

Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 10161998](#)¹ for the full IEEE Recommended Practice for Software Design Descriptions.

¹ <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieeeSDD.pdf>

(El Muchachos)
(Málacar)
Software Design Document

Name (s): Cergan Radu-Mihai; Plaisanu Alexandru-Ciprian; Micu Cristian-Mihai

Lab Section:

Workstation:

Date: (21/03/2021)

Software Design Document

Software Design Document

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 Purpose
- 1.2 Scope
- 1.3 Overview
- 1.4 Reference Material
- 1.5 Definitions and Acronyms

2. SYSTEM OVERVIEW

3. SYSTEM ARCHITECTURE

- 3.1 Architectural Design
- 3.2 Decomposition Description
- 3.3 Design Rationale

4. DATA DESIGN

- 4.1 Data Description
- 4.2 Data Dictionary

5. COMPONENT DESIGN

6. HUMAN INTERFACE DESIGN

- 6.1 Overview of User Interface
- 6.2 Screen Images
- 6.3 Screen Objects and Actions

7. REQUIREMENTS MATRIX

8. APPENDICES

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for Car Rental software application. It will illustrate the purpose and complete declaration for the development of the system.

The document is written for a more general audience, this document is intended for individuals directly involved in the development of Car Rental Application: MáláCar. This includes software developers, testers, project managers, teachers, users and team members.

1.2 Scope

MáláCar is a software system that automates the process of managing the car rental activity for a company, which helps people to rent cars at any moment from anywhere based on some preferences like price, car type, car details and more. The website can be accessed by any person, but for renting a car, the access is made through an account, creating one or log in into an existing account.

1.3 Overview

This document provides information about the software system of a Car Rental Application, describes some features about the designing implementation and its overall architecture.

The first part of this document is a short description of our application, some references, acronyms and definitions.

The second part of this document is a short overview of our application.

The third part of this document are some diagrams about the functionality of our program.(Architecture Design, Decomposition Description Design).

The fourth part is not applicable yet.

The fifth part of the document is not applicable.

The sixth part of the documents presents an overview of graphical interfaces and all of the functionalities of the website presented through mockups.

The seventh part of the document is not applicable..

The eighth part of the document is not applicable.

1.4 Reference Material

SPRING WEB MVC	<ul style="list-style-type: none">● https://www.javatpoint.com/spring-mvc-tutorial● https://spring.io/guides/gs/serving-web-content● https://www.tutorialspoint.com/spring/spring_web_mvc_framework.html
DATABASES	<ul style="list-style-type: none">● https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15● https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15
RENTAL CAR	<ul style="list-style-type: none">● https://en.wikipedia.org/wiki/Car_rental
FRAMEWORK CSS	<ul style="list-style-type: none">● https://getbootstrap.com/● https://devdocs.io/
OTHER REFERENCES	<ul style="list-style-type: none">● https://colorhunt.co/● https://docs.google.com/document/u/0/?tgif=d● https://app.diagrams.net/● https://www.w3schools.com/● https://www.codecademy.com/● https://javascript.info/

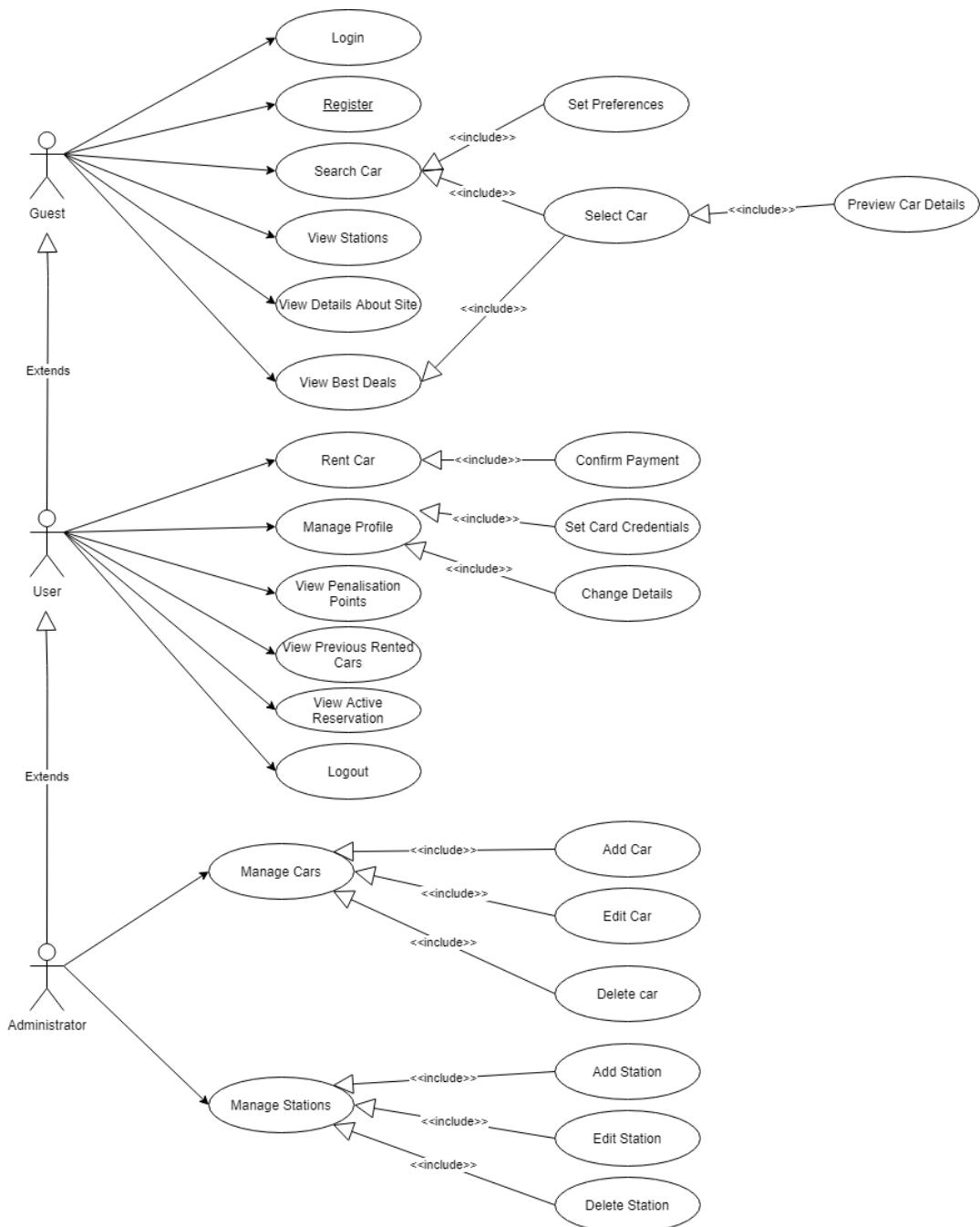
1.5 Definitions and Acronyms

User	Any person who has interaction with the system who is not a developer
Database	A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via database management system
SDD	Software Design Document
APP	Application
SQL	Structured Query Language

CSS	Cascading Style Sheets
Server	A central computer(server) which provides services such as file storage, printing and communications in a network computing system
Software requirement	A software capability needed by a user to solve a problem to achieve an objective; A software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed document.
System	A composite of equipment, skills and techniques capable of performing or supporting an operational role or both. A complete system includes all equipment, related facilities, material, software, services and personnel required for its operation and support to the degree that it can be considered a self-sufficient item in its intended operational environment.

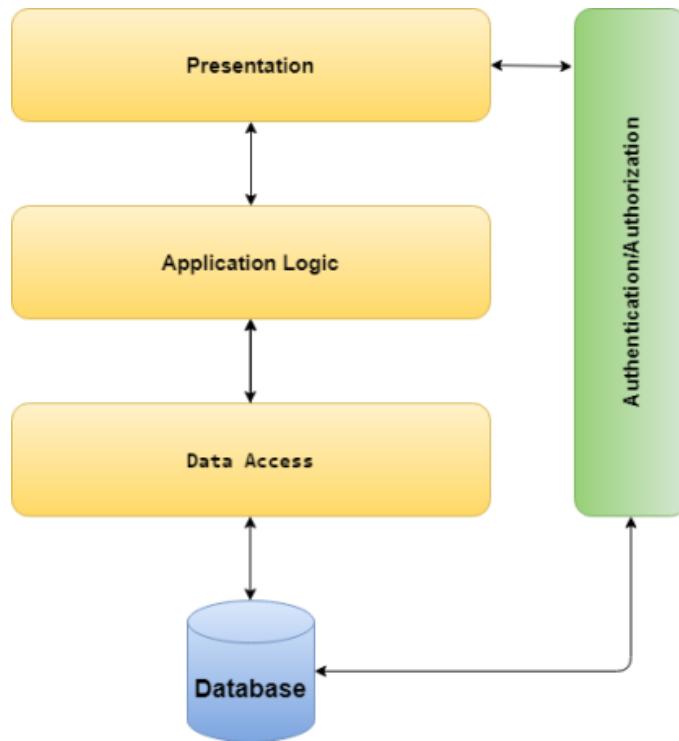
2. SYSTEM OVERVIEW

The Rental System is a new, self-contained system intended for web application. This system is intended for clients that want to rent cars, just with the help of the internet, from anywhere to everywhere. The users can rent a car, pay for renting, view penalization points, previous rented cars and transactions.



3. SYSTEM ARCHITECTURE

3.1 Architectural Design



The most common architecture pattern is the layered architecture pattern, otherwise known as the n-tier architecture pattern. This pattern is the de facto standard for most Java EE applications and therefore is widely known by most architects, designers, and developers.

Each layer of the layered architecture pattern has a specific role and responsibility within the application. For example, a **presentation layer** would be responsible for handling all user interface and browser communication logic, whereas a business layer would be responsible for executing specific business rules associated with the request. Also the presentation layer doesn't need to know or worry about how to get customer data; it only needs to display that information on a screen in particular format.

The **application layer** doesn't need to be concerned about how to format customer data for display on a screen or even where the customer data is coming from; it only needs to get the data from the data access layer, perform application logic against the data (e.g., calculate values or aggregate data), and pass that information up to the presentation layer.

Data access layer in computer software is a layer of a computer program which provides simplified access to data stored in persistent storage of some kind, such as an entity-relational

database.

