MyTaxiService

Design Document

Software Engineering 2

Authors:

Hasan Aliyev
Madat Mustafayev

Milan —2015

Keynote

Contents

Contents

1. Introduction

1.1 Scope

This document is prepared considering requirements analysis and specification document of MyTaxiservice Project which is published before. This document covers software design stage of Taxi Project and it also contains important information relevant to software implementation. System structure and the components of the project is also described in this document.

1.2 Purpose

The purpose of this software design descriptions document is to describe how the software will be structured to satisfy the requirements of the MyTaxiservice. This document will explain the design details of the system.

1.3 Intended audience

The intended audience of this software design descriptions document is both the developers who will build the system and the stakeholders

2. Definition

Terms	Definitions
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
SDD	Software Design Description
SRS	Software Requirements Specification
UML	Unified Modeling Language
Args	Arguments
Use Case Diagram	it represents user's interaction with the system
Deployment Diagram	It models the physical deployment of artefacts on nodes
Component Diagram	It is used to illustrate the structure of arbitrarily complex systems
State Diagram	It is used to dene the behavior of the system
Activity Diagram	Activity diagrams are graphical representations of workows of stepwise activities and actions
Class Diagram	It describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects

3.Conceptual model for software design descriptions

3.1 Software design in context

In MyTaxiservice project, object oriented approach will be used as a design method. Sequently, it will be easier to implement the project and add possible future features. Since this project is a social oriented application, this property is critically important. Furthermore multi-layered system architecture will be used. Database, business and client layers will help modularity and adaptability of the software. With object oriented design and multi-layered architecture, portability and integrability between components will be improved.

3.2 Software design descriptions within the life cycle

3.2.1 Influences on DD preparation

The key software life cycle product that drives a software design is typically the requirements analysis and specification (RASD) document. RASD captures the software requirements (product perspective, functional and non-functional requirements and interface requirements) that will drive the design and it constraints to be considered or observed.

3.2.2 Influences on software life cycle products

The software Design document influences the content of several major software life cycle work products. During the preparation of DD and during the implementation stage of the project requirements may change. Therefore, DD influences will lead to requirements changes. However, DD will impact test plans and test documentation of the Online Taxi project.

3.2.3 Design verification and design role in validation

Test cases will be prepared after the development phase. Verification of the software will be tested with these test cases and all parts will be evaluated. Success of the software system will be determined with test cases. After the test results, validation of the software will be checked if requirements of the system are fulfilled or not.

4. Design description information content

4. 1 Introduction

Software Design Document of MyTaxiService project identifies how this system will be implemented and designed. Online Taxi project has a modular object-oriented structure. Furthermore a qualified interface will be designed in a way that system will represent events of the simulation successfully

4. 2 DD identification

This DD contains these features:

- Summary
- Glossary
- Change history
- Date of issue and status
- Scope
- Issuing organisation
- Authorship (responsibility or copyright information)
- References
- Context
- One or more design languages for each design viewpoint used
- Body

4.3 Design stakeholders and their concerns

Design stakeholders of MyTaxiService project is the developer team of the system and their advisors. Design concern of the stakeholders is implementing the project in a modular structure according to Design Document. End product has to contain design features that are described at Design Document. Modular approach is essential for the project since there will be multi types of clients; namely, web client and mobile clients.

4. Design description information content

4. 4 Design views

This Project will be implemented as web and mobile service application with simple interface. In this document contextual, composition, interface, logical, interaction and state dynamics view will be explained in next sections. Detailed description and diagrams about these views will clarify them. Each view is given with its corresponding viewpoint.

4. 5 Design Viewpoints

Software design description identifies context, composition, interface, logical, interaction and state dynamics viewpoints. Context viewpoint species the system boundaries and actors interacting with the system. Composition viewpoint identifies the system modules, components, frameworks and system repositories. Interface viewpoint explains the interaction between different software interfaces. Logical viewpoint includes the detailed description of data design and class diagrams. Interaction viewpoint gives the sequence of events in the system and the state dynamics viewpoint models the system as a state machine and shows the state transitions and conditions on these transitions.

4.6 Design rationale

In this project, design choices are made according to performance concerns and integrability of the system. System has to be designed in a way that future models and features can be added and current models can be changed and updated independently. Stakeholders may have and request further requirements, therefore system parts have to be modular. Developers of the system has to document development process and use comments in their code frequently, so that in the future other developers may understand code and the structure of the system.

4.7 Design languages

In this project, Unified Modelling Language (UML) is selected as a part of design viewpoint and it will be used for clarifying design viewpoints..

5. Design Viewpoints

5.1 Introduction

In this part we are going to explain 6 main design viewpoints

- Context Viewpoint
- Composition Viewpoint
- Logical Viewpoint
- Dependency Viewpoint
- Interaction Viewpoint
- State Dynamics Viewpoint

During this section, UML diagrams will be used to increase understandability.

5.2 Context Viewpoint

User functions:

Register to the system: The user will be able to register into system manually to access feature of the application.

Login: The user will be able to login to the system after successful registration.

Create new Order: The user will be able to create new order.

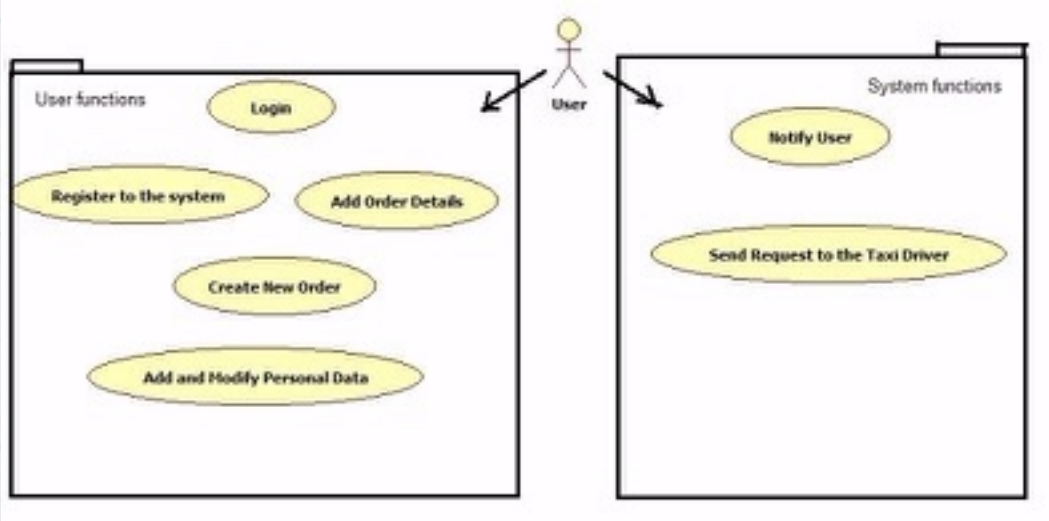
Add and Modify Personal Data: User will be able add and modify his personal information. It includes profile photo, credit/debit card, phone number etc.

Add Order Details: User will be able to add order details. It includes Pickup and Destination Locations, Car Class.

System Functions:

Notify users: The system will notify user after getting order from customer side and also confirmation from driver side, the system can notify the customer before 2 hours of its trip and also 10 minutes before

Send Request to Taxi Driver: The System will send the request to the taxi driver about new order. If taxi driver accepts the request order will be completed, otherwise, the request will have been sent to the other taxi driver in queue.



5. Design Viewpoints

5.3 Composition Viewpoint

Our application is consists of 2 part. User Mobile application and Web application. Mobile application part is composed of following components. We are going use Java, for IOS C++. For database system we will use local database SQLite and RSA as an Encrypting algorithm for data transfer Database is placed at the lowest level of the system. For Web application we will use HTTPS, Server side scripting is Java JSP or JSF, DBS MYSQL, Client side scripting is JavaScript, jQuery. The application will work on the Glassfish Server v4.1. (Web Services on Java).

5. 4 Logical Viewpoint

This part of the Design Document explains the classes, which we will use in our project and the relationship between classes. To begin with, all classes are explained with class diagram separately. After that the entity relationship diagram, which defines the database structure of our project, will be provided. There are four classes which are User, Order a Taxi, View and edit profile, and Taxi queue. In the following sections, we will explain each class separately.

.

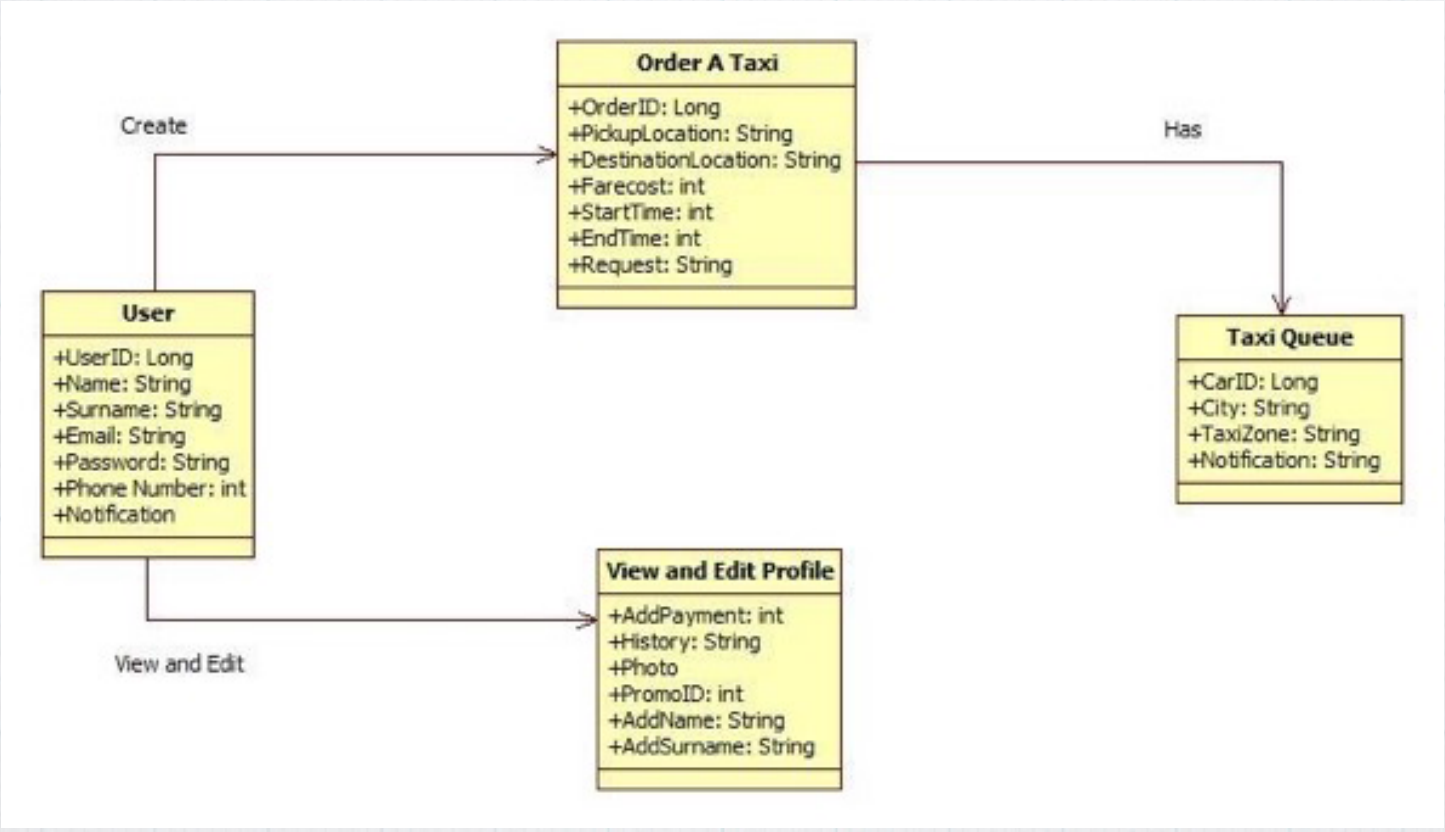
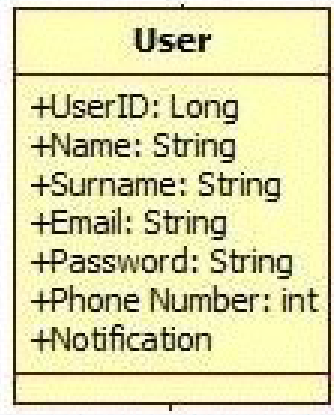


Figure 1. Logical Viewpoint

5. Design Viewpoints

5. 4. 1 User Class

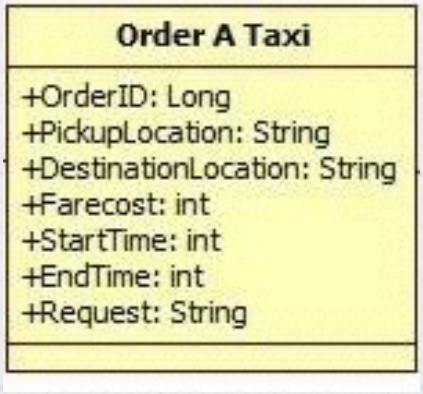
User class will handle user related data operations of the system and contain all user information with invited participated and created events



Name	Type/Return Value Type	Visibility	Definition
User ID	Long	Private	Unique id of the user
Name	String	Private	Name of the user
Surname	String	Private	Surname of the user
Email	String	Private	Unique email address of the user
Password	String	Private	Password of the user account
Phone Number	Integer	Private	Unique phone num. of the user
Notification	String	Private	Notifications specific to this user

5. 4. 2 Taxi order class

Taxi Order class will handle order related data operations of the system and contains information about order.



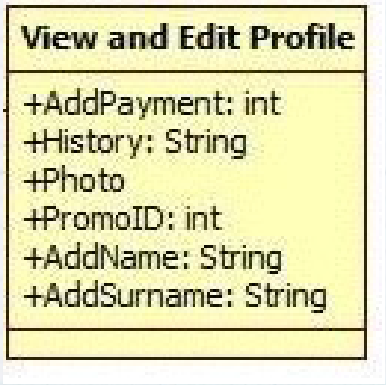
5. Design Viewpoints

5. 4. 2 Taxi order class

Name	Type/Return Value Type	Visibility	Definition
Order_ID	Long	Private	Unique ID of order
PickupLoc.	String	Private	Location that identifies the beginning of the route.
DestinationLoc.	String	Private	Location that identifies the ending of the route.
Farecost	Integer	Private	Estimated cost of the order
StartTime	Integer	Private	The beginning time of the route
EndTime	Integer	Private	The ending time of the route
Request	Integer	Public	Request is sent to Taxi Driver

5. 4. 3 View and Edit profile

View and Edit profile class will handle account information of customers.



Name	Type/Return Value Type	Visibility	Definition
AddPayment	Integer	Private	Payment method of Customer
History	String	Private	Shows previous orders of Customer
Photo	*Jpeg file	Private	Profile Photo of Customer
PromoID	Integer	Private	Unique ID of Promo Code
AddName	String	Private	Name of Customer
AddSurname	String	Private	Surname of Customer

5. Design Viewpoints

5. 5 Dependency Viewpoint

In this viewpoint the relationships of interconnections and access among packages are explained in detail and with UML component diagram.

Three-tier architecture is composed of three main packages. These are presentation tier, middle tier and data management tier.

- **Presentation Tier**: A layer that users can access directly, such as web page and client application.
- **Middle Tier**: This layer encapsulates the business such as business rules and data validation, domain concept, data access logic.
- **Data Management Tier**: The external data source to store the application data such as database server, mainframe or other legacy systems. The one we meet often today is database server.

In our system, we can divide into three basic packages. These are User Interface, Business Logic and Database. The explanation of these packages is below:

User Interface (Presentation Tier): It is top most module of our system. Basically its function is translating tasks and showing the results to the user. In User Interface package we have only one module:

- **View**: The view can be separated into two main modules according to usage. Actually, all of these have almost same features. The View renders a presentation of modelled data. We will use HTML5 technology for client web interface and Java for mobile application

5. Design Viewpoints

5. 5 Dependency Viewpoint

Business Logic (Middle Tier): It is the transition module of our system. The Business Logic coordinates the application, processes commands, makes logical decision and evaluation and performs calculations. It also moves and processes data between the modules, User Interface and Database in our system.

- **Model**: The model represents the part of our application that implements the business logic. It is responsible for the retrieving data and converting it into meaningful data for our application.This includes processing, validating, association or other tasks relate to handling data. We will use Java programming language for this module.
- **Controller**: The Controller handles requests from user. It is responsible for rendering back a response with the aid of Model module. Controller can be seen as managers taking care that all needed resources for completing a task are delegated to the correct workers. We will use Java programming language for this module.

Database (Data Management Tier) : It is the lowest module in our system. In this module the information is stored and retrieved from a database. The information is then passed back to the Business Logic for processing and then eventually back to the user. We will use MySQL for database package.

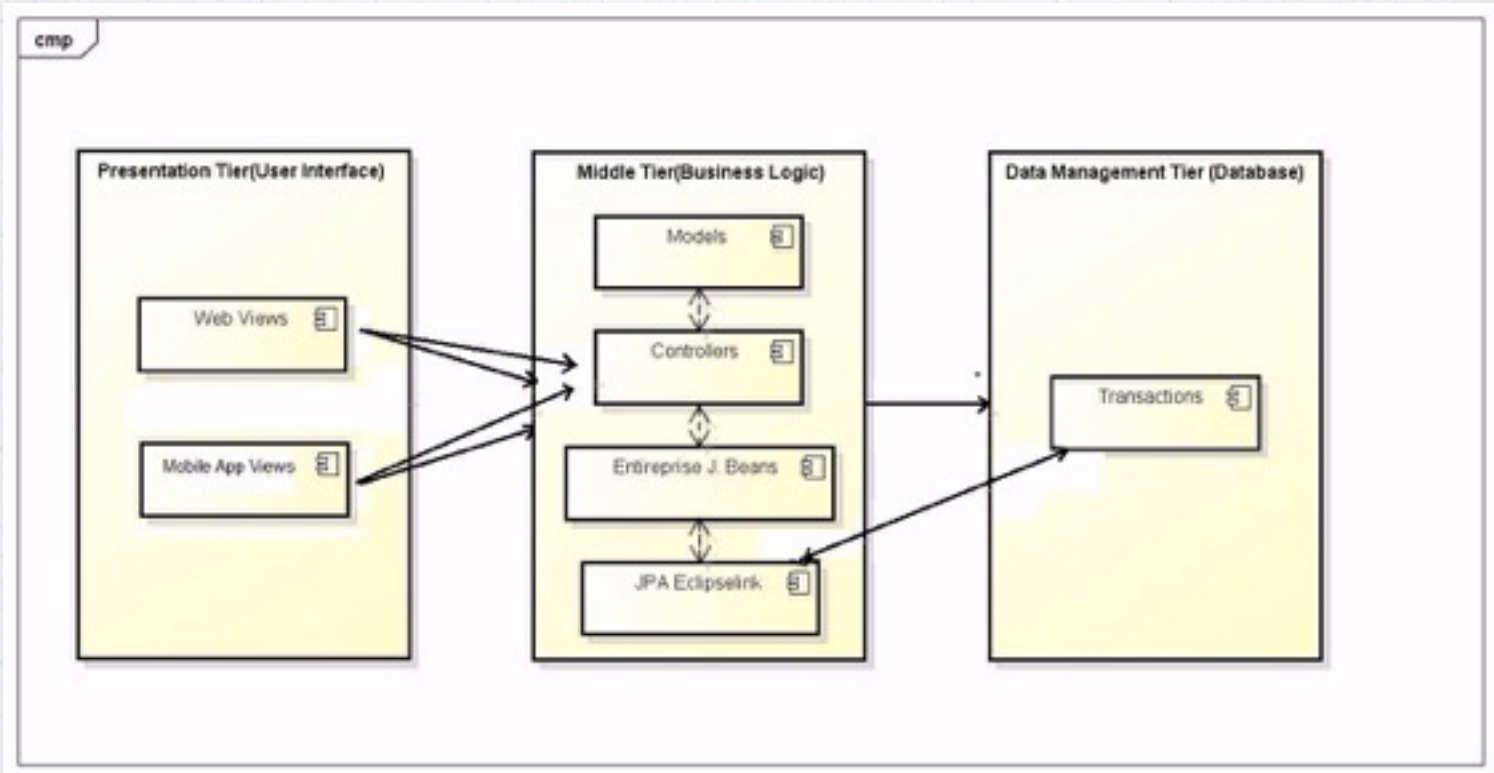


Figure 2. Component Diagram

5. Design Viewpoints

5. 6 State Dynamics Viewpoint

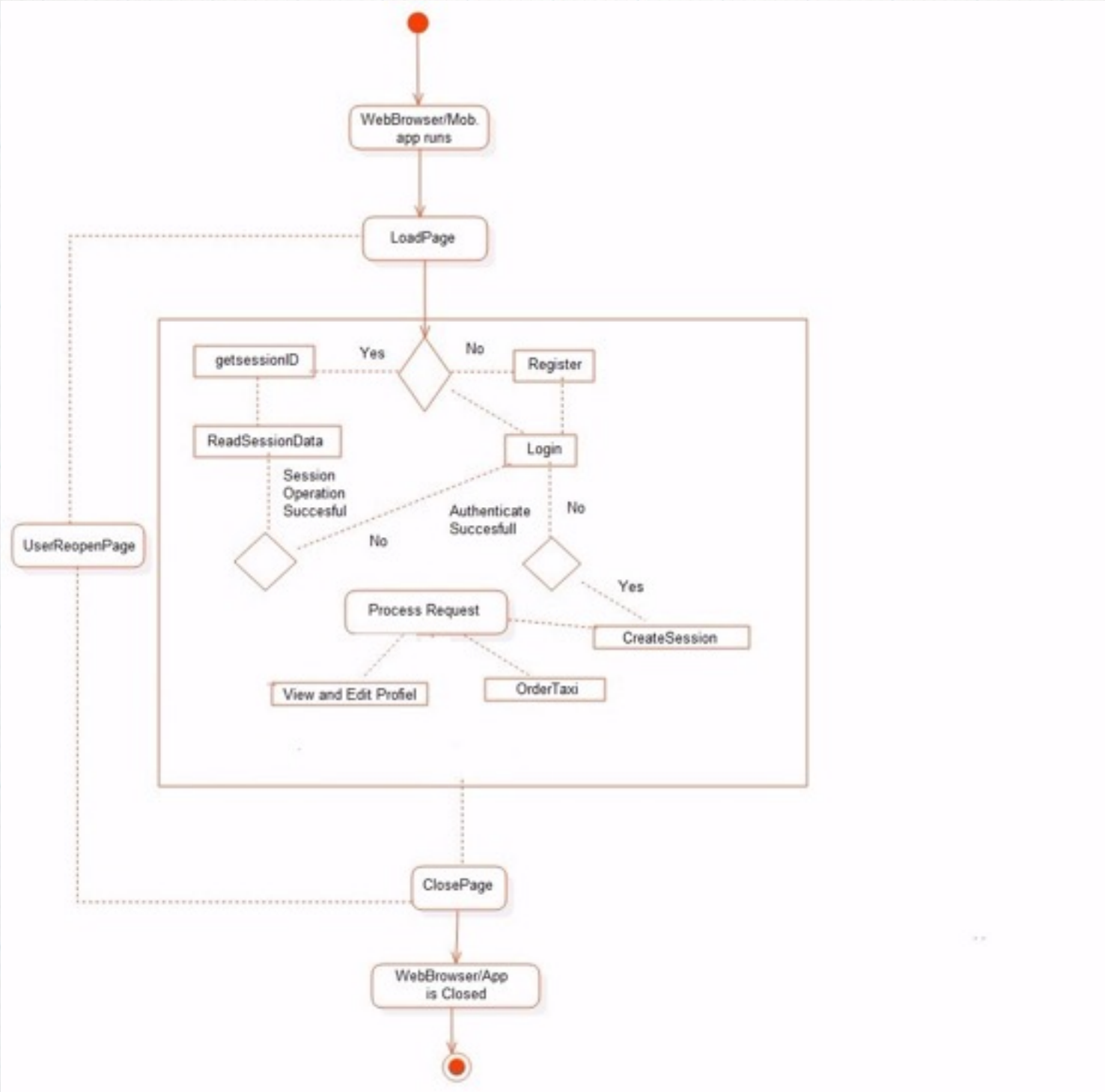


Figure 3. Active Diagram

5. Design Viewpoints

5. 7 Interaction Viewpoint

5. 7. 1 Register to the system

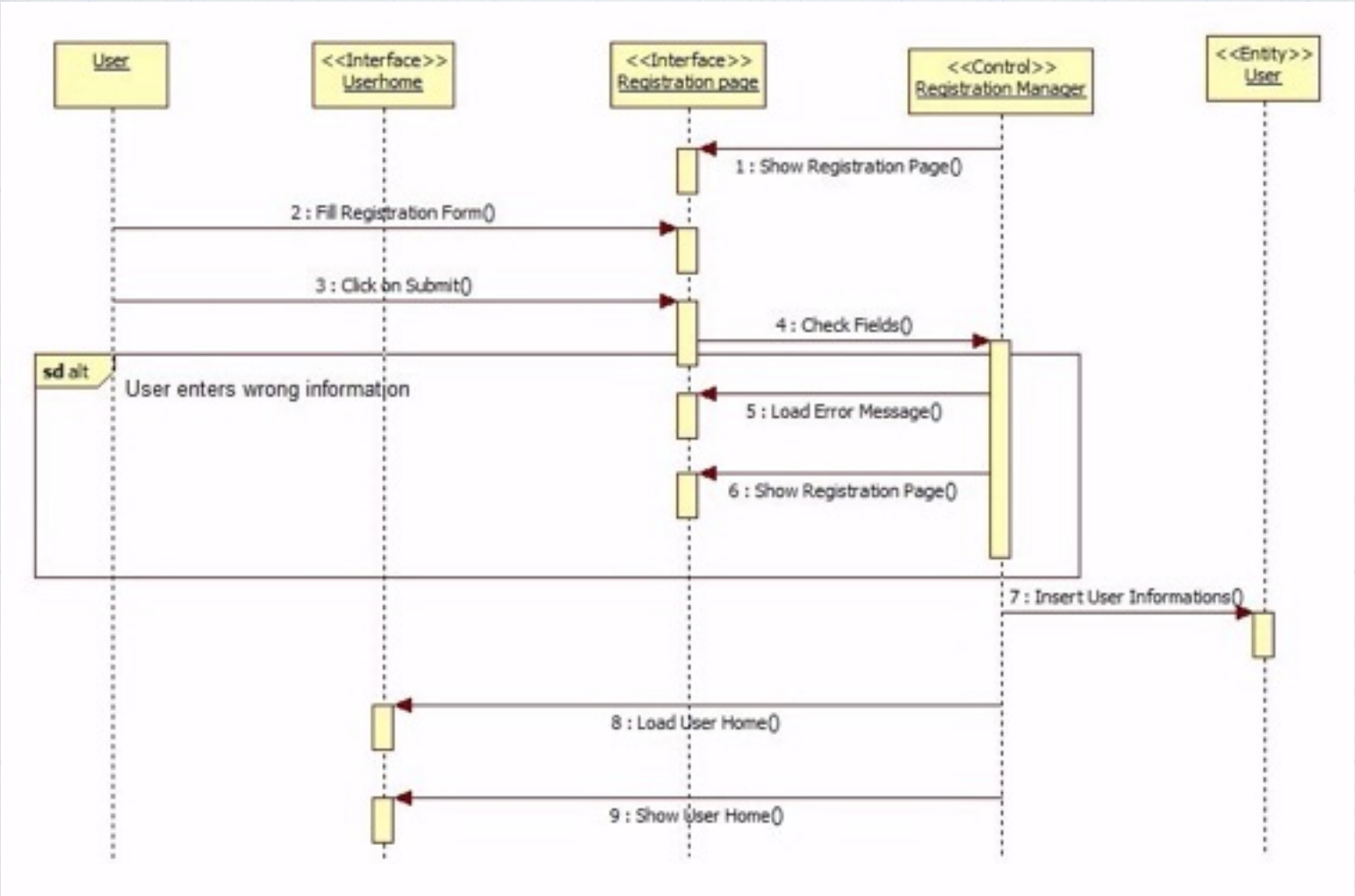


Figure 4 Register to the system

5. Design Viewpoints

5. 7 Interaction Viewpoint

5. 7. 2 Login

