POLITECNICO
DI MILANO

Software Engineering II

myTaxiService

DD

Design Document

Authors:

Hamidreza Hanafi

Navid Heidari

December 4th 2015

Table of Contents

TABLE OF CONTENTS 2

1. INTRODUCTION 5

1.1 PURPOSE OF DOCUMENT 5

1.2 SCOPE 5

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS 5

1.4 REFERENCE DOCUMENTS 5

1.5 DOCUMENT STRUCTURE 6

2. ARCHITECTURE DESCRIPTION 6

2.1 JEE ARCHITECTURE 6

2.2 IDENTIFYING SUB-SYSTEMS 7

2.3 PERSISTENT DATA MANAGEMENT 9

2.3.1 CONCEPTUAL DESIGN 9

2.3.2 LOGICAL DESIGN 10

2.4 USER EXPERIENCE 11

2.5 BCE 13

3. ALGORITHM DESIGN 15

4. USER INTERFACE DESIGN 15

5. REQUIREMENTS TRACEABILITY 15

6. TOOLS 15

1. INTRODUCTION

1.1 PURPOSE OF DOCUMENT

This document represents the design description of the “myTaxiService” project assigned during the course of Software Engineering II at Politecnico di Milano. Indeed it contains the functional description of the system with using of diagrams (such UML diagrams).

1.2 SCOPE

The scope of this document is to give a representation of the software design to be used for recording design information and communicating them. The architectural descriptions concern the data layer, business logic and the user interface described in the RASD.

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

- RASD : Requirements Analysis and Specification Document
- JEE : Java Enterprise Edition
- DB : Data Base
- DBMS : Data Base Management System
- DD : Design Document
- EIS : Enterprise information system
- BCE : Boundary-Control-Entity
- UX : User eXperience

1.4 REFERENCE DOCUMENTS

Here is the list of the references which are used in this document:

- Java EE - Distribuited Multitiered Applications:

  http://docs.oracle.com/javaee/6/tutorial/doc/bnaay.html (Dec 4th 2015);

- Requirements Analysis and Specification Document : RASD.pdf;
- A template for creating this document (DD) provided by Prof. Di Nitto, during the Software Engineering II lectures.

1.5 DOCUMENT STRUCTURE

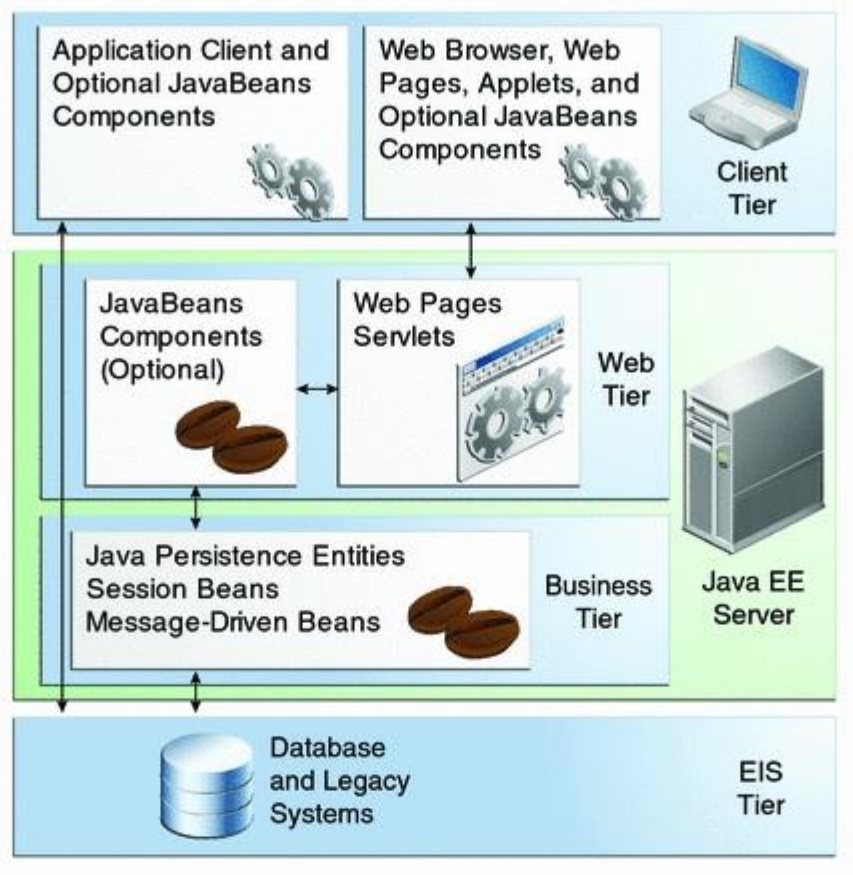
.

2. ARCHITECTURE DESCRIPTION

2.1 JEE ARCHITECTURE

The Java EE platform uses a distributed multi-tiered application model for enterprise applications. Application logic is divided into components according to function, and the application components that make up a Java EE application are installed on various machines, depending on the tier in the multi-tiered Java EE environment to which the application component belongs. Java EE applications divided into the tiers described in the following list:

- Client-tier components run on the client machine.
- Web-tier components run on the Java EE server.
- Business-tier components run on the Java EE server.
- Enterprise information system (EIS)-tier software runs on the EIS server.


Application Client and Optional JavaBeans Components
Web Browser, Web Pages, Applets, and Optional JavaBeans Components
Client Tier
JavaBeans Components (Optional)
Web Pages Servlets
Web Tier
Java Persistence Entities Session Beans Message-Driven Beans
Business Tier
Java EE Server
Database and Legacy Systems
EIS Tier

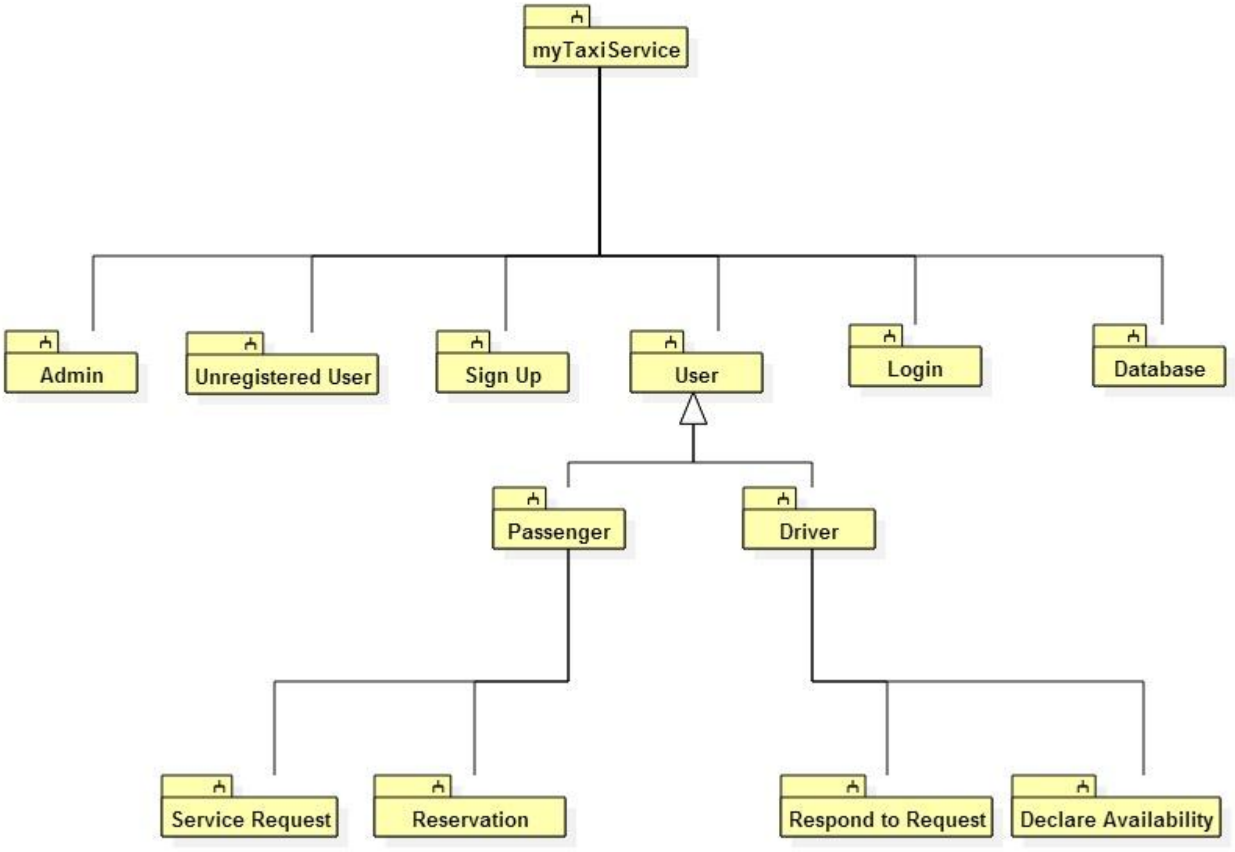
2.2 IDENTIFYING SUB-SYSTEMS

Here we can see the Sub-Systems in our project:

- Sign up sub-system;
- Log in sub-system;
- Unregistered User;
- Admin;
- Database;
- User Sub-Systems;
  - Passenger
  - Driver

- Passenger Sub-Systems
  - Service Request
  - Reservation

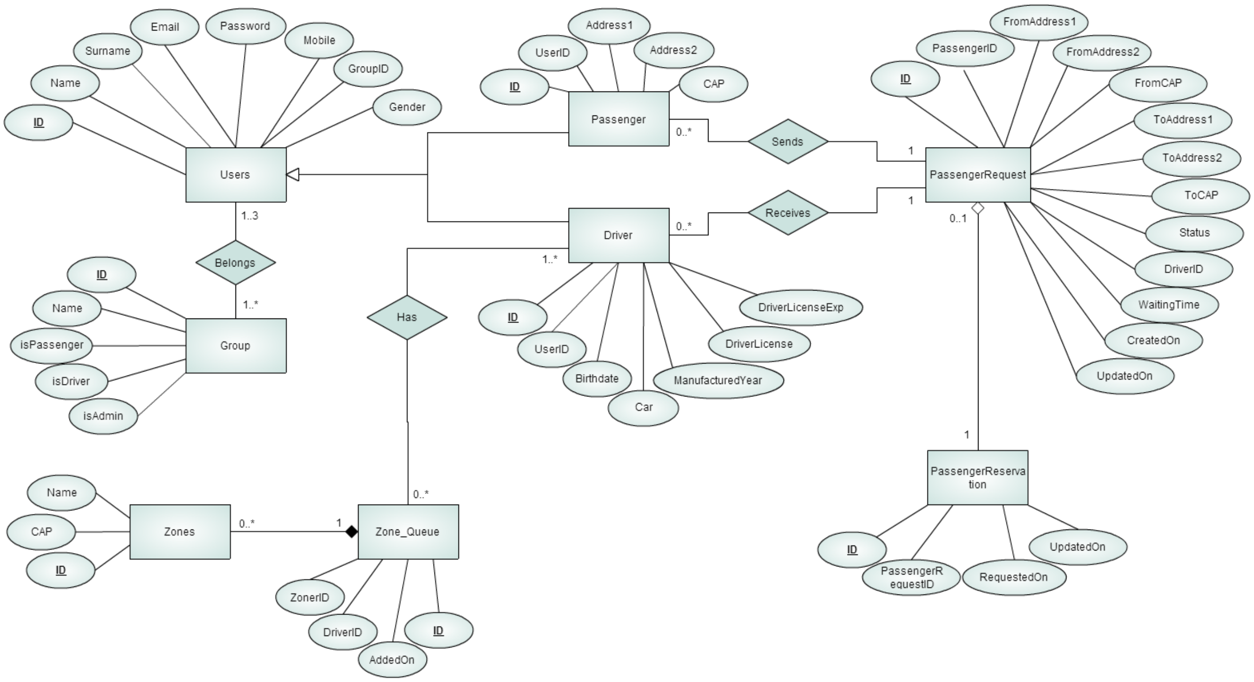
- Driver Sub-System
    - Respond to Request
    - Declare Availability



2.3 PERSISTENT DATA MANAGEMENT

2.3.1 CONCEPTUAL DESIGN

Here we present the relation between entities and introduce their attributes.

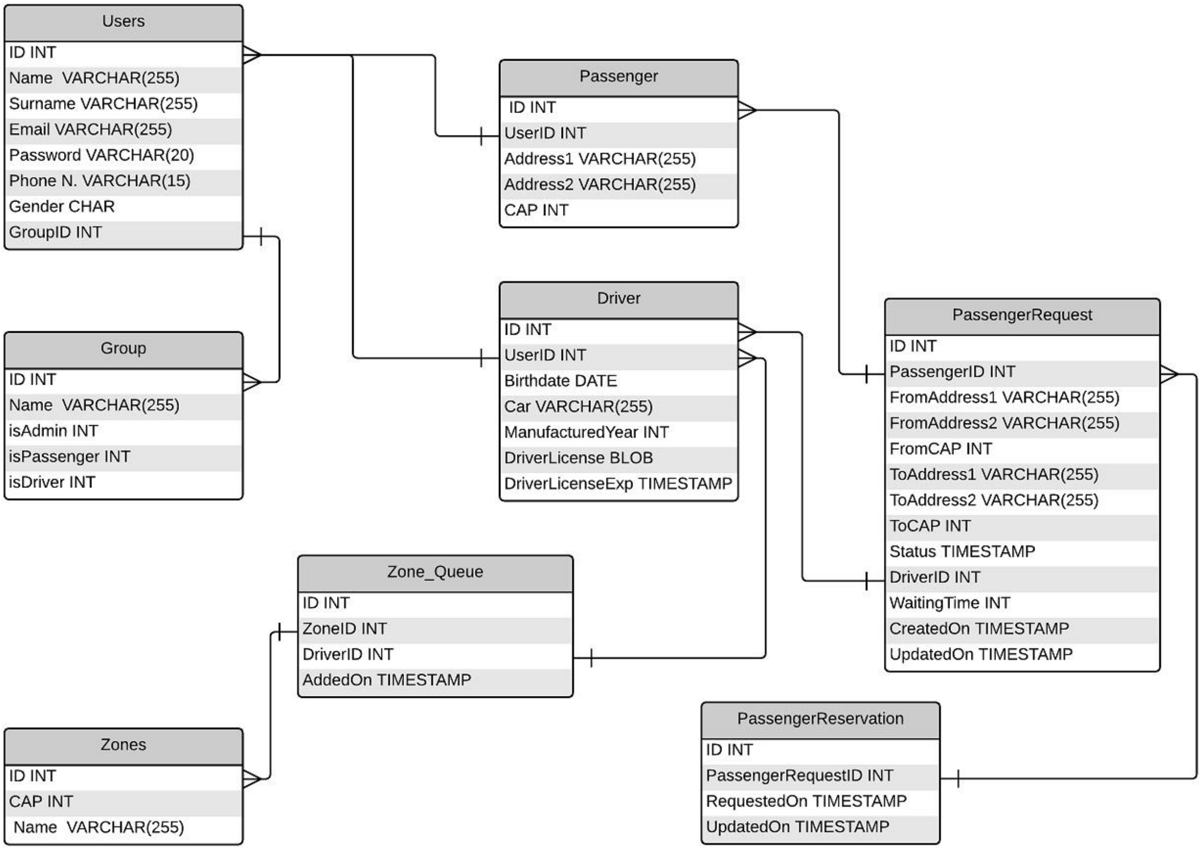


Here is better to describe the relations between Entities:

- *Belongs:* there are three kinds of user, (admin, passenger and driver), so each user belongs to one of these groups.
- *Has:* each Zone-Queue has a queue of available drivers of that specific zone at that certain moment.
- *Sends and Receives:* Passengers can send their request (either service or reservation) to the system and Drivers can receive requests from the system.

2.3.2 LOGICAL DESIGN

This is the Logical Model of the project according to the ER model. In the each table, ID will be the primary key of the table.



Users
ID INT
Name VARCHAR(255)
Surname VARCHAR(255)
Email VARCHAR(255)
Password VARCHAR(20)
Phone N. VARCHAR(15)
Gender CHAR
GroupID INT

Passenger
ID INT
UserID INT
Address1 VARCHAR(255)
Address2 VARCHAR(255)
CAP INT

Group
ID INT
Name VARCHAR(255)
isAdmin INT
isPassenger INT
isDriver INT

Driver
ID INT
UserID INT
Birthdate DATE
Car VARCHAR(255)
ManufacturedYear INT
DriverLicense BLOB
DriverLicenseExp TIMESTAMP

PassengerRequest
ID INT
PassengerID INT
FromAddress1 VARCHAR(255)
FromAddress2 VARCHAR(255)
FromCAP INT
ToAddress1 VARCHAR(255)
ToAddress2 VARCHAR(255)
ToCAP INT
Status TIMESTAMP
DriverID INT
WaitingTime INT
CreatedOn TIMESTAMP
UpdatedOn TIMESTAMP

Zone_Queue
ID INT
ZoneID INT
DriverID INT
AddedOn TIMESTAMP

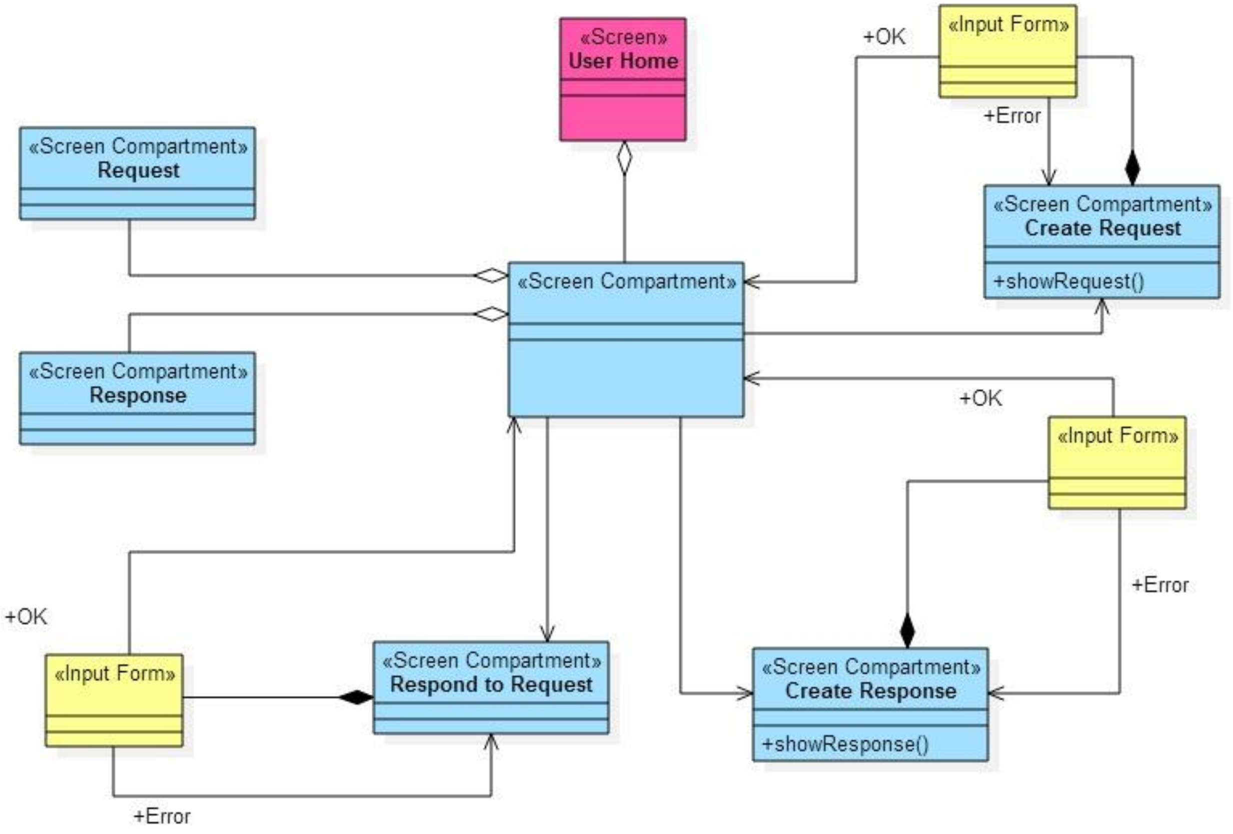
Zones
ID INT
CAP INT
Name VARCHAR(255)

PassengerReservation
ID INT
PassengerRequestID INT
RequestedOn TIMESTAMP
UpdatedOn TIMESTAMP

