto, oltre alle funzionalità di ricercare e visualizzare un'auto.

Lo stile architetturale usato è di tipo repository in quanto i sottosistemi che compongono il software accedono e modificano una singola struttura dati (nel nostro caso un database MySQL). L'architettura implementa un pattern di tipo MVC, diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented in grado di separare la logica di presentazione dei dati dalla logica di business. Questo tipo di architettura è multi-tier, ovvero le funzionalità del sito sono separate e suddivise in più sottosistemi su più livelli in comunicazione tra loro.

3.2 Decomposizione in sottosistemi

La decomposizione adottata per il sistema è il pattern MVC composto da tre layer, ognuno dei quali gestisce determinate funzionalità:

- **Controller:** trasforma le interazioni dell'utente con la View in azioni che possono essere eseguite dal Model;
- **Model:** definisce la gestione dei dati e la gestione della logica del sistema, cioè le operazioni che possono essere eseguite;
- **View:** gestisce l'implementazione dell'interfaccia grafica, la quale rappresenta il mezzo tramite il quale gli utenti possono interagire col sistema.

Dopo attente valutazioni, abbiamo ritenuto opportuno adottare questo tipo di architettura per il nostro sistema poiché lo migliorerebbe sia in termini di efficienza che di manutenibilità.

Per via dei criteri mostrati nella sezione dei Design Trade-off, e quindi dalle esigenze del nostro sistema, tale decomposizione mira alla suddivisione delle componenti, facendo sì che risultino con un basso accoppiamento e un'alta coesione.

Component Diagram

Nella gestione delle componenti, ogni layer ha in gestione determinate componenti, di seguito espone.

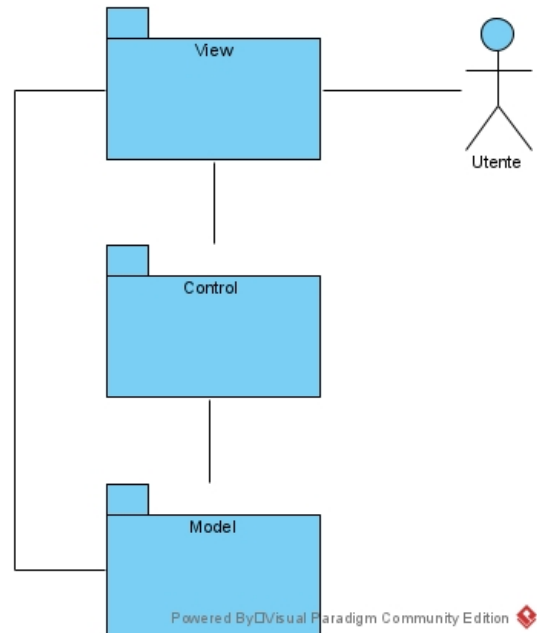
Il layer Controller gestisce cinque componenti:

- Gestione Amministratore
- Gestione Consulente
- Gestione Cliente
- Gestione Auto
- Gestione Autenticazione

