|  |  |
| --- | --- |
| Spring NetCracker Project |  |
| Taxi Service System |
|  |

|  |  |
| --- | --- |
| Version Number | [1.00] |
| Status | [Draft] |
| Document Date | [19.05.2015] |
| Owner | TrackSee (Group B) |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Changes |
| 1.0 | 19.05 | Vitalii Diravka | My general version of documentation |
| 2.0 | 20.05 | Sharaban Sasha | General user design subsection description |
| 3.0 | 25.05 | Vitalii Diravka | Added Class Cases, ER-diagram |
| 4.0 | 25.05 | Vitalii Diravka | Added Use Cases |

Document Reference

|  |  |  |
| --- | --- | --- |
| # | Document Name | Description |
| 1 | UseCases.docx, | UseCase diagrams and descriptions |
| 4 | TrackSee-Project.png | Structure of project |
| 5 | ERD v 6.1.png | DataBase ER-diagram |
| 6 | traceability-matrix.xlsx | Traceability matrix |
| 7 | reports | Weekly reports of each team member |
| 8 | Logic\_detailed.pdf,  Presentation\_detailed.pdf,  Storage\_detailed.pdf | Class diagram |

Glossary of Terms

|  |  |  |
| --- | --- | --- |
| # | Acronym | Interpretation |
| 1 | User | End-user of the system (Customer, Driver) |
| 2 | TO | Taxi Order |
| 3 | RI | Resource Inventory |
| 4 | M | Mandatory requirement |
| 5 | O | Optional requirements |
| 6 | Admin | Administrator |
| 7 | DDD | Domain-driven design |

Оглавление

[1 Introduction 5](#_Toc420395679)

[1.1 Document Purpose 5](#_Toc420395680)

[1.2 Document Objectives 5](#_Toc420395681)

[1.3 Intended Audience 5](#_Toc420395682)

[2 Project timeline 6](#_Toc420395683)

[3 Database ER-diagram 7](#_Toc420395684)

[4 Class diagram 8](#_Toc420395685)

[4.1 Storage layout 8](#_Toc420395686)

[4.2 Logic layout 9](#_Toc420395687)

[4.3 Presentation layout 10](#_Toc420395688)

[5 Use Cases 11](#_Toc420395689)

[6 TrackSee Web Application Overview 22](#_Toc420395690)

[6.1 General User Design Subsection 22](#_Toc420395691)

[6.2 Personal Administrator Design Subsection 22](#_Toc420395692)

[6.3 Personal Driver Design Subsection 22](#_Toc420395693)

[7 TrackSee Web Application Detailed Description 23](#_Toc420395694)

[7.1 General User Design Subsection 23](#_Toc420395695)

[7.1.1 Home Page 23](#_Toc420395696)

[7.1.2 Sign In Page 23](#_Toc420395697)

[7.1.3 Order Page 23](#_Toc420395698)

[7.1.4 Dashboard Page 23](#_Toc420395699)

[7.1.5 Order Information Page 23](#_Toc420395700)

[7.2 Personal Administrator Design Subsection 24](#_Toc420395701)

[7.2.1 Driver Accounts Page 24](#_Toc420395702)

[7.2.2 Cars Page 24](#_Toc420395703)

[7.2.3 Reports Page 24](#_Toc420395704)

[7.2.4 Tariffs Page 25](#_Toc420395705)

[7.2.5 Groups And Roles 25](#_Toc420395706)

[7.3 Personal Driver Design Subsection 25](#_Toc420395707)

[7.3.1 Free Orders Page 25](#_Toc420395708)

[7.3.2 Assigned Orders Page 25](#_Toc420395709)

[7.3.3 History Of Orders Page 26](#_Toc420395710)

[7.3.4 Own Profile Page 26](#_Toc420395711)

# Introduction

This document represents the Design Specification for TrackSee web application for the “Taxi Service System” project.

## Document Purpose

The Taxi Service System Project Documentation provides a definition of the project, including the project’s goals and objectives.

## Document Objectives

The objectives of this document are to define the following:

Project timeline

Database ER-diagram

Class diagram

The conceptual design of web application

Detailed description of web application

## Intended Audience

This document is intended for use by:

Business Analysts

Project Manager

Enterprise Architects

Solution Engineers

# Project timeline

|  |  |
| --- | --- |
| 09.04.15 - 16.04.15 | Use Case diagrams for administrator, client and driver |
| Database structure (ER diagram) |
| Prototypes of such web pages: registration, authorization, taxi ordering, client page, admin page, driver page |
| Choose maven archetype |
| Develop SQL database scripts |
| 16.04.15-23.04.15 | Realization of DAO pattern |
| Registration and user management |
| User roles |
| E-mail notification |
| 23.04.15 - 30.04.15 | RI management |
| Taxi order |
| Availability check |
| Tracking and TO processing |
| 30.04.15 - 07.05.15 | Dashboards and TO lifecycle |
| Integration with 3rd party system |
| Reports |
| 07.05.15 - 14.05.15 | Data preparation |
| Testing |
| Bag fixing |
| Refactoring |
| 14.05.15 - 20.05.15 | Deploy |

# Database ER-diagram

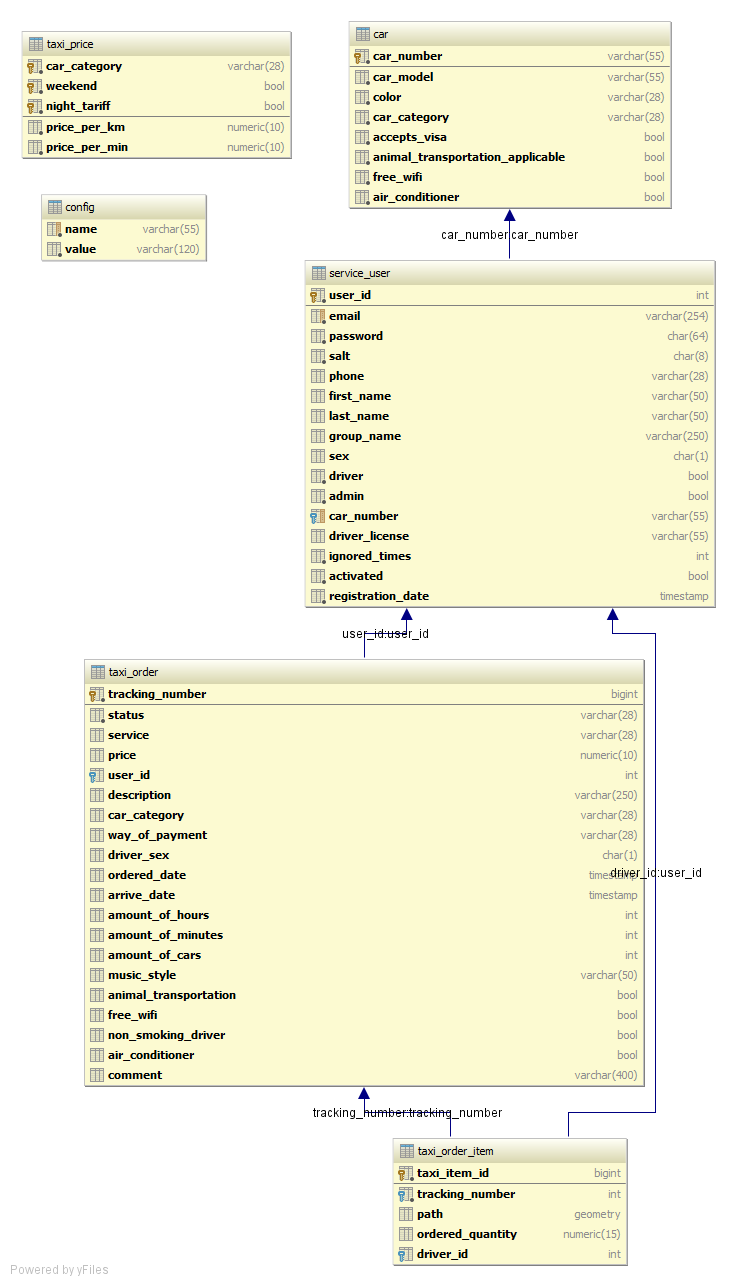


Figure 1 – Database ER-diagram

# Class diagram

## Storage layout

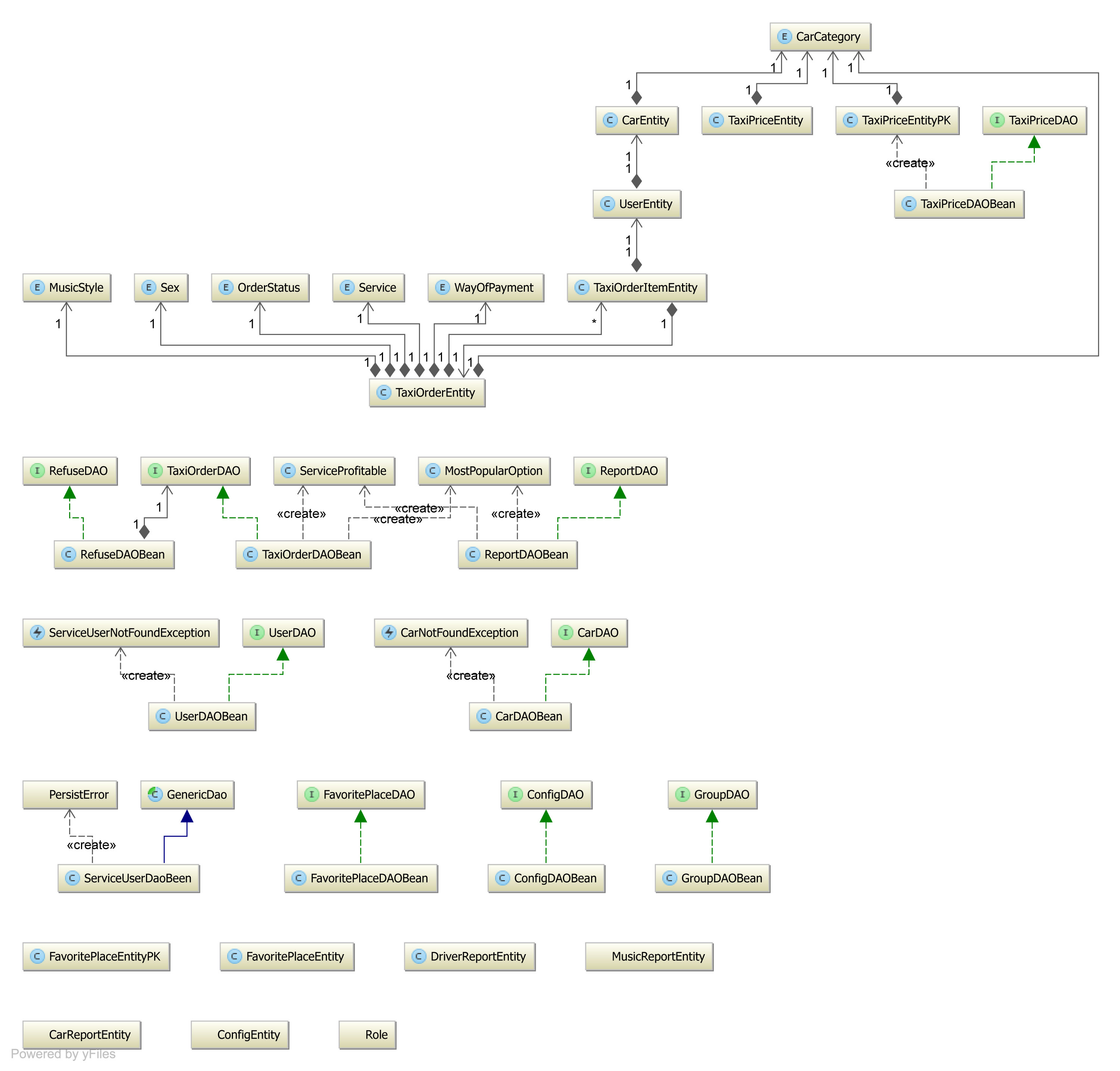


Figure 2 – Storage layout class diagram

## Logic layout

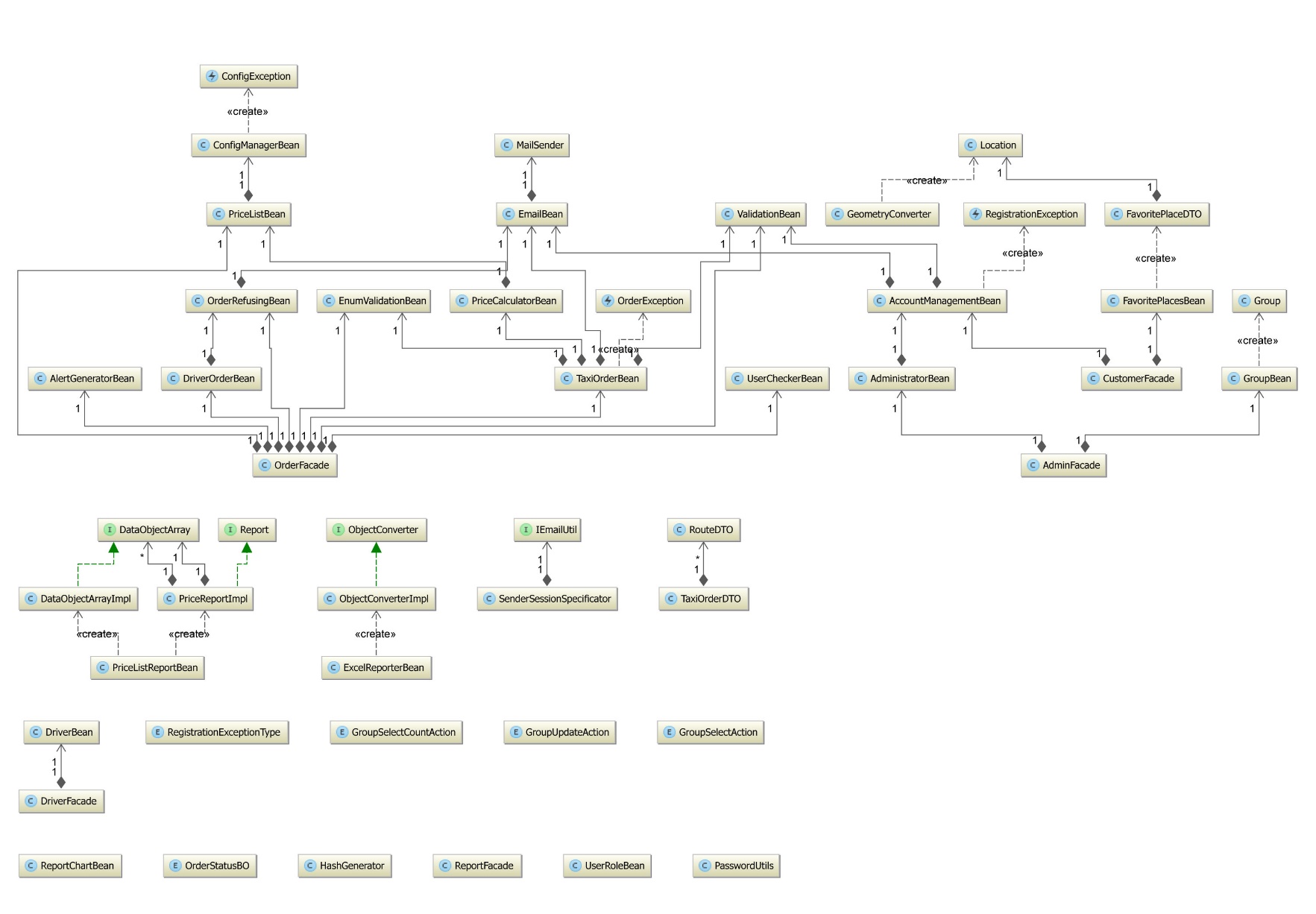


Figure 3 – Logic layout class diagram

## Presentation layout

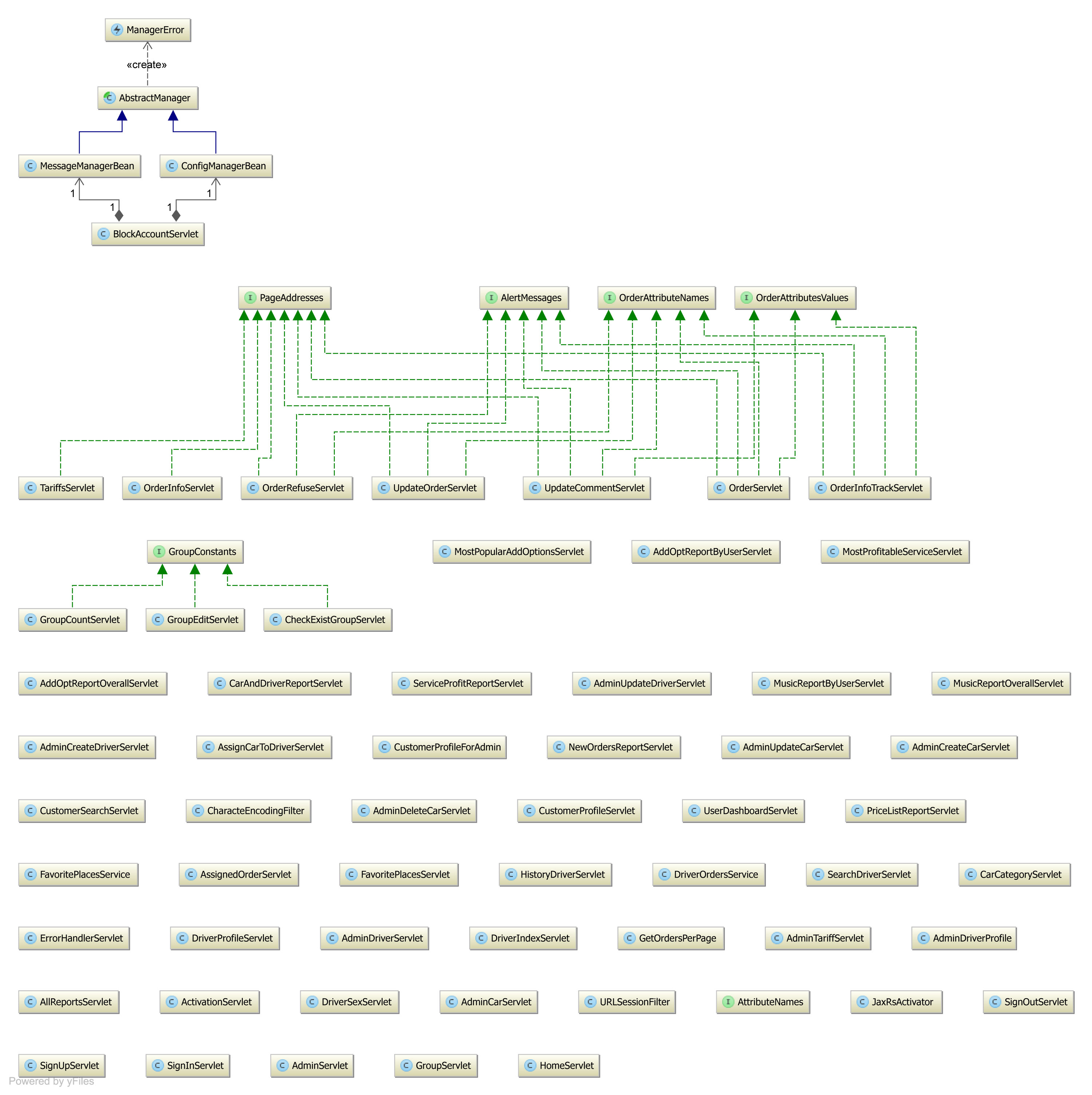


Figure 4 – Storage layout class diagram

# Use Cases

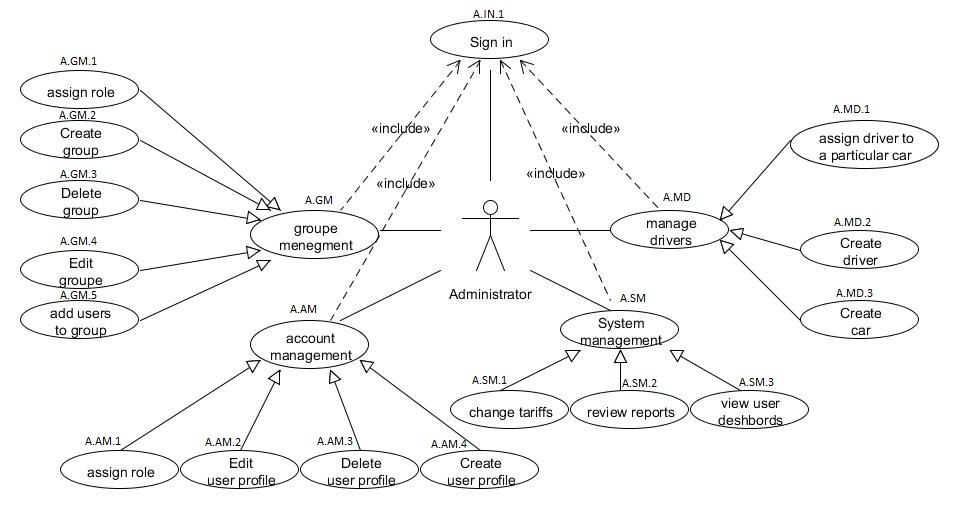


Figure 5 – Administrator Use Case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Case ID** | **Use Case Name** | **Actor** | **Main Success Scenario** |  | **Alternative Flow** |  | **Precondition** |
|  |  |  | **Step** | **Step Description** | **Step** | **Step Description** |  |
| A.IN.1 | Sign In | 1) Admin A | 1 | Admin open login page | 2.a | A didn’t fill login fields |  |
|  |  | 2) System S | 2 | Admin enter login and password | 2.a.1 | S displays message about invalid input |  |
|  |  |  | 3 | A press button Log in | 2.b | A didn’t enter valid data |  |
|  |  |  | 4 | S checks if administrator exist | 2.b.1 | S displays message about invalid input |  |
|  |  |  | 5 | S redirects Admin to Admin Dash Board | 4.a | Admin with specified login data not exists in the system |  |
|  |  |  |  |  | 4.a.1 | S displays message that no user exists |  |
| A.GM.1 | assign role | 1) Admin A | 1 | Admin click edit groupe from groupe list | 3.a | groupe already has role | A.GM.2 |
|  |  | 2) System S | 2 | Admin select role from  role list | 4.a | no existing role |  |
|  |  |  | 3 | System link groupe to role | 4.b | no existing groupe |  |
| A.GM.2 | Create groupe | 1) Admin A | 1 | Admin input name of groupe | 1.a | name alrady exist |  |
