Software Requirements Specification

for

FindMy**College**

Version 3.2

Prepared by SC2006

Nanyang Technological University

01 April 2023

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 User Documentation
- 2.6 Assumptions and Dependencies

3. External Interface Requirements
- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces

4. System Features
- 4.1 Data sets
- 4.2 Sign up and login page
- 4.3 Personality Test page
- 4.4 Input page
- 4.5 Recommendations page
- 4.6 User preferences
- 4.7 Course and College Introduction Page

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Usability Requirements
- 5.3 Reliability Requirements
- 5.4 Security Requirements
- 5.5 Maintainability Requirements
- 5.6 Software Quality Attributes

6. Use case descriptions

7. Diagrams
- 7.1 Initial Use Case diagram
- 7.2 Dialog map
- 7.3 System architecture
- 7.4 Class diagram
- 7.5 Sequence diagrams

Revision History

Name	Date	Reason For Changes	Version
Decided on main topic of project	16 Jan 2023	Set the direction for our project to be a college recommender website after a team discussion.	1.0
Edited the flow of events of user experience on the website	30 Jan 2023	Refined sequence of events when user loads into website to make it more intuitive to use. Hence we added a separate button to take personality test	1.1
Changed certain attributes that the user can input, which affects the final recommended list of courses	4 Feb 2023	Instead of indicating the value of monetary benefits from a scholarship, we changed it into a yes/no option.	1.2
Changed the type of personality test that is to be used	13 Feb 2023	After even more in-depth research, we understood that the original idea of using the 16personalities test is not very relevant and accurate when using it to decide a career path. Hence, we changed it to use the Holland Codes which is used by the RIASEC test.	1.3
Removed the API for google maps	13 Feb 2023	Decided that it was not very impactful for our project.	1.4
Changed classes in the class diagram	27 Feb 2023	Realised that there were some important classes were missing and some redundant classes in the original class diagram. So we edited it to ensure that all the classes were properly defined and present so we can maintain modularity of the individual components, so there will not be an overlap of functionalities, ensuring good software design.	2.0
Refined sequence diagram	27 Feb 2023	After improving the class diagram, the sequence diagram has to be edited. Other changes were made to ensure a smoother user experience.	2.0.1
Changed the layout of the RİASEC test on the website	15 Mar 2023	At first, the user must click the 'Next' button to go to the next page when doing the test. We changed the test such that it fits into a single page, allowing for a smoother and hassle-free user experience.	2.1
Added in a link to display all the course information	20 Mar 2023	More information about the course can be provided so users are able to select their courses just by using our website. They do not have to search for more information outside, then our website would not be very useful.	2.2

Developed a new mathematical algorithm to recommend courses	23 Mar 2023	Previously, the recommended courses were not really accurate during testing. Hence, we had to come up with a better algorithm.	3.0
Added a feature where users have their information automatically saved	25 Mar 2023	It is more logical for user data to be saved, so they do not need to retake the test every time they logout and re-enter the website.	3.1
Added new images and re-designed the website	30 Mar 2023	Made the website more visually appealing	3.2

1. Introduction

1.1 Purpose

The SC2006 team will be responsible for developing the website, *FindMyCollege*, an online college and course recommendation metasearch platform. *FindMyCollege* aims to enable users, primarily high school graduates, to make more informed decisions about which universities and what courses to apply for within Singapore. *FindMyCollege* will recommend particular courses and universities to users based on their qualifications and personality.

1.2 Document Conventions

NTU	:	Nanyang Technological University
NUS	:	National University of Singapore
SMU	:	Singapore Management University
SIT	:	Singapore Institute of Technology
SUTD	:	Singapore University of Technology and Design
SUSS	:	Singapore University of Social Sciences
API	:	App Programming Interface
IGP	:	Indicative Grade Profile, in Singapore, it is known as a university's cut off points. It reflects the grades of successful applicants to that university. The respective grades at the 10^{th} and 90^{th} percentile of a cohort in each degree programme is displayed.
RIASEC	:	The Holland Codes six personality types namely **Realistic, Investigative, Artistic, Social, Enterprising and Conventional**

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

1.3 Intended Audience and Reading Suggestions

The software requirements specification (SRS) document serves as a blueprint for anyone who wants to understand the scope and functionality of the website. This documentation can be useful and is intended for developers, users, and testers.

Developers can understand the purpose of the project from reading the detailed information about the requirements and functionality of the website. This understanding enables developers to create a clear plan for the development process. As such, this ensures that the end product is as useful for the targeted audience as possible.

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.

Through this document, users can understand the features of the website, what the website offers and how it works. Users can use this information to evaluate whether the website is useful for them to choose the right school and course.

In summary, this SRS document provides essential information for all stakeholders involved in the project. It helps the reader understand more about the website.

1.4 Product Scope

FindMyCollege website is a user-friendly free-of-charge application where users can sign up for an account. An account is required for each user to input their academic qualifications, namely their subject interests and grades. The website covers 3 types of students, those from 'Advanced ' Level curriculum, 'International Baccalaureate' Diploma and 'Polytechnic' students. Users can take a popular career personality test, the RIASEC Profiling Tool. It consists of 6 categories known as the "Holland Code". It groups people on the basis of their suitability for six different categories of occupations. The six categories have the 'RIASEC' acronym.

- **R: Realistic**
    - Individuals who possess a Realistic inclination prefer work that involves dealing with tangible, practical problems and finding solutions through hands-on activities. Generally, those with Realistic interests tend to dislike careers that require paperwork or extensive collaboration with others. They derive satisfaction from tasks that involve working with living organisms such as plants and animals, real-life resources such as wood, as well as using equipment and machinery, and working outdoors.
    - Eg. Architecture, Arts, Law, Science, Engineering

- **I: Investigative**
    - Individuals who have a preference for Investigative pursuits are inclined towards tasks that involve contemplation and mental exertion, as opposed to physical labor or management responsibilities. They derive satisfaction from exploring information and devising solutions to complex issues.
    - Eg. Information technology, Law, Science, Engineering, Mathematics

- **A: Artistic**
    - Individuals who possess an inclination towards the arts are fond of tasks that relate to the creative aspect of fields such as acting, music, art, and design. They appreciate innovation in their work and prefer tasks that allow them to work without rigid guidelines.
    - Eg. Arts, Education, Hospitality and tourism, Marketing

- **S: Social**
    - Individuals who possess social inclinations enjoy collaborating with others to facilitate their personal and professional development. They tend to prefer interacting with individuals over dealing with inanimate objects, machinery, or data. They derive pleasure from educating, offering guidance, assisting, and providing service to others.
    - Eg. Education, Government and public administration, Law, Sales and service

- **E: Enterprising**
    - Individuals with an enterprising inclination are interested in tasks related to initiating and executing business ventures. Such individuals prefer to take

action instead of contemplating ideas. They derive satisfaction from convincing and directing others, making judgments, and engaging in risky endeavors for the purpose of gaining profits.
    - Eg. Business, Finance, Marketing

- **C: Conventional**
    - Individuals who possess Conventional interests tend to favor tasks that involve established procedures and routines. They generally prefer dealing with concrete information and focusing on specifics rather than working with abstract ideas. Additionally, they tend to enjoy having clear guidelines and instructions to follow, as well as working with a decisive authority figure.
    - Eg. Architecture, Business management, Finance, Manufacturing

2. Overall Description

2.1 Product Perspective

Choosing the right course in university can be a daunting task for many high school students. There are various factors to consider such as personal interest, academic scores, career goals, and more. However, the decision making process can be simplified with the help of our web application, FindMyCollege. Our web application aims to provide students with a personalized recommendation of the courses based on their personality type, academic performance, and career preferences. By taking a comprehensive personality test, students will receive a list of recommended courses that match their personal interests and strengths. Our web application aims to help students make informed decisions about their academic future and help them achieve their career goals.

2.2 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.

2.3 User Classes and Characteristics

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

2.4 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.

2.5 User Documentation

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

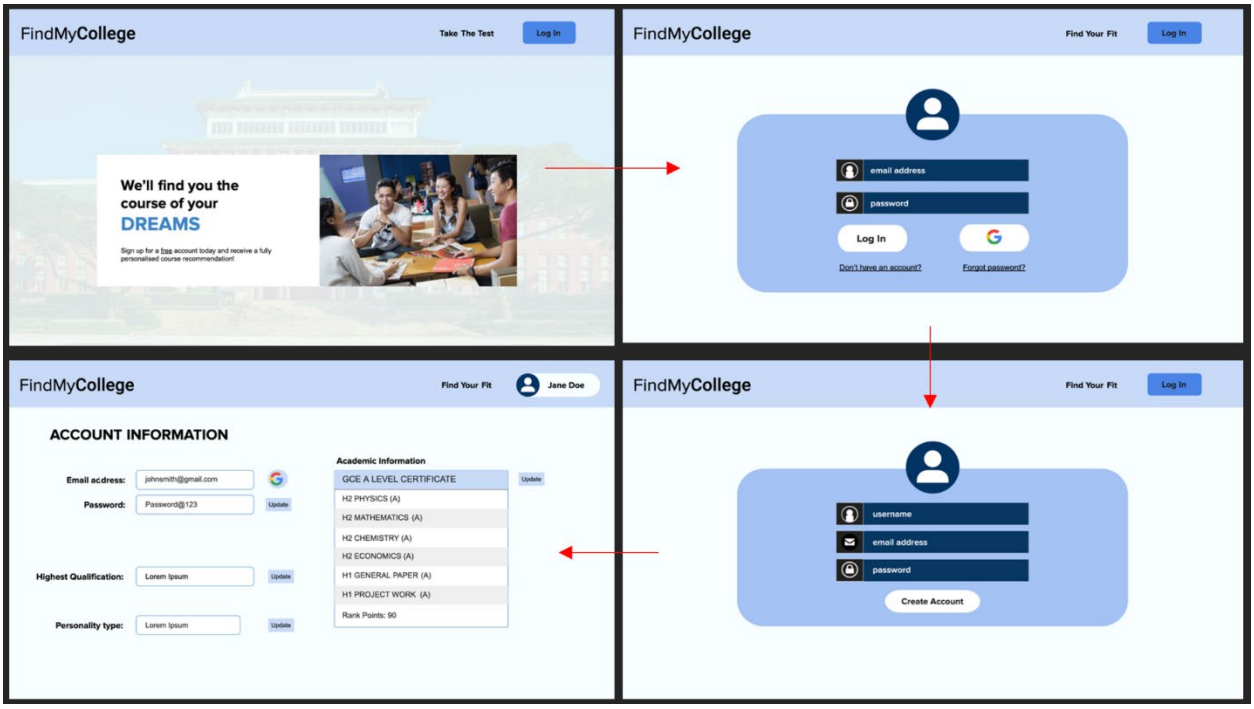
2.6 Assumptions and Dependencies

- 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

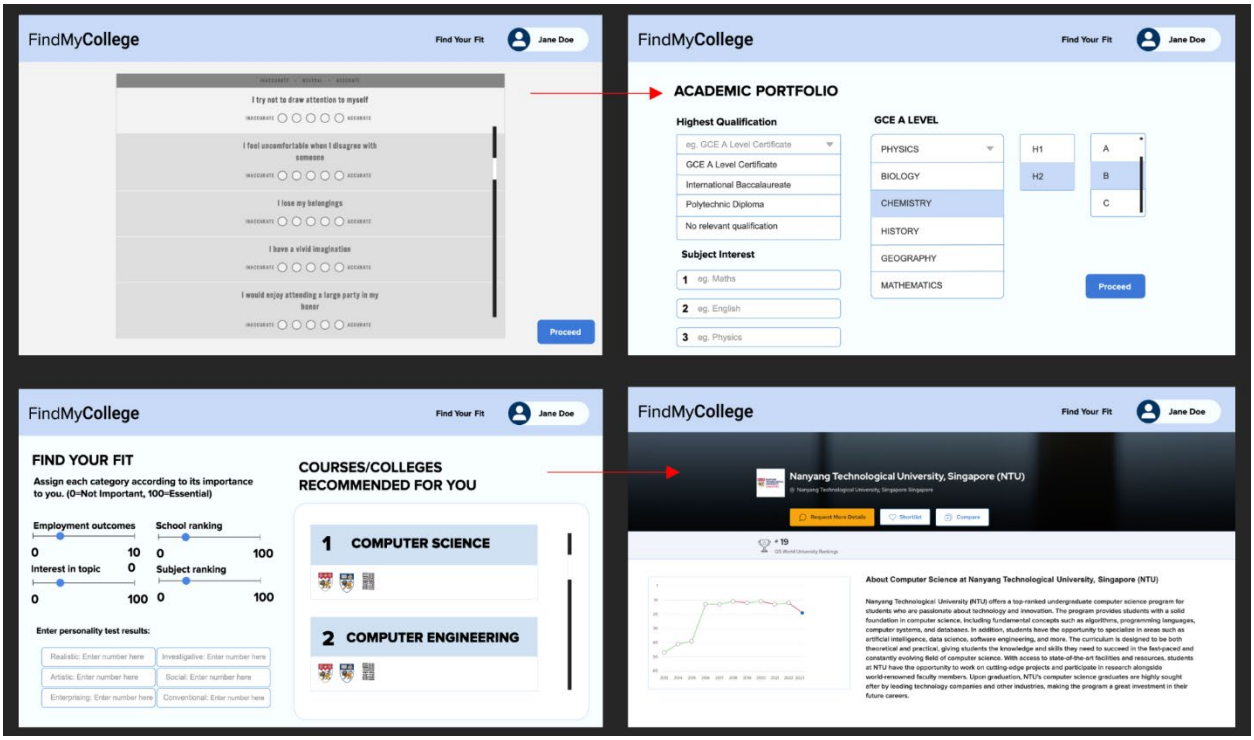
3. External Interface Requirements

3.1 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.



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.



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).

3.2 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.

3.3 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.

4. System Features

4.1 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.

4.2 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.

4.3 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

4.4 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.

4.5 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.

4.6 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.

4.7 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.

5. Other Nonfunctional Requirements

5.1 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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.

5.6 Software Quality Attributes

FindMyCollege adopts a 3-Tier Architecture design, namely:

Presentation Layer	Application Layer	Data Layer
1. HTML 2. CSS 3. JavaScript	1. Python 2. Flask	1. SQLAlchemy

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
    a. 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.
    b. 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.
    c. Maintainability of features
        - Ensured readability of code
        - Included comments to help anyone else reading the source code understand what each function does.
    d. Extensibility of features
        - The flexible architecture allows new modules or components to be added without affecting the existing system.
2. Design Patterns
    a. 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
    a. High cohesion
        - The modules in the system have a single, well-defined responsibility.
    b. 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.
    c. 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)

d. 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.

6. Use case descriptions

Use Case ID:	UC01		
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 a. UC05: RIASEC Personality test b. UC06: Input subject interests

	c. 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. 1. 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 a. University b. IGP c. UC08: Display RIASEC Results d. Related Subjects e. Prerequisite Subjects f. Additional Requirements

	g. 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 a. Course Employment Rate (%) b. Graduate Mean Starting Income (SGD) c. 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 a. Graduate Employment Rate b. Starting Salary c. A level cutoff grade d. 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:	-

7.

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

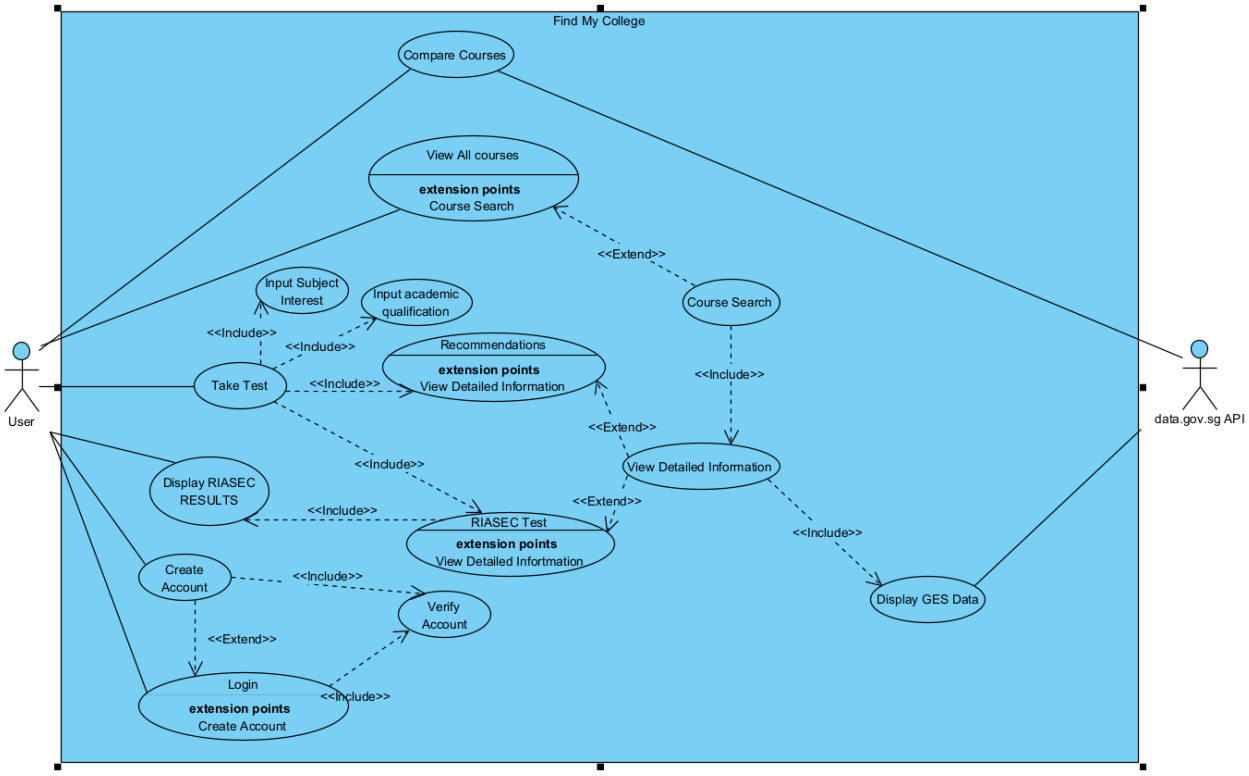
Actor	User
Description	Users can search for specific information about a particular course.
Preconditions	1. Device is connected to Wi-Fi or Mobile Data 2. User must be logged into an existing account. 3. User must be in “View all Courses” or “Recommendations” page
Postconditions	1. Users will be brought 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

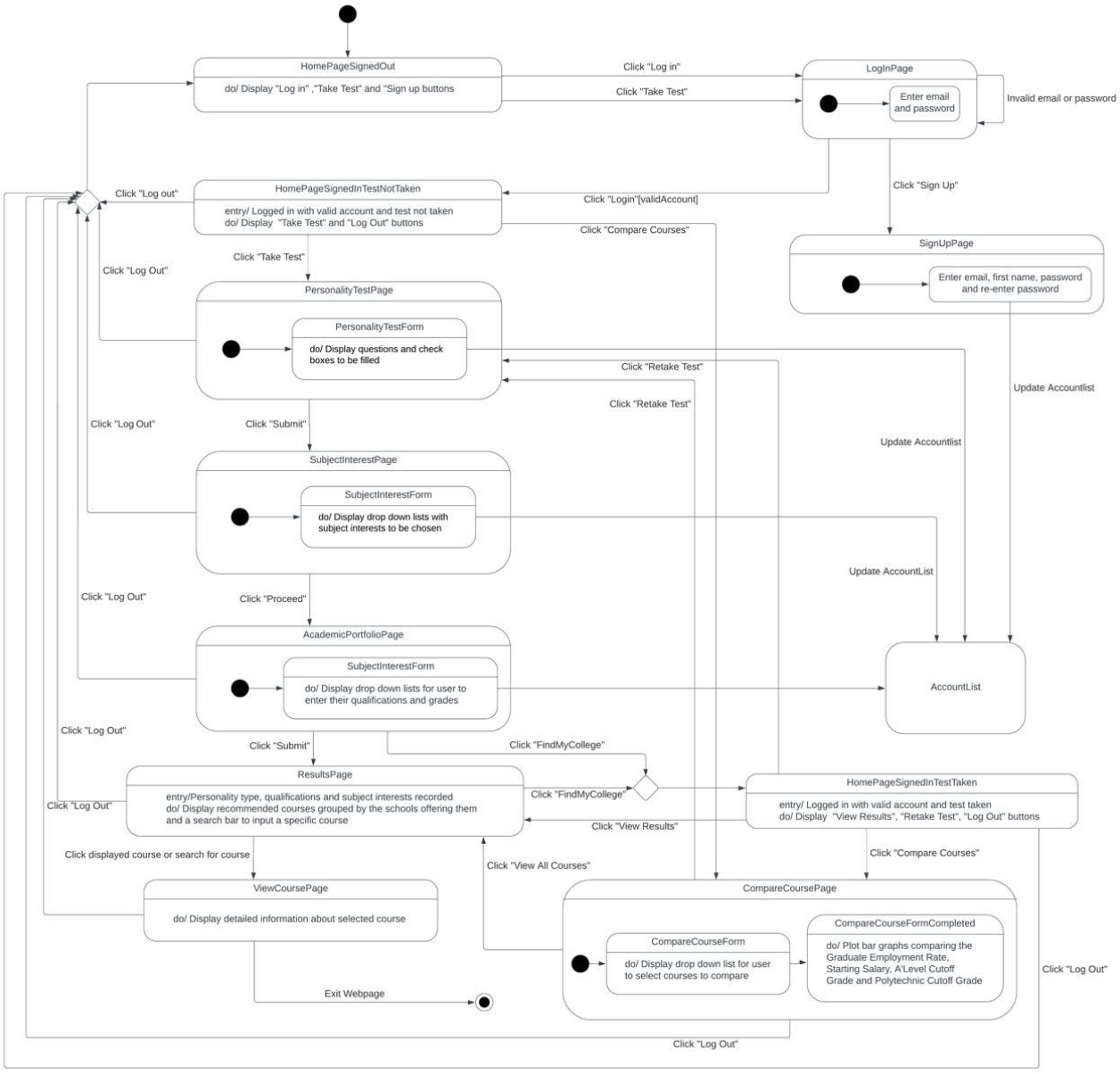
Term	Definition	Example
High School Graduates	An individual who has received a high school diploma or passed the general educational development (GED) diploma test.	-
Extra-curricular activities	Activities, sports and programs that either complement or extend what students are learning in school.	Basketball
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

8. System Diagrams

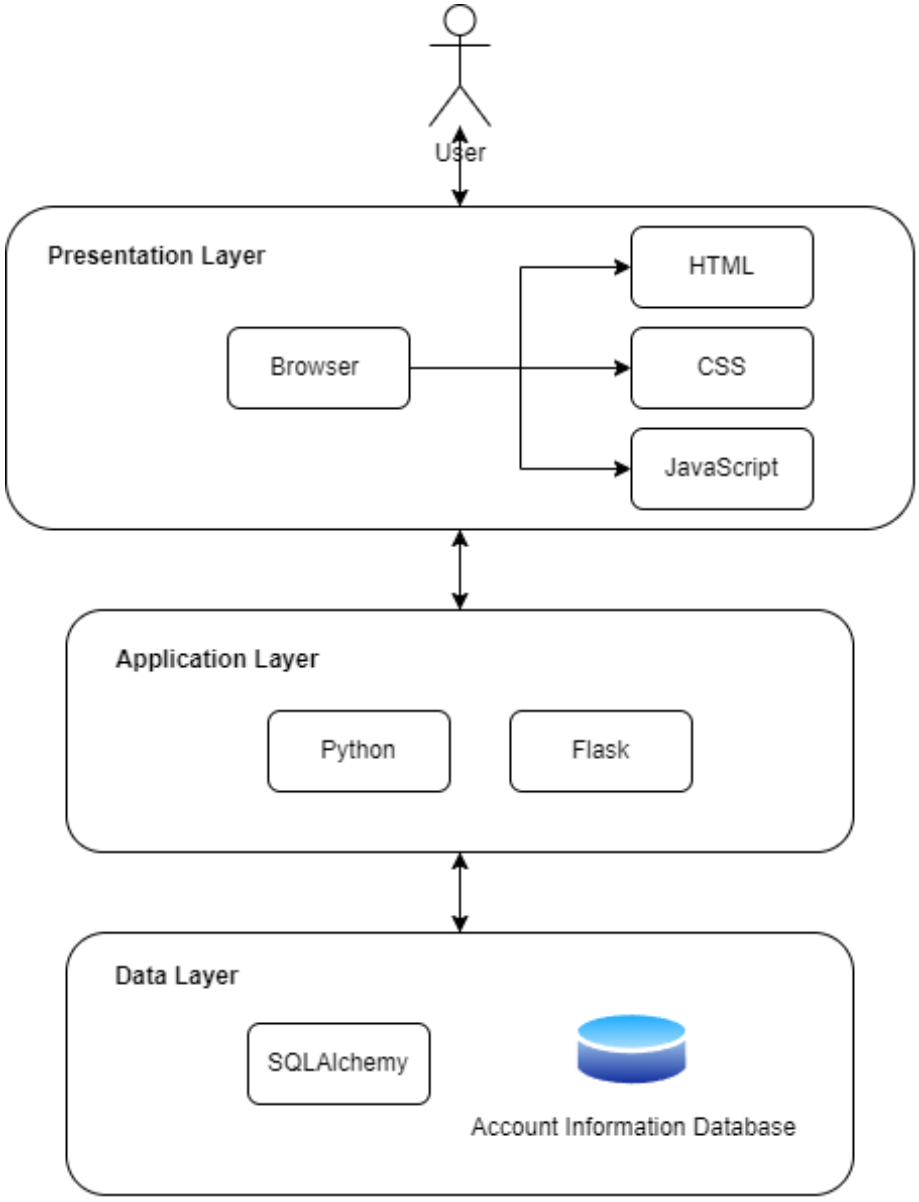
8.1 Use Case diagram



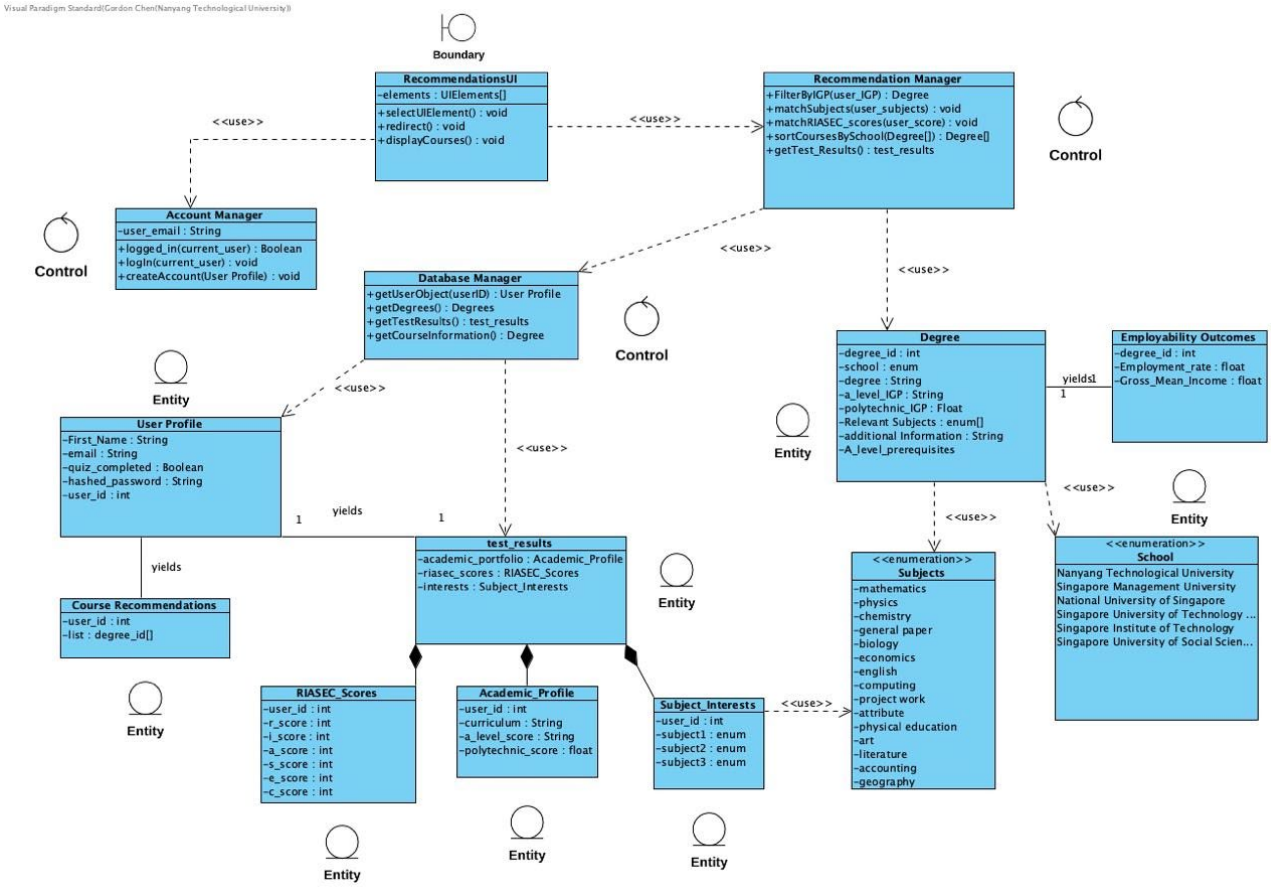
8.2 Dialog map

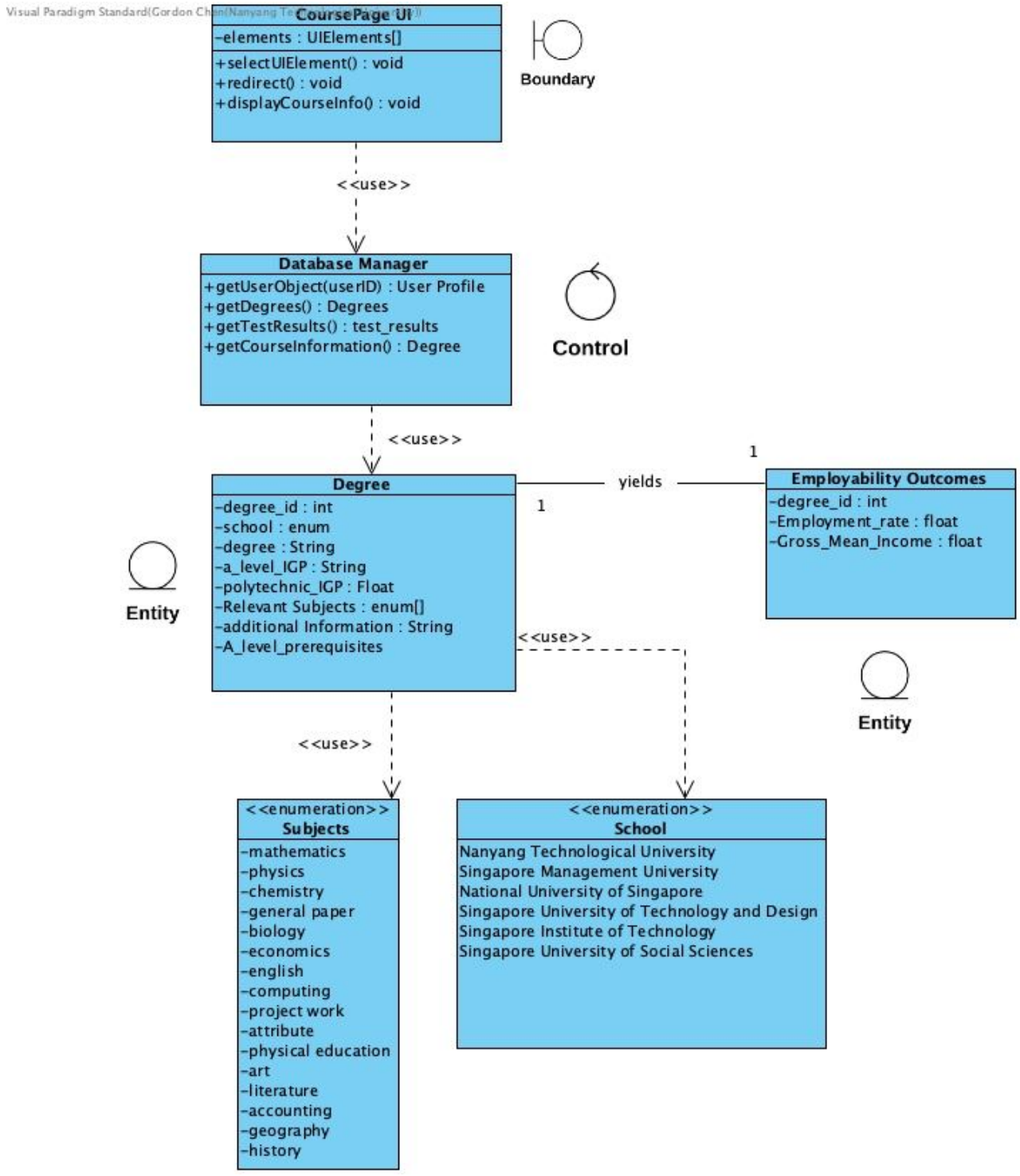


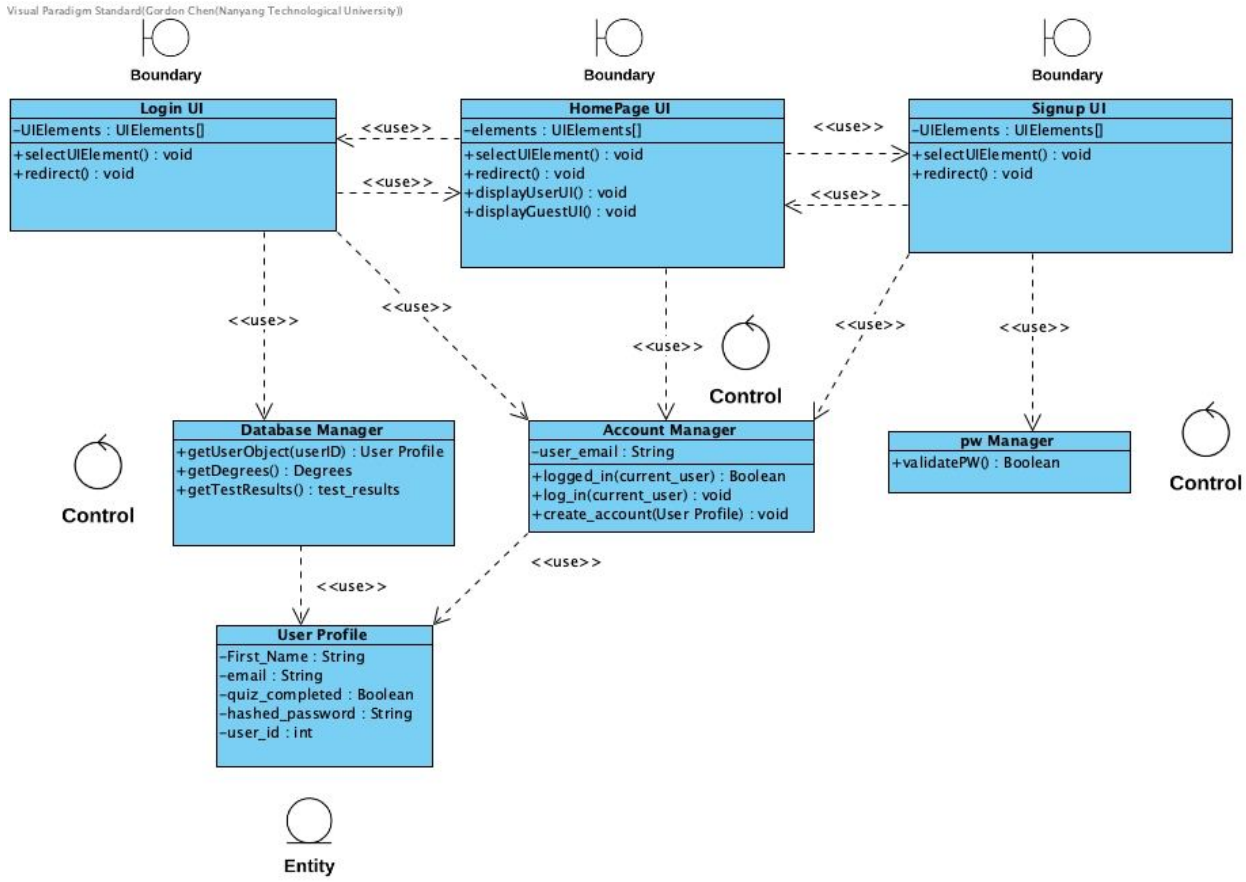
8.3 System architecture

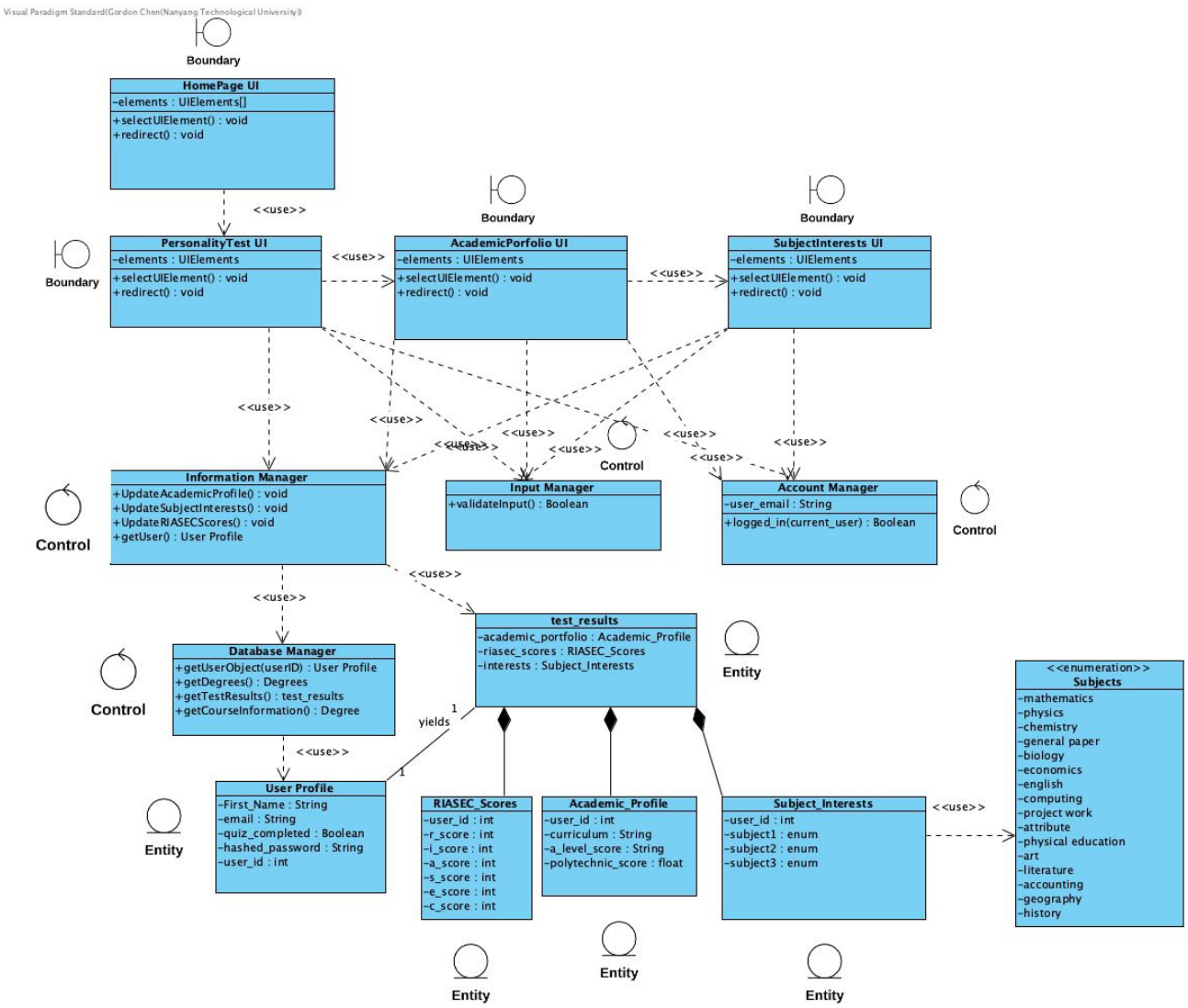


8.4 Class diagram

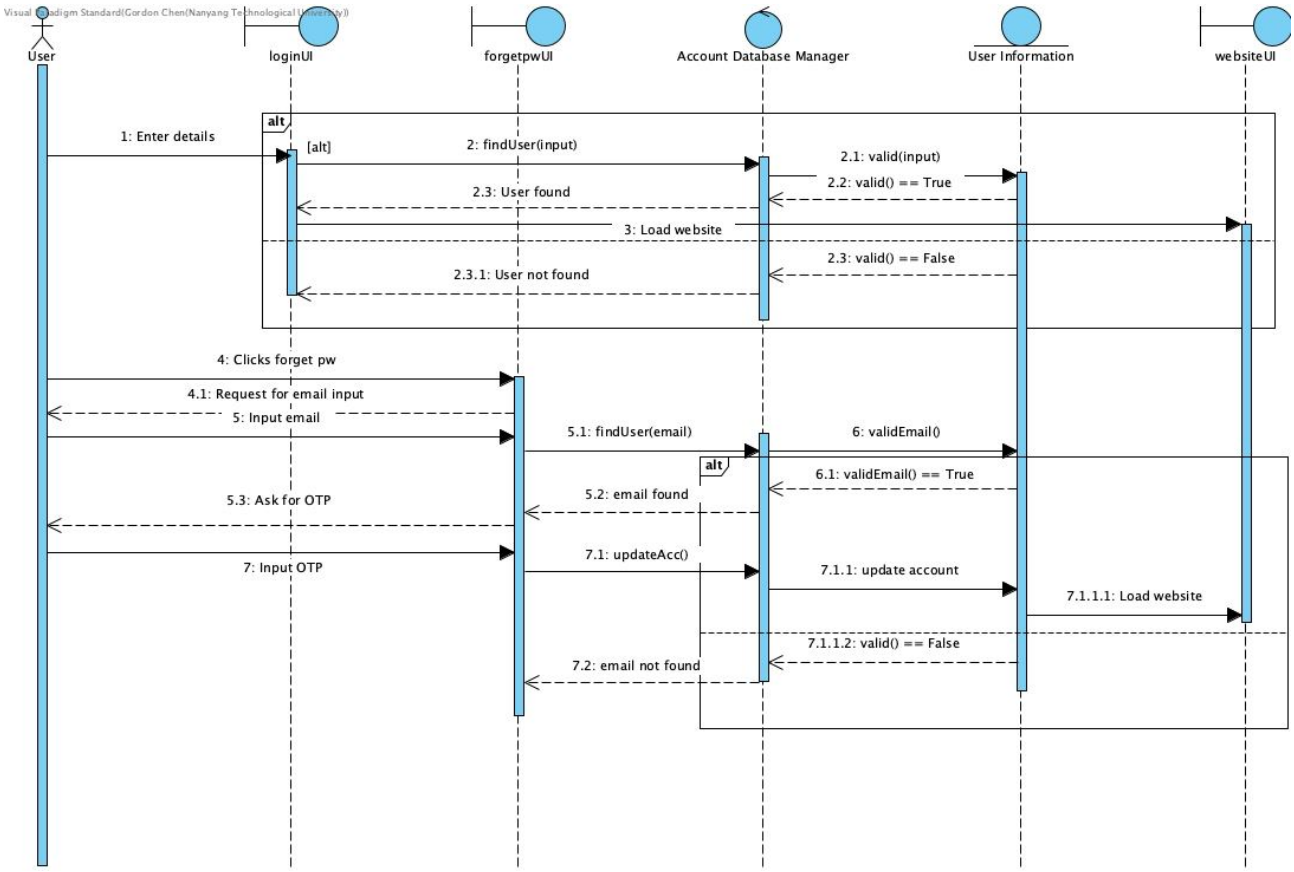


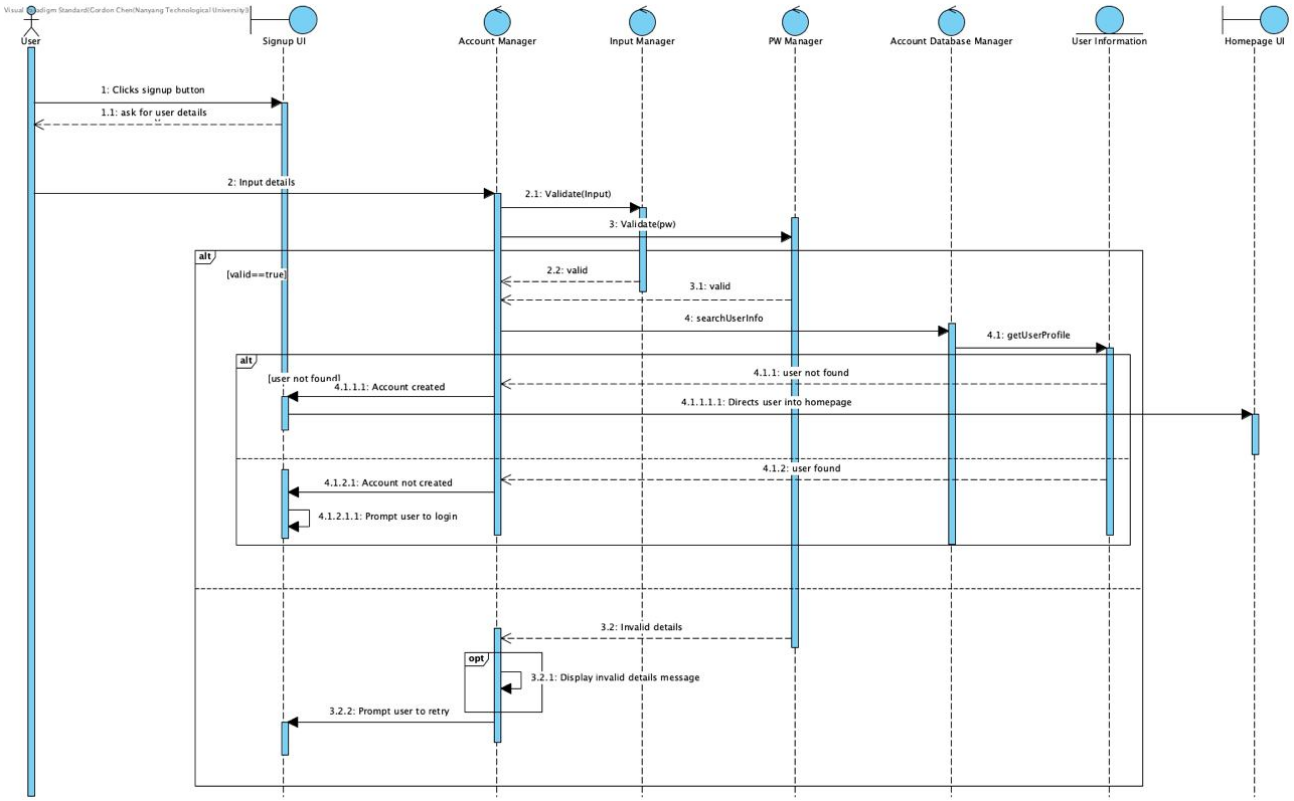

Visual Paradigm Standard(Gordon Chan(Nanyang Technological University))
CoursePage UI
-elements : UIElements[]
+selectUIElement() : void
+redirect() : void
+displayCourseInfo() : void
Boundary
<<use>>
Database Manager
+getUserObject(userID) : User Profile
+getDegrees() : Degrees
+getTestResults() : test_results
+getCourseInformation() : Degree
Control
<<use>>
1
yields
1
Degree
-degree_id : int
-school : enum
-degree : String
-a_level_IGP : String
-polytechnic_IGP : Float
-Relevant Subjects : enum[]
-additional Information : String
-A_level_prerequisites
Employability Outcomes
-degree_id : int
-Employment_rate : float
-Gross_Mean_Income : float
Entity
<<use>>
Entity
<<use>>
<<enumeration>>
Subjects
-mathematics
-physics
-chemistry
-general paper
-biology
-economics
-english
-computing
-project work
-attribute
-physical education
-art
-literature
-accounting
-geography
-history
<<enumeration>>
School
Nanyang Technological University
Singapore Management University
National University of Singapore
Singapore University of Technology and Design
Singapore Institute of Technology
Singapore University of Social Sciences

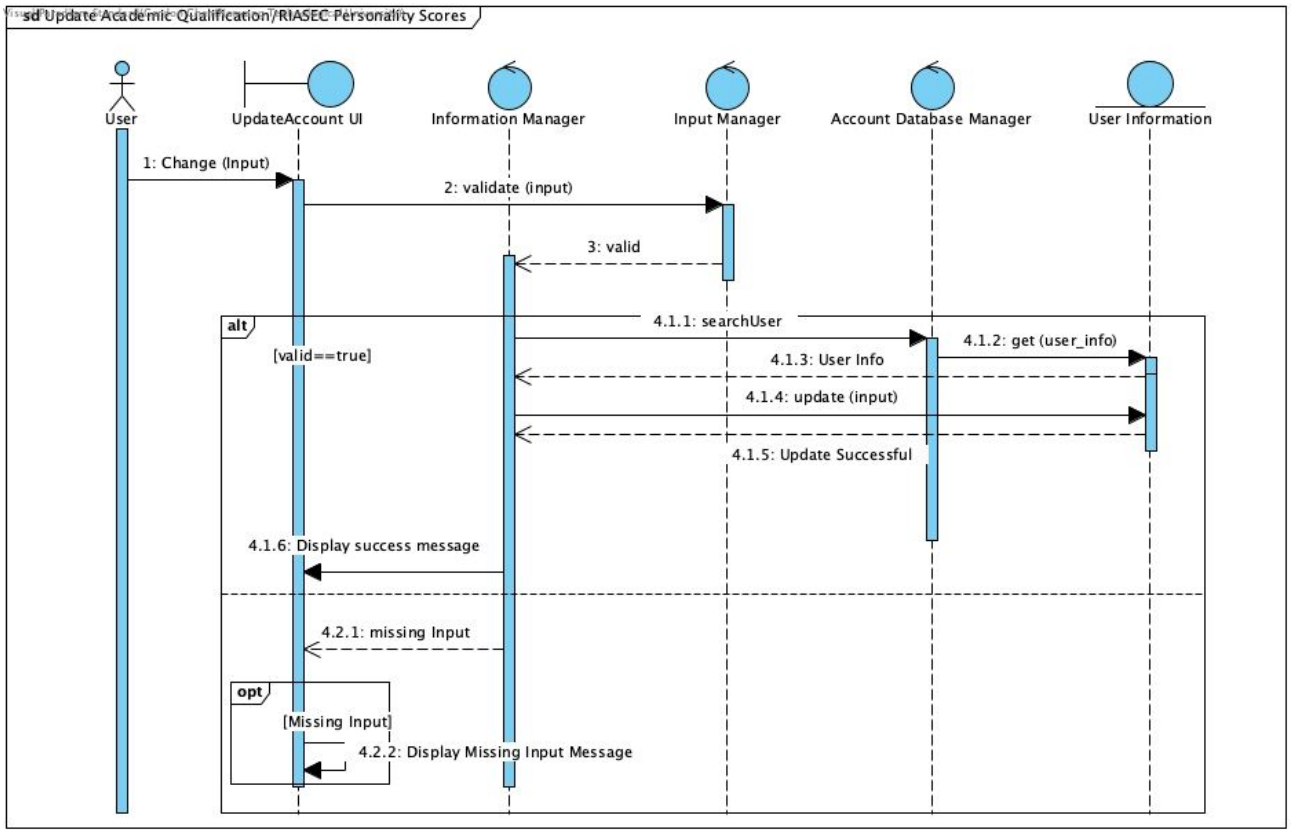

Visual Paradigm Standard(Gordon Chen(Nanyang Technological University))
Boundary
Boundary
Boundary
Login UI
-UIElements : UIElements[]
+selectUIElement() : void
+redirect() : void
HomePage UI
-elements : UIElements[]
+selectUIElement() : void
+redirect() : void
+displayUserUI() : void
+displayGuestUI() : void
Signup UI
-UIElements : UIElements[]
+selectUIElement() : void
+redirect() : void
<<use>>
Control
Control
Control
Database Manager
+getUserObject(userID) : User Profile
+getDegrees() : Degrees
+getTestResults() : test_results
Account Manager
-user_email : String
+logged_in(current_user) : Boolean
+log_in(current_user) : void
+create_account(User Profile) : void
pw Manager
+validatePW() : Boolean
User Profile
-First_Name : String
-email : String
-quiz_completed : Boolean
-hashed_password : String
-user_id : int
Entity

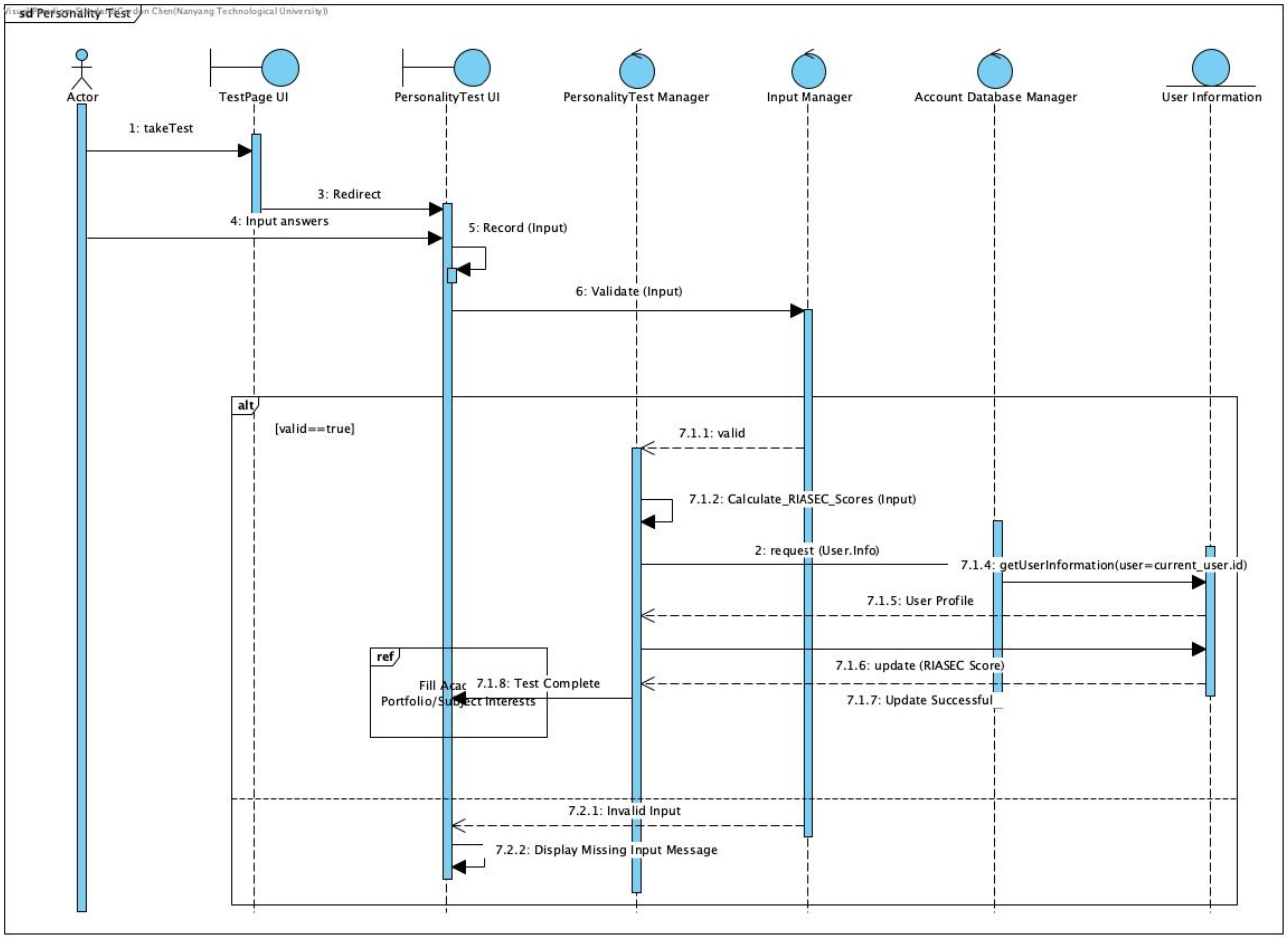

Visual Paradigm Standard(Gordon Chen(Nanyang Technological University))
Boundary
HomePage UI
-elements : UIElements[]
+selectUIElement() : void
+redirect() : void
<<use>>
Boundary
Boundary
Boundary
PersonalityTest UI
-elements : UIElements
+selectUIElement() : void
+redirect() : void
<<use>>
AcademicPorfolio UI
-elements : UIElements
+selectUIElement() : void
+redirect() : void
<<use>>
SubjectInterests UI
-elements : UIElements
+selectUIElement() : void
+redirect() : void
Control
Information Manager
+UpdateAcademicProfile() : void
+UpdateSubjectInterests() : void
+UpdateRIASECScores() : void
+getUser() : User Profile
Input Manager
+validateInput() : Boolean
Account Manager
-user_email : String
+logged_in(current_user) : Boolean
Control
Control
test_results
-academic_portfolio : Academic_Profile
-riasec_scores : RIASEC_Scores
-interests : Subject_Interests
Entity
Control
Database Manager
+getUserObject(userID) : User Profile
+getDegrees() : Degrees
+getTestResults() : test_results
+getCourseInformation() : Degree
1
yields
1
<<enumeration>>
Subjects
-mathematics
-physics
-chemistry
-general paper
-biology
-economics
-english
-computing
-project work
-attribute
-physical education
-art
-literature
-accounting
-geography
-history
Entity
User Profile
-First_Name : String
-email : String
-quiz_completed : Boolean
-hashed_password : String
-user_id : int
RIASEC_Scores
-user_id : int
-r_score : int
-i_score : int
-a_score : int
-s_score : int
-e_score : int
-c_score : int
Academic_Profile
-user_id : int
-curriculum : String
-a_level_score : String
-polytechnic_score : float
Subject_Interests
-user_id : int
-subject1 : enum
-subject2 : enum
-subject3 : enum
Entity
Entity
Entity

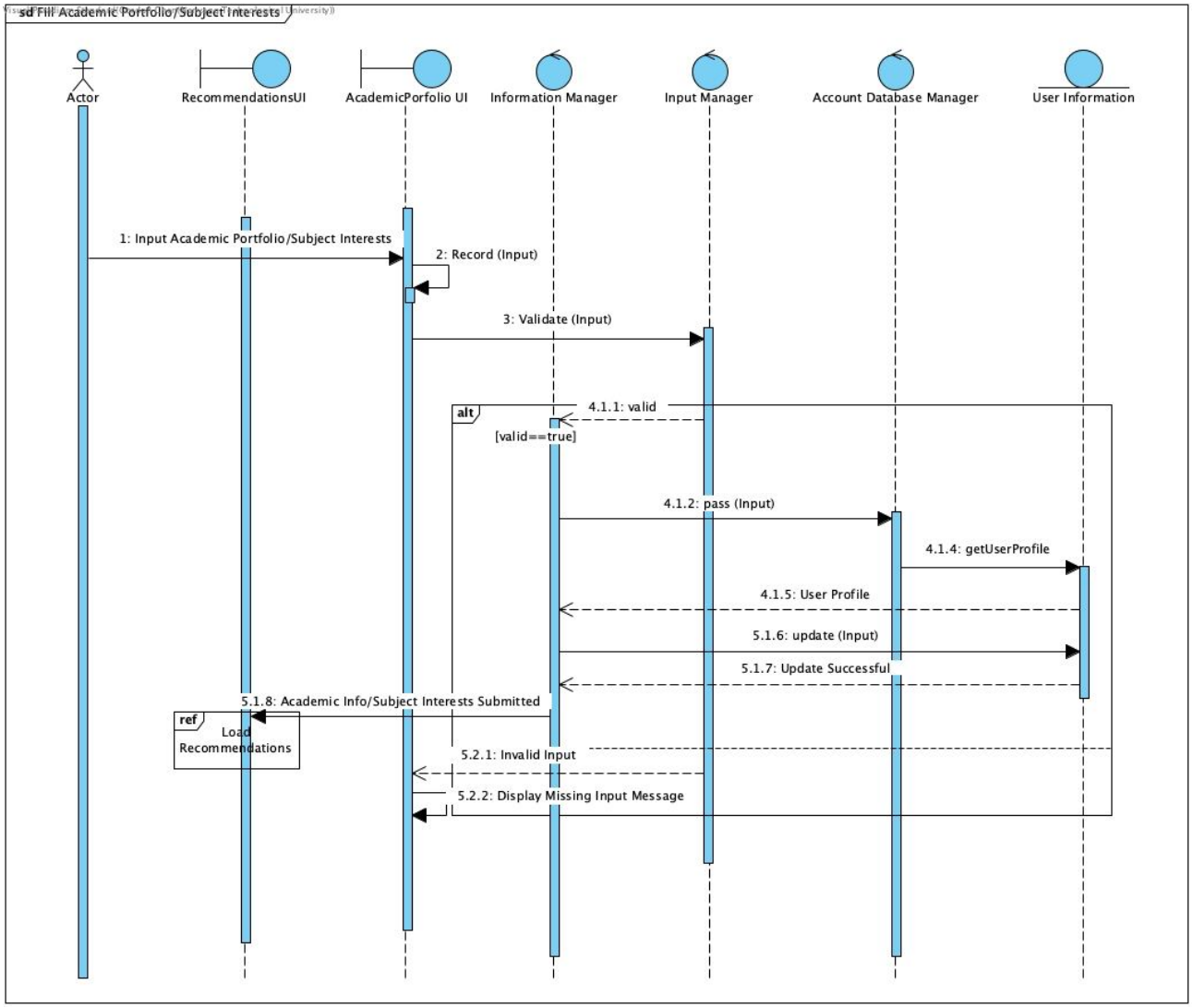
8.5 Sequence diagrams

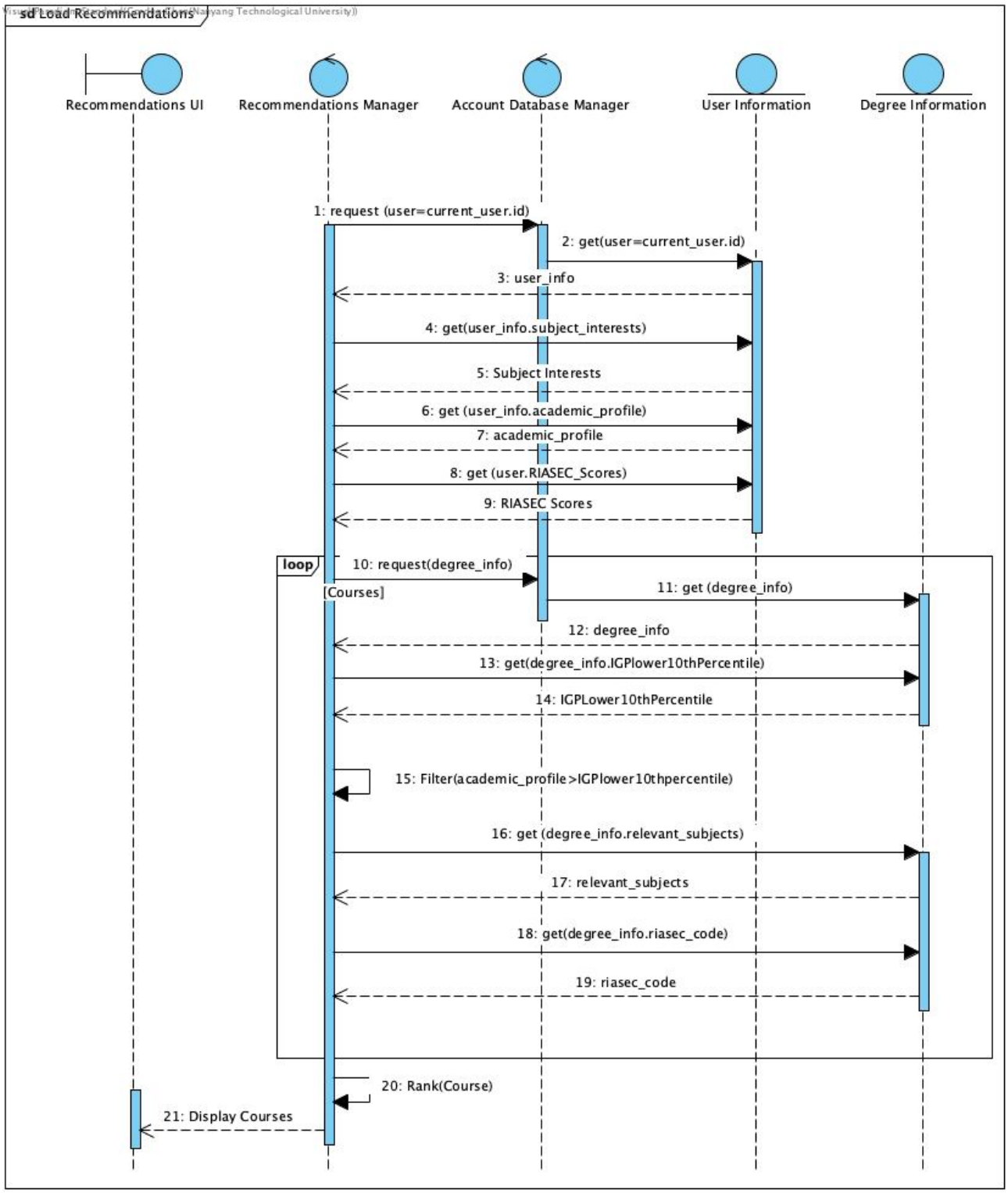


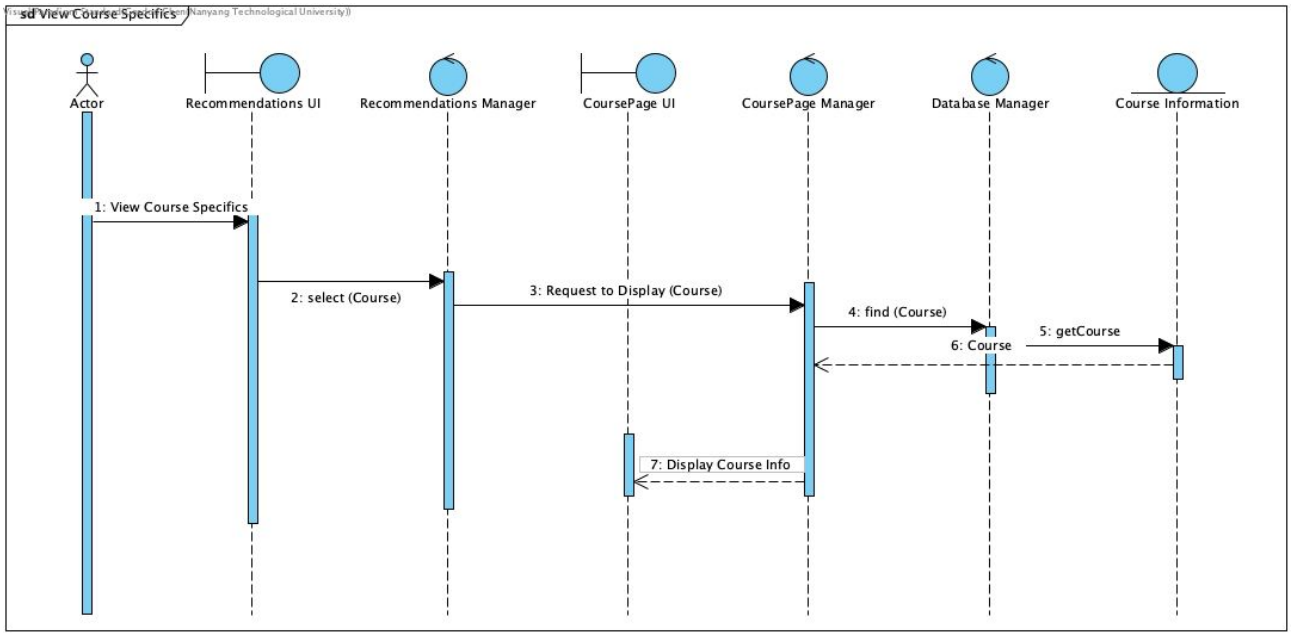

User
Signup UI
Account Manager
Input Manager
PW Manager
Account Database Manager
User Information
Homepage UI
1: Clicks signup button
1.1: ask for user details
2: Input details
2.1: Validate(Input)
3: Validate(pw)
alt
[valid==true]
2.2: valid
3.1: valid
4: searchUserInfo
4.1: getUserProfile
alt
[user not found]
4.1.1.1: Account created
4.1.1: user not found
4.1.1.1.1: Directs user into homepage
4.1.2: user found
4.1.2.1: Account not created
4.1.2.1.1: Prompt user to login
3.2: Invalid details
opt
3.2.1: Display invalid details message
3.2.2: Prompt user to retry


sd Update Academic Qualification/RIASEC Personality Scores
User
UpdateAccount UI
Information Manager
Input Manager
Account Database Manager
User Information
1: Change (Input)
2: validate (input)
3: valid
alt
[valid==true]
4.1.1: searchUser
4.1.2: get (user_info)
4.1.3: User Info
4.1.4: update (input)
4.1.5: Update Successful
4.1.6: Display success message
4.2.1: missing Input
opt
[Missing Input]
4.2.2: Display Missing Input Message


sd Personality Test
Actor
TestPage UI
PersonalityTest UI
PersonalityTest Manager
Input Manager
Account Database Manager
User Information
1: takeTest
3: Redirect
4: Input answers
5: Record (Input)
6: Validate (Input)
alt
[valid==true]
7.1.1: valid
7.1.2: Calculate_RIASEC_Scores (Input)
2: request (User.Info)
7.1.4: getUserInformation(user=current_user.id)
7.1.5: User Profile
ref
7.1.6: update (RIASEC Score)
7.1.8: Test Complete
7.1.7: Update Successful
7.2.1: Invalid Input
7.2.2: Display Missing Input Message


sd Fill Academic Portfolio/Subject Interests
Actor
RecommendationsUI
AcademicPorfolio UI
Information Manager
Input Manager
Account Database Manager
User Information
1: Input Academic Portfolio/Subject Interests
2: Record (Input)
3: Validate (Input)
alt
4.1.1: valid
[valid==true]
4.1.2: pass (Input)
4.1.4: getUserProfile
4.1.5: User Profile
5.1.6: update (Input)
5.1.7: Update Successful
5.1.8: Academic Info/Subject Interests Submitted
ref
Load Recommendations
5.2.1: Invalid Input
5.2.2: Display Missing Input Message


sd Load Recommendations
Recommendations UI
Recommendations Manager
Account Database Manager
User Information
Degree Information
1: request (user=current_user.id)
2: get(user=current_user.id)
3: user_info
4: get(user_info.subject_interests)
5: Subject Interests
6: get (user_info.academic_profile)
7: academic_profile
8: get (user.RIASEC_Scores)
9: RIASEC Scores
loop
10: request(degree_info)
[Courses]
11: get (degree_info)
12: degree_info
13: get(degree_info.IGPlower10thPercentile)
14: IGPLower10thPercentile
15: Filter(academic_profile>IGPlower10thpercentile)
16: get (degree_info.relevant_subjects)
17: relevant_subjects
18: get(degree_info.riasec_code)
19: riasec_code
20: Rank(Course)
21: Display Courses


sd View Course Specifics
Actor
Recommendations UI
Recommendations Manager
CoursePage UI
CoursePage Manager
Database Manager
Course Information
1: View Course Specifics
2: select (Course)
3: Request to Display (Course)
4: find (Course)
5: getCourse
6: Course
7: Display Course Info

Appendix B: Analysis Models

B1: Black box testing

B1.1.1. Sign up for account

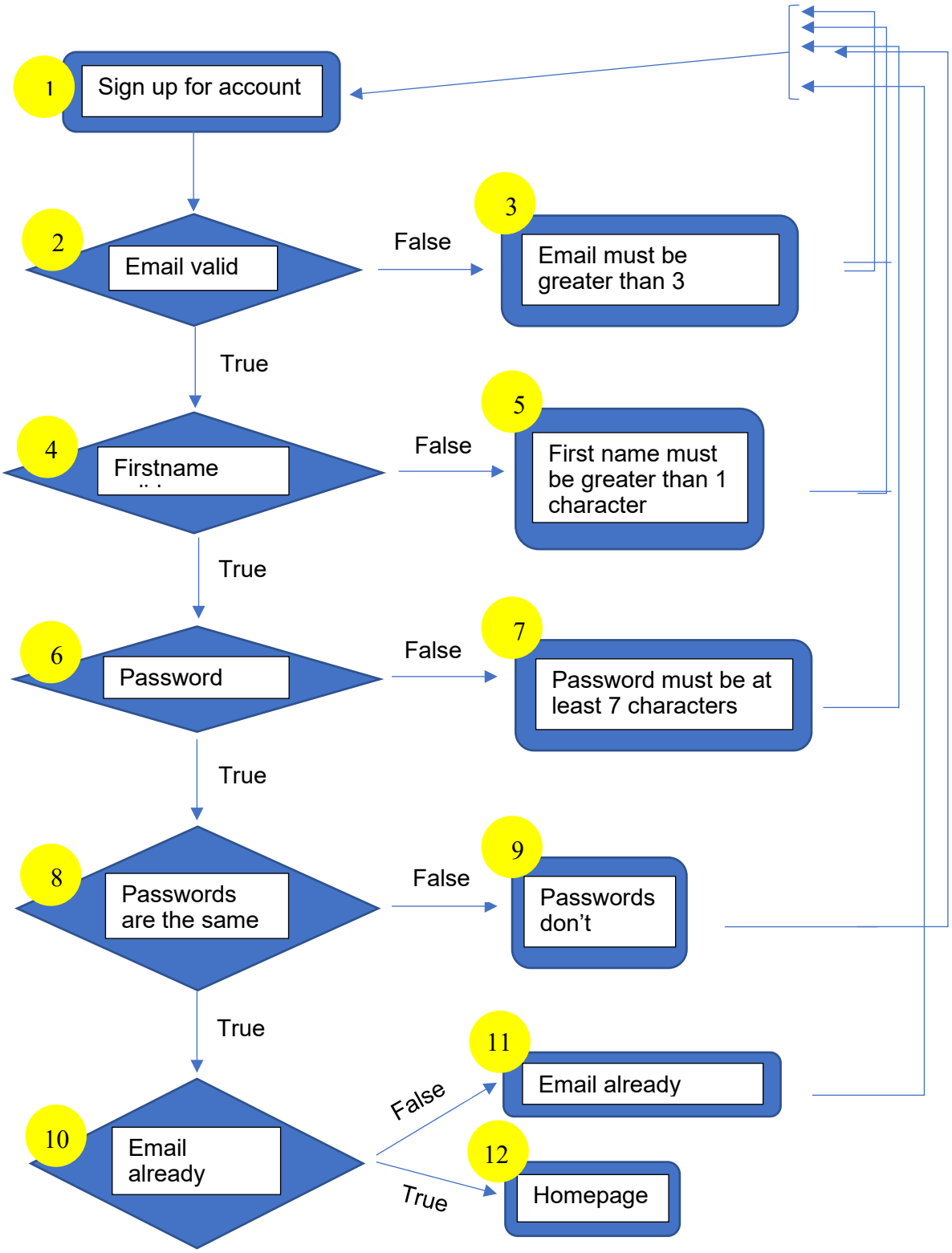
User Email	First Name	Password	Confirm Password	Expected Result (Oracle)	Test Result (Log)
user1@email.com	User	Password1		Passwords don't match	Pass
user1@email.com	User		Password1	Passwords don't match	Pass
user1@email.com		Password1	Password1	First name must be greater than 1 character	Pass
	User	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



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

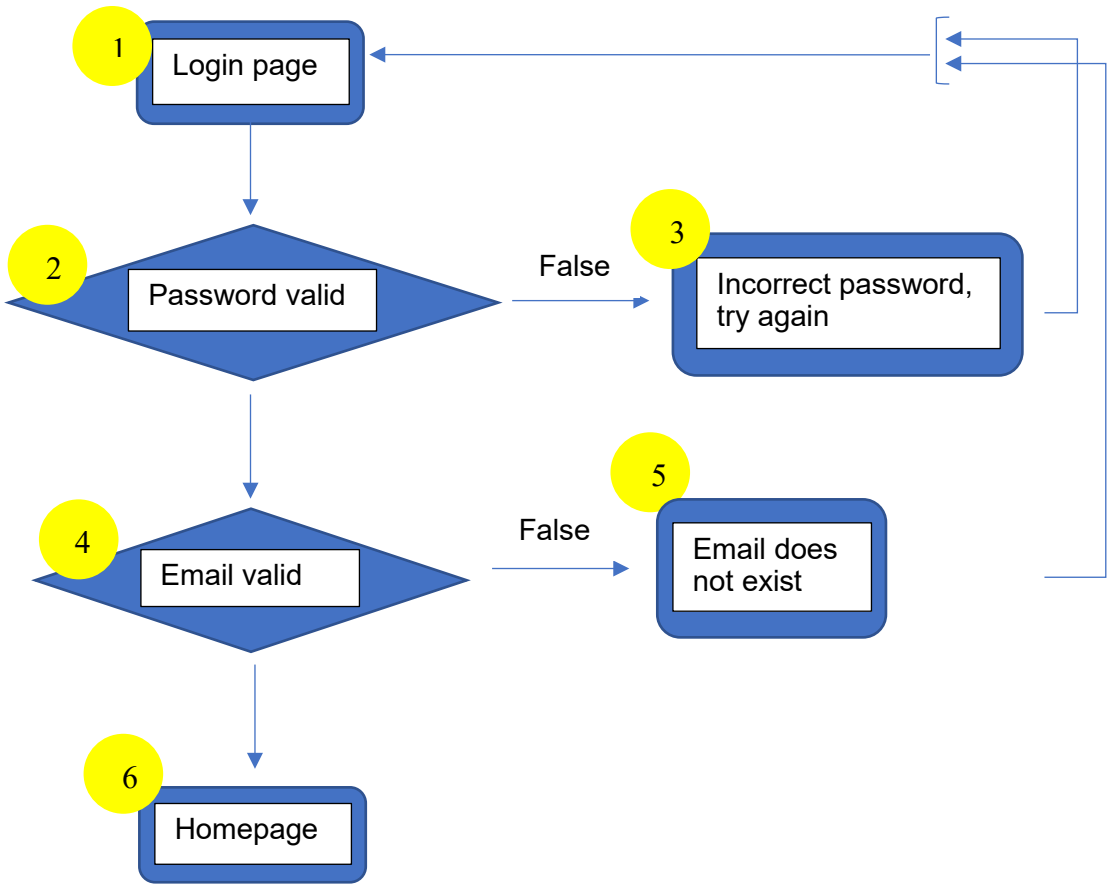
Cyclomatic Complexity:

CC_1 = |edges| - |nodes| + 2 = 16 – 12 + 2 = 6

CC_2 = |decisionpoint| + 1 = 5 + 1 = 6

➔ $CC_1 = CC_2 := CC_{B1.2.1.}$, hence all decision points are binary

B2.2: Login



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:
CC_3 = |edges| - |nodes| + 2 = 7 – 6 + 2 = 3
CC_4 = |decisionpoint| + 1 = 2 + 1 = 3
➔ $CC_3 = CC_4 := CC_{B1.2.2.}$, 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