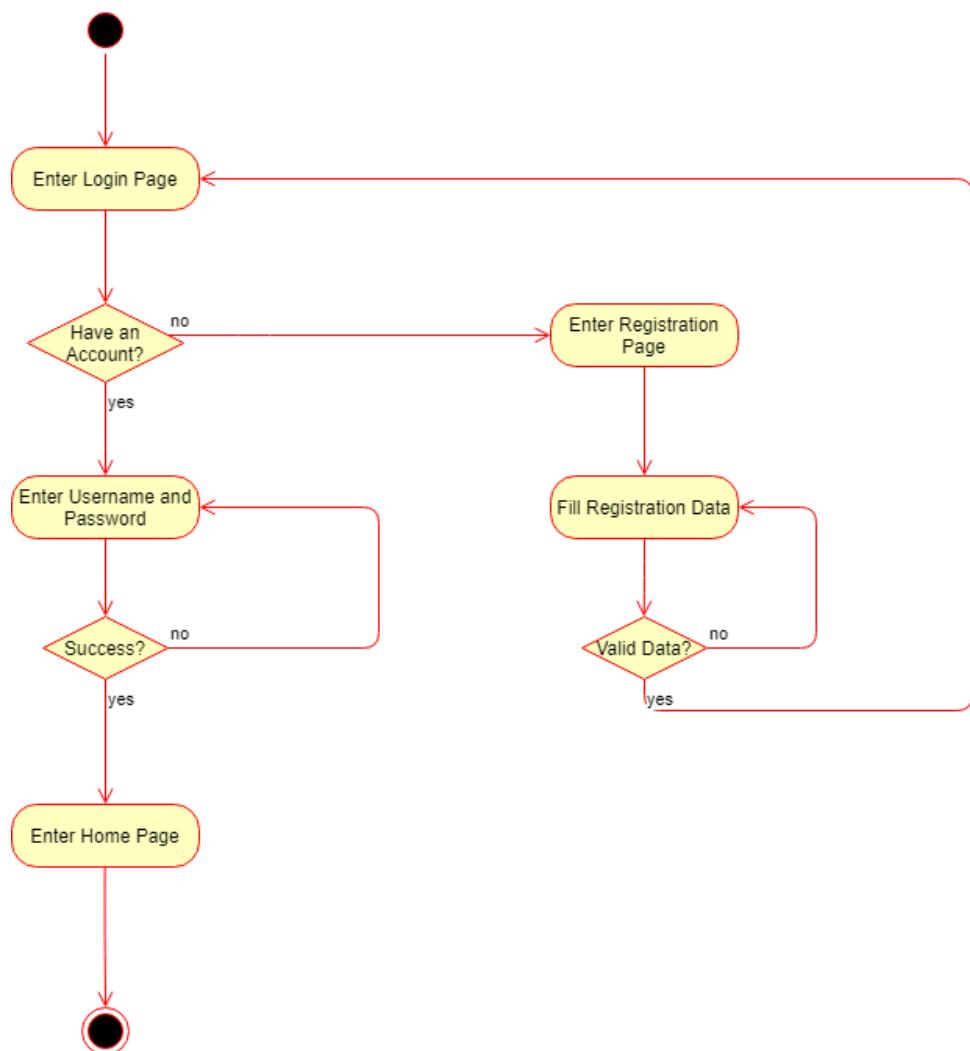
The **database layer** takes care of data access. An object from the database layer can write itself to one or more tables. In the database layer, you'll find things like database, connection, table, SQL, and result set.

Authentication is the process of verifying the identity of an individual. A unique identifier is associated with a user which is the username or userid. Traditionally, we use a combination of username and password to authenticate a user. A user can interact with a web application using multiple actions. Access to certain actions or pages can be restricted using user levels.

Authorization is the process of controlling user access via assigned roles & privileges.

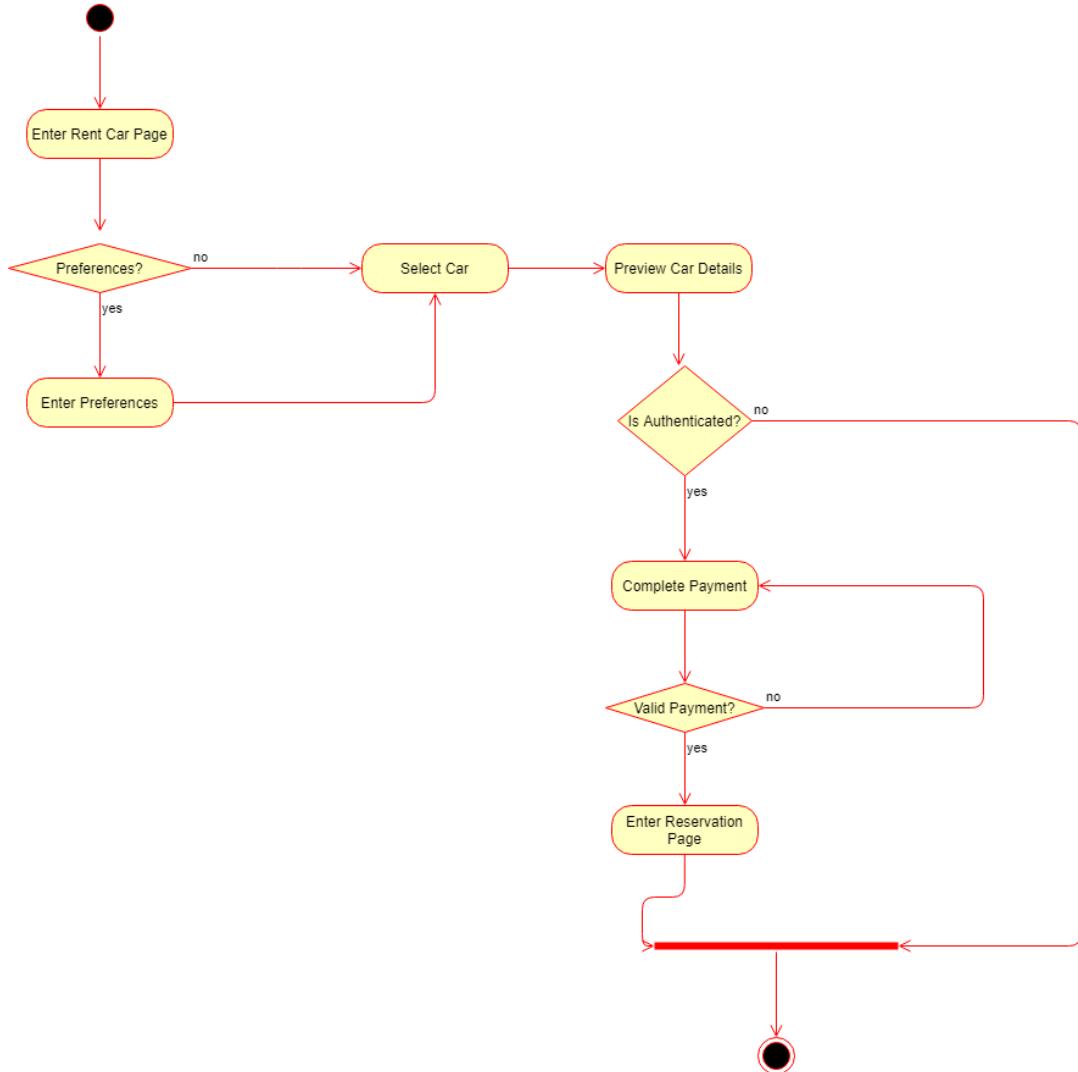
The layered architecture pattern is a solid general-purpose pattern, making it a good starting point for most applications, particularly when you are not sure what architecture pattern is best suited for your application.

3.2 Decomposition Description



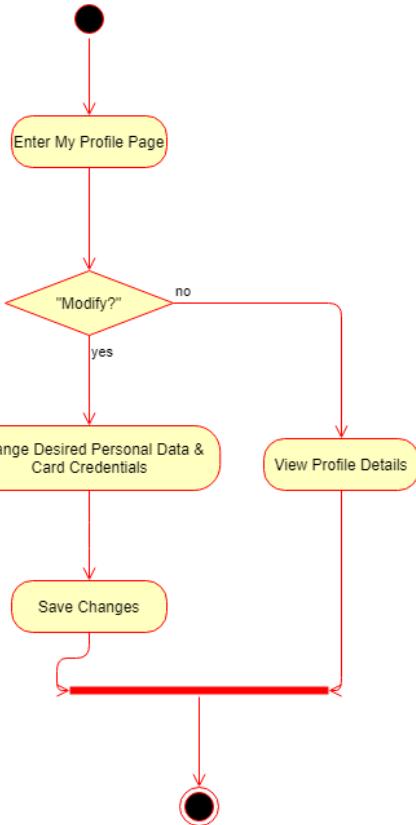
USER

RENT CAR ACTIVITY DIAGRAM



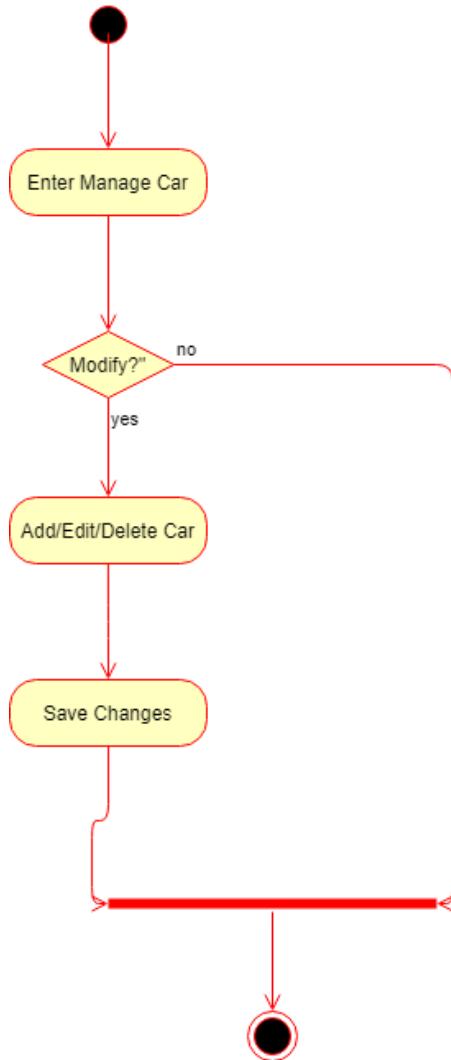
USER AND ADMINISTRATOR

MANAGE PROFILE



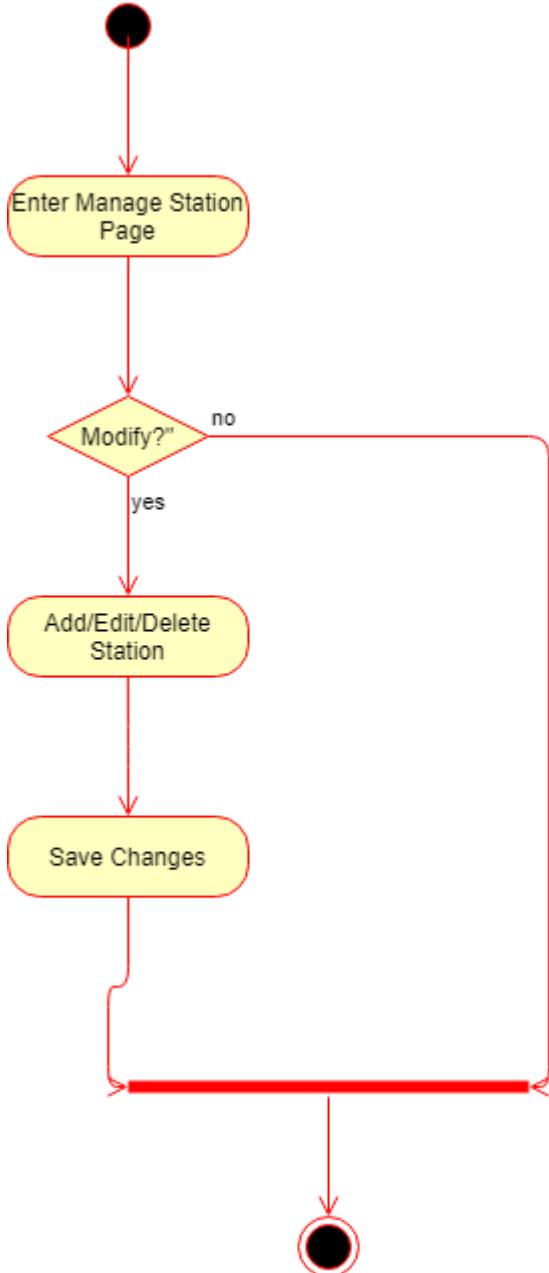
ADMINISTRATOR

MANAGE CARS



ADMINISTRATOR

MANAGE STATIONS

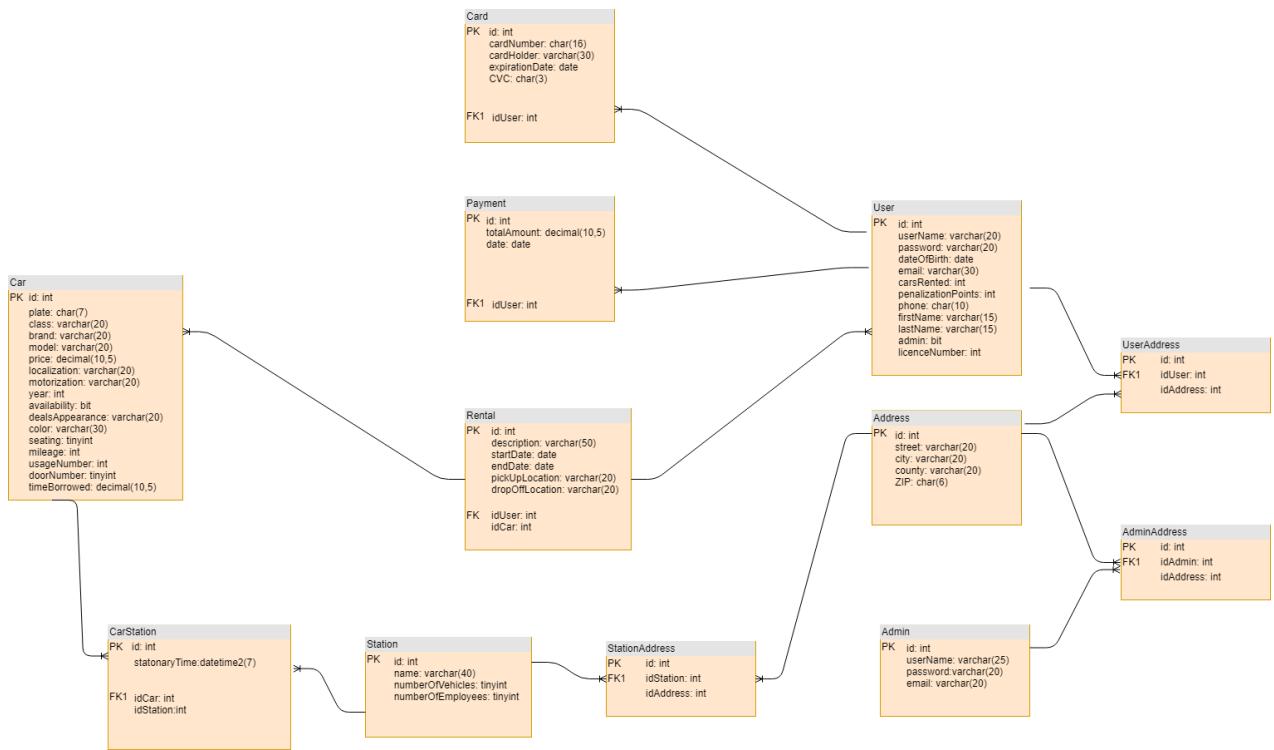


3.3 Design Rationale

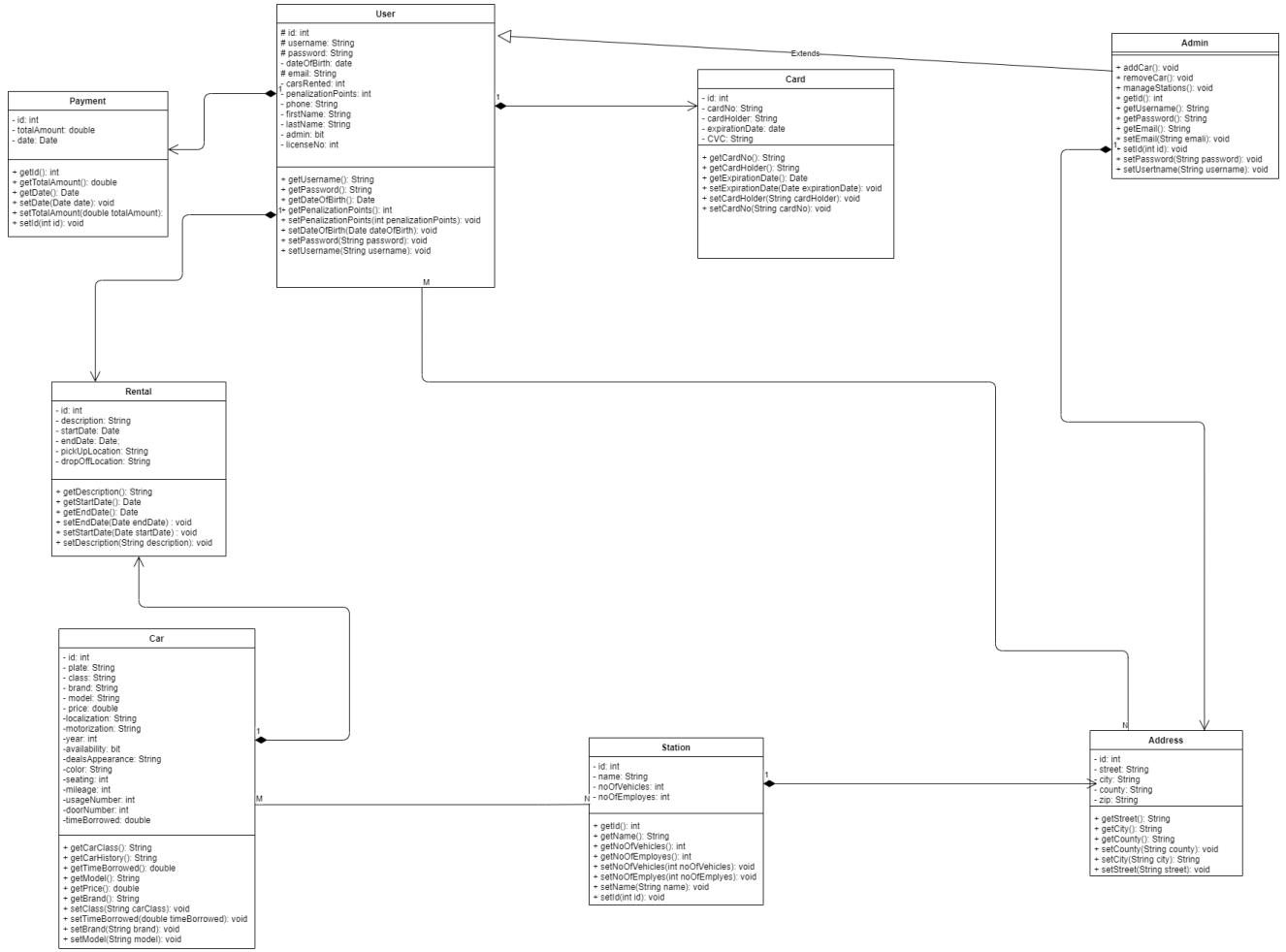
<NOT APPLICABLE>

4. DATA DESIGN

4.1 Data Description



4.2 Data Dictionary



6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The APP has two account types: one as a user(after you register you become a user and you can rent a car or do some other features) and the other one as an admin(which can manage the cars, manage the rental of the cars and more). Also you can enter on the site without being a user, as a guest, but you have some restrictions(limited access).

The functionalities are presented below:

- guests can register
- all types of users can see the available cars for renting
- guests and users can see the details about a car(class, motorization, price, location, etc.)

- all types of users can search a car
- all types of users can set the preferences for listing the cars
- all types of users can access the best deals available
- all types of users can view the renting stations from each city
- all types of users can access the website information page(About Us section)
- users can rent a car
- users can see the history of his rented cars
- users can see the reservation details of appointment(date to pick/bring back the car, hour , location, etc.)
- users can be penalized if something happens to the car by his fault(this can lead to a three months suspension ban, by accumulating a number of penalization points)
- users and administrators can log in
- users and administrator can see his profile information
- users and administrator can change the profile information
- users and administrator can log out
- the administrator can see the rented cars
- the administrator can change the details about a car(motorization, price, location, etc.)
- the administrator can modify the preferences for listing the cars
- the administrator can modify the stations(add/delete a station)
- the administrator modify the website information page(About Us section)
- the administrator can add/delete a car

When opening the website, the guest can see the start-up page. This page should act like a welcome page. The guest can see the login button and the menu button, which is a pop-up list of other buttons such as “Rent a Car”, “About us”, etc. If logged in as a user, instead of the login button should be the “My profile” button. In the first page is also presented a set of “Best Deals” that should act like a slider in JS, there are located cars sorted by different criterias such as a best selling car.

Note ! The website’s logo located at the top-left corner of the tab should act like a “Home” button on every page related to our site.

After pressing the “Login” button from the homepage, another page where we have to log in appears. If the guest doesn’t have an account, he should press “Don’t have an account? Click here to register”.

When the guest presses the “Don’t have an account? “Click here to register” button, he will be redirected to the register page, where he must complete all the fields in order to advance. Note that the field “Password” must match with “Confirm Password”. After the registration completes, the guest can log in as a user.

After a successful login, the username should be displayed at the top-right. This should also act as a button too, showing a pop-up list of “My profile” - this leads to the information of the account, where we can change the password, mail, etc. , “History” - where the user can check

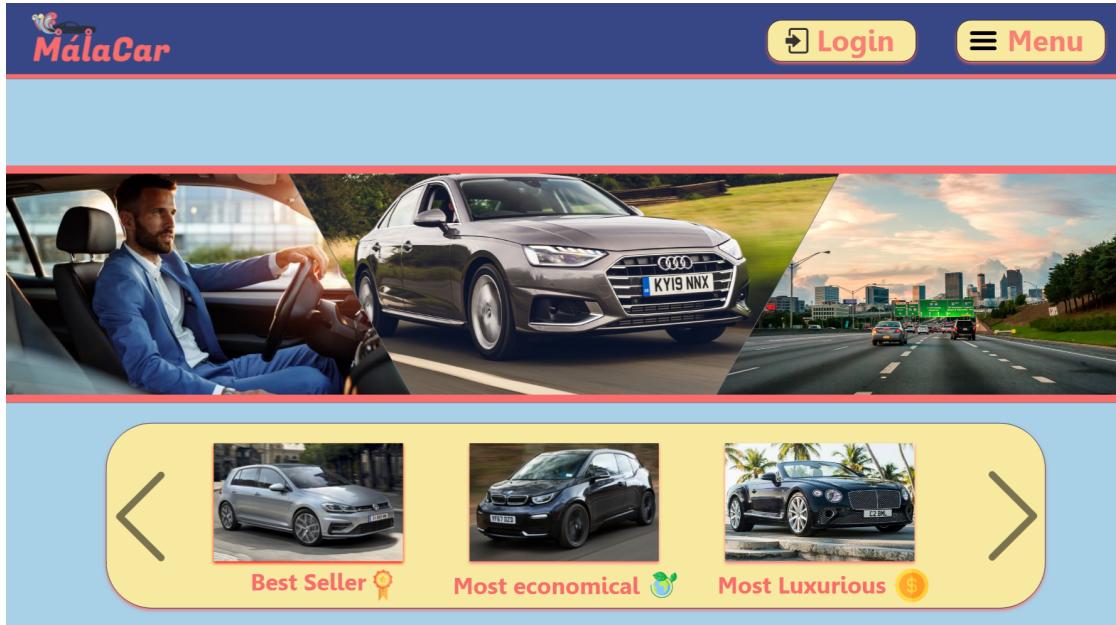
his renting history and the penalization points, “Reservation” - where the user can see the information about the car he wants to rent(location, hour to pick up), and the “Logout” button.

The page where we browse the listed cars is the “brain” of the website. The user should be able to search for a car regarding some preferences such as type, brand, model, etc. . There will also be listed all the cars, before entering any selecting criteria.

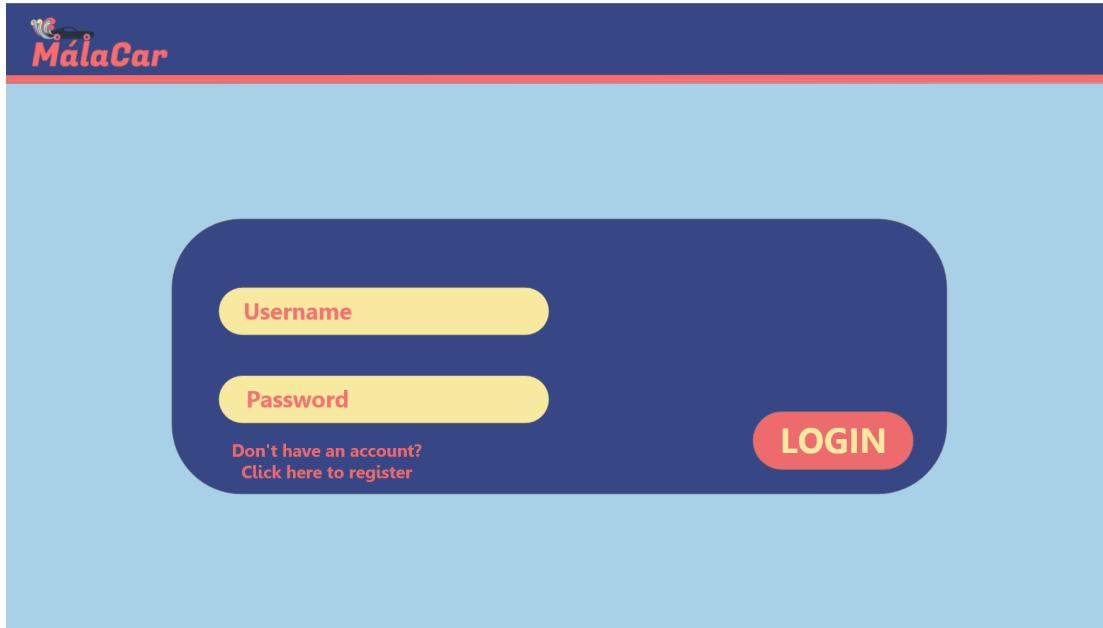
By pressing the “Preview” button from the previous screen, the user should be redirected to another page, where he should be able to see all the features about the car he selected. After completing the two mandatory fields of “start date” and “end date”, the user can press “Book Now” for advancing into the payment section.

6.2 Screen Images

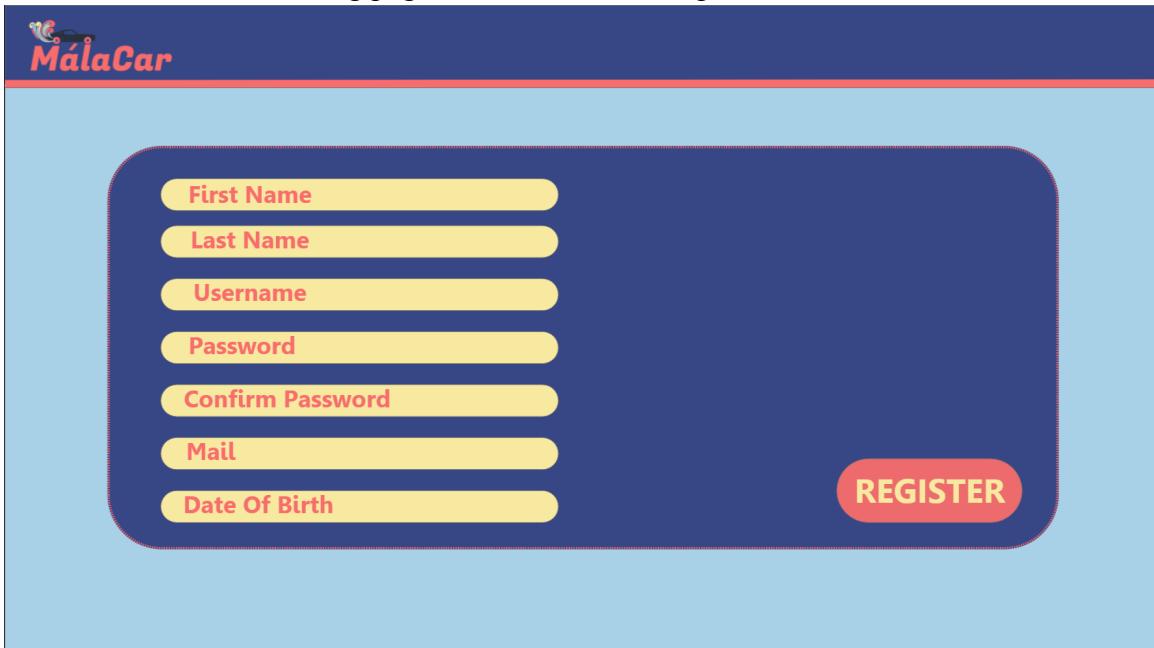
When opening the website, the guest can see the start-up page. This page should act like a welcome page.



After pressing the “Login” button from the homepage, the following page should appear. If the guest doesn't have an account, he should press “Don't have an account? Click here to register



When the guest presses the “Don’t have an account? “Click here to register” button, he will be redirected to the following page, where he must complete all the fields in order to advance.



After a successful login, the user should be redirected to the welcome page.

MalaCar

(8) **marian.87**

Menu

Best Seller

Most economical

Most Luxurious

The following page is the “brain” of the website. The user should be able to search for a car regarding some preferences such as type, brand, model, etc.

Search

Checkbox Checkbox Checkbox Checkbox

Motorization **Type** **Doors** **Capacity**

Mercedes	A-Class		Compact Hatchback	Technical Specifications	Price/day	Preview
Mercedes	A-Class		Compact Hatchback	Technical Specifications	Price/day	Preview
Mercedes	A-Class		Compact Hatchback	Technical Specifications	Price/day	Preview

By pressing the “Preview” button from the previous screen, the user should be redirected to another page, where he should be able to see all the features about the car he selected.



Volkswagen Golf VII



Technical Specifications

Start date

End date

BOOK NOW

7. REQUIREMENTS MATRIX

<NOT APPLICABLE>

8. APPENDICES

<NOT APPLICABLE>