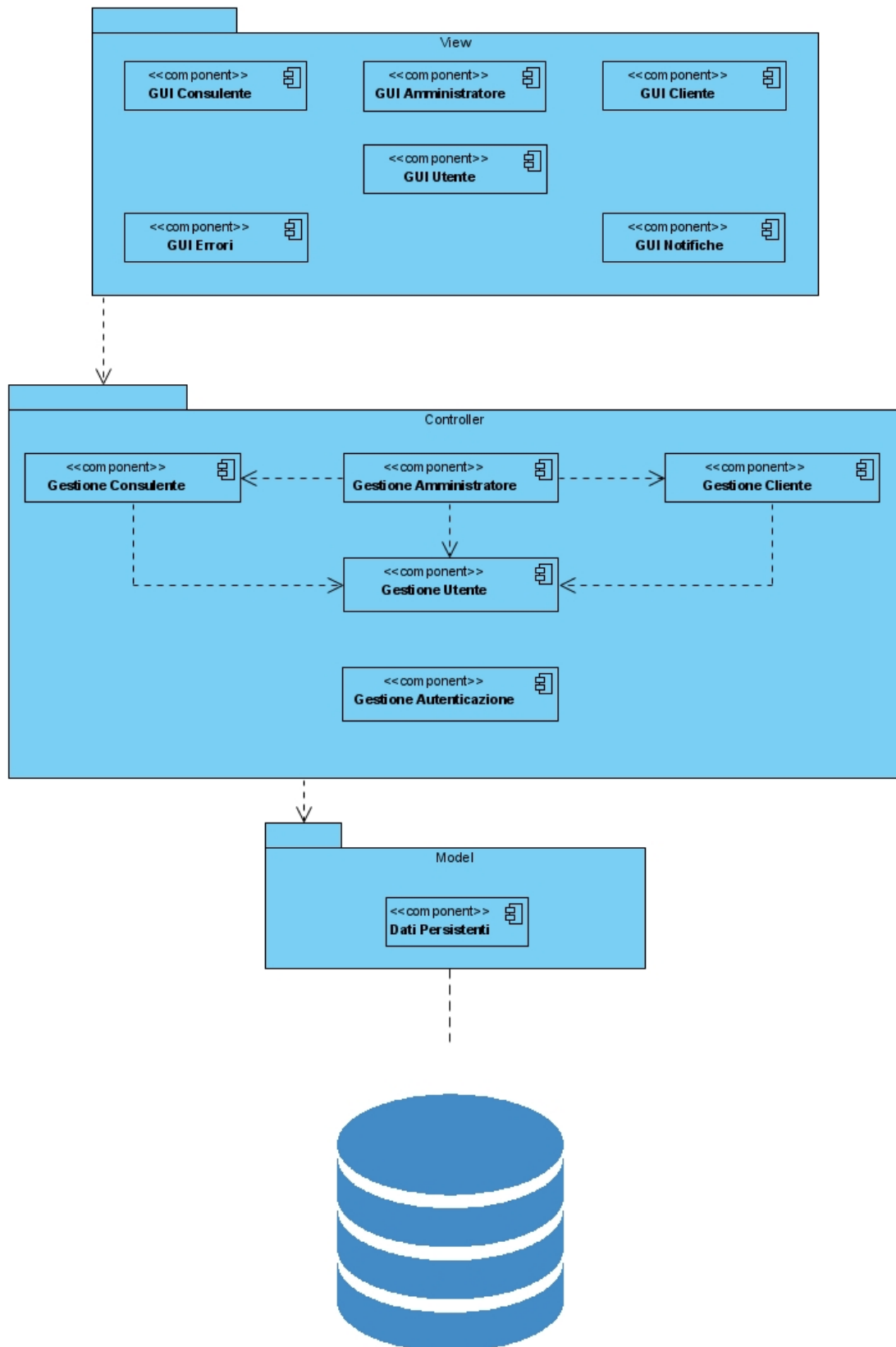
Il layer Model gestisce una sola componente, cioè quella che si occupa di gestire i dati persistenti nel sistema, interfacciandosi con il Database.

Il layer View gestisce sei componenti:

- GUI Amministratore
- GUI Consulente
- GUI Cliente
- GUI Utente
- GUI Errori

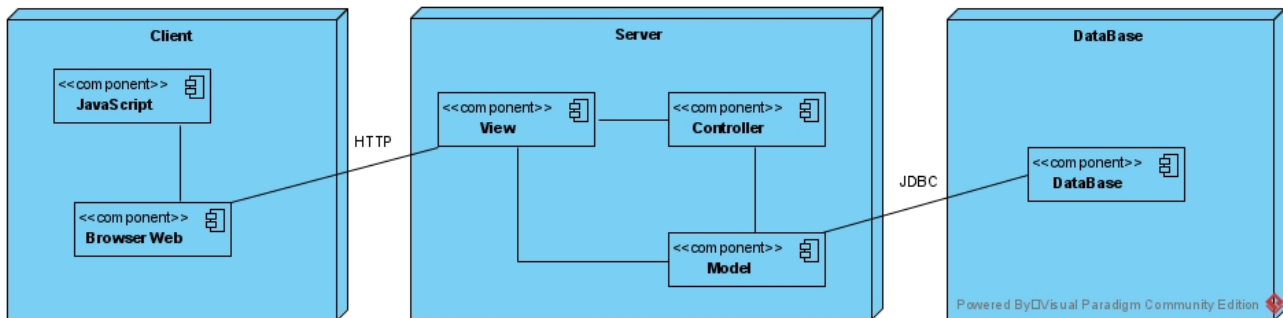


- GUI Notifiche



Deployment Diagram

L'utente, tramite un browser compatibile con l'interpretazione delle funzionalità di JavaScript, può richiedere le funzionalità del sistema tramite l'interfaccia presente sul Web. Le richieste dell'utente vengono ricevute ed elaborate poi dal sistema tramite il modello MVC, che si occupa di comunicare con il Database ed i dati persistenti.

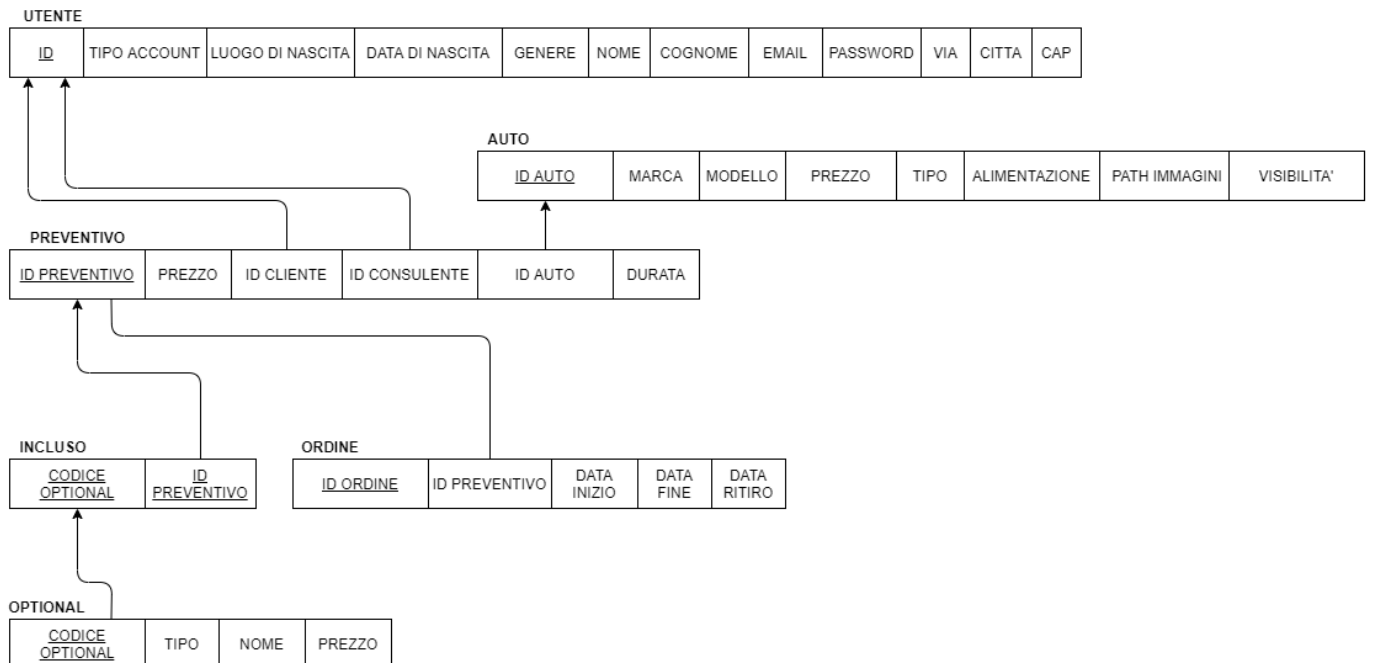
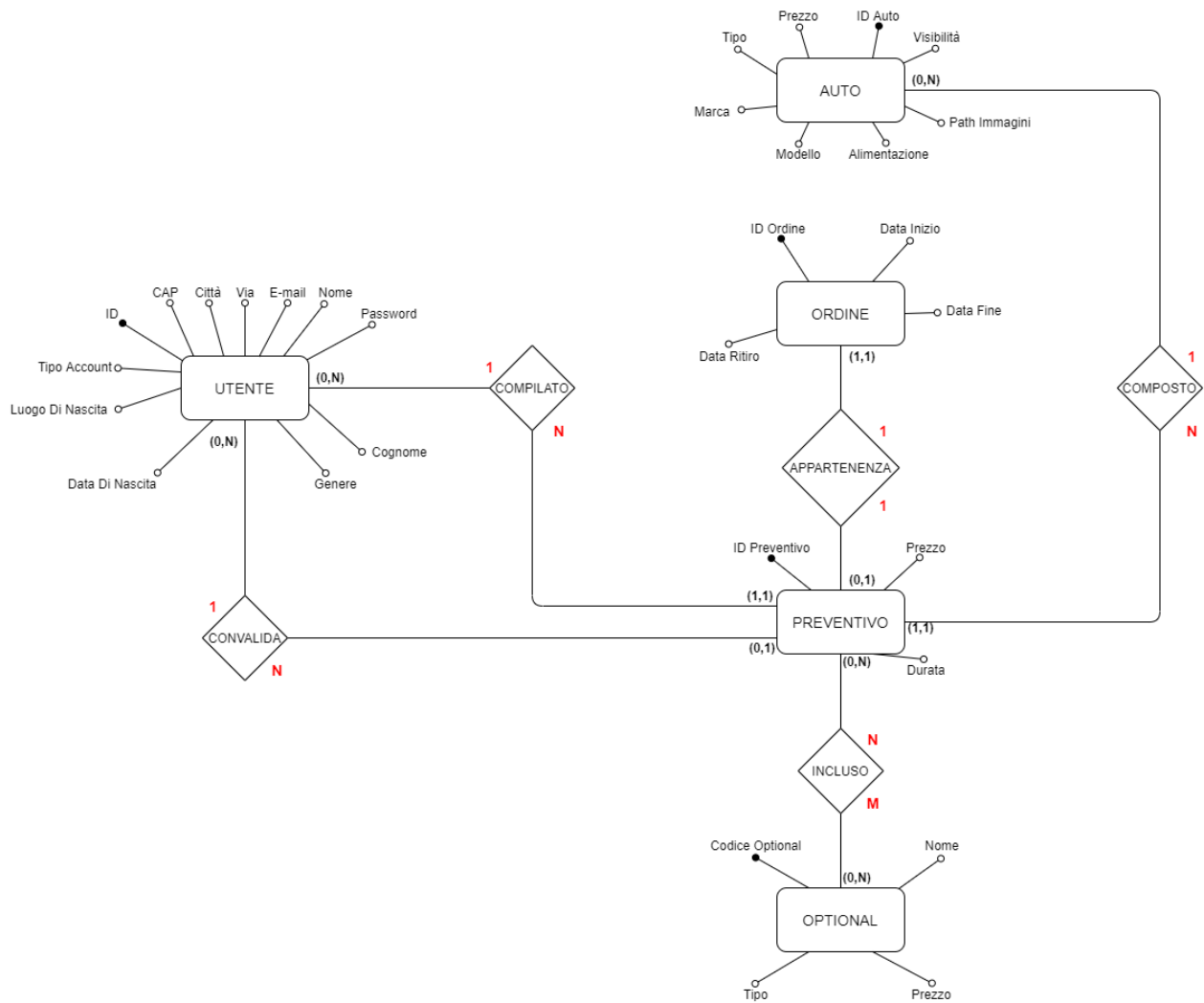


3.3 Mapping Hardware/Software

Il sistema proposto adopererà come fondamenta della propria struttura un sistema basato sull'architettura client-server, dove il server si occuperà di offrire informazioni e ogni tipologia di servizio disponibile ai client che ne faranno richiesta collegandosi ad esso. Il server sopracitato, per poter svolgere il proprio compito, dovrà predisporre di una connessione ad Internet con una grande larghezza di banda in modo da soddisfare eventuali picchi di richieste. Inoltre, il server dovrà disporre di una memoria abbastanza grande da poter contenere un alto numero di informazioni. Invece, il client per poter svolgere il proprio compito all'interno del sistema necessita delle seguenti architetture hardware e software: un computer, un tablet oppure uno smartphone, poiché di fatti il server potrà comunicare tramite il protocollo http con qualsiasi tipo di hardware, l'importante è che esso disponga di una connessione ad Internet e di un motore di ricerca che permetta di effettuare le richieste. Indipendentemente dalla disponibilità da parte di un qualsiasi client nell'effettuare delle richieste, il server dovrà sempre essere disponibile alla comunicazione con esso, il servizio quindi dovrà offrire un servizio attivo e disponibile in qualsiasi momento, per fare ciò il server dovrà utilizzare un web-server, che nel nostro caso sarà Tomcat. Inoltre, il server utilizzerà un DBMS per consentire la creazione, la manipolazione e l'interrogazione efficiente dei dati in modo persistente del database in uso, che nel nostro caso sarà MySQL.

