Lift above yourself (break bad habits)

Analysis and Design Document

Student: Maghiar Catalin Andrei

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# The project is an application that will help the user break their bad habits depending on the severity and the time they want to manage to forget that old bad habit. The application will send regular notifications reminding the user what they should be doing. Depending on their penalties (if they miss one day or multiple consecutive days) the applications will send notifications more often.

# Elaboration – Iteration 1.1

# Domain Model

![A close up of text on a white surface

Description automatically generated]()

# Architectural Design

## Conceptual Architecture

I will use the MVC approach (Model, View, Controller) for my application. As seen in the picture above, the classes that will be the in the Model are the User and the Vice classes. The Controller will have the UserProfile and the Planner. As for the view, there will be just the interface that the user will see to create the profile and choose their vice and a plan for their bad habit that they should follow. This pattern seemed easier for the kind of application that I wanted to make, it will probably be changed depending on how the application will evolve, but for the moment it seems easier especially because the application is mostly intended for use with mobile phones.

## Package Design

![A close up of a map

Description automatically generated]()

## Component and Deployment Diagrams

![A close up of a whiteboard

Description automatically generated]()

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography