Vidzeme University of Applied Sciences

**Faculty of Engineering**

Python OOP and Modelling

group 01

practical work #3

Valmiera, 2021

Table of contents

[1 Use case description development 4](#_Toc90591085)

[1.1 Research and clarifications 4](#_Toc90591086)

[1.2 Use case diagram update 4](#_Toc90591087)

[2 Class definition 5](#_Toc90591088)

[3 Used resources 6](#_Toc90591089)

|  |  |  |  |
| --- | --- | --- | --- |
| Document versions | | | |
| Version | Status / Changes | Date | Author |
| 1.0 | Initial document release | 05.12.2021 | A.Jenerts |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Contacts and responsible (-s) | | | |
| Name Surname | Department | Position | Contact information (e-mail) |
| Signe Grosberga | Group 01 | Member | signe.grosberga@va.lv |
| Edmunds Sprūdžs | Group 01 | Member | edmunds.sprudzs@va.lv |
| Andris Jenerts | Group 01 | Member | andris.jenerts@va.lv |

# Use case description development

Detailed use case descriptions are provided in separated document attached together with this document

## Research and clarifications

Once team had researched online and investigated given template and examples only thing that needed clarification for the team members and for better documentation ware differences between alternative sequences and error events.

Alternative sequence term is used for user or system caused situations that require different action path, e.g., invokes different use case. Simple example for such situation is when user tries to log in to the app, but system can’t recognize such email so redirects to the registration. Another example is when user tries to use GPS location, but app hasn’t received permission to use user location, app initializes permission request for the user.

Error events are events that are caused by user or system error. The basic example is wrong password or inputting nonnumeric value in only numeric value field. In these cases, system just informs user about that error occurred and doesn’t take any action.

## Use case diagram update

Diagram

Description automatically generated Once team started to produce more specific descriptions for the use cases it became clear that more use cases and references should be added to the use case diagram. Most notably authentication use case was missing and was necessary to complete this task in best possible quality. Updates use case diagram can be seen in the 1. Image below.

*Image 1* Updated Use Case diagram

# Class definition

Diagram

Description automatically generatedFive basic classes of the system were identified, three of them are subclasses of other one. In the 2nd image you can see all defined classes and simple relations between them.

*Image 2* Defined Classes

All users in the system have some similarities, e.g., they have unique identifiers, emails, passwords, information about their authentication status etc. That is the reason for the abstract class with the name User.

Drivers, Operators and Admins although all share same attributes as users, they have different specific attributes. That is the reason for separate classes for each of them. For example, all drivers will specify what type of fuel they are using, what loyalty cards they own, etc., but these are not parameters that Operator or Admin will specify.

Similarly, all fuel prices have same structure, they all have price, fuel type, location, confidence coefficient, etc.

# Used resources

1. Fakhroutdinov, K. Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors). Retrieved 16 December 2021, from <https://www.uml-diagrams.org/use-case-diagrams.html>
2. Fakhroutdinov, K. UML use case include relationship shows that behavior of the included use case is inserted into the behavior of the including use case. Retrieved 16 December 2021, from <https://www.uml-diagrams.org/use-case-include.html>