|  |  | 2) System S | 2 | add some user to  group | 1.b | A don’t add any user |  |
|  |  |  | 3 | Click create | 1.b.1 | S displays message that  admin must add user to create groupe |  |
|  |  |  | 4 | System create groupe |  |  |  |
| A.GM.3 | Delete groupe | 1) Admin A | 1 | Admin select groupe from groupe list | 3.a | Groupe alrady deleted | A.GM.2 |
|  |  | 2) System S | 2 | Click delete |  |  |  |
|  |  |  | 3 | System delete groupe |  |  |  |
| A.GM.4 | Edit groupe | 1) Admin A | 1 | Admin select groupe from groupe list |  |  | A.GM.2 |
|  |  | 2) System S | 2 | System showing fields of selected groupe |  |  |  |
|  |  |  | 3 | Admin changes the  group field that he needs |  |  |  |
|  |  |  | 4 | Click update |  |  |  |
|  |  |  | 5 | System showing groupe  with edited fields |  |  |  |
| A.GM.5 | add users  to groupe | 1) Admin A | 1 | S show user list | 2.a | user was already added |  |
|  |  | 2) System S | 2 | A select users that will  be added |  |  |  |
|  |  |  | 3 | S redirect on  groupe page |  |  |  |
| A.AM.1 | assign role | 1) Admin A | 1 | A select edit user  from user list |  |  |  |
|  |  | 2) System S | 2 | A select role from list |  |  |  |
|  |  |  | 3 | S assign role to user |  |  |  |
|  |  |  |  |  |  |  |  |
| A.AM.2 | edit user profile | 1) Admin A | 1 | Admin select user from user list | 3.a | A input invalid values | A.AM.4 |
|  |  | 2) System S | 2 | System showing fields of selected user | 3.a.1 | S displays message  about invalid input |  |
|  |  |  | 3 | Admin changes the  user field that he needs |  |  |  |
|  |  |  | 4 | Click update |  |  |  |
|  |  |  | 5 | System showing user  with edited fields |  |  |  |
| A.AM.3 | Delete user  profile | 1) Admin A | 1 | Admin select user from user list |  |  | A.AM.4 |
|  |  | 2) System S | 2 | Click delete |  |  |  |
|  |  |  | 3 | System delete user |  |  |  |
|  |  |  |  |  |  |  |  |
| A.AM.4 | create user  profile | 1) Admin A | 1 | Admin fills all required fields | 1.a | A input invalid values |  |
|  |  | 2) System S | 2 | Click create | 1.a.1 | S displays message  about invalid input |  |
|  |  |  | 3 | System create user |  |  |  |
|  |  |  |  |  |  |  |  |
| A.SM.1 | change tariffs | 1) Admin A |  | A open tarrifs page | 2.a | A input invalid values |  |
|  |  | 2) System S |  | A edit fileds of tarrif | 2.a.1 | S displays message  about invalid input |  |
|  |  |  |  | A click save |  |  |  |
|  |  |  |  | S save changes |  |  |  |
| A.SM.2 | review reports | 1) Admin A | 1 | A open reports page |  |  |  |
|  |  | 2) System S | 2 | A select type of report |  |  |  |
|  |  |  | 3 | S show reports |  |  |  |
|  |  |  |  |  |  |  |  |
| A.SM.3 | view users  dashboard | 1) Admin A | 1 | A open user dashboard |  |  | A.AM.4 |
|  |  | 2) System S | 2 | S shows dashboard |  |  |  |
|  |  |  |  |  |  |  |  |
| A.MD.1 | assign driver to a particular  car | 1) Admin A | 1 | Admin press edit driver from driver list |  |  | A.MD.2,A.MD.3 |
|  |  | 2) System S | 2 | A click button choose  car |  |  |  |
|  |  |  | 3 | Admin select car from  car list |  |  |  |
|  |  |  | 4 | System assign driver to selected car |  |  |  |
| A.MD.2 | create driver | 1) Admin A | 1 | Admin fills all required fields | 2.a | driver alrady exist |  |
|  |  | 2) System S | 2 | Click create |  |  |  |
|  |  |  | 3 | System create driver |  |  |  |
| A.MD.3 | create car | 1) Admin A | 1 | Admin fills all required fields | 2.a | car alrady exist |  |
|  |  | 2) System S | 2 | Click create |  |  |  |
|  |  |  | 3 | System create car |  |  |  |

Table 1 – Administrator Use Case description

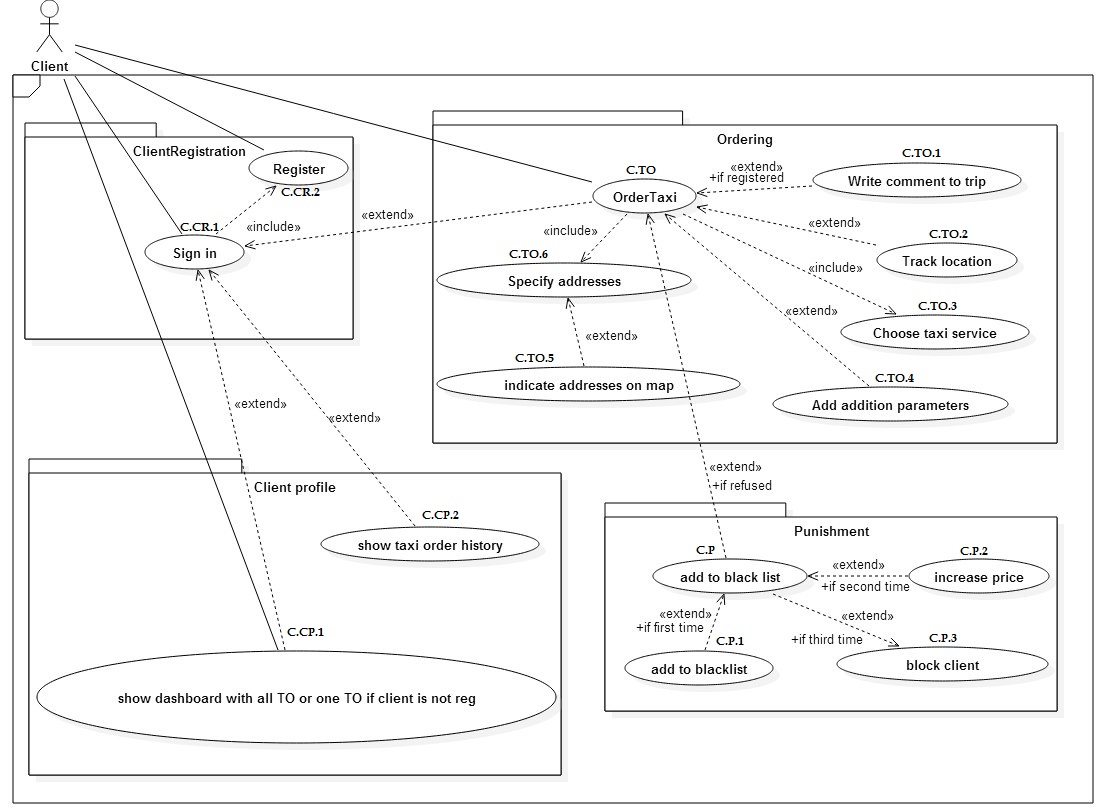


Figure 6 – Customer Use Case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Case ID** | **Use Case Name** | **Actor** | **Main Success Scenario** | | **Alternative Flow** | | **Precondition** |
| **Step** | **Step Description** | **Step** | **Step Description** |
| C.CR.1 | Sign In | 1) Client C | 1 | C open login page | 2.a | C didn't fill login fields | C.CR.2 |
|  |  | 2) System S | 2 | C enter login and password | 2.a.1 | S displays message about invalid input |  |
|  |  |  | 3 | C press button Log in | 2.b | C didn't enter valid data |  |
|  |  |  | 4 | S checks if user exist | 2.b.1 | S displays message about invalid input |  |
|  |  |  | 5 | S checks if client's account is not blocked | 4.a | Client with specified login data not exists in the system |  |
|  |  |  | 6 | S redirects client to Main Menu | 4.a.1 | S displays message that no user exists |  |
|  |  |  |  |  | 5.a | Client with specified login data is blocked |  |
|  |  |  |  |  | 5.a.1 | S displays message that user is blocked |  |
| C.CR.2 | Register | 1) Client C | 1 | C open login page | 2.a | C didn’t' fill mandatory fields |  |
|  |  | 2) System S | 2 | C enter mandatory registration data: email, password; and optional: phone number, first name, last name | 2.a.1 | S displays message about invalid input |  |
|  |  |  | 3 | C press button Register | 2.b | C didn't enter valid data |  |
|  |  |  | 4 | S checks if specified email not exists | 2.b.1 | S displays message about invalid input |  |
|  |  |  | 5 | S create new Client in the system | 4.a | Client with specified email exists in the system |  |
|  |  |  | 4 | S sends confirmation email with confirmation link | 4.a.1 | S displays message that user already exists |  |
|  |  |  | 5 | C gets confirmation email and click confirmation link |  |  |  |
|  |  |  | 6 | S redirects client to confirmation page |  |  |  |
|  |  |  | 7 | S mark C as activated |  |  |  |
|  |  |  | 6 | S redirects client to Main Menu |  |  |  |
| C.CP.1 | Show dashboard with all Tos or one TO if client is not registered | 1) Unregistered Client UC | 1 | C.CR.1 | 1.a | UC is not registered in the system |  |
|  | 2) Registered Client RC | 2 | UC open Client Dashboard page | 1.a.1 | UC open TO Information page |  |
|  | 3) System S | 3 | S shows the dasboard with TO | 1.a.2 | UC enters the tracking number in the field |  |
|  |  |  |  | 1.a.3 | UC clicks on submit |  |
|  |  |  |  | 1.a.4 | S searchs order by tracking number |  |
|  |  |  |  |  | 1.a.5 | S shows info about TO |  |
|  |  |  |  |  | 1.a.4.a | No TO was found by tracking number |  |
|  |  |  |  |  | 1.a.4.a.1 | S display message about no order existence |  |
| C.CP.2 | Show taxi order history | 1) Client C | 1 | C open Client Dashboard page |  |  | C.CR.1 |
|  | 2) System S | 2 | C clicks on Old Tos button |  |  |  |
|  |  | 3 | S shows the dasboard with all TOs |  |  |  |
| C.TO (I) | Order Taxi (part 1: filling fields and forming order) |  | 1 | C open Order page | 2.a | C didn't input phone number |  |
|  |  | 2 | C inputs phone number | 2.a.1 | S displays message that phone number is mandatory field |  |
|  |  | 3 | C inputs email | 3.a | C is logged in |  |
|  |  |  |  | 3.a.1 | S automatically fills email field |  |
|  |  |  |  | 3.a.1.b | C is not logged in |  |
|  |  |  |  |  | 3.a.1.b.1 | C didn't input email address |  |
|  |  |  |  |  | 3.a.1.b.2 | S displays message that email address is mandatory field |  |
| C.TO.1 | Write comment to trip | 1) Client C | 1 | C open Client Dashboard page | 5.a | C has already left comment about trip | C.CR.1 |
|  | 2) System S | 2 | S shows the dasboard with all TOs | 5.a.1 | S blocks comment input form |  |
|  |  |  | 3 | C clicks on any TO record |  |  |  |
|  |  |  | 4 | S show information about order |  |  |  |
|  |  |  | 5 | C inputs comment abot trip and submit |  |  |  |
|  |  |  | 6 | S add coment to TO information |  |  |  |
| C.TO.2 | Track Location | 1) Client C | 1 | S automatically tracks C location via Google Maps | 1.a | C prohibited automatically location tracking via Google Maps | C.TO |
|  |  | 2) System S | 2 | S marks C location with flag A | 1.a.1 | S does't track C location |  |
|  |  |  |  |  | 1.a.2 | S specifies default location marked with flag A |  |
| C.TO.3 | Choose taxi service | 1) Client C | 1 | S shows drop down list with existing services | 2.a | C didn't choose service | C.TO |
|  | 2) System S | 2 | C click on list and choose right service | 2.a.1 | S specifies service "Simple taxi" as default |  |
| C.TO.4 | Add aditional parameters | 1) Client C | 1 | S shows button Aditional Options |  |  | C.TO |
|  | 2) System S | 2 | C clicks on this button |  |  |  |
|  |  | 3 | S shows additional options menu under main order body |  |  |  |
|  |  |  | 4 | C specifies additional option if nessesary |  |  |  |
| C.TO.6 | Specify address | 1) Client C | 1 | S shows fields to input address from and address to and map | 3.a | C input invalid address | C.TO |
|  | 2) System S | 2 | C fill one or both fields to specifies address from and address to | 3.a.2 | S is not able to parse address via Google Map |  |
|  |  |  | 3 | S parses address via Google Map | 3.a.3 | S displays message about invalid address |  |
|  |  |  | 4 | S marks specified addresses with flag A for address from and flag B for address to on map |  |  |  |
| C.TO.5 | Indicate address on map | 1) Client C | 1 | S shows map with to flags: A for address from, B for address to |  |  | C.TO |
|  | 2) System S | 2 | C drags and drops any flag cpecifying some address |  |  |  |
|  |  | 3 | S automatically change adresses in the address fields |  |  |  |
| C.TO (II) | Order Taxi (part 2: submiting order) | 1) Client C | 1 | S shows button to submit order |  |  | C.TO.2-5 |
|  | 2) System S | 2 | C click on submit button |  |  |  |
|  |  | 3 | S create new TO with status "Queued" |  |  |  |
|  |  | 4 | S sends to C email with tracking number of TO |  |  |  |
| C.P | Add to blacklist | 1) Client C | 1 | C change status of TO to "Refused" | 4.a | If second time then C.P.2 | C.CP.1 |
|  | 2) System S | 2 | S increase the amount of refusal | 4.b | If second time then C.P.3 |  |
|  |  |  | 3 | S checks how many times C has refused TO |  |  |  |
|  |  |  | 4 | If first time then C.P.1 |  |  |  |
| C.P.1 | Add to black list | 1) Client C | 1 | S add C to blacklist |  |  |  |
|  |  | 2) System S |  |  |  |  |  |
| C.P.2 | Add to black list | 1) Client C | 1 | S increase price for C |  |  |  |
|  |  | 2) System S |  |  |  |  |  |
| C.P.3 | Add to black list | 1) Client C | 1 | S blocks C ( since this time C is not able to use TO Services |  |  |  |
|  |  | 2) System S |  |  |  |  |  |

