­

Software Requirements Specification

for

Glucometer App

Version 0.8.5.2

Prepared by Wong Chu Feng

Trilogy Technologies

13 July 2023

Table of Contents

[**1. Introduction**](#_1fob9te) **Error! Bookmark not defined.**

[1.1 Purpose](#_3znysh7) **Error! Bookmark not defined.**

[1.2 Document Conventions](#_2et92p0) **Error! Bookmark not defined.**

[1.3 Intended Audience and Reading Suggestions](#_tyjcwt) **Error! Bookmark not defined.**

[1.4 Product Scope](#_3dy6vkm) **Error! Bookmark not defined.**

[**2. Overall Description**](#_1t3h5sf) **Error! Bookmark not defined.**

[2.1 Product Perspective](#_4d34og8) **Error! Bookmark not defined.**

[2.2 Product Functions](#_2s8eyo1) **Error! Bookmark not defined.**

[2.3 User Classes and Characteristics](#_17dp8vu) **Error! Bookmark not defined.**

[2.4 Operating Environment](#_3rdcrjn) **Error! Bookmark not defined.**

[2.5 User Documentation](#_lnxbz9) **Error! Bookmark not defined.**

[2.6 Assumptions and Dependencies](#_35nkun2) **Error! Bookmark not defined.**

**3. External Interface Requirements Error! No bookmark name given.**

[3.1 User Interfaces](#_44sinio) **Error! Bookmark not defined.**

[3.2 Hardware Interfaces](#_2jxsxqh) **Error! Bookmark not defined.**

[3.3 Software Interfaces](#_z337ya) **Error! Bookmark not defined.**

[**4. System Features**](#_3j2qqm3) **Error! Bookmark not defined.**

[4.1 Data sets](#_qqcq74xd5eh2) **Error! Bookmark not defined.**

[4.2 Sign up and login page](#_js3t2oam3zpe) **Error! Bookmark not defined.**

[4.3 Personality Test page](#_xs9udbqkic31) **Error! Bookmark not defined.**

[4.4 Input page](#_9t4tmtvt2x7d) **Error! Bookmark not defined.**

[4.5 Recommendations page](#_s7nm01xhayjj) **Error! Bookmark not defined.**

[4.6 User preferences](#_dhnmcdqg6tq9) **Error! Bookmark not defined.**

[4.7 Course and College Introduction Page](#_n2cy9xyae8pp) **Error! Bookmark not defined.**

[**5. Other Nonfunctional Requirements**](#_2xcytpi) **Error! Bookmark not defined.**

[5.1 Performance Requirements](#_1ci93xb) **Error! Bookmark not defined.**

[5.2 Usability Requirements](#_3whwml4) **Error! Bookmark not defined.**

5.3 Reliability Requirements **Error! No bookmark name given.**

[5.4 Security Requirements](#_2bn6wsx) **Error! Bookmark not defined.**

5.5 Maintainability Requirements **Error! No bookmark name given.**

[5.6 Software Quality Attributes](#_qsh70q) **Error! Bookmark not defined.**

**6. Use case descriptions Error! No bookmark name given.**

[**7. Diagrams**](#_b3ctkcdyopuv) **Error! Bookmark not defined.**

[7.1 Initial Use Case diagram](#_ryusi7ffq89t) **Error! Bookmark not defined.**

[7.2 Dialog map](#_nc049jpofbur) **Error! Bookmark not defined.**

[7.3 System architecture](#_r7fba2xnbimo) **Error! Bookmark not defined.**

[7.4 Class diagram](#_wexf45gsb1i7) **Error! Bookmark not defined.**

[7.5 Sequence diagrams](#_qgey3lbnqemd) **Error! Bookmark not defined.**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Description** | **Version** |
| App connection with glucometer | 10 May 2023 | At this point, the app has the following functionalities:   1. Splash screen 2. Establishes BLE connection to glucometer 3. Receives and displays biodata. | 0.5 |
| Add Personal Information page and bug fix | 22 May 2023 | Added a page for users to key in their personal information which includes their name, age, gender, height, and weight. Fixed a bug that prevented the saving of entered personal information. | 0.5.1 |
| Export and Storage options | 23 May 2023 | Users can now choose to export and save biodata as .xls or .csv into their mobile phone storage. Fixed an issue where exported file could not be saved. | 0.6 |
| Add email function | 23 May 2023 | Users can choose to send the saved data via email. | 0.6.1 |
| Basic graph display | 25 May 2023 | Added the most basic graph that reads data from the local database and plots a graph to display all the data. | 0.7 |
| Graph package update | 26 Mar 2023 | Updated graph package from charts\_flutter to syncfusion\_flutter\_charts due to deprecation. | 0.8 |
| System compatibility | 1 June 2023 | Migrated app from Android 8.0 to Andorid 10.0 including necessary dependency updates. | 0.8.0.1 |
| Graph display and data format fixes | 8 June 2023 | Fixed DateTime format to ensure proper graph display. Resolved a problem with time format when inserting data into the database | 0.8.1 |
| iOS Development | 12 June 2023 | Built iOS development app using Xcode 13. | 0.8.2 |
| Dashboard enhancement | 15 June 2023 | Created a new, visually appealing dashboard to display biodata. | 0.8.3 |
| BLE compatibility improvement | 22 June 2023 | Fixed problem where newer devices cannot detect BLE | 0.8.4 |
| Data filtering and graph visualisation | 28 June 2023 | Users are now able to filter the data and view their data on the graph by the day, week, month, or year. | 0.8.5 |
| Refactor code | 29 June 2023 | This version change focuses on code refactoring, primarily to improve readability of the codebase. This reorganization streamlines the development process, increase code efficiency and ensures a more robust foundation for future enhancements to any component in the app. | 0.8.5.1 |
| Graph colour highlighting | 30 June 2023 | Graph now visually distinguishes glucose levels above the threshold. | 0.8.5.2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Glucometer App or Blood-Glucose monitoring app serves as a comprehensive digital companion, enabling users to record and track essential health metrics such as heart rate, cholesterol level, uric acid level, glucose level and SpO2 level. With user-friendly features like data storage and data sharing with healthcare professionals and visual trend analysis through graphs, this app paves the way to revolutionise and elevate the current state of teleconsultation and remote health monitoring between patients and doctors. By providing a convenient and efficient platform for tracking and sharing essential health metrics, it bridges the gap between individuals and healthcare providers, enabling more informed and personalised remote consultations.

## Document Conventions

* Glucometer
  + A blood glucose meter is a (medical) device used to measure the concentration of glucose in a person’s blood. It is commonly used by diabetic patients to monitor their glucose levels.
  + Specific to Trilogy Technologies’ glucometer, it can measure not only glucose levels but also uric acid, heartrate, SpO2 and cholesterol levels.
  + The Arduino code defines the healthy range to be <140mg/dL or <7.8mmol/L.
* SpO2
  + The measurement of oxygen saturation in a person’s blood. It generally should range from 96% - 99%.
* Cholesterol
  + The fatty substance found in our bodies which can cause health issues in excess.
  + The Arduino code defines the healthy range to be <200mg/dL or <6.6mmol/L.
* Uric Acid
  + A waste product produced by the body’s metabolism of certain foods. Elevated uric acid levels can lead to health issues.
  + The Arduino code defines a healthy range for both men and women.
  + Range for men: 4.0 – 8.5mg/dL or 0.24 – 0.51mmol/L.
  + Range for women: 2.7 – 7.3mg/dL or 0.16 – 0.43mmol/L.
* MAX30100 PulseOximeter
  + Heart rate sensor module that measures heart rate.
* Adafruit SSD1306
  + OLED graphic display.
* Bluetooth Low Energy (BLE)
  + A wireless communication technology designed for energy-efficient data exchange between devices. More commonly used in devices that transmits data over short distances.
  + Unlike traditional Bluetooth, BLE is more optimised for low-power, less data-intensive applications where energy efficiency is crucial, making it ideal for devices that require intermittent data exchange or have limited power resources.

This SRS document is adapted from the IEEE standard for Software Requirements Specification.

## Intended Audience and Reading Suggestions

The software requirements specification (SRS) document serves as a blueprint for developers and testers.  
  
Developers can understand the purpose of this project and find any necessary information about the requirements or functionality of the app through this document. A clear description of the development process will be described and made easy for any to follow along.

Extending from this understanding to testers, they are able to identify the test cases that need to be executed to ensure the website is able to carry out its desired function. Test scenarios and test cases are to help validate the functionality and performance of the website.

## Product Scope

Glucometer App is a user-friendly free-of-charge application that targets diabetic patients or those at risk and encompasses a range of features designed to empower users in monitoring and managing their glucose levels and aiding their healthcare providers. This app serves as a digital health companion, enabling users to effortlessly view their glucose levels. With a user-friendly interface, users can securely store their data or conveniently email it to their healthcare provider for analysis and consultation, all done within the app. The app also incorporates an interactive graph, allowing users to visually track and analyse their glucose level over time, fostering a better understanding of their health status and facilitating informed decision-making regarding lifestyle choices or treatment plans. However, this app will not provide medical advice, nor will it replace the guidance of a healthcare professional.

# Overall Description

## Product Perspective

The Glucometer app fits into a broader healthcare ecosystem, playing a pivotal role in empowering individuals to manage their health effectively. Within this perspective, the app interacts with various entities, including Trilogy Technologies’ glucometers for data recording and input, user devices such as smartphones for data storage and communication channels for data sharing with healthcare professionals. Furthermore, it aligns with prevailing privacy regulations and data protection measures to safeguard sensitive health information. By considering the app’s place within a larger ecosystem, it ensures a seamless user-centric approach to health management.

## Product Functions

* Website must utilise Graduate Employment Survey conducted by NTU, NUS, SMU, SIT, SUTD, SUSS.
* Website shall refer to the indicative grade profiles across all public universities including NTU, NUS, SMU, SIT, SUTD, SUSS.
* Users must be able to create an account with their Google account or a valid email address.
* Users must be able to login with their registered email address and password.
* The test page must prompt users to take the RIASEC personality test.
* User’s personality test scores must be stored in the database.
* The website must feature a page for users to input their academic qualifications and interest subjects.
* The website must support at least 3 curriculums recognised by Singapore’s public universities.
* The website must prompt users to input their subjects taken and respective grades obtained.
* User’s input must be stored in the database.
* User’s input must be checked against the indicative grade profiles stored in the database to optimise user recommendations.
* The Recommendations Page must feature 3 course suggestions, each at exactly one university.
* Users must be able to view details by selecting recommended course and college.
* Website must suggest different courses and colleges to complement user preference.

## User Classes and Characteristics

We will have one user class that will be able to interact with all features of FindMyCollege including the APIs.

## Operating Environment

FindMyCollege will be a website that supports all browsers on any operating system such as Windows or MacOS. It will feature external APIs such as Google Charts to plot graphs.

## User Documentation

A tutorial video alongside this SRS documentation will be uploaded so prospective users are able to understand how to use the website.

## Assumptions and Dependencies

* Users must be connected to Wi-Fi or mobile data
* Users must use an internet browser that supports the FindMyCollege website
* The website will only be in English
* The website is primarily designed for desktop usage

# External Interface Requirements

## User Interfaces

FindMyCollege is a website built specifically for pre-university students who are unsure of what course they should enroll for university. The website features a login and password verification, where the system only accepts a valid password and checks through the database to ensure each user has only a unique account.

Users can also interact with input fields on the website for their login details and personal information such as their academic qualifications and to select their choices during the personality test.

Graphical user interface, website

Description automatically generated

Users will first see the main page (1) upon entering the website. If the user selects “login” (2), they will be prompted to enter their email and password. Otherwise, users can sign up as a new user (3) if they “don’t have an account”. Users will be prompted to input a username, a valid email address and password. Users can modify their “account information” (4) afterwards.

Graphical user interface, text, application

Description automatically generated

Upon taking the test, users are prompted to take a personality test (1). Afterwards, users will key in their academic information (2). Users can adjust their preferences and key in their personality test results. Based on user’s preferences (3), FindMyCollege will recommend a set of different courses at different universities. Details can be found by selecting a particular course (4).

## Hardware Interfaces

The hardware interface depends solely on their device the user is using to access the website. If the user is on a computer, a keyboard, mouse and monitor screen with internet access via Wi-Fi or Ethernet connection is necessary. If on a mobile device, the device also has to have internet access via Wi-Fi or mobile data and more commonly, a multi-touch capacitive screen.

## Software Interfaces

The website is based on HTML 5 built using JavaScript. (ECMAScript 2020), Flask (2.2.3), reliant on Python (3.11.2), CSS (2.1) for the frontend designs and SQLAlchemy (2.0.5) for our database management. It also includes an API from Google Charts to plot a graph of salary changes over time.

# System Features

## Data sets

4.1.1 FindMyCollege must utilise the Graduate Employment survey conducted by NTU, NUS, SMU, SIT, SUTD, SUSS retrieved from data.gov.sg

4.1.2 FindMyCollege shall refer to the indicative grade profiles across all public universities including NTU, NUS, SMU, SIT, SUTD, SUSS.

## Sign up and login page

4.2.1 Users must be able to create an account with their Google account or a valid email address

4.2.2 Users must be able to login with their registered email address and password.

## Personality Test page

4.3.1 The test page must prompt users to take the RIASEC personality test

4.3.2 Users’ personality test scores must be stored in the database

## Input page

4.4.1 The website must feature a page for users to input their academic qualifications and interest subjects

4.4.2 The website must support at least 3 curriculums recognised by Singapore’s public universities.

4.4.3 The website must prompt users to input their subjects taken and respective grades obtained.

4.4.4 Users’ inputs must be stored in the database

4.4.5 Users’ inputs must be checked against the IGP stored in the database to optimise the recommendations to users.

## Recommendations page

4.5.1 The Recommendations page must feature at least 3 course suggestions, each at exactly one university.

4.5.2 Users must be able to view details of a recommended course and college by clicking on a link.

## User preferences

4.6.1 Users must be able to modify their preferences by re-taking the personality test.

4.6.2 The website must suggest different courses and colleges to complement user preferences.

## Course and College Introduction Page

4.7.1 The website must display course ranking and description, employment outcomes and campus location.

4.7.2 The website must implement Google Charts API to display campus location.

# Other Nonfunctional Requirements

## Performance Requirements

5.1.1. The system must not crash when the user opens the application.

5.1.2. The system must be able to display recommendations within 3 seconds of the user submitting the personality test.

## Usability Requirements

5.2.1. The website design must be intuitively illustrated for ease of navigation.

5.2.2. All features of the website are to be clearly displayed.

5.2.3. The website must offer informative recommendations.

5.2.4. The system must provide necessary feedback to the user when invalid inputs are detected.

5.2.5. The system must display appropriate error messages with certain processes fail.

5.2.6. The system will only support the English language.

## Reliability Requirements

5.3.1. The website must have an uptime of more than 98%.

5.3.2. The website must not break due to user’s invalid input.

## Security Requirements

5.4.1. User data should not be disclosed without their consent.

5.4.1.1. User’s data must not be seen or accessible by other users.

5.4.1.2. User data must be deleted from the database when the user deletes the account.

5.4.2. The password must be at least 12 characters long and a combination of uppercase letters, lowercase letters, numbers and symbols.

5.4.3. Users must be automatically logged out if there is a change in account information.

## Maintainability Requirements

5.5.1. Maintenance shall be conducted regularly every month to ensure website is up to date.

5.5.2. An electronic announcement must be released on the website as a pop up message to inform the users, days prior to maintenance, the date and details of the maintenance accurately.

## Software Quality Attributes

FindMyCollege adopts a 3-Tier Architecture design, namely:

|  |  |  |
| --- | --- | --- |
| **Presentation Layer** | **Application Layer** | **Data Layer** |
| 1. Dart | 1. C++ | 1. SQLite |

This tiered architecture means that each layer is independent of the other layers. Hence, each layer can be modified or scaled without affecting the system’s performance, therefore also promotes modularity and increases separation of concerns. Furthermore, having the data layer separated from the other layers adds security to the system, making it difficult for any user to gain unwanted access to the database with everyone’s private information.

We also adhered to good software engineering principles of

* 1. Good design
     1. Reusability of features
        + Google Charts API is reusable to make any graphs the programmer needs.
        + Methods for user account creation, the authentication process and storing of user data in a database can be reused for another project.
     2. Testability of features
        + Each feature can be tested independently.
        + Features like signup, personality test page, storing of data in the database can all be tested separately.
     3. Maintainability of features
        + Ensured readability of code
        + Included comments to help anyone else reading the source code understand what each function does.
     4. Extensibility of features
        + The flexible architecture allows new modules or components to be added without affecting the existing system.
  2. Design Patterns
     1. Single responsibility principle
        + Our application is broken down into smaller, individual functions, each with a single responsibility.
        + Improves reusability as each function can be reused in other modules.
        + Improves maintainability as each function is less likely to affect other functions.
  3. Design Principles
     1. High cohesion
        + The modules in the system have a single, well-defined responsibility.
     2. Low coupling
        + Changes in one module will not affect other functions.
        + Eg. In take\_test and Login modules, the former facilitates displaying test questions and storing of user data while the latter facilitates user authentication. These modules do not have overlapping responsibilities and does not affect each other at all.
     3. Open-Closed
        + Existing code allows user data (email, password, website-specific data) to be used simultaneously.
        + Eg. The code needs no further modification when trying to get any user data. (Closedness)
        + Extension of existing classes is possible.
        + Eg. Website-specific data can be modified to store whatever the programmer requires. (Openedness)
     4. Separation of concerns
        + The application consists of independent modular component, having adopted a 3-tier architecture.
        + The code is more readable since all components do not have overlapping functionalities.

Overall, a tiered architecture and good software design principles in FindMyCollege provides benefits like scalability, modularity, maintainability, reusability and security.

# Use case descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: |  | | |
| Use Case Name: | Login | | |
| Created by: | Hazel | Last Updated by: | Hazel |
| Date created: | 29 Jan 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor | User |
| Description | Users can log into an account on the website application with a unique username and password. |
| Preconditions | 1. Device must be connected to Wi-Fi or Mobile Data |
| Postconditions | 1. Users will be able to login with their email address and password. 2. Users will be able to take the RIASEC test and logout |
| Priority | High |
| Frequency of Use | Medium |
| Flow of events | 1. System requests input for email and password 2. User enters email address and password 3. User clicks “Log In” button 4. Information is validated against database 5. User logs in successfully |
| Alternative flow: | **AF-S1: Email Address is invalid**   1. User is presented with an error message 2. User is prompted to enter their username again 3. Return to step 1   **AF-S2: Password is incorrect**   1. User is presented with an error message 2. User is prompted to enter their password again 3. Return to step 2 |
| Exceptions: | - |
| Includes | * Verify Account (UC02) |
| Special Requirements: | - |
| Assumptions | User has a personal email |
| Notes and Issues | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC02 | | |
| Use Case Name: | Verify Account | | |
| Created By: | Wee Teck | Last Updated By: | Gordon |
| Date Created: | 13 Feb 2023 | Date Last Updated: | 14 Feb 2023 |

|  |  |
| --- | --- |
| Actor: | Database |
| Description: | System database verifies that the user account email and password exists and are matched. |
| Preconditions: | 1. Device must be connected to Wi-Fi or Mobile Data. 2. User must input their email and password. |
| Postconditions: | 1. User will be able to log in with their email account and password. |
| Priority: | High |
| Frequency of Use: | Medium |
| Flow of Events: | 1. System queries the database to find a match between the submitted user email and password. 2. On a successful match of the email and password, the user is able to log in. |
| Alternative Flows: | **AF-S1:** **No Match Found**   1. There does not exist an email with such a corresponding password in the database. 2. System displays the message “Invalid email account or password”. 3. User is redirected back to the login page. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC03 | | |
| Use Case Name: | Sign Up | | |
| Created By: | Hazel | Last Updated By: | Hazel |
| Date Created: | 13 Feb 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, Account List |
| Description: | User signs up as a new user. System will store details of user into the database upon successful registration. User will be able to sign in as existing user. |
| Preconditions: | 1. Device must be connected to Wi-Fi or Mobile Data |
| Postconditions: | 1. New user data must be saved into account database 2. User must be able to log in after this |
| Priority: | High |
| Frequency of Use: | Medium |
| Flow of Events: | 1. User clicks on “Sign Up” 2. System shows a sign up page with input fields to accept username, email, password and confirm password 3. User inputs a username, valid email, password and confirms password 4. System checks if email and password is valid 5. If yes, System logs new account data into database 6. System redirects and loads website, logging in to the newly created account |
| Alternative Flows: | **AF-S1: Email Address already exists**   1. If the email address already exists, system will display error message “Email already exists.” 2. Return to step 2   **AF-S2: Email Address is missing ‘@’**   1. If the email address is missing a @, system will display error message “Please include an ‘@’ in the email address.” 2. Return to step 2   **AF-S3: Password is not strong enough**   1. System displays error message “Password must be at least 7 characters.” 2. Return to step 2 |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | User has an email address |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC04 | | |
| Use Case Name: | Take Test | | |
| Created By: | Chu Feng | Last Updated By: | Hazel |
| Date Created: | 13 Feb 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User selects “Take Test” to take the test. The test will allow the system to recommend to the user a set of courses and universities catered for them. |
| Preconditions: | 1. Device is connected to Wi-Fi or mobile data 2. User is currently logged into an account |
| Postconditions: | 1. Recommended schools and courses are displayed to the user after completing the test. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User selects “Take Test” to begin taking the test via the following use cases    1. UC05: RIASEC Personality test    2. UC06: Input subject interests    3. UC07: Input academic qualification 2. System outputs course recommendations to the user via UC08: Recommendations |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | UC05: RIASEC Personality test  UC06: Input subject interests  UC07: Input academic qualifications  UC08: Recommendations |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC05 | | |
| Use Case Name: | RIASEC Personality Test | | |
| Created By: | Gordon | Last Updated By: | Hazel |
| Date Created: | 15 Feb 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, Database |
| Description: | User answers questions about their personality. Users' results are saved to the database. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data. 2. User is currently logged into an account. |
| Postconditions: | 1. User’s RIASEC personality test scores are recorded in the database 2. User is directed to the page to input subject interests |
| Priority: | High |
| Frequency of Use: | Medium |
| Flow of Events: | 1. User is presented with a set of questions, and needs to check the checkboxes if they agree with the statement 2. User selects “Submit” to submit their information 3. System saves RIASEC personality test scores into the database. 4. System will then guide users to UC06: Input academic qualification |
| Alternative Flows: | **AF-S1: Missing input**   1. System displays error message “Missing input, please questions above” 2. Return to step 2. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC06 | | |
| Use Case Name: | Input subject interests | | |
| Created By: | Hazel | Last Updated By: | Hazel |
| Date Created: | 7 Apr 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, Database |
| Description: | User inputs their top three subject interests |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data. 2. User is currently logged into an account. 3. User has completed the RIASEC personality test. |
| Postconditions: | 1. User’s top three interested subjects are recorded in the database 2. User is directed to the page to input their academic portfolio |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User is presented with 3 dropboxes to select their favorite subjects from the options presented 2. User selects “Submit” to submit their information 3. System saves user input into database. 4. System will then guide users to UC07: Input academic qualifications |
| Alternative Flows: | **AF-S1: Missing input**   1. System displays error message “Please select an item in the list” 2. Return to step 1. |
| Exceptions: | **-** |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC07 | | |
| Use Case Name: | Input academic qualifications | | |
| Created By: | Gordon | Last Updated By: | Hazel |
| Date Created: | 14 Feb 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, Database |
| Description: | User inputs their academic qualifications |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data. 2. User is currently logged into an account. 3. User has completed the RIASEC personality test. 4. User has submitted their top three subject interests. |
| Postconditions: | 1. User’s academic qualifications, either in the form of GCE A Level grades or Polytechnic GPA, must be inputted 2. User will be directed to course recommendation page |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User is presented with fields to submit their personal information including their academic qualifications. 2. They can toggle between ‘A Levels’ and ‘Polytechnic’. 3. The ‘A Levels’ option displays four dropdown boxes to select grades for their 3 H2 and 1 H1 subjects, while the ‘Polytechnic’ option displays an input field for User to input their GPA. 4. User selects “Submit” to submit their information. 5. System saves user input into database. 6. System will then guide users to UC08: Recommendations |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | School Diplomas include GCE A Levels and Polytechnic |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC08 | | |
| Use Case Name: | Display RIASEC results | | |
| Created By: | Nichele | Last Updated By: | Nichele |
| Date Created: | 7 Apr 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, Database |
| Description: | Users are able to view their RIASEC test results and their descriptions. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data. 2. User is currently logged into an account. 3. User has completed the personality test. 4. User has submitted their top three subject interests. 5. User has submitted their academic qualifications. |
| Postconditions: | 1. User’s top 3 RIASEC types are displayed. 2. Detailed description of each RIASEC type displayed. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. System retrieves User’s top 3 RIASEC types and their descriptions. 2. User is presented with their top 3 RIASEC types and their descriptions. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC09 | | |
| Use Case Name: | Recommendations | | |
| Created By: | Gordon | Last Updated By: | Nichele |
| Date Created: | 15 Feb 2023 | Date Last Update: | 9 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Users are able to view their Holland Code as well as the course and university recommendations for them. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data. 2. User is currently logged into an account. |
| Postconditions: | 1. User can view Holland Code, course and university recommendations 2. User can use the Course Search function to efficiently look for specific course recommendations. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. System outputs a set of recommendations to the user 2. User searches for specific course recommendations via UC14: Course Search. 3. User views detailed information of course and respective universities via UC10: View course-specific information |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC10 | | |
| Use Case Name: | View course-specific information | | |
| Created By: | Gordon | Last Updated By: | Hazel |
| Date Created: | 15 Feb 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | System displays information about course and university |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account 3. User has selected a recommendation |
| Postconditions: | System displays university description, ranking, Course description, ranking, IGP and employability outcomes of selected course as well as campus location. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. System displays    1. University    2. IGP    3. UC08: Display RIASEC Results    4. Related Subjects    5. Prerequisite Subjects    6. Additional Requirements    7. UC11: Display GES Data |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | 1. UC08: Display RIASEC Results 2. UC11: Display GES Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC11 | | |
| Use Case Name: | Display GES Data | | |
| Created By: | Nichele | Last Updated By: | Nichele |
| Date Created: | 7 Apr 2023 | Date Last Updated: | 7 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, data.gov.sg API, Google Charts API |
| Description: | System displays more information about relevant course and university from the Graduate Employment Survey (GES) data retrieved from data.gov.sg. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account 3. User has selected a recommended course from a university. |
| Postconditions: | System displays information on the course employment rate (%), mean starting income (SGD), and a graph visualization of the change of starting salary over the years. |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. System receives GES data from the data.gov.sg API. 2. System displays    1. Course Employment Rate (%)    2. Graduate Mean Starting Income (SGD)    3. Graph visualization of starting salary over the years. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC12 | | |
| Use Case Name: | Compare Courses | | |
| Created By: | Nichele | Last Updated By: | Stephen |
| Date Created: | 9 Apr 2023 | Date Last Updated: | 9 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User, data.gov.sg API, Google Charts API |
| Description: | Users can compare employment outcomes, starting salary and cut off grades between selected courses from specific universities. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account |
| Postconditions: | System displays bar graphs that compare employment outcomes, starting salary and cut off grades between selected courses from the specified university. |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. User selects courses from specific universities that he/she wishes to compare. 2. System receives GES data from the data.gov.sg API to get information on employment outcomes. 3. System plots and displays bar graphs for the respective courses, comparing    1. Graduate Employment Rate    2. Starting Salary    3. A level cutoff grade    4. Polytechnic cutoff grade |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC13 | | |
| Use Case Name: | View All Courses | | |
| Created By: | Nichele | Last Updated By: | Nichele |
| Date Created: | 9 Apr 2023 | Date Last Updated: | 9 Apr 2023 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User can view all courses that each university offers. |
| Preconditions: | 1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account |
| Postconditions: | System displays a search bar, as well as lists of all courses, grouped by university. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. System displays the lists of all courses under each university. 2. User can search for a specific course via UC14: Course Search. 3. User can click on the course they want to know more, and then clicking on the magnifying glass icon will take them to course specific page 4. User will be directed to UC10: View course-specific information page to view information about course |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

# 

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC14 | | |
| Use Case Name: | Course Search | | |
| Created by: | Stephen | Last Updated by: | Stephen |
| Date created: | 9 April 2023 | Date Last Updated: | 9 Apr 2023 |

|  |  |
| --- | --- |
| Actor | User |
| Description | Users can search for specific information about a particular course. |
| Preconditions | 1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account. 3. User must be in “View all Courses” or “Recommendations” page |
| Postconditions | 1. Users will be brought to page with detailed information |
| Priority | Medium |
| Frequency of Use | Medium |
| Flow of events | 1. The user is required to enter the name of a particular course into the dropdown search bar. 2. The user selects one of the available options and will be directed to UC10: View course-specific information page to view information about the course. |
| Alternative flow: | AF-S1: The course does not exist   1. If there are no available courses to select, then the particular course does not exist 2. Return to step 1. |
| Exceptions: | - |
| Includes | UC10: View course-specific information |
| Special Requirements: | - |
| Assumptions | User has a personal email. |
| Notes and Issues | - |

Appendix A: Glossary

|  |  |  |
| --- | --- | --- |
| **Term** | **Definition** | **Example** |
| High School Graduates | An individual who has received a high school diploma or passed the general educational development (GED) diploma test. | - |
| Extra-curricular activities | Activities, sports and programs that either complement or extend what students are learning in school. | Basketball |
| Region of residence | The area that the user resides in | West Region of Singapore. |
| Course | An extended period of organized study, a College major, often leading to a qualification | Bachelor Degree in Computer Science |
| College/University | An educational institution. These are usually degree-awarding tertiary educational institutions. | Nanyang Technological University |
| Myers Briggs Type Indicator | An instrument to analyze an individual's strengths and preferences, yielding 1 of 16 personality types which can be used to identify one's ideal career. | ISTJ |
| School ranking | The graded ranking of a school in comparison to others in the world. This project will be using university rankings by Quacquarelli Symonds | - |
| Account | A unique profile that stores a user’s personal detail and saved information on the website | - |
| Account Email | User’s registered email, which can uniquely identify the user to gain access to a computer network or online system | chengordon 8@gmail.com |
| Account Password | A string of characters to verify the identity of a user during the authentication process | 9fy931!Jh |

# System Diagrams

## Diagram Description automatically generatedUse Case diagram

## Diagram Description automatically generatedDialog map

## Diagram Description automatically generatedSystem architecture

## Diagram Description automatically generatedClass diagram

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

## Sequence diagrams

Diagram

Description automatically generated with medium confidence

**A picture containing chart

Description automatically generated**

Graphical user interface

Description automatically generated with low confidence

**Chart

Description automatically generated with medium confidence**

Diagram

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Timeline

Description automatically generated

Appendix B: Analysis Models

B1: Black box testing

B1.1.1. Sign up for account

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User Email | First Name | Password | Confirm Password | Expected Result (Oracle) | Test Result (Log) |
| user1@email.com | User | Password1 |  | Passwords don’t match | Pass |
| user1@email.com | User |  | Password1 | Passwords don’t match | Pass |
| user1@email.com |  | Password1 | Password1 | First name must be greater than 1 character | Pass |
|  | User | Password1 | Password1 | Email must be greater than 3 characters | Pass |
| U | User | Password1 | Password1 | Email must be greater than 3 characters | Pass |
| existing@email.com | User | Password1 | Password1 | Email already exists | Pass |
| user1@email.com | User | Password1 | Password0 | Passwords don’t match | Pass |
| user1@email.com | User | Pas1 | Pas1 | Password must be at least 7 characters | Pass |
| user1@email.com | User | Password1 | Password1 | Account created! | Pass |

B1.1.2. Login

|  |  |  |  |
| --- | --- | --- | --- |
| User Email | Password | Expected Result (Oracle) | Test Result (Log) |
| user10101@email.com | Password1 | Email does not exist | Pass |
| user1@email.com | Password10101 | Incorrect password, try again | Pass |
|  | Password1 | Email does not exist | Pass |
| user1@email.com |  | Email does not exist | Pass |
| user1@email.com | Password1 | Done loading | Pass |

B2: White box testing

B2.1: Sign up for account

Sign up for account

Email valid

Firstname valid

Password valid

Passwords are the same

False

Email must be greater than 3 characters

Email already exists

First name must be greater than 1 character

Password must be at least 7 characters

Passwords don’t match

Email already exists

False

False

False

Homepage

False

True

True

True

True

True

1

2

3

4

5

7

6

8

9

10

11

12

|  |  |
| --- | --- |
| Test cases | Execution Paths |
| Field is left blank or email is too short | 1, 2, 3 |
| Field for firstname is left blank or not longer than 1 character | 1, 2, 4, 5 |
| Field for password is left blank or password is not longer than 7 characters | 1, 2, 4, 6, 7 |
| Input in “Password” and “Confirm password” fields are not the same | 1, 2, 4, 6, 8, 9 |
| There exists an account in the database with the input email | 1, 2, 4, 6, 8, 10, 11 |
| Account is successfully created | 1, 2, 4, 6, 8, 10, 12 |

**Cyclomatic Complexity:**  
CC1 = |edges| - |nodes| + 2 = 16 – 12 + 2 = 6

CC2 = |decisionpoint| + 1 = 5 + 1 = 6

🡺 CC1 = CC2 := CCB1.2.1. , hence all decision points are binary

B2.2: Login

Login page

Password valid

Incorrect password, try again

Email does not exist

False

Email valid

False

1

2

3

4

Homepage

6

5

|  |  |
| --- | --- |
| Test cases | Execution Paths |
| Wrong password | 1, 2, 3 |
| Email is not associated with any account in the database | 1, 2, 4, 5 |
| Login successful | 1, 2, 4, 6 |

**Cyclomatic Complexity:**  
CC3 = |edges| - |nodes| + 2 = 7 – 6 + 2 = 3

CC4 = |decisionpoint| + 1 = 2 + 1 = 3

🡺 CC3 = CC4 := CCB1.2.2. , hence all decision points are binary

Appendix C: To Be Determined List

In this appendix, we will provide links to all images found in this documentation. The demonstration video and the source code in a GitHub repository will also be linked.

Application Walkthrough

https://www.youtube.com/watch?v=M2Z-7QSoXwM&t=37s

Source Code

https://github.com/GordonChen19/FindMyCollege-Source-Code