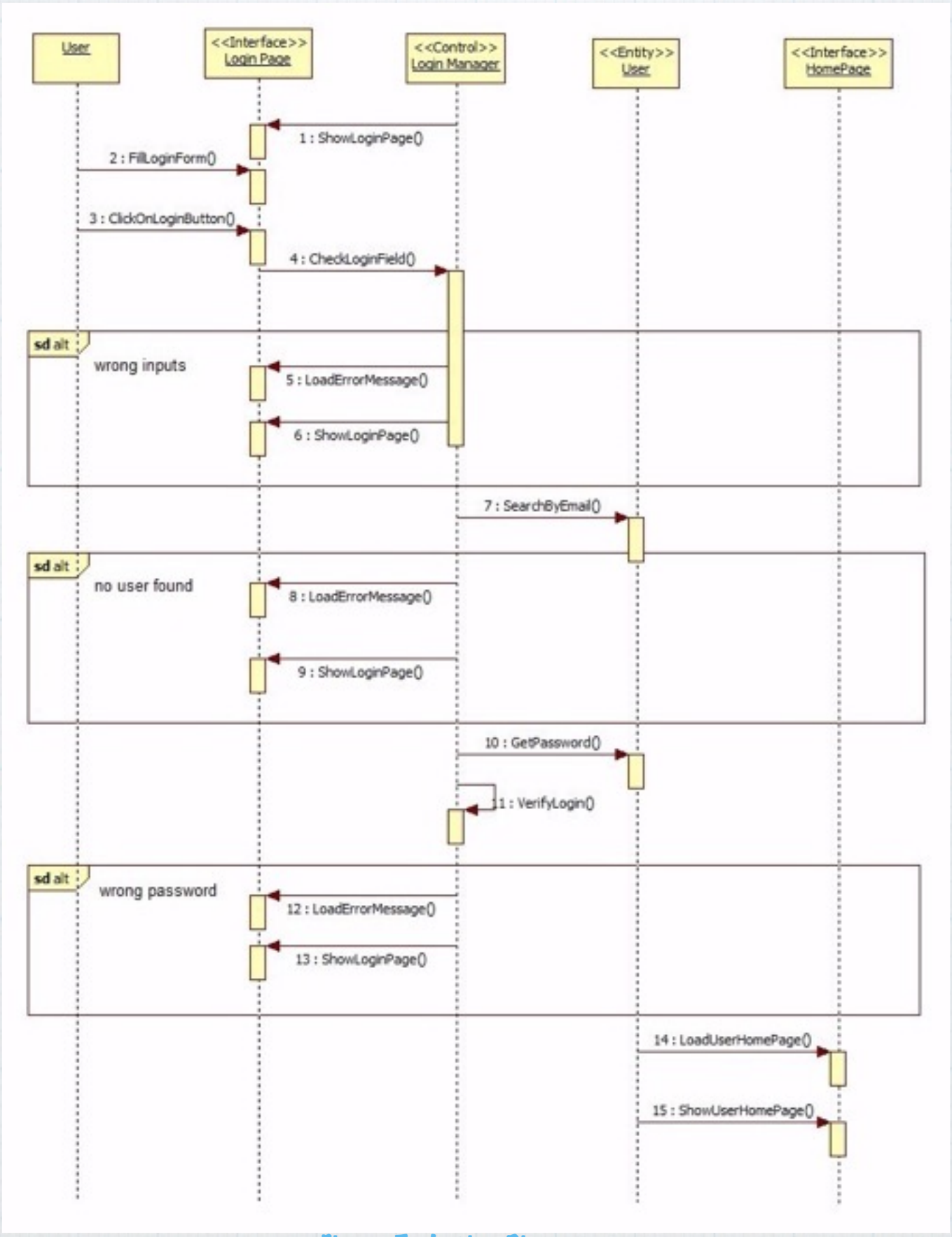


Figure 5. Active Diagram

5. Design Viewpoints

5.7.3 New Order

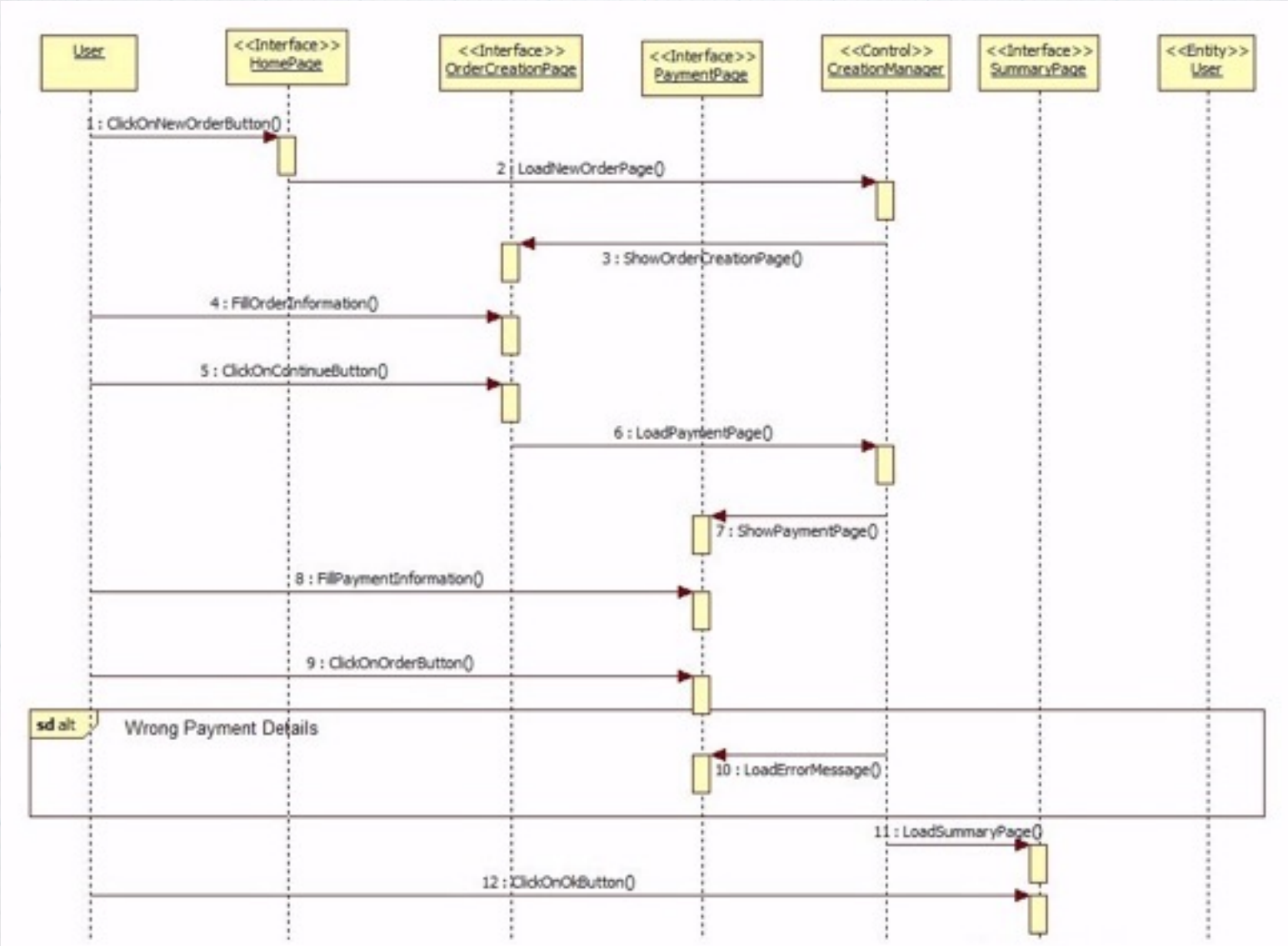


Figure 6. New Order

5. Design Viewpoints

5.7.4 Modify Profile

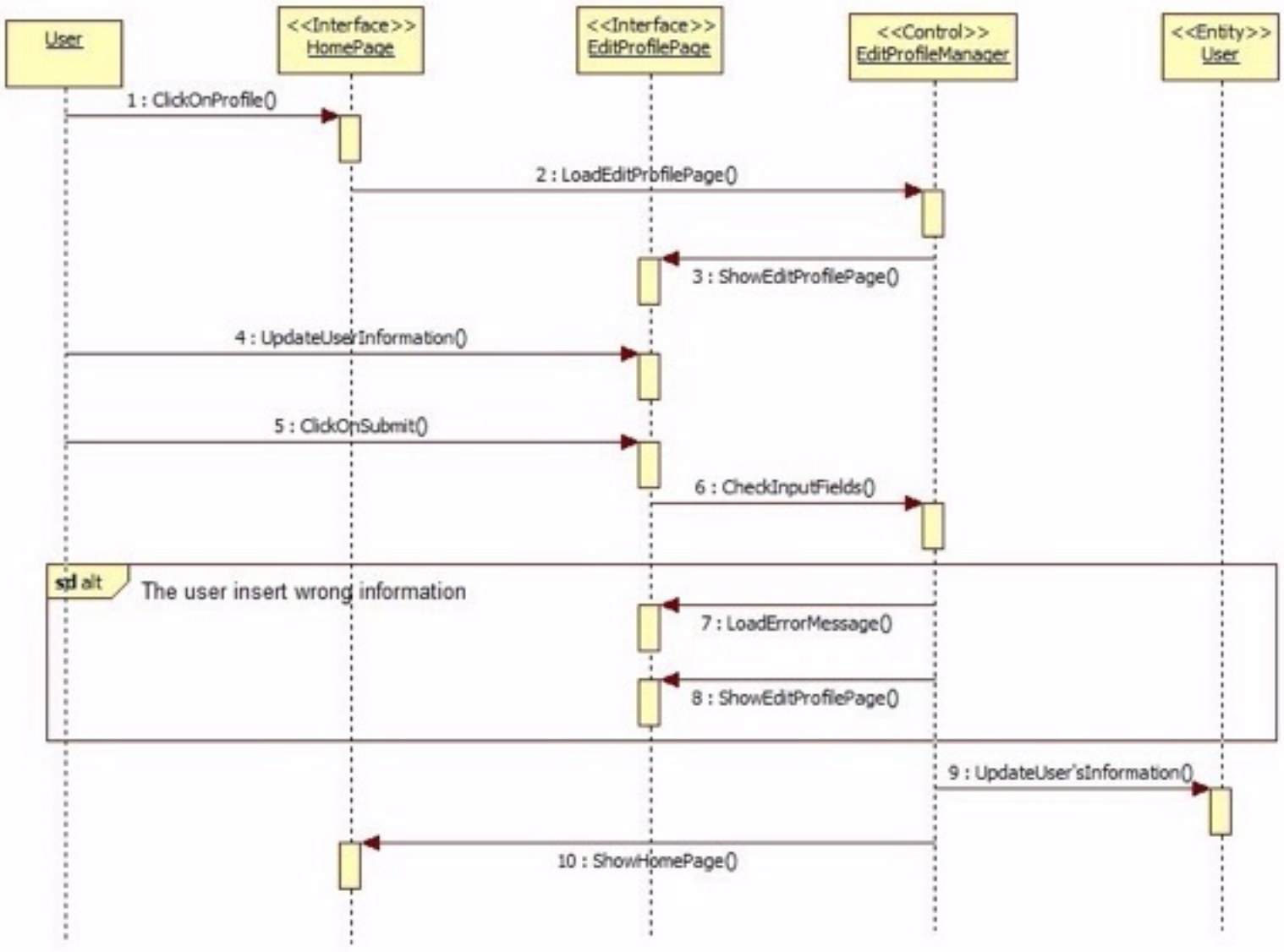


Figure 7. Modify Profile

5. Design Viewpoints

5.7.8 Notifications

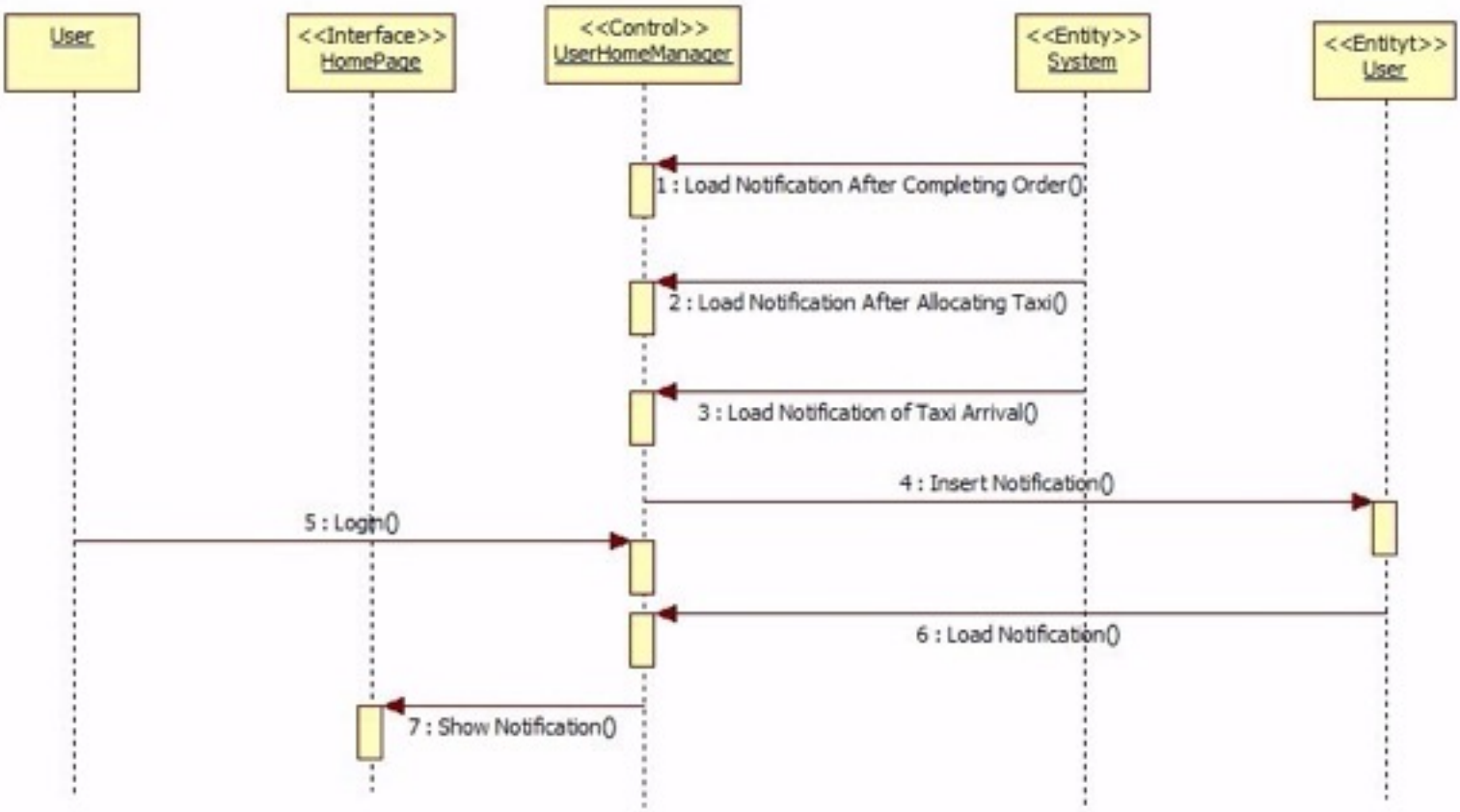


Figure 7. Notifications

6. Planning

Our team consists of two students. Since it is a small team it is possible to decide and develop everything together. However, ever member has his responsibilities stated below with the working hours until the end of requirement document.

Hasan Aliyev: Documentation 23h
Madat Mustafayev: Back-end and Front-end design development, 23h

7. Conclusion

This software Design Document is prepared for giving detailed design information of MyTaxiService Project. Furthermore, basics of data design, modules and design viewpoints of the system are described. The tools and libraries that will be used while designing and implementing the system are also provided.