Table 2 – Customer Use Case description

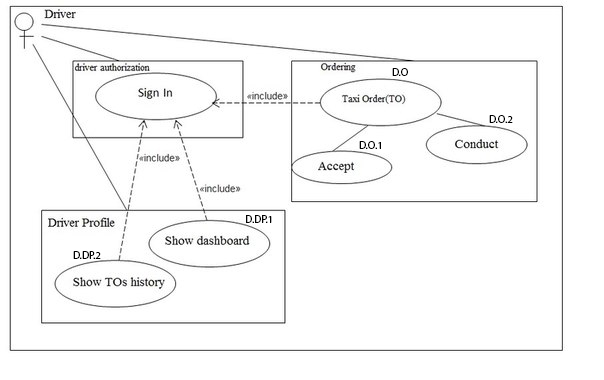


Figure 6 – Driver Use Case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Case ID** | **Use Case Name** | **Actor** | **Main Success Scenario** | | **Alternative Flow** | | **Precondition** |
| **Step** | **Step Description** | **Step** | **Step Description** |
| D.DA | Sign In | 1) Driver D | 1 | D opens login page | 2.a | D didn't fill login fields |  |
|  |  | 2) System S | 2 | D enters login and password | 2.a.1 | S displays message about invalid input |  |
|  |  |  | 3 | D press button Log in | 2.b | D didn't enter valid data |  |
|  |  |  | 4 | S checks if user exist | 2.b.1 | S displays message about invalid input |  |
|  |  |  | 5 | S redirects D to Main Menu | 4.a | Driver with specified login data not exists in the system |  |
|  |  |  |  |  | 4.a.1 | S displays message that no user exists |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| D.DP.1 | Show dassboard | 1) Driver D | 1 | D opens Driver Dashboard page |  |  | D.DA |
|  |  | 2) System S | 2 | S shows the dasboard |  |  |  |
| D.DP.2 | Show Tos history | 1) Driver D | 1 | D opens Driver TO history page |  |  | D.DP.1 |
|  |  | 2) System S | 2 | S shows the TO history |  |  |  |
| D.DO.1 | Accept | 1) Driver D | 1 | S shows to driver TO in status "Queued" |  |  | D.DP.1 |
|  |  | 2) System S | 2 | Driver press Assign |  |  |  |
|  |  |  | 3 | S changes status of TO to "Assigned" |  |  |  |
| D.DO.2 | Conduct | 1) Driver D | 1 | D opens active TO page | 3.a | Driver presses Refuse | D.DO.1 |
|  |  | 2) System S | 2 | S shows all active Tos assigned to D | 3.a.1 | S changes status of TO to "Refused" |  |
|  |  |  | 3 | D picks up Client and presses In Progress | 3.a.2 | S notifies client via email about refusal |  |
|  |  |  | 4 | S changes status of TO to "In progress" |  |  |  |
|  |  |  | 5 | S notifies client via emaiL |  |  |  |
|  |  |  | 6 | D completes TO and press Complete |  |  |  |
|  |  |  | 7 | S changes status of TO to "Completed" |  |  |  |
|  |  |  |  | S notifies client via email |  |  |  |

Table 3 – Driver Use Case description

# TrackSee Web Application Overview

TrackSee web application allows to fulfil and to manage customer orders. It has three main design subsections according to user roles in the system: customer, administrator and driver.

## General User Design Subsection

This is the main design subsection for all user roles: administrator, driver, registered customer and non-registered customer.

The main bookmarks of this design module are:

Home page

Log in page

Sign in page

Order page

Dashboard page

Order information page

## Personal Administrator Design Subsection

This is the personal administrator subsection with only admin user role permission.

The main pages of this design module are:

Main page

Driver accounts page

Cars page

Reports page

Tariffs page

Block user page

Groups and roles

## Personal Driver Design Subsection

This is the personal driver subsection with driver and admin user roles permission.

The main pages of this design module are:

Freeorders page

Assigned orders page

History of orders page

Own profile page

# TrackSee Web Application Detailed Description

This Document section describes functionality of user interface.

## General User Design Subsection

All users types can use this subsection of web application for log in, sign in, placing order, tracking order and going to the personal cabinet.

### Home Page

On the main page paced slider that provides ability for quick redirect to taxi ordering, registration and order tracking.

|  |  |  |
| --- | --- | --- |
| Fast taxi order | Sign up | Track your taxi order |
| Provides redirect to order page where user can order taxi. | Provides redirect to sign up page where user can sign up. | Provides redirect to order information page where user can track his order. |

Table 4 – Slides

### Sign In Page

Login page allows user to login his account.

### Order Page

Order page allows user to make fast taxi order or extended taxi order with additional options.  
To make fast taxi order user must fill such required fields: phone number, email, address from and address destination. Address origin and address destination can be selected through Google maps or written by hand. In case selecting address on Google maps address fields filling automatically and vice versa. Price calculation is performed continuously at each change of address origin and address destination. Option way of payment is set to default value “Cash”, if user changed way of payment this will be an informative character for the driver. In case user wants to make extended taxi order he can open a block of additional options the filling of which is mandatory. In this block user can change the value of such options: service, car category, driver sex, music style, animal transportation, free Wi-Fi, non smoking driver, air conditioner. Option “service” has different meanings, most of which has its own characteristics.

### Dashboard Page

Client dashboard allows user to view his orders and get price list. User can go to extended form of each order that he made, where he can view his order. Also it is possible to change, refuse and leave comments to this order depending on the status of the order.

### Order Information Page

Order information page allows user to track his order. He can find and view his order by tracking number. Registered users can view only their orders when they authorized. Non registered and non authorized users can’t view orders authorized users, they can view only orders that made non registered users. Also it is possible to change, refuse and leave comments to this order depending on the status of the order.

## Personal Administrator Design Subsection

### Driver Accounts Page

Driver account page allows to create/review/update/delete every driver. Administrator must click on menu panel and choose a driver management option. Selecting “create driver” administrator is being redirected to form, where he must fill in the fields (email, password, phone). Selecting “Driver List” administrator is being redirected page where he can review any driver and choose driver to review driver profile.

### Cars Page

Cars page allows to create/review/update/delete cars. Administrator must click on menu panel and choose a car management option. Selecting “create car” administrator is being redirected to form where he must fill in the fields (car number, car model, color, car category, way of payment, animal transportation applicable, free wi-fi, air conditioner). Selecting “Car List” administrator is being redirected to page where he can review any car and choose car for updating car entity. There are such buttons as “delete” and “edit”. Selecting “delete” administrator deletes car(if this car is not assigned to any driver). If car is assigned to a driver, administrator sees the warning window. Selecting “edit” administrator is being redirected to form, where he can update car fields.

### Reports Page

1. Services reports page allows to see all system reports in one page, without transitions to other pages. It consists of such report blocks:

* New orders per period – in this block administrator can review the number of new orders for a certain period. Administrator must choose time range in date picker panel to view report for selected period.
* Service profitability by month – in this block administrator can review profitability of the system for the selected month. Administrator must choose month and year in date picker panel to view report for selected period.
* Most profitable taxi service – in this block administrator can review profitability for each taxi service separately. Table consist of four rows - periods (week, month, year, decade) and nine columns - services (simple taxi, sober driver, convey employees, guest delivery, cargo taxi, taxi for long term, meet my guest, celebration taxi, foodstuff delivery). Information is available immediately after the page is loaded.
* Most popular driver and car category – in this block administrator can review most popular driver and car category from all existing orders in the system. Information is available immediately after the page is loaded.
* Most popular music style and additional car options overall – in this block administrator can review most popular music style and additional car options (credit card, animal transportation, free Wi-Fi, non-smoking driver, air-conditioner) from all existing orders in the system. Information is available immediately after the page is loaded.
* Most popular music style and additional car options for current customer – in this block administrator can review most popular music style and additional car options (credit card, animal transportation, free Wi-Fi, non-smoking driver, air-conditioner) from all existing orders in the system for current customer. Administrator must choose customer from drop-down menu to view his statistics.

It is possible to save all reports presented on the page in the form of an Excel spreadsheet by pressing “Save as XLSX” button.

Data are transmitted to server with AJAX technology for dynamic refresh page.

1. Most popular additional options. In this page administrator can review most popular option for every customer. Administrator must choose customer from drop-down menu to review his statistics.
2. Most profitable service. In this page administrator can review most profitable service for any period. Administrator must choose start date and end date in date-panel to review statistics for selecting period.

### Tariffs Page

Admin page allows to see and to change tariffs for all available services. It is displayed on the “Tariff” page. It consists data from “taxi\_price” table on DB (price per km, price per min, car category, weekend, night tariff) and provides an ability to change some tariff. All input fields have validation for input number format. Data are transmitted to server with AJAX technology for dynamic refresh page.

### Groups And Roles

“Group and roles” page allows administrator to create, remove and modify groups. It is also possible to add/remove users to group. Administrator can change roles for users here. There is a validation for group name when administrator creates group (name has to consist of two – twenty eight symbols). Data to this page is saved in table “service\_user” in DB. All data transmit works through AJAX technology.

## Personal Driver Design Subsection

### Free Orders Page

“Free orders page” allows driver to see all appropriate orders and assign them. Available orders are selected on the basis of a comparison of order options and parameters of the car. Such parameters are availability of free wi-fi and air conditioner in car and possibility to accept visa card payments and animal transportation. Driver's sex is also considered if it is listed in the customer's request. For driver is also available orders with lower car class than his. For business class and van cars – economy class orders. The decision whether to accept the order with lower price takes the driver. If customer pointed time of car arrival it displayed in order details. If date and time not pointed by customer, to accept order, driver must pick up time when he planning to arrive to customer. Once order accepted, order is in state “Assigned” and driver redirects to “assigned orders page”. Data to this page is saved in tables “taxi\_order” and “taxi\_order\_item” in DB.

### Assigned Orders Page

“Assigned orders page” allows driver to process orders. When driver picks up customer order changes its state to “In Progress”As long as driver and customer are getting to destination, TO has status “In progress”. When Customer User gets his destination, TO changes its state from “In progress” to “Completed”. Driver has ability to refuse order in two cases. Firs case is when customer not arrived to car – order change his status to “Refused” and customer receive plus one to number of orders he refused. Second case when driver accepted order but no longer has ability to arrive to customer. In this case order change his status to “Queued” and customer is notified by email about this. Data to this page is saved in tables “taxi\_order”, “taxi\_order\_item” and “service\_user” in DB.

### History Of Orders Page

“History of orders” page display history of drivers “Refused” and “Completed” orders. Data to this page is saved in tables “taxi\_order” and “taxi\_order\_item” in DB.

### Own Profile Page

“Profile page” display drivers profile and information about his car. Data to this page is saved in tables “service\_user” and “car” in DB.