3.4 Gestione dati persistenti

I dati nel nostro contesto verranno memorizzati con l'ausilio di un database relazionale per agire in un modo intuitivo e diretto con dati nelle tabelle e per gestire velocemente grandi quantità di dati. Per garantire che i dati siano sempre accurati e accessibili, i database relazionali seguono regole di integrità specifiche, difatti, l'integrità dei dati sarà poi garantita da backup periodici che in caso di perdita di dati fungeranno da supporto. Inoltre, i dati che si troveranno nel database saranno privatizzati, cioè verrà definito un accesso protetto a tali dati e quindi solo un amministratore potrà accedervi per apportare delle modifiche, evitando in questo modo possibili accessi pericolosi da parte di utenti non autorizzati.





UTENTE

NOME	TIPO	CONSTRAINTS	KEY
ID	VARCHAR(7)	NOT NULL	PRIMARY KEY
TIPO ACCOUNT	ENUM('CLIENTE', 'AMMINISTRATORE', 'CONSULENTE')	NOT NULL	
DATA DI NASCITA	DATE		
GENERE	ENUM('UOMO', 'DONNA', 'ALTRO', 'PREFERISCO NON SPECIFICARLO')		
NOME	VARCHAR(16)	NOT NULL	
COGNOME	VARCHAR(16)	NOT NULL	
EMAIL	VARCHAR(30)	NOT NULL	
PASSWORD	VARCHAR(16)	NOT NULL	
VIA	VARCHAR(16)		
CITTA	VARCHAR(16)		
CAP	VARCHAR(5)		

AUTO

NOME	TIPO	CONSTRAINTS	KEY
ID AUTO	VARCHAR(7)	NOT NULL	PRIMARY KEY
MARCA	VARCHAR(16)	NOT NULL	



MODELLO	VARCHAR(16)	NOT NULL	
PREZZO	FLOAT(8,2)	NOT NULL	
TIPO	VARCHAR(16)	NOT NULL	
ALIMENTAZIONE	VARCHAR(16)	NOT NULL	
PATH IMMAGINI	VARCHAR(16)	NOT NULL	
VISIBILITA	BOOLEAN	NOT NULL	

PREVENTIVO

NOME	TIPO	CONSTRAINTS	KEY
ID PREVENTIVO	VARCHAR(7)	NOT NULL	PRIMARY KEY
PREZZO	FLOAT(8,2)		
ID CLIENTE	VARCHAR(7)	NOT NULL	FOREIGN KEY
ID CONSULENTE	VARCHAR(7)	NOT NULL	FOREIGN KEY
ID AUTO	VARCHAR(7)	NOT NULL	FOREIGN KEY
DURATA	INTEGER	NOT NULL	

INCLUSO

NOME	TIPO	CONSTRAINTS	KEY
CODICE OPTIONAL	VARCHAR(7)	NOT NULL	FOREIGN KEY

ID PREVENTIVO	VARCHAR(7)	NOT NULL	FOREIGN KEY
----------------------	------------	----------	-------------

OPTIONAL

NOME	TIPO	CONSTRAINTS	KEY
CODICE OPTIONAL	VARCHAR(7)	NOT NULL	PRIMARY KEY
TIPO	VARCHAR(16)	NOT NULL	
NOME	VARCHAR(16)	NOT NULL	
PREZZO	FLOAT(8,2)	NOT NULL	

ORDINE

NOME	TIPO	CONSTRAINTS	KEY
ID ORDINE	VARCHAR(7)	NOT NULL	PRIMARY KEY
ID PREVENTIVO	VARCHAR(7)	NOT NULL	FOREIGN KEY
DATA INIZIO	DATE	NOT NULL	
DATA FINE	DATE	NOT NULL	
DATA RITIRO	DATE		

3.5 Controllo degli accessi e sicurezza

Nel sistema “EasyLease” il controllo degli accessi viene garantito grazie all'utilizzo dell'e-mail e la password, le quali permettono all'utente che si interfaccia con le funzionalità ad esso predisposte di creare o modificare gli oggetti che modellano le entità di dominio, così da poter evitare accessi non desiderati con la consecutiva violazione di informazioni sensibili.

Le operazioni che gli utenti del sistema possono effettuare sugli oggetti sono riportate nella seguente tabella, dove:



- In alto, sono presenti le astrazioni delle classi presenti nel sistema.
- Sul lato, sono presenti gli attori.
- La cella che incrocia l'istanza e l'attore rappresenta il permesso che quest'ultimo ha su quella determinata istanza.

Istanze Attori	Utente	Amministra tore	Consule nte	Cliente	Auto	Preventiv o	Ordine
Utente Non Registrato				Registrazion e Account	Ricerca Auto Consultazi one Catalogo Visualizza Modello		
Utente Registrato	Login Logout				Ricerca Auto Consultazi one Catalogo Visualizza Modello		
Amministra tore		Modifica Account Visualizza Dati Personalì	Aggiunta Consulen te Eliminazio ne Consulen te		Aggiunta Auto Modifica Auto Eliminazion e Auto		
Consulente			Modifica Account Visualizza Dati Personalì	Visualizzazio ne Clienti		Gestione Preventivo Stipulazion e Preventivo	Gestione Ordine Approvazi one Ordine
Cliente	Eliminazio ne Account			Modifica Account Visualizza Dati Personalì		Gestione Preventivo Richiesta Preventivo Conferma Preventivo	Gestione Ordine Conferma Ordine

3.6 Controllo flusso globale del sistema

Il sistema “EasyLease” adotta un controllo di flusso globale di tipo event-driven, ovvero che è richiesta l’interazione dell’utente con il sistema per poter innescare le risposte di quest’ultimo.

Questa soluzione, quindi, permette al sistema di poter soddisfare le richieste di molteplici utenti contemporaneamente, ma necessita che gli accessi per la scrittura dei dati persistenti vengano eseguite sequenzialmente, gestendo opportunamente la concorrenza. Tutto ciò è realizzabile tramite l’adozione di un thread dedicato per ogni utente che interagisce con il sistema.

3.7 Condizione limite

Identificativo <i>UC_BC_1</i>	System Start-up	Data	28/11/2020
		Vers.	0.00.001
		Autore	Sarro Antonio
Descrizione	Lo UC fornisce la funzionalità di avviare il server.		
Attore Principale	Amministratore		
Attori secondari	N.a.		
Entry Condition	L'amministratore ha accesso alla macchina sulla quale è installato il sistema.		
Exit condition <i>On success</i>	Il server viene avviato correttamente.		
Exit condition <i>On failure</i>	Il server non viene avviato.		
Rilevanza/User Priority	Elevata		
Frequenza stimata	1 usi/anno		
Extension point	N.a.		
Generalization of	N.a.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	Richiede al sistema di avviare il DBMS per poter accedere alle risorse salvate nel database tramite l'apposito comando.	
2	Sistema:	Avvia il DBMS.	
3	Sistema:	Mostra un messaggio all'amministratore notificando che il DBMS è stato avviato con successo.	
4	Amministratore:	Richiede al sistema di avviare il web container tramite l'apposito comando.	
5	Sistema:	Avvia il web container.	
6	Sistema:	Mostra un messaggio all'amministratore notificando che il web container è stato avviato con successo.	
I Scenario/Flusso di eventi di ERRORE: Il sistema non riesce ad avviare il DBMS			
2.a1	Sistema	Mostra un messaggio di errore all'amministratore. Il messaggio segnala che non è stato possibile effettuare l'azione richiesta.	
2.a2	Sistema	Termina con un insuccesso.	
Il Scenario/Flusso di eventi di ERRORE: Il sistema non riesce ad avviare il web container			



5.a1	Sistema	Mostra un messaggio di errore all'amministratore. Il messaggio segnala che non è stato possibile effettuare l'azione richiesta.
5.a2	Sistema	Termina con un insuccesso.
Note		
Special Requirements		
1	N.a.	

