**VAQ Healthy Living Guide**

**Software Design Documentation**

Version 1.1

November 15, 2017

**Lead Software Engineer:**

Stephen Izquierdo

**Project Team:**

Gabriel Garcia

Dulio Gomez

Edward Jones

Reyna Maximino

Yessenia Rodriguez

Jacob Shircliff

Edson Trejo Hernandez

Prepared for

Software Engineering

Instructor: MK Quweider, Ph.D.

Fall 2017

**Table of Contents**

**1. Introduction 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

**2. References 2**

**3. Detailed Design 2**

3.1 Module Detailed Design 2

*3.1.1 LoginController.java*

*3.1.2 HomeController.java*

*3.1.3 PersonalController.java*

*3.1.4 MedicalController.java*

*3.1.5 ExerciseProfileController.java*

*3.1.6 RoutineCreatorController.java*

*3.1.7 ExportController.java*

*2*

3.2 Data Detailed Design 2

*3.2.1 TabManager.java 2*

*3.2.2 Medical.java*

*3.2.3 Allergy.java*

*3.2.4 Medical.java*

*3.2.5 Personal.java*

*3.2.6 Exercise.java*

*3.2.7 ExerciseRoutine.java*

*3.2.8 WeeklyRoutine.java*

*3.2.9 SaveProfile.java*

*2*

**4. User Interfaces By Use Case 2**

4.1 Create new Account 2

4.2 Login to Vaq\_Health

4.3 Update Personal

4.4 Update Medical

4.5 Update Exercise

4.6 Create Weekly Routine

4.7 Export Profile 2

**A. Appendices 2**

A.1 Appendix 1 2

A.2 Appendix 2 3

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Design Document is to provide an explicit outline of the software architecture and high-level design of the VAQ Healthy Living Guide application. Various data structures, databases and interfaces will be depicted to accomplish this goal. This is being done with the intention of providing developers an insight to how our Client’s needs are being meet efficiently and effectively regarding the requirements set in the SRS. This document’s intended audience will be developers, current company, and future organizations.

## 1.2 Scope

The software application being described in this document is the VAQ Healthy Living Guide application. This SDD will provide a proof of concept for building a prototype that demonstrates the functions specified in its respective SRS. Using use-case, state, class, and data flow models, this document will explain the developing team’s thought process, achieving the overall purpose of this SDD.

## 1.3 Definitions, Acronyms, and Abbreviations

The following terms, acronyms, and abbreviations are used throughout this document and are presented in the table below.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SDD | Software Design Document |
| SRS | Software Requirement Specification |
| GUI | Graphical User Interface |
| Database | A structured set of data held in a computer, especially one that is accessible in various ways. |
| MySQL | My Structured Query Language: open source relational database management system. |
| XML | Extensible Markup Language |
| XSL | Extensible Stylesheet Language |
| HTML | HyperText Markup Language |
| PDF | Portable Document Format |
| GitHub | GitHub is a web-based Git or version control repository and Internet hosting service, used by software developing teams. |
| Java Virtual Machine | Allows a java program to run on any platform, regardless of the operating system, without having to modify the code. |
| Wizard | A sequential set of prompts used in data collection, organized for ease of use for the user. |

# 2. References

GitHub – <https://github.com/>

MySQL – <https://www.mysql.com/>

JavaFX – <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

NetBeans – <https://netbeans.org/>

Gluon - <http://gluonhq.com>

# 3. Detailed Design

## 3.1 Module Detailed Design

**3.1.1**

Class: LoginContoller.java

|  |  |
| --- | --- |
| **LoginController** | |
| Manages the GUI components for the login tab | |
| **Attributes** | **Description** |
| TextField usernameTF | username input |
| TextField passwordTF | password input |
| Label errorL | displays errors to user |
| String password | saves current password entered |
| **Methods** | **Description** |
| initialize() | initilize fields |
| OpenCreateAccount() | opens create account tab |
| OpenHome() | opens home tab |
| Login() | Logs into account and opens home tab |
| RetreiveProfile() | gets profile from database |

**3.1.2**

Class: HomeContoller.java

|  |  |
| --- | --- |
| **Home Controller** | |
| Manages the GUI components for the home tab | |
| **Methods** | **Description** |
| OpenExercise() | opens exercise tab |
| OpenRoutine() | opens routine tab |
| Logout() | signs out and opens login tab |

### 

**3.1.3**

Class: PersonalController.java

|  |  |
| --- | --- |
| **PersonalController** | |
| Manages the GUI components for the profile personal tab | |
| **Attributes** | **Description** |
| ArrayList<String> states | list of the states in the U.S. |
| TextField fNameTF | first name input |
| TextField lNameTF | last name input |
| TextField zipTF | zipcode input |
| TextField addressTF | address input |
| TexField cityTF | city input |
| TextField emailTF | email input |
| TextField birthdayMonthTF | birthday month input |
| TextField birthdayDayTF | birthday day input |
| TextField birthdayYearTF | birthday year input |
| ComboBox sexCB | sex input |
| ComboBox stateCB | state input |
| ImageView profielIV | displays profile image |
| ImageFile imageFile | stores uploaded image file |
| **Methods** | **Description** |
| initialize() | initilize fields and calls DisplayPersonal() |
| OpenMedical() | opens profile medical tab |
| OpenHome() | opens home tab |
| OpenExercise() | opens profile exercise tab |
| OpenExport() | opens export tab |
| SavePersonal() | saves personal information from GUI fields to user’s profile |
| DisplayPersonal() | displays user personal information to the GUI |
| UploadProfilePic() | allows user to upload profile pic |

### 

**3.1.4**

**Class:**MedicalController.java

|  |  |
| --- | --- |
| **MedicalController** | |
| Manages the GUI components for the medicalpersonal tab | |
| **Attributes** | **Description** |
| TextField weighTF | input weight |
| TextField heightTF | height input |
| CheckListView allergyCLV | allergy check box list |
| CheckListView diseaseCLV | disease check box list |
| TextArea diseaseTA | displays information about disease |
| TextArea allergyTA | displays information about allergy |
| ArrayList<Disease> diseaseList | disease list to store user diseases |
| ArrayList<Allergy> allergyList | disease list to store user allergies |
| **Methods** | **Description** |
| initialize() | initilize fields and calls DisplayMedical() |
| OpenMedical() | opens profile medical tab |
| DisplayMedicalProfile() | displays user medical information to the GUI |
| OpenPersonal() | opens profile personal tab |
| OpenExercise() | opens profile exercise tab |
| OpenHome() | opens home tab |
| SaveMedical() | saves medical information from GUI fields to user’s profile |

**3.1.5**

**Class:**ExerciseProfileController.java

|  |  |
| --- | --- |
| **ExerciseProfileController** | |
| Manages the GUI components for the Exercise Profile tab | |
| **Attributes** | **Description** |
| CheckListView equipmentCCB | first name |
| ImageView equipmentIV | last name |
| ArrayList<Equipment> equipmentList | sex |
| **Methods** | **Description** |
| DisplayExerciseProfile() | state (U.S.) |
| Save() | saves the checked off equipment to a list |
| inititalize() | adds equipment list to GUI and calls DisplayExerciseProfile() |

### 

**3.1.6**

Class:RoutineCreatorController.java

|  |  |
| --- | --- |
| **RoutineCreatorController** | |
| Manages the GUI components for the Routine Creator tab | |
| **Attributes** | **Description** |
| ComboBox routineCB | list of routines to use as templates |
| ListView exerciseLV | list of exercises to choose from |
| ListView mondayRoutineLV | list of exercises for monday |
| ListView tuesdayRoutineLV | list of exercises for tuesday |
| ListView wednesdayRoutineLV | list of exercises for wednesday |
| ListView thursdayRoutineLV | list of exercises for thursday |
| ListView fridayRoutineLV | list of exercises for friday |
| ListView saturdayRoutineLV | list of exercises for saturday |
| ListView sundayRoutineLV | list of exercises for sunday |
| TextArea routineNameTA | input for name of the routine being creator |
| TabPane weekTabPane | holds the week day tabs |
| ListView currentDayLV | used to save the access to the current day list view |
| ArrayList<exercise> exerciseList | list of exercises |
| String currentDay | used to keep track of the current tab selected |
| **Methods** | **Description** |
| AddExercise() | adds the selected exercise to the current day list view |
| RemoveExercise() | removes the selected exercise in the current day list view |
| CreateRoutine() | creates a weekly routine and saves to user profile |
| DisplayRoutine() | displays the selected routine selected from the routineCB |
| wdTabChanged() | Tab event handler |

### 

**3.2.7**

Class:ExportController.java

|  |  |
| --- | --- |
| **ExportController** | |
| Manages the GUI components for the Export tab | |
| **Attributes** | **Description** |
| Button exportButton | export button |
| AnchorPane pdfBackgroundAP | used to highlight pdf icon when selected |
| AnchorPane htmlBackgroundAP | used to highlight html icon when selected |
| AnchorPane xmlBackgroundAP | used to highlight xml icon when selected |
| TextArea folderTA | input directory to save files |
| boolean pdfClicked | used to keep track if pdf has been clicked |
| boolean htmlClicked | used to keep track if html has been clicked |
| boolean xmlClicked | used to keep track if xml has been clicked |
| **Methods** | **Description** |
| ClickPDF() | pdf button event handler |
| ClickHTML() | html button event handler |
| ClickXML() | xml button event handler |
| SelectFolder() | event handler for selecting directory |
| HighLight(boolean highlight,Anchor pane) | highlights the passed in anchor pane |
| Export() | saveProfile event handler |

### 

.