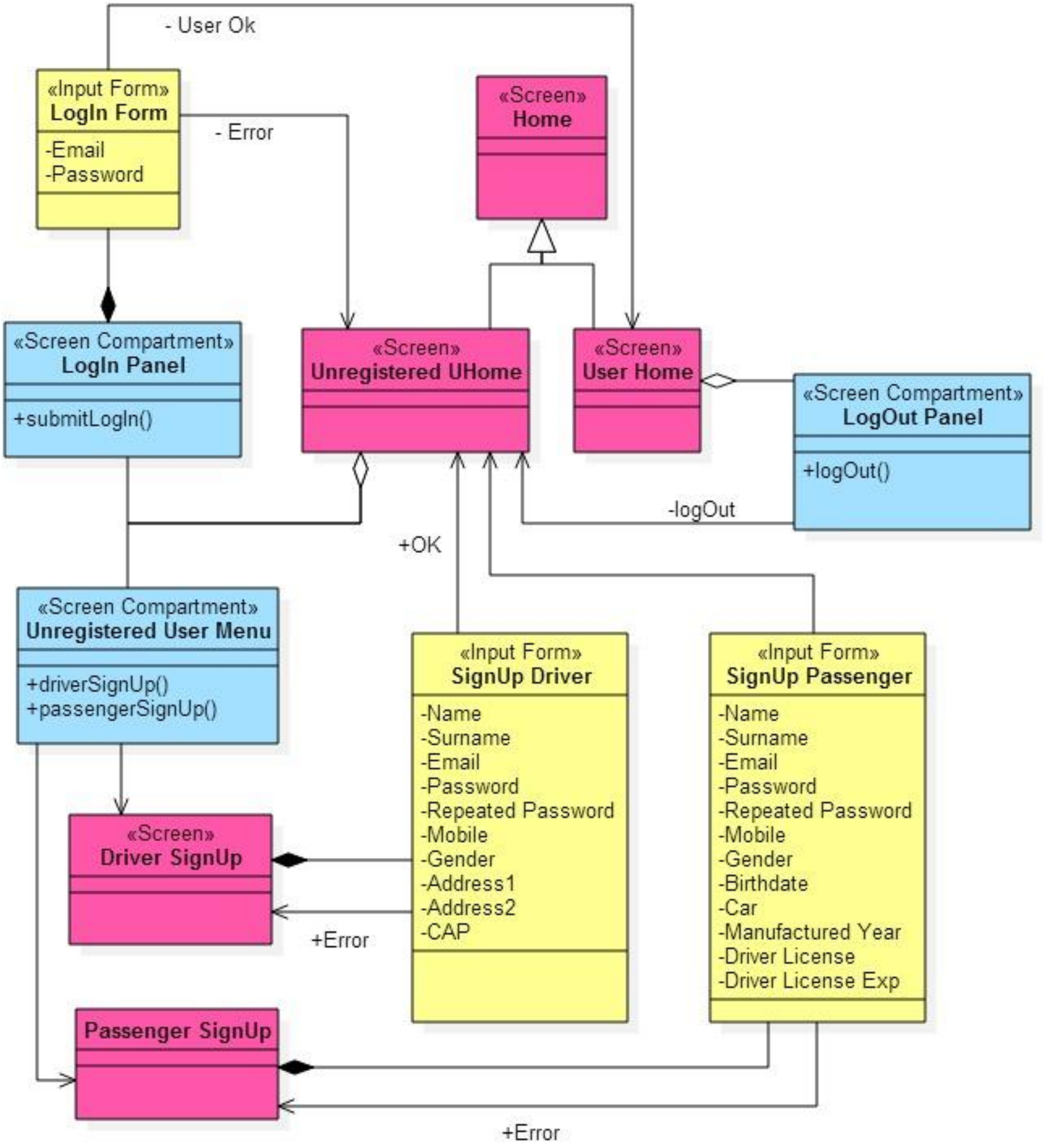
2.4 USER EXPERIENCE

In this part of the document we will see the User Experience (UX) given by the system to users. The class diagrams consist of some stereotypes <<Screen>>, <<Screen Compartment>> and <<Input Form>> and also normal classes. <<Screen>> represents page, <<Screen Compartment>> represents parts of the page that can be shared with others. <<Input Form>>, represents some input fields that can be fulfilled by a user. Here we see two UX diagrams:

Here is the UX of Response to a Request by Driver:



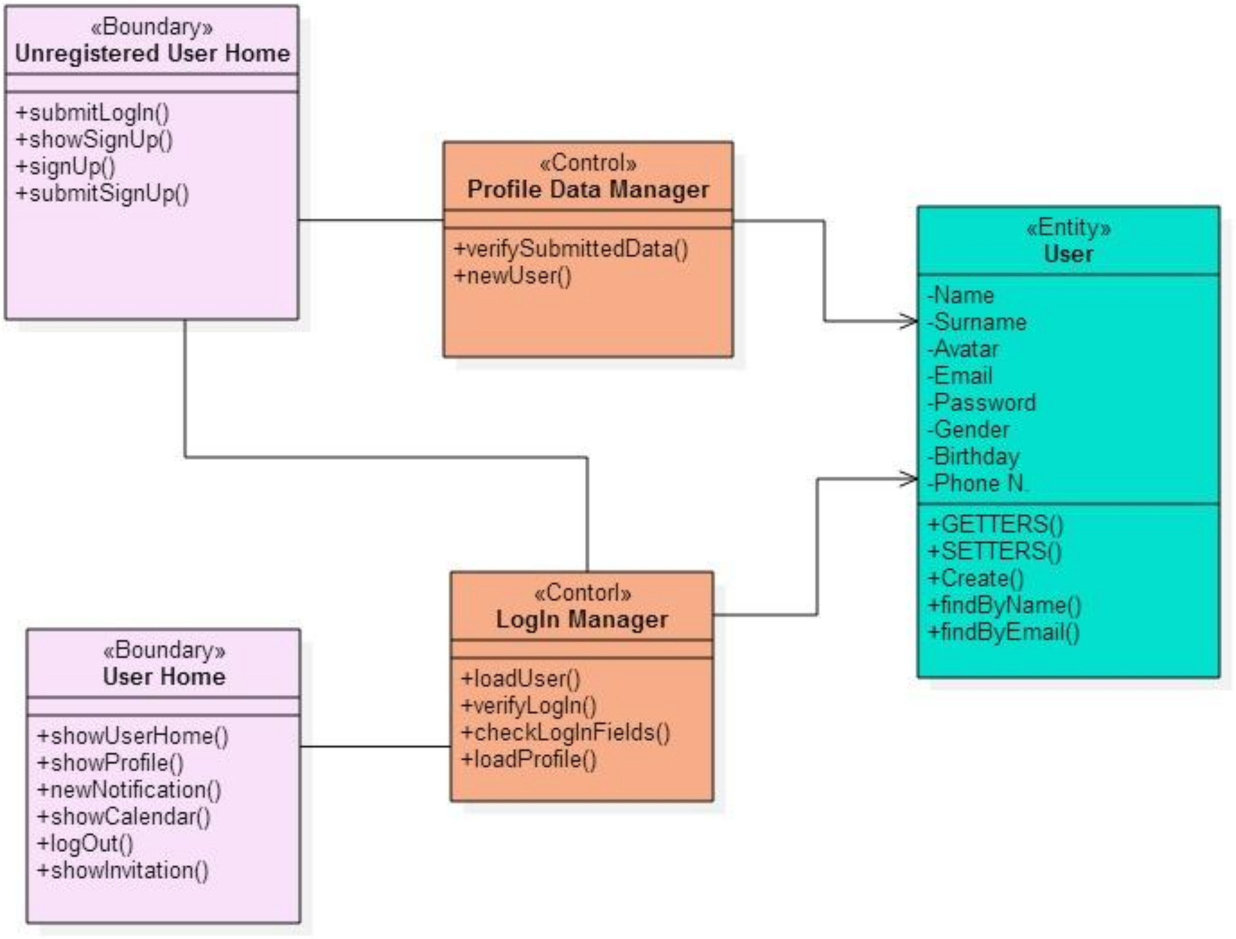
This one shows the UX of Sign Up and Log In to the profile:



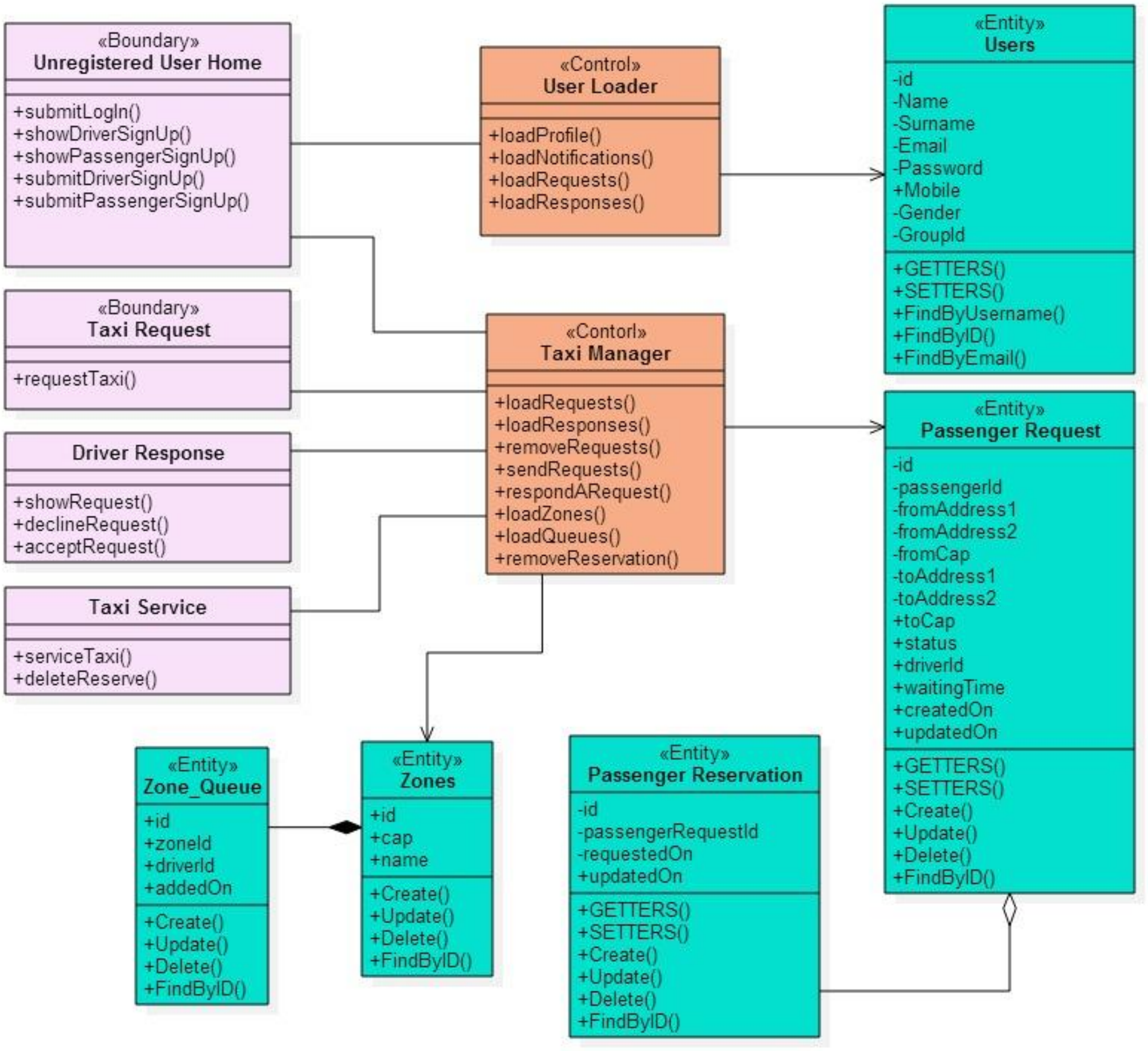
2.5 BCE

This representation is realized through the UML diagram BCE (Boundary-Control-Entity). Boundary refers to the presentation level, to what user can see. It is created starting from the UX model, grouping all functionalities with the same thematic area. Control is a 'broker' between the user interface and the data, represented through the Entity.

Here is the figure of BCE of Sign Up and Log In:



And here is the figure of Taxi Service Request :



3. ALGORITHM DESIGN

4. USER INTERFACE DESIGN

5. REQUIREMENTS TRACEABILITY

6. TOOLS

- https://www.lucidchart.com (for ER and Logic Model)
- https://cacoo.com (for ER and Logic Model)
- StarUML (Sub-Systems, UX, BCE and Sequence Diagrams)