Identificativo <i>UC_BC_2</i>	System shut-down		Data	28/11/2020
			Vers.	0.00.001
			Autore	Sarro Antonio
Descrizione		<i>Lo UC fornisce la funzionalità di terminare il server.</i>		
Attore Principale		Amministratore		
Attori secondari		N.a.		
Entry Condition		L'amministratore ha accesso alla macchina sulla quale il sistema è in esecuzione.		
Exit condition <i>On success</i>		<i>Il server viene terminato correttamente.</i>		
Exit condition <i>On failure</i>		Il server non viene terminato.		
Rilevanza/User Priority		Elevata		
Frequenza stimata		1 usi/anno		
Extension point		N.a.		
Generalization of		N.a.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Amministratore:	Richiede al sistema di terminare il web container tramite l'apposito comando.
2	Sistema:	Termina il web container.
3	Sistema:	Mostra un messaggio all'amministratore notificando che il web container è stato terminato con successo.
4	Amministratore:	Richiede al sistema di terminare il DBMS tramite l'apposito comando.
5	Sistema:	Termina il DBMS.
6	Sistema:	Mostra un messaggio all'amministratore notificando che il DMS è stato terminato con successo.

I Scenario/Flusso di eventi di ERRORE: Il sistema non riesce a terminare il web container

2.a1	Sistema	Mostra un messaggio di errore all'amministratore. Il messaggio segnala che non è stato possibile effettuare l'azione richiesta.
2.a2	Sistema	Termina con un insuccesso.

Il Scenario/Flusso di eventi di ERRORE: Il sistema non riesce ad avviare il DBMS

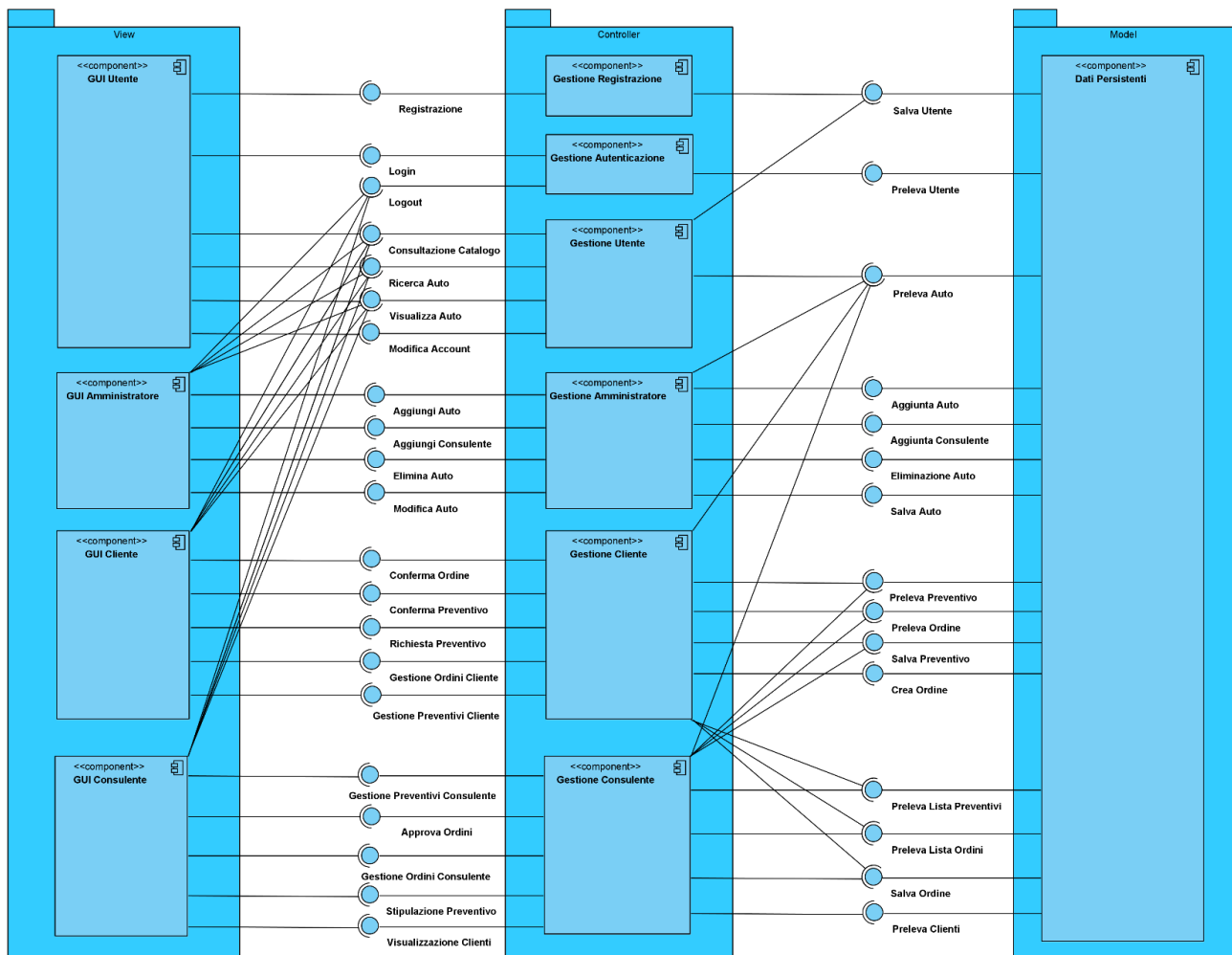
5.a1	Sistema	Mostra un messaggio di errore all'amministratore. Il messaggio segnala che non è stato possibile effettuare l'azione richiesta.
5.a2	Sistema	Termina con un insuccesso.



Note			
Special Requirements			
1	N.a.		

Identificativo <i>UC_BC_3</i>	System Faliure	Data	28/11/2020
		Vers.	0.00.001
		Autore	Sarro Antonio
Descrizione	Lo UC fornisce al sistema la funzionalità di notificare eventuali errori ad un utente.		
Attore Principale	Amministratore		
Attori secondari	Utente		
Entry Condition	Il sistema gestisce l'errore e lo notifica all'utente.		
Exit condition <i>On success</i>	Il sistema gestisce l'errore e lo notifica all'utente.		
Exit condition <i>On failure</i>	L'errore non viene gestito dal sistema.		
Rilevanza/User Priority	Elevata		
Frequenza stimata	10 usi/anno		
Extension point	N.a.		
Generalization of	N.a.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Rileva l'errore avvenuto e salva la sessione attuale.	
2	Sistema:	Mostra un messaggio all'utente che segnala il problema.	
3	Utente:	Prende visione del messaggio.	
Note			
Special Requirements			
1	N.g.		

4. Servizi dei sottosistemi



Servizi offerti da Controller per View

Servizi offerti per GUI Amministratore

- Logout
- Aggiungi Auto
- Consultazione Catalogo
- Aggiungi Consulente
- Elimina Auto
- Modifica Auto
- Ricerca Auto
- Visualizza Auto

Servizi offerti per GUI Cliente



- Conferma Ordine
- Conferma Preventivo
- Consultazione Catalogo
- Ricerca Auto
- Richiesta Preventivo
- Visualizzazione Auto
- Logout

Servizi offerti per GUI Utente

- Consultazione Catalogo
- Login
- Registrazione
- Ricerca Auto
- Visualizza Auto
- Modifica Account

Servizi offerti per GUI Consulente

- Approva Ordini
- Consultazione Catalogo
- Ricerca Auto
- Gestione Ordini
- Stipulazione Preventivo
- Visualizzazione Clienti
- Logout

Servizi offerti da Model per Controller

Servizi offerti per Gestione Autenticazione

- Preleva Utente

Servizi offerti per Gestione Amministratore

- Aggiunta Auto



- Aggiunta Consulente
- Preleva Auto
- Eliminazione Auto
- Salva Auto

Servizi offerti per Gestione Cliente

- Preleva Preventivo
- Preleva Ordine
- Preleva Auto
- Salva Preventivo
- Crea Ordine
- Salva Ordine
- Preleva Lista Ordini
- Preleva Lista Preventivi

Servizi offerti per Gestione Utente

- Preleva Auto
- Salva Utente

Servizi offerti per Gestione Consulente

- Preleva Ordine
- Preleva Auto
- Salva Preventivo
- Preleva Clienti
- Preleva preventivo
- Preleva Lista Ordini
- Preleva Lista Preventivi