**3.2.1**

Class: TabManager.java

|  |  |
| --- | --- |
| **TabManager** | |
| The Tab Manager is used to navigate through the application. | |
| **Attributes** | **Description** |
| instance | singleton |
| getInstance() | returns the current instance |
| OpenHome() | Opens home tab |
| OpenProfilePersonal() | Opens profile personal tab |
| OpenProfileMedical() | Opens profile medical tab |
| OpenProfileExercise() | Opens profile exercise tab |
| OpenProfileExport() | Opens profile export tab |
| OpenRoutineHome() | Opens routine home tab |
| OpenRoutineCreator() | Opens routine creator tab |
| OpenRoutineManger() | Opens routine manager tab |
| OpenExercises() | Opens exercies tab |

**3.2.2**

Class:Medical.java

|  |  |
| --- | --- |
| **Medical** | |
| Holds the user medical information | |
| **Attributes** | **Description** |
| int weight | weight |
| int height | height |
| ArrayList<Disease> diseaseList | disease list to store user diseases |
| ArrayList<Allergy> allergyList | disease list to store user allergies |

### 

**3.2.3**

Class:Allergy.java

|  |  |
| --- | --- |
| **Allergy** | |
| Holds Allergy information | |
| **Attributes** | **Description** |
| String name | name of the allergy |
| String description | description of the allergy |

**3.2.4**

**Class:**Disease.java

|  |  |
| --- | --- |
| **Disease** | |
| Holds Disease information | |
| **Attributes** | **Description** |
| String name | name of the disease |
| String description | description of the disease |

**3.2.5**

Class:Personal.java

|  |  |
| --- | --- |
| **Personal** | |
| Holds users personal information | |
| **Attributes** | **Description** |
| String fName | first name |
| String lName | last name |
| SexEnum sex | sex |
| String address | street address |
| String state | state (U.S.) |
| String city | city |
| String zipCode | zip code |
| String email | email address |
| Date birthday | birthday |
| **Methods** | **Description** |
| getFname() | returns first name |
| getlName() | returns last name |
| getSex() | returns sex |
| getAddress() | returns address |
| getState() | returns state |
| getCity() | returns city |
| getZipCode() | returns zip code |
| getEmail() | returns email |
| getBirthday() | returns brithday |

### 

**3.2.6**

Class:Exercise.java

|  |  |
| --- | --- |
| **Exercise** | |
| Exercise Model | |
| **Attributes** | **Description** |
| String name | name of the exercise |
| String description | description of the exercise |
| ExerciseDifficultyE difficulty | difficulty of the exercise |
| Image image | image of the exercise |
| Equipment equipment | required equipment |
| int met | metabolic equivalent task |
| **Methods** | **Description** |
| getName() | state (U.S.) |
| getDescription() | saves the checked off equipment to a list |
| getDifficulty() | adds equipment list to GUI and calls DisplayExerciseProfile() |
| getImage() |  |
| getEquipment() |  |
| getMet() |  |

### 

**3.2.7**

Class:ExerciseRoutine.java

|  |  |
| --- | --- |
| **ExerciseRoutine** | |
| Exercise Routine Model | |
| **Attributes** | **Description** |
| Exercise exercise | holds the exercise |
| Double workOutTime | work out time |
| **Methods** | **Description** |
| ExerciseRoutine(Exercise exercise ,Double workOutTime) | constructor |

### 

**3.2.8**

Class:WeeklyRoutine.java

|  |  |
| --- | --- |
| **WeeklyRoutine** | |
| Weekly Routine Model | |
| **Attributes** | **Description** |
| String name | name of the weekly routine |
| ArrayList<ExerciseRoutine> monday | monday exercise routines |
| ArrayList<ExerciseRoutine> tuesday | tuesday exercise routines |
| ArrayList<ExerciseRoutine> wednesday | wednesday exercise routines |
| ArrayList<ExerciseRoutine> thursday | thursday exercise routines |
| ArrayList<ExerciseRoutine> friday | friday exercise routines |
| ArrayList<ExerciseRoutine> saturday | saturday exercise routines |
| ArrayList<ExerciseRoutine> sunday | sunday exercise routines |
| **Methods** | **Description** |
| toString() | returns the name of the routine |

### 

**3.2.9**

Class:SaveProfile.java

|  |  |
| --- | --- |
| **Save Profile** | |
| Converts user profile to a html/xml/pdf file | |
| **Methods** | **Description** |
| Converter(String directory) | Converts user profile to a html/xml/pdf file |
| HTML(String directory) | Converts user profile to a html file |
| PDF() | Converts user profile to a pdf file |

### 

**4. User Interfaces by Use Case Scenarios**

**4.1 Create New Account**

Vaq\_Health will require users to create an account.

**A Valid Scenario**

1. User runs application

2. User clicks create account link

3. User Fills out username and password

4. User logs into Vaq\_Health

**An invalid Scenario 1**

1. User runs application

2. User clicks create account link

3. User enters a username that already exists

4. Vaq\_Health prompts error “Username Already Exists”

**An invalid Scenario 2**

1. User runs application

2. User clicks create account link

3. User enters a password that doesn’t meet requirements

4. Vaq\_Health prompts error “Password Doesn't Meet Requirements (Requirements..)”

**4.2 Login to Vaq\_Health**

Vaq\_Health will allow users that have created an account to log in.

**Case Scenario**

1. User in login window

2. User successfully inputs username and password

3. User logs into Vaq\_Health

**4.3 Update Personal**

Vaq\_Health will allow users update the profile which is saved into a database.

**Valid Scenario**

1. Starting at home screen user clicks profile button.

2. User clicks personal button

3. User fills out personal information

4. User clicks save button

5. Vaq\_Healthupdates the profile which is saved into a database

**4.4 Update Medical**

Vaq\_Health will allow users update the medical information in the user's profile which is saved into a database.

**Valid Scenario**

1. Starting at home screen user clicks profile button.

2. User clicks medical button

3. User checks off which allergies and diseases they have

4. User clicks save button

5. Vaq\_Healthupdates the profile which is saved into a database

**4.4 Update Exercise**

Vaq\_Health will allow users update the exercise information in the user's profile which is saved into a database.

**Valid Scenario**

1. Starting at home screen user clicks profile button.

2. User clicks exercise button

3. User checks which exercise equipment they have

4. User clicks save button

5. Vaq\_Healthupdates the profile which is saved into a database

**4.5 Create Weekly Routine**

Vaq\_Health will allow users to create weekly routines that will allow users manage their exercise schedule

.

**Valid Scenario**

1. Starting at home screen user clicks Routine button.

2. User clicks Create button

3. User correctly fills out routine information

4. User clicks save button

5. Vaq\_Health saves the routine to the users weekly routine list and updates the database.

**InValid Scenario**

1. Starting at home screen user clicks Routine button.

2. User clicks create button

3. User fills out routine information but leaves out the name

4.User clicks save button

5. Vaq\_Health prompts error “Routine Name Has Not Been Entered”

**4.6 Export Profile**

Vaq\_Health will allow users to export their profile information to a pdf,xml and html file to a chosen directory.

.

**Valid Scenario**

1. Starting at home screen user clicks profile button.

2. User clicks export button

3. User selects which formats they want to export(pdf,html,xml)

4. User clicks “select directory” label and successfully chooses a directory

5. User clicks export button

6. Vaq\_Health exports the chosen formats to the selected directory

**Invalid Scenario**

1. Starting at home screen user clicks profile button.

2. User clicks export button

3. User selects which formats they want to export(pdf,html,xml)

4. User clicks “select directory” label and successfully chooses a directory

5. User clicks export button

6. Vaq\_Health exports the chosen formats to the selected directory

**InValid Scenario**

1. Starting at home screen user clicks Routine button.

2. User clicks create button

3. User fills out routine information but leaves out the name

4.User clicks save button

5. Vaq\_Health prompts error “Routine Name Has Not Been Entered”

# 

# A. Appendices

## A.1 Appendix 1

Required form for approval of changes to this document.

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Izquierdo Stephen | Lead Software Eng. |  |
|  |  |  |  |
|  |  |  |  |

## A.2 Appendix 2

Required form needed if there are any changes after the document is completed.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | | **Author** | **Comments** |
| 11/2/2017 | Version 1.0 | Yessenia Rodriguez, Stephen Izquierdo | | First Draft |
| 11/27/2017 | Version 1.1 | Stephen Izquierdo | | Second Draft |
|  |  |  | |  |
|  |  |  | |  |