Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering II**

BeHealthy

Software Requirements Specifications

Donia Khaled

Safwa Salman

Mahmoud Samy Saeed

Hager Sobeah

Fall-2016

Contents

[Team 3](#_Toc447313429)

[Document Purpose and Audience 3](#_Toc447313430)

[Introduction 3](#_Toc447313431)

[Software Purpose 3](#_Toc447313432)

[Software Scope 4](#_Toc447313433)

[Definitions, acronyms, and abbreviations 4](#_Toc447313434)

[Requirements 4](#_Toc447313435)

[Functional Requirements 4](#_Toc447313436)

[Non Functional Requirements 5](#_Toc447313437)

[System Models 4](#_Toc447313438)

[Use Case Model 6](#_Toc447313439)

[Use Case Tables 7](#_Toc447313440)

[Ownership Report 20](#_Toc447313441)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20146011 | Mahmoud Samy | mahmoudsamy995@gmail.com | 01097690138 |
| 20146013 | Hager Sobeah | hagersobeah@gmail.com | 01202590820 |
| 20146018 | Donia Khaled | doniaakhaled@gmail.com | 01001366636 |
| 20146009 | Safwa Mohammed |  | 01125549487 |

# Document Purpose and Audience

* **Software Requirements Specification is a Formal document that describes the functionality of the software using the use cases model and table.**
* **Expected Audience : Product Owner - Customer**

# Introduction

# Software Purpose

**The purpose of this software is to help users lead a healthy lifestyle as the name suggests. It can help with remembering medicine times, when they’re supposed to drink water, keeps regular checkups on a schedule and calculates the entered calories for every day.**

# Software Scope

**The BeHealthy app will be running on an Android interface mainly, backed with a web server that holds all needed information (for example: a table with all food calories) and user database.**

# Definitions, acronyms, and abbreviations

* **None**

# Requirements

# Functional Requirements

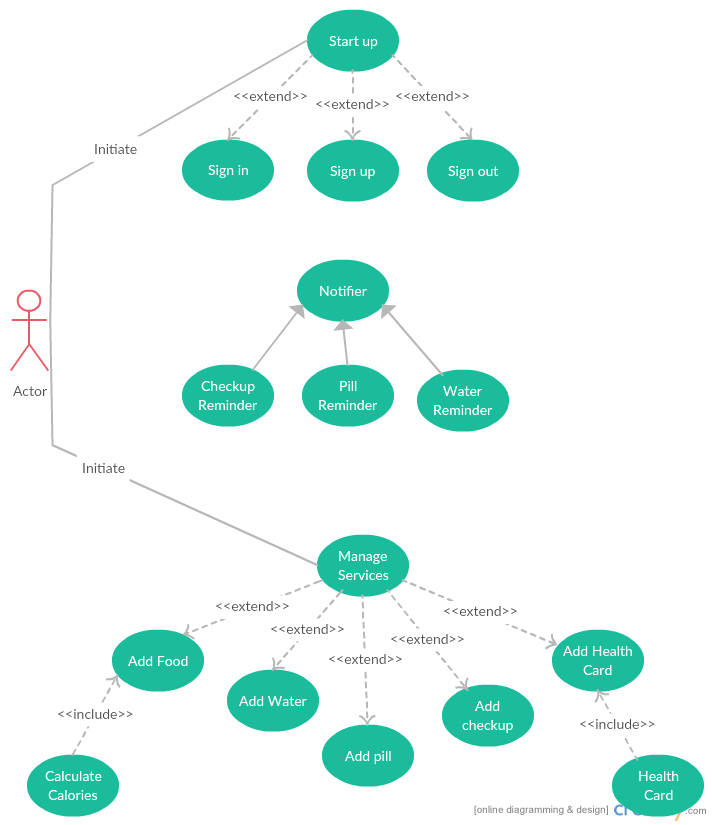
* **The user first creates an account to login.**
* **Adds information to their health card.**
* **Adds pills, if any.**
* **Adds/searches for food items throughout the day.**
* **Adds cups of water drunk.**
* **Views and calculates number of calories consumed per day.**
* **System notifies user to drink water.**
* **System notifies user to take pill and go to the doctor.**

# Non Functional Requirements

1. **Documentation: to be as a guide for users.**
2. **Compatibility: Compatible with android and web platforms.**
3. **Response Time: at most 5 seconds for each request.**
4. **Security: proves security to personal information.**
5. **Reliability: software's ability to function, given environmental conditions, for a particular amount of time**
6. **Usability: startup a use guide will load to go through the whole system and how it could be used**
7. **Performance**

* **Adding an item to the database is done in at most 3 seconds.**
* **Searching an item in the database takes at most 5 seconds.**
* **Notifications are sent constantly, and at exact given times.**

**System Models**

**Use case Model**

<<include>>

<<include>>

<<include>>

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Sign in | |
| Actors: | User | |
| Pre-conditions: | User starts App | |
| Post-conditions: | User Sign in the app successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter Username and Password. |  |
|  | 2- System checks database |
| 3- User sign in successfully |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters username and password. |  |
|  | 2- e-mail or password is incorrect. |
|  |  | 3- System rejects entry. |
|  |  | 4-Asks user to enter data again. |
|  | 5-User enters data again. |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Sign up | |
| Actors: | User | |
| Pre-conditions: | User starts App | |
| Post-conditions: | User Signs up successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters e-mail, password and other details. |  |
|  | 2- System verifies data. |
| 3- User signs up successfully |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters e-mail, password and other details. |  |
|  | 2- System checks database.  3- System rejects entry.  4-Asks user to enter data again. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Sign Out | |
| Actors: | User | |
| Pre-conditions: | User is signed in | |
| Post-conditions: | User logout successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User presses sign out button. |  |
|  | 2- System allows the user to sign out and saves a backup of his data. |
| Exceptions: | **User Action** | **System Action** |
| 1- User presses logout button. |  |
|  | 2- System fails to save a backup. |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Pill Reminder | |
| Actors: | User | |
| Pre-conditions: | User added pill name and time. | |
| Post-conditions: | System sends a notification to the user. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-System sends notification to the user. |
| 2-User accepts the notification. |  |
| Exceptions: | **User Action** | **System Action** |
|  |  | 1-System sends notification to the user. |
|  | 2-User didn’t accept the notification. |  |
|  |  | 3-System notifies the user again after 15 mins until it gets user’s response. |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Checkup Reminder | |
| Actors: | User | |
| Pre-conditions: | A checkup time is already assigned. | |
| Post-conditions: | A checkup notification is sent to the user. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-System sends the checkup notification at the exact time that was already assigned. |
|  | 2-User accepts the notification. |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1-System sends a checkup notification. |
| 2-no response from the user. |  |
|  | 3-System notifies the user again after 1 day. |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Water Reminder | |
| Actors: | User | |
| Pre-conditions: | User is signed in. | |
| Post-conditions: | Notification is sent to the user. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-System sends a notification with a health tip every 2 hrs. |
| 2-User accepts the notification. |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1-System doesn’t reach the minimum amount of water determined. |
|  | 2-System notifies the user to add water. |
|  | 3-User adds water. |  |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Add Water | |
| Actors: | User | |
| Pre-conditions: | User is signed in. | |
| Post-conditions: | Water added successfully | |
| Flow of e-vents: | **User Action** | **System Action** |
| 1- User enters amount of water drank. |  |
|  | 2- System adds water. |
|  |  | 3-Verify addition of water. |
| Exceptions: | **User Action** | **System Action** |
| NULL |  |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Add Pill | |
| Actors: | User | |
| Pre-conditions: | User is signed in. | |
| Post-conditions: | Pill name and time is added successfully. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-User enters pill name and time. |  |
|  |  | 2-System checks database for name. |
|  |  | 2-System doesn’t find pill name. |
|  |  | 4-Verify addition. |
| Exceptions: | **User Action** | **System Action** |
| 1-User enters pill name and time. |  |
|  | 2-Pill name already exists. |
|  |  | 3-System checks status. |
|  |  | 4-If turned on; print “Pill exists and turned on, please enter another” |
|  | 5-User enters another pill name and time. |  |
|  |  | 6- repeat 2 and 3 |
|  |  | 7-if turned off; print “Pill exists but turned off, please turned it on or enter another.” |
|  | 8-Users turns it on or enter another pill name and time. |  |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Calculate calories | |
| Actors: | User | |
| Pre-conditions: | Type of food exists. | |
| Post-conditions: | System calculate the calories gained. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters the meal name. |  |
|  | 2- System checks the database. |
|  |  | 3-System calculate calories. |
| Exceptions: | **User Action** | **System Action** |
|  | 1-User enters the meal name. |  |
|  |  | 2-System didn’t find the food in the database. |
|  |  | 3-asks the user to enter the food. |
|  | 4-User adds food. |  |
|  |  | 5-system calculate calories. |
| Includes: | Add Food | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Add food | |
| Actors: | User | |
| Pre-conditions: | User has signed in. | |
| Post-conditions: | Meal name is added. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User adds food name and calories. |  |
|  | 2- System adds data entered to the database. |
| Exceptions: | **User Action** | **System Action** |
|  | 1-User adds food name. |  |
|  |  | 2-System checks database |
|  |  | 3-Food name exists |
|  |  | 4-Asks the user to add another name. |
|  | 5-User adds another name. |  |
|  |  | 6-Food name is added. |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Health card | |
| Actors: | User | |
| Pre-conditions: | User is signed in. | |
| Post-conditions: | Shows health card information. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User chooses type of information needed. |  |
|  | 2-System displays information chosen. |
| Exceptions: | **User Action** | **System Action** |
|  |  | 1-Health card is empty. |
|  |  | 2-System displays a blank page. |
| Includes: | Add Health card | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Add Checkup | |
| Actors: | User | |
| Pre-conditions: | User has signed in. | |
| Post-conditions: | Checkup date and time is added. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User adds checkup date and time. |  |
|  |  | 2-System verifies. |
| Exceptions: | **User Action** | **System Action** |
|  | 1-User adds checkup date and time. |  |
|  |  | 2-Date or time has already passed. |
|  |  | 3-asks user to try again. |
|  | 4-User enters data again |  |
|  |  | 5-System verifies. |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 13 | |
| Use Case Name: | Add Health Card | |
| Actors: | User | |
| Pre-conditions: | Blood type, Medical prescriptions, vaccinations taken, weight, height. | |
| Post-conditions: | Health card information is filled. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User adds any type of data related to his medical history. |  |
|  | 2- System adds data into the database. |
| Exceptions: | **User Action** | **System Action** |
| NULL |  |
|  | NULL |  |
| Includes: | NULL | |
| Notes and Issues: | NULL | |

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |