**System High-Level Design Document**

**Mobile Opensource**

**Healthly Ever After Project**

|  |  |
| --- | --- |
| 지도 교수 **:** | 정우진 교수님 |
| 조 원**:** | 송민수 안현수 쿠엔틴 일리자테오 |
| Group : | E |

Contents

1. Introduction …………………………………………………………… 1
   1. Purpose of document …………………………………………… 2
   2. Project scope ……………………………………………………… 3
   3. Subject Matter Experts Agreement List ……………………… 4
   4. Definition
   5. Desired Behavior
2. General Description………………………………………………… 5
   1. Product Perspective……………………………………………… 6
   2. Tools used………………………………………………………… 7
   3. General Constraints……………………………………………… 8

2.4. Special Assumptions……………………………………………… 9

1. System Architecture ………………………………………………… 10
   1. Requirement Table ………………………………………………… 11
   2. Usecase Diagram ………………………………………………… 12
   3. Sequence Diagram ………………………………………………… 13 3.4 Class Digram …………………………………………………… 14
2. Project Propulsion System ……………………………………… 15
   1. Organizational Role ………………………………………………… 16
   2. Detailed Schedule Promotion Plan …………………………… 17
   3. Documentation Planning ………………………………………… 18
   4. Commercialization Test Plan …………………………………… 19
3. Project Management ………………………………………… 20
   1. Coding Style Guide Line …………………………………………… 21

**1. Introduction**

## 1.1 Purpose of Document

**T**his document will help narrow and specify the requirements and high level design of our health and fitness program, HealthilyEverAfter. This document will help assign roles to all members of our group and what each member will be responsible for in this document and program.

The outcome of this document is the enable all members to understand their roles which would enable them to produce the best and most effective program possible.

## 1.2 Project scope

Healthily Ever After is a flexible, user-friendly mobile application to help users track calorie intake, and amount walked. It will also give the user dietary and exercise recommendations based on height, weight and user activity. Healthily Ever After will be able to eliminate the need for multiple different applications to track calories, measurements and exercise amount.

## 1.3 Subject Matter Experts Agreement List

This section should list all key people involved in the project. It will ensure that all mandatory reviewers have reviewed and agree to the requirements and proposed architecture.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title/Role** | **Mandatory Reviewer (Y/N)** | **Agreed To** |
| Quentin Pionnier | Programmer | Y |  |
| Ahn Hyunsu | Programmer | Y |  |
| Song Minsoo | Programmer | Y |  |
| Elijah Teo | Programmer | Y |  |
| Woojin Jeong | Client | Y | Give appropriate comments and review |

1.4 Definition

**Blog**: Journal that is assigned to students by the staff

**Entry:** Attribute of a blog, which is a post in response to the blog topic

**Comment:** A different response to the overall entry and blog

## 1.5 Desired Behavior

This program is designed to allow any person to make an account and start tracking their weight, food intake, exercise and sleep. Allowing them to make more informed decisions about their health.

They will have motivation through competition among their friends and other users. This program is also designed to allow administrators to have minimal interaction with users other than removing users that want to be removed.

# General Description

2.1. Product Perspective

The application HealthlyEverAfter is comprised of several different components.

Some of these components will be programmed, while others will be implementations of open-source programs. The language implementation will be dictated by it’s purpose. Since our application has the login progress but we only use the phone server not the actual external database, the administrative and user interface are not devided. There’s no administrator version, only the basic users, which can only view their running history and diet. This page is automatically displayed based on their phone ip address.

2.2. Tools used

1. Draw.io based Sequence design program, is used to generate all of the diagrams used in analysis and design phases of the project.
2. Visual Paradigm Online (VP Online) supports UML, ERD and Organization Chart. we can draw Use Case Diagram quickly through the intuitive UML drawing editor.
3. The project have created apps for IOS and Android using HTML language, in Visual Cordova.
4. A kind of Web-based API (EG a REST API) is required to implement a login server. We can implement it using visual studios and ASP.NET. This API will serve as a link between app and database.
   1. General Constraints

The HealthlyEverAfter must be user friendly and as automated as possible. There’s no administrator so the developer should do all the initial setup. And users should not be required to know any of the working. Without logging in, the user will only have the ability to view that IP’s current average and history. After logging in, that user then has the ability to change settings, start running and see the user history.

* 1. Special Assumptions

This project is based on the idea of tracking users running record and the goal is to make this idea a reality using Software Engineering practices. In doing so, many documents are created, and it is executed with these following assumptions………

- Users only **run** to exercise and record their exercise.

- Users don’t rest during the jogging.

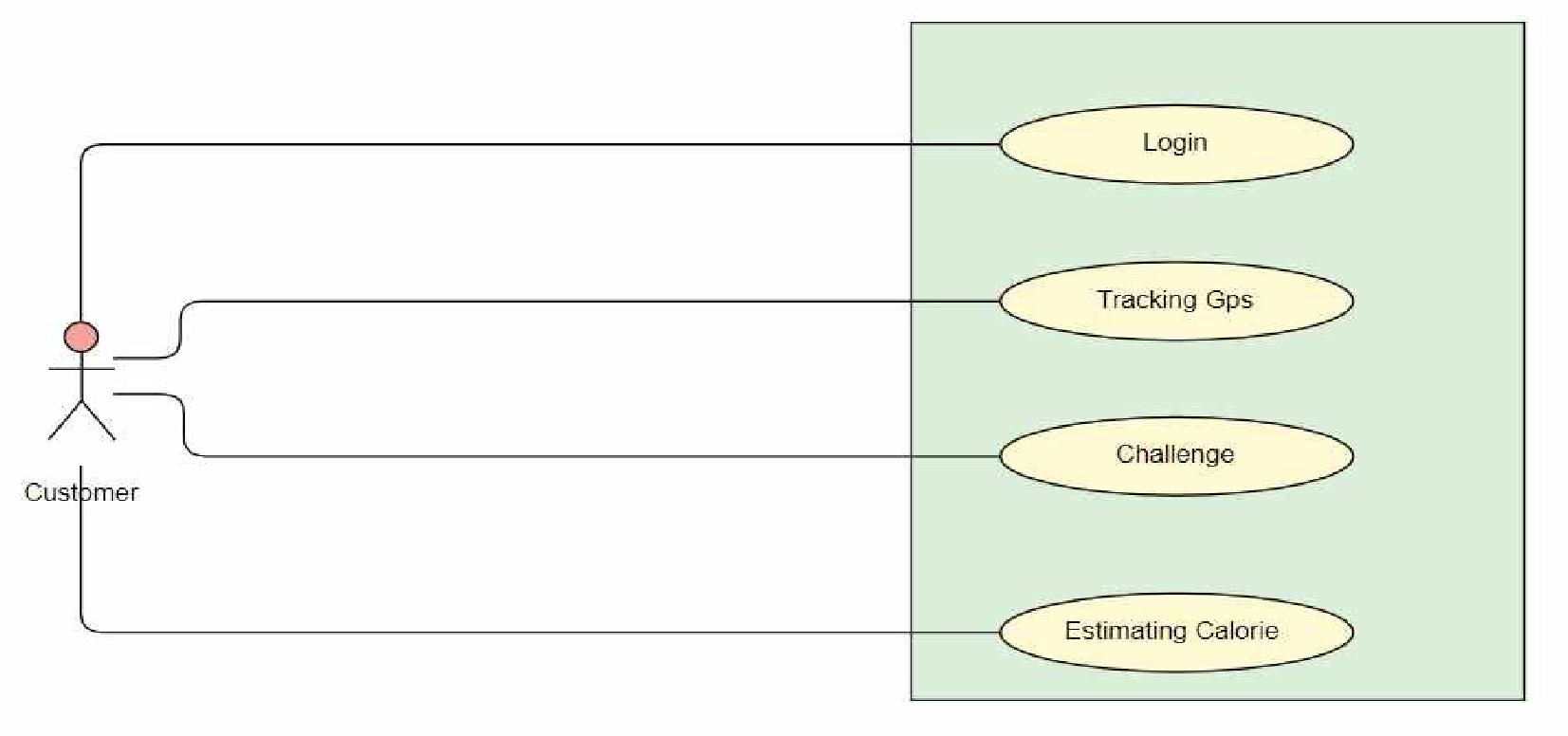
- Users only run in one steady speed.

- Users only login through their own phone which allow us not to make the actual external DB

1. **System Architecture**
   1. Requirements Table:

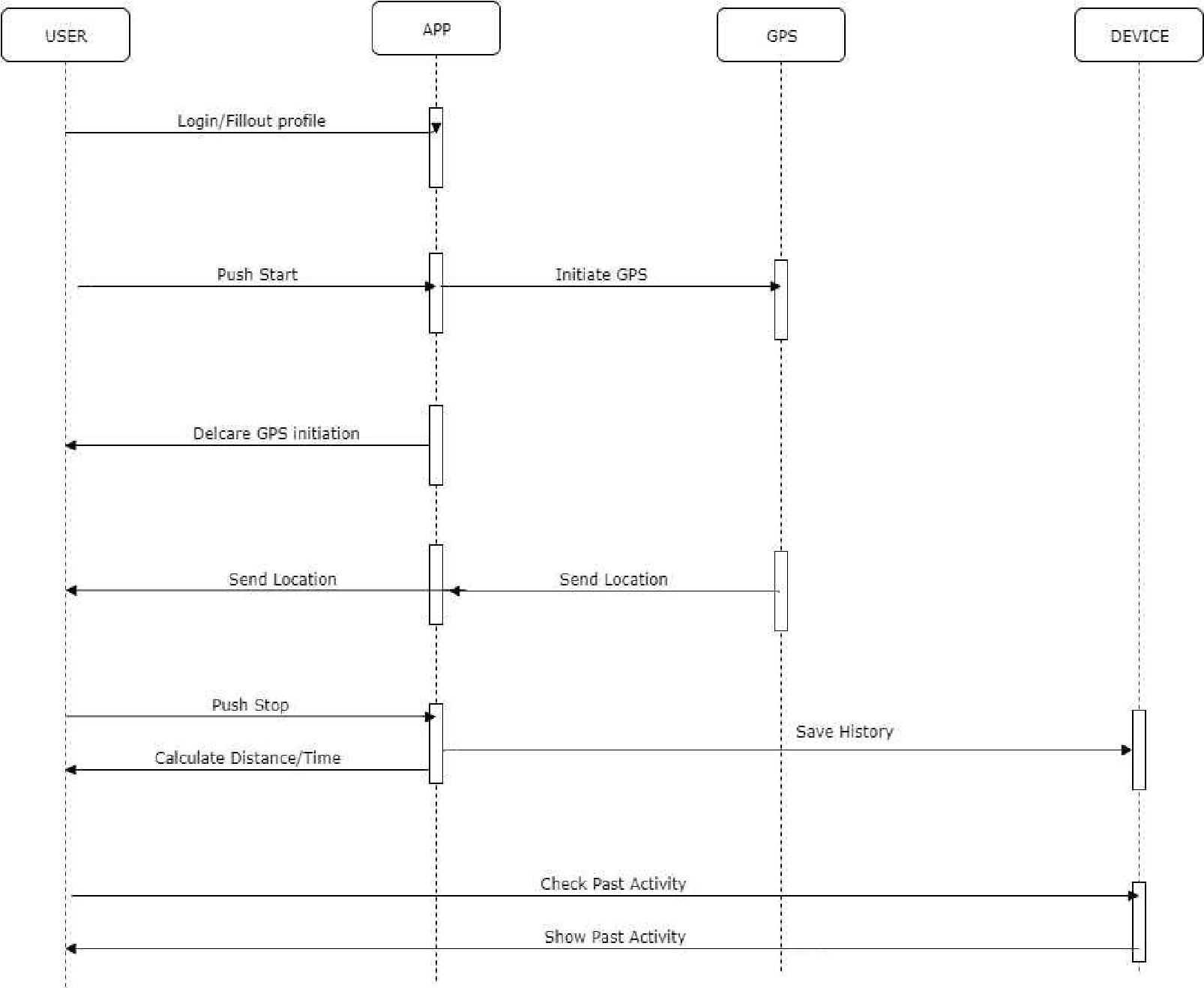
|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Description | Requirement Class | V&V Method |
| Log in | Any person can access this page, however, only existing users can successfully log in. | Required |  |
| Sign up | Any person can access this page and make an account. Assumption is that one user per person. | Required |  |
| GPS distance tracker | Any person can access this page, it will track the distance the user has moved through GPS. | Required |  |
| Measurements | Only users can access this page as it will track and measure the users height, weight and fat percentage | Required |  |
| Challenges | Any person can access this page, however, only  users can interact with and see other users completed challenges | Required |  |
| Calorie tracker | Any person can access this page, however, only users can track daily intake and output.  Guests can only see the amount of calories of a food item | Required  (The difference between user and guest is optional) |  |
| R e c o m m e n d e d  exercise | Based on measurements and amount of times user exercises a week. Only users can access this page. | Required |  |
| R e c o m m e n d e d  meals | Based on measurements, and food intake. Only users can access this page. | Required |  |
| Sleep Pattern | Tracks sleep timing. Only users can access this page. | Optional |  |
| Social media sharing | Connects application to selected social media and makes a post. Only users can access this page | Optional |  |
| Log out | The users logged in are able to log out | Required |  |

* 1. Use Case Diagram



A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. Then, our project need Login, Tracking GPS, Challenge, Estimating Calorie

* 1. Sequence Diagram



* + 1. sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

3.3 Class diagram![](data:application/octet-stream;base64,)

A type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

1. **Project Propulsion System**
   1. Organizational Role

|  |  |
| --- | --- |
| **구분** | **역할 및 책임** |
| **송민수** | * Draft and Management project deliverables * System building direction adjustment * Management/Supervision in project process * User history interface implementation |
| **쿠엔틴** | * Overall user interface and design implementation * Development work support and implementation review/ advice * Design and implementation of location tracking module interface with real-time location display |
| **엘라이자**  **테오** | * Draft and Management project deliverables * User management and login process * Design and implementation of login module and detailed interface |
| **안현수** | * Final decision of project overall and key issues * Devise ideas and topic/ Find and support project-related information * Project Proposal and SRS, High Level Design Document General |

* 1. Detailed Schedule Promotion Plan

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | **Activity** | **October** | | |  |  | **November** | |  | **Dec** |
| **1st** | **2nd** | **3rd** | **4th** | **1st** | **2nd** | **3rd** | **4th** | **1st** |
| **Plan** | Initiation Report |  |  |  |  |  |  |  |  |  |
| Project Proposal |  |  |  |  |  |  |  |  |  |
| Requirements Definition |  |  |  |  |  |  |  |  |  |
| **Analysis** | System Analysis |  |  |  |  |  |  |  |  |  |
| **Design** | Module Interface Design |  |  |  |  |  |  |  |  |  |
| Test Design |  |  |  |  |  |  |  |  |  |
| **Implementation** | UI implementation |  |  |  |  |  |  |  |  |  |
| Location tracking function using GPS |  |  |  |  |  |  |  |  |  |
| Unit test planning and testing |  |  |  |  |  |  |  |  |  |
| **Test** | Integration Test |  |  |  |  |  |  |  |  |  |
| System Test |  |  |  |  |  |  |  |  |  |
| Compatibility Test |  |  |  |  |  |  |  |  |  |
| **Termination** | Inspection and Maintenance |  |  |  |  |  |  |  |  |  |
| SRS Trace Sheet |  |  |  |  |  |  |  |  |  |

* 1. Documentation Planning

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Results** | Progress date | **Remark** |
| Planning | Project Proposal  (SRS) | 2019.09.24-2019.10.08 |  |
| **Analysis** | Usecase Diagram | 2019. 10. 7 |  |
| System Sequence Diagram | 2019. 10. 7 |  |
| **Design** | Class Diagram | 2019. 10. 17 |  |
| Sequence Diagram | 2019. 10. 17 |  |
| High Level Design Document | 2019.10.17-  2019.11.28 |  |
| **Test** | Test Result Document | 2019. 11.30-2019.12.2 |  |
| **Installation&**  **Execution** | User Manual | 2019. 12. 04 |  |
| SRS Trace Sheet | 2019. 12. 04 |  |
|  |

* 1. Commercialization Test Plan

|  |  |
| --- | --- |
| **Test Type** | Test Content |
| **Unit**  **Testing** | The test is conducted individually based on the individual part of each member, and it finds and corrects bugs and malfunctions of the UI screen output and code in real time.  - Is the UI displayed on the screen at the correct rate without clipping?  - Does the correct operation correspond to the click of a button to activate the function?  - Do you correctly map your location using your smartphone's IP address?  - Do you output route tracking and distance calculations according to movement within the specified error range?   ·  - Does the usage history show up correctly in the usage history?  - Does the user get the appropriate error message when using the application in the wrong way?  The goal is to reduce the number of bugs that can be fixed at this stage to reduce time for future maintenance and integration testing, system testing, and usability testing. |
| Integration Testing | Check how well each module we implement integrates with the UI, and works well without any bounces.   * Does it output the correct activity screen based on your actions?   · * Does it have the necessary functions activated in a short time when moving between modules? * Does the overall rental and return process of the application work well? |

|  |  |
| --- | --- |
| System Testing | Assess how closely you follow the requirements definitions we made during the project planning phase.  Evaluate the requirements we created in the project proposal in a detail table. |
| Usability  Testing | Before the program is completed, the inconvenience of the process is compensated for from the user's point of view, and the design aspect for a cleaner UI is evaluated. Everyone in the group shares their experiences with using the application, and complements the common areas.   * Can users use key features from app launch to exit without difficulty? * Can users use each service without misunderstanding through clear expression and UI design? * Does the user have difficulty clicking the button and entering text? * Can the user receive major system messages in each operation? |

1. Project Management
   1. Coding Style Guide Line

< Basic Guideline >

□ Lines within the class do not exceed 1000 lines.

□ All classes and methods start with annotations.

□ Write only one declaration on one line.

□ Insert an enter character between codes that can be logically separated.

□ When setting the return value, clearly state it in the notes.

< Naming guidelines >

□ The choice of variable name uses meaningful words to identify its intended use.

□ Avoid excessive omissions when selecting variable names.

□ The class and interface name must be a noun and for compound words, the first letter of each word is capitalized.

□ Classes and interface names are written simply and explicitly.

□ If the name of the method is a compound word, the first word begins with a lowercase letter and the first letter of the subsequent word uses capital letters.

□ The first character in the variable name starts with a lowercase letter, and the first letter in each inner word starts with an uppercase letter.

□ Do not use the underbar (\_) in the variable name as a start.

□ All classes and methods start with annotations.

□ The names of the variables declared class constants and ANSI constants are all capitalized, and each word is separated by an underbar.