

Abstract

This One-Stop Student System aims to solve common challenges faced by students such as browser cluttering and memory usage, physical health concerns, time management issues, lack of motivation and goal setting, information access and communication barriers problem. This system provides a comprehensive study system that simplifies and enhances the student experience by integrating academic, health and personal management tools into a single interface. The system scope is the three main modules, study, health and personal categories, each of the categories containing multiple features to meet a wide range of student needs and expectations. For instance, the study module consists of note taking, grade calculator, measurement converter, video teaching pronunciation and calendar reminder function for academic support, the health module offers BMI calculator, calorie calculator and exercise workout schedule for health monitoring and the personal module include to-do list, goal getter and AI chatbot to assist with various daily tasks.

The Feature-Driven Development (Agile methodology) is adopted throughout the development process to ensure frequent progress updates, early feedback and adaptability to changing requirements. Thorough testing is conducted to ensure the system's usability, reliability and functionality to ensure that the system is user-friendly and well-functioning.

The four objectives of the system have been achieved with all the functions provided in the system. However, the system also highlighted some potential areas for future improvement, such as adding a game module to help students relax and integrating VR technology to allow students to explore the campus environment virtually which are outside the current project scope. Overall, the One-Stop Student System provides students with convenient access to a wide range of study, health and personal category functions which ease the students' study journey.

Table of Contents

Abstract.....	i
Table of Contents	ii
1 Introduction	2
1.1 Problem Statement.....	3
1.2 Objectives	6
1.2.1 To Allow Predict Next Semester CGPA Grade to Motivate to Achieve Better Results	6
1.2.2 To Identify Who are at Health Risk at Earlier Stage	6
1.2.3 To Improve Pronunciation Accuracy and Skills	6
1.2.4 To Simplify the Access in Functions, Reduce Navigation and Learning Time	7
1.3 Benefits	7
1.3.1 To reduce stressness level, avoid browser crashing and enhance efficiency by simplifying multiple platform.....	7
1.3.2 To improve academic performance by having clear motivation	8
1.3.3 To prioritise the tasks effectively by applying Eisenhower Matrix	8
1.3.4 To enhance in personal health and physical condition.....	9
1.3.5 To improve language learning	9
1.3.6 Ease of school information or query retrieval process	9
1.4 Solution.....	9
1.4.1 Study Category	10
1.4.2 Health Category	10
1.4.3 Persona Category	11
1.5 Competitor	12
1.5.1 Sub-similar functions	12
1.6 Project Plan	13
1.6.1 Proposed Development Model / Research Method.....	13
1.6.2 Role.....	14
1.6.3 Project Schedule	15
1.7 Software Requirement	16
1.8 Chapter Summary and Evaluation	17
2 Literature Review.....	19
2.1 Literature Review	19
2.1.1 Review of Existing Similar System	28
2.2 Feasibility Study	30
2.2.1 Technical Feasibility	30
2.2.2 Economical Feasibility	30
2.2.3 Legal Feasibility	30
2.2.4 Operational Feasibility.....	30
2.2.5 Schedule Feasibility	31
2.3 Chapter Summary and Evaluation	31

3 Methodology and Requirements Analysis.....	33
3.1 Methodology	33
3.1.1 Agile methodology (Feature-Driven Development)	33
3.2 Requirements Gathering Technique	34
3.2.1 Online Questionnaire	34
3.3 Requirements Analysis	40
3.3.1 Functional Requirements	40
3.3.2 Non-functional Requirements	43
3.3.3 Critical evaluation.....	44
3.4 Development Environment.....	50
3.4.1 Hardware Requirement	50
3.4.2 Software Requirement	50
3.5 Operational Environment.....	51
3.5.1 Hardware Requirement	51
3.5.2 Software Requirement	51
3.6 Chapter Summary and Evaluation	51
4 System Design	53
4.1 Use Case Model	54
4.1.1 Use Case Description Table.....	58
4.2 Activity Diagram	87
4.3 Entity Relationship Diagram	118
4.2 User Interface Flow	119
4.3 Chapter Summary and Evaluation	129
5 Implementation and Testing	131
5.1 Implementation	131
5.1.1 Code Snippets	132
5.2 Testing	172
5.2.1 Test case	172
5.3 Chapter Summary and Evaluation	264
6 Discussions and Conclusion.....	266
6.1 Summary	266
6.2 Achievements.....	267
6.3 Contributions.....	268
6.4 Limitations and Future Improvements	269
6.5 Issues and Solutions	270
References	271
Appendices.....	275

Chapter 1

Introduction

1 Introduction

In order to solve the cluttering of browsers, anxiety of recalling multiple passwords when an account is needed for different functions, stressness when opening multiple web pages and so on, this one-stop student system provides a comprehensive study system which includes 3 categories. The study category integrates a range of academic tools such as note-taking, video pronunciation teaching, GPA calculator, measurement converters and calendar reminder to make students foster the interest in study via easily accessible functions. In the health category, students are able to know the BMI and calories value is important to maintaining a healthy lifestyle by adjusting the daily eating habit and setting an exercise workout schedule for regular physical activity. Hence, it could make gradual adjustments to students' daily routine in a better way. Besides, in the personal category, this system will be able to help students to prioritise tasks and set goals as a motivation. New students are also able to ask for the chatbot about any query efficiency. With all these features, it is able to foster a positive and successful study journey, with a balance between academic and personal life in university. In this chapter, the current problem faced by university students, the objective of the system, the benefits of the system, solutions that solve the problem, competitor and project plan for this project will be covered.

1.1 Problem Statement

Students face challenges when accessing multiple web pages for different functions. This way not only clutters the Internet Browser but also leads to increased memory usage. Students may need to memorise multiple passwords for accessing different services platforms, thus adopting insecure password practices such as password reuse is a high possibility (Woods & Siponen, 2018). Biometric authentication such as fingerprint, facial, voice, retina recognitions are easy methods to login but are not widely found in web applications. Students will be anxious about the memories' capabilities to recall passwords (Woods & Siponen, 2024). A study identified pressures to close tabs, such as limited attention and screen space, and pressures to maintain open tabs (Chang et al., 2021 as cited in Ma et al., 2023). The clutter-related stress increases when students keep a large number of tabs and windows open. Exceeding the user's tab limit also decreases work efficiency, causes distraction, leads to computer crashes, induces feelings of stress, anxiety, annoyance and so on (Ma et al., 2023). Feeling of frustration and impatience will be raised from slow browser performance. In the interview conducted by Chang et al. (2021), participants expressed several negative feelings associated with the accumulation of numerous browser tabs. The interactive elements like Ads and pop-up windows when students need to use a function in a website will also cause students to be stressed. Thus, with this one-stop system containing functions such as note taking, measurement converter without any annoying Ads will be the best.

Physical education is not a mandatory requirement for graduation among university students, excluding students enrolled in sports-related courses. In recent years, the overall physical health of students is not as good as that of secondary school students, and the reports show a rise in illness and abnormal psychological state among students (Wang, 2020). Sedentary lifestyle of students due to not having a structured and proper schedule for exercise activities and without a fixed time slot for physical activity. According to Castro et al. (2020), most university students engage in higher levels of self-reported sedentary time compared to the general young adult population. Students' sedentary behaviour levels are similar to desk-based office workers such as students seated in lectures, tutorial and practical classes, library and at home for hours at a time (Moulin & Irwin, 2017). This is impacting students' overall health and wellness. Besides, students who do not focus on BMI and calories have the possibility of leading to being overweight by unknowingly adopting unhealthy eating habits. On top of that, calorie tracker and calculator, body mass index (BMI) calculator and exercise schedule functions are implemented in this system.

Besides, prioritising the tasks effectively can be challenging, especially considering the relative urgency and importance of each (Kennedy & Porter, 2022). University students find it challenging to manage academic responsibilities. Students often struggle to prioritise tasks when multiple assignments, practicals, and tests come together at one time, sometimes it will lead to procrastination and time wastage. Balancing coursework, exam preparation, part-time job and personal life can be overwhelming for students (Nicklin et al., 2018). Moreover, students find it hard to balance academic and personal life which requires effective organisation and time management skills. Students have challenges in managing multiple tasks, meeting deadlines for academic tasks, procrastination and maintaining study-life balance (Soares et al., 2023). Students find it difficult to plan what to do next and always struggle and rush to finish the practical or assignment. According to Adams & Blair (2019), students will be procrastinating on tasks students do not like but must be done. Hence, to-do list and deadline reminder features are a solution of the above mentioned.

Students have a lack of motivation and clear goals for lives and academic performance. At any stage of a student's life in university can experience a loss of motivation, some may experience it earlier, while some may experience it in the middle or even at the very end of university life (Kamal & Baqui, 2022). Lack of motivation factors can cause students to have poor academic performance (Mauliya et al., 2020). Based on Clark et al. (2017), asking students to set goals for the number of past year papers students completed can increase students' performance. For example, students can set a goal and strive for a better grade after predicting students' results. Therefore, a grade point average (GPA) calculator with predictive grades function letting students know the aiming grade of each subject to reach target Cumulative Grade Point Average (CGPA) provided the motivation to study and a goal getter function letting students to jot down grade goals are important.

New students always have some questions about school such as where to find past year papers and do not fully know about TAR UMT culture and background, and may not be willing to search the official website and find the information needed one by one. Besides, some query or information is not readily available online but new students need to ask from seniors or school authorities. In order to address this issue, an AI chatbot function is implemented.

Students may have difficulties in pronouncing the English words when communicating with colleagues or lecturers. Pronunciations are mainly the problems that occur with the students whose native language is other than English (Toçi, 2020). If a student's pronunciation is not clear, it can be very difficult to understand what he or she speaks (Antaris & Omolu, 2020). Thus, integrating YouTube API in video teaching pronunciation function can help students to pronounce a word correctly.

1.2 Objectives

1.2.1 To Allow Predict Next Semester CGPA Grade to Motivate to Achieve Better Results

GPA calculator allows the students to specify the target CGPA and will suggest the possible grades in each course that is required to realise the target CGPA based on the subject that will be taken for the coming semester (Azubuike Ezenwoke et al., 2018). The result can be an indicator for students' coming academic performance and this can motivate students to work harder in order to achieve good results and at the same time to be graduated on time (Ismail et al., 2016). Several formulas will be used to predict the CGPA and will be implemented in the GPA calculator.

1.2.2 To Identify Who are at Health Risk at Earlier Stage

Body mass index (BMI) is a measurement based on a person's height and weight, allowing the classification of individuals into categories such as obese or overweight (Khanna et al., 2022). Addressing classification of obesity or underweight at earlier stage can be possible to prevent or reduce the risk of negative health consequences such as diabetes, cancer, hypercholesterolemia, and other chronic diseases, thereby improve overall health and well-being (Sable et al., 2023). Hence, BMI serves as a prediction method for health and disease later in life (Khanna et al., 2022).

1.2.3 To Improve Pronunciation Accuracy and Skills

According to Simanullang (2018), using videos has a significant effect on the students' pronunciation accuracy. Based on the research conclusion, the researcher suggests using videos in dealing particularly with English Phonology and Pronunciation (Simanullang, 2018). Moreover, based on the research conducted by Rachmawati & Cahyani (2020), the use of YouTube videos have a positive impact on students' ability to pronounce words correctly. Hence, YouTube can be one of the influential English pronunciation learning media for students (Rachmawati & Cahyani, 2020). Hence, integrated YouTube API into video teaching pronunciation function is able to improve English pronunciation accuracy. Students can input the word that confuses pronunciation and the video teaching about pronunciation will be shown.

1.2.4 To Simplify the Access in Functions, Reduce Navigation and Learning Time

This system simplifies task completion and provides a one-stop access for various functions. From the study category, to health category then to personal category, there are many functions contained in each category. Students do not need to open multiple tabs or applications to use specific functions. Students can find and use each function quickly in one system, and this will reduce the navigation time by eliminating the need to search for functions individually through websites or bookmarks in the web browser. Moreover, functions within this system will have a consistent user interface such as similar position of navigation list or button position. It will be able to improve the usability and reduce learning time because students do not need to adapt to different layouts and the system is easy to use. It can be accessed by the number of errors made and speed of operation when the user uses the system without training and uses the system after training 2 days. Robustness which is when a major error happens, whether the user can continue to the next process or the system will stop or hang there and recoverability which is how good is the system at recovering from user errors could be used to measure usability as well.

1.3 Benefits

1.3.1 To reduce stressness level, avoid browser crashing and enhance efficiency by simplifying multiple platform

The one-stop student system enhances the efficiency in accessing different academic tools such as timetables, unit converter and note taking functions through multiple platforms. It eliminates the need of students to login to TARC App or TAR UMT Intranet for seeing timetables, converter apps or online unit converter to convert any unit and so on with this system. It is able to reduce browser clutter when multitasking using different web pages or leave tabs unrelated open without thinking about it. Then, one of the problems of opening multiple tabs, drains processing power causing browsers to slow down, will be reduced (Chang et al., 2021). The stressness level caused by slow browser performance will be reduced too to a certain extent. Besides, anxiety about the memories' capabilities to recall passwords for multiple accounts to access the specific functions can be reduced as well (Woods & Siponen, 2024). Hence, a one-stop student system with various functions allowing students to open only what functions needed, avoiding browser crash and reducing the stress for students to recall passwords and stress from hindering students to complete tasks using specific functions. It can help students stay focused on tasks, eventually improving students productivity.

1.3.2 To improve academic performance by having clear motivation

Predicting a student's grades can enhance students' performance (Awotunde et al., 2024). The CGPA Prediction function assists students in predicting the grades students need to strive for in the upcoming subjects to achieve the targeted CGPA. This can provide students a sense of motivation and direction, knowing the specific grades students need to obtain to reach the academic grade. Thus, realistic goals can be set. Goal setting is able to motivate students to work harder and achieve better outcomes (Clark et al., 2020). Students put efforts on particular subjects to strive for a goal. It serves as a proactive planning tool, allowing students to tailor students' study methods, allocate time and effort more effectively for each subject, and prioritise subjects strategically to meet the academic objectives. A goal getter with a countdown timer allows students to record down the long-term or short-term goal such as completing past year papers or revising specific subjects and to set the time frames for assignments, past year papers, or subject revisions.

1.3.3 To prioritise the tasks effectively by applying Eisenhower Matrix

The to-do list apply Eisenhower Matrix, as known as Urgent-Important Matrix helps students to prioritise tasks by urgency and importance, sorting out less urgent and important tasks which one should either do first, schedule a time for it, delegate or not do at all (Panchali, 2021). It helps students manage time effectively, students can focus on tasks that align with the long-term goals which are quadrants I and II and to minimise time spent on tasks that are not contributing to those goals which are quadrants III and IV. Deadline reminder features will help to remind students that some task will be due soon to avoid late submission. Students can make decisions about how to allocate time and resources more effectively and avoid procrastination in managing multiple tasks, strive to meet deadlines for academic tasks and maintain study-life balance (Soares et al., 2023).

1.3.4 To enhance in personal health and physical condition

The calorie calculator helps students to know what calorie students should eat in order to reach the target weight in a few choices of duration to gain or lose weight. It also provides personalised nutrition. Students can monitor and adjust calorie intake of protein, fat and carbohydrate accordingly based on the recommended calorie intake. Besides, the BMI calculator allows students to track changes in BMI over time. It gives a quick assessment of whether a student weight falls within a healthy range or high health risk in severe thinness or obsessed class III. Students are able to know the amount of weight needed to gain or loss and set goals based on BMI recommendations. Besides, monitoring BMI and calories can contribute to the prevention and management of health conditions such as obesity, diabetes, and cardiovascular diseases. Then, the exercise schedule facilitates structured exercise routines for improved fitness. Seeing exercise progress and completing scheduled workouts can boost motivation. It significantly improves students overall well-being by knowing the health range category, promoting healthy eating according to calorie, and goal-oriented progress tracking.

1.3.5 To improve language learning

The video teaching pronunciation function integrated with YouTube API helps students learn how to pronounce words correctly by watching the teaching video. It exposes students to accurate pronunciation easily. This has effectively addressed the issue of students struggling to grasp accurate pronunciation and decrease the miscommunication among students.

1.3.6 Ease of school information or query retrieval process

AI chatbot allows students to quickly and easily access information about the sports venue, canteen, school culture and services of the school without navigating various websites and searching multiple sources. It also offers immediate responses, enabling students to get the information needed quickly. For new students unfamiliar with the school's culture and background, it can guide students by providing insights into campus life, culture, and available services. This personalised chatbot enhances the students' understanding of the school's services and amenities.

1.4 Solution

One-stop student system is a comprehensive system for students, reducing the use of multiple content web pages to streamline the student experience. It consists of 3 main categories which are study, health, and personal.

1.4.1 Study Category

The study category consists of

- A. Note-taking: This feature allows students input, add attachment, save and share notes.
- B. Video teaching pronunciation: This function integrating with the YouTube API enables students to access video tutorials for accurate pronunciation guidance. Students can input any word and the system will output the related video tutorial.
- C. GPA calculator: The calculation will be separated on before and after July 2023 intake due to introducing new grades. It allows input of target CGPA, total credit hours, next semester subjects' credit hours and so on to predict the next semester subject grades in order to achieve the target CGPA.
- D. Calendar reminder: Students can input the event details and the system will send notification via Google Calendar and email as a reminder before the 30 minutes event starts or anytime preferable.
- E. Measurement converters: The conversion covers various units such as length, temperature, numbers, case and so on. It allows students to select notation and decimal places to display the result. Besides, a base calculator function is also included.

1.4.2 Health Category

The health category consists of

- A. Calorie tracker and calculator: It will calculate calorie need to lose or gain or maintain based on target weight such as mild weight gain (calorie for gaining 0.25 kg per week) or extreme weight loss (losing 1 kg per week). It will also calculate calories of target weight and whether to increase or decrease daily caloric intake, as well as number of days required to achieve the desired weight change based on gaining or losing 0.25, 0.5 or 1 kg per week. It will also provide the recommended calorie intake, students can monitor the weight progress over the next few weeks and adjust the calorie intake accordingly based on how quickly students lose or gain weight. It also provides optimal macronutrient distribution by suggesting protein, fat and carbohydrate daily intake ranges, ensuring a balanced and sustainable dietary plan.
- B. Body mass index (BMI) calculator: It will calculate the BMI and save the BMI over time, allowing students to track the BMI from time to time.
- C. Exercise schedule: Students can input the workout activities on a particular date and time, then the system will provide suggested video links for each exercise such as tips

and techniques. Besides, it includes a function for students to search for videos related to specific exercise concerns, unfamiliar exercise details or fitness goals by inputting keywords or body parts. Additionally, it provided a 30-day fitness challenge with workout details for students to mark for the daily workout completion, rest days and view some workout video links.

- D. A fixed schedule for sports and knowing calories and BMI can help students focus on personal health and physical condition.

1.4.3 Persona Category

Personal category features consists of

- A. Personal profile: It includes some students personal details such as name, age, pictures and timetable schedule. To access this feature, students need to register and login to the system. While students logout the account, the personal profile will be temporarily disabled for privacy and security.
- B. To-do list: It consists of 4 categories based on urgent and important that apply to The Eisenhower Matrix to let students input the tasks in different categories to make easy decisions on what task to do first. A timer countdown function is included for students to set the duration to complete the task.
- C. Goal getter: Students can input the goals and strive to achieve the goals. A timer countdown function is included for students to set the duration to complete the small goal like completing one past year paper.
- D. AI chatbot: It assists new students by answering common questions. The chatbot provides relevant information based on the input given.

Overall, this system will provide students with a balanced and convenient lifestyle.

1.5 Competitor

- A. iStudies is a system that helps to organise academic schedules and keep up with all activities. It includes records of daily schedules and tasks in a calendar, GPA calculator, to-do list and so on. The UI of iStudies is more colourful and interesting.
- B. BMI calculator in calculator.net will calculate the BMI and determine the BMI range such as underweight, normal weight, overweight, or obese. It provides quick calculations. However, it could be improved by suggesting the weight needed to lose or gain instead of just telling the BMI figure that might not be understood by students. Therefore, this improvement will be implemented in this system.
- C. Calorie calculator in calculator.net will calculate the calorie and show results for how many calories to consume each day to maintain, lose or gain weight at a chosen rate such as 0.25 kg per week, 0.5 kg per week and 1 kg per week.
- D. Notion is an all-in-one app that contains a note-taking, event reminder, workout schedule, to-do list and so on. However, its interface is complex for the novice user because it consists of many content block types and most functions are costly.

1.5.1 Sub-similar functions

- A. Studious is able to let students organise study timetable schedules and record down the homework details with different colour separation. However, it has very basic features and many more features could be added to it such as to-do-list and note-taking. Hence, to-do-list implemented with The Eisenhower Matrix and reminder features will be implemented in this system. Besides, a note-taking function could be added to jot down notes with interactive features of adjusting colour of note.
- B. Todoist is an app to help students keep track of all the important tasks and projects. The user interface is simple and easy for beginners to use. However, jotting down many tasks is not efficient yet but just leads to the headache of having so many tasks to be completed, students find it difficult to plan what to do next. Hence, a to-do-list implemented with The Eisenhower Matrix will be implemented in this system.
- C. According to Jslim (2011), the person has published a CGPA calculator for Tunku Abdul Rahman College students in GitHub, with input grades and credit hours to calculate GPA. However, it is outdated and has no separate calculation for new grades students. Hence, this system will take this into consideration and provide a GPA calculator by predicting students grades. Besides, a goal getter function can be included to let students create the goal after knowing the expected grades.

- D. NASM's Calorie Calculator will calculate the macronutrient ratio recommendations to fit users needs. This system will combine this function with the function of calorie calculator in calculator.net together as an enhancement.
- E. California State University, San Bernardino (CSUSB) has a chatbot, Cody, to answer common questions such as campus, orientation, services, departments, links for campus maps, calendar and so on. It is convenient for accessing campus information quickly and able to provide answers that go far beyond department-specific questions. However, there is no such chatbot on the TAR UMT website.
- F. UnitConverters.net provided many types of unit converter and a basic calculator. However, it will output many decimal numbers for the results and be hard for students to readily use the results. Hence, this system will be improved by letting students choose different notation and preferred decimal places to display the results.
- G. Moreover, students are more likely to use Microsoft Excel or any online tools or even write it out on a paper to plan the workout schedule using table property. However, it could be enhanced by capturing user input and providing relevant video links for exercise activities.

1.6 Project Plan

1.6.1 Proposed Development Model / Research Method

This proposed system adopts Agile methodology.

- A. Iterative Development: Agile focuses on iterative development, where software is developed incrementally, part by part in small, manageable chunks. Each category which is the study, health and personal category of the system is considered as an iteration. Solutions for each category are broken down into features. Features of each category will be developed incrementally within a period of time, usually two to three weeks which is called a sprint to get frequent improvement or feedback from stakeholders.
- B. User Involvement: Agile encourages regular communication and feedback loops, continuous collaboration and interaction between stakeholders. It will gather comments from students about features, usability and user experience (UX) aspects throughout the development stages of the system to maintain efficiency and effectiveness. Thus, less resources are required and cost effective due to decreased rework.
- C. Adaptability to Change: Agile embraces changes in requirements in every stage of software development life cycle (SDLC) even. New features may need to be added and

existing features may need enhancements based on user feedback. It adapts to these changes effectively and ensures that the system is what the user wants.

- D. Continuous Delivery and Continuous Integration (CI/CD): Agile encourages continuous integration and delivery practices, where code is integrated frequently and enables rapid release to users in small increments. Thus, users no longer need to wait longer to get the functionality.
- E. Emphasis on Working Software: Agile only delivers the working software to users. Each iteration of this system should deliver a set of fully functional features that students can use immediately and it does not focus only on documentation or planning, it more focuses on coding.

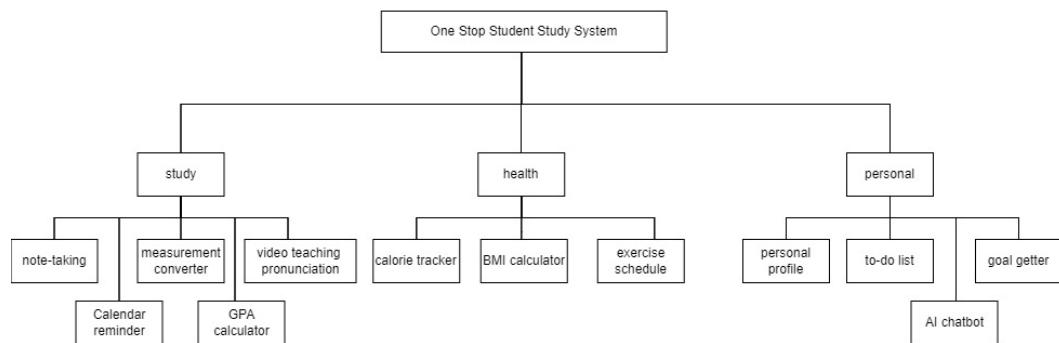


Figure 1.1 : System Structure Chart

1.6.2 Role

- A. Admin is responsible to unlock an account that has been locked due to 3 times of invalid login attempts. Besides, the admin is responsible for managing and maintaining the student database.
- B. All university students will have unrestricted access to all the modules with its features mentioned in section 1.2.

1.6.3 Project Schedule

Title	Description	Timeline - Start	Timeline - End	Dependency
Proposal Idea Development	Come out with few ideas for fyp.	2023-11-30	2023-12-08	
Fill in Form 1, Form 2 and Form 3	Fill out all the personal details and some information required.	2023-12-05	2023-12-05	
Proposal Writing		2023-12-08	2023-12-30	Proposal Idea Development
Project Schedule	Plan for the schedule to develop the proposed system in table form	2023-12-08	2023-12-09	
Problem Statement	Find with journal articles and list down the problems in point form	2023-12-10	2023-12-14	
Solution	Come out with solutions to solve the problems	2023-12-15	2023-12-19	
Target Market	List down for the target audience for this proposed system	2023-12-20	2023-12-20	
Competition/Contribution	Search for the relevant competitor for each function	2023-12-21	2023-12-24	
Proposed Development Model / Research Method	Figure out the methodology used, structure chart and user access role	2023-12-25	2023-12-28	
Abstract	Write down a brief summary for the system	2023-12-29	2023-12-30	
FYP 1 Chapter 1 : Introduction		2024-03-19	2024-03-15	Proposal Writing
Introduction	Brief introduction about the system	2024-02-19	2024-02-20	
Problem Statement	Refine for the problem statement stated in proposal	2024-02-21	2024-02-24	
Objective	State about the objective using SMART	2024-02-25	2024-02-29	
Solution	Refine the solution stated in proposal	2024-03-01	2024-03-03	
Competitor	Refine the competitor stated in proposal	2024-03-04	2024-03-05	
Proposed Development Model / Research Method	Refine the methodology used, structure chart and user access role	2024-03-06	2024-03-08	
Project Schedule	Refine the schedule stated in proposal using Gantt chart and table	2024-03-09	2024-03-12	
Software Requirement	List down the tools that using throughout the project	2024-03-13	2024-03-14	
Chapter Summary and Evaluation	Provide a summary about chapter 1 and the introduction of next chapter	2024-03-15	2024-03-15	
FYP 1 Chapter 2 : Literature Review		2024-03-16	2024-03-29	FYP 1 Chapter 1 : Introduction
Literature Review	Research on past study	2024-03-16	2024-03-25	
Feasibility Study	Mention about feasibility such as technical, schedule and so on	2024-03-26	2024-03-28	
Chapter Summary and Evaluation	Provide a summary about chapter 2 and the introduction of next chapter	2024-03-29	2024-03-29	
FYP 1 Chapter 3 : Methodology & Requirement Analysis		2024-03-30	2024-04-17	FYP 1 Chapter 2 : Literature Review
Methodology	Explain about the methodology planned to use in this project	2024-03-30	2024-04-03	
Requirements Gathering Technique	Explain for the requirement elicitation technique planned to use in this project	2024-04-04	2024-04-07	
Requirements Analysis	List down functional and non-functional requirement	2024-04-08	2024-04-10	
Critical Evaluation	Discuss how methodology implement in this system	2024-04-11	2024-04-14	
Development Environment	Discuss the hardware and software required for development environment	2024-04-15	2024-04-15	
Operational Environment	Discuss the hardware and software required for operational environment	2024-04-16	2024-04-16	
Chapter Summary and Evaluation	Provide a summary about chapter 3 and the introduction of next chapter	2024-04-17	2024-04-17	
FYP 2 Chapter 4 : Design		2024-04-18	2024-05-10	FYP 1 Chapter 3 : Methodology & Requirement Analysis
Diagram	Draw diagrams such as use case diagram and activity diagram to illustrate about the system	2024-04-18	2024-04-28	
Database	Design for the database using entity relationship diagram	2024-04-29	2024-05-05	
User Interface and Navigation Design	Design the interface and navigation of the system	2024-05-06	2024-05-10	
Chapter Summary and Evaluation	Provide a summary about chapter 4 and the introduction of next chapter	2024-05-10	2024-05-10	
FYP 2 Chapter 5 : Development & Testing		2024-07-01	2024-08-04	FYP 2 Chapter 4 : Design
Implementation	Provide description of implementation (code)	2024-07-01	2024-07-27	
Testing	Prepare test cases	2024-07-28	2024-08-03	
Chapter Summary and Evaluation	Provide a summary about chapter 5 and the introduction of next chapter	2024-08-04	2024-08-04	
FYP 2 Chapter 6 Discussions and Conclusion		2024-08-05	2024-08-18	FYP 2 Chapter 5 : Development & Testing
Summary	Summarize the project	2024-08-05	2024-08-07	
Achievements	Evaluate the projects achievement	2024-08-08	2024-08-10	
Contributions	Discuss the creativity, innovativeness, contribution of the proposed system	2024-08-11	2024-08-13	
Limitations and Future Improvements	Identify the limitations of the project	2024-08-14	2024-08-16	
Issues and Solutions	Describe the various problems faced during project development	2024-08-17	2024-08-18	

Table 1.1: Project timeline



Figure 1.2: Gantt Chart for Project Timeline

1.7 Software Requirement

This project is completed using a laptop.

In the planning stage and design stage, Google Docs is used to do the proposal. Visual Paradigm is used to do the structure chart, class diagram, entity relationship diagram and so on. Monday's website is used to do the gantt chart of project timeline. Then, Microsoft Excel is used to modify the export data. Canva, webflow and figma are used to create the user interface flow in the design stage.

In the analysis stage, the journal articles are found mainly from reliable web journal database websites, which are IEEE explore and Google Scholar.

In the development stage, Visual Studio is used in order to code with Hypertext Markup Language(HTML), Cascading Style Sheets (CSS), Javascript (JS), C-Sharp (C#) and Python. The AI chatbot can be developed by giving a dataset to train the chatbot. SQL Server Express LocalDB will be used to design the database.

1.8 Chapter Summary and Evaluation

This chapter has addressed the problem currently faced by students such as students having difficulties when accessing multiple web pages leading to recalling pressure, computer crashes and frustration and several problems, highlighting the need for the 3 categories solutions which are study, health and personal categories. It also introduces the objectives of the one-stop student system such as to predict expected grades to achieve targets CGPA. The functionalities or solutions that are included in this system will provide several benefits such as reduce stressness, improve academic performance and improve language learning and solve the identified issues. Competitors with the pros, cons and unique are identified based on functions integrated into this system. Agile methodology, system structure chart, role for 3 different users which are admin, university students and software developer and project timeline are included as an early planning to avoid any delaying of the project. The software tools that will be used throughout this project such as Google Docs and Visual Studio is listed down as well.

The next chapter will drive into the literature review, examining the past studies related to the proposed system. Moreover, a feasibility study will be conducted to ensure the viability of the project.

Chapter 2

Literature Review

2 Literature Review

2.1 Literature Review

Students use tabs for different types of tasks. Some functions require a user account for access, students need to recall the passwords for various functions, and using the same password across multiple sites is the common preference (Woods & Siponen, 2024). This practice poses a security risk. If one account's password is leaked, people may access the other accounts and the sensitive information will be exposed. The layout for different websites will be different as well. Besides, students will have productivity and feel in control when students maintain a manageable number of open browser tabs. Based on the interview findings, people have mixed feelings about the browser tab, such as having open tabs serves as a reminder to work on tasks and stay updated on progress (Chang et al., 2021). However, all participants in the interview also expressed experiencing bad emotions and pressure when dealing with the unmanageable number of open tabs which is more than 5 or 8 tabs (Chang et al., 2021). However due to the functionality in each tab, it is hard for students to reduce the number of open tabs. Besides, when students are required to look for any function, students have to search using keywords and sift through various web pages to find the suitable and useful function with the free version. This process is time-consuming and troublesome. Hence, the implementation of this system will address those issues. Students will be able to reduce the needs of multiple tabs and navigate to each function in the system through the navigation bar easily with less navigation time and low learning time in studying different layouts of user interface. Students can easily locate and access various functions in a more organised and efficient way.

Procrastination has always been an issue in university, and academic performance will undoubtedly have a negative impact (Louise et al., 2022). One of the reasons for procrastination is students have difficulties in prioritising the task effectively. Besides, academic procrastination is the cause of failing to meet deadlines within the specific time frame (Louise et al., 2022). Hence, students use different applications or functions to jot down the task students need to do. For example, Studious is an application with minimalist and simple user interface that organises study schedules by adding subjects through the 'Subjects' option, the timetable, homework and examination section can be edited and viewed (Studious - Apps on Google Play, n.d.). Notion also provided functions to take note and list down tasks that need to be done (Home Page | Notion Template, n.d.). Thus, a note-taking function could be added to jot down notes with interactive features of adjusting colour of note.

Moreover, Todoist is an app to help users keep track of all the important tasks and projects. It excels in breaking down large projects into manageable tasks and subtasks which will enhance productivity (TrustRadius, 2023). However, jotting down all tasks is inefficient, merely causing the overwhelming headache of a lengthy to-do list. Consequently, students struggle to effectively prioritise the next steps or tasks. Besides, when clicking the complete task button, the task is directly disappeared, it does not take into consideration of misclicking the button. Hence, this system will let students upload the timetable and jot down the assignment or task based on priority by implementing a to-do list which applies to the Eisenhower Matrix. Besides, the system will make the task become strikethrough if clicking any area of the task, the record is still there and is able to undo the action and a trash button to directly delete the task putting at the end of the task to avoid any misclicking. The Eisenhower matrix is widely used among the methods of time management, which will divide the tasks into 4 degrees of importance such as important and urgent, important and not urgent, not important and urgent, not important and not urgent (Maksymov & Tryus, 2023). For university students who need to balance the academic responsibilities with personal life, time management is the crucial aspect (Calonia et al., 2023). Students will be able to manage the time wisely by organising the tasks according to the Eisenhower Matrix and lead to academic excellence. Besides, a timer countdown will be included to let students to set for the timing without opening another countdown tab.

Predicting future grades to meet target CGPA is a forward-thinking approach. It is because forecast grades needed has become important in education due to the growing interest in identifying the underlying factors that influence academic performance (Zollanvari et al., 2017), and this will encourage students to set academic goals based on predictions. This predictive feature will foster motivation by providing students with a clear academic goal, fostering a sense of direction and purpose in the study journey. iStudiez provides a function to calculate GPA based on a custom GPA scale but needs to be paid (iStudiez Pro for Windows – Best App for Students, n.d.). According to Jslim (2011), the person has published a CGPA calculator for Tunku Abdul Rahman College students in GitHub, but just to input grades and credit hours to calculate CGPA. Hence, this system includes a special function of predicting the student's grades and helping students to work better on the coming subjects. Besides, this system will also separate the calculation due to new grades introduced for students after 2023 July intake. Besides, after knowing a big goal which is to strive for expected grades, a goal getter function will be implemented with a timer counter for students to set the timing to complete a small goal like doing revision everyday for one hour each subject.

GRADE	MARKS RANGE	GRADE POINT
A	80 – 100	4.0000
A-	75 - 79	3.7500
B+	70 - 74	3.5000
B	65 - 69	3.0000
B-	60 - 64	2.7500
C+	55 - 59	2.5000
C	50 - 54	2.0000
F	0 - 49	0.0000

Note: Passing grade is C and above.

Figure 2.1 Grading Scheme Applicable to Year 1 Semester 2 & Semester 3, Year 2 & Year 3 & Year 4 in Academic Year 2023/2024 in TAR UMT

Grade	Mark Range	Grade Point	Description
A+	90 - 100	4.0000	High Distinction
A	80 - 89	4.0000	Distinction
A-	75 - 79	3.6700	Distinction
B+	70 - 74	3.3300	Merit
B	65 - 69	3.0000	Merit
B-	60 - 64	2.6700	Merit
C+	55 - 59	2.3300	Pass
C	50 - 54	2.0000	Pass
F	0 - 49	0.0000	Fail

Note: Passing grade is C and above.

Figure 2.2 Grading Scheme Applicable to July 2023 Year 1 Semester 1 New Intake in Academic Year 2023/2024 and onwards in TAR UMT

Quality Point = Grade Point x Credits of the course

$$\text{GPA} = \frac{\text{Total Quality Points for course(s) with CGPA bearing registered in a Semester}}{\text{Total Credits for all course(s) with CGPA bearing in a Semester}}$$

$$\text{CGPA} = \frac{\text{Total Quality Points for course(s) with CGPA bearing for all Semesters}}{\text{Total Credits for all course(s) with CGPA bearing in all Semesters}}$$

Figure 2.3 Formulas for calculating Quality Point, GPA and CGPA in TAR UMT

Step to predicting subject grades:

- 1) accept input for Current CGPA, Target CGPA, Total Credit Hours Earned, Each Subject's Credit Hour for Next Semester
- 2) Total Quality Points = Current CGPA * Total Credit Hours Earned
- 3) Total Credit Hours for Next Semester = Sum of Credit Hour for All Next Semester Subject
- 4) Total Quality Points Needed for Target CGPA = Target CGPA * (Total Credit Hours Earned + Total Credit Hours for Next Semester)
- 5) Quality Points Needed = Total Quality Points Needed for Target CGPA - Total Quality Points
- 6) Loop For Each Subject for Next Semester
 - i. Grade Point = Quality Points Needed / Each Subject's Credit Hour for Next Semester
 - ii. Subject Grade = Convert Grade Point to Grade
 - iii. Quality Points Needed = Quality Points Needed - (gradePoint * Each Subject's Credit Hour for Next Semester)
- 7) If Quality Points Needed still have after loop through all subject, then it is consider cannot reach the Target CGPA in next semester

Students may require different types of information regarding the university and sometimes getting this information is rather cumbersome and lengthy. To address this issue, California State University, San Bernardino (CSUSB) uses chatbot Cody that will efficiently provide comprehensive campus information beyond department-specific queries. Cody is able to answer common query such as campus, orientation, services, departments, links for campus maps and academic calendar (Apply to CSUSB | Admissions | CSUSB, n.d.). An online chatbot system implemented for college websites uses Artificial Intelligence techniques like Natural Language Processing(NLP), allowing users to communicate with chatbot using natural language input and trains it using Machine Learning(ML) to provide helpful responses (K. et al., 2020). Hence, a TAR UMT chatbot could be implemented in this system to simplify the information retrieval process.

In this system, NLP and ML are used to develop the chatbot. NLP techniques are essential for understanding and processing human language inputs. A dataset is given to the model. First, it has to be preprocessed into clean text before letting the model understand and analyse the data. To clean the data, any capital letters, punctuation, numbers and stop words should be removed. Then, tokenization needs to be done to break down a phrase, sentence, paragraph, or an entire text document into smaller units. Lemmatization is needed to cut words down to base form as well. After that, splitting the dataset into train and test data and training the model can be done. TF-IDF vectorization will be used as an NLP technique to transform the text data into numerical features. Then, it is needed to compare the performance of the ML algorithm and choose the best model to implement as the chatbot. Decision Tree that makes decisions based on feature values, splitting data into subsets to create a tree-like model, Support Vector Machine that separates data points with a linear boundary to maximise the margin between classes, Naive Bayes that classifies text data based on the frequency of words using Bayes' theorem, Random Forest that combines the output of multiple decision trees to reach a single result and Logistic Regression that predicts binary outcomes using linear regression and a logistic function will be used as ML algorithms.

UnitConverters.net provided many types of unit converter and a basic calculator (Unit Converter, n.d.). In this system, this system will offer enhanced customization options, allowing students to select the preferred number notation, such as normal (123.45) or exponential (1.2345E2), as well as specify the desired decimal places for display, which improve readability and usability. A base calculator for add, subtract, multiple and divide numbers of any base will be included as well ($78+58=148$).

BMI is more accurate than the body weight in the measurement of the body overweight that leads to heart disease, high blood pressure and other risks (OuYang, 2018). Adults in the youngest age group between 18 and 24 years old had the highest Odds Ratio (OR) of 4.22 and greatest absolute risk of 37% compared to other ages, which is 24% of transitioning from normal weight to overweight or obesity over a 10-year period. Adults in the youngest age group with overweight or obesity at baseline were also at highest risk to transition to a higher BMI category; OR 4.60 and absolute risk 42% which significantly higher than those without baseline overweight or obesity of 18% risk of transitioning from overweight to class 1 and 2 obesity, While OR 5.87 and absolute risk 22% compared to only 5% for others of transitioning from class 1 and 2 obesity to class 3 obesity (Katsoulis et al., 2021). In short, young adults (18–24 years) face higher risks of weight transitions, especially for those who start with being overweight or obese. The BMI calculator in calculator.net accepts input of age, gender, height, weight and will calculate the BMI and determine the BMI range classification such as underweight, normal weight, overweight, or obese (BMI Calculator, n.d.). In this system, it provides additional functions which will show the picture of the BMI category such as severe thinness, moderate thinness, obese class I and so on for students to have a clear visualisation of the BMI range. Besides, the system will suggest how much weight students should lose or gain to reach a healthy BMI of 24.9 kg/m^2 . It also provides a chart to automatically track the changes of BMI overtime.

BMI can be calculated by ratio of height and weight (kg/m^2) (Misra & Dhurandhar, 2019).

There are several parameters (Golubnitschaja et al., 2021):

- Overweight parameters are as follows:
 - o for females, body mass index (BMI) 25–30
 - o for males, BMI 26–30
- Obesity parameters are as follows:
 - o class I—BMI 30–35 (Obese Class I)

- class II—BMI 35–40 (Obese Class II)
 - class III—BMI > 40 (Obese Class III)
- Underweight parameters are as follows:
- grade 1: females, BMI < 19, males, BMI < 20 (Mild Thinness)
 - grade 2: BMI ≤ 17 (Moderate Thinness)
 - grade 3: BMI ≤ 16 (Severe Thinness)

A low BMI increases the risk for several cancer types, metastatic disease, malnutrition, lower haemoglobin and so on. A high BMI increases the risk for hypertensive heart disease, stroke, hypertension, chronic inflammation and so on.

Reducing caloric intake is important for those at high risk for developing type 2 diabetes (“5. Prevention or Delay of Type 2 Diabetes: Standards of Medical Care in Diabetes—2018,” 2018). Calorie calculator will be able to help to calculate daily calorie intake based on the collected data age, gender, weight, height and activity factor (K.R.C Bandara et al., 2022). The calorie calculator in calculator.net will calculate the calorie and show result for how many calories to consume each day to maintain, lose or gain weight at a chosen rate such as 0.25 kg per week, 0.5 kg per week and 1 kg per week (Calorie Calculator, n.d.). NASM's Calorie Calculator could give the recommendation for macronutrients by inputting target weight (Calorie Intake and Weight Loss Calculator - NASM Bodyweight Tool, n.d.). Hence, this system will merge the 2 functions together by calculating the calorie needed to lose or gain or maintain based on target weight. Besides, calories of target weight and the amount whether to increase or decrease daily caloric intake, as well as number of days required to achieve the desired weight change will be shown as results. It will also provide the recommended calorie intake for students to self-adjust the calorie. An optimal macronutrient distribution is also implemented to ensure a healthy and balanced diet by including the breakdown of consumed calories from fats, carbohydrates, and protein.

Knowing calories value, increasing in doing exercise and knowing caloric consumption intake such as carbohydrates, proteins, fats each day will be able to maintain a healthy weight and help to prevent diseases such as diabetes, heart disease, and various cancers (Osilla et al., 2022). To lose 0.25 kg a week, people need to consume 250 calories below daily calorie requirements (Seng, 2024).

Hence, it can be derived that:

- Mild weight loss 0.25 kg/week : consume 250 calories below daily calorie requirements.
-

- Weight loss 0.5 kg/week : consume 500 calories below daily calorie requirements.
- Extreme weight loss 1 kg/week: consume 1000 calories below daily calorie requirements.

Vice versa for weight gain.

To calculate the daily calorie intake of the body, first is to calculate basal metabolic rate (BMR), Mifflin-St Jeor Equation which is more reliable than Harris Benedict Formula can be used in this calculation (K.R.C Bandara et al., 2022).

Male: $BMR = (9.99 \times \text{weight (kg)}) + (6.25 \times \text{height (cm)}) - (4.92 \times \text{age (y)}) + 5$

Female: $BMR = (9.99 \times \text{weight (kg)}) + (6.25 \times \text{height (cm)}) - (4.92 \times \text{age (y)}) - 161$

- 1.2 for people who are inactive (little or no exercise at all)
- 1.375 for people with mild activity (light work or exercise 1-3 times a week)
- 1.55 for people with moderate activity (work is being or sports 3-5 times a week)
- 1.725 for active people (heavy work or exercise 6-7 times a week)
- 1.9 for people who are extra active (very hard work or jobs that require heavy physical labor)

Figure 2.4: Activity factor

After calculating the BMR, it is multiplied by the activity factor (eg: $BMR * 1.2 =$ total calorie need daily) shown in figure 1.3, to determine the total calorie intake required to maintain the current weight of the user (K.R.C Bandara et al., 2022).

Students can understand physical health after calculating BMI and calorie value. Then, a workout schedule can be written down to further improve health by gaining or reducing or maintaining weight.

Physical activities and exercise are able to maintain physical and psychological health and help the body to respond to the negative effects of several diseases such as diabetes, hypertension, cardiovascular diseases, and respiratory disease (Kaur et al., 2020). The positive attitudes created towards fitness videos could influence positive attitudes towards fitness (Sokolova & Pérez, 2021). Videos that are provided on YouTube could be used to help sustain exercise activities of people that need motivation or social and informational support by reinforcing the motivation (Sokolova & Pérez, 2021). Instead of using Microsoft Excel or any tools to do the workout schedule, this system will not only provide students with an exercise schedule function which can let students input the activities, but also includes a special function which will

capture the user input and output the YouTube video link related to the exercise activities. Students can also directly search for any doubt about exercise and the system will provide the link. A 30-day fitness challenge is also implemented in this system with some interactive and digital features to monitor and mark that challenges have been completed instead of printing and manual tracking or using any annotation tools that students might have messed up due to large numbers of files and pictures.

Accurate pronunciation is crucial if someone recites a word such as ‘live’ that has a different meaning than what is intended (Rachmawati & Cahyani, 2020). Learning and mastering correct pronunciation skills is necessary (Rachmawati & Cahyani, 2020). A result showed that students have a good attitude towards using YouTube as a positive approach to enhance pronunciation skills (Mehmood et al., 2021). The help of YouTube videos will be useful by providing learning videos for students. Hence, a video teaching pronunciation function integrating YouTube API will be included to improve pronunciation accuracy.

This proposed project’s uniqueness is it has provided most of the free functions in one system with a friendly user interface instead of the existing systems that provided limited functions. This system encourages time management and grades prediction which can contribute to academic success. It can also encourage a healthy lifestyle for students so that students will not be tired or lack energy for classes. The chatbot can answer the students’ common doubts about school efficiently. Overall, this system provides a platform that caters to various aspects of student life, contributing to a balanced lifestyle.

2.1.1 Review of Existing Similar System

Similar System	iStudiez	calculator.net	Notion	Proposed system have everything
Note-taking	Not applicable	Not applicable	/ (can jot down note)	/
Calendar Reminder	/ (only sent notification to Google Calendar)	Not applicable	/ (only sent notification to Google Calendar)	/ (will sent notification to Google Calendar and email)
Measurement converter	Not applicable	Not applicable	Not applicable	/
GPA calculator	/ (only can calculate GPA)	Not applicable	Not applicable	/ (able calculate and predict grades)
video teaching pronunciation	Not applicable	Not applicable	Not applicable	/ (able to output a YouTube video teach pronunciation based on user input)
calorie tracker	Not applicable	/ (only can calculate the calorie and show result of calories to consume each day)	Not applicable	/ (provide the recommended calorie intake and will also calculate the macronutrient ratio recommendations)
BMI calculator	Not applicable	/ (only can calculate the BMI and determine the	Not applicable	/ (will suggest how much weight students should lose or gain to

		BMI range classification)		reach a healthy BMI and provides a chart to automatically track the changes)
exercise schedule	Not applicable	Not applicable	/ (have exercise tracker but just check box to check)	/ (after calculate BMI and calorie can set a workout schedule to stick to it and will provided a interactive interface)
personal profile	/	Not applicable	/	/
to-do list	/ (to-do-list but not categories task)	Not applicable	/ (only list down task to do and due date)	/ (to-do-list implemented with The Eisenhower Matrix)
goal getter	Not applicable	Not applicable	/	/ (after grades prediction can set the goal as a motivation)
AI chatbot	Not applicable	Not applicable	/ (normal chatbot)	/ (to solve new students' query)

Table 2.1 : Competitors With Proposed System

2.2 Feasibility Study

2.2.1 Technical Feasibility

Technical feasibility involves the evaluation of the hardware, software, and other technical requirements of this proposed system. Laptop will be used to develop this system.

Multiple software tools will be used to develop the system. The system can be built at all with available free software tools such as visual studio with available programming languages such as HTML, CSS, JS, C# and Python. Besides, Google Docs, Visual Paradigm, Monday's website, Microsoft Excel, Webflow, Figma and Canva will be used in this project.

2.2.2 Economical Feasibility

This proposed system will be cost-effective and can be developed for free. The development costs for designing, coding, and testing the web-based application will be free. Application Programming Interface (API) and software tools that are used in this system will be free-of-charge. No money will be spent for this proposed project.

2.2.3 Legal Feasibility

This system is using legal software and hardware to be developed. Any private and confidential data will not be revealed to unauthorised third parties such as personal details provided by students during registration or any text data inputted will be protected. Besides, all the software components used in this system comply with licensing agreements. There will not be illegal use of software components. Students will be informed through the system about the privacy policy or terms and conditions. This system will also be aware of intellectual property rights such as copyright.

2.2.4 Operational Feasibility

This system will work well in TAR UMT students circle to solve the problem and provide benefits mentioned in the previous chapter. Besides, students can access this web-based system conveniently by using a laptop, computer or desktop. No extra training is needed to use this system because students have experience using laptops. The user-friendly interface allows students to navigate and interact with the system easily without encountering challenges. This system definitely can fulfil and meet the students' requirements and achieve satisfaction.

2.2.5 Schedule Feasibility

A project development milestone is planned at an early stage in order to track the project to prevent any delay. Each stage of the project starting from planning, analysis, design, development and testing will be documented to capture the ongoing progress of the development. The project team (author) will meet with the client (supervisor) weekly to review documentation, get the enhancement and address any problems faced throughout the development process. The project is planned and well-structured.

2.3 Chapter Summary and Evaluation

In this chapter, it has covered the literature review of the system and the methods to implement the functions. The proposed project aims to integrate a wide range of free functions into one system with a user-friendly interface, addressing various aspects of student life. It emphasises time management, grade prediction, healthy lifestyle promotion, efficient information retrieval through a chatbot, and balanced task organisation. Comparing the 12 features between most similar systems such as iStudiez, calculator.net and Notion with this proposed system is covered as well. Besides, the feasibility study assesses the viability of the project from different perspectives such as technical, economical, legal, operational and schedule also included.

Chapter 3

Methodology and Requirements Analysis

3 Methodology and Requirements Analysis

In this chapter, the methodology that will be used to develop the proposed system, the techniques used to gather the requirements from stakeholders and the functional and non-functional requirements will be listed out. Besides, critical evaluation, development environment and operational environment will be covered in following sections.

3.1 Methodology

3.1.1 Agile methodology (Feature-Driven Development)

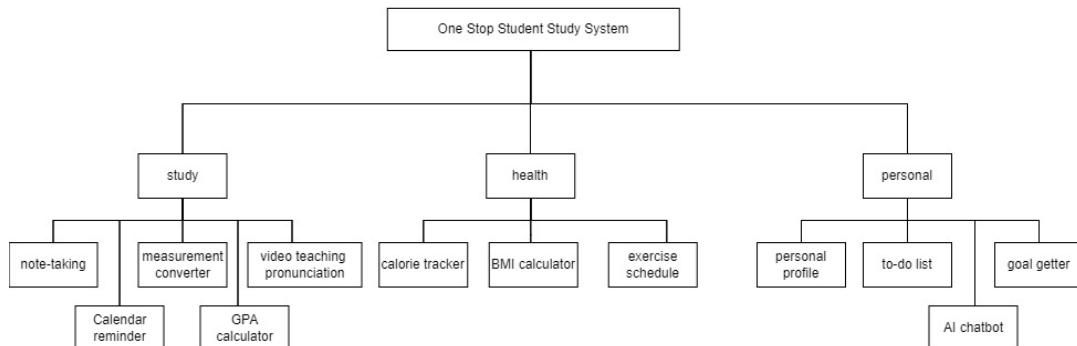


Figure 3.1 Structure Chart of Proposed System

Feature-Driven Development (FDD) is an iterative and incremental software development methodology that focuses on gradually developing features in a series of short, time-boxed development cycles. Unlike other Agile Frameworks like Scrum, FDD is a scalable model that delivers features on a much shorter time frame. For example, Scrum needs a 4-week iteration cycle, but FDD aims to deliver features every 2-10 days. This rapid delivery schedule makes it easier to update the project regularly with supervisor, track and address errors quickly and adapt to supervisor changes and requests. FDD encourages status reporting at all levels, which helps to track progress and results (Lynn, 2021). It relies on documentation to communicate rather than having daily meetings which can be time-consuming. FDD is suitable for large-scale, long-term projects as it enables it to manage and adapt to changing requirements on an ongoing basis. Hence, in this project, there are weekly meetings with the supervisor on Google Meet to discuss documentation and get feedback for improvement.

- A. Focused scope: FDD focuses on the delivery of small features incrementally. It helps to stay focused on a well-defined project's scope, which can help to reduce the risk of scope creep, focus on delivering high-quality features and keep the project on track.
- B. Early and frequent delivery: FDD emphasises the delivery of working software as fast as possible which 2 to 10 days deliver a feature (sprint). This allows the supervisor to see regular progress, receive the early feedback and make necessary changes to ensure that the final system meets the project goals.

- C. Increased flexibility: FDD encourages changing requirements. It allows flexibility in adding, modifying or removing features or requirements. It helps adapt to changing circumstances and deliver a product that meets the needs and expectations.
- D. Improved quality: FDD focuses on quality assurance and testing, which helps to ensure that software is reliable, high quality standards and meets the user needs. Issues can be identified and addressed early to prevent a large problem from happening in later stages.
- E. Enhanced predictability: FDD provides a clear roadmap for development efforts to increase predictability and reduce risk of delays. The clarity of this project helps in planning and developing the features within a short and specific timeframe. Thus, ensure delivery on time.

Overall, FDD keeps development efforts on track and ensures progress within set timeframes by using timeboxed development cycles.

3.2 Requirements Gathering Technique

3.2.1 Online Questionnaire

Due to the target stakeholders being university students, which is a large number of the population, questionnaires will be used as the data collection method. Google Form will be used to do the survey questions and distribute to university students through social media platforms like WhatsApp and Instagram. Questionnaires provide anonymity, students names or details are not required in the questionnaires, allow students to express opinions freely without any fear of judgement. Online surveys also provide convenience because students can complete the questionnaires anytime and anywhere during leisure time.

To provide students with a sense of relaxation before starting the questionnaire, the survey starts by asking for students' age, level of education, and faculty.

The survey questions include below:

1.

Do you use any applications to ease your study, health and personal plaining?
66 responses

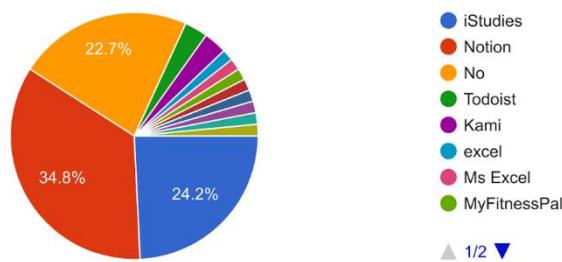
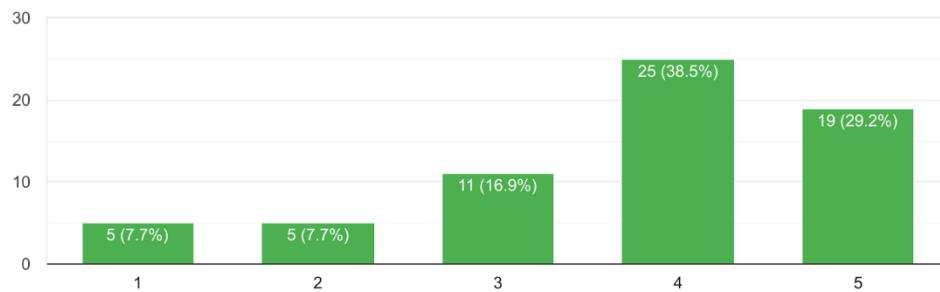


Figure 3.2 : Question 1

This question provides insight into current students' application for organising university study, health and personal life such as Notion, iStudies.

How was your satisfaction in using it to help you during your university life?
65 responses

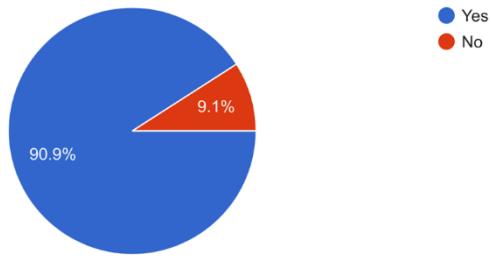


2.

Figure 3.3 : Question 2

This question showed how well these applications meet students' needs and expectations. It can help to identify areas for improvement or determine which applications are most effective and usable in supporting university life.

Do you need a function to predict your next semester grades in achieving target CGPA?
66 responses

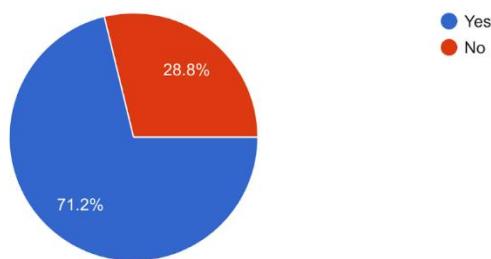


3.

Figure 3.4 : Question 3

This question showed that the majority of the students need a function to predict your next semester grades in achieving target CGPA

Do you face any issue of mispronunciation?
66 responses

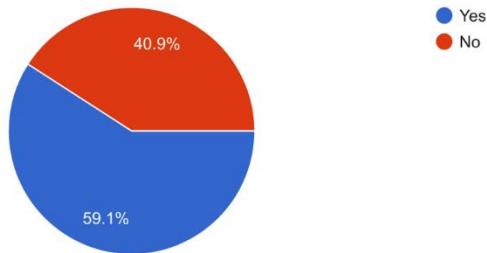


4.

Figure 3.5 : Question 4

This question showed that the students are encountering a problem with mispronunciation, and hence, a pronunciation teaching function could be useful to address this issue.

Do you currently using any BMI or calorie tracker app?
66 responses

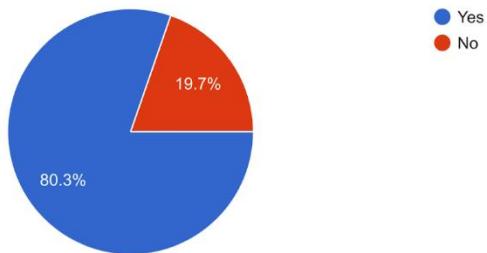


5.

Figure 3.6 : Question 5

This question provides insight such as whether students are taking care of the health and well-being.

Do you using to-do list function to plan your task?
66 responses



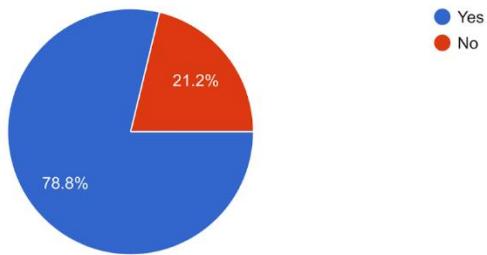
6.

Figure 3.7 : Question 6

This question showed that most of the students use to-do lists to function to plan tasks and thus, to-do list functions could be useful.

Do you set any goal to achieve your aim?

66 responses



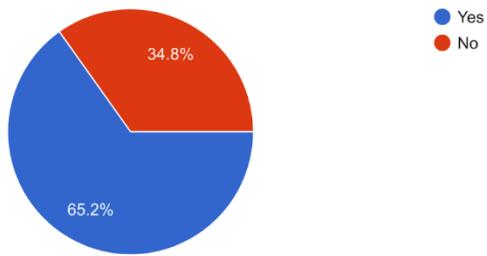
7.

Figure 3.8 : Question 7

This question shows that students would set a goal to strive to achieve the aim, hence, goal getter function could be included in this proposed system.

Do you face any queries about school when you are a freshies?

66 responses



8.

Figure 3.9 : Question 8

This question showed that most students are facing problems, thus, those questions may be solved.

According to previous question, what are the doubts/problems you encountered?

School too big

cannot find classroom!!!

where to find past year

where is eprint

lecture and tutor keep saying find past year to do but where

missing on campus...

where are all those canteen

citc where?

can't find redbrick

cannot find parking at parking lot

too many people at canteen

need to use google map to walk to classroom

Calculation of Cgpa, hard time keeping track of deadlines

I do not have senior to ask regarding academic assistance

Notes and tutorials lack of organize and no clear target on studies

Figure 3.10 : Question 9

This question highlights potential pain points or challenges faced by new students, and hence a chatbot for addressing common queries could be implemented.

3.3 Requirements Analysis

3.3.1 Functional Requirements

Module	Role	Function
study module	Student	The system should allow student to create note by typing into text box
		The system should allow students to edit notes after creating.
		The system should allow students to delete notes after creating.
		The system should allow students to input numbers for conversion in a text box.
		The system should be able to calculate the unit conversion according to the input given by students.
		The system should allow students to input words in a text box to solve pronunciation confusion.
		The system should be able to output YouTube videos related to the students' input.
		The system should allow students to create events as a reminder.
		The system should be able to set the Calendar reminder to students' Google calendar.
		The system should be able to send the reminder to students' email addresses.
		The system should allow students to check expected grades by inputting details.
		The system should be able to calculate the expected grades according to the target CGPA input by students.

Health module	Student	The system should allow students to check calorie values by inputting details.
		The system should be able to calculate the calorie based on age, gender, height, weight, target weight, number of weeks to reach target weight and activity level.
		The system should allow students to check BMI values by inputting details.
		The system should be able to calculate the BMI based on age, gender, height and weight.
		The system should allow users to input exercises into the table.
		The system should be able to output YouTube links related to the user input exercise.
Personal module	Student	The system should allow students to register for new accounts by creating a username and password.
		The system should allow students to login with a valid username and password or existing Google account.
		The system should be able to lock the student account after 3 invalid login attempts.
		The system should allow students to reset passwords by clicking the reset password button.
		The system should allow students to logout accounts by clicking on the logout button or directly closing the webpage by clicking 'X' located at the top right corner of the browser.
		The system should allow students to view the personal information in the profile.

	<p>The system should allow students to edit the personal information in the profile.</p> <p>The system should allow student to create to-do list by typing into text box</p> <p>The system should allow students to edit to-do lists after creating.</p> <p>The system should allow students to delete to-do lists after creating.</p> <p>The system should allow student to create goal by typing into text box</p> <p>The system should allow students to edit goals after creating.</p> <p>The system should allow students to delete goals after creating.</p> <p>The system should allow students to ask queries in AI chatbots.</p> <p>The system should be able to generate query responses that are asked by students.</p>
Admin	<p>The system should allow admin to register for new accounts by creating a username and password.</p> <p>The system should allow the admin to login to the account with a valid username and password.</p> <p>The system should allow the admin to view the student database.</p> <p>The system should allow admin to update the student database record.</p> <p>The system should allow admin to delete the student database record.</p> <p>The system should allow admin to unlock an account that has been locked due to 3 times of invalid login attempts.</p>

Table 3.1: Functional Requirements

3.3.2 Non-functional Requirements

1. Performance: The system should be able to handle a large number of students using the system at the same time.
2. Reliability: The system should have not more than 0.05% downtime error per day except for maintenance hours.
3. Availability: The system should be available for 24/7 except for maintenance hours.
4. Security: The system should be able to provide security on private and confidential information without leaking out to unauthorised third parties.
5. Usability: The system should be attractive and user friendly with low learning time by users.
6. Integrity: The system should give authorisation to specific roles and prevent unauthorised access.

3.3.3 Critical evaluation

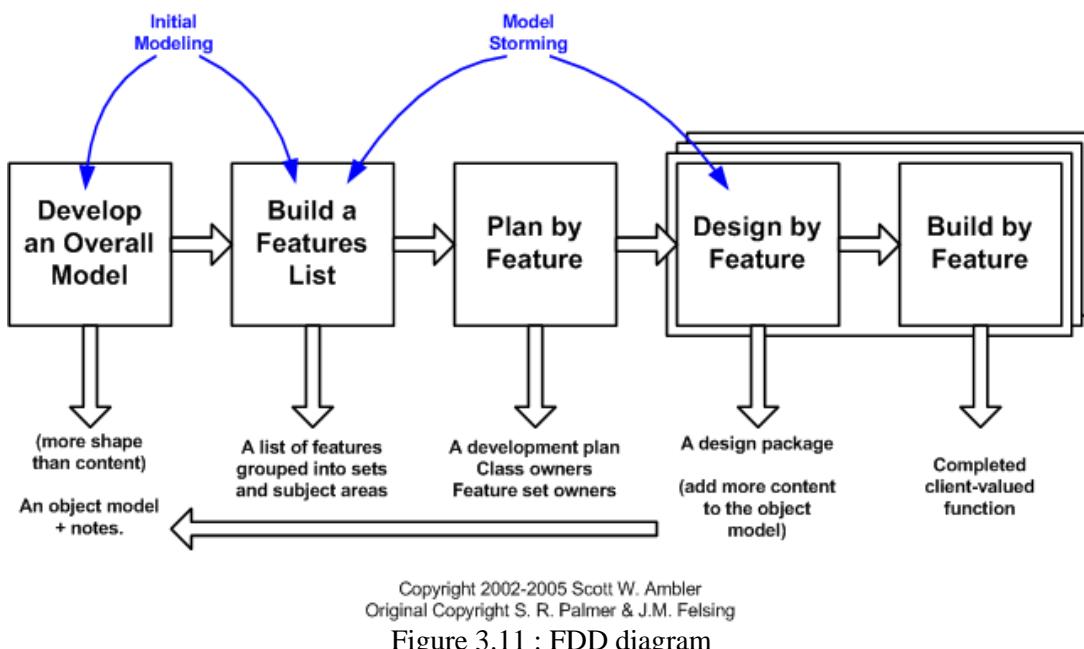


Figure 3.11 : FDD diagram

5-Step Process of Feature-Driven Development:

1. Develop overall model

FDD starts with creating a high-level model of the system to serve as a foundation for visual representation of the system which includes identifying the main components of the software, the relationships between components, and the key functionalities provided. In this project, this would involve identifying the main component of the system which is study, health and personal categories. Each of the category key functionalities provided is different.

2. Build feature list

Once the overall model is established, the development team creates a feature list by breaking down the categories into smaller, more manageable features. For example, under the study category, features like note-taking, video teaching pronunciation, GPA calculator, calendar reminder, measurement converters are identified. Under the health category, calorie tracker, BMI calculator and exercise workout schedule are included. Under the personal category, personal profile, to-do list, goal getter and AI chatbot are covered. Each feature will be able to be clear, specific and completed within a short period of time.

Plan by feature

After defining the feature list, it can begin to plan the development process or iteration. This involves assigning features to specific iterations based on priorities and resource availability. In this system, the core functionality of the system, which is personal profile in personal category will be developed first, then followed by functions in study category, health category and personal category.

3. Design by feature

In the design phase, detailed designs for each feature are created, taking into consideration the overall model and any dependencies between features. It will determine any necessary APIs or integrations such as YouTube API for video teaching function, workout schedule. User interface design, class diagram and flowchart will be created in this phase.

4. Build by feature

After finishing designing, it will proceed to build the features incrementally. Each Features are implemented one at a time independently with regular tests and inspections to ensure quality and adherence to design specifications. For example, when the note-taking feature is developed, it will be tested to verify its functionality and then move on to the next feature.

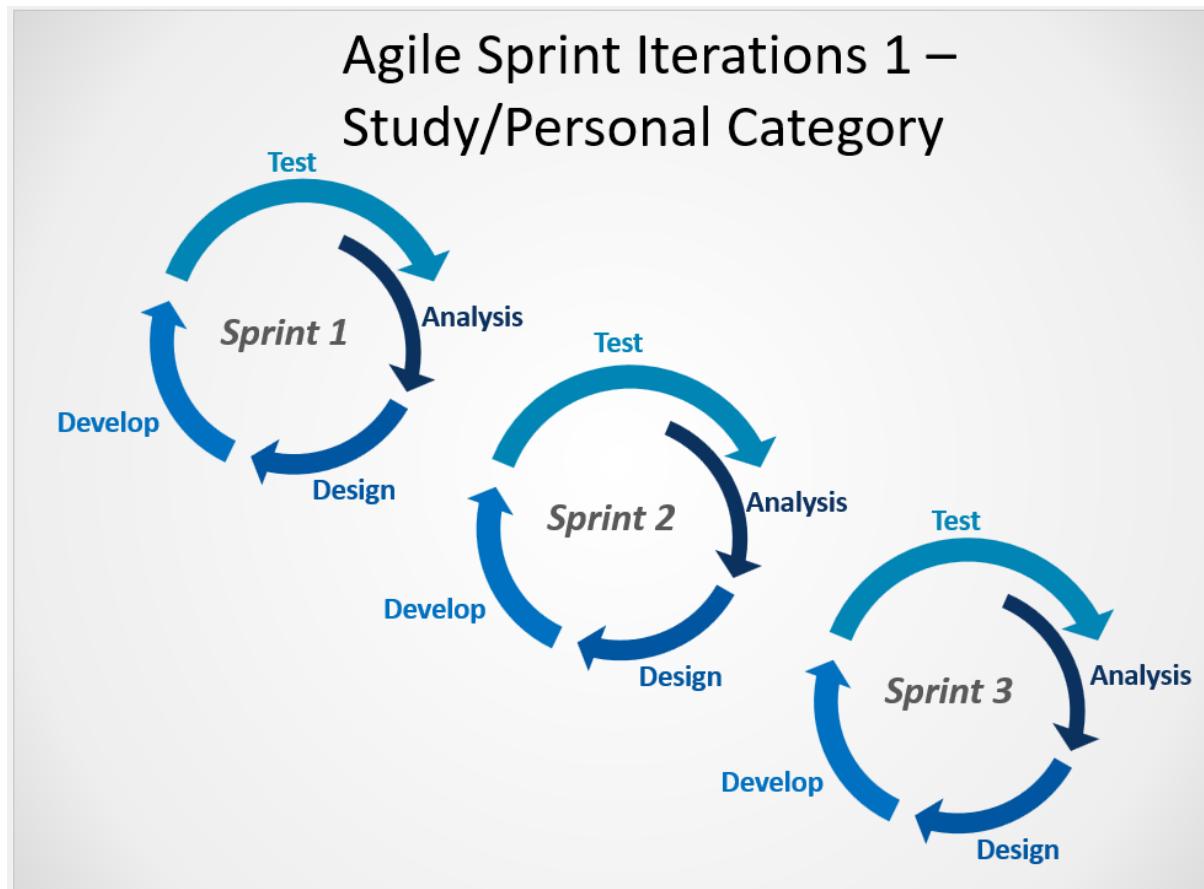


Figure 3.12 : Iteration 1

Iteration 1: Study Category, Personal Category

Sprint 1: personal profile (register, login, logout), note-taking

Sprint 2: measurement converter, video teaching pronunciation

Sprint 3: GPA calculator, calendar reminder

In this sprint, these functions will be considered as one cycle of the sprint which will go through the process of analysis, design, development and testing in 2 to 10 days. After 10 days, the next sprint (sprint 2) is executed, the 4 processes will be repeated within the timeframe. The same goes with sprint 3. The process is iterative.

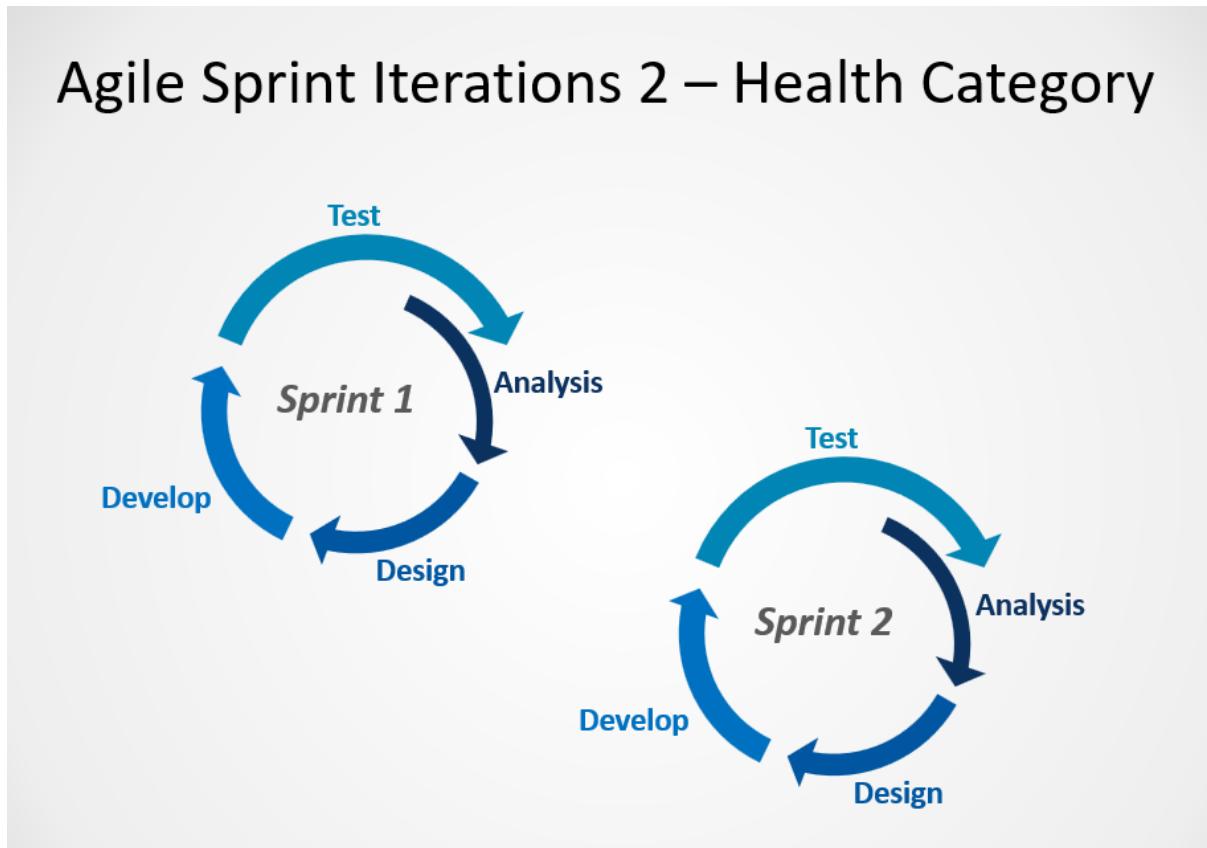


Figure 3.12 : Iteration 2

Iteration 2: Health Category

Sprint 1: calorie tracker, BMI calculator

Sprint 2: exercise schedule

In this sprint, these functions will be considered as one cycle of the sprint which will go through the process of analysis, design, development and testing in 2 to 10 days. After 10 days, the next sprint (sprint 2) is executed.

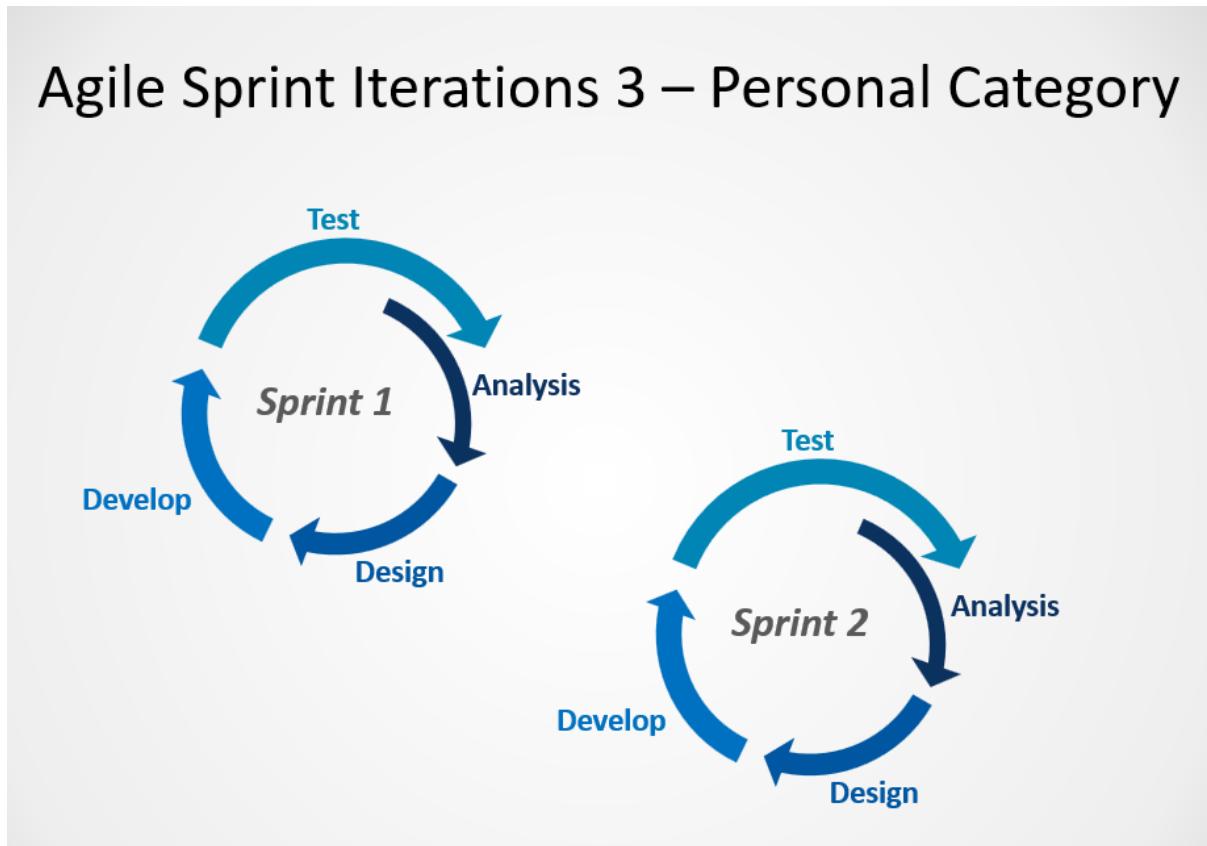


Figure 3.13 : Iteration 3

Iteration 3: Personal Category

Sprint 1: to-do list

Sprint 2: goal getter, AI chatbot

In this sprint, these functions will be considered as one cycle of the sprint which will go through the process of analysis, design, development and testing in 2 to 10 days. After 10 days, the next sprint (sprint 2) is executed.

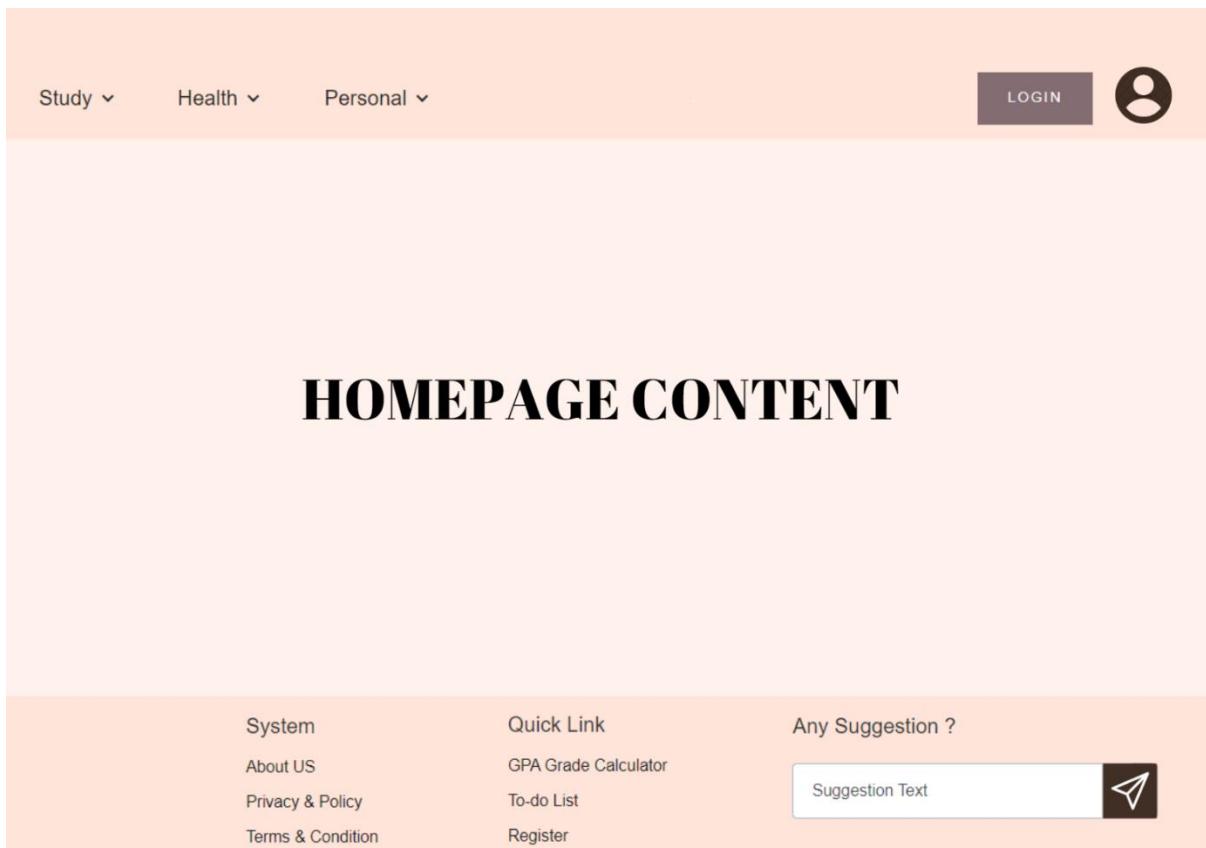


Figure 3.14 : Sample of User Interface

3.4 Development Environment

In order to establish the development of the proposed system, there are development resources such as hardware, software, programming tools and databases required to develop this proposed system.

3.4.1 Hardware Requirement

Device Model	ASUS TUF Gaming A17 FA707RC_FA707RC
Mouse	MIMouse 2
CPU Processor	AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz
CPU System Type	64-bit operating system, x64-based processor
RAM Capacity	40.0 GB
Operating System Edition	Windows 11 Home Single Language
Hard disk drive (HDD)	512 Byte
Free Capacity	

Table 3.1 : Hardware Requirement for Development Environment

3.4.2 Software Requirement

Integrated Development Environment	Visual Studio 2022
Web Browser	Google Chrome
Database	SQL Server Express LocalDB
Software Development Framework	.NET Framework
Programming Language	C-Sharp (C#) HyperText Markup Language (HTML) Cascading Style Sheet (CSS) JavaScript (JS) Python

Table 3.2 : Software Requirement for Development Environment

3.5 Operational Environment

It requires several hardware and software to use this system.

3.5.1 Hardware Requirement

Device Model	Any desktop or laptop that able to access the internet
CPU Processor	Any processor
CPU System Type	Any system type
Operating System Edition	Windows 10, Window 11

Table 3.3 : Hardware Requirement for Operational Environment

3.5.2 Software Requirement

Integrated Development Environment	Visual Studio 2022
Web Browser	Google Chrome, Internal Web Browser, Microsoft Edge
Database	SQL Server Express LocalDB

Table 3.4 : Software Requirement for Operational Environment

3.6 Chapter Summary and Evaluation

This chapter has shown an agile methodology, FDD, which allows features being developed and delivered incrementally in short iterations usually 2 to 10 days has been chosen as the software development model for this project. The ways of FDD implemented in developing this system is included. Online questionnaires, Google Form are used to gather information from the target users. A total of 66 feedback was gathered from 20/4/2024 until 25/4/2024. Functional and non-functional requirements for each module, a rough design on what the interface should look like are covered in this chapter as well. Lastly, the development environment and operational environment are discussed in this chapter.

Chapter 4

System Design

4 System Design

This chapter will cover all the diagrams and user interface flow diagrams to provide clearer illustration of the system. The use case in use case diagrams align with the functional requirements mentioned in section 3.3.1. A use case description table is created based on each use case in the diagram to tell the flow of processes. Activity diagrams are created to illustrate the user and system's process done based on each use case. Entity relationship diagrams (ERD) are used to design the structure of a database and show the tables and relationship between tables in a system. User interfaces are used to show the flow of the system in a more interactive way.

4.1 Use Case Model

Overview Use Case Diagram:

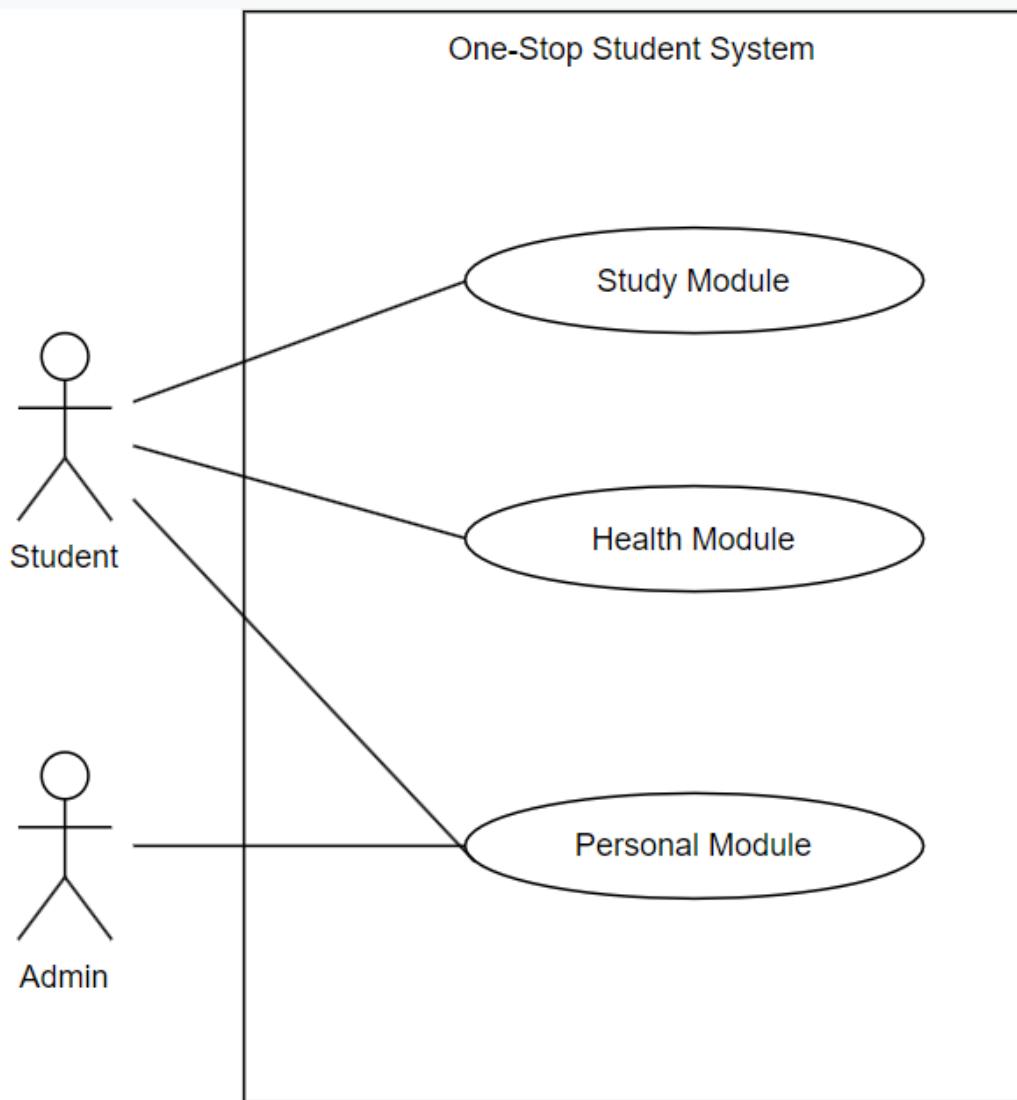


Figure 4.1: Overview Use Case Diagram

Detailed Use Case Diagram:

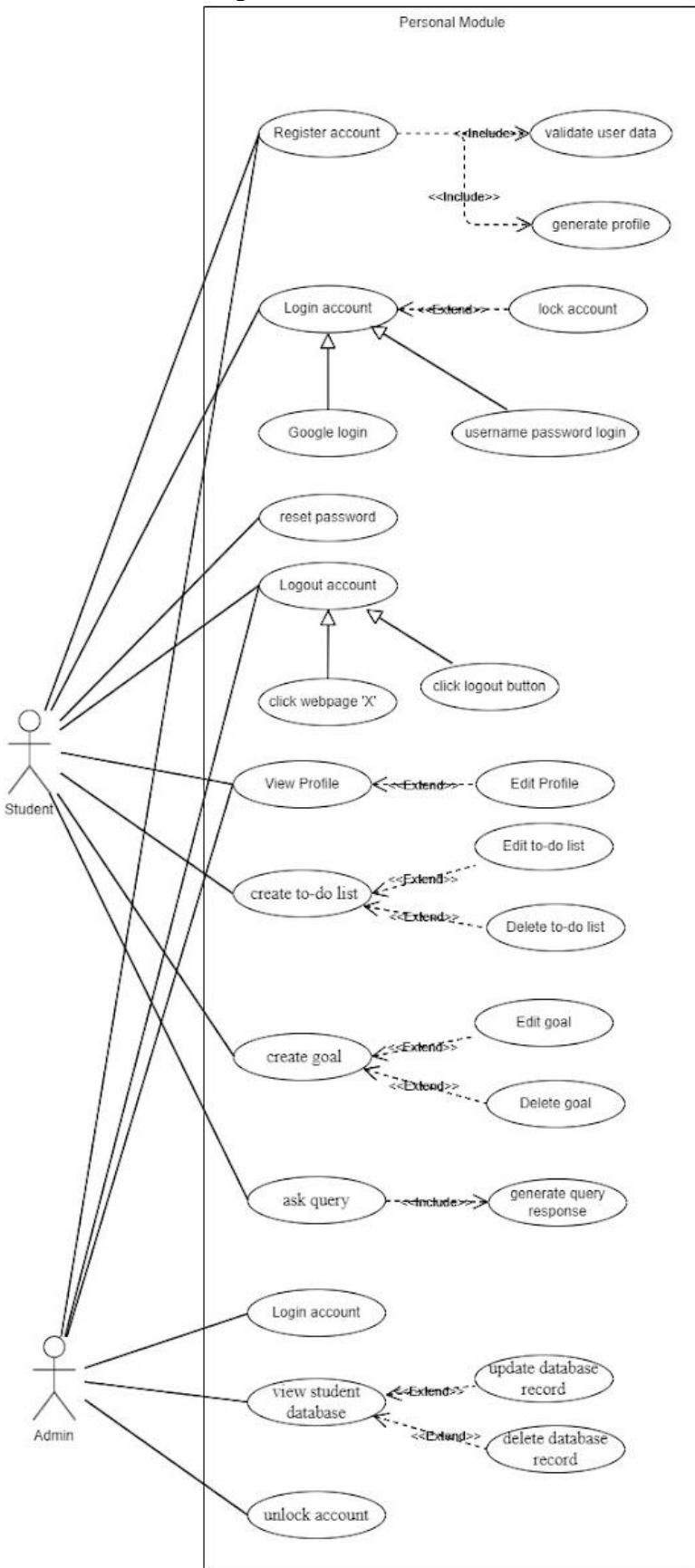


Figure 4.2: Detailed Use Case Diagram for personal category

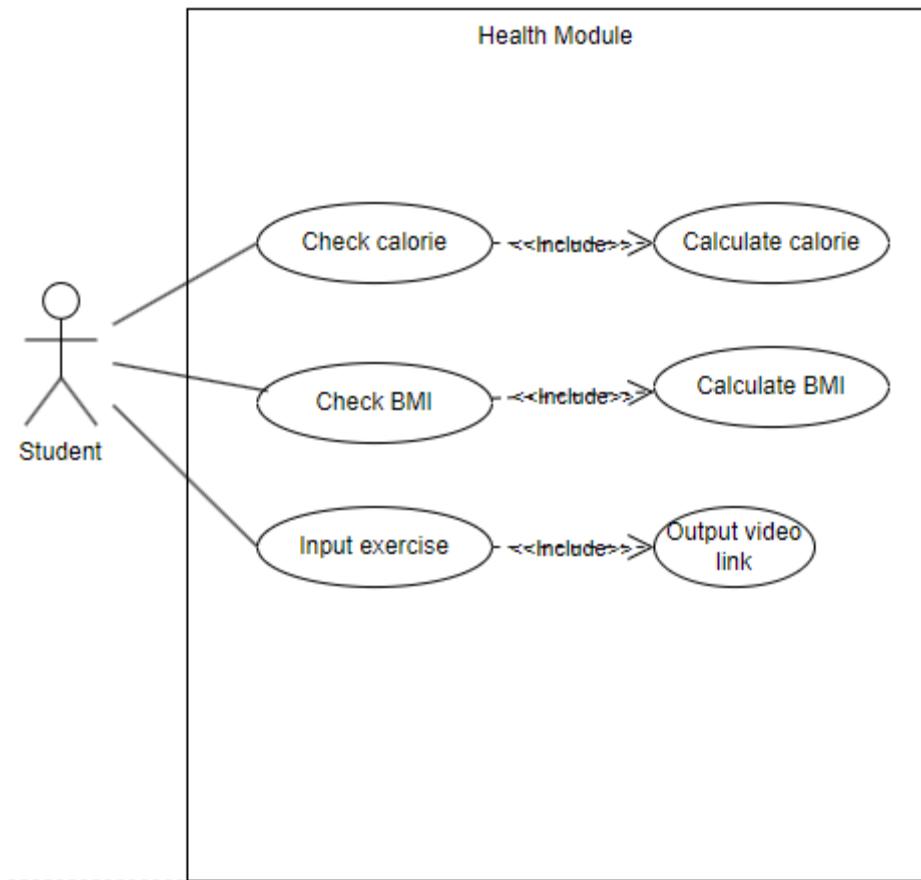


Figure 4.3: Detailed Use Case Diagram for health category

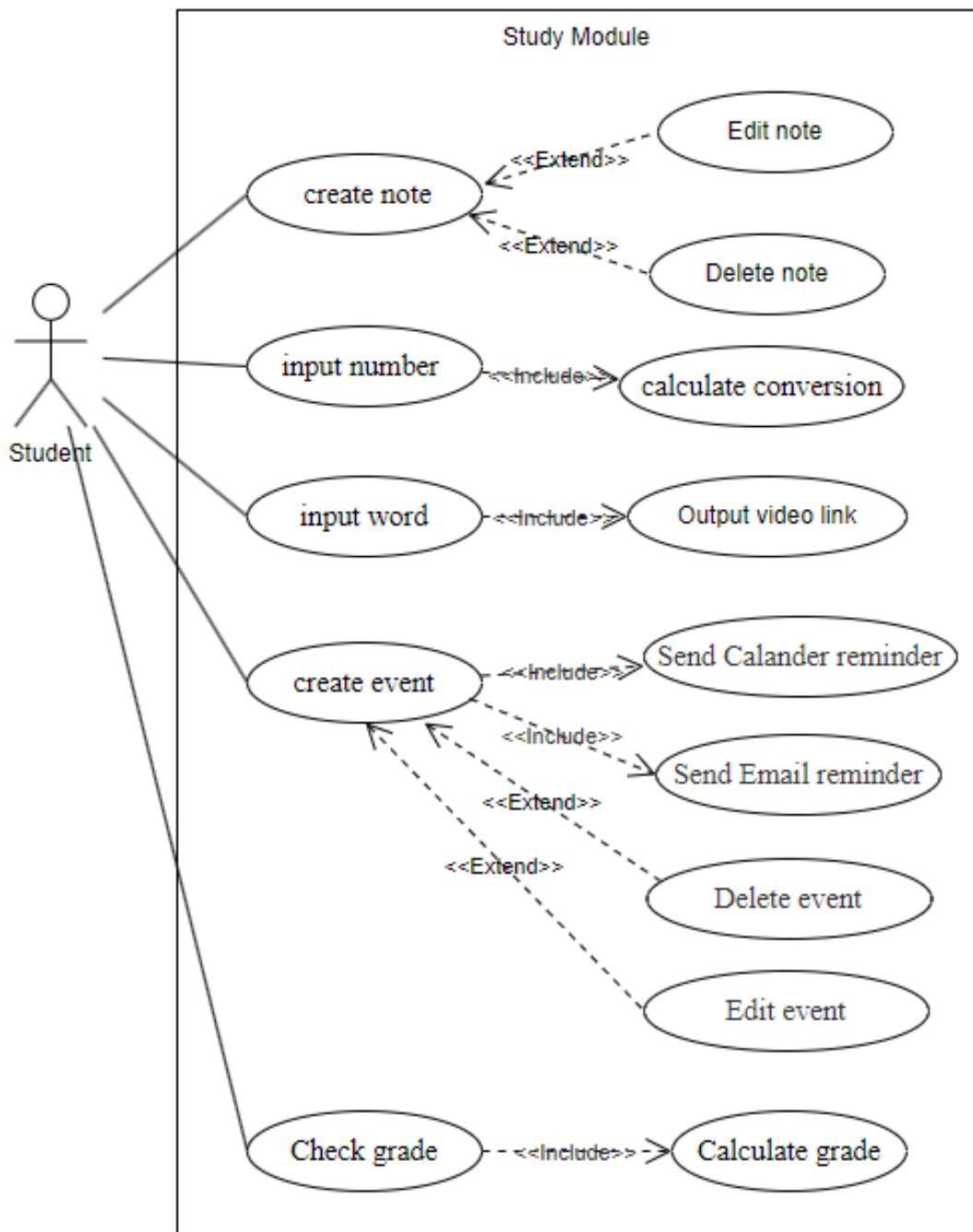


Figure 4.4: Detailed Use Case Diagram for study category

4.1.1 Use Case Description Table

Use Case Name: Register account	
Actor : Student, Admin	
Brief Description: This module allows the student or admin to register a new account	
Pre-condition: The student or admin must not have an account in the system.	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “register account”	4. Display relevant form
5. Enter registration details 6. click confirm button	7. Validate input details If the enter value is correct, the system will prompt for confirmation
8. Click submit button	9. The system will add the account details into database 10. The system will generate a profile and redirect to the login page.
Alternative flow of events:	
A1 Step 7: If the registration details is invalid, the system will display an error message “invalid registration details, please re-enter” and go back to step 4.	
A2 Step 8: If students or admin cancel the registration, the system will redirect student or admin to step 4.	
Post-condition: The student or admin will redirect to the login page	

Table 4.1 : Use case description table for register account

Use Case Name: Login student account

Actor : Student

Brief Description: This module allows the student to login to the one-stop student system.

Pre-condition: The student must register an account or have an existing Google account.

Main Flow of Events:

Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “login account”	4. Display relevant form
5. Enter username and password 6. click login button	7. Validate login details If the details are correct, the system will change the student's login status to “login” and redirect the student to the homepage.

Alternative flow of events:

A1 Step 5:

If a student chooses not to enter username and password to login, but chooses to use existing Google to login, the system will continue to step 7.

A2 Step 6:

If a student cancels the login function, the system will redirect the student to step 4.

A3 Step 7:

If the enter value is incorrect and reaches 3 times invalid login attempts, the system will display an error message “Your account has been locked because you have reached the maximum number of invalid attempts, please contact the system administrator.”, update account status to ‘locked’ in the database and cannot login again.

A4 Step 7:

If the enter value is incorrect but less than 3 times invalid login attempts, the system will display error message “Invalid username or password, please reenter” and go back to step 4.

A5 Step 7:

For students that choose to login with an existing account, if the account is invalid, the system will display an error message “Invalid account, please reselect” and go back to step 4.

Post-condition: Student will redirect to homepage.

Table 4.2 : Use case description table for login student account

Use Case Name: Login admin account

Actor : Admin

Brief Description: This module allows the admin to login to the one-stop student system.

Pre-condition: The admin must register an account.

Main Flow of Events:

Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “login account”	4. Display relevant form
5. Enter username and password 6. click login button	7. Validate input details If the enter value is correct, the system will change the admin’s login status to “login” and redirect the admin to the homepage.

Alternative flow of events:

A1 Step 6:

If an admin cancels the login function, the system will redirect the admin to step 4.

A2 Step 7:

If the enter value is incorrect, the system will display error message “Invalid username or password, please reenter” and go back to step 4.

Post-condition: The admin will redirect to the homepage

Table 4.3 : Use case description table for login admin account

<p>Use Case Name: Reset password</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to reset the password once the student forgets the password.</p> <p>Pre-condition: The student must register an account</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “reset password”	4. Display relevant form
5. Enter password and confirmed password 6. click confirm button	7. Validate input details If the enter value is correct, the system will update the student account password to a new password in the database.
<p>Alternative flow of events:</p> <p>A1 Step 6: If a student cancels the reset password function, the system will redirect the student to step 4.</p> <p>A2 Step 7: If the enter value is incorrect, the system will display error message “Invalid input, please reenter” and go back to step 4.</p>	
<p>Post-condition: Student can login with a new password</p>	

Table 4.4 : Use case description table for reset password

<p>Use Case Name: Logout account</p> <p>Actor : Student, Admin</p> <p>Brief Description: This module allows the student or admin to logout account</p> <p>Pre-condition: The student must register an account and is in login state.</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “logout account”	4. Change student’s or admin’s login status to “logout” and return to homepage
<p>Alternative flow of events:</p> <p>A1 Step 3: If the student or admin directly closes the window, the system will continue to step 4.</p>	
<p>Post-condition: The student or admin will redirect to homepage</p>	

Table 4.5 : Use case description table for logout

<p>Use Case Name: View profile</p> <p>Actor : Student, Admin</p> <p>Brief Description: This module allows the student to view profile</p> <p>Pre-condition: The student must login to account</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “view profile”	4. Display profile
<p>Alternative flow of events:</p>	
<p>Post-condition: Student or admin can see own profile</p>	

Table 4.6 : Use case description table for view profile

Use Case Name: Update profile	
Actor : Student, Admin	
Brief Description: This module allows the student to edit profile	
Pre-condition: The student must login to account and in profile page	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “view profile”	4. Display profile
5. Enter profile details that need to be updated 6. Select ‘Save’ button	7. Validate input details If the entered value is correct, the system will update the record in the database.
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
A2 Step 7. If the input value is incorrect, the system will display error message, “Invalid input value, please reenter” and go back to step 4.	
Post-condition: Student or admin will be able to view the new profile details	

Table 4.7 : Use case description table for update profile

Use Case Name: Create to-do list	
Actor : Student	
Brief Description: This module allows the student to create a to-do list by inputting in a textbox.	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “to-do list”	4. Display to-do list page
5. Enter to-do details 6. Click ‘Add’ button	7. add the details in the database 8. display the to-do list details
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a new to-do list added	

Table 4.8 : Use case description table for create to-do list

Use Case Name: Edit to-do list	
Actor : Student	
Brief Description: This module allows the student to edit a to-do list record.	
Pre-condition: The student must login to account and the to-do list record must exist	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “to-do list”	4. Display to-do list page
5. Click “edit” of one to-do list record	6. Display pop up window to enter edit details
7. Enter details that need to be updated 8. Select ‘Confirm’ button	9. update the record in the database 10. display the new to-do list details
Alternative flow of events:	
A1 Step 8. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a updated to-do list details	

Table 4.9 : Use case description table for edit to-do list

Use Case Name: Delete to-do list

Actor : Student

Brief Description: This module allows the student to delete a to-do list record.

Pre-condition: The student must login to account and the to-do list record must exist

Main Flow of Events:

Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “to-do list”	4. Display to-do list page
5. Click “delete” of one to-do list record	6. Display confirmation message
7. Click ‘confirm’ button	8. The system will delete the record in the database.

Alternative flow of events:

A1 Step 7.

If the student selects the ‘cancel’ button, the system will redirect the student to step 4.

Post-condition: Student cannot see the deleted to-do list record

Table 4.10 : Use case description table for delete to-do list

Use Case Name: Create goal	
Actor : Student	
Brief Description: This module allows the student to create a goal by inputting in a textbox.	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “goal getter”	4. Display goal getter page
5. Enter goal details 6. Select ‘Add’ button	7. add the details in the database 8. display the goal
<u>Alternative flow of events:</u>	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a new goal added	

Table 4.11 : Use case description table for create goal

Use Case Name: Edit goal	
Actor : Student	
Brief Description: This module allows the student to edit a goal record.	
Pre-condition: The student must login to account and the goal record must exist	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “goal getter”	4. Display goal page
5. Click “edit” of one goal record	6. Display pop up window to enter edit details
7. Enter details that need to updated	9. update the record in the database
8. Select ‘Confirm’ button	10. display the new goal
Alternative flow of events:	
A1 Step 8. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a updated goal details	

Table 4.12 : Use case description table for edit goal

Use Case Name: Delete goal	
Actor : Student	
Brief Description: This module allows the student to delete a goal record.	
Pre-condition: The student must login to account and the goal record must exist	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “goal getter”	4. Display goal page
5. Click “delete” of one goal record	6. Display confirmation message
7. Click ‘confirm’ button	8. The system will delete the record in the database.
Alternative flow of events:	
A1 Step 7. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student cannot see the deleted goal record	

Table 4.13 : Use case description table for delete goal

<p>Use Case Name: Ask query</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to ask query about school</p> <p>Pre-condition: The student must login to account</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “AI chatbot”	4. Display AI chatbot page
5. Enter query details 6. Select Submit button	7. The system will verify and generate query response
<p>Alternative flow of events:</p> <p>A1 Step 6.</p> <p>If the student selects the ‘cancel’ button, the system will redirect the student to step 4.</p> <p>Post-condition: Students will get the answer for the query.</p>	

Table 4.14 : Use case description table for ask query

Use Case Name: View student database Actor : Admin Brief Description: This module allows the admin to view student database Pre-condition: The admin must login to account Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “view student database”	4. Display database record
Alternative flow of events:	
Post-condition: Admin can view student database	

Table 4.15 : Use case description table for view student database

<p>Use Case Name: Update student database</p> <p>Actor : Admin</p> <p>Brief Description: This module allows the admin to update student database</p> <p>Pre-condition: The admin must login to account and the student record must exist in database</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “view student database”	4. Display database record
5. Click “update” for a particular student details record	6. Display relevant form
7. Enter student details that need to be updated 8. Click ‘Confirm’ button	9. Validate input details If the entered value is correct, the system will update the record in the database.
<p>Alternative flow of events:</p> <p>A1 Step 8. If the admin selects the ‘cancel’ button, the system will redirect the admin to step 4.</p> <p>A2 Step 9. If the input value is incorrect, the system will display error message, “Invalid input value, please reenter” and go back to step 4.</p>	
<p>Post-condition: Admin can see the updated student database details</p>	

Table 4.16 : Use case description table for update student database

Use Case Name: Delete student database	
Actor : Admin	
Brief Description: This module allows the admin to delete student database	
Pre-condition: The admin must login to account and the student record must exist in database	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Select “view student database”	4. Display database record
5. Click “delete” for a particular student record	6. Display confirmation message
7. Click ‘confirm’ button	8. The system will delete the record in the database.
Alternative flow of events:	
A1 Step 7. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Admin cannot see the deleted student database record	

Table 4.17 : Use case description table for delete student database

Use Case Name: Unlock locked account								
Actor : Admin								
Brief Description: This module allows the admin to unlock locked accounts due to reaching maximum invalid login attempts.								
Pre-condition: The admin must have a system administrator account.								
Main Flow of Events:								
<table border="1"> <thead> <tr> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1. Browse the main menu</td> <td>2. Display main menu</td> </tr> <tr> <td>3. Select “view locked account”</td> <td>4. Display locked account</td> </tr> <tr> <td>5. Click “unlock” for a particular locked account</td> <td>6. Update account status to ‘unlocked’ in database for that record</td> </tr> </tbody> </table>	Actor Action	System Response	1. Browse the main menu	2. Display main menu	3. Select “view locked account”	4. Display locked account	5. Click “unlock” for a particular locked account	6. Update account status to ‘unlocked’ in database for that record
Actor Action	System Response							
1. Browse the main menu	2. Display main menu							
3. Select “view locked account”	4. Display locked account							
5. Click “unlock” for a particular locked account	6. Update account status to ‘unlocked’ in database for that record							
Alternative flow of events:								
A1 Step 5. If the admin does not choose to unlock the lock account, it will go back and stop at step 4.								
Post-condition: The particular student account can be used to login again								

Table 4.18 : Use case description table for unlock account

Use Case Name: Create note	
Actor : Student	
Brief Description: This module allows the student to create a note by inputting in a textbox.	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “note-taking”	4. Display note-taking page
5. Enter note details 6. Select ‘Add’ button	7. add the details in the database 8. display the note
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a new note added	

Table 4.19 : Use case description table for create note

Use Case Name: Edit note	
Actor : Student	
Brief Description: This module allows the student to edit a note record.	
Pre-condition: The student must login to account and the note record must exist	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “note-taking”	4. Display note-taking page
5. Click “edit” of one note record	6. Display pop up window to enter edit details
7. Enter details that need to updated 8. Click ‘Confirm’ button	9. update the record in the database 10. display the new note
Alternative flow of events:	
A1 Step 8. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can see a updated note details	

Table 4.20 : Use case description table for edit note

Use Case Name: Delete note	
Actor : Student	
Brief Description: This module allows the student to delete a note record.	
Pre-condition: The student must login to account and the note record must exist	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “note-taking”	4. Display note-taking page
5. Click “delete” of one note record	6. Display confirmation message
7. Click ‘confirm’ button	8. The system will delete the record in the database.
Alternative flow of events:	
A1 Step 7. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student cannot see the deleted note record	

Table 4.21 : Use case description table for delete note

Use Case Name: Input number for unit conversion

Actor : Student

Brief Description: This module allows the student to input numbers for unit conversion by inputting in a textbox.

Pre-condition: The student must login to account

Main Flow of Events:

Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “unit conversion”	4. Display unit conversion page
5. Enter number	7. calculate the conversion result
6. Click ‘Confirm’ button	8. display the conversion result.

Alternative flow of events:

A1 Step 6.

If the student selects the ‘cancel’ button, the system will redirect the student to step 4.

Post-condition: Student will get the conversion result

Table 4.22 : Use case description table for input number for unit conversion

<p>Use Case Name: Input input confused word for proper pronunciation</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to input proper pronunciation for a confused word by inputting in a textbox.</p> <p>Pre-condition: The student must login to account</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “video pronunciation function”	4. Display video pronunciation function page
5. Enter word 6. Click ‘Confirm’ button	7. search YouTube video 8. display the video result
<p>Alternative flow of events:</p> <p>A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.</p> <p>Post-condition: Student will get the video result</p>	

Table 4.23 : Use case description table for input pronunciation confusing word

<p>Use Case Name: Create event reminder</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to create events by inputting in a textbox.</p> <p>Pre-condition: The student must login to account</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “calendar reminder”	4. Display calendar reminder page
5. Enter event details 6. Select ‘Add’ button	7. add the details in the database 8. display event details and send reminders to Google Calendar and email address
<p>Alternative flow of events:</p> <p>A1 Step 6.</p> <p>If the student selects the ‘cancel’ button, the system will redirect the student to step 4.</p>	
<p>Post-condition: Student will get the notification in Google Calendar and email</p>	

Table 4.24 : Use case description table for create event reminder

<p>Use Case Name: Edit event reminder</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to edit an event record.</p> <p>Pre-condition: The student must login to account and the event record must exist</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “calendar reminder”	4. Display calendar reminder page
5. Click “edit” of one event record	6. Display pop up window to enter edit details
7. Enter details that need to updated	9. update the record in the database
8. Select ‘Confirm’ button	10. display the new event details
<p>Alternative flow of events:</p> <p>A1 Step 8.</p> <p>If the student selects the ‘cancel’ button, the system will redirect the student to step 4.</p>	
<p>Post-condition: Student can see a updated event details</p>	

Table 4.25 : Use case description table for edit event

<p>Use Case Name: Delete event reminder</p> <p>Actor : Student</p> <p>Brief Description: This module allows the student to delete an event record.</p> <p>Pre-condition: The student must login to account and the event record must exist</p> <p>Main Flow of Events:</p>	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “calendar reminder”	4. Display calendar reminder page
5. Click “delete” of one event record	6. Display confirmation message
7. Click ‘confirm’ button	8. The system will delete the record in the database.
<p>Alternative flow of events:</p> <p>A1 Step 7.</p> <p>If the student selects the ‘cancel’ button, the system will redirect the student to step 4.</p>	
<p>Post-condition: Student cannot see the deleted event record</p>	

Table 4.26 : Use case description table for delete event

Use Case Name: Check grade	
Actor : Student	
Brief Description: This module allows the student to check expected grade	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “CGPA grade calculator”	4. Display CGPA grade calculator page
5. Enter relevant details 6. Click ‘Confirm’ button	7. calculate expected grade 8. display the expected grade result.
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can get the expected grade result	

Table 4.27 : Use case description table for check grade

Use Case Name: Check calorie	
Actor : Student	
Brief Description: This module allows the student to check calorie	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “calorie calculator”	4. Display calorie calculator page
5. Enter relevant details 6. Click ‘Confirm’ button	7. Calculate calorie 8. Save record in database 9. display the calorie result
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can get the calorie result	

Table 4.28 : Use case description table for check calorie

Use Case Name: Check BMI	
Actor : Student	
Brief Description: This module allows the student to check BMI	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “BMI calculator”	4. Display BMI calculator page
5. Enter relevant details 6. Click ‘Confirm’ button	7. Calculate BMI 8. Save record in database 9. display the BMI result
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can get the expected BMI result	

Table 4.29 : Use case description table for check BMI

Use Case Name: Input exercise name	
Actor : Student	
Brief Description: This module allows the student to input exercise and the system will output relevant video links.	
Pre-condition: The student must login to account	
Main Flow of Events:	
Actor Action	System Response
1. Browse the main menu	2. Display main menu
3. Click “exercise workout schedule”	4. Display exercise workout schedule page
5. Enter exercise name 6. Click ‘Confirm’ button	7. Search the related video link 8. Save record in database 9. Display the video link result
Alternative flow of events:	
A1 Step 6. If the student selects the ‘cancel’ button, the system will redirect the student to step 4.	
Post-condition: Student can get the video link result	

Table 4.30 : Use case description table for input exercise name

4.2 Activity Diagram

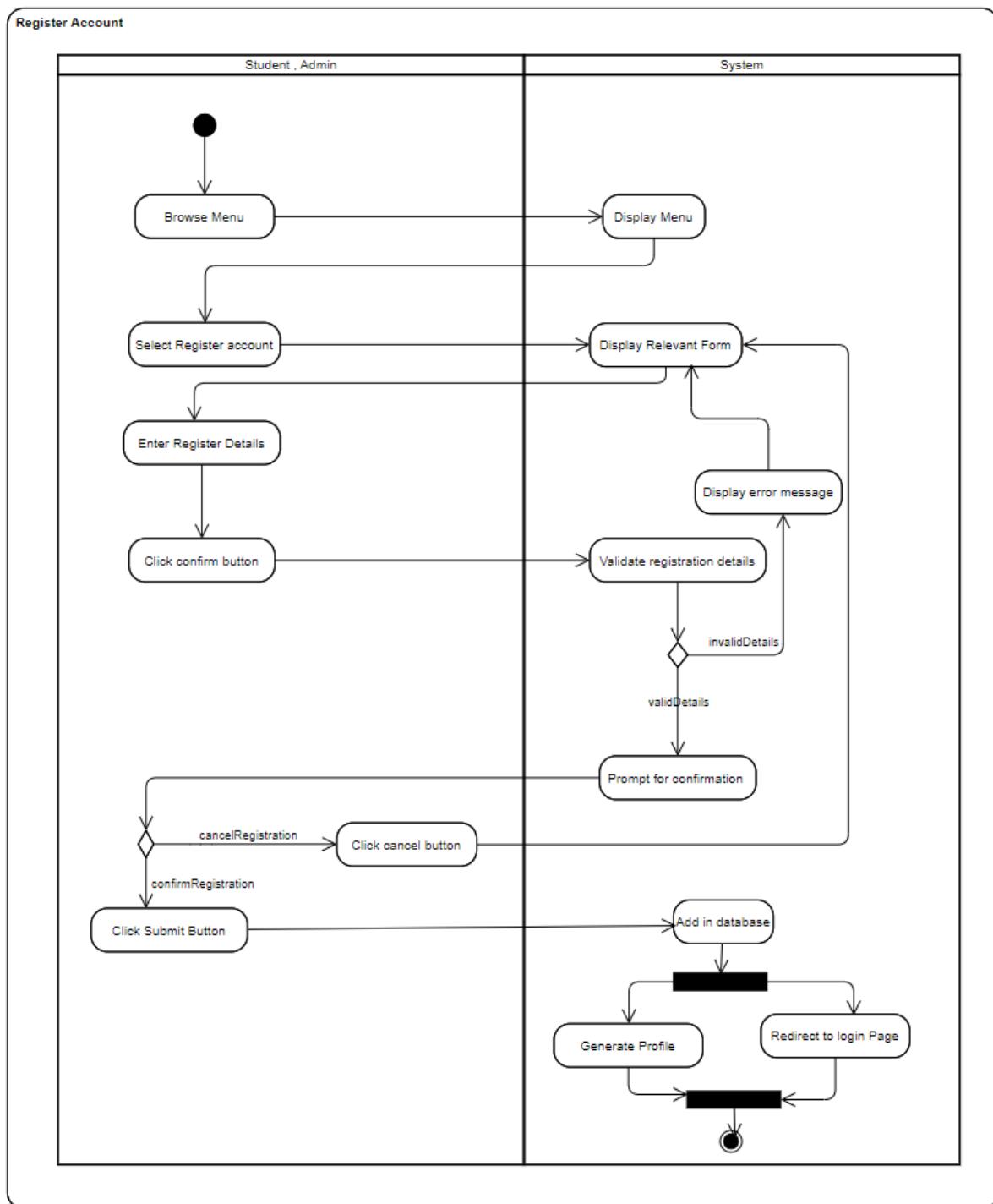


Diagram 4.5: Activity diagram for register account

The student or admin begins by browsing the main menu, the system will then display the main menu options. Student or admin then selects "register account", the system will display the relevant registration form. The student or admin proceeds to enter the registration details and clicks the confirm button. Then, the system validates the input details, and if entered values are correct, it prompts for confirmation. If a student or admin confirms registration, the student or admin will click the submit

button, the system will then add the account details into the database. Then, the system will generate a profile for the student or admin and redirect the student or admin to the login page concurrently. However, if the registration details is invalid, the system will display an error message and go back to display the registration form. Besides, if students or admin decides to cancel the registration by clicking the cancel button, the system will redirect student or admin to display the registration form.

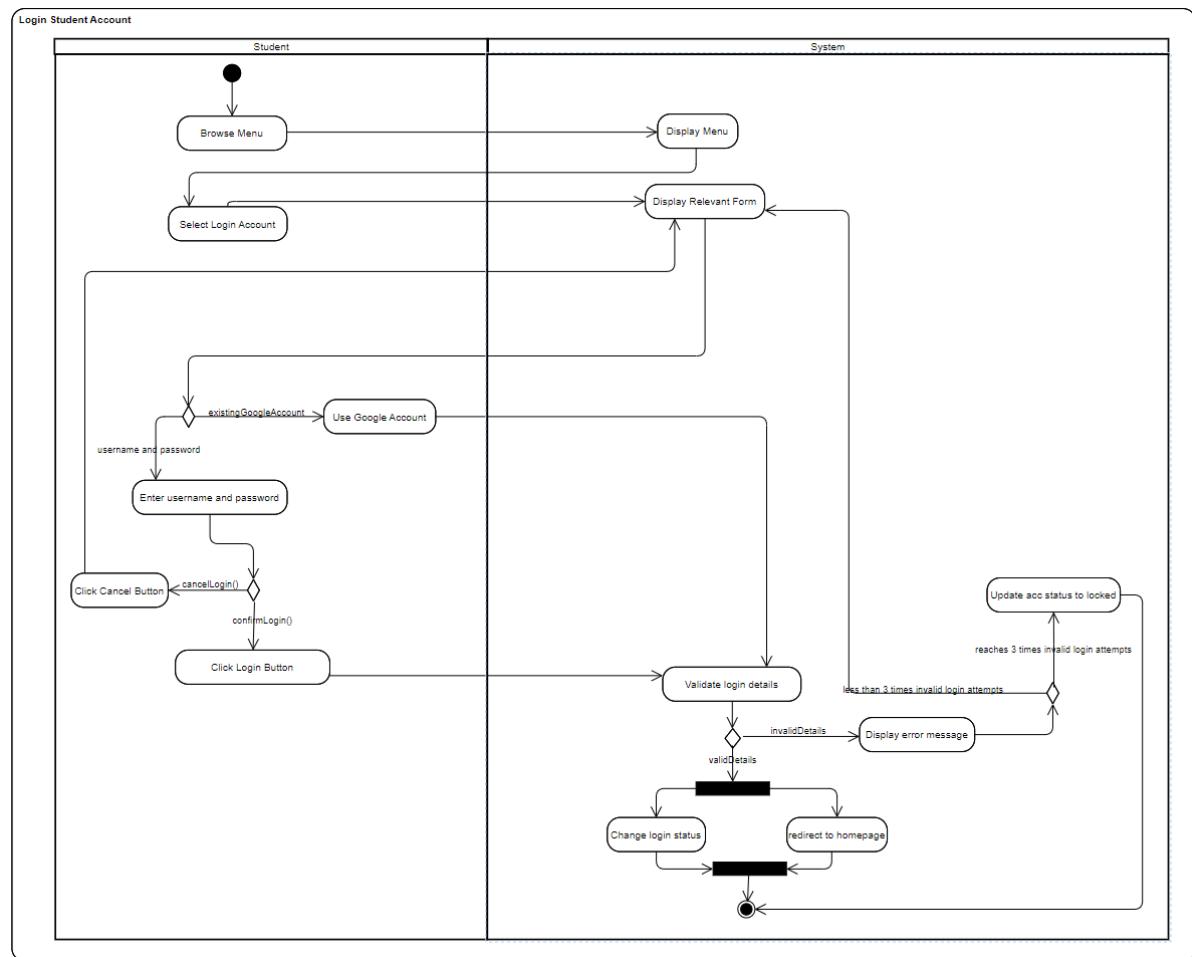


Diagram 4.6: Activity diagram for login student account

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects "login account", the system will display the login page. There are two login selections for students. The student can either proceed to enter username and password and click the login button when the student decides to login or the student chooses to use existing Google to login. Then, the system validates the login details, if details are correct, the system will change the student's login status to "login" and redirect the student to the homepage concurrently. However, if a student decides to cancel the login function by clicking the cancel button, the system will redirect the student to display the login page. Besides, if the details are incorrect, the system will display error messages. Then, it will check whether it has reached 3 time invalid login attempts. If it reaches 3 times invalid login attempts, the system will update account status to 'locked' in the database and the account cannot be used to login again. If there are less than 3 times invalid login attempts, the system will redirect the student to display login forms to repeat again.

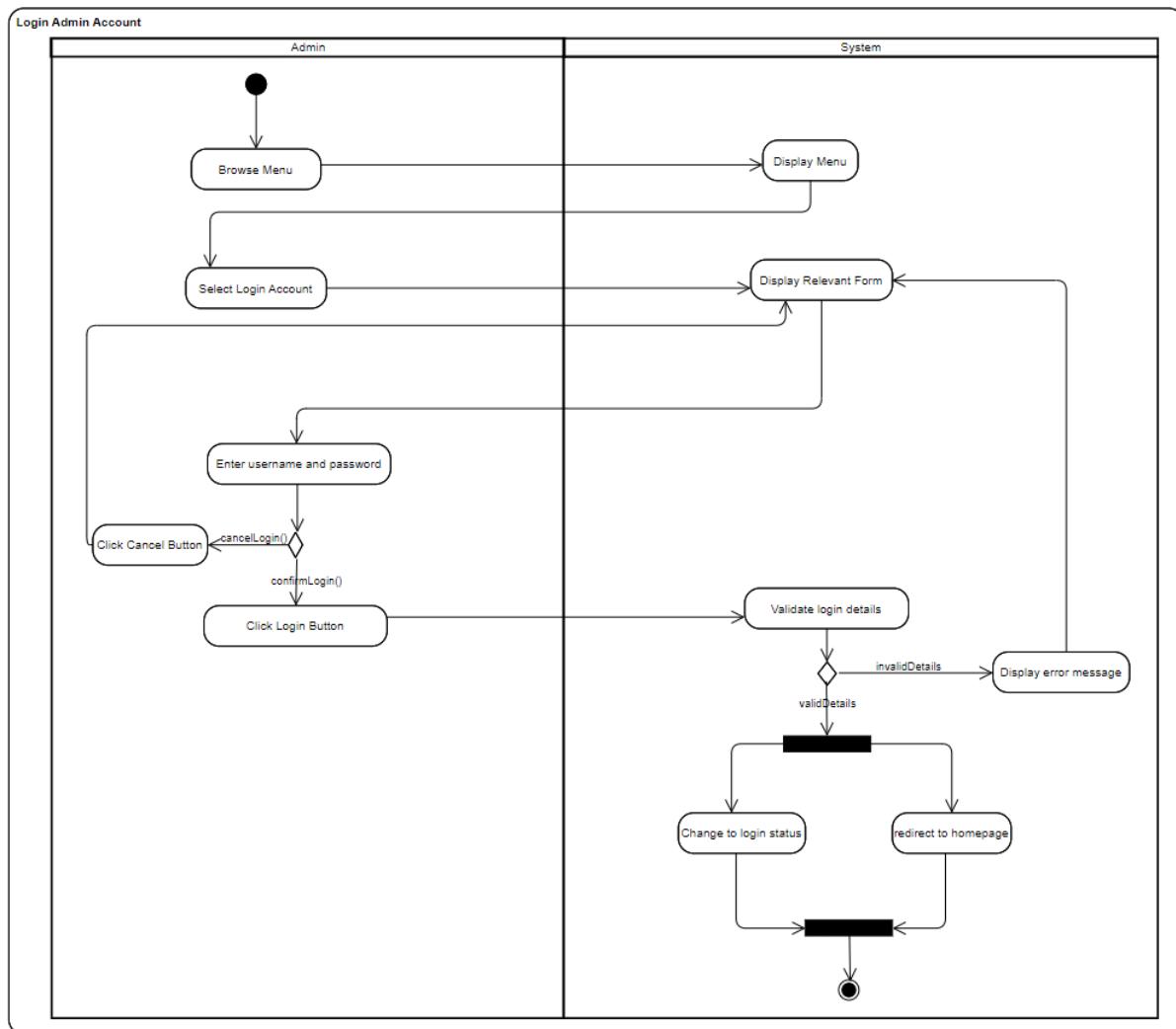


Diagram 4.7: Activity diagram for login admin account

The admin begins by browsing the main menu, the system will then display the main menu options. Admin then selects "login account", the system will display the login page. The admin can proceed to enter username and password and click the login button when the admin decides to login. Then, the system validates the login details, if details are correct, the system will change the admin's login status to "login" and redirect the admin to the homepage. However, if an admin decides to cancel the login function by clicking the cancel button, the system will redirect the admin to display the login page. Besides, if the details are incorrect, the system will display error messages and redirect the admin to display the login page as well to re-enter again.

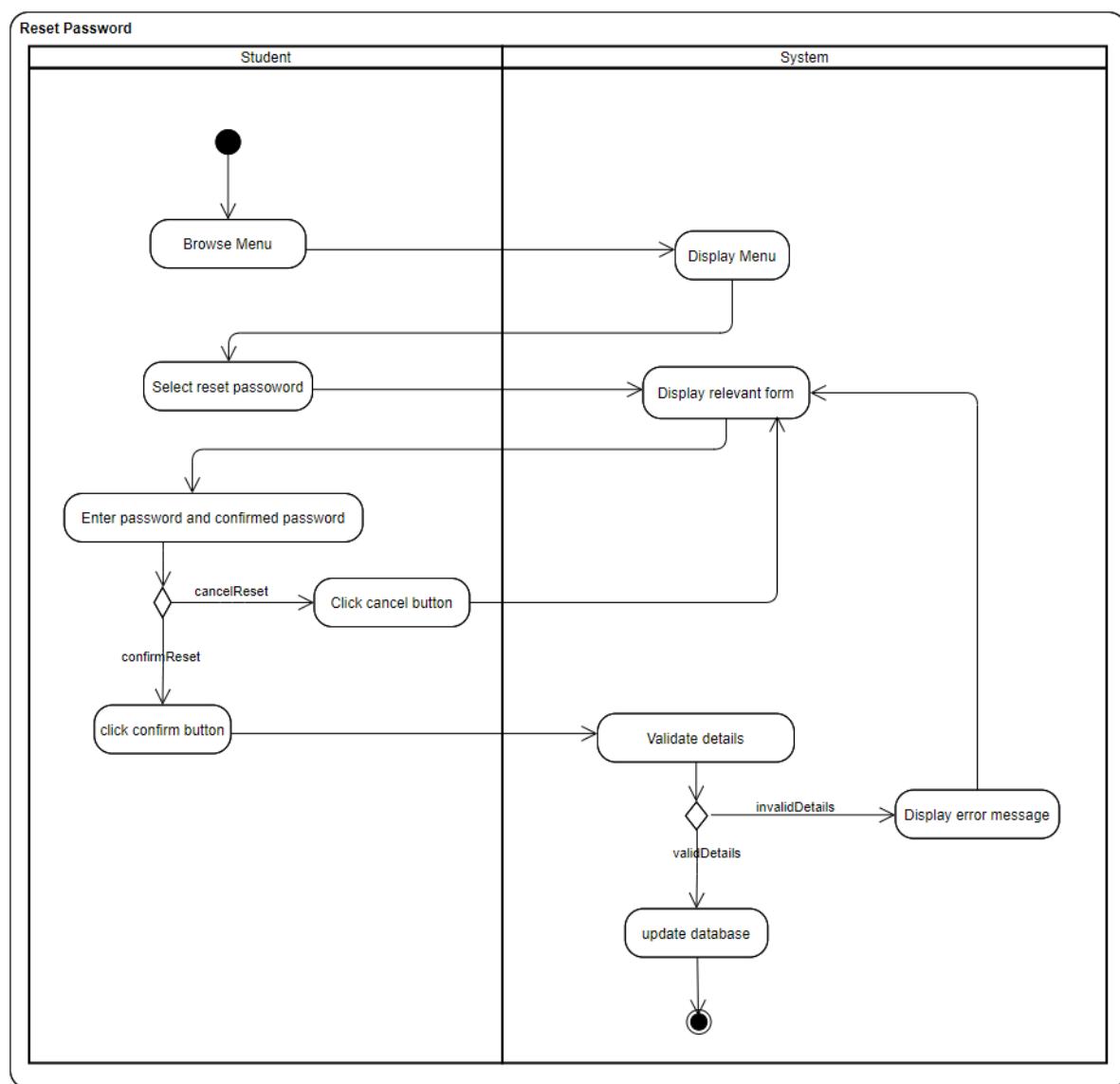


Diagram 4.8: Activity diagram for reset password

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects "reset password", the system will display the reset password page. The student can proceed to enter username and password and click the confirm button when the student confirms to reset. Then, the system validates the input details, if details are correct, the system will update the student account password to a new password in the database. However, if a student decides to cancel the reset password function by clicking the cancel button, the system will redirect the student to display the reset password page. Besides, if the details are incorrect, the system will display error messages and redirect the student to display the reset password page to re-enter again.

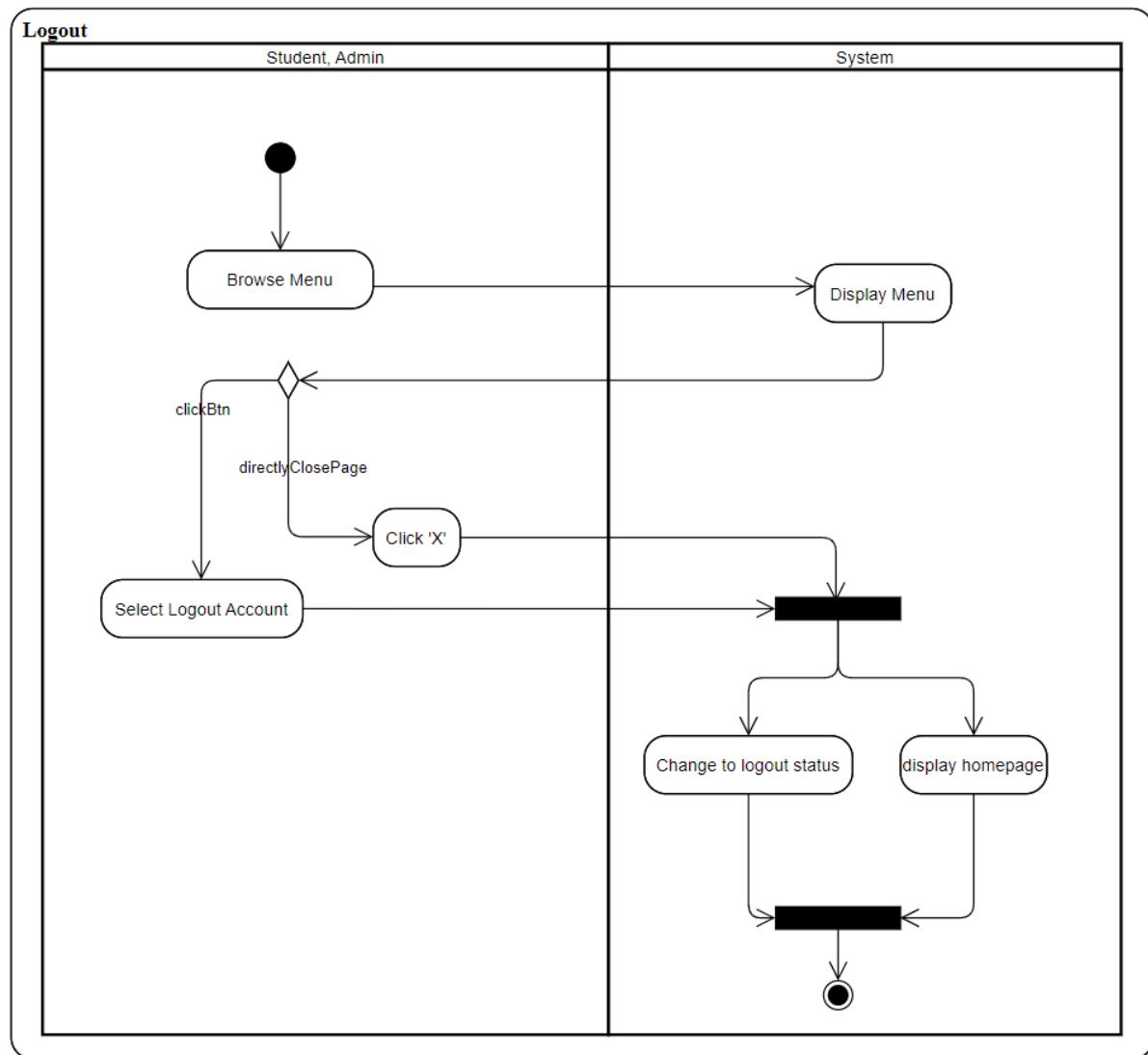


Diagram 4.9: Activity diagram for logout

The student or admin begins by browsing the main menu, the system will then display the main menu options. Student or admin can either selects "logout account" or directly close the window by clicking the 'X' button to perform the logout function, then the system will change the student's or admin's login status to "logout" and redirect to the homepage in parallel.

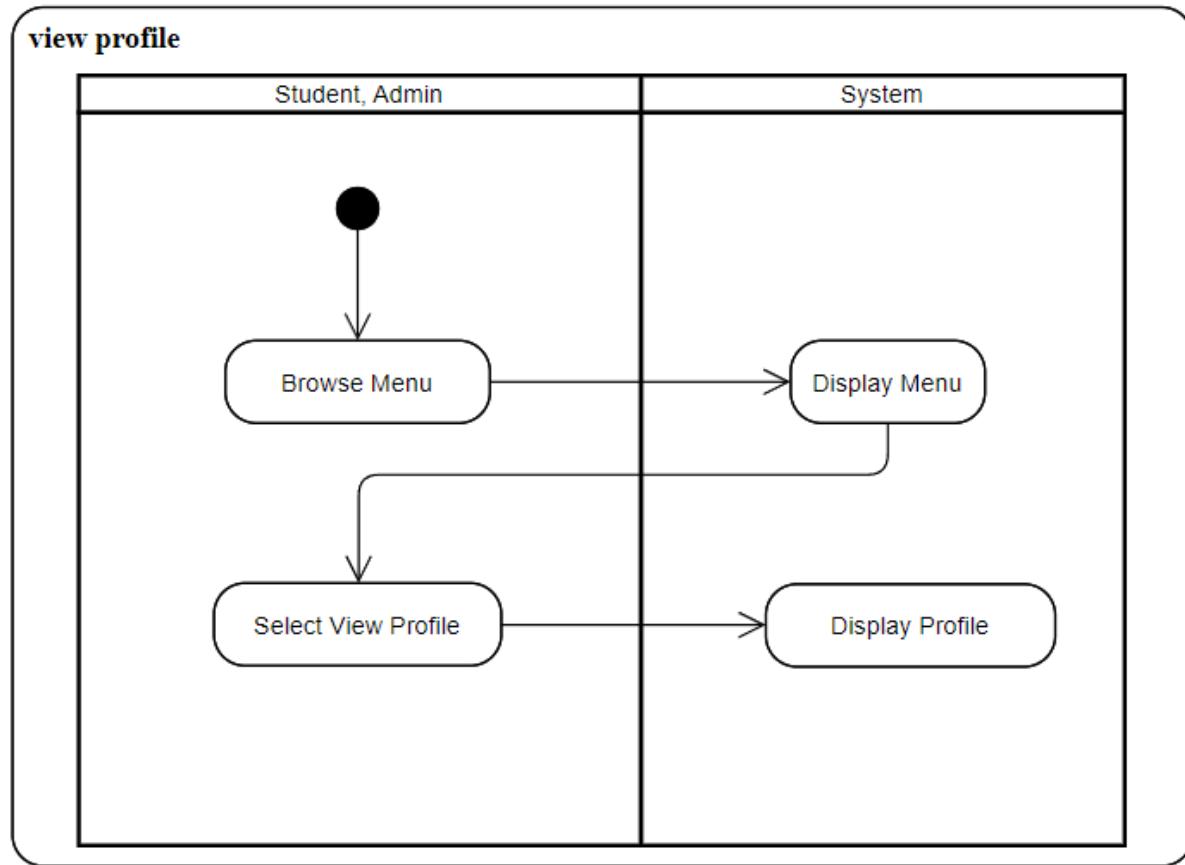


Diagram 4.10: Activity diagram for view profile

The student or admin begins by browsing the main menu, the system will then display the main menu options. Student or admin then selects "view profile", the system will display the profile page.

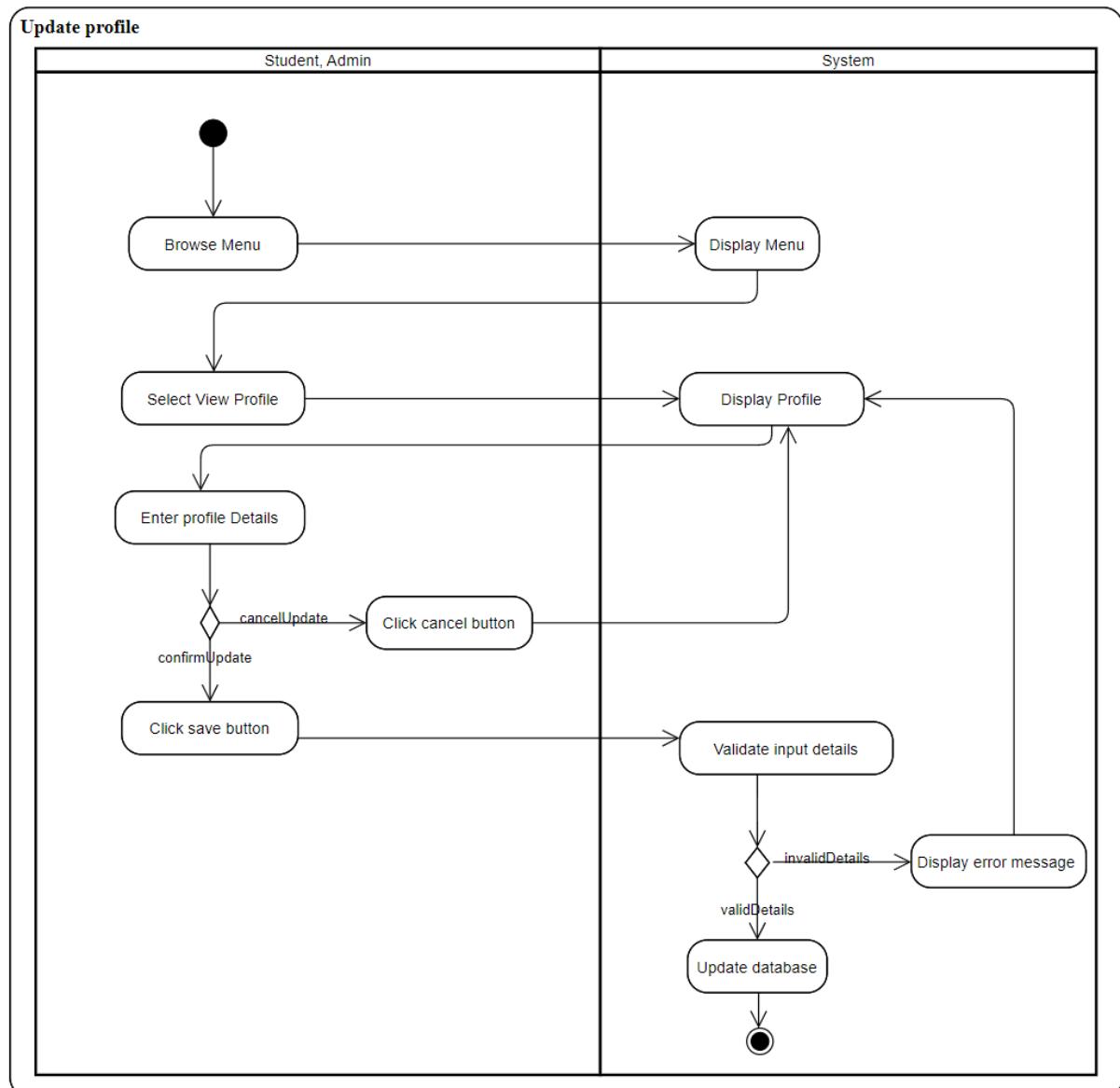


Diagram 4.11: Activity diagram for update profile

The student or admin begins by browsing the main menu, the system will then display the main menu options. Student or admin then selects "view profile", the system will display the profile page. The student can proceed to enter profile details that need to update and click the save button when the student confirms to update. Then, the system validates the input details, if details are correct, the system will update the record in the database. However, if a student decides to cancel the update function by clicking the cancel button, the system will redirect the student to display the profile page. Besides, if the input details are incorrect, the system will display error messages and redirect the student to display the profile page to re-enter again.

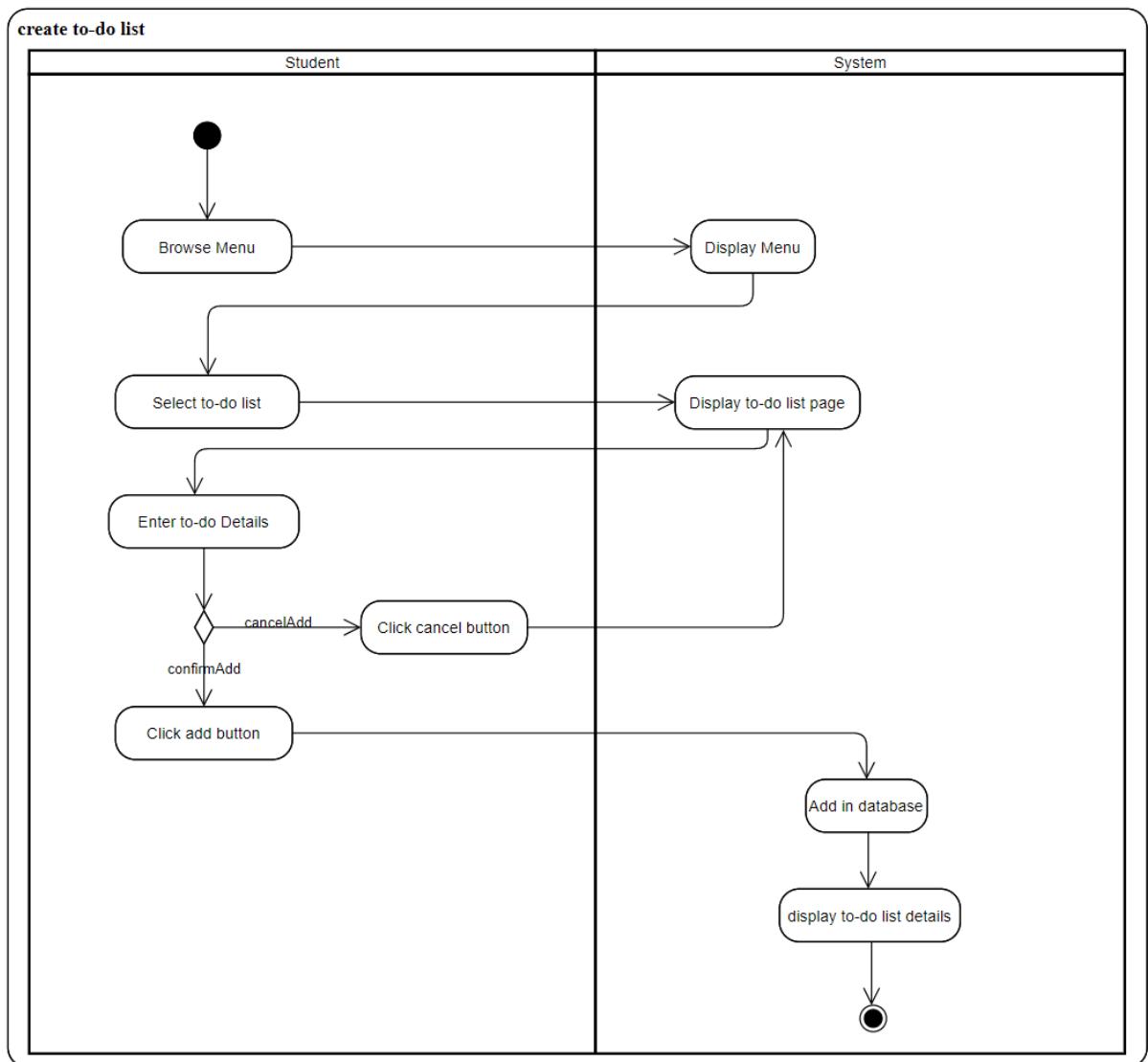


Diagram 4.12: Activity diagram for create to-do list

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “to-do list”, the system will display the to-do list page. The student can proceed to enter to-do details and click the add button when the student confirms to add to-do list. Then, the system will add the details in the database and display the to-do list details to student. However, if a student decides to cancel the add function by clicking the cancel button, the system will redirect the student to display the to-do list page.

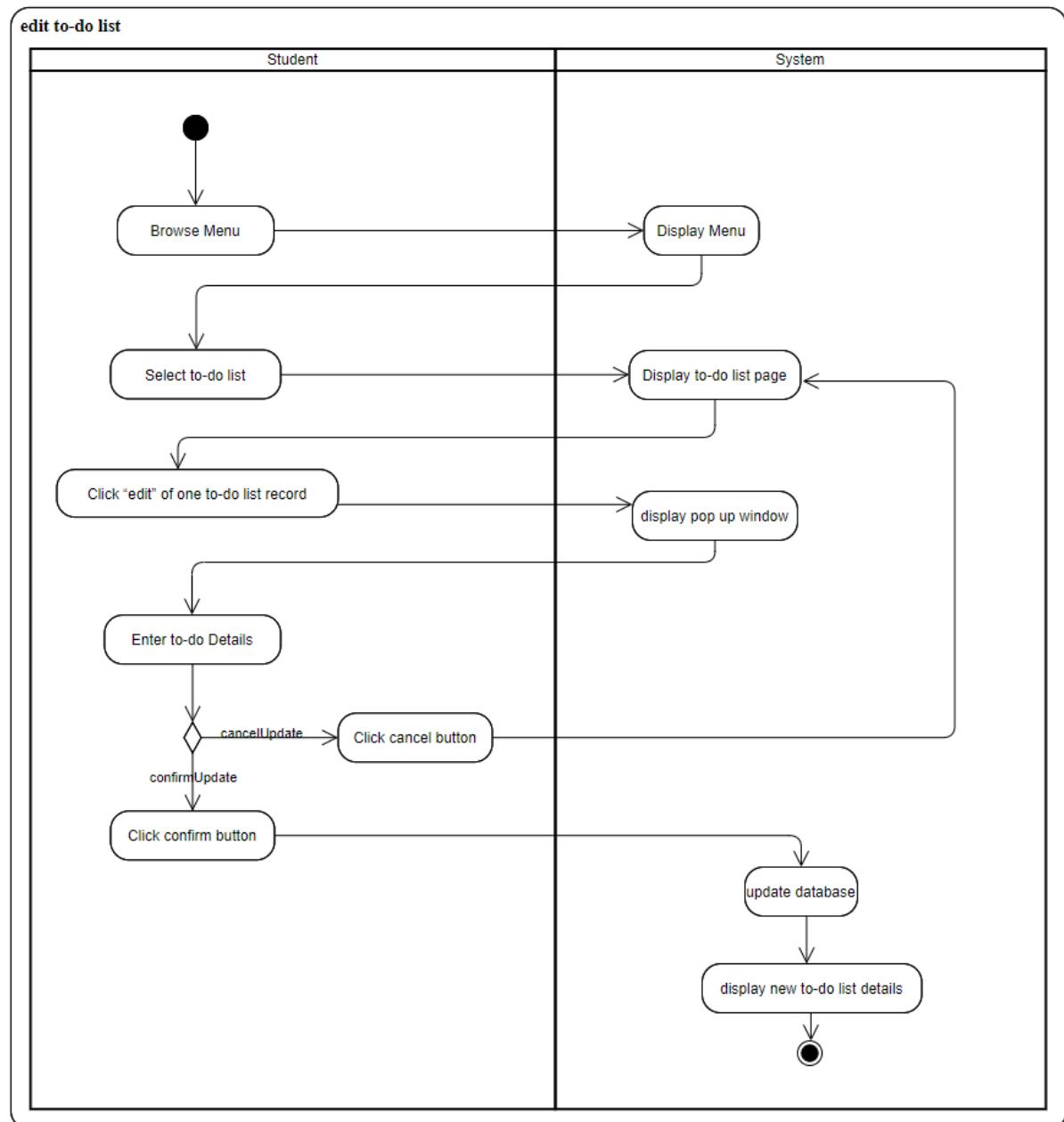


Diagram 4.13: Activity diagram for edit to-do list

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “to-do list”, the system will display the to-do list page. The student can click “edit” of one to-do list record and the system will display a pop up window to enter edit details. Then, student can proceed to enter details that need to be updated and click the confirm button when the student confirms to update. Then, the system will update the record in the database and display the new to-do list details to student. However, if a student decides to cancel the update function by clicking the cancel button, the system will redirect the student to display the to-do list page.

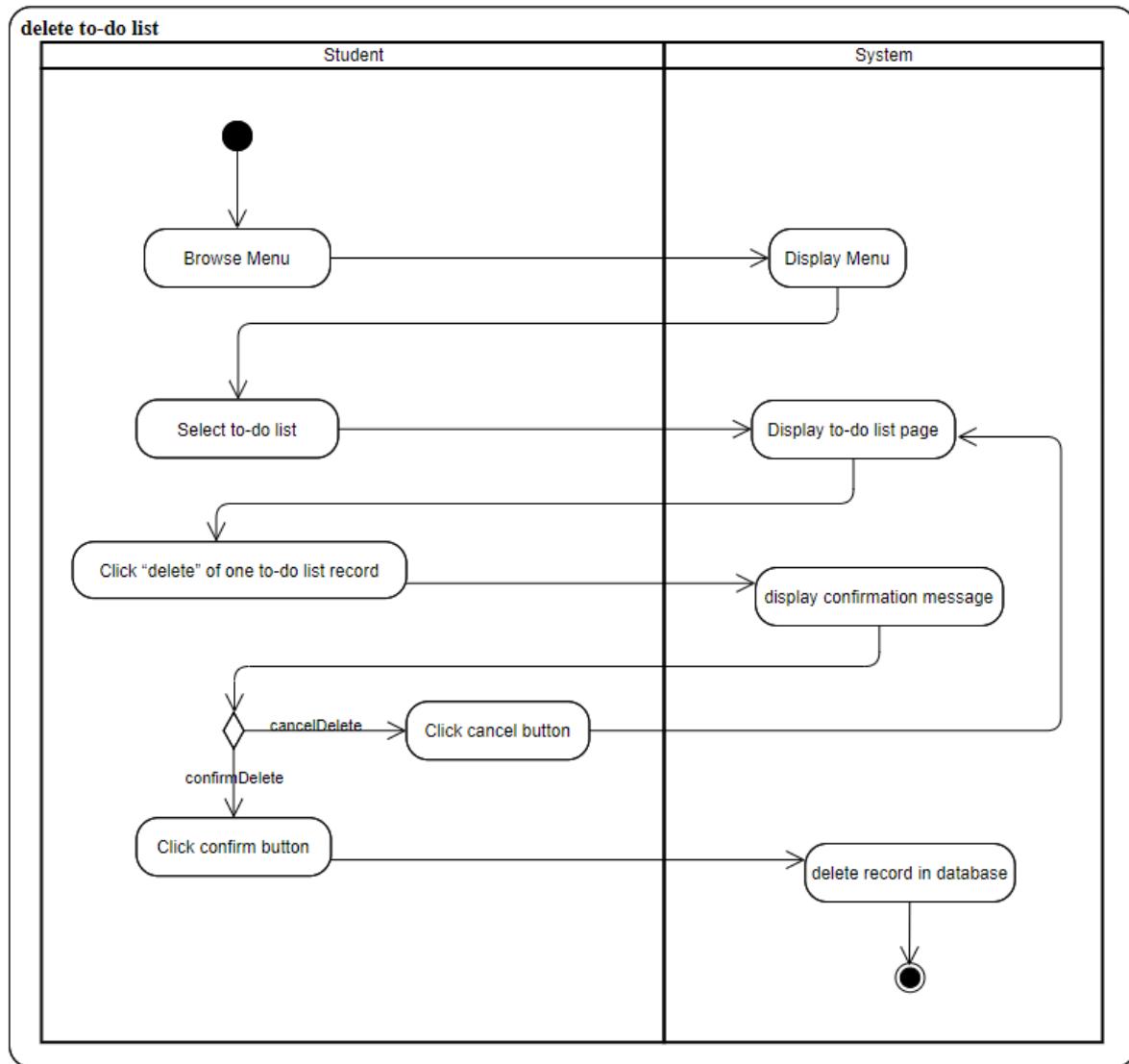


Diagram 4.14: Activity diagram for delete to-do list

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “to-do list”, the system will display the to-do list page. The student can click “delete” of one to-do list record and the system will display a confirmation message. Then, student can click the confirm button when the student confirms to delete the record. Then, the system will delete the record in the database. However, if a student decides to cancel the delete function by clicking the cancel button, the system will redirect the student to display the to-do list page.

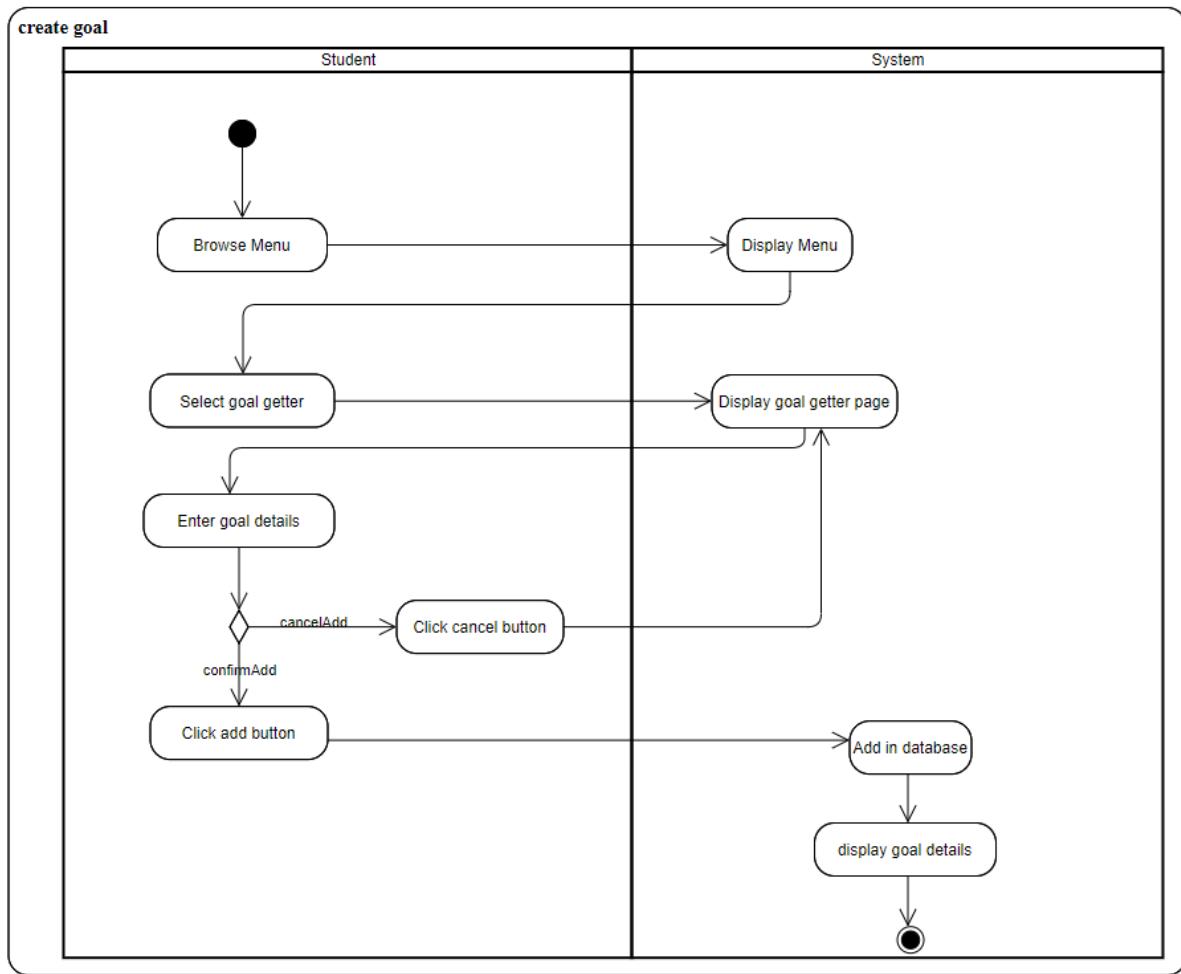


Diagram 4.15: Activity diagram for create goal

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “goal getter”, the system will display the goal getter page. The student can proceed to enter goal details and click the add button when the student confirms to add a goal. Then, the system will add the details in the database and display the goal details to student. However, if a student decides to cancel the add function by clicking the cancel button, the system will redirect the student to display the goal getter page.

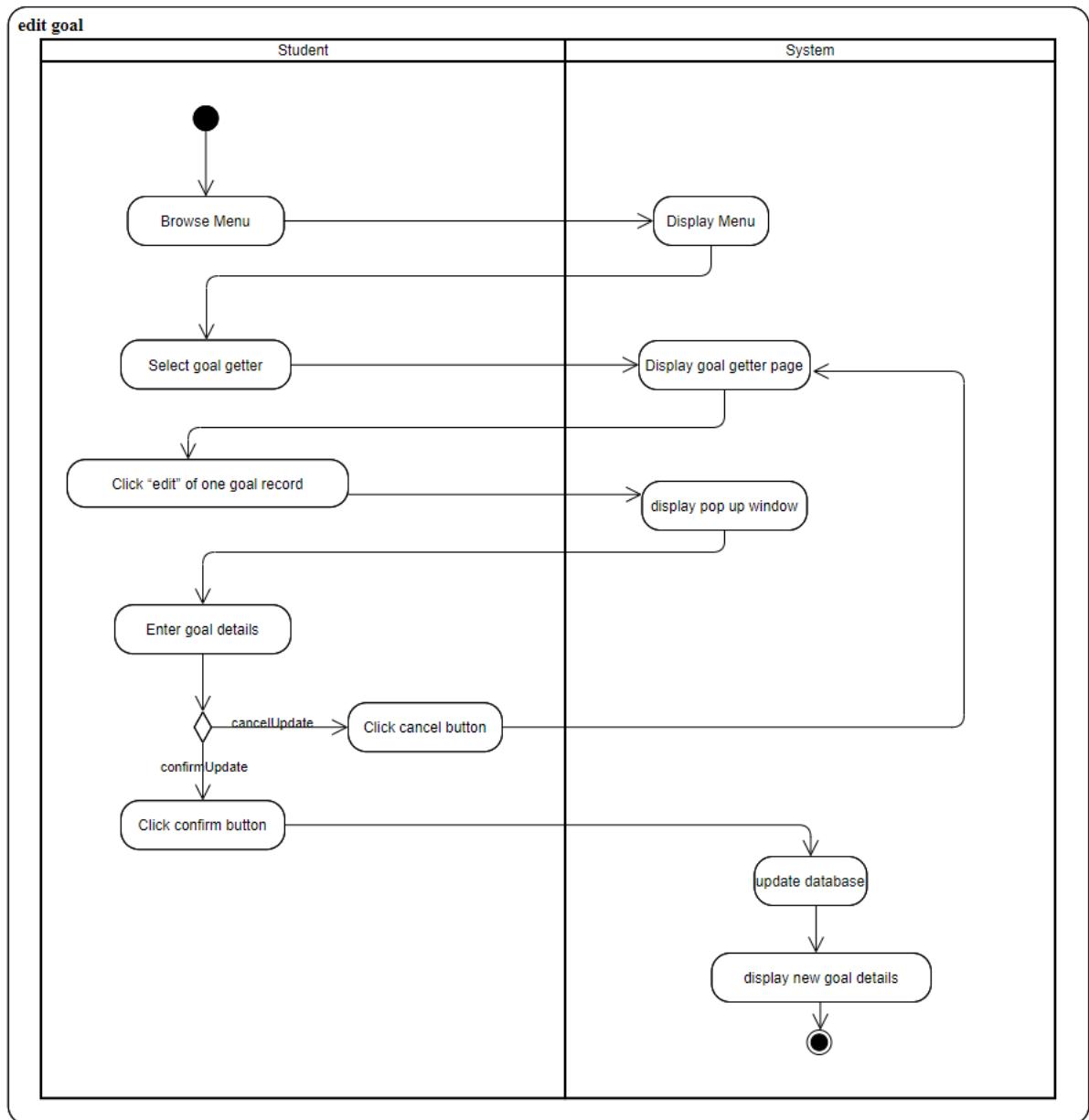


Diagram 4.16: Activity diagram for edit goal

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “goal getter”, the system will display the goal getter page. The student can click “edit” of one goal record and the system will display a pop up window to enter edit details. Then, student can proceed to enter details that need to be updated and click the confirm button when the student confirms to update. Then, the system will update the record in the database and display the new goal details to student. However, if a student decides to cancel the update function by clicking the cancel button, the system will redirect the student to display the goal getter page.

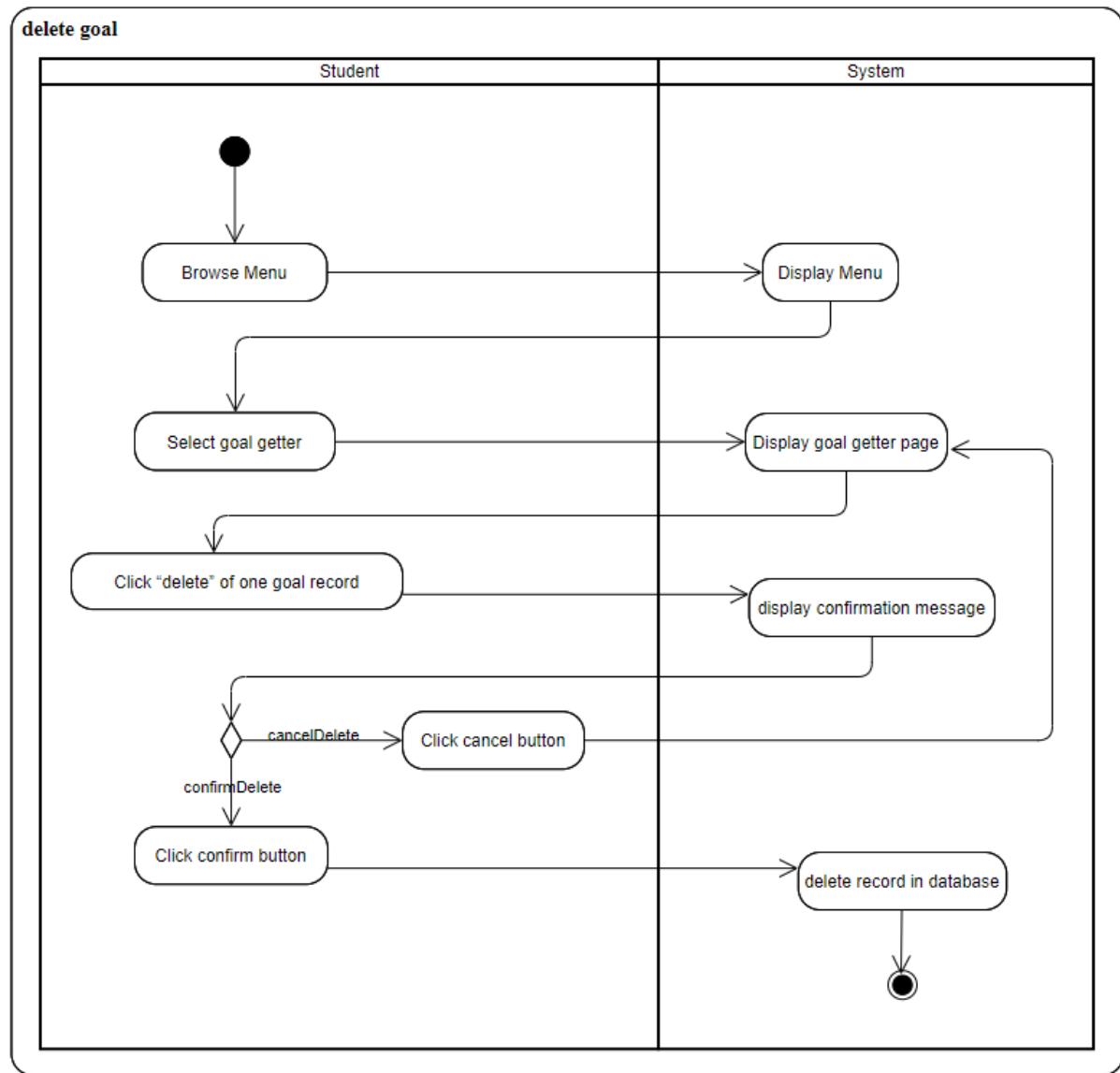


Diagram 4.17: Activity diagram for delete goal

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “goal getter”, the system will display the goal getter page. The student can click “delete” of one goal record and the system will display a confirmation message. Then, student can click the confirm button when the student confirms to delete the record. Then, the system will delete the record in the database. However, if a student decides to cancel the delete function by clicking the cancel button, the system will redirect the student to display the goal getter page.

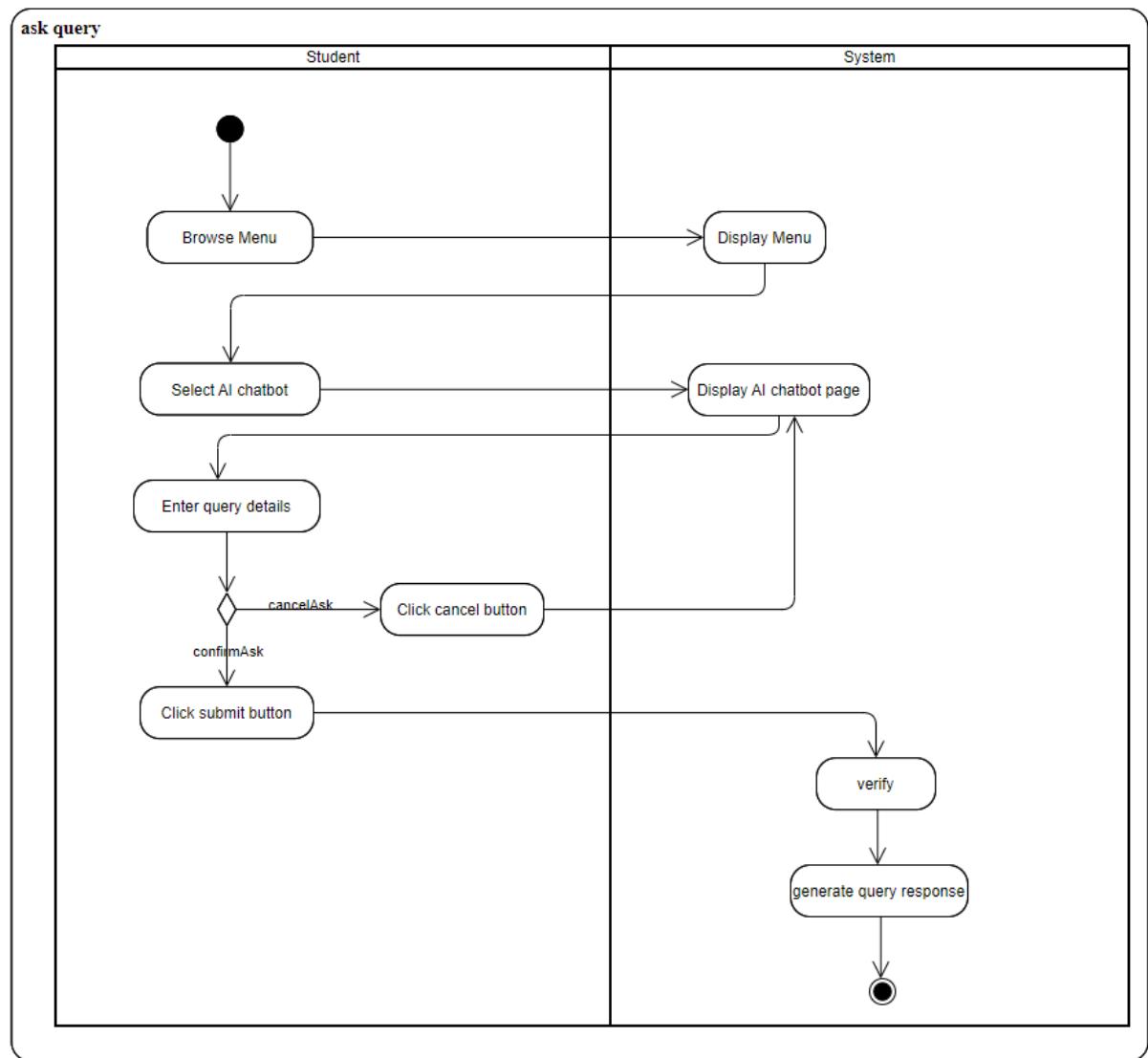


Diagram 4.18: Activity diagram for ask query

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “AI chatbot”, the system will display the AI chatbot page. The student can proceed to enter query details and click the submit button when the student confirms to ask the query. Then, the system will verify and generate query responses. However, if a student decides to cancel the ask function by clicking the cancel button, the system will redirect the student to display the AI chatbot page.

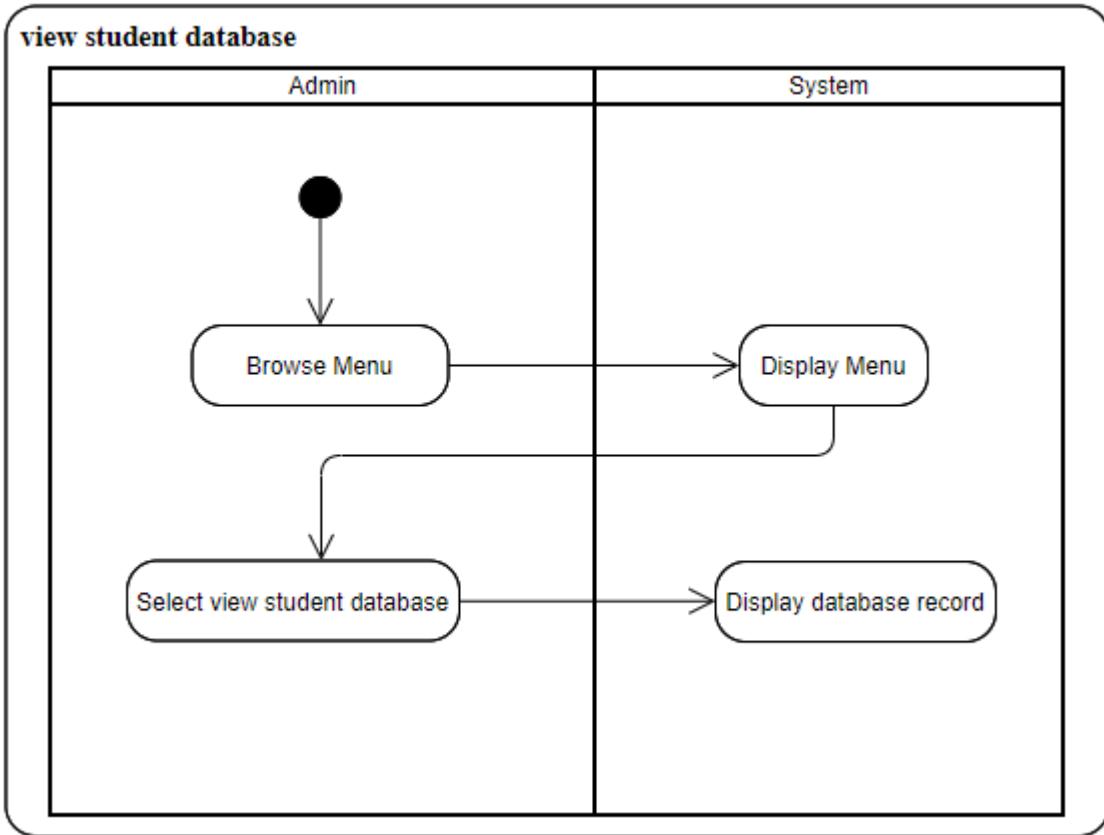


Diagram 4.19: Activity diagram for view student database

The admin begins by browsing the main menu, the system will then display the main menu options. Admin then selects "view student database", the system will display the database record.

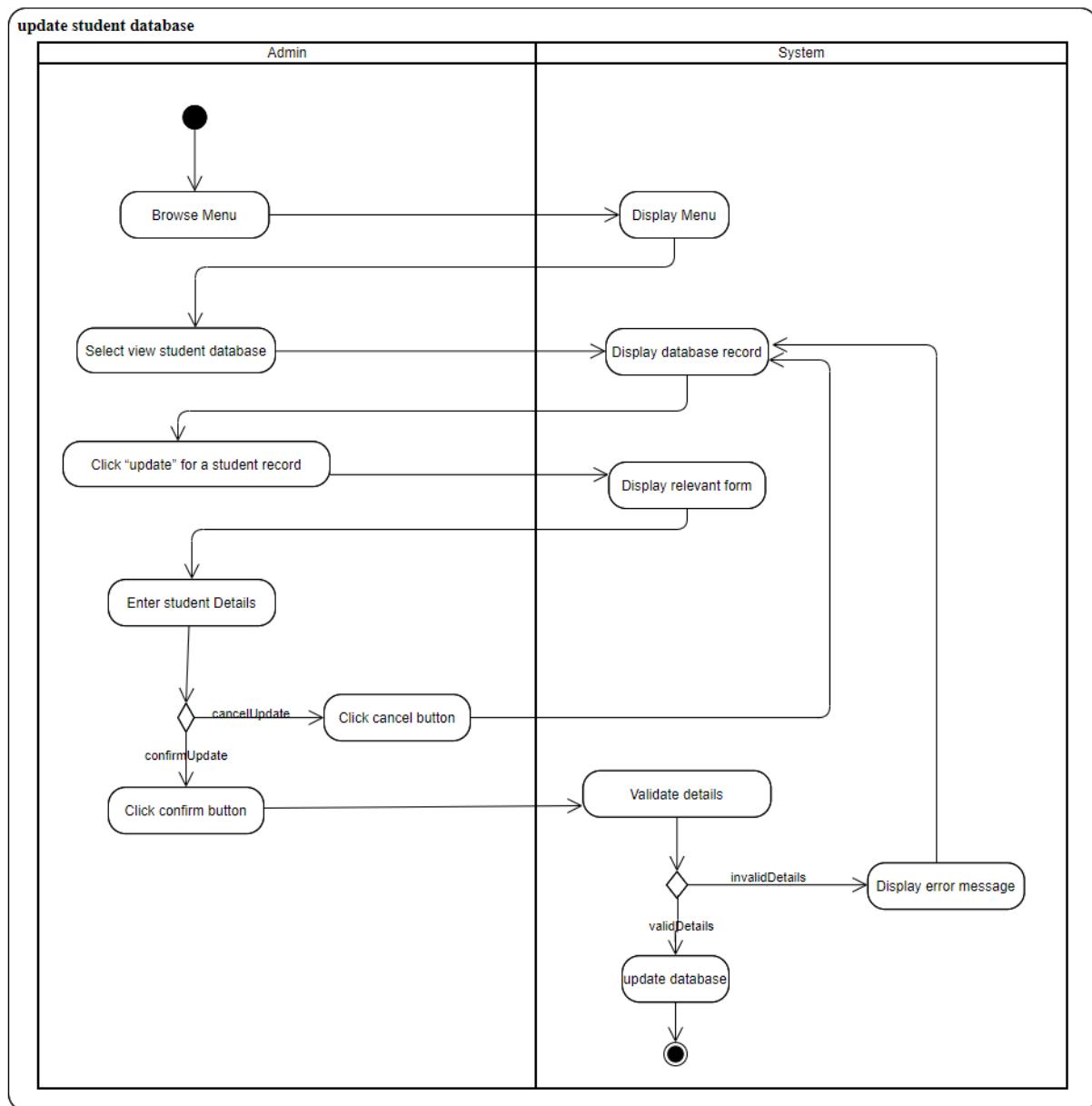


Diagram 4.20: Activity diagram for update student database

The admin begins by browsing the main menu, the system will then display the main menu options. Admin then selects "view student database", the system will display the database record. The admin can click "update" for a particular student details record and the system will then display relevant form to update. Then, the admin can proceed to enter student details that need to be updated and click the confirm button when the admin confirms to update. Then, the system validates the input details, if details are correct, the system will update the record in the database. However, if an admin decides to cancel the update function by clicking the cancel button, the system will redirect the admin to display the database record page. Besides, if the input details are incorrect, the system will display error messages and redirect the admin to display the database record page to re-enter again.

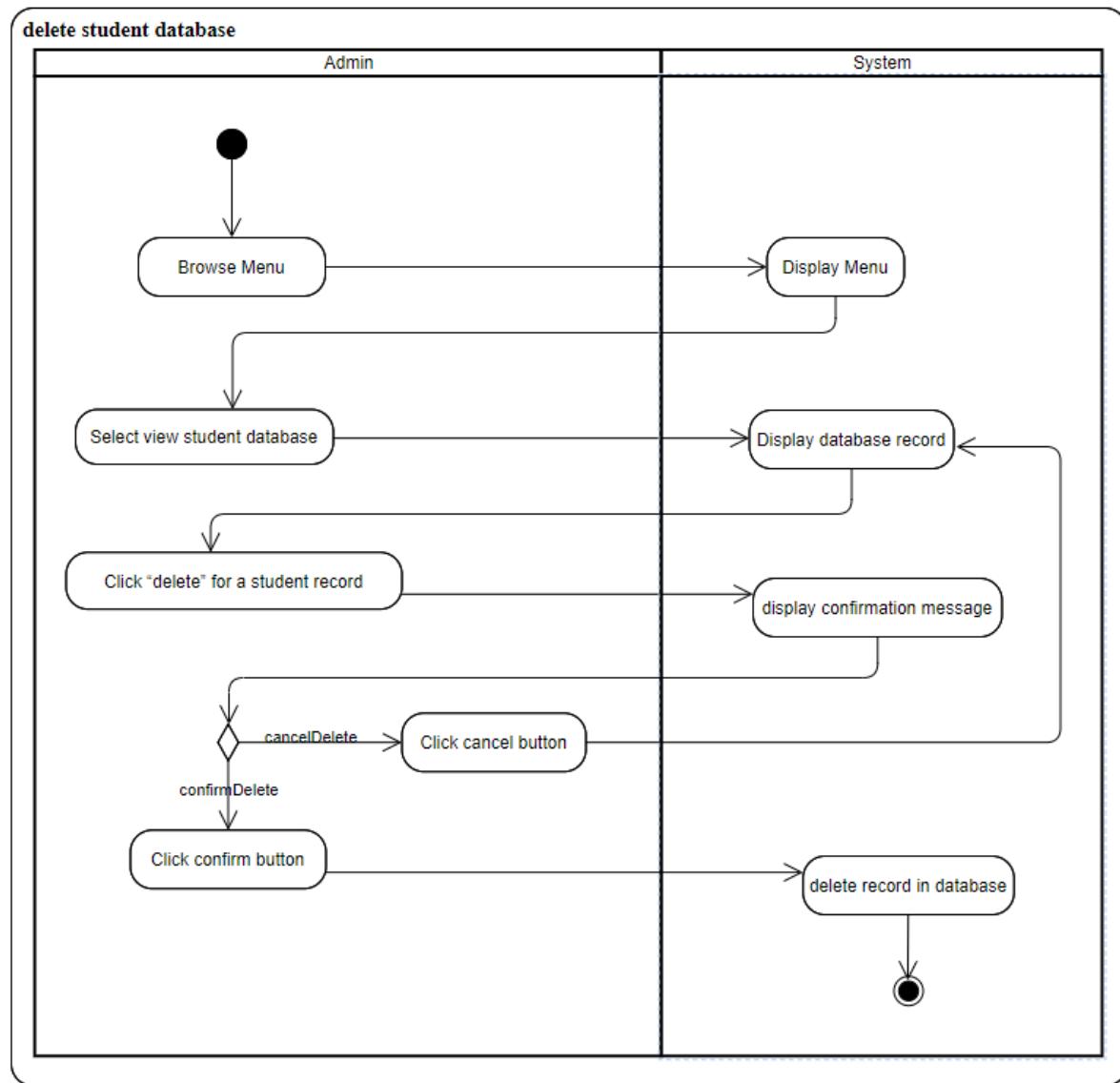


Diagram 4.21: Activity diagram for delete student database

The admin begins by browsing the main menu, the system will then display the main menu options. Admin then selects "view student database", the system will display the database record. The admin can click "delete" for a particular student record and the system will display a confirmation message. Then, the admin can click the confirm button when the admin confirms to delete the record. Then, the system will delete the record in the database. However, if an admin decides to cancel the delete function by clicking the cancel button, the system will redirect the admin to display the database record page.

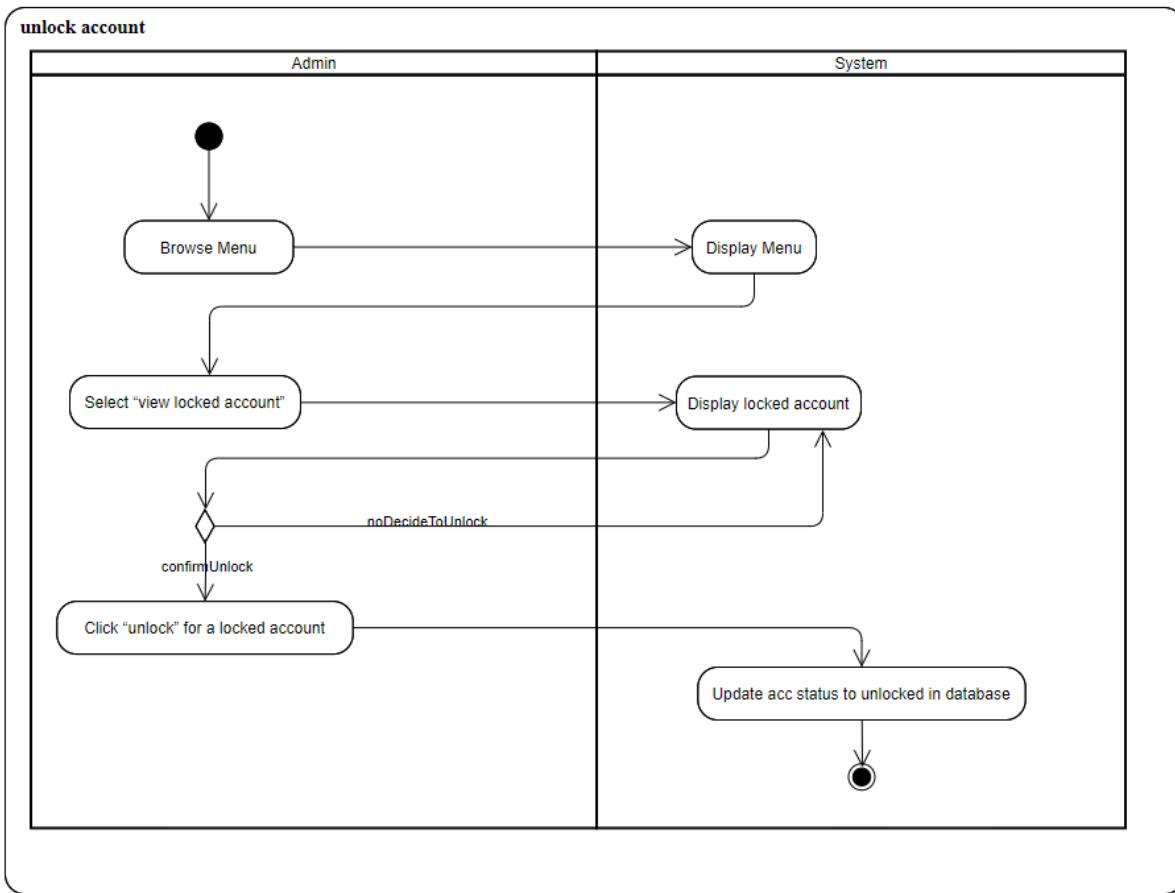


Diagram 4.22: Activity diagram for unlock account

The admin begins by browsing the main menu, the system will then display the main menu options. Admin then selects “view locked account” the system will display the locked account. The admin can click “unlock” for a particular locked account and then the system will update account status to ‘unlocked’ in the database for that record. However, if an admin decides not to unlock the lock account, the system will redirect the admin to display the locked account page.

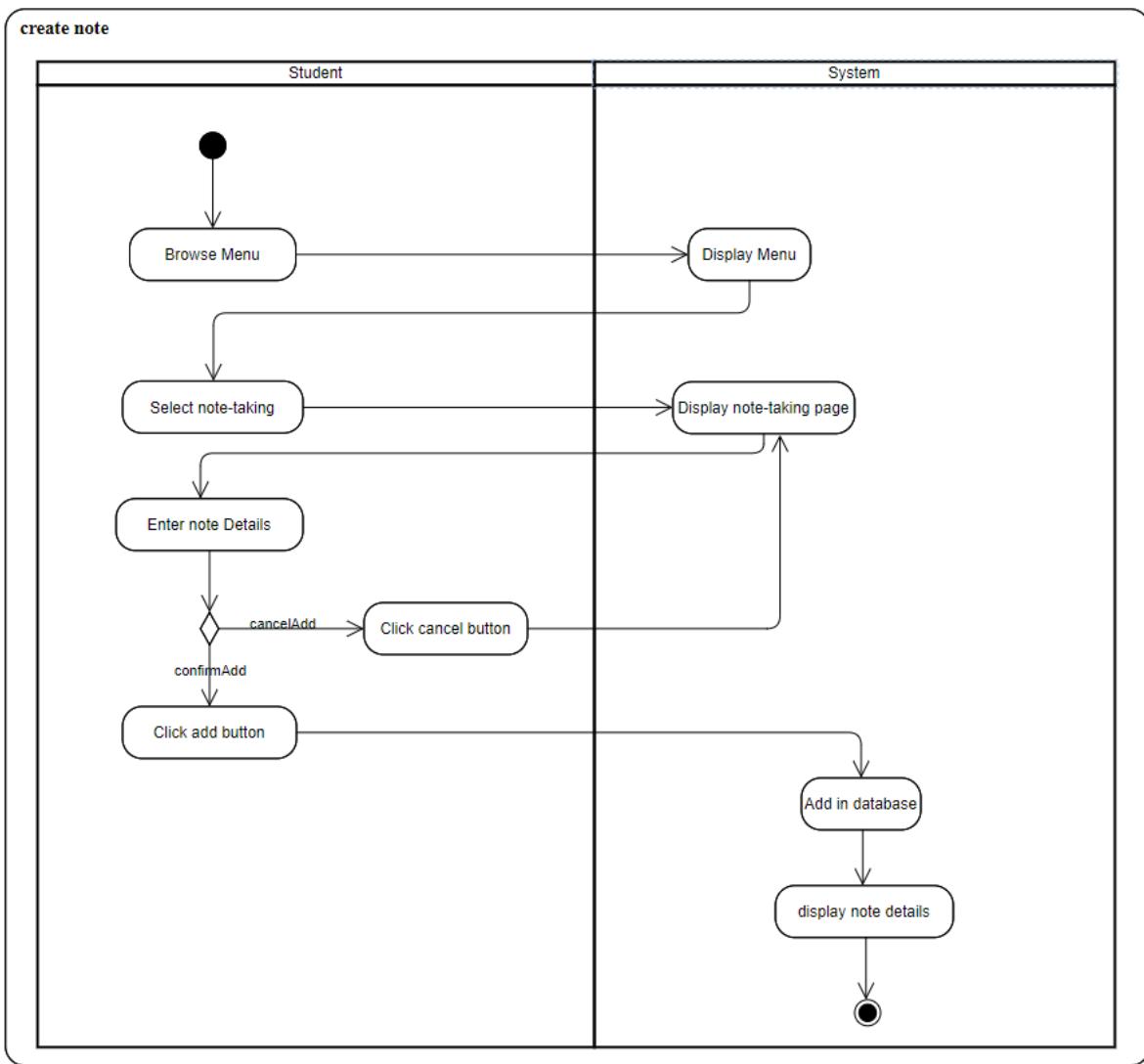


Diagram 4.23: Activity diagram for create note

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “note-taking”, the system will display the note-taking page. The student can proceed to enter note details and click the add button when the student confirms to add a note. Then, the system will add the details in the database and display the note details to student. However, if a student decides to cancel the add function by clicking the cancel button, the system will redirect the student to display the note-taking page.

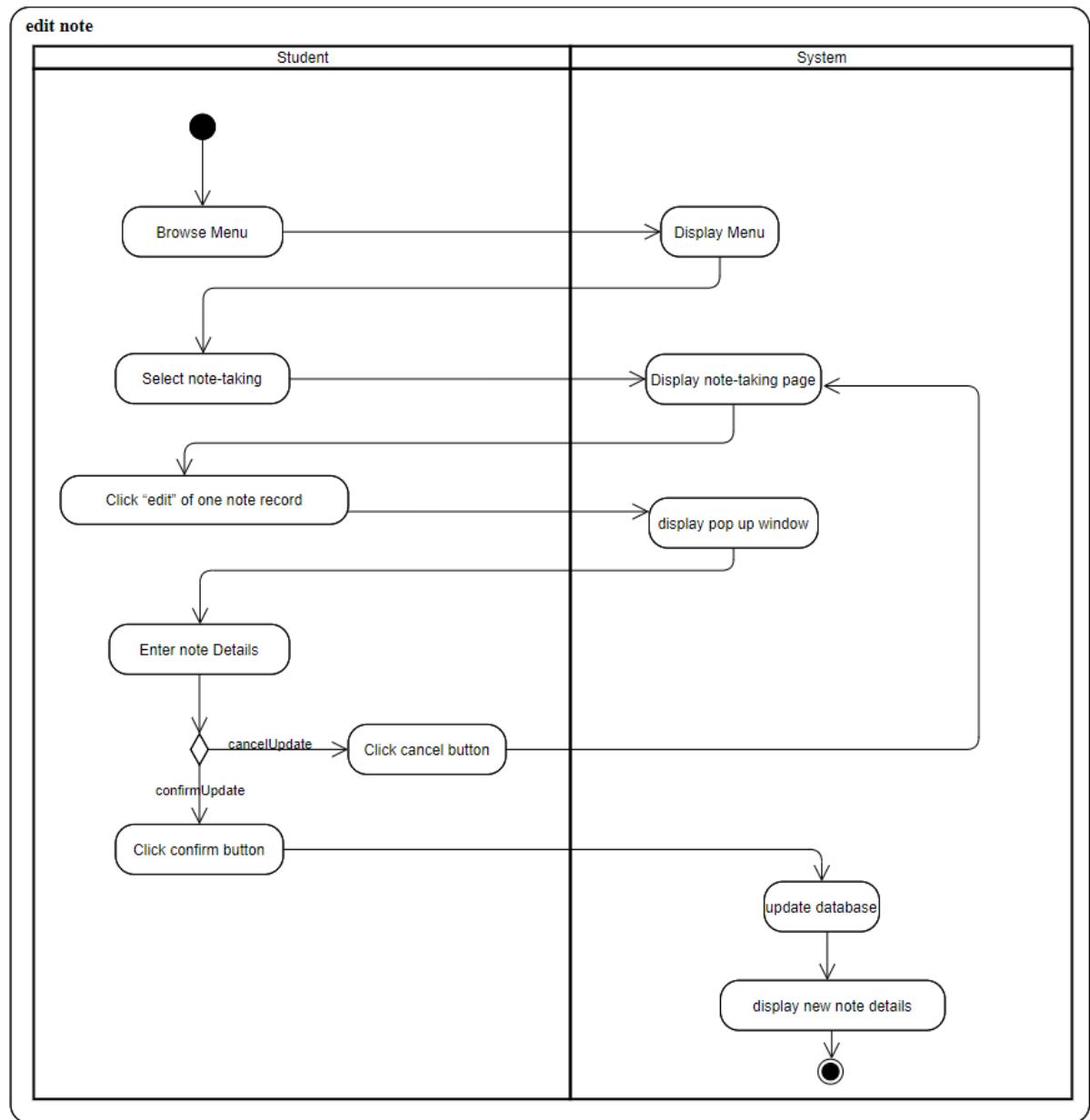


Diagram 4.24: Activity diagram for edit note

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “note-taking”, the system will display the note-taking page. The student can click “edit” of one note record and the system will display a pop up window to enter edit details. Then, student can proceed to enter details that need to be updated and click the confirm button when the student confirms to update. Then, the system will update the record in the database and display the new note details to student. However, if a student decides to cancel the update function by clicking the cancel button, the system will redirect the student to display the note-taking page.

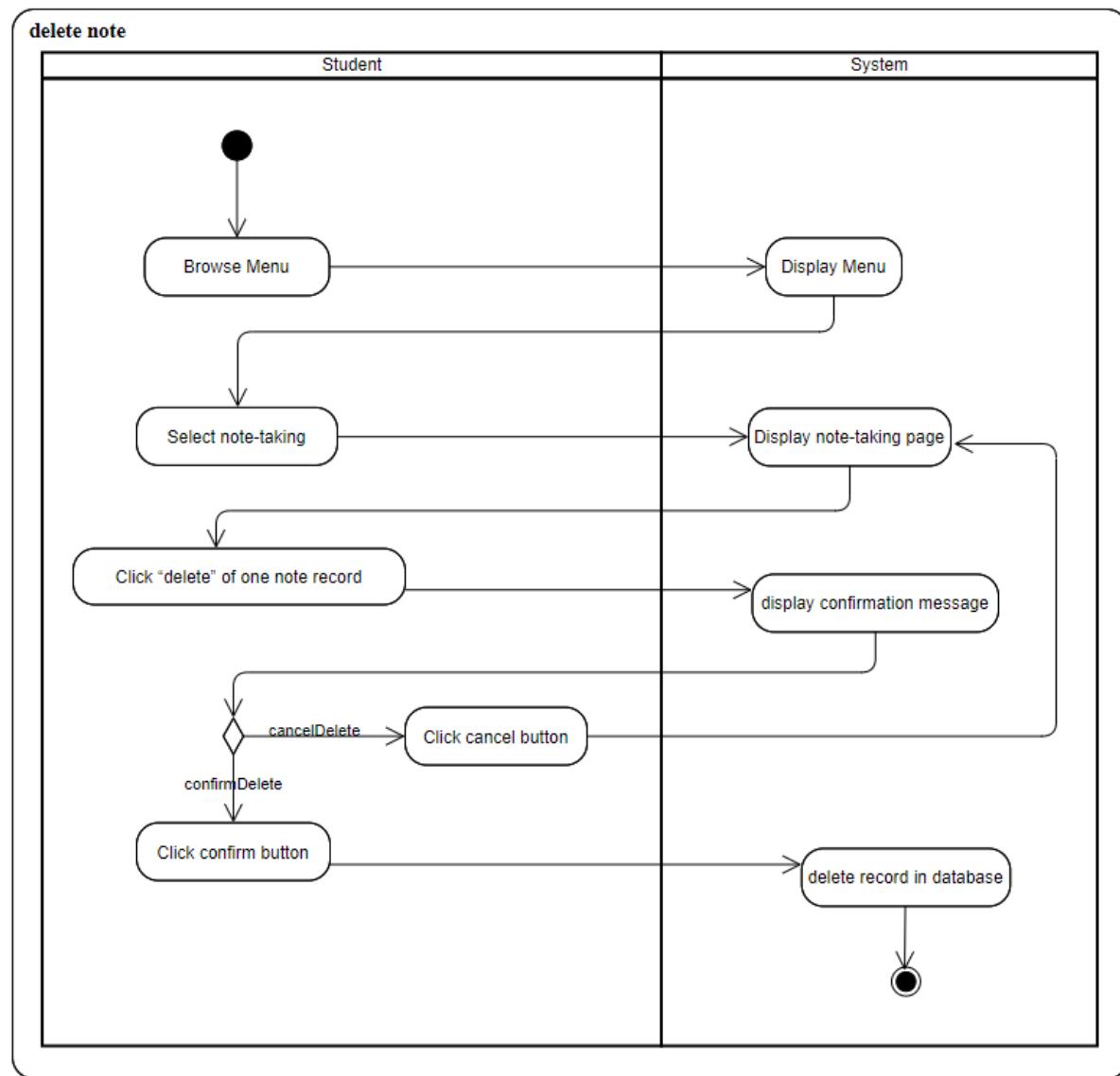


Diagram 4.25: Activity diagram for delete note

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “note-taking”, the system will display the note page. The student can click “delete” of one note record and the system will display a confirmation message. Then, student can click the confirm button when the student confirms to delete the record. Then, the system will delete the record in the database. However, if a student decides to cancel the delete function by clicking the cancel button, the system will redirect the student to display the note-taking page.

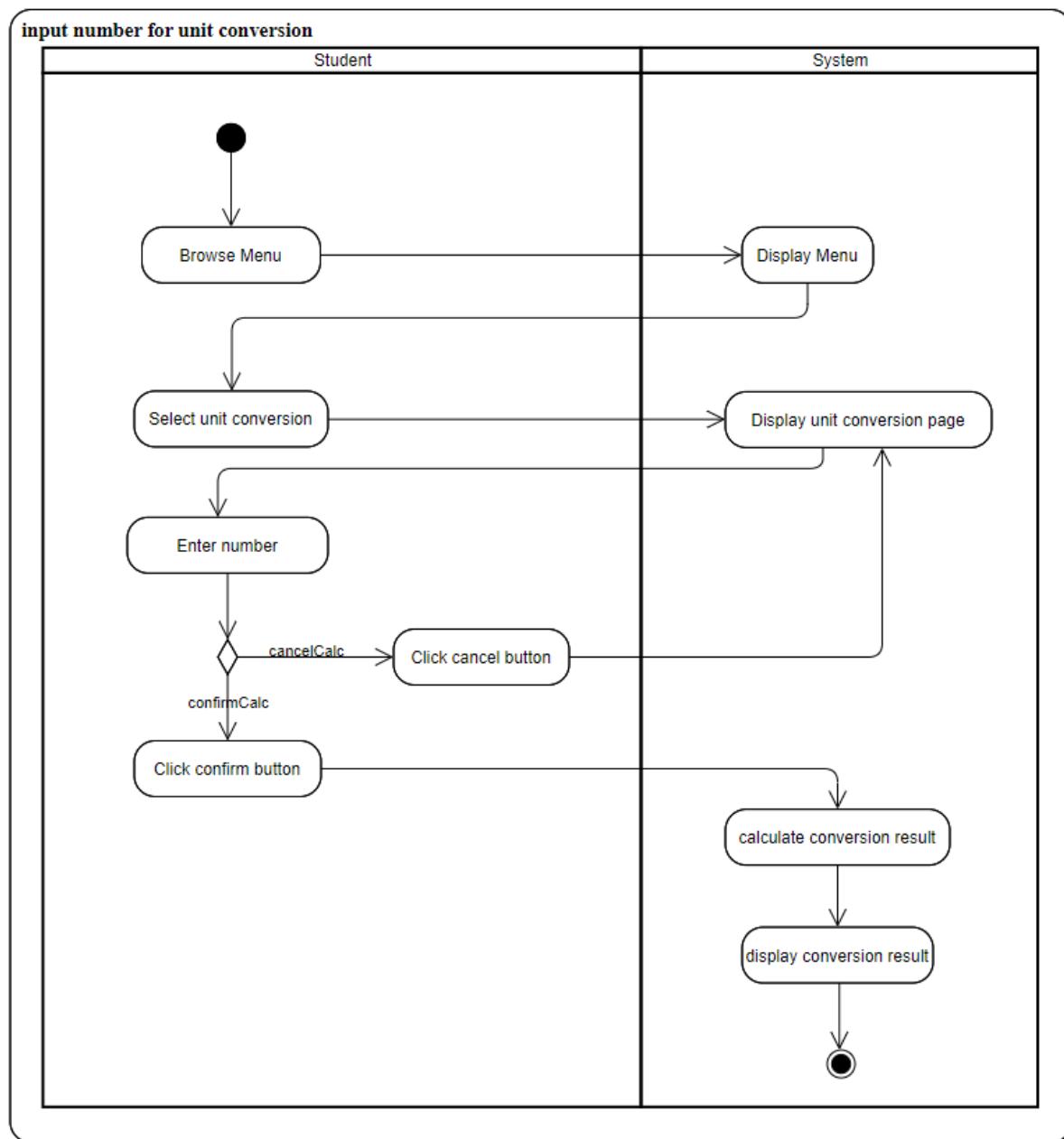


Diagram 4.26: Activity diagram for input number for unit conversion

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “unit conversion”, the system will display the unit conversion page. The student can proceed to enter a number and click the confirm button when the student confirms to calculate the conversion result. Then, the system will calculate the conversion result and display the conversion result to the student. However, if a student decides to cancel the calculated conversion function by clicking the cancel button, the system will redirect the student to display the unit conversion page.

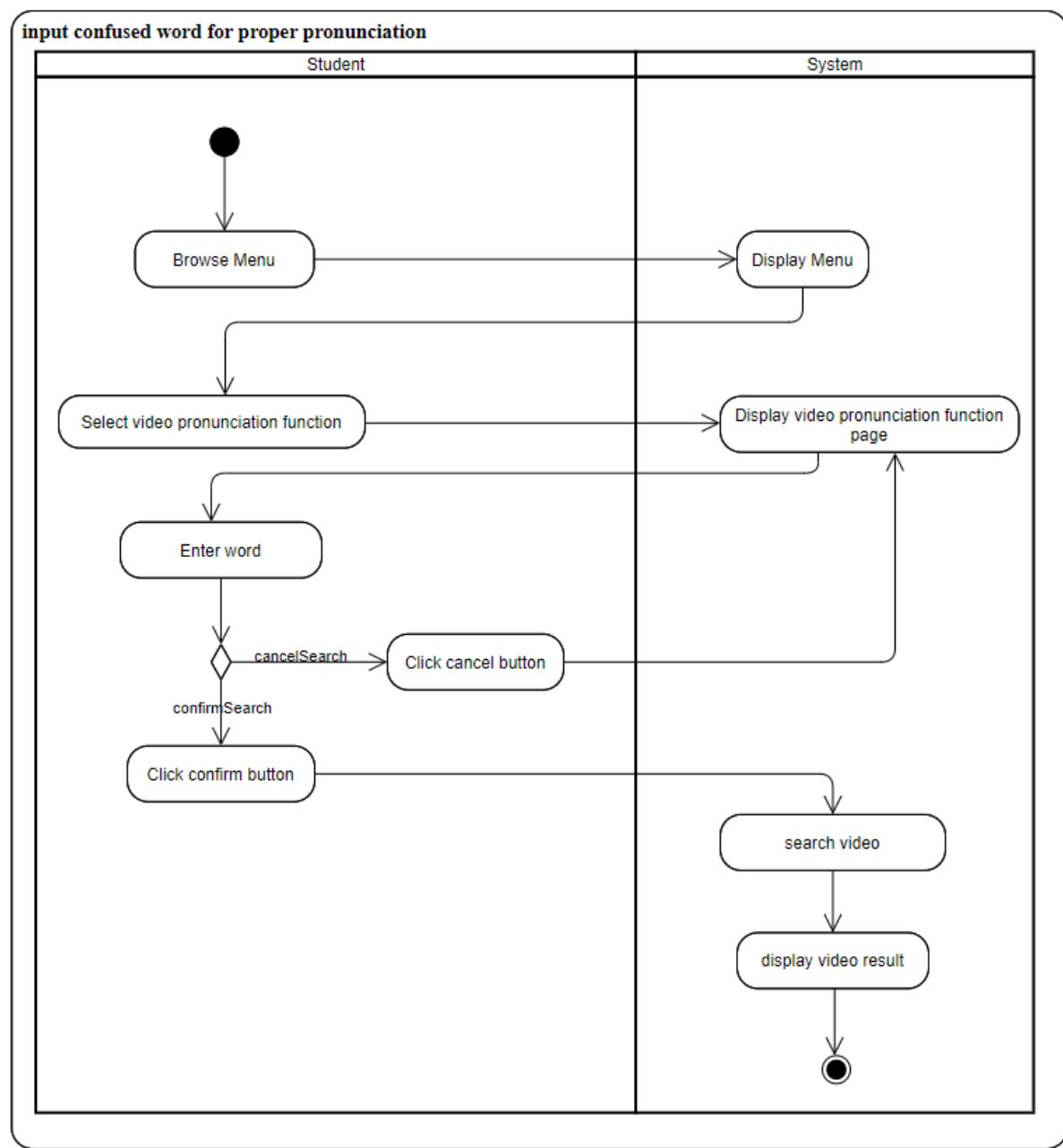


Diagram 4.27: Activity diagram for input confused word for proper pronunciation

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “video pronunciation function”, the system will display the video pronunciation function page. The student can proceed to enter a word and click the confirm button when the student confirms to search the pronunciation teaching video. Then, the system will search YouTube videos and display the video result to the student. However, if a student decides to cancel the searching function by clicking the cancel button, the system will redirect the student to display the video pronunciation function page.

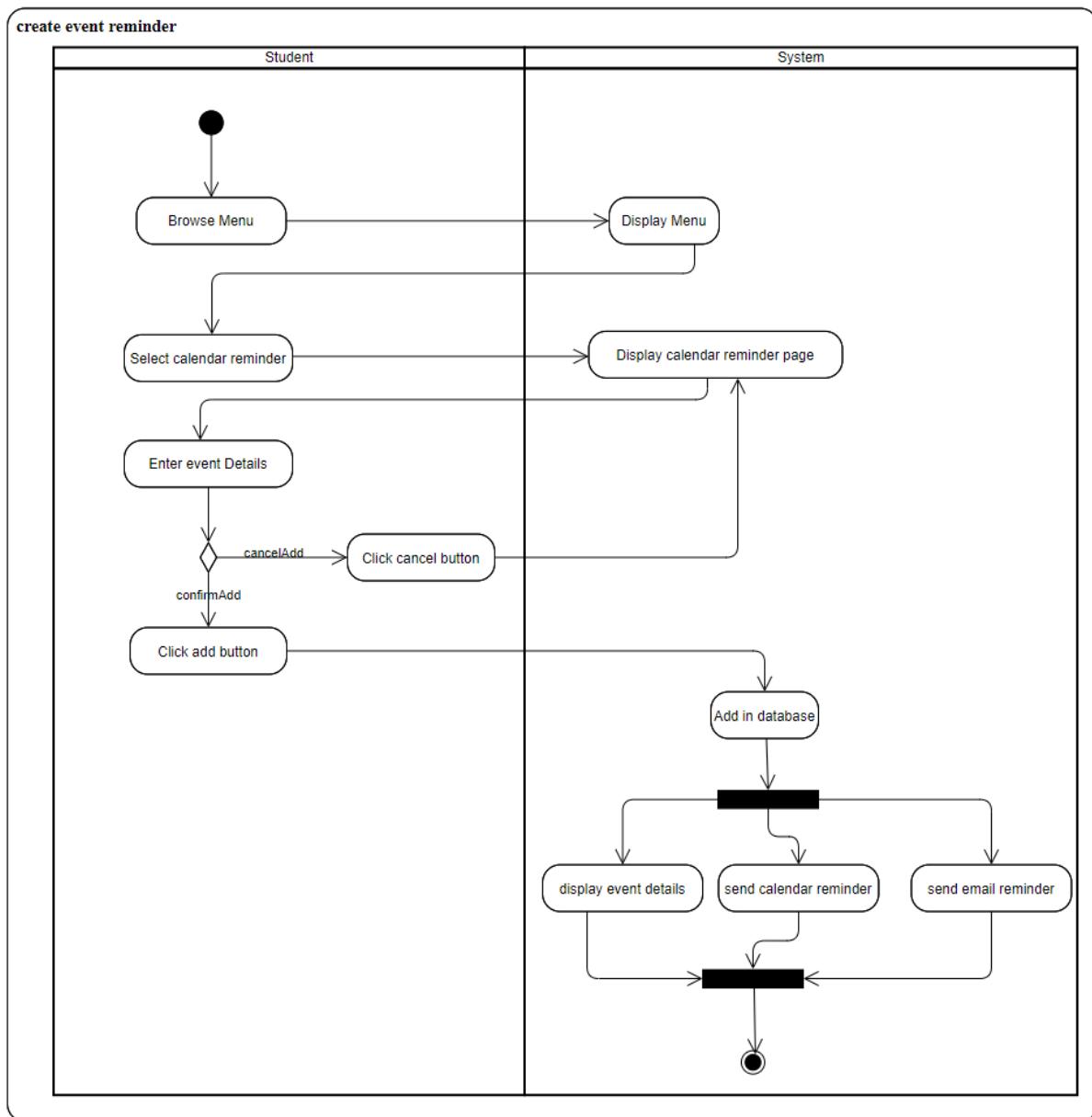


Diagram 4.28: Activity diagram for create event reminder

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “calendar reminder”, the system will display the calendar reminder page. The student can proceed to enter event details and click the add button when the student confirms to add an event. Then, the system will add the details in the database. Then, the system will display the event details to student and send reminders to Google Calendar and email address concurrently. However, if a student decides to cancel the add function by clicking the cancel button, the system will redirect the student to display the calendar reminder page.

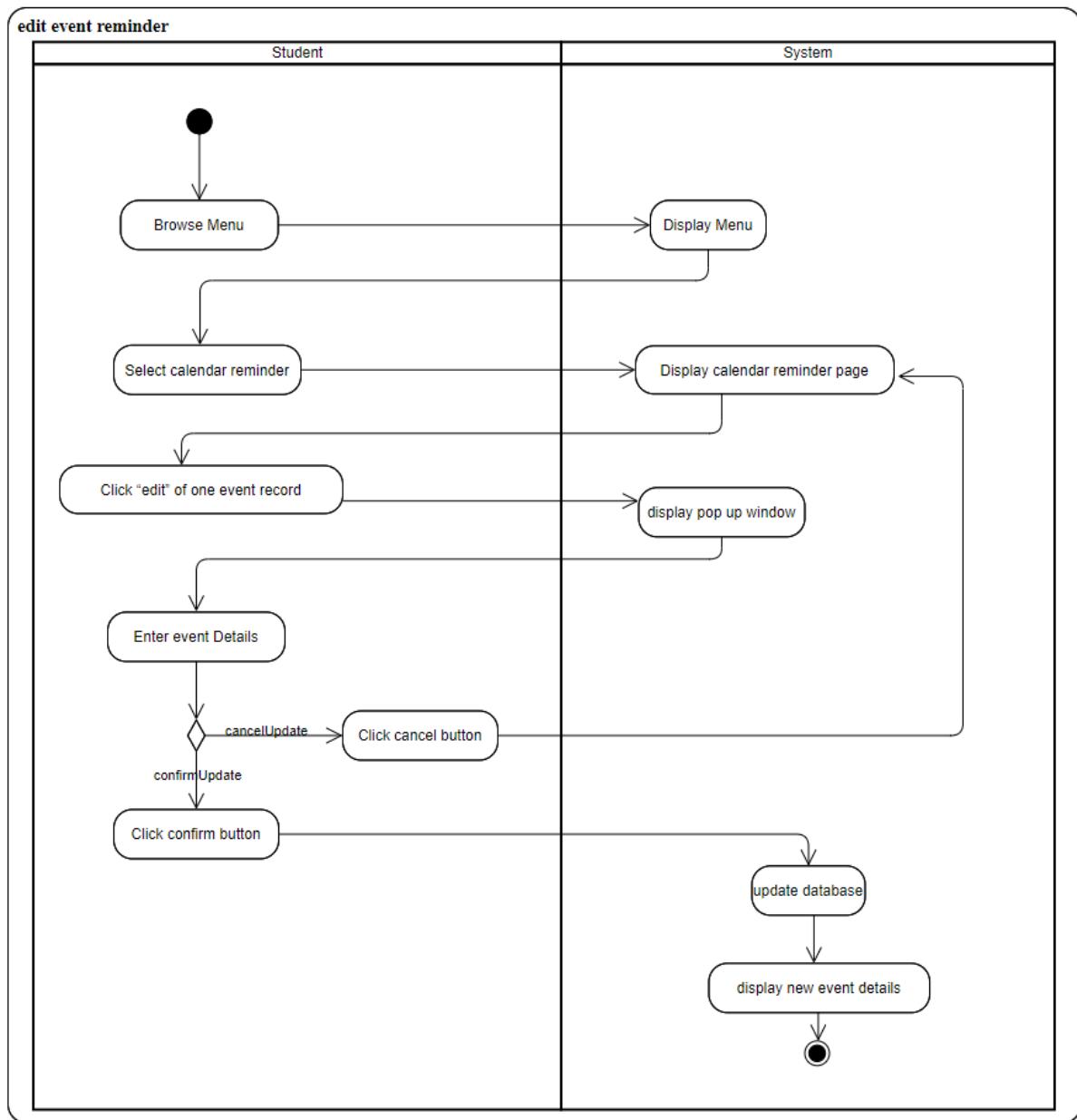


Diagram 4.29: Activity diagram for edit event reminder

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “calendar reminder”, the system will display the calendar reminder page. The student can click “edit” of one event record and the system will display a pop up window to enter edit details. Then, student can proceed to enter details that need to be updated and click the confirm button when the student confirms to update. Then, the system will update the record in the database and display the new event details to student. However, if a student decides to cancel the update function by clicking the cancel button, the system will redirect the student to display the calendar reminder page.

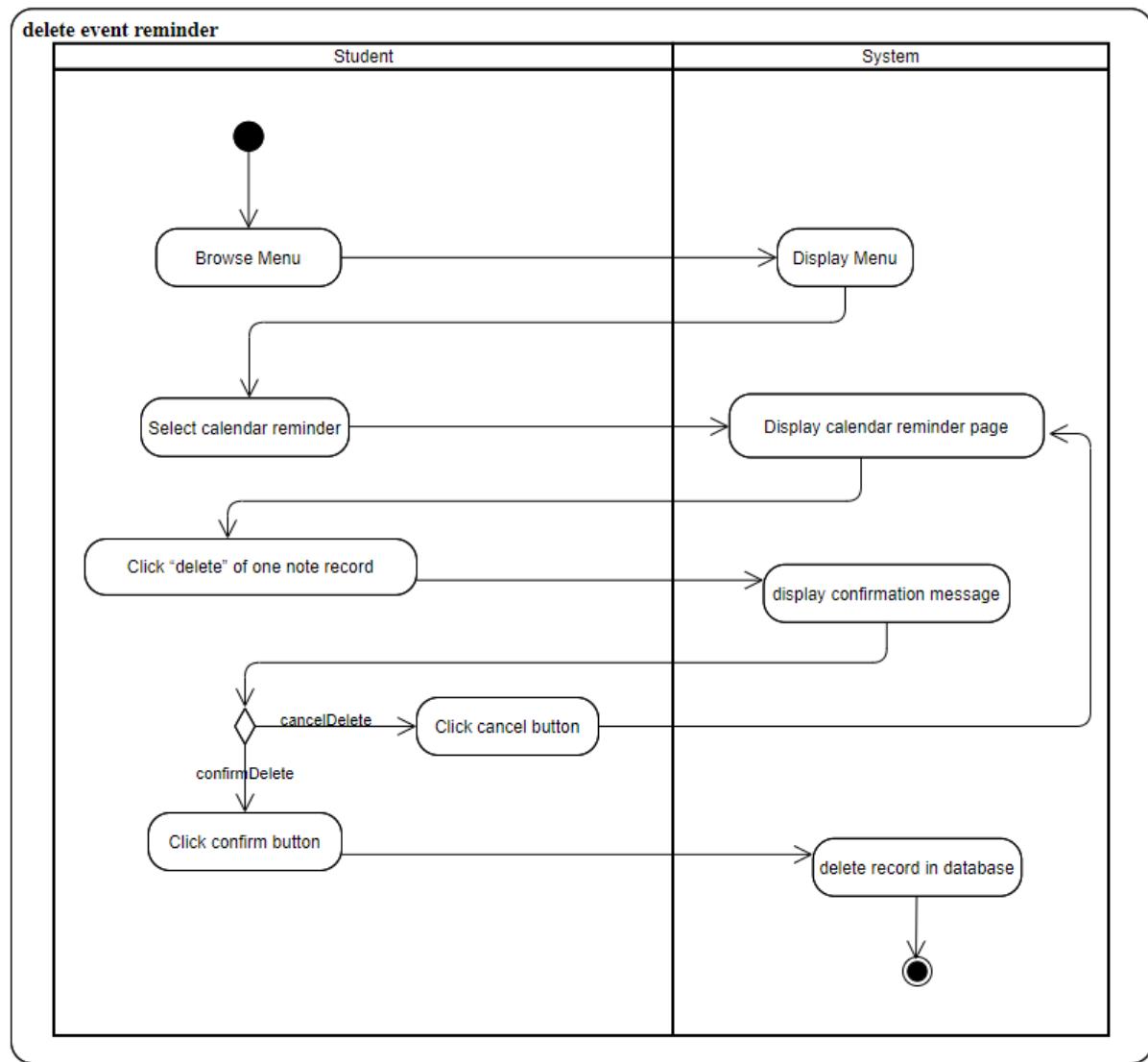


Diagram 4.30: Activity diagram for delete event reminder

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “calendar reminder”, the system will display the calendar reminder page. The student can click “delete” of one event record and the system will display a confirmation message. Then, student can click the confirm button when the student confirms to delete the record. Then, the system will delete the record in the database. However, if a student decides to cancel the delete function by clicking the cancel button, the system will redirect the student to display the calendar reminder page.

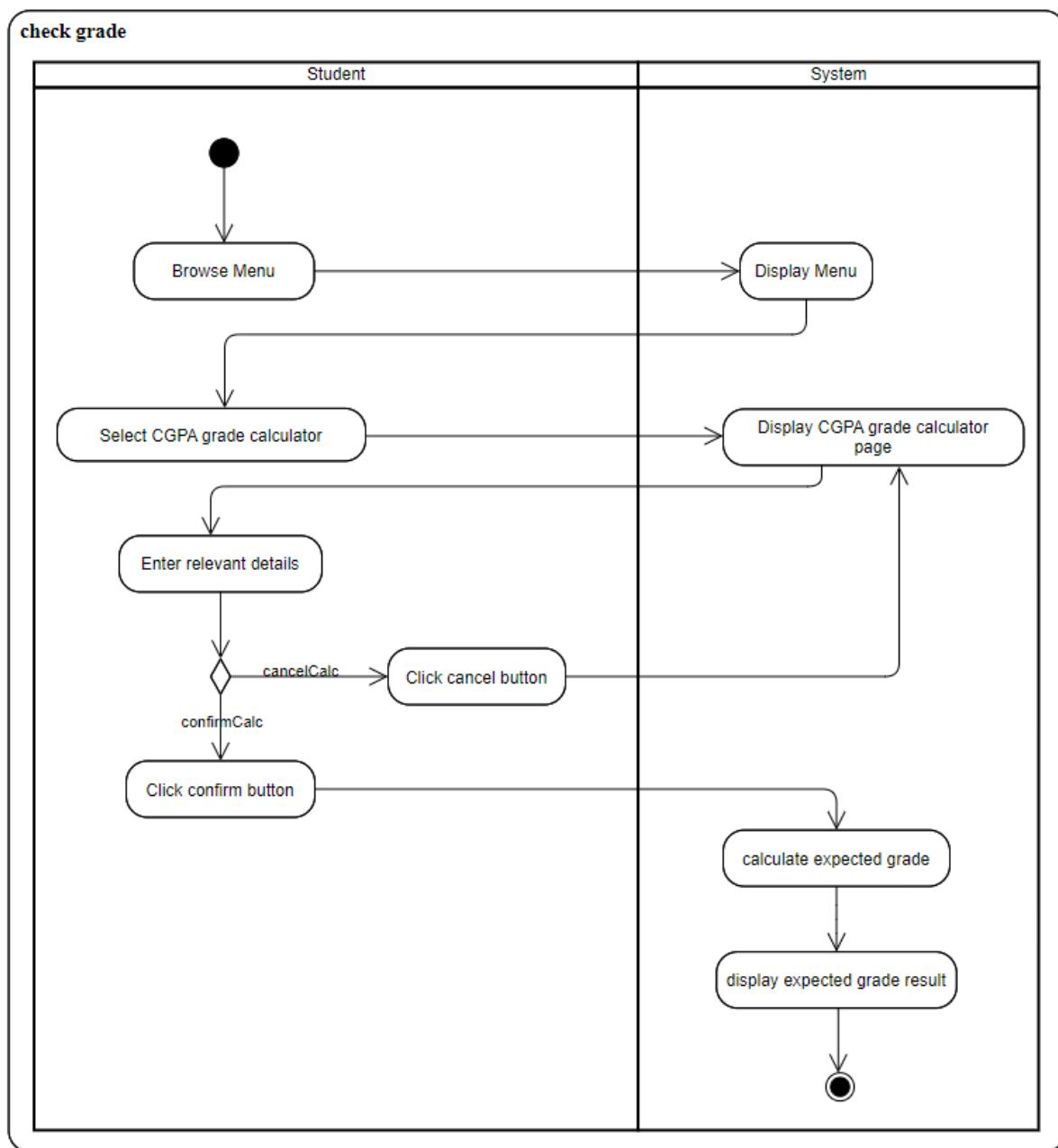


Diagram 4.31: Activity diagram for check grade

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “CGPA grade calculator”, the system will display the CGPA grade calculator page. The student can proceed to enter relevant details and click the confirm button when the student confirms to calculate expected grade. Then, the system will calculate expected grades and display the expected grade result to student. However, if a student decides to cancel the calculate function by clicking the cancel button, the system will redirect the student to display the CGPA grade calculator page.

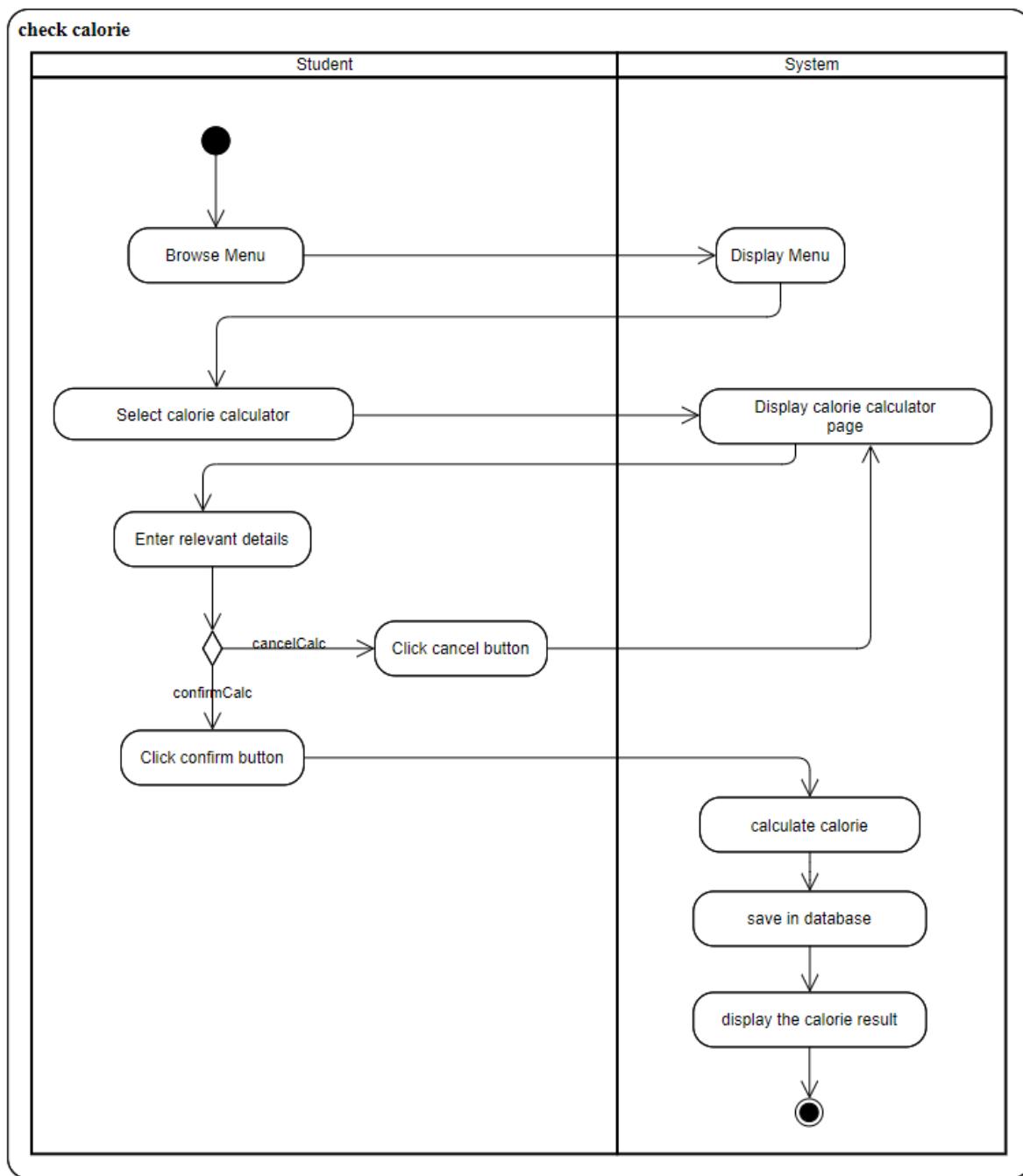


Diagram 4.32: Activity diagram for check calorie

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “calorie calculator”, the system will display the calorie calculator page. The student can proceed to enter relevant details and click the confirm button when the student confirms to calculate calorie results. Then, the system will calculate calories, save calorie records in the database and display the calorie result to student. However, if a student decides to cancel the calculate function by clicking the cancel button, the system will redirect the student to display the calorie calculator page.

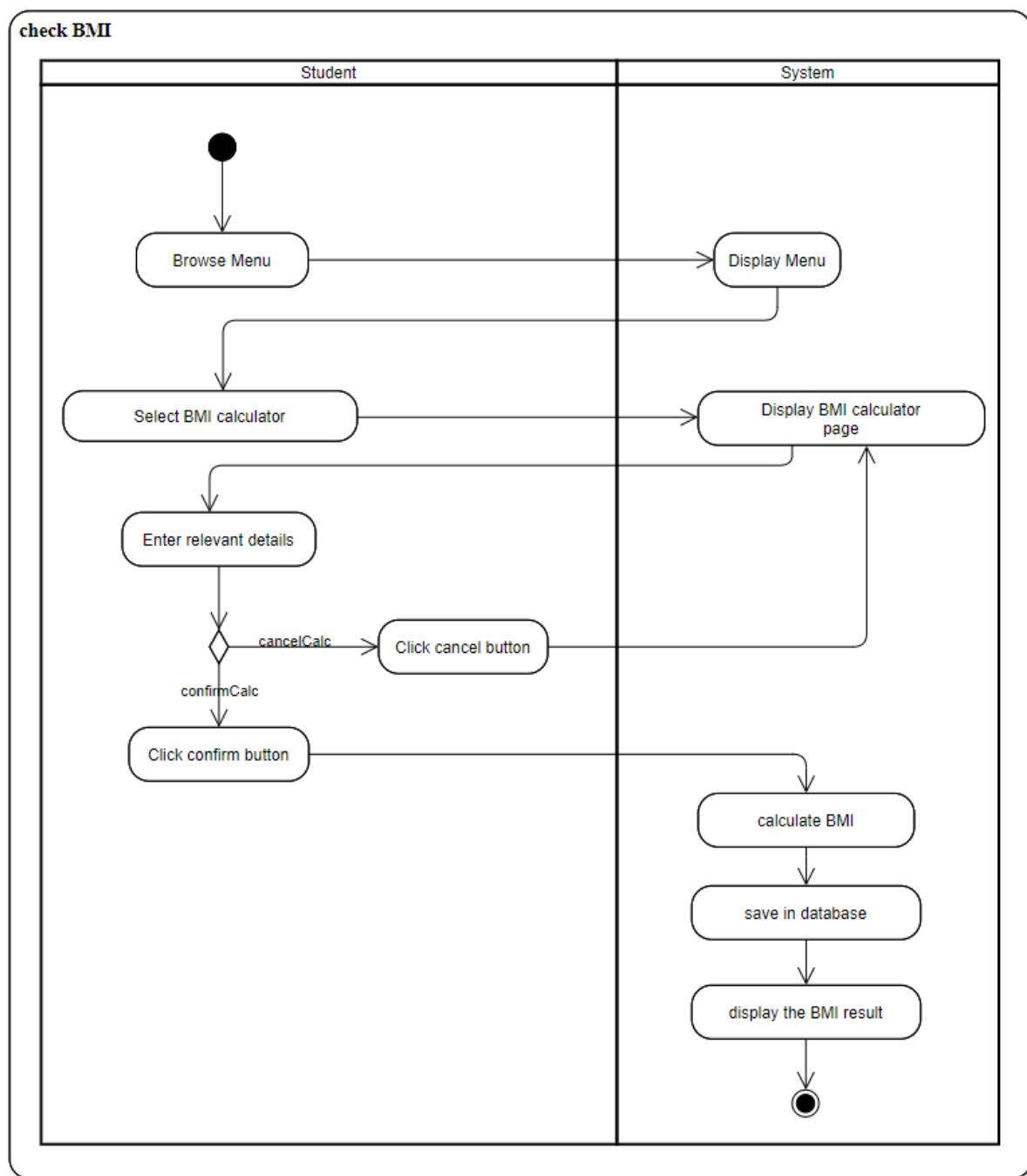


Diagram 4.33: Activity diagram for check BMI

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “BMI calculator”, the system will display the BMI calculator page. The student can proceed to enter relevant details and click the confirm button when the student confirms to calculate BMI results. Then, the system will calculate BMI, save BMI records in the database and display the BMI result to student. However, if a student decides to cancel the calculate function by clicking the cancel button, the system will redirect the student to display the BMI calculator page.

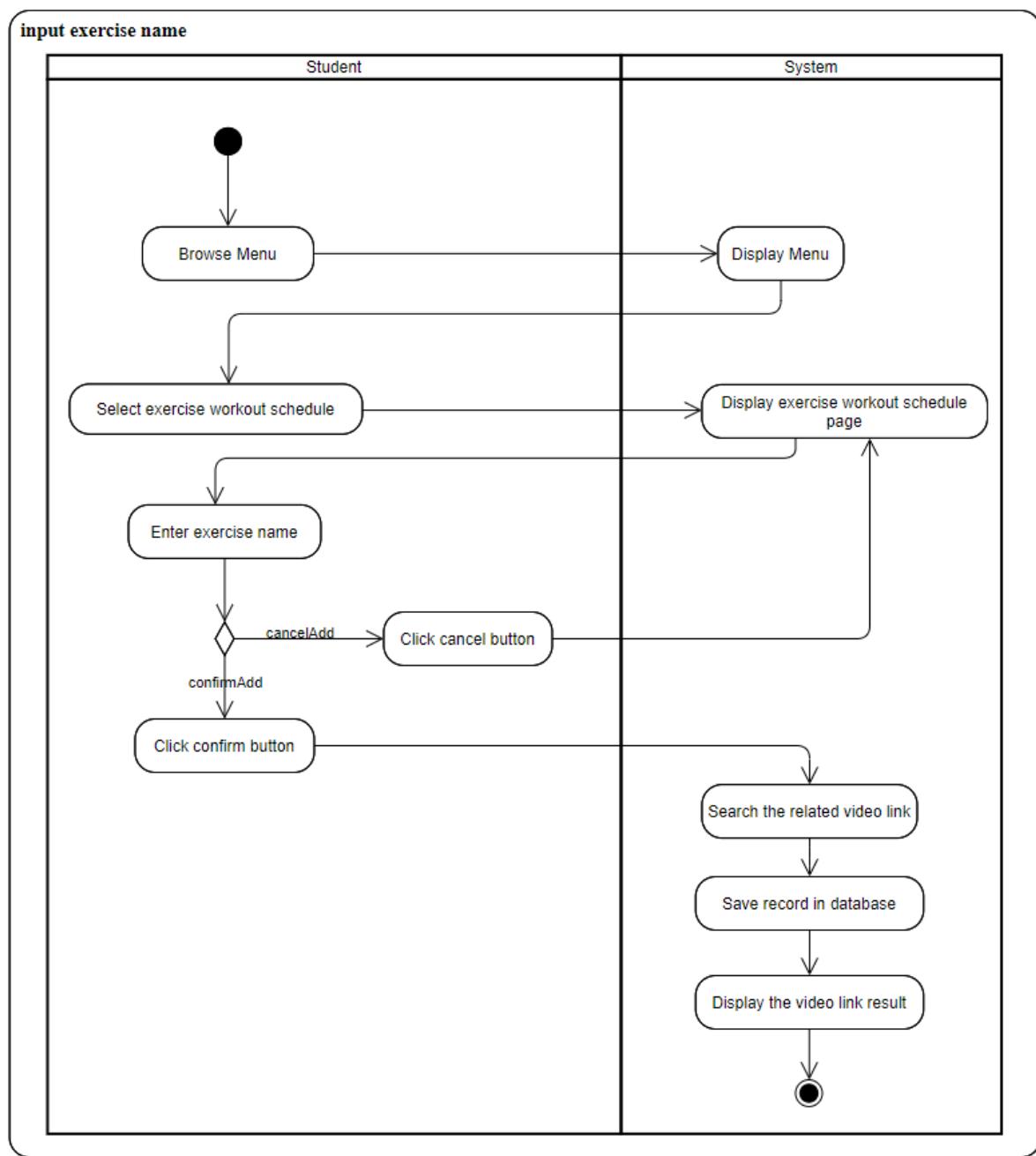


Diagram 4.34: Activity diagram for input exercise name

The student begins by browsing the main menu, the system will then display the main menu options. Student then selects “exercise workout schedule”, the system will display the exercise workout schedule page. The student can proceed to enter the exercise name and click the confirm button when the student confirms to add the exercise in schedule. Then, the system searches the related video link, saves the exercise record in the database and displays the video link result to the student. However, if a student decides to cancel the add function by clicking the cancel button, the system will redirect the student to display the exercise workout schedule page.

4.3 Entity Relationship Diagram

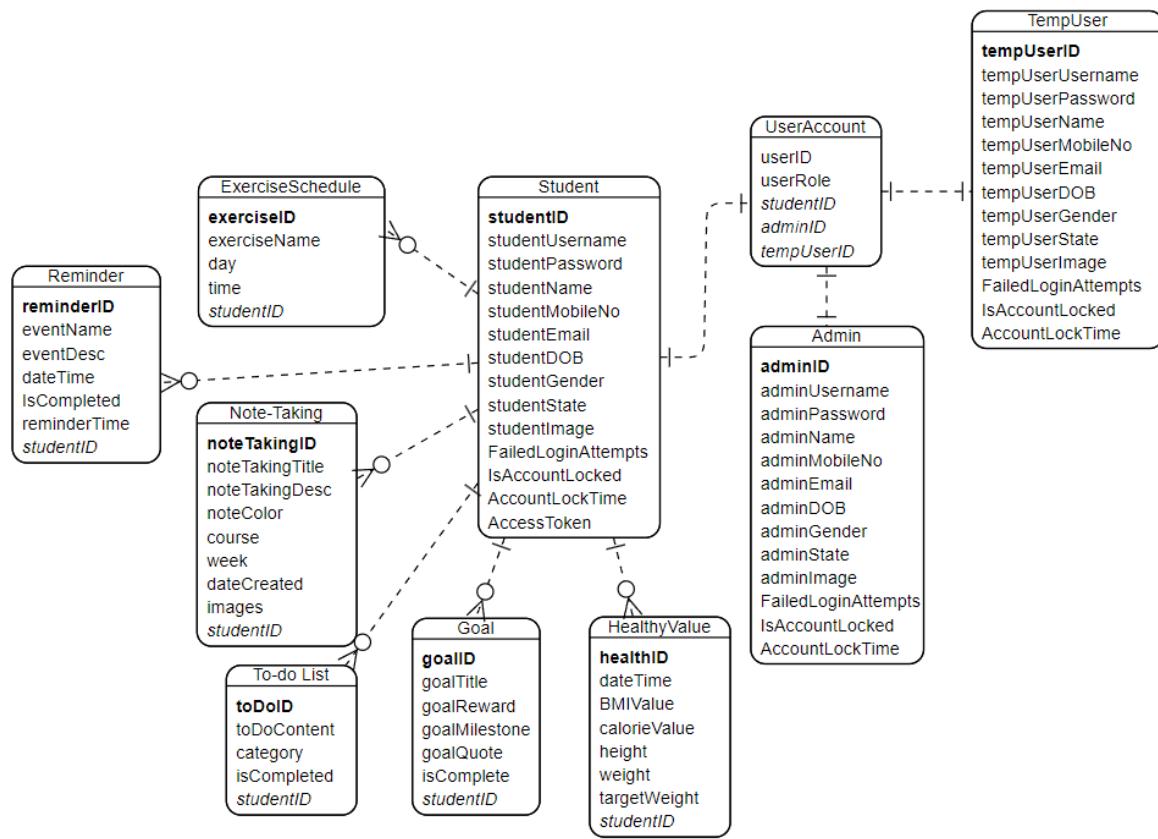


Diagram 4.35: ER diagram

4.2 User Interface Flow

Firstly, users need to register before they can use all the functionalities.

The screenshot shows the 'Register' page of the One-Stop Student System. At the top, there is a navigation bar with three dropdown menus: 'Study', 'Health', and 'Personal'. To the right of the menus are a 'LOGIN' button and a user icon. The main content area is titled 'Register' and contains several input fields:

- Full Name
- Mobile Number
- Email Address
- Username
- Password
- Retype Password
- Date of Birth (with placeholder 'mm/dd/yyyy')
- Gender (with radio buttons for Male and Female)
- Location (with a dropdown menu labeled 'Select State')

At the bottom of the form are two buttons: 'Register' (dark grey) and 'Cancel' (red).

Below the form is a footer section with three columns:

- System**: About US, Privacy & Policy, Terms & Condition
- Quick Link**: GPA Grade Calculator, To-do List, Register
- Any Suggestion ?**: A text input field labeled 'Suggestion Text' and a send icon.

Diagram 4.36: Register Page

Study ▾ Health ▾ Personal ▾

LOGIN

Register

Full Name

Mobile Number

Email Address

Username

Password

Retype Password

Date of Birth
mm/dd/yyyy

Gender
 Male Female

Location

Selangor

Kuala Lumpur

Putrajaya

System

About US

Privacy & Policy

Terms & Condition

Quick Link

GPA Grade Calculator

To-do List

Register

Any Suggestion ?

Suggestion Text

Send icon

Diagram 4.37: When toggle drop down at Register Page

After registering, the user will be redirected to the login page.

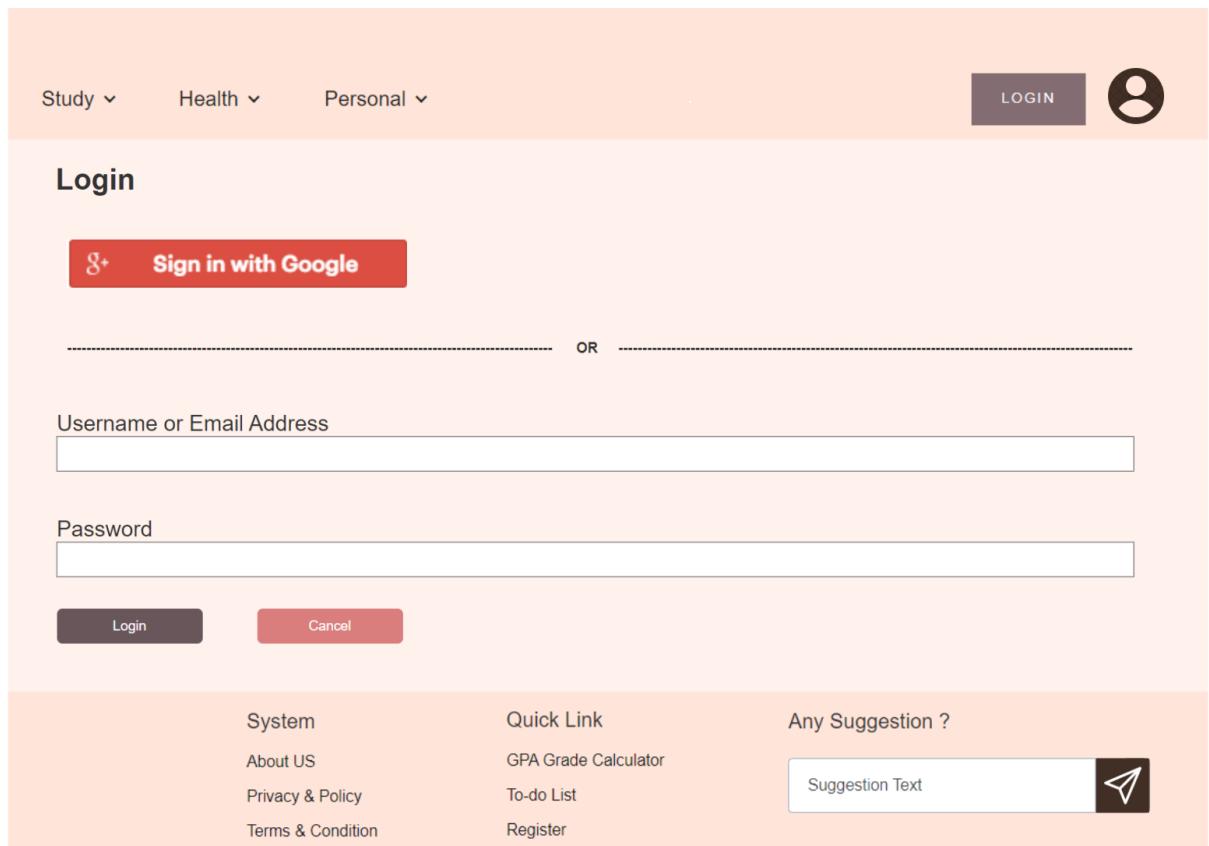


Diagram 4.38: Login Page

After login, the user will be redirected to the homepage.

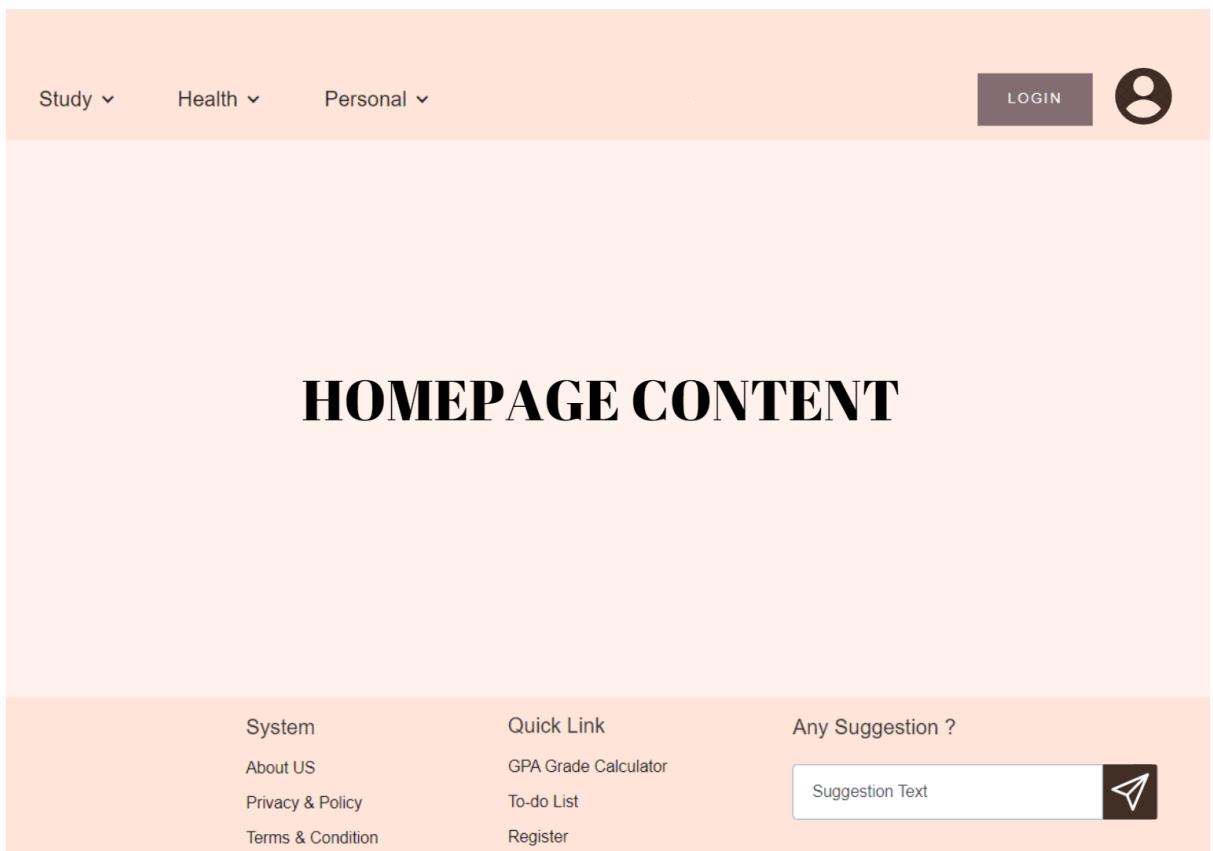


Diagram 4.39: Homepage

When the user hovers or clicks the drop down or navigation bar, a drop down list will appear.

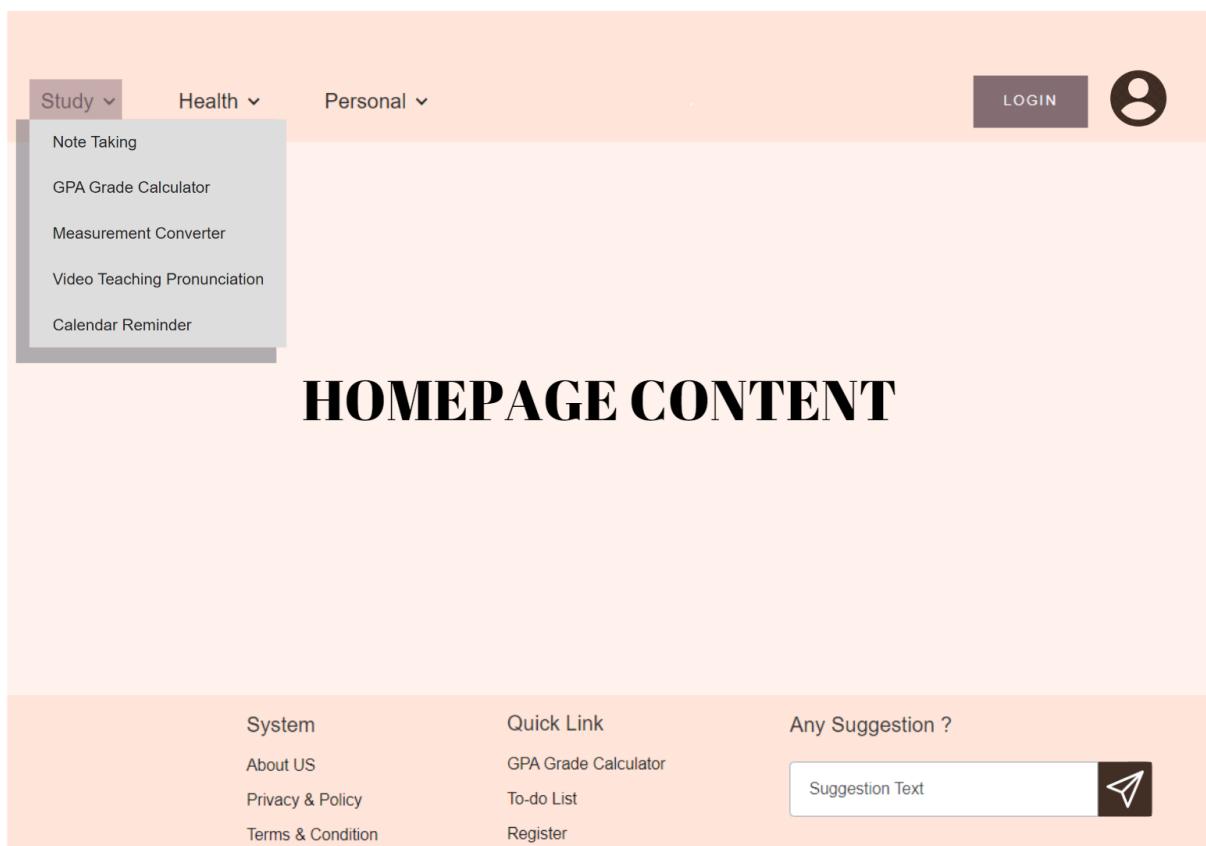


Diagram 4.40: Hover Study Category Navigation Link

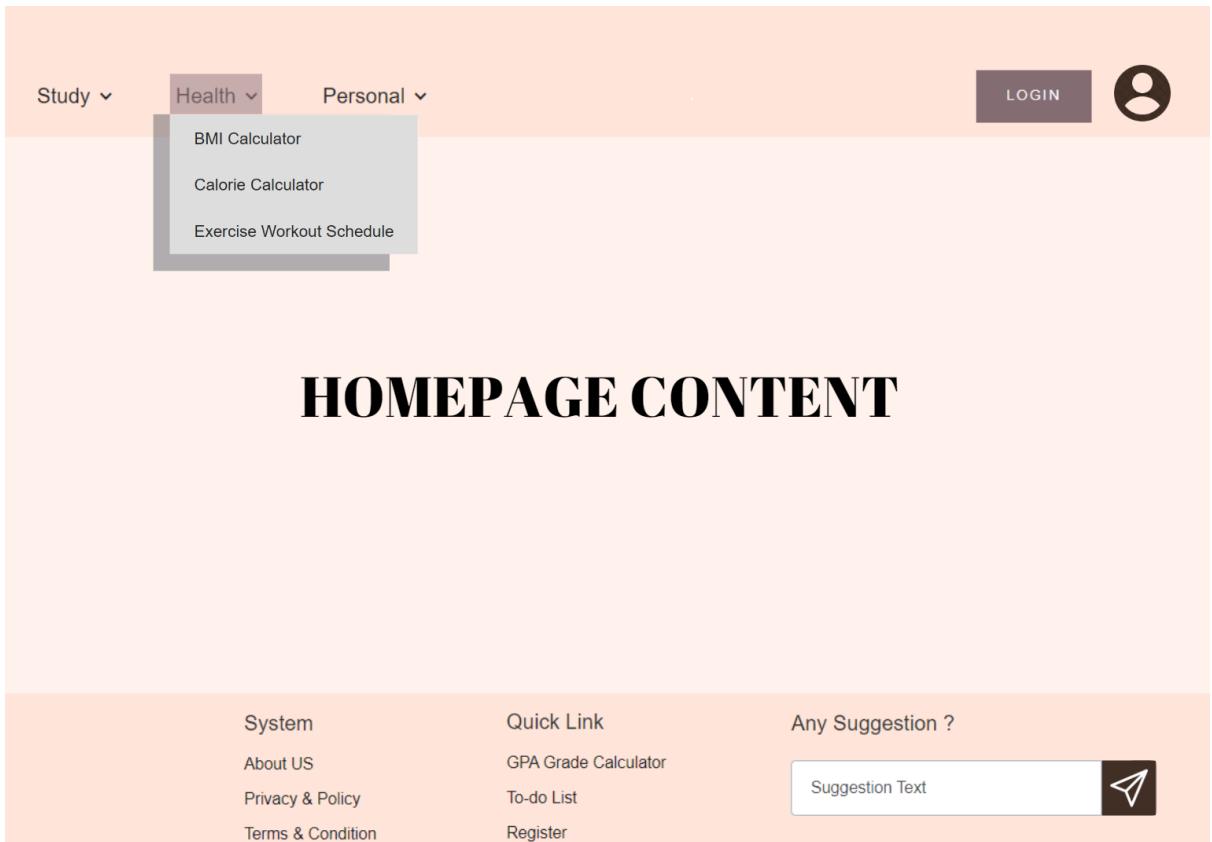


Diagram 4.41: Hover Health Category Navigation Link

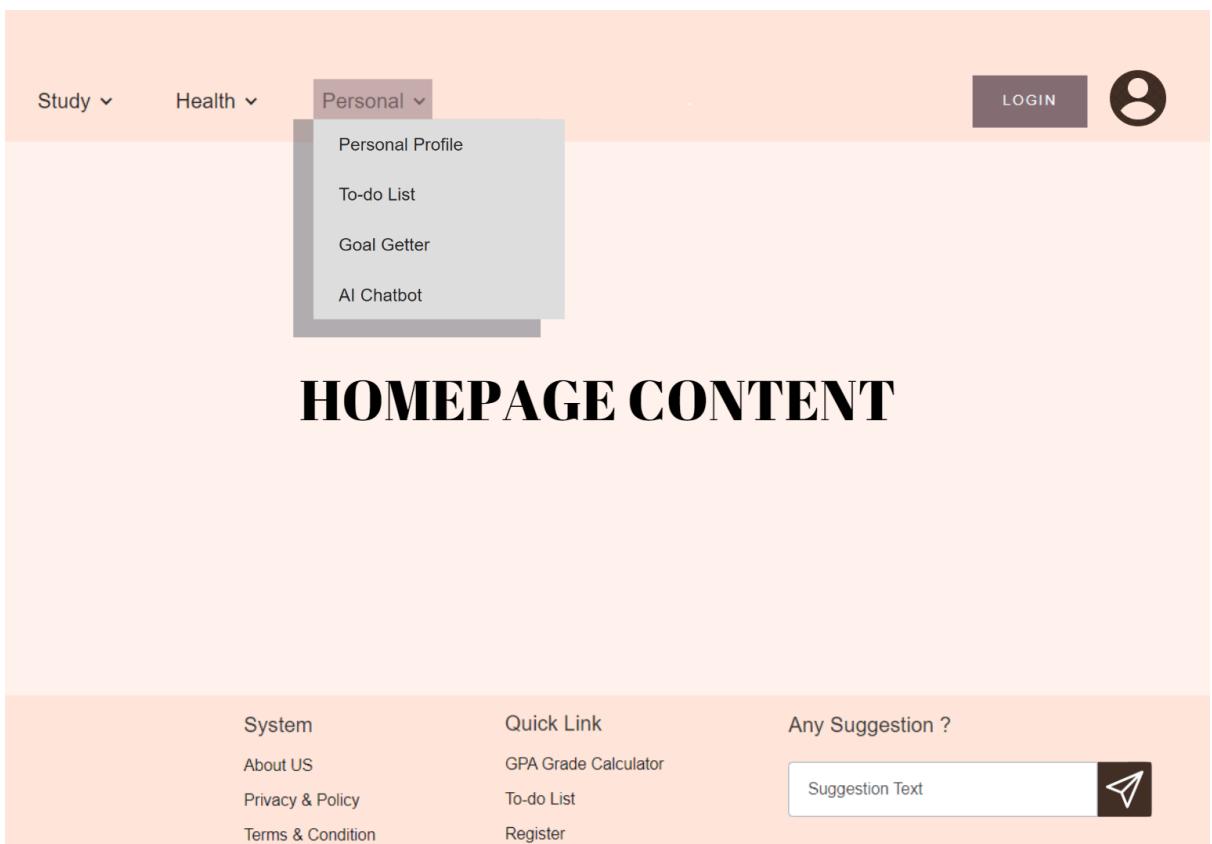


Diagram 4.42: Hover Personal Category Navigation Link

When user click on the drop down navigation, it will bring user to specific page, ex: click Note Taking will bring to Note Taking Page

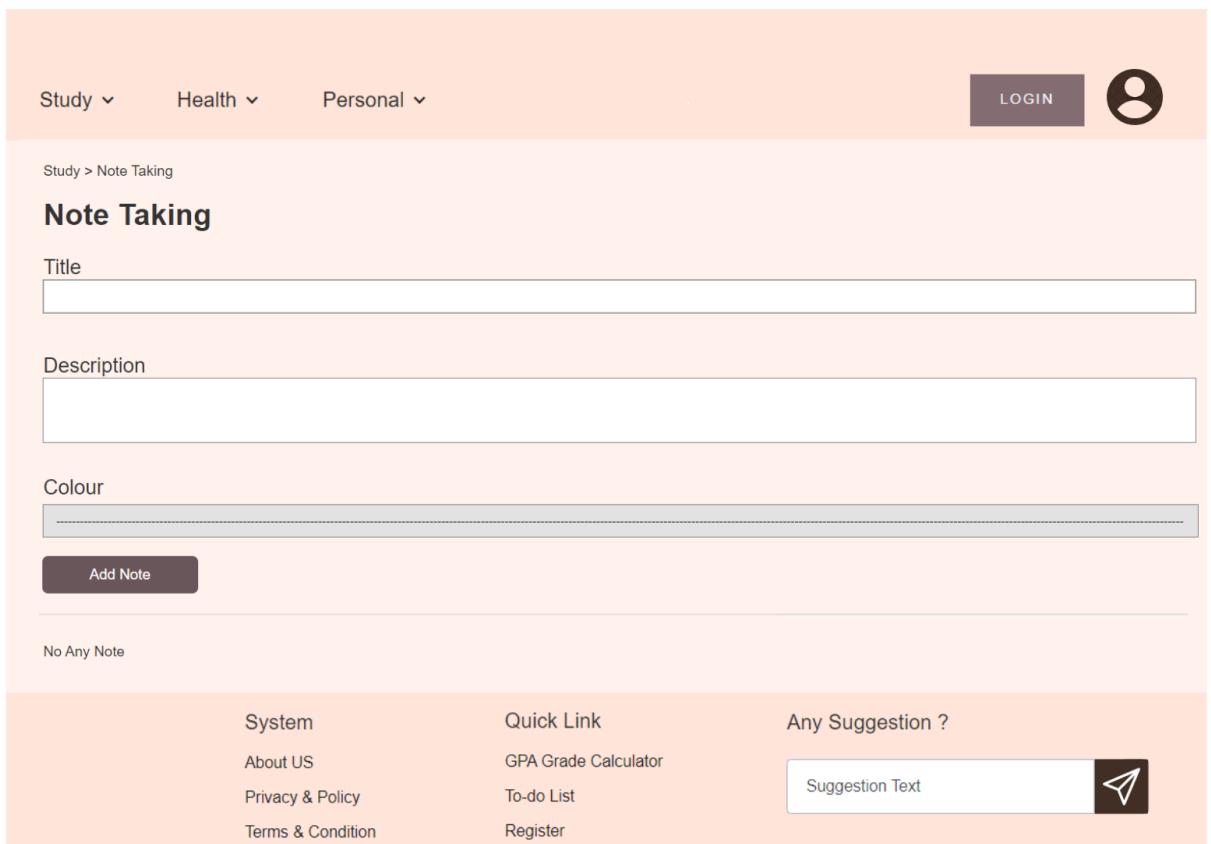


Diagram 4.43: Note Taking Page

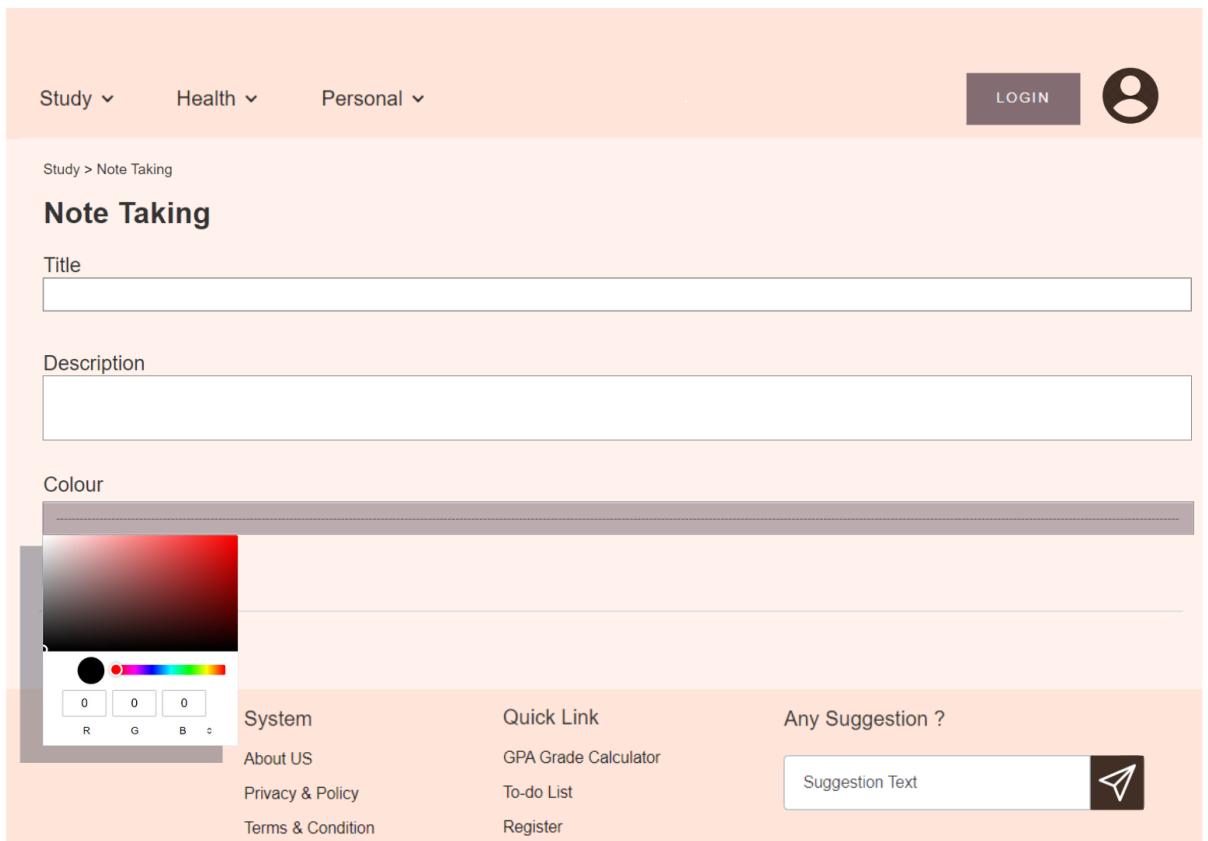


Diagram 4.44: Colour picker at note taking page

The screenshot shows the 'Calorie Calculator' page. At the top, there are navigation links for 'Study', 'Health', and 'Personal'. On the right side, there are 'LOGIN' and user profile icons. Below the header, the page title 'Calorie Calculator' is displayed. The form consists of several input fields with dropdown arrows: 'Age (years)', 'Gender' (with options 'Male' and 'Female'), 'Height (cm)', 'Weight (kg)', 'Target Weight (kg)', and 'Number of weeks to Reach Target Weight'. Below these fields is a section titled 'Activity Level' with a dropdown menu containing three levels: 'Exercise: 15-30 minutes of elevated heart rate activity.', 'Intense exercise: 45-120 minutes of elevated heart rate activity.', and 'Very Intense exercise: 2+ hours of elevated heart rate activity.'. At the bottom of the form are two buttons: 'Calculate' (dark grey) and 'Cancel' (red). At the very bottom of the page, there is a footer bar with links: 'System' (About US, Privacy & Policy, Terms & Condition), 'Quick Link' (GPA Grade Calculator, To-do List, Register), and 'Any Suggestion ?' (with a suggestion text input field and a send icon).

Diagram 4.45: Calorie Calculator Page

The screenshot shows the 'Calorie Calculator' page of the One-Stop Student System. At the top, there are navigation links for 'Study', 'Health', and 'Personal'. On the right side, there are 'LOGIN' and user profile icons. Below the header, the page title 'Calorie Calculator' is displayed. The form consists of several input fields with dropdown arrows for selecting values: 'Age (years)', 'Gender' (with options 'Male' and 'Female'), 'Height (cm)', 'Weight (kg)', 'Target Weight (kg)', and 'Number of weeks to Reach Target Weight'. Below the 'Activity Level' section, there is a detailed description of exercise levels: 'Exercise: 15-30 minutes of elevated heart rate activity.', 'Intense exercise: 45-120 minutes of elevated heart rate activity.', and 'Very intense exercise: 2+ hours of elevated heart rate activity.' A dropdown menu titled 'Select Activity Level' lists activity levels from 'Sedentary: little or no exercise' to 'Extra Active: very intense exercise daily, or physical job'. At the bottom, there is a footer with 'System' links ('About US', 'Privacy & Policy', 'Terms & Condition'), 'Quick Link' links ('GPA Grade Calculator', 'To-do List', 'Register'), and a 'Any Suggestion?' section with a text input field and a send icon.

System	Quick Link	Any Suggestion ?
About US	GPA Grade Calculator	Suggestion Text
Privacy & Policy	To-do List	
Terms & Condition	Register	

Diagram 4.46: Toggle drop down at Calorie Calculator Page

The screenshot shows the 'Personal > Profile' section of the system. At the top, there are navigation links for 'Study', 'Health', and 'Personal'. On the right, there are 'LOGIN' and a user icon buttons. Below the header, the page title 'Profile' is displayed. On the left, there is a placeholder for a profile picture with a camera icon. The main form contains fields for 'Name' (text input), 'Date Of Birth' (text input), 'Gender' (radio buttons for Male and Female), 'Location' (text input), 'Email Address' (text input), and 'Phone Number' (text input). Below the form are 'Save' and 'Cancel' buttons. At the bottom, there is a footer with three columns: 'System' (links to 'About US', 'Privacy & Policy', and 'Terms & Condition'), 'Quick Link' (links to 'GPA Grade Calculator', 'To-do List', and 'Register'), and 'Any Suggestion ?' (a text input field for suggestions and a send icon).

Diagram 4.47: Personal Profile Page

4.3 Chapter Summary and Evaluation

In this chapter, all the relevant diagrams will be covered such as use case diagrams, use case description tables, activity diagrams, and ERD diagrams to provide a better understanding of this proposed system. One overall use case diagram and three detailed use case diagrams are provided by the modules. Actors for this system are students and admins. A total of thirty use case description tables and activity diagrams are shown for each use case. Explanation for each activity diagram is provided as well. A user interface flow is provided to have a clear visualisation of the system user interface and interaction.

Chapter 5

Implementation and Testing

5 Implementation and Testing

This chapter will provide a comprehensive description of the One-Stop Student System's implementation. It includes the detailed explanations of how various components of the system are developed and integrated, accompanied by relevant code snippets and descriptions. These code snippets illustrate how the requirements of system has met and explain the functionalities and workings of each part of the system. Furthermore, test cases of each function will be included in this chapter.

5.1 Implementation

The One-Stop Student system is developed using Microsoft Visual Studio, a free and user-friendly integrated development environment (IDE). ASP.NET is used which is a free web framework for building great websites and web applications using HTML, CSS, and JavaScript.

HTML is used for the structural layout of web pages, CSS for styling and JS for adding interactivity and responsiveness to ensure a dynamic user experience.

It also uses the AjaxControlToolkit which is a set of controls and extenders for ASP.NET applications that enhance the capabilities of standard web controls. This toolkit provides AJAX (Asynchronous JavaScript and XML) functionality which allowed data to be fetched and updated dynamically without the need of a full page reload.

It also uses OWIN (Open Web Interface for .NET) middleware for managing authentication and authorization. OWIN provides a standard interface between .NET web applications and web servers which allow for modular and flexible authentication mechanisms. The system integrates with Google for user authentication using OWIN's external authentication capabilities. This integration is achieved using the GoogleOAuth2AuthenticationOptions and OWIN's authentication middleware which facilitate the implementation of OAuth 2.0 for secure authentication via Google accounts. Hence, students can log in using their Google credentials.

The application includes a chatbot feature developed using Python. The chatbot is trained using the Natural Language Toolkit (NLTK) and scikit-learn libraries for intent classification. To integrate Python scripts with the ASP.NET application, batch files (.bat) are used to execute Python scripts from the server-side code. The batch file handles the execution of Python scripts, captures user input and returns responses to the web application.

This system uses the Google Calendar API to integrate with Google Calendar which allows student to add, edit, delete events directly from this system. Youtube API is used to allow students to search for YouTube videos and display the results directly on the page. It also uses

Gmail's SMTP (Simple Mail Transfer Protocol) server for sending email reminders which implemented using .NET's SmtpClient class.

5.1.1 Code Snippets

Login

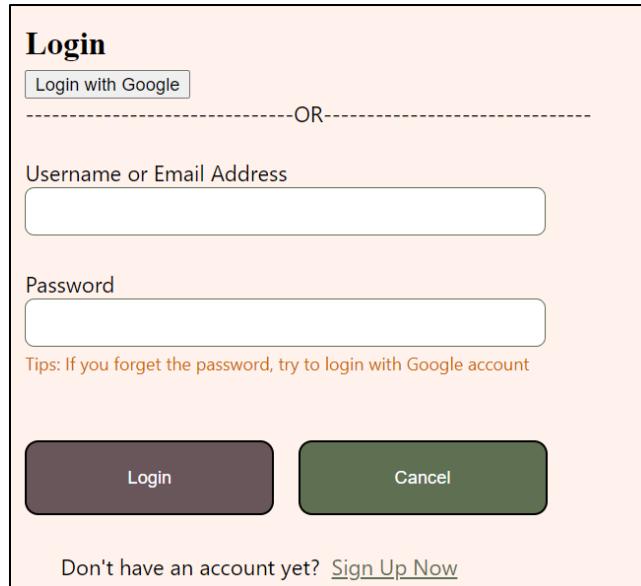


Figure 5.1 Login Page

```
protected void BtnGoogleLogin_Click(object sender, EventArgs e)
{
    var properties = new AuthenticationProperties { RedirectUri = "https://localhost:44355/GLCallback.aspx" };
    Context.GetOwinContext().Authentication.Challenge(properties, "Google");
}
```

The AuthenticationProperties object is used to set properties related to the authentication process, RedirectUri, which specifies the URL where the student will be redirected after successfully logging in with Google. The GLCallback.aspx is to handle the response from Google. Context.GetOwinContext().Authentication.Challenge(properties, "Google") initiates the OAuth authentication challenge with Google. The Challenge method is called with the specified properties containing the RedirectUri and the authentication provider, Google. It redirects the student to Google's login page. After the student logs in and grants permissions, Google will redirect them back to the specified RedirectUri.

```

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        var authenticationManager = Context.GetOwinContext().Authentication;
        var loginInfo = authenticationManager.GetExternalLoginInfo();

        if (loginInfo != null)
        {
            var claimsIdentity = new ClaimsIdentity(loginInfo.ExternalIdentity.Claims, DefaultAuthenticationTypes.ApplicationCookie);
            authenticationManager.SignIn(new AuthenticationProperties { IsPersistent = true }, claimsIdentity);

            var identity = HttpContext.Current.User.Identity as ClaimsIdentity;

            var emailClaim = claimsIdentity.FindFirst(ClaimTypes.Email);
            if (emailClaim != null)
            {
                Session["Username"] = emailClaim.Value;
                Session["UserType"] = "Student";
                var email = Session["Username"].ToString();
                var studentId = GenerateNextStudentID();
                var username = email;

                if (!UserExists(email))
                {
                    InsertStudent(studentId, username, email);
                    string userID = GetNextUserID();
                    Session["UserID"] = studentId;
                    SaveIntoUser(userID, studentId, "Student");
                }
                else
                {
                    string studentId1 = GetStudentIDFromDatabase(email);
                    Session["UserID"] = studentId1;
                }
            }

            // RequestCalendarPermissions();

            Response.Redirect("~/Homepage.aspx", false);
        }
        else
        {
            Response.Redirect("~/Login.aspx", false);
        }
    }
    else
    {
        Response.Redirect("~/Login.aspx");
    }
}
catch (Exception ex)
{
    Debug.WriteLine($"Exception in GoogleLoginCallback Page_Load: {ex.Message}");
    Response.Redirect("~/Error.aspx", false);
}
}

```

The code `var authenticationManager = Context.GetOwinContext().Authentication` retrieves the OWIN authentication manager to interact with the authentication process. `var loginInfo = authenticationManager.GetExternalLoginInfo()` attempts to get information about the user from the external login provider, Google. `loginInfo != null` checks whether login information was successfully retrieved from Google. If `loginInfo` is null, the student has not authenticated successfully and they are redirected back to the login page. It also creates a new `ClaimsIdentity` based on the claims which is user information provided by Google. This identity is configured to use the default application cookie authentication type. While `authenticationManager.SignIn()` signs in the student using the created identity with the option to persist the authentication across sessions (`IsPersistent`). Then it searches the claims for the student's email address to identify whether to store inside database. If successful, it will redirect authenticated students to the homepage. Overall, it is the integration of Google authentication into the ASP.NET application, managing user sessions and data securely using OWIN middleware.

Note Taking

The screenshot shows a web-based application for managing notes. On the left, there is a table with columns for Course, Color, and a large 'Apply Course & Color' button. Each row corresponds to a course listed in the 'Course' column. To the right of the table are several input fields: 'Title', 'Description', 'Image' (with a 'Choose Files' button and a note 'No file chosen'), 'Colour', 'Course', 'Week', and two buttons: 'Add Note' and 'Cancel'. Below the main form is a search section with 'Search by Week' and 'Search by Course' fields, and a 'Search' button. At the bottom is a preview area showing a note entry for 'Week1' with content 'SDA TEST asdd' and buttons for 'Edit' and 'Delete'.

Figure 5.2 Note Taking Page

```

List<string> imagePaths = new List<string>();

if (fileUpload.HasFiles)
{
    foreach (HttpPostedFile uploadedFile in fileUpload.PostedFiles)
    {
        string filename = Path.GetFileName(uploadedFile.FileName);

        string serverPath = Server.MapPath("~/Image/") + filename;

        uploadedFile.SaveAs(serverPath);
        imagePaths.Add(filename);
    }
}

```

This is the code is to check if the user has upload image, then to handle multiple file uploads by saving each file to a server directory and store all the files in imagePaths list.

```
using (SqlConnection connection = new SqlConnection(connectionString))
{
    string query = "INSERT INTO NoteTaking (NoteTakingID, NoteTakingTitle, NoteTakingDesc, studentID, noteColor, Course, Week, DateCreated, Images) VALUES (@ID, @title, @desc, @studentID, @color, @course, @week, @date, @images)";

    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@ID", newNoteID);
        command.Parameters.AddWithValue("@title", NoteTitle);
        command.Parameters.AddWithValue("@desc", NoteDesc);
        command.Parameters.AddWithValue("@studentID", studentID);
        command.Parameters.AddWithValue("@color", colour);
        command.Parameters.AddWithValue("@course", course);
        command.Parameters.AddWithValue("@week", week);
        command.Parameters.AddWithValue("@date", DateTime.Now);
        command.Parameters.AddWithValue("@images", string.Join(",", imagePaths));

        connection.Open();
        command.ExecuteNonQuery();
    }
    GridView1.DataBind();
}
```

This is the code snippet to add note details into database using sql insert statement, the images stored is taking the images in imagePaths list separating with coma.

```

private void AddNoteToPage(string noteID, string title, string desc, string color, string course, string week, string date, string image)
{
    Panel notePanel = new Panel();
    notePanel.CssClass = "note";
    notePanel.Style.Add("background-color", color);

    Label titleLabel = new Label();
    titleLabel.Text = title;
    titleLabel.Font.Bold = true;

    Label descLabel = new Label();
    descLabel.Text = desc;

    Label dateLabel = new Label();
    dateLabel.Text = " (" + date + ")";

    Label courseLabel = new Label();
    courseLabel.Text = course;
    courseLabel.Font.Bold = true;
    courseLabel.Font.Underline = true;

    Label weekLabel = new Label();
    weekLabel.Text = "Week" + week + " ";

    notePanel.Controls.Add(weekLabel);
    notePanel.Controls.Add(dateLabel);
    notePanel.Controls.Add(new LiteralControl("<br />"));
    notePanel.Controls.Add(courseLabel);
    notePanel.Controls.Add(new LiteralControl("<br />"));

    if (!string.IsNullOrWhiteSpace(image))
    {
        string[] images = image.Split(new char[] { ',' }, StringSplitOptions.RemoveEmptyEntries);
        foreach (string imagePath in images)
        {
            Image img = new Image();
            img.ImageUrl = "~/Image/" + imagePath;
            img.Width = 100;
            img.Height = 100;
            notePanel.Controls.Add(img);
        }
        notePanel.Controls.Add(new LiteralControl("<br />"));
    }

    notePanel.Controls.Add(titleLabel);
    notePanel.Controls.Add(new LiteralControl("<br />"));
    notePanel.Controls.Add(descLabel);
    notePanel.Controls.Add(new LiteralControl("<br />"));

    // Add Edit Button
    Button editButton = new Button();
    editButton.ID = $"editButton_{noteID}";
    editButton.Text = "Edit";
    editButton.CssClass = "edit-button";
    editButton.CommandArgument = noteID;
    editButton.Click += EditButton_Click;
    notePanel.Controls.Add(editButton);

    notePanel.Controls.Add(new LiteralControl(" "));
    // Add Delete Button
    Button deleteButton = new Button();
    deleteButton.ID = $"deleteButton_{noteID}";
    deleteButton.Text = "Delete";
    deleteButton.CssClass = "delete-button";
    deleteButton.CommandArgument = noteID;
    deleteButton.Click += DeleteButton_Click;
    notePanel.Controls.Add(deleteButton);

    noteContainer.Controls.Add(notePanel);
}

```

This code snippet is to sets up the panel with various controls such as labels for the note's title, description, date, course and week with the specific background colour. If images exist then they are added as image controls. The "Edit" and "Delete" buttons is added to allow edit and delete of the note. The panel is added to a container control, so the notes are visible to student.

GPA Grade Calculator

GPA Grade Calculator

Grade Table for before July 2023 intake students

Grade	Mark Range	Grade Point
A	80-100	4.0000
A-	75-79	3.7500
B+	70-74	3.5000
B	65-69	3.0000
B-	60-64	2.7500
C+	55-59	2.5000
C	50-54	2.0000
F	0-49	0.0000

Current CGPA:

Target CGPA:

Total Credits Hours Earned:

Tips: Next Semester Subjects (Arrange it according to ability - confident to less confident)

Subject 1 Name: SEM	Credit for Subject 1: 3
Subject 2 Name: CC	Credit for Subject 2: 4
Subject 3 Name: SRE	Credit for Subject 3: 3
Subject 4 Name: SE	Credit for Subject 4: 5
Subject 5 Name: FM	Credit for Subject 5: 2

Grades needed for each subject:

Subject 4 (SE, 5 credits): A

Subject 2 (CC, 4 credits): A

Subject 1 (SEM, 3 credits): A

Subject 3 (SRE, 3 credits): A

Subject 5 (FM, 2 credits): A

Unfortunately, even though you achieve A's in all subjects, you still cannot achieve the target CGPA in the next semester with the provided subjects.

The maximum achievable CGPA is 3.9906.

Keep trying hard, and you can improve in the next semester!

Figure 5.3 GPA Grade Calculator Page

```

function calculatePredictions() {
    // Retrieve user inputs
    const currentCGPA = parseFloat(document.getElementById('currentCGPA').value);
    const aimedCGPA = parseFloat(document.getElementById('aimedCGPA').value);
    const totalCredit = parseFloat(document.getElementById('totalCredit').value);

    const subjectCredits = [];

    // Initialize error message
    let errorMessage = "";

    // Validate totalCredit
    if (isNaN(totalCredit) || totalCredit <= 0) {
        errorMessage += "Please enter a valid positive integer value for total credit field\n";
    }

    // Validate currentCGPA
    if (isNaN(currentCGPA)) {
        errorMessage += "Please enter a valid numerical value for current CGPA\n";
    } else if (currentCGPA < 0.0 || currentCGPA > 4.0) {
        errorMessage += "Current CGPA should be between 0.0 and 4.0\n";
    }

    // Validate aimedCGPA
    if (isNaN(aimedCGPA)) {
        errorMessage += "Please enter a valid numerical value for target CGPA\n";
    } else if (aimedCGPA < 0.0 || aimedCGPA > 4.0) {
        errorMessage += "Target CGPA should be between 0.0 and 4.0\n";
    }

    // Loop through all dynamically added subject inputs
    for (let i = 1; i <= subjectCount; i++) {
        const subjectName = document.getElementById('subjectName${i}').value.trim();
        const credit = parseFloat(document.getElementById('credit${i}').value);
        if (subjectName === "") {
            errorMessage += "Please enter a valid subject name for subject ${i}\n";
        }
        if (isNaN(credit) || credit <= 0) {
            errorMessage += "Please enter a valid positive numerical value for credit of subject ${i}.\n";
        } else {
            subjectCredits.push({
                subject: i,
                subjectName: subjectName, // Store subject name here
                subjectCredit: credit // Store subject credit here
            });
        }
    }

    if (errorMessage) {
        alert(errorMessage);
        return;
    }

    // Define grade points for different grades
    const gradePoints = [
        'A': 4.0,
        'A-': 3.75,
        'B+': 3.5,
        'B': 3.0,
        'B-': 2.75,
        'C+': 2.5,
        'C': 2.0,
        'F': 0.0
    ];

    // Calculate total quality points for the current CGPA
    const totalQualityPoints = currentCGPA * totalCredit;

    // Calculate total credits for the next semester dynamically
    const totalNextSemesterCredits = subjectCredits.reduce((total, subject) => total + subject.subjectCredit, 0);

    // Calculate total quality points needed for the aimed CGPA
    const totalAimedQualityPoints = aimedCGPA * (totalCredit + totalNextSemesterCredits);

    // Calculate the difference in quality points needed
    let requiredQualityPoints = totalAimedQualityPoints - totalQualityPoints;

    // Sort subjects by credit hours in descending order
    subjectCredits.sort((a, b) => b.subjectCredit - a.subjectCredit);

    // Initialize an array to store grade and remaining quality points for each subject
    const subjectResults = [];

    // Loop through each subject to calculate grade and remaining quality points
    for (const subject of subjectCredits) {
        const subjectCredit = subject.subjectCredit;
        let gradePoint = requiredQualityPoints / subjectCredit;

        // Ensure the calculated grade point is within the valid range
        gradePoint = Math.max(0, Math.min(4.0, gradePoint));

        // Ensure the calculated grade is not lower than 'C'
        const grade = markGrade(Math.max(gradePoint, gradePoints['C']));

        subjectResults.push({
            subject: subject.subject,
            subjectName: subject.subjectName, // Include subject name in results
            subjectCredit: subject.subjectCredit, // Include subject credit in results
            grade: grade,
        });

        // Update remaining quality points for the next iteration
        requiredQualityPoints -= gradePoint * subjectCredit;
    }

    // Stop if we have met or exceeded the required quality points
    if (requiredQualityPoints <= 0) break;
}

// Calculate maximum achievable CGPA
let maxAchievableQualityPoints = totalQualityPoints;
for (const subject of subjectCredits) {
    maxAchievableQualityPoints += subject.subjectCredit * gradePoints['A'];
}
const maxAchievableCGPA = maxAchievableQualityPoints / (totalCredit + totalNextSemesterCredits);

// Output results
let resultsHTML = '<p>Grades needed for each subject:</p>';
subjectResults.forEach(result => {
    resultsHTML += '<p>Subject: ${result.subject} (${result.subjectName}), ${result.subjectCredit} credits: ${result.grade}</p>';
});

if (requiredQualityPoints > 0) {
    resultsHTML += '<p style="color: red; font-weight: bold;">Unfortunately, even though you achieve A's in all subjects, you still cannot achieve the target CGPA in the next semester with the provided subjects.</p>';
    resultsHTML += '<p style="color: red; font-weight: bold;">The maximum achievable CGPA is ${maxAchievableCGPA.toFixed(4)}</p>';
    resultsHTML += '<p style="color: red; font-weight: bold;">Keep trying hard, and you can improve in the next semester!</p>';
    resultsHTML += '';
} else {
    resultsHTML += '<p style="color: green; font-weight: bold;">Congratulations! You can achieve your target CGPA with the provided subject grade!</p>';
    resultsHTML += '';
}

document.getElementById('resultsContainer').innerHTML = resultsHTML;
}

```

This JavaScript function calculates the subjects grades needed for next semester to achieve a target CGPA based on user inputs. It first validates the inputs for current CGPA, aimed CGPA, and subject credits. If inputs are valid, it calculates the total quality points required to reach the

target CGPA and then determines the grades needed for each subject by comparing the required quality points with available credits. The function sorts subjects by credit hours then assigns grades and checks if achieving the target CGPA is possible. It will show different messages based on achievable and unachievable target CGPA.

Measurement Converter

Unit Converter

Select Category: Weight and Mass

Weight and Mass Conversion

Enter Value: 3 Sun's Mass

Notation: Normal Decimal Places: 10

Convert

Result: Gram: 5.965410000e+33 g
Kilogram: 5.965410000e+30 kg
Milligram: 5.965410000e+36 mg
Decigram: 5.965410000e+31 dg
Centigram: 5.965410000e+32 cg
Metric Ton: 5.965410000e+27 t
Long Ton: 5.872680000e+27
Short Ton: 6.529800000e+27
Pound: 1.315500000e+31 lbs
Ounce: 2.104800000e+32 oz
Carat: 3.021000000e+33 ct
Atomic Mass Unit: 1.787700000e+56 u
Dalton: 1.788300000e+56 Da
Gamma: 5.965410000e+52 γ
Planck Mass: 0.000000000
Electron Mass: 2.470470000e+42
Muon Mass: 1.155750000e+40
Proton Mass: 1.298040000e+39
Neutron Mass: 1.295700000e+39
Deuteron Mass: 6.476700000e+39
Earth Mass: 0.000000000
Sun Mass: 3.000000000

Figure 5.4 Unit Converter Page

This is one of the converter cases, convert energy, it will specify all the formulas for each case, then display the result based on certain format.

Video Teaching Pronunciation

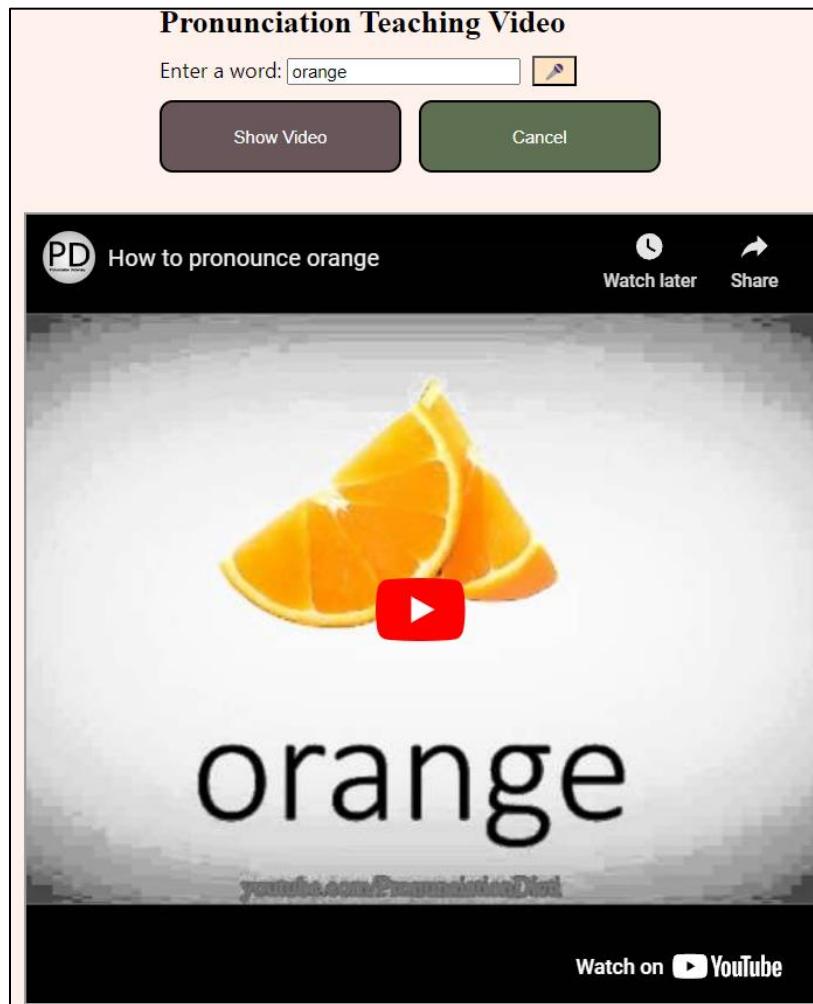


Figure 5.5 Video Teaching Pronunciation Page

```
protected async void SearchAndShowVideo(object sender, EventArgs e)
{
    string apiKey = ConfigurationManager.AppSettings["YouTubeApiKey"];
    string word = wordInput.Text;
    if (!string.IsNullOrEmpty(word))
    {
        string searchQuery = $"{word} pronunciation";

        string videoId = await FetchYouTubeVideoId(apiKey, searchQuery);

        if (!string.IsNullOrEmpty(videoId))
        {
            ClientScript.RegisterStartupScript(this.GetType(), "showVideo", $"showVideo('{videoId}');", true);
        }
        else
        {
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('No pronunciation videos found.');//", true);
        }
    }
    else
    {
        ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('Please Input WORD');//", true);
    }
}
```

This code is triggered when the user clicks the search button to search for a video. First, it retrieves the YouTube API key from configuration settings and the word that needs to be searched from user input. Then, it constructs a search query for "pronunciation" videos based on the input word such as

apple pronunciation. Then it calls the FetchYouTubeVideoId method to get the video ID asynchronously without blocking the execution of other code so that to let students have a smoother experience. Then, depending on whether a video ID is returned, it either shows the video on the page or displays an alert that no videos found or please input word. To display video, this server-side code uses ClientScript.RegisterStartupScript to inject the showVideo function call into the client-side JavaScript.

```
private async Task<string> FetchYouTubeVideoId(string apiKey, string searchQuery)
{
    string apiUrl = $"https://www.googleapis.com/youtube/v3/search?part=snippet&q={searchQuery}&type=video&key={apiKey}";

    using (HttpClient client = new HttpClient())
    {
        HttpResponseMessage response = await client.GetAsync(apiUrl);

        if (response.IsSuccessStatusCode)
        {
            string json = await response.Content.ReadAsStringAsync();
            JObject data = JObject.Parse(json);

            if (data["items"] != null && data["items"].HasValues)
            {
                string videoId = data["items"][0]["id"]["videoId"].ToString();
                return videoId;
            }
        }
    }

    return null;
}
```

This code asynchronously fetches the YouTube video ID for a given search query like “apple pronunciation” using the YouTube API. It constructs a URL for the YouTube API search request. Then it uses HttpClient to send a GET request to the API. If the response is successful, parses the JSON response to extract the video ID from the first search result. It will returns the video ID if found result or return null if no results are found.

Calendar Reminder

Reminder

Event Name:	<input type="text"/>
Event Description:	<input type="text"/>
Event Date and Time (ex: 07-16-2024 00:29:58):	<input type="text"/>
Email Reminder Sent Again x min before event start (ex: 30):	<input type="text"/>
Add	Cancel

	dateTime	eventName	eventDesc	reminderMinBefore
Edit Delete	8/30/2024 10:43:32 AM	qwe	wqe	1
Edit Delete	8/30/2024 10:41:31 AM	fghj	fgh	1
Edit Delete	8/30/2024 1:14:35 AM	987	876	2
Edit Delete	8/29/2024 7:44:04 PM	qqqk	qqqk	1
Edit Delete	8/26/2024 10:05:45 PM	22345	2233	1
Edit Delete	8/26/2024 10:04:42 PM	asd12	fd	1
Edit Delete	8/26/2024 9:42:16 PM	666	666	1
Edit Delete	8/26/2024 9:08:50 PM	888	888	1
Edit Delete	8/26/2024 6:12:57 PM	ko	ko	3
Edit Delete	8/5/2024 10:58:01 AM	po	so	1

12

Figure 5.6 Calendar Reminder Page

```

<asp:TextBox ID="txtEventDate" runat="server" Height="27px" Width="690px"></asp:TextBox>
<ajaxToolkit:CalendarExtender
    ID="CalendarExtender1"
    runat="server"
    TargetControlID="txtEventDate"
    Format="MM-dd-yyyy HH:mm:ss"
    OnClientShown="disablePastDates" />
<br />
```

This code snippet is using the CalendarExtender control from the Ajax Control Toolkit. This control extends the functionality of a standard TextBox to include a calendar popup for date selection in the txtEventDate textbox field. When student clicks on the event date field, the calendar popup will appear. When the calendar popup, it will trigger the disablePastDates client-side JavaScript function.

```

function disablePastDates(sender, args) {
    var today = new Date();
    today.setHours(0, 0, 0, 0);

    var days = sender._days.body.rows; //Accesses the rows of the calendar table that contains the days of the month
    for (var i = 0; i < days.length; i++) {
        for (var j = 0; j < days[i].cells.length; j++) {
            var cell = days[i].cells[j];
            var cellDate = new Date(cell.date);

            if (cellDate < today) {
                cell.className += " disabled-date";
                cell.style.pointerEvents = "none";
            } else {
                cell.className = cell.className.replace("disabled-date", "");
                cell.style.pointerEvents = "auto";
            }
        }
    }
}

```

This function is to disable past dates in the CalendarExtender control so that students cannot select dates earlier than today. It will iterate over each cell in the calendar, if the cell's date is earlier than today which the date is in the past, it will adds a CSS class to the cell to apply the disabled-date styling and disables mouse pointer for the cell to prevent user interaction. If the date is today or in the future, it will remove the disabled-date class and enables mouse pointer for the cell.

```

private string GetAccessToken(string userId)
{
    string accessToken = null;
    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string query = "SELECT AccessToken FROM Student WHERE studentID = @StudentID";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@StudentID", userId);
            connection.Open();
            var result = command.ExecuteScalar();
            if (result != null)
            {
                accessToken = result.ToString();
            }
        }
    }

    return accessToken;
}

```

This method retrieves the OAuth 2.0 access token for Google Calendar API for a specific student from the database because this is necessary for add, edit and delete the event in the student's Google Calendar.

```

using (SqlConnection connection = new SqlConnection(connectionString))
{
    string query = "INSERT INTO Reminder (ReminderID, eventName, eventDesc, studentID, dateTime, reminderTime) " +
                   "VALUES (@ID, @name, @desc, @studentID, @dateTime, @reminderTime); " +
                   "SELECT @ID";

    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@ID", newReminderID);
        command.Parameters.AddWithValue("@name", name);
        command.Parameters.AddWithValue("@desc", desc);
        command.Parameters.AddWithValue("@studentID", studentID);
        command.Parameters.AddWithValue("@dateTime", dateTime);
        command.Parameters.AddWithValue("@reminderTime", reminderTime);

        connection.Open();

        object result = command.ExecuteScalar();

        if (result != null)
        {
            return result.ToString();
        }
        else
        {
            Console.WriteLine("Failed to retrieve the new ReminderID.");
            return null;
        }
    }
}

```

This code snippet is to add event to database using insert query.

```

protected void SendReminderEmail(string toEmail, string eventName, string dateTime, string eventDesc)
{
    try
    {
        // Create the mail message
        MailMessage msg = new MailMessage();
        msg.To.Add(toEmail);
        msg.From = new MailAddress("abc@gmail.com");
        msg.Subject = $"Reminder: {eventName}";
        msg.Body = $"Dear Student,\n\nYou have a reminder for '{eventName}' scheduled on {dateTime}.\n\nEvent Description: {eventDesc}";

        // Configure the SMTP client
        SmtpClient client = new SmtpClient("smtp.gmail.com", 587)
        {
            EnableSsl = true,
            Credentials = new NetworkCredential("abc@gmail.com", "<--Password-->")
        };

        client.Send(msg);
        Console.WriteLine("Reminder email sent successfully.");
    }
    catch (SmtpException ex)
    {
        Console.WriteLine($"Failed to send reminder email: {ex.Message}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"An error occurred: {ex.Message}");
    }
}

```

This is to sends an email to the student email to remind them about the event created. It creates a `MailMessage` object with the recipient's email, subject and body. Then it configures an `SmtpClient` which is a class provided by .NET for sending emails using the Simple Mail Transfer Protocol (SMTP) to send the email through Gmail's SMTP server. It will ensure that the communication between this system and the SMTP server is encrypted using SSL (Secure Sockets Layer) for securing email transmission. `NetworkCredential` is a class used to provide

credentials with username and password for authenticating with the SMTP server. In here, app password is generated and used instead of using regular Gmail account password because Gmail might block sign-in attempts from less secure apps. Then it sends the email message to the configured SMTP server and logs the success or failure.

```
// Send reminder x min before event start
int reminderMinutes;
if (int.TryParse(txtTime.Text, out reminderMinutes))
{
    DateTime reminderTime = eventDate.AddMinutes(-reminderMinutes);
    // Schedule the reminder only if it is in the future
    if (reminderTime > DateTime.Now)
    {
        ScheduleReminder(reminderTime, studentEmail, eventName, eventDesc, txtEventDate.Text, reminderID, txtTime.Text);
        lblPass.Visible = false;
    }
    else
    {
        lblPass.Text = "Reminder time has already passed. No reminder will be send again to remind before event start.";
        lblPass.Visible = true;
    }
}
```

This code shows how many minutes before an event start, the reminder should be sent and calls ScheduleReminder to set up the reminder email. It first converts the user input from a text box (txtTime.Text) into an integer representing how many minutes before the event the reminder should be sent. Then it subtracts the number of reminder minutes from the event date to determine the exact time the reminder should be sent. If the reminder time is in the future then the reminder is scheduled using ScheduleReminder, else display a message to student saying that will not send again the email reminder.

```
private void ScheduleReminder(DateTime reminderTime, string toEmail, string subject, string body, string oriDateTime, string reminderID, string min)
{
    TimeSpan delay = reminderTime - DateTime.Now;
    bool eventsSame = CheckIfEventDetailSame(reminderID, subject, body, oriDateTime, min);

    // Check if the reminder time is in the future
    if (delay.TotalMilliseconds > 0)
    {
        // Using a timer to send reminder at specified time
        System.Threading.Timer reminderTimer = new System.Threading.Timer((state) =>
        {
            if (eventsSame)
            {
                SendReminderEmailAgain(toEmail, subject, oriDateTime, body);
            }
            else
            {
                return;
            }
        }, null, delay, TimeSpan.Zero);
        lblPass.Visible = false;
    }
}
```

This method is used to send again a reminder email to a user at a specified time before an event starts. It calculates the delay until the reminder time, checks if the event details have changed, and uses a timer to trigger the email sending at the appropriate time. First it computes the time difference between the current time and the reminder time. This TimeSpan object represents

how long to wait before sending the reminder. Then it checks if the event details have remained the same to avoid sending incorrect or outdated information. If the delay is positive means the reminder time is in the future. Then it creates a new timer that triggers an action after the specified delay (delay). The function inside the timer checks if the event details are the same then sends the reminder email again.

```
private void CreateGoogleCalendarEvent(string accessToken, string eventName, string eventDesc, DateTime eventDate)
{
    var credential = GoogleCredential.FromAccessToken(accessToken)
        .CreateScoped(new[] { CalendarService.Scope.Calendar });

    var service = new CalendarService(new BaseClientService.Initializer()
    {
        HttpClientInitializer = credential,
        ApplicationName = "OneStopStudentSystem"
    });

    var newEvent = new Event()
    {
        Summary = eventName,
        Description = eventDesc,
        Start = new EventDateTime()
        {
            DateTime = eventDate,
            TimeZone = "Asia/Kuala_Lumpur"
        },
        End = new EventDateTime()
        {
            DateTime = eventDate.AddHours(1),
            TimeZone = "Asia/Kuala_Lumpur"
        }
    };

    EventsResource.InsertRequest request = service.Events.Insert(newEvent, "primary");
    request.Execute();
}
```

This method creates an event on Google Calendar using the Google Calendar API. It creates a GoogleCredential object from the provided OAuth 2.0 access token to authenticate API requests and specifies the scope of the credentials to give permission to manage the student's calendar. Then it initializes Google Calendar Service with credentials to authenticate requests and name of this system making the request. It constructs an event object with the specified details such as title, description, start time and end time. It creates an insert request for adding a new event to the user's primary calendar, executes the insert request which sends the event data to Google Calendar's servers. Then the new event is added to the calendar visibly and accessibly within the Google Calendar interface.

BMI Calculator

BMI Calculator

Recent Data :

Apply Height & Weight	dateTime	Height	Weight	BMIValue
Apply	9/1/2024 12:28:32 AM	150	67	29.78
Apply	8/31/2024 12:40:33 AM	150	69	30.67
Apply	8/31/2024 12:28:05 AM	150	70	31.11
Apply	8/31/2024 12:21:22 AM	150	72	32.00
Apply	8/30/2024 10:43:36 AM	150	74	32.89

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#)

Apply Age & Gender studentGender Age

Apply	studentGender	Age
Apply	Female	21

Age:

Gender:

Height (cm):

Weight (kg):



Figure 5.7 BMI Calculator Page

```

function calculateBMI() {
    var age = parseInt(document.getElementById("age").value);
    var gender = document.getElementById("gender").value;
    var height = parseFloat(document.getElementById("height").value);
    var weight = parseFloat(document.getElementById("weight").value);

    document.getElementById('<%= hiddenWeight.ClientID %>').value = weight.toFixed(2);
    document.getElementById('<%= hiddenHeight.ClientID %>').value = height.toFixed(2);
    document.getElementById('<%= hiddenAge.ClientID %>').value = age;
    document.getElementById('<%= hiddenGender.ClientID %>').value = gender;

    var error = "";
    if (isNaN(age) || isNaN(height) || isNaN(weight) || age <= 0 || height <= 0 || weight <= 0) {
        if (isNaN(age)) {
            error += "Please enter age\n";
        }
        if (isNaN(height)) {
            error += "Please enter height\n";
        }
        if (isNaN(weight)) {
            error += "Please enter weight\n";
        }
        if (height <= 0) {
            error += "Please enter valid height\n";
        }
        if (weight <= 0) {
            error += "Please enter valid weight\n";
        }
        if (age < 18) {
            error += "Please enter valid university student age (more than 18 years old)\n";
        }
    }
    alert(error);
    return;
}

var bmi = weight / (Math.pow((height / 100), 2));
document.getElementById('<%= hiddenBMI.ClientID %>').value = bmi.toFixed(2);

var interpretation = getInterpretation(bmi, age, gender);

var resultElement = document.getElementById("result");
resultElement.innerHTML = `Your BMI is: ${bmi.toFixed(2)}  
Interpretation: ${interpretation}`;

var textBodyColor = getBodyColor(age, bmi, gender, interpretation);
resultElement.className = textBodyColor;

var pic = getPic(age, bmi, gender, interpretation);
var BMIBodyPic = document.getElementById("BMIBodyPic");
BMIBodyPic.innerHTML = `![BMI body Picture](Image/${pic})`;

var recommendation = getRecommendation(interpretation);
var BMIreco = document.getElementById("BMIreco");
BMIreco.innerHTML = `Recommendation: ${recommendation}`;

var healthyMinWeight, healthyMaxWeight;

if (interpretation == "Thinness") {
    healthyMinWeight = (18.5 * Math.pow((height / 100), 2));
    healthyMaxWeight = (24.9 * Math.pow((height / 100), 2));
    var gainWeight = ((healthyMinWeight - weight));
    BMIreco.innerHTML = `Recommendation: ${recommendation}  
`;
    BMIreco.innerHTML += `Healthy weight for the height: ${healthyMinWeight.toFixed(2)} kg - ${healthyMaxWeight.toFixed(2)} kg  
`;
    BMIreco.innerHTML += `Gain ${gainWeight.toFixed(2)} kg to reach a BMI of 18.5 kg/m².`;
} else if (interpretation == "Overweight" || interpretation == "Obese") {
    healthyMinWeight = (18.5 * Math.pow((height / 100), 2));
    healthyMaxWeight = (24.9 * Math.pow((height / 100), 2));
    var loseWeight = (weight - healthyMaxWeight);
    BMIreco.innerHTML = `Recommendation: ${recommendation}  
`;
    BMIreco.innerHTML += `Healthy weight for the height: ${healthyMinWeight.toFixed(2)} kg - ${healthyMaxWeight.toFixed(2)} kg  
`;
    BMIreco.innerHTML += `Lose ${loseWeight.toFixed(2)} kg to reach a BMI of 24.9 kg/m².`;
} else {
    healthyMinWeight = (18.5 * Math.pow((height / 100), 2));
    healthyMaxWeight = (24.9 * Math.pow((height / 100), 2));
    BMIreco.innerHTML = `Recommendation: ${recommendation}  
`;
    BMIreco.innerHTML += `Healthy weight for the height: ${healthyMinWeight.toFixed(2)} kg - ${healthyMaxWeight.toFixed(2)} kg`;
}

var bmiPrime = (bmi / 25).toFixed(2);
BMIreco.innerHTML += `  
BMI Prime: ${bmiPrime}`;

var ponderalIndex = (weight / Math.pow((height / 100), 3)).toFixed(2);
BMIreco.innerHTML += `  
Ponderal Index: ${ponderalIndex}`;

document.getElementById('<%= btnSave.ClientID %>').classList.remove('hidden');

PageMethods.CompareBMI(onSuccess, onError);
}

```

This code collects user inputs, validates them, calculates the BMI, stores necessary data in hidden fields and then updates the user interface with the calculated BMI, its interpretation and a visual representation through category images.

```
function updateChart(date, bmi) {
    var currentDate = new Date().toLocaleString(); // Get current date and time
    bmiChart.data.labels.push(currentDate);
    bmiChart.data.datasets[0].data.push(bmi);
    bmiChart.update();

    //save updated data to localStorage
    var savedData = {
        labels: bmiChart.data.labels,
        data: bmiChart.data.datasets[0].data
    };
    localStorage.setItem("bmiChartData", JSON.stringify(savedData));
}

function resetChart() {
    //clear the chart data and update
    bmiChart.data.labels = [];
    bmiChart.data.datasets[0].data = [];
    bmiChart.update();

    //clear localStorage
    localStorage.removeItem("bmiChartData");
}

function redoChart() {
    //remove latest data point
    if (bmiChart.data.labels.length > 0) {
        bmiChart.data.labels.pop();
        bmiChart.data.datasets[0].data.pop();
        bmiChart.update();

        //save updated data to localStorage
        var savedData = {
            labels: bmiChart.data.labels,
            data: bmiChart.data.datasets[0].data
        };
        localStorage.setItem("bmiChartData", JSON.stringify(savedData));
    }
}
```

This is the code snippet of the BMI chart to show line chart BMI data visualisation.

```

private void compareBMI()
{
    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string studentID = HttpContext.Current.Session["UserID"].ToString();

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            // Get the latest record
            string queryLatest = @""
SELECT TOP 1 Weight, BMIValue
FROM HealthyValue
WHERE studentID = @studentID AND BMIValue IS NOT NULL
ORDER BY dateTime DESC";

            SqlCommand commandLatest = new SqlCommand(queryLatest, connection);
            commandLatest.Parameters.AddWithValue("@studentID", studentID);

            SqlDataReader readerLatest = commandLatest.ExecuteReader();
            if (readerLatest.Read())
            {
                decimal latestWeight = readerLatest.GetDecimal(0);
                decimal latestBMI = readerLatest.GetDecimal(1);
                readerLatest.Close();

                // Get the previous record
                string queryPrevious = @""
SELECT TOP 1 Weight, BMIValue
FROM HealthyValue
WHERE studentID = @studentID AND BMIValue IS NOT NULL
AND dateTime < (
                SELECT MAX(dateTime)
                FROM HealthyValue
                WHERE studentID = @studentID AND BMIValue IS NOT NULL)
ORDER BY dateTime DESC";

                SqlCommand commandPrevious = new SqlCommand(queryPrevious, connection);
                commandPrevious.Parameters.AddWithValue("@studentID", studentID);

                SqlDataReader readerPrevious = commandPrevious.ExecuteReader();
                if (readerPrevious.Read())
                {
                    decimal previousWeight = readerPrevious.GetDecimal(0);
                    decimal previousBMI = readerPrevious.GetDecimal(1);

                    if (previousBMI < 18.5m) // Underweight
                    {
                        if (latestWeight > previousWeight)
                        {
                            lblCong.Text = "Congratulations! You have increased your weight from underweight (" + previousWeight + "kg -> " + latestWeight + "kg ) keep it up!";
                            lblCong.CssClass = "glow-bold";
                        }
                        else
                        {
                            lblCong.Text = "Unfortunately, You have decreased your weight from underweight (" + previousWeight + "kg -> " + latestWeight + "kg )";
                            lblCong.CssClass = "glow-bold";
                        }
                    }
                    else if (previousBMI >= 30m) // Obese
                    {
                        if (latestWeight < previousWeight)
                        {
                            lblCong.Text = "Congratulations! You have successfully reduced your weight from obese (" + previousWeight + "kg -> " + latestWeight + "kg ) keep it up!";
                            lblCong.CssClass = "glow-bold";
                        }
                        else
                        {
                            lblCong.Text = "Unfortunately, You have increased your weight from obese (" + previousWeight + "kg -> " + latestWeight + "kg )";
                            lblCong.CssClass = "glow-bold";
                        }
                    }
                    else if (previousBMI >= 25m) // Overweight
                    {
                        if (latestWeight < previousWeight)
                        {
                            lblCong.Text = "Congratulations! You have successfully reduced your weight from overweight (" + previousWeight + "kg -> " + latestWeight + "kg ) keep it up!";
                            lblCong.CssClass = "glow-bold";
                        }
                        else
                        {
                            lblCong.Text = "Unfortunately, You have increased your weight from overweight (" + previousWeight + "kg -> " + latestWeight + "kg )";
                            lblCong.CssClass = "glow-bold";
                        }
                    }
                    else
                    {
                        lblCong.Text = "No previous record found for comparison.";
                    }
                }
                else
                {
                    lblCong.Text = "No latest record found for comparison.";
                }
            }
        }
    catch (Exception ex)
    {
        lblCong.Text = "An error occurred while comparing BMI data: " + ex.Message;
    }
}

```

This is the code to get latest and previous BMI record using SQL query, it compares students' latest weight to the previous weight and based on the three categories to output different feedback accordingly.

Calorie Calculator

Calorie Calculator

Recent Data :

Apply Height & Weight	dateTime	Height	Weight	CalorieValue	TargetWeight
Apply	8/30/2024 10:47:33 AM	150	80	1766.86	80
Apply	8/30/2024 10:47:12 AM	150	78	1742.88	80
Apply	8/30/2024 10:46:50 AM	150	79	1754.87	80
Apply	8/30/2024 10:45:50 AM	150	70	1646.98	80
Apply	8/26/2024 9:11:05 PM	150	80	1766.86	80

12345

Apply Age & Gender	studentGender	Age
Apply	Female	21

Age (years):

21

Gender:

Female

Height (cm):

150

Weight (kg):

76

Target Weight (kg):

80

Number of weeks to Reach Target Weight:

34

Activity Level:

Exercise: 15-30 minutes of elevated heart rate activity.
Intense exercise: 45-120 minutes of elevated heart rate activity.
Very intense exercise: 2+ hours of elevated heart rate activity.

Sedentary: little or no exercise

Calculate

Cancel

Save

Maintain Current Weight: 1718.90 kcal/day

Mild Weight Gain (0.25 kg/week): 1968.90 kcal/day

Weight Gain (0.5 kg/week): 2218.90 kcal/day

Extreme Weight Gain (1 kg/week): 2718.90 kcal/day

=====

Calories for Target Weight: 1766.86 kcal/day

You need to eat 47.95 more kcal/day

Number of days to continue eating according to Calories to Reach Target Weight: 1343 days (192 weeks) for gaining 0.25 kg/week

Number of days to continue eating according to Calories to Reach Target Weight: 671 days (96 weeks) for gaining 0.5 kg/week

Number of days to continue eating according to Calories to Reach Target Weight: 336 days (48 weeks) for gaining 1 kg/week

Sorry, you cannot reach your target weight within 34 week(s)

====RECOMENDED/OPTIMUM CALORIES(NOT ACCORDING YOUR TARGET WEIGHT)====

You can monitor your weight progress over the next few weeks and adjust your calorie intake accordingly based on how quickly you lose or gain weight.

Adjusted Calories for Goal: 2018.90 to 2118.90 (more better) kcal/day

Protein Goal: 76.00 to 91.20 grams/day

Fat Intake: 38.20 to 76.40 grams/day

COMMON FOOD Calories Chart			
FRUIT	PORTION	CALORIES	FIBER
Apple	1 medium	70	4.0
Banana	1 medium	96	3.0
Berries	1/2 cup	20-33	15-4.6
Cherries	1/2 cup	55	1.0
Grapefruit	1/2 medium	30	0.8
Grapes	20	75	1.0
Melon, honeydew, cantaloupe	1/2 cup	42-76	1.5-2
Orange	1 large	70	2.4
Peach	1 medium	38	2.3
Pear	1 medium	88	4.0
Pineapple	1/2 cup	41	0.8
Plums	2 or 3 small	38-45	2.0
Raisins	1 tbsp	29	1.0

GRAINS, RICE, BREADS, POTATOES			
PORTION	CALORIES	FIBER	
Bread, wheat	2 slices	120	3.6
Bread, white	2 slices	160	1.9
Lentils, brown	2/3 cup	144	4.4
Macaroni pasta	1 cup	200	5.7
Potato, baked	1 small	120	4.2
Sweet potato, baked	1 small	146	4.0
Rice, white (before cooking)	1/2 cup	79	2.0
Rice, brown (before cooking)	1/2 cup	83	5.5

PROTEINS SOURCES (COOKED)			
PORTION	CALORIES	FIBER	
Chicken or turkey breast	3 oz.	135-140	25.0
Chicken thigh	3.5 oz.	120	10.0
Lean red meat	3 oz.	145-160	25.0
Beef hot dog	1/4 lb.	352	4.0
Ham	2 thin slices	75	22.0
Shrimp, crab, lobster	4 oz.	120-150	22-24
Tuna	1/2 cup	90-120	25.0
Ocean fish	4 oz.	130-170	25-30
Egg whites	7 egg whites	115	25.0
Nonfat cottage cheese	1 cup	140	28.0

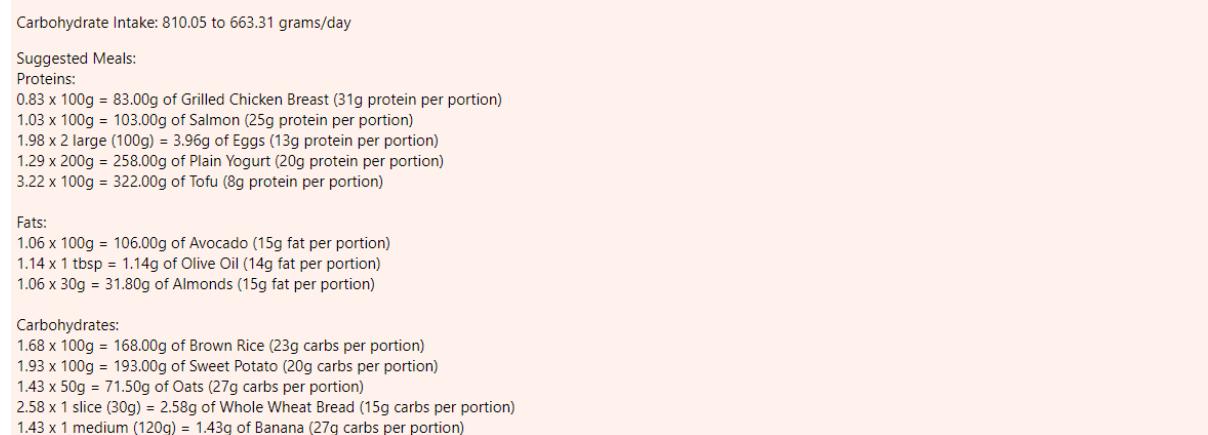


Figure 5.8 Calorie Calculator Page

This code firstly shows the input validation and error handling, checks if all required inputs are valid numbers and within acceptable ranges like age more than and equal to 18, weight and height more than 0. Then it uses the Mifflin-St Jeor formula to calculate the BMR for both males and females based on the provided weight, height, age and gender. It calculates the number of calories needed for different weight change rates (mild, normal and extreme weight gain/loss) depends on whether the target weight is greater or less than the current weight. Then it estimates the number of days and weeks required to reach the target weight based on the difference between current and target maintenance calories. It also determines the protein, fat and carbohydrate intake needed to meet the adjusted calorie goals for weight gain, loss, or maintenance with suggestion of food portions.

```

private decimal GetPreviousTargetWeight(string studentID)
{
    decimal previousTargetWeight = 0;
    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "SELECT TOP 1 TargetWeight FROM HealthyValue WHERE studentID = @StudentID AND CalorieValue IS NOT NULL ORDER BY dateCreated DESC";
            SqlCommand command = new SqlCommand(query, connection);
            {
                command.Parameters.AddWithValue("@StudentID", studentID);
                connection.Open();
                object result = command.ExecuteScalar();
                if (result != null)
                {
                    previousTargetWeight = Convert.ToDecimal(result);
                }
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred while retrieving previous target weight: " + ex.Message);
    }

    return previousTargetWeight;
}
private decimal GetPreviousWeight(string studentID)
{
    decimal previousWeight = 0;
    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

    try
    {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            conn.Open();
            string query = "SELECT TOP 1 Weight FROM HealthyValue WHERE studentID = @StudentID AND CalorieValue IS NOT NULL ORDER BY dateCreated DESC";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@StudentID", studentID);
            var result = cmd.ExecuteScalar();

            if (result != null)
            {
                previousWeight = Convert.ToDecimal(result);
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred while fetching the previous weight: " + ex.Message);
    }

    return previousWeight;
}

```

It retrieves the previous target weight and previous weight using SQL Query to retrieve from database.

```
if (targetWeight1 == previousTargetWeight)
{
    if (currentWeight == targetWeight1)
    {
        message = "Congratulations! You have achieved your target weight, " + targetWeight1.ToString("F0") + " kg";
        isTargetAchieved = true;
    }
    else if (Math.Abs(currentWeight - targetWeight1) < Math.Abs(previousWeight - targetWeight1))
    {
        message = "You are moving towards your target weight, " + targetWeight1.ToString("F0") + " kg from " + previousWeight.ToString("F0") + " kg -> " + currentWeight.ToString("F0") + " kg. Nice!";
        isTargetAchieved = true;
    }
    else
    {
        message = "Sadly, you are further away from your target weight, " + targetWeight1.ToString("F0") + " kg than before (" + previousWeight.ToString("F0") + " kg -> " + currentWeight.ToString("F0") + " kg). Please keep it up.";
        isTargetAchieved = false;
    }
}

lblCong.Text = message;
lblCong.CssClass = "glow-bold";
}
```

Then it will check if the current target weight is same with previous target weight, it will compare the current weight and the target weight to check whether the students has moving towards, away from or achieve the target weight and output different message based on different situations.

Exercise Workout Schedule

Workout Schedule

Suggestion:

1. Low-Impact Cardio: walking, cycling, and swimming.
Goal: Burn calories while minimizing joint stress.
2. Strength Training: bodyweight exercises (e.g., squats, lunges) and resistance band training.
Goal: Build muscle mass and boost metabolism.
3. Flexibility and Mobility: stretching and yoga.
Goal: Improve flexibility and support overall physical activity.

Day:

Time:

Exercise:

Add Exercise

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12:00 AM - 1:00 AM	swim	running	squats	shoulder workout	jogging	jog	slim leg
1:00 AM - 2:00 AM	plank	crunch				squat	
2:00 AM - 3:00 AM	crunch		jog				lunge
3:00 AM - 4:00 AM	run		jog			leg raises	
4:00 AM - 5:00 AM					squats		
5:00 AM - 6:00 AM							
6:00 AM - 7:00 AM	run						
7:00 AM - 8:00 AM							
8:00 AM - 9:00 AM							
9:00 AM - 10:00 AM							
10:00 AM - 11:00 AM							

11:00 AM - 12:00 PM							
12:00 PM - 1:00 PM							
1:00 PM - 2:00 PM							jogging
2:00 PM - 3:00 PM							
3:00 PM - 4:00 PM							
4:00 PM - 5:00 PM							
5:00 PM - 6:00 PM							
6:00 PM - 7:00 PM							
7:00 PM - 8:00 PM							
8:00 PM - 9:00 PM							
9:00 PM - 10:00 PM							
10:00 PM - 11:00 PM							
11:00 PM - 12:00 AM							

Enter Exercise Concern or Any Unknown Exercise Details:

Search

Figure 5.9 Exercise Workout Schedule Page

```

window.addExercise = function () {
    var dayInput = document.getElementById("day");
    var timeInput = document.getElementById("time");
    var exerciseInput = document.getElementById("exercise");

    if (exerciseInput.value.trim() === "") {
        alert("Please enter exercise name.");
        return;
    }

    var timeSlot = timeInput.value;
    var selectedDay = dayInput.value;
    var exerciseName = exerciseInput.value.trim();

    dayInput.selectedIndex = 0;
    timeInput.selectedIndex = 0;
    exerciseInput.value = "";

    var tr = tbody.querySelector(`tr[data-time="${timeSlot}"]`);
    if (!tr) {
        console.error("Time slot not found:", timeSlot);
        return;
    }

    var dayCell = tr.querySelector(`td[data-day="${selectedDay}"]`);
    if (!dayCell) {
        console.error("Day cell not found:", selectedDay);
        return;
    }

    dayCell.classList.add("exercise-cell");

    PageMethods.SaveIntoExerciseSchedule(selectedDay, timeSlot, exerciseName, function (result) {
        if (result === "EXISTS") {
            var replace = confirm("An exercise already exists for this day and time. Do you want to replace it?");
            if (replace) {
                PageMethods.ReplaceExercise(selectedDay, timeSlot, exerciseName, function (replaceResult) {
                    alert(replaceResult);
                    location.reload();
                }, function (error) {
                    console.error(error);
                });
            } else {
                alert("No replace");
                location.reload();
            }
        } else {
            location.reload();
        }
    }, function (error) {
        console.error(error);
    });
};

function onSaveSuccess(response) {
    alert(response);
    document.getElementById('<%= lblMessage.ClientID %>').innerText = response;
    document.getElementById('<%= lblMessage.ClientID %>').style.color = "green";
}

function onSaveFailure(error) {
    alert("Error: " + error.get_message());
    document.getElementById('<%= lblMessage.ClientID %>').innerText = "Error: " + error.get_message();
    document.getElementById('<%= lblMessage.ClientID %>').style.color = "red";
}

```

This code handles user input which are date time and exercise name and validates the data. The PageMethods.SaveIntoExerciseSchedule function is called to save the exercise into the schedule. This function communicates with the server using ASP.NET AJAX PageMethods. It takes the day, time slot and exercise name as parameters along with two callback functions, one for success and one for failure error handling. If the server response (result) is "EXISTS", it means an exercise is already scheduled for the selected day and time. The function prompts the user with a confirm dialog asking if they want to replace the existing exercise. If the user confirms, PageMethods.ReplaceExercise is called to replace the exercise,

and the page is reloaded after the operation completes. If no exercise exists at the specified timeslot the page is reloaded to reflect the updated schedule.

```
function displayYouTubeResultsKeyword(query) {
    var apiKey = '<--APIKEY-->';
    var apiUrl = `https://www.googleapis.com/youtube/v3/search?part=snippet&q=${query}&type=video&key=${apiKey}&channelId=${channelID}`;

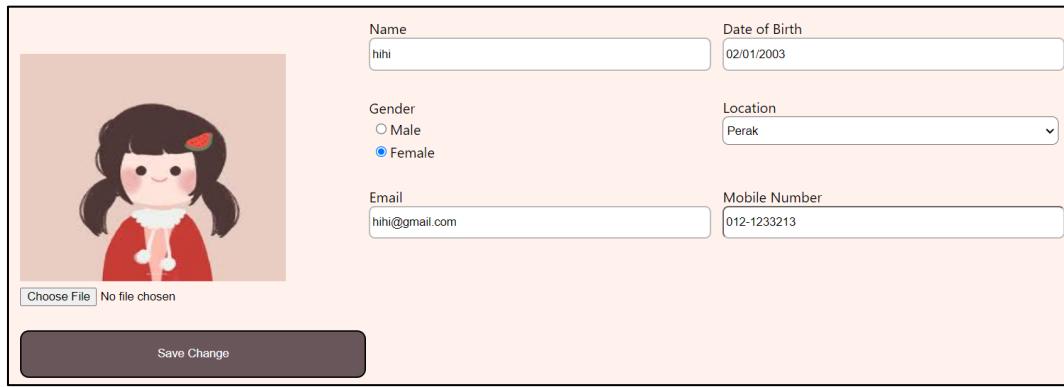
    fetch(apiUrl)
        .then(response => response.json())
        .then(data => {
            youtubeResultsKeywordDiv.innerHTML = "";
            data.items.forEach(item => {
                var videoUrl = `https://www.youtube.com/watch?v=${item.id.videoId}`;
                var videoTitle = item.snippet.title;

                if (isExerciseVideo(videoTitle)) {
                    var resultItem = document.createElement("div");
                    resultItem.innerHTML = `\${videoTitle}`;
                    youtubeResultsKeywordDiv.appendChild(resultItem);
                }
            });
        })
        .catch(error => console.error('Error fetching YouTube results:', error));
}

channelID = 'UCDU1DJcPP00QK-3UrxEyhAQ' || 'UC7t6QJ4u8qF8pI-vibX-BUQ' || 'UCPw6LgHyAHu17_Khqd1LyXw';
```

The API key used to authenticate requests to the YouTube Data API. It constructs the URL for the YouTube Data API search endpoint, including query parameters for the search term (query), the type of result (video), the API key and a channel ID. It uses the fetch API to make a GET request to the YouTube Data API. The response is converted to JSON format. If the fetch request fails, it logs an error to the console. Any existing content in the youtubeResultsKeywordDiv element will be clear to ensure that only the latest results are displayed. The forEach loop iterates over each item in the returned data.items array from the YouTube API response. For each item, the video URL and title are constructed using the video ID and snippet title from the API response. The isExerciseVideo(videoTitle) function checks if the video title matches certain criteria, then a new div element is created containing a link to the video then appended to youtubeResultsKeywordDiv. Overall, YouTube Data API and JavaScript fetch API are used to retrieve and display the videos dynamically on the webpage.

Personal Profile



The screenshot shows a user profile form. At the top left is a placeholder image of a girl with pigtails. To the right of the image are fields for Name (hihi), Date of Birth (02/01/2003), Gender (Female selected), Location (Perak), Email (hihi@gmail.com), and Mobile Number (012-1233213). Below these fields is a "Choose File" button with the message "No file chosen". At the bottom is a large dark grey "Save Change" button.

Figure 5.10 Personal Profile Page

```

private void UpdateUserProfile(string username, string name, string dob, string gender, string state, string email, string mobileNo, string imagePath)
{
    DateTime dobDate;
    if (!DateTime.TryParseExact(dob, "dd/MM/yyyy", CultureInfo.InvariantCulture, DateTimeStyles.None, out dobDate))
    {
        lblMessage.Text = "Invalid date format for date of birth.(20/12/2002)";
        lblMessage.Visible = true;
        return;
    }

    TimeSpan ageDifference = DateTime.Now - dobDate;
    int age = (int)(ageDifference.Days / 365.25);
    if (age < 18)
    {
        lblMessage.Text = "Only university students who are 18 years old or older are eligible to register an account.";
        lblMessage.Visible = true;
        return;
    }

    string phonePattern = @"^01[0-9]{2}[0-9]{7}$|^01[0-9]{8}$";
    if (!Regex.IsMatch(mobileNo, phonePattern))
    {
        lblMessage.Text = "Invalid phone number format. Please enter correct mobile number 01x-xxxxxx or 01x-xxxxxxx";
        lblMessage.Visible = true;
        return;
    }

    string userType = Session["UserType"].ToString();

    String mobileNum = GetMobileNum(username, userType);
    if (mobileNum != mobileNo)
    {
        if (!IsPhoneUnique(mobileNo))
        {
            lblMessage.Text = "Mobile number already exists. Please choose a different mobile number.";
            lblMessage.Visible = true;
            return;
        }
    }
    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string query = "";

    switch (userType)
    {
        case "Student":
            query = "UPDATE Student SET studentName = @Name, studentDOB = @DOB, studentGender = @Gender, studentState = @State, studentEmail = @Email, studentMobileNo = @MobileNo";
            if (!string.IsNullOrEmpty(imagePath))
            {
                query += ", studentImage = @ProfileImage";
            }
            else
            {
                query += ", studentImage = @ProfileImage";
            }
            query += " WHERE studentUsername = @Username OR studentEmail = @Email";
            break;
        case "Admin":
            query = "UPDATE Admin SET adminName = @Name, adminDOB = @DOB, adminGender = @Gender, adminState = @State, adminEmail = @Email, adminMobileNo = @MobileNo";
            if (!string.IsNullOrEmpty(imagePath))
            {
                query += ", adminImage = @ProfileImage";
            }
            else
            {
                query += ", adminImage = @ProfileImage";
            }
            query += " WHERE adminUsername = @Username OR adminEmail = @Email";
            break;
    }
}

```

```
default:  
    Response.Write("Invalid user type.");  
    return;  
}  
  
using (SqlConnection connection = new SqlConnection(connectionString))  
{  
    using (SqlCommand command = new SqlCommand(query, connection))  
    {  
        command.Parameters.AddWithValue("@Name", name);  
        command.Parameters.AddWithValue("@DOB", dobDate);  
        command.Parameters.AddWithValue("@Sex", sex);  
        command.Parameters.AddWithValue("@State", state);  
        command.Parameters.AddWithValue("@Email", email);  
        command.Parameters.AddWithValue("@MobileNo", mobileNo);  
        command.Parameters.AddWithValue("@Username", username);  
        if (!string.IsNullOrEmpty(imagePath))  
        {  
            command.Parameters.AddWithValue("@ProfileImage", imagePath);  
        }  
        else  
        {  
            Image1.ImageUrl = "/Image/" + imagePath;  
            Image1.Visible = true;  
            boyProfile.Visible = false;  
            girlProfile.Visible = false;  
            command.Parameters.AddWithValue("@ProfileImage", imagePath);  
        }  
        try  
        {  
            connection.Open();  
            int rowsAffected = command.ExecuteNonQuery();  
            if (rowsAffected > 0)  
            {  
                lblMessage.Text = "Profile updated successfully!";  
            }  
            else  
            {  
                lblMessage.Text = "Failed to update profile. Please try again.";  
            }  
        }  
        catch (Exception ex)  
        {  
            lblMessage.Text = "An error occurred: " + ex.Message;  
        }  
    }  
}
```

This is the code used to update profile, it ensures that user input is correctly formatted and valid before attempting to update the database. It also uses parameterized queries to avoid SQL injection attacks. It uses SQL queries to update the database based on different role.

To-Do List

The screenshot shows a 'To-Do List' interface with the following sections:

- Important & Urgent - Do first:** Contains two tasks: "play3457u" (checked) and "ghngf".
- Important & Not Urgent - Schedule:** Contains two tasks: "wewr" (checked) and "er".
- Not Important & Urgent - Delegate:** Contains two tasks: "popo" and "43".
- Not Important & Not Urgent - Delete/Postpone:** Contains two tasks: "54634" (checked) and "uyi76".

Below the categories, there is a search section with dropdowns for 'Important' and 'Search Urgency', and a checkbox for 'Not Completed Task'. The search results show 'er' and 'ghngf'.

Figure 5.11 To-Do List Page

```

private bool AddTaskToDatabase(string taskContent, int category)
{
    if (string.IsNullOrWhiteSpace(taskContent))
    {
        string alertMessage = "Please write something.";
        ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('{alertMessage}'); window.location.href='ToDo.aspx';", true);
        return false;
    }

    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string studentID = Session["UserID"].ToString();
    string newToDoID = GenerateNextID();

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "INSERT INTO ToDoList (ToDoID, ToDoContent, studentID, Category) VALUES (@ToDoID, @ToDoContent, @studentID, @Category)";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@ToDoID", newToDoID);
                command.Parameters.AddWithValue("@ToDoContent", taskContent);
                command.Parameters.AddWithValue("@studentID", studentID);
                command.Parameters.AddWithValue("@Category", category);

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
        return false;
    }
}

```

This is the code to add task in database using SQL query.

```

protected void chkComplete7_CheckedChanged(object sender, EventArgs e)
{
    CheckBox chkComplete = (CheckBox)sender;
    GridViewRow row = (GridViewRow)chkComplete.NamingContainer;
    string ToDoID = GridView7.DataKeys[row.RowIndex].Value.ToString();
    bool isCompleted = chkComplete.Checked;

    UpdateTaskCompletionStatus(ToDoID, isCompleted);

    row.Style["text-decoration"] = isCompleted ? "line-through" : "none";
}

```

This is the code snippet to allow student to mark tasks as complete or incomplete directly from the grid view. When a student checks or unchecks a checkbox, it identifies the row and task associated with the checkbox. Then it updates the task's completion status in the database. It will visually reflects this change by striking through the task in the grid view row to have immediate feedback on the task's status.

```

protected void GridView7_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        CheckBox chkComplete = (CheckBox)e.Row.FindControl("chkComplete");

        // Ensure chkComplete is not null
        if (chkComplete != null)
        {
            // Check if DataItem is not null
            if (e.Row.DataItem != null)
            {
                var isCompletedObj = DataBinder.Eval(e.Row.DataItem, "IsCompleted");
                bool isCompleted = isCompletedObj != null && Convert.ToBoolean(isCompletedObj);

                chkComplete.Checked = isCompleted;
                e.Row.Style["text-decoration"] = isCompleted ? "line-through" : "none";
            }
        }
    }
}

```

This code retrieves and checks a checkbox for each task in the row to reflect its completion status. It visually indicates completed tasks by striking through the text in the row. It ensure students have a immediate and accurate information about each task's status.

Goal Getter

Goal Getter

Goal:

Reward:

Milestones:

Motivational Quote:

Add Goal **Cancel**

	GoalTitle	GoalReward	GoalMilestone	GoalQuote	Completed
Edit Delete	Maintain president list	play game	read hard, work hard, revise hard	Don't limit myself	<input checked="" type="checkbox"/>
Edit Delete	ABC	efgewr	fdwer	r	<input type="checkbox"/>
Edit Delete	sfd	wer	re		<input type="checkbox"/>
Edit Delete	a	a	a	a	<input checked="" type="checkbox"/>
Edit Delete	ewq	123	sda		<input type="checkbox"/>
Edit Delete	ret	34	345		<input type="checkbox"/>

Figure 5.12 Goal Getter Page

```

private bool AddGoalToDatabase(string goal, string reward, string milestone, string quote)
{
    if (string.IsNullOrWhiteSpace(goal))
    {
        string alertMessage = "Please write something.";
        ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('{alertMessage}');", true);
        return false;
    }

    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string studentID = Session["UserID"].ToString();
    string newGoalID = GenerateNextID();

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "INSERT INTO Goal (GoalID, GoalTitle, GoalReward, GoalMilestone, GoalQuote, studentID) VALUES (@ID, @title, @reward, @milestone, @quote, @studentID)";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@ID", newGoalID);
                command.Parameters.AddWithValue("@title", goal);
                command.Parameters.AddWithValue("@reward", reward);
                command.Parameters.AddWithValue("@milestone", milestone);
                command.Parameters.AddWithValue("@quote", quote);
                command.Parameters.AddWithValue("@studentID", studentID);

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
        return false;
    }
}

```

This is the SQL query to add goal into database.

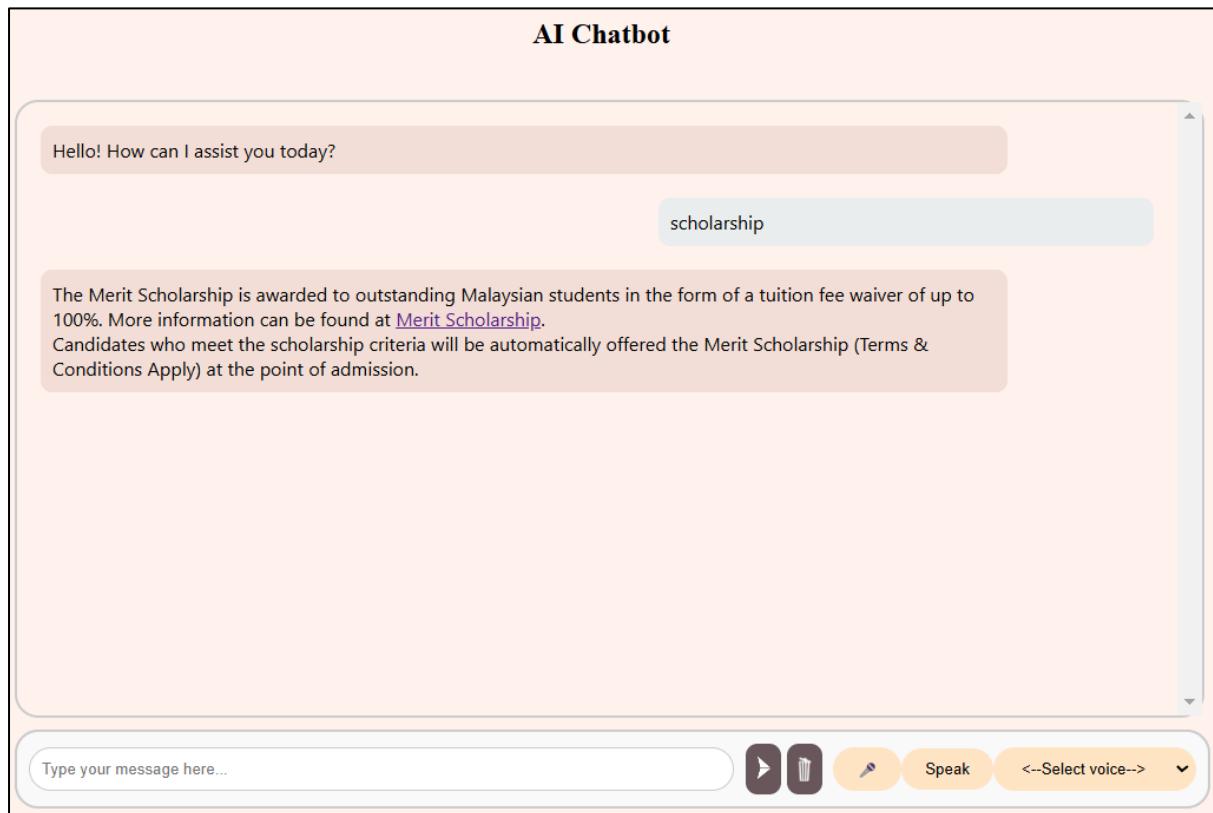
AI Chatbot

Figure 5.13 AI Chatbot Page

```

# Global variables
vectorizer = None
best_model = None

def train_models():
    global vectorizer, best_model

    # Function to perform synonym replacement
    def synonym_replacement(tokens, limit):
        augmented_sentences = []
        for i in range(len(tokens)):
            synonyms = []
            for syn in wordnet.synsets(tokens[i]):
                for lemma in syn.lemmas():
                    synonyms.append(lemma.name())
            if len(synonyms) > 0:
                num_augmentations = min(limit, len(synonyms))
                sampled_synonyms = random.sample(synonyms, num_augmentations)
                for syn in sampled_synonyms:
                    augmented_tokens = tokens[:i] + [synonyms] + tokens[i+1:]
                    augmented_sentences.append(" ".join(augmented_tokens))
        return augmented_sentences

    text_data = []
    labels = []
    stopwords = set(nltk.corpus.stopwords.words("english"))
    lemmatizer = WordNetLemmatizer()

    limit_per_tag = 40

    for intent in intents["intents"]:
        augmented_sentences_per_tag = 0
        for example in intent["examples"]:
            tokens = nltk.word_tokenize(example.lower())
            filtered_tokens = [
                lemmatizer.lemmatize(token)
                for token in tokens
                if token not in stopwords and token.isalpha()
            ]
            if filtered_tokens:
                text_data.append(" ".join(filtered_tokens))
                labels.append(intent["tag"])
                augmented_sentences_per_tag += 1
            if augmented_sentences_per_tag > limit_per_tag:
                break

        augmented_sentences = synonym_replacement(
            filtered_tokens, limit_per_tag - augmented_sentences_per_tag
        )
        for augmented_sentence in augmented_sentences:
            text_data.append(augmented_sentence)
            labels.append(intent["tag"])
        augmented_sentences_per_tag = 1
        if augmented_sentences_per_tag > limit_per_tag:
            break

    vectorizer = TfidfVectorizer()
    X = vectorizer.fit_transform(text_data)
    y = labels

    def find_best_model(X, y, test_size=0.2):
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size=test_size, random_state=100
        )

        models = [
            {
                "name": "Logistic Regression",
                "model": LogisticRegression(),
                "params": {
                    "penalty": ["L2"],
                    "C": [0.1, 1, 10],
                    "solver": ["liblinear"],
                    "max_iter": [100, 1000, 10000]
                }
            },
            {
                "name": "Multinomial Naive Bayes",
                "model": MultinomialNB(),
                "params": {"alpha": [0.1, 0.5, 1.0]}
            },
            {
                "name": "Linear SVC",
                "model": LinearSVC(),
                "params": {
                    "penalty": ["L2"],
                    "loss": ["hinge", "squared_hinge"],
                    "C": [0.1, 1, 10],
                    "max_iter": [100, 1000, 10000]
                }
            },
            {
                "name": "Decision Tree",
                "model": DecisionTreeClassifier(),
                "params": {
                    "max_depth": [5, 10, 20, None],
                    "min_samples_split": [2, 5, 10],
                    "min_samples_leaf": [1, 2, 4],
                    "criterion": ["gini", "entropy"]
                }
            },
            {
                "name": "Random Forest",
                "model": RandomForestClassifier(),
                "params": {
                    "n_estimators": [100, 200, 500],
                    "max_depth": [10, 20, None],
                    "min_samples_split": [2, 5, 10],
                    "min_samples_leaf": [1, 2, 4]
                }
            }
        ]

        best_score = 0
        best_model = None

        for name, model, param_grid in models:
            grid = GridSearchCV(model, param_grid, cv=3, n_jobs=-1)
            grid.fit(X_train, y_train)
            y_pred = grid.predict(X_test)
            score = accuracy_score(y_test, y_pred)
            print(f"({name}): {score:.4f} (best parameters: {grid.best_params_})")

            if score > best_score:
                best_score = score
                best_model = grid.best_estimator_

        print(f"\nBest model: {type(best_model).__name__} with score {best_score:.4f}")

    # Fit the best model to the full training data
    best_model.fit(X, y)

    # Save the trained model and vectorizer
    joblib.dump(vectorizer, "vectorizer.pkl")
    joblib.dump(best_model, "best_model.pkl")

    return best_model

best_model = find_best_model(X, y)

def chatbot_response(user_input):
    global vectorizer, best_model

    if vectorizer is None or best_model is None:
        vectorizer_path = (
            "C:/Users/SD01/Desktop/OneStopStudentSystem/vectorizer.pkl"
        )
        vectorizer = joblib.load(vectorizer_path)
        best_model_path = (
            "C:/Users/SD01/Desktop/OneStopStudentSystem/best_model.pkl"
        )
        best_model = joblib.load(best_model_path)

    input_text = vectorizer.transform([user_input])
    predicted_intent = best_model.predict(input_text)[0]

    for intent in intents["intents"]:
        if intent["tag"] == predicted_intent:
            response = random.choice(intent["responses"])
            return response

    return "Sorry, I didn't understand that."

if __name__ == "__main__":
    train_models()

```

This code uses synonym replacement to augment training data. It converts text data to numerical format using TF-IDF. Then it trains multiple machine learning models, evaluates them and selects the best one. It will save the vectorizer and best model for later use. It predicts the intent of user input and generates a response based on the best model.

```

private string GetChatbotResponse(string userInput)
{
    string batchFilePath = @"C:\Users\SEOW HUI CHEE\source\repos\OneStopStudentSystem\run_chatbot.bat";
    string command = $"{batchFilePath}\\" "{userInput}\\";

    ProcessStartInfo startInfo = new ProcessStartInfo
    {
        FileName = "cmd.exe",
        Arguments = $"/c \"{command}\"",
        RedirectStandardOutput = true,
        RedirectStandardError = true,
        UseShellExecute = false,
        CreateNoWindow = true
    };

    try
    {
        using (Process process = Process.Start(startInfo))
        {
            using (StreamReader outputReader = process.StandardOutput)
            {
                using (StreamReader errorReader = process.StandardError)
                {
                    string output = outputReader.ReadToEnd();
                    string error = errorReader.ReadToEnd();

                    process.WaitForExit();

                    if (!string.IsNullOrEmpty(error))
                    {
                        File.WriteAllText(@"C:\Users\SEOW HUI CHEE\source\repos\OneStopStudentSystem\error.txt", error);
                        return "An error occurred while processing your request.";
                    }

                    return output;
                }
            }
        }
    }
    catch (Exception ex)
    {
        File.WriteAllText(@"C:\Users\SEOW HUI CHEE\source\repos\OneStopStudentSystem\exception.txt", ex.ToString());
        return "An error occurred while processing your request.";
    }
}

```

It execute a batch file to get a response from the python script file.

```

@echo off
setlocal

:: Define the path to the Python script
set PYTHON_SCRIPT=chatbot.py
set WORKING_DIR="C:\Users\SEOW HUI CHEE\source\repos\OneStopStudentSystem"
set USER_INPUT=%1

:: Change to the working directory
cd /d %WORKING_DIR%

:: Run the Python script and capture output
python %PYTHON_SCRIPT% "%USER_INPUT%" > response.txt

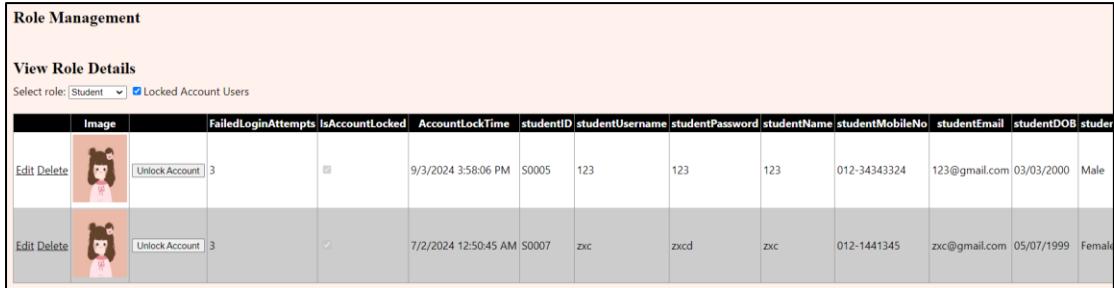
:: Check if response.txt was created and is not empty
if exist response.txt (
    type response.txt
) else (
    echo No response received or response.txt not found
)

endlocal

```

The batch file captures the output from the Python script and saves it to response.txt. The C# method reads the output from response.txt or handles any errors or exceptions that occur during execution and returns the chatbot's response.

Role Management



The screenshot shows a web-based application titled "Role Management". Under the heading "View Role Details", there is a dropdown menu set to "Student" and a checked checkbox labeled "Locked Account Users". Below this is a table with two rows of data. Each row contains columns for "Edit", "Delete", "Image" (showing a student icon), "FailedLoginAttempts" (3), "IsAccountLocked" (unchecked), "AccountLockTime" (9/3/2024 3:58:06 PM), "studentID" (S0005), "studentUsername" (123), "studentPassword" (123), "studentName" (123), "studentMobileNo" (012-34343324), "studentEmail" (123@gmail.com), "studentDOB" (03/03/2000), and "studentGender" (Male). The second row has similar values but with different student details.

Figure 5.14 Role Management Page

```
private void ResetAccount(string userType, string userID)
{
    string tableName = userType == "Student" ? "Student" : "TempUser";
    string idColumnName = userType == "Student" ? "studentID" : "tempUserID";

    string connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    string query = $"UPDATE {tableName} SET FailedLoginAttempts = 0, IsAccountLocked = 0, AccountLockTime = NULL WHERE {idColumnName} = @userID";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@userID", userID);
            connection.Open();
            command.ExecuteNonQuery();
        }
    }

    BindGridView();
}
```

This is the code for admin to reset account which is to reset the failed login attempt to be 0, set the account locked to false and set the account lock time to null.

```
if (Session["Username"] == null)
{
    string alertMessage = "Please login first.";
    ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('{alertMessage}'); window.location.href='Login.aspx';", true);
    return;
}
else if ((string)Session["UserType"] != "Admin")
{
    string alertMessage = "Please login as a Admin.";
    ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('{alertMessage}'); window.location.href='Login.aspx';", true);
    return;
}
```

This is the code where it will detect for the valid user type to access admin page. When user login successfully, it will create a session for the user that store username and user type. Then this session values are used to verify if the user is logged in and has the appropriate user type to ensure proper access control and security. If it does not fulfil the conditions, then it will show alert messages and redirect user to login page.

5.2 Testing

5.2.1 Test case

Sub-section numbering should be limited to a maximum of 3 levels (e.g. 5.3.1) in order to avoid confusion.

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC001	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, valid email address, valid username, valid password, valid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.
--

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hoho			

4	Enter valid mobile number Mobile number: 019-6502346			
5	Enter valid email address Email address: hoho@gmail.com			
6	Enter valid username Username: hoho			
7	Enter valid password Password: Hoho@123			
8	Enter valid retype password RePassword: Hoho@123			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Successful register an account	Pass	

Post-conditions: System will display Login page
--

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC002	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, invalid mobile number, valid email address, valid username, valid password, valid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter invalid mobile number Mobile number: 012-3234345			
5	Enter valid email address Email address: hohoho@gmail.com			
6	Enter valid username Username: hohoho			

7	Enter valid password Password: Hohoho@123			
8	Enter valid retype password RePassword: Hohoho@123			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Mobile number already exists. Please choose a different mobile number.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC003	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, invalid email address, valid username, valid password, valid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter valid mobile number Mobile number: 016-2745596			
5	Enter invalid email address Email address: hoho@gmail.com			
6	Enter valid username Username: hohoho			
7	Enter valid password Password: Hohoho@123			

8	Enter valid retype password RePassword: Hohoho@123			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Email already exists. Please choose a different email.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC004	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, valid email address, invalid username, valid password, valid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter valid mobile number Mobile number: 016-2745596			
5	Enter valid email address Email address: hohoho@gmail.com			
6	Enter invalid username Username: hoho			
7	Enter valid password Password: Hohoho@123			

8	Enter valid retype password RePassword: Hohoho@123			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Username already exists. Please choose a different username.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC005	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, valid email address, valid username, invalid password, valid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter valid mobile number Mobile number: 016-2745596			
5	Enter valid email address Email address: hohoho@gmail.com			
6	Enter valid username Username: hohoho			
7	Enter invalid password Password: hoho			

8	Enter valid retype password RePassword: hoho			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Password must be at least 8 characters long, contain at least one uppercase letter, one lowercase letter, one number, and one special character”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC006	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, valid email address, valid username, valid password, invalid retype Password, valid date of birth (select year of 2006 or before), select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.
--

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter valid mobile number Mobile number: 016-2745596			
5	Enter invalid email address Email address: hohoho@gmail.com			
6	Enter valid username			

	Username: hohoho			
7	Enter valid password Password: Hohoho@123			
8	Enter invalid retype password RePassword: hoho			
9	Enter valid DOB DOB: 01/01/2000			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Password does not match”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC007	Test Case Name: Register
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test register function with full name, valid mobile number, valid email address, valid username, valid password, valid retype Password, invalid date of birth, select gender, select location, select check box	

Pre-conditions: The username must be unique and email and mobile number must not be registered yet.

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	display login page	Pass	
2	Click register button			
3	Enter full name Full name: hohoho			
4	Enter valid mobile number Mobile number: 016-2745596			
5	Enter invalid email address Email address: hohoho@gmail.com			
6	Enter valid username Username: hohoho			
7	Enter valid password			

	Password: Hohoho@123			
8	Enter valid retype password RePassword: Hohoho@123			
9	Enter invalid DOB DOB: 01/01/2020			
10	Select gender Ex: Female			
11	Select location Ex: Kedah			
12	Select checkbox			
13	Click Register button	Prompt error message “Only university students who are 18 years old or older are eligible to register an account.”	Pass	

Post-conditions: System remain to display error message and remain in the same page
--

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC008	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with valid email address and valid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter valid email address email address : hoho@gmail.com			
2	Enter valid password password : Hoho@123			
3	Click login button	Display system main page	Pass	

Post-conditions: System allow user to successfully login and display home page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC009	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with invalid email address and valid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter invalid email address email address : hohok@gmail.com			
2	Enter valid password password : Hohoho@123			
3	Click login button	prompt error message “Invalid username/email or password.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC010	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with valid email address and invalid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter valid email address email address : hoho@gmail.com			
2	Enter invalid password password : hohok			
3	Click login button	prompt error message “Invalid username/email or password.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC011	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with valid username and valid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter valid username username: hoho			
2	Enter valid password password : Hoho@123			
3	Click login button	Display system main page	Pass	

Post-conditions: System allow user to successfully login and display main page

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC012	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with invalid username and valid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter invalid username username: hohok			
2	Enter valid password password : Hoho@123			
3	Click login button	prompt error message “Invalid username/email or password.”	Pass	

Post-conditions: System remain to display error message and remain in the same page
--

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC013	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with valid username and invalid password	

Pre-conditions: OneStopStudentSystem_AccountModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Enter valid username username : hoho			
2	Enter invalid password password : hohok			
3	Click login button	prompt error message “Invalid username/email or password.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC014	Test Case Name: Login
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test login with Google account.	

Pre-conditions: Must have a Google account

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	Display login page	Pass	
2	Click Login with Google button	Display Choose an account page	Pass	
3	Select one of the Google account that have listed in Test Users in the Google Cloud Console	Display request additional access page	Pass	
4	Click continue	Display home page	Pass	

Post-conditions: System allow user to successfully login and display main page

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC015	Test Case Name: Reset password
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test reset password function with valid old password, valid new password and valid confirmed password.	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	Display login page	Pass	
2	Click reset password navigation link	Display reset password page	Pass	
3	Enter valid old password Old password: Hoho@123			
4	Enter valid new password New password: Hoho@1234			
5	Enter valid confirmed password Confirmed password: Hoho@1234			
6	Click reset password button	Successfully reset password	Pass	

Post-conditions: System allow user to successfully update new password in database.

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC016	Test Case Name: Reset password
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test reset password function with invalid old password, valid new password and valid confirmed password.	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	Display login page	Pass	
2	Click reset password navigation link	Display reset password page	Pass	
3	Enter invalid old password Old password: HoA@123			
4	Enter valid new password New password: Hoho@1234			
5	Enter valid confirmed password Confirmed password: Hoho@1234			
6	Click reset password button	Prompt error message “Old password does not match. Please try again.”	Pass	

Post-conditions: System remain to display error message and remain in the same page.

Test Case Template	
Test Case #: OneStopStudentSystem_AccountModule_TC017	Test Case Name: Reset password
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test reset password function with valid old password, invalid new password and valid confirmed password.	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	Display login page	Pass	
2	Click reset password navigation link	Display reset password page	Pass	
3	Enter valid old password Old password: Hoho@123			
4	Enter invalid new password New password: Ho1234			
5	Enter valid confirmed password Confirmed password: Ho1234			
6	Click reset password button	Prompt error message “Password must be at least 8 characters long, contain at least one uppercase letter, one lowercase letter, one number, and one	Pass	

	special character."		
--	---------------------	--	--

Post-conditions: System remain to display error message and remain in the same page.

Test Case Template

Test Case #: OneStopStudentSystem_AccountModule_TC018	Test Case Name: Reset password
System: OneStopStudentSystem	Module: Account Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test reset password function with valid old password, valid new password and invalid confirmed password.	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Run system	Display login page	Pass	
2	Click reset password navigation link	Display reset password page	Pass	
3	Enter valid old password Old password: Hoho@123			
4	Enter valid new password New password: Hoho@1234			
5	Enter invalid confirmed password Confirmed password: Hoho			
6	Click reset password button	Prompt error message “Passwords do not match.”	Pass	

Post-conditions: System remain to display error message and remain in the same page.

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC001	Test Case Name: ToDoList
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test to-do list function by adding to-do task	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click to-do list navigation link	Display to-do list page	Pass	
2	Enter to-do task task: read			
3	Click add button	Successful add a task and show the task	Pass	

Post-conditions: System allow user to successfully add to-do task in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC002	Test Case Name: ToDoList
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test to-do list function by edit to-do task	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click edit button	Display column to edit content	Pass	
2	Enter to-do task task: reading			
3	Click update button	Successful update the task and show the new task	Pass	

Post-conditions: System allow user to successfully update to-do task in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC003	Test Case Name: ToDoList
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test to-do list function by strikethrough to-do task	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click checkbox beside a task	Successful strikethrough the task	Pass	

Post-conditions: System allow user to successfully strikethrough to-do task

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC004	Test Case Name: ToDoList
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test to-do list function by delete to-do task	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click delete button	Successful delete the task	Pass	

Post-conditions: System allow user to successfully delete to-do task in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC005	Test Case Name: GoalGetter
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test goal getter function by adding goal task	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click goalGetter navigation link	Display goalGetter page	Pass	
2	Enter goal goal: SE midterm get A			
3	Enter reward reward: play 1hrs phone			
4	Enter milestones milestones: do revision			
5	Enter motivational quote motivational quote: work hard			
6	Click add goal button	Successful add a goal and show the goal	Pass	

Post-conditions: System allow user to successfully add goal in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC006	Test Case Name: GoalGetter
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test goal getter function by edit goal quote	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC005 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click edit button	Display column to edit content	Pass	
2	Enter goal quote goal quote: work more hard			
3	Click update button	Successful update the goal and show the new goal details	Pass	

Post-conditions: System allow user to successfully update goal quote in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC007	Test Case Name: GoalGetter
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test goal getter function by strikethrough goal	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC005 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click checkbox beside a goal	Successful strikethrough the goal	Pass	

Post-conditions: System allow user to successfully strikethrough goal

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC008	Test Case Name: GoalGetter
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test goal getter function by delete goal	

Pre-conditions: OneStopStudentSystem_PersonalModule_TC005 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click delete button	Successful delete the goal	Pass	

Post-conditions: System allow user to successfully delete goal details in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC009	Test Case Name: AIChatbot
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test AI chatbot function by input query	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click AI Chatbot navigation link	Display AI Chatbot page	Pass	
2	Enter query query: where is library			
3	Click sent button	Successful send query and show reply	Pass	

Post-conditions: System allow user to successfully ask query and get reply

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC010	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update name	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Enter name name: Hihi			
3	Click save change button	Successful update the name and show new name	Pass	

Post-conditions: System allow user to successfully update name in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC011	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update with valid date of birth	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Enter valid DOB name: 02/01/2003			
3	Click save change button	Successful update the DOB and show new DOB	Pass	

Post-conditions: System allow user to successfully update DOB in database

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC012	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update with invalid date of birth	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Enter invalid DOB name: 02/01/2024			
3	Click save change button	prompt error message “Only university students who are 18 years old or older are eligible to register an account.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC013	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update with valid mobile number	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Enter valid mobile number name: 019-9999999			
3	Click save change button	Successful update the mobile number and show new mobile number	Pass	

Post-conditions: System allow user to successfully update mobile number in database

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC014	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update with invalid mobile number	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Enter invalid mobile number name: 012-1233213			
3	Click save change button	prompt error message “Mobile number already exists. Please choose a different mobile number.”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC015	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update profile picture	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Click choose file button	Display file explorer	Pass	
3	Select picture			
3	Click save change button	Successful update the profile picture and show new profile picture	Pass	

Post-conditions: System allow user to successfully update profile picture in database

Test Case Template	
Test Case #: OneStopStudentSystem_PersonalModule_TC016	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update gender	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Select gender gender: Male			
3	Click save change button	Successful update the gender and show new gender	Pass	

Post-conditions: System allow user to successfully update gender in database

Test Case Template

Test Case #: OneStopStudentSystem_PersonalModule_TC016	Test Case Name: PersonalProfile
System: OneStopStudentSystem	Module: Personal Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test personal profile function by update location	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click personal profile navigation link	Display personal profile page with all the personal details	Pass	
2	Select location location: Putrajaya			
3	Click save change button	Successful update the location and show new location	Pass	

Post-conditions: System allow user to successfully update location in database

Test Case Template

Test Case #: OneStopStudentSystem_AdminModule_TC001	Test Case Name: RoleManagement
System: OneStopStudentSystem	Module: Admin Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test role management function by view admin role	

Pre-conditions: OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click role management navigation link	Display role management page	Pass	
2	Select role role: Admin	Successful display all the admin details	Pass	

Post-conditions: System allow user to successfully view all the admin role details

Test Case Template

Test Case #: OneStopStudentSystem_AdminModule_TC002	Test Case Name: RoleManagement
System: OneStopStudentSystem	Module: Admin Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test role management function by view student role	

Pre-conditions: OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click role management navigation link	Display role management page	Pass	
2	Select role role: Student	Successful display all the student details	Pass	

Post-conditions: System allow user to successfully view all the student role details

Test Case Template	
Test Case #: OneStopStudentSystem_AdminModule_TC003	Test Case Name: RoleManagement
System: OneStopStudentSystem	Module: Admin Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test role management function by view locked account for student role.	

Pre-conditions: OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click role management navigation link	Display role management page	Pass	
2	Select role role: Student	Successful display all the student details	Pass	
3	Select checkbox for locked account users	Successful display all the locked account of student details	Pass	

Post-conditions: System allow user to successfully view all the locked account for student role details.

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC001	Test Case Name: BMICalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test BMI Calculator function with valid age, gender, valid height, valid weight	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click BMI calculator navigation link	Display BMI calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter valid height Height: 160			
5	Enter valid weight Weight: 40			
6	Click calculate BMI button	Successful calculate and show the result	Pass	

Post-conditions: System successful calculate and display the result

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC002	Test Case Name: BMICalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test BMI Calculator function with invalid age, gender, valid height, valid weight	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click BMI calculator navigation link	Display BMI calculator page	Pass	
2	Enter invalid age Age: 2			
3	Select gender Gender: female			
4	Enter valid height Height: 160			
5	Enter valid weight Weight: 40			
6	Click calculate BMI button	Prompt error message “Please enter valid university student age (more than 18 years old)”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC003	Test Case Name: BMICalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test BMI Calculator function with valid age, gender, invalid height, valid weight	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click BMI calculator navigation link	Display BMI calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter invalid height Height: -2			
5	Enter valid weight Weight: 40			
6	Click calculate BMI button	Prompt error message “Please enter valid height”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC004	Test Case Name: BMICalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test BMI Calculator function with valid age, gender, valid height, invalid weight	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click BMI calculator navigation link	Display BMI calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter invalid height Height: 150			
5	Enter valid weight Weight: -2			
6	Click calculate BMI button	Prompt error message “Please enter valid weight”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC005	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with valid age, gender, valid height, valid weight, valid target weight, valid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter valid height Height: 160			
5	Enter valid weight Weight: 40			
6	Enter valid target weight Target weight: 50			

7	Enter valid weeks to reach target weight Weeks: 100			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Successful calculate and show the result	Pass	

Post-conditions: System successful calculate and display the result

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC006	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with invalid age, gender, valid height, valid weight, valid target weight, valid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter invalid age Age: 10			
3	Select gender Gender: female			
4	Enter valid height Height: 160			
5	Enter valid weight Weight: 40			
6	Enter valid target weight Target weight: 50			

7	Enter valid weeks to reach target weight Weeks: 100			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Prompt error message “Please enter valid university student age (more than 18 years old)”	Pass	

Post-conditions: System remain to display error message and remain in the same page
--

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC007	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with valid age, gender, invalid height, valid weight, valid target weight, valid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter invalid height Height: 0			
5	Enter valid weight Weight: 40			
6	Enter valid target weight Target weight: 50			

7	Enter valid weeks to reach target weight Weeks: 100			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Prompt error message “Please enter valid height”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC008	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with valid age, gender, valid height, invalid weight, valid target weight, valid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter valid height Height: 150			
5	Enter invalid weight Weight: 0			
6	Enter valid target weight Target weight: 50			

7	Enter valid weeks to reach target weight Weeks: 100			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Prompt error message “Please enter valid weight”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC009	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with valid age, gender, valid height, valid weight, invalid target weight, valid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter valid height Height: 150			
5	Enter valid weight Weight: 70			
6	Enter invalid target weight Target weight: 0			

7	Enter valid weeks to reach target weight Weeks: 100			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Prompt error message “Please enter valid target weight”	Pass	

Post-conditions: System remain to display error message and remain in the same page

Test Case Template	
Test Case #: OneStopStudentSystem_HealthModule_TC010	Test Case Name: CalorieCalculator
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calorie calculator function with valid age, gender, valid height, valid weight, valid target weight, invalid weeks to reach target weight, activity level	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calorie calculator navigation link	Display calorie calculator page	Pass	
2	Enter valid age Age: 20			
3	Select gender Gender: female			
4	Enter valid height Height: 150			
5	Enter valid weight Weight: 70			
6	Enter valid target weight Target weight: 60			

7	Enter invalid weeks to reach target weight Weeks: -3			
8	Select activity level Activity level: Sedentary			
9	Click calculate button	Prompt error message “Please enter valid weeks to reach target”	Pass	

Post-conditions: System remain to display error message and remain in the same page
--

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC011	Test Case Name: WorkoutSchedule
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test workout schedule function with input day, time and exercise name in blank timeslot	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click exercise workout schedule navigation link	Display exercise workout schedule page	Pass	
2	Select day Day: Monday			
3	Select time Time: 12:00am-1:00am			
4	Enter exercise name Name: run			
5	Click add exercise button	Successful add exercise and show the exercise in respective timeslot	Pass	

Post-conditions: System successful add the exercise name with the timeslot in database

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC012	Test Case Name: WorkoutSchedule
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test workout schedule function with input day, time and exercise name in existing timeslot	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click exercise workout schedule navigation link	Display exercise workout schedule page	Pass	
2	Select day Day: Monday			
3	Select time Time: 12:00am-1:00am			
4	Enter exercise name Name: run			
5	Click add exercise button	Prompt alert “An exercise already exists for this day and time. Do you want to replace it?”	Pass	
6	Click ok button	Successful update exercise and show the exercise in respective timeslot	Pass	

Post-conditions: System successful update the exercise name with the timeslot in database

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC013	Test Case Name: WorkoutSchedule
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test workout schedule function with click exercise name in timeslot	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click exercise workout schedule navigation link	Display exercise workout schedule page	Pass	
2	Click existing exercise name in timeslot	Successful display YouTube link	Pass	

Post-conditions: System successful display the YouTube link related to the particular exercise

Test Case Template

Test Case #: OneStopStudentSystem_HealthModule_TC014	Test Case Name: WorkoutSchedule
System: OneStopStudentSystem	Module: Health Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test workout schedule function with enter exercise concern	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click exercise workout schedule navigation link	Display exercise workout schedule page	Pass	
2	Enter exercise concern Concern: burn fat			
3	Click existing exercise name in timeslot	Successful display YouTube link	Pass	

Post-conditions: System successful display the YouTube link related to the particular exercise

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC001	Test Case Name: NoteTaking
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test note taking function by enter title, description, image, colour, course, valid week	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click note taking navigation link	Display note taking page	Pass	
2	Enter title Title: waterfall model			
3	Enter description Description: no iterative			
4	Upload 2 images			
5	Select colour in colour picker			
6	Enter course Course: SE			
7	Enter valid week Week: 1			

8	Click add note button	Successful add note and show the note details	Pass	
---	-----------------------	---	------	--

Post-conditions: System allow user to successfully add note in database

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC002	Test Case Name: NoteTaking
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test note taking function by enter title, description, image, colour, course, invalid week	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click note taking navigation link	Display note taking page	Pass	
2	Enter title Title: waterfall model			
3	Enter description Description: no iterative			
4	Upload images			
5	Select colour in colour picker			
6	Enter course Course: SE			
7	Enter invalid week Week: 20			

8	Click add note button	Prompt error message “value must be less than or equal to 14”	Pass	
---	-----------------------	---	------	--

Post-conditions: System remain to display error message and remain in the same page

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC003	Test Case Name: NoteTaking
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test note taking function of edit note by adding note image that are not duplicated	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass and OneStopStudentSystem_StudyModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click note taking navigation link	Display note taking page	Pass	
2	Click edit note button for a particular note	Navigate to edit note page	Pass	
3	Click choose file button	Pop up file explorer	Pass	
4	Select images that are not same with existing image			
5	Click update button	Navigate to main note taking page and show new note details	Pass	

Post-conditions: System allow user to successfully update note in database

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC004	Test Case Name: NoteTaking
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test note taking function of edit note by adding note image that are duplicated	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass and OneStopStudentSystem_StudyModule_TC001 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click note taking navigation link	Display note taking page	Pass	
2	Click edit note button for a particular note	Navigate to edit note page	Pass	
3	Click choose file button	Pop up file explorer	Pass	
4	Select images and at least one image are same with existing image			
5	Click update button	Prompt error message “Some images are duplicates and will not be added again.” and show only image that are not duplicated in image container	Pass	

Post-conditions: System remain in the same page but add the image that are not duplicated in the temporary container

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC005	Test Case Name: GradeCalculator
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test grade calculator function with valid current CGPA, valid target CGPA, valid total credit hours, valid 3 subject credit	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click Before July 2023 intake navigation link	Display grade prediction before July 2023 intake page	Pass	
2	Enter valid current CGPA Current CGPA: 3.9882			
3	Enter valid target CGPA Target CGPA: 3.9897			
4	Enter valid total credit hours Total credit hours: 66			
5	Click add subject button	Show extra 1 subject column	Pass	
6	Click add subject button	Show extra 1 subject column	Pass	
7	Enter valid subject credit for			

	subject 1 Credit: 3			
8	Enter valid subject credit for subject 2 Credit: 4			
9	Enter valid subject credit for subject 3 Credit: 3			
10	Click calculate button	Successful calculate and display result	Pass	

Post-conditions: System successfully calculate each subject grade

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC006	Test Case Name: GradeCalculator
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test grade calculator function with invalid current CGPA, valid target CGPA, valid total credit hours, valid 3 subject credit	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click Before July 2023 intake navigation link	Display grade prediction before July 2023 intake page	Pass	
2	Enter invalid current CGPA Current CGPA: 6			
3	Enter valid target CGPA Target CGPA: 3.9897			
4	Enter valid total credit hours Total credit hours: 66			
5	Click add subject button	Show extra 1 subject column	Pass	
6	Click add subject button	Show extra 1 subject column	Pass	
7	Enter valid subject credit for			

	subject 1 Credit: 3			
8	Enter valid subject credit for subject 2 Credit: 4			
9	Enter valid subject credit for subject 3 Credit: 3			
10	Click calculate button	Prompt error message “Current CGPA should be between 0.0 and 4.0”	Pass	

Post-conditions: System remain in the same page
--

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC007	Test Case Name: GradeCalculator
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test grade calculator function with valid current CGPA, invalid target CGPA, valid total credit hours, valid 3 subject credit	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click Before July 2023 intake navigation link	Display grade prediction before July 2023 intake page	Pass	
2	Enter valid current CGPA Current CGPA: 3.9897			
3	Enter invalid target CGPA Target CGPA: 6			
4	Enter valid total credit hours Total credit hours: 66			
5	Click add subject button	Show extra 1 subject column	Pass	

6	Click add subject button	Show extra 1 subject column	Pass	
7	Enter valid subject credit for subject 1 Credit: 3			
8	Enter valid subject credit for subject 2 Credit: 4			
9	Enter valid subject credit for subject 3 Credit: 3			
10	Click calculate button	Prompt error message “Target CGPA should be between 0.0 and 4.0”	Pass	

Post-conditions: System remain in the same page
--

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC008	Test Case Name: GradeCalculator
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test grade calculator function with valid current CGPA, valid target CGPA, invalid total credit hours, valid 3 subject credit	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click Before July 2023 intake navigation link	Display grade prediction before July 2023 intake page	Pass	
2	Enter valid current CGPA Current CGPA: 3.9897			
3	Enter valid target CGPA Target CGPA: 3.9892			
4	Enter invalid total credit hours Total credit hours: a			
5	Click add subject button	Show extra 1 subject column	Pass	

6	Click add subject button	Show extra 1 subject column	Pass	
7	Enter valid subject credit for subject 1 Credit: 3			
8	Enter valid subject credit for subject 2 Credit: 4			
9	Enter valid subject credit for subject 3 Credit: 3			
10	Click calculate button	Prompt error message “Please enter a valid positive integer value for total credit field”	Pass	

Post-conditions: System remain in the same page
--

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC009	Test Case Name: GradeCalculator
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test grade calculator function with valid current CGPA, valid target CGPA, valid total credit hours, invalid 3 subject credit	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click Before July 2023 intake navigation link	Display grade prediction before July 2023 intake page	Pass	
2	Enter valid current CGPA Current CGPA: 3.9897			
3	Enter valid target CGPA Target CGPA: 3.9892			
4	Enter valid total credit hours Total credit hours: 66			
5	Click add subject button	Show extra 1 subject column	Pass	

6	Click add subject button	Show extra 1 subject column	Pass	
7	Enter invalid subject credit for subject 1 Credit: a			
8	Enter valid subject credit for subject 2 Credit: 4			
9	Enter valid subject credit for subject 3 Credit: 3			
10	Click calculate button	Prompt error message “Please enter a valid positive numerical value for credit of subject 1”	Pass	

Post-conditions: System remain in the same page
--

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC010	Test Case Name: VideoTeachingPronunciation
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test video teaching pronunciation function by inputting a word	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click video teaching pronunciation navigation link	Display video teaching pronunciation page	Pass	
2	Enter a word Word: apple			
3	Click show video button	Successfully display video related to user input	Pass	

Post-conditions: System successfully search YouTube and display related pronunciation video

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC011	Test Case Name: CalanderReminder
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calendar reminder function with event name, event description, valid date time, reminder minute before event start	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calendar reminder navigation link	Display calendar reminder page	Pass	
2	Enter event name Name: read book			
3	Enter event description Description: SE book			
4	Enter valid date time which is greater than now Date time: 08-15-2024 00:05:26			
5	Enter reminder minute before event start Minute : 1			

6	Click add event button	Successfully add and display event, set event to Google Calendar and sent reminder email to Gmail	Pass	
---	------------------------	---	------	--

Post-conditions: System allow user to successfully create event and set reminder in Google Calander and sent reminder email to Gmail

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC012	Test Case Name: CalanderReminder
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calendar reminder function by updating event name and event description	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass and OneStopStudentSystem_StudyModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calendar reminder navigation link	Display calendar reminder page	Pass	
2	Click edit of one existing event			
3	Enter event name Name: read books			
4	Enter event description Description: SE books			
5	Click update	Successfully update and display event, update event in Google Calendar and sent update reminder email to Gmail	Pass	

Post-conditions: System allow user to successfully update event in database and update reminder in Google Calander and sent update reminder email to Gmail

Test Case Template	
Test Case #: OneStopStudentSystem_StudyModule_TC013	Test Case Name: CalanderReminder
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test calendar reminder function by deleting existing event	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass and OneStopStudentSystem_StudyModule_TC011 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click calendar reminder navigation link	Display calendar reminder page	Pass	
2	Click delete of one existing event	Successfully delete event in database, delete event in Google Calendar, sent deleted email reminder to Gmail	Pass	

Post-conditions: System allow user to delete event in database, delete reminder in Google Calander and sent deleted reminder email to Gmail

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC014	Test Case Name: UnitConverter
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test unit converter function by inputting number in number related converter	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click unit converter navigation link	Display unit converter page	Pass	
2	Select one category about number conversion converter	Display related conversion section	Pass	
3	Input number			
4	Optional to choose unit, notation, decimal places			
5	Click convert button	Display result	Pass	

Post-conditions: System allow user to calculate conversion about number and display result

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC015	Test Case Name: UnitConverter
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test unit converter function by inputting text in case converter	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click unit converter navigation link	Display unit converter page	Pass	
2	Select category case converter	Display case conversion section	Pass	
3	Input text			
4	Optional to choose conversion type			
5	Click convert button	Display result	Pass	

Post-conditions: System allow user to convert text into different conversion type and display result

Test Case Template

Test Case #: OneStopStudentSystem_StudyModule_TC016	Test Case Name: UnitConverter
System: OneStopStudentSystem	Module: Study Module
Design By: Seow Hui Chee	Design Date: 10/8/2024
Executed By: Seow Hui Chee	Execution Date: 14/8/2024
Short Description: to test unit converter function by inputting number in calculator	

Pre-conditions: OneStopStudentSystem_AccountModule_TC008 must pass or OneStopStudentSystem_AccountModule_TC011 must pass or OneStopStudentSystem_AccountModule_TC014 must pass

Step	Action	Expected System Response	Pass/Fail	Comments
1	Click unit converter navigation link	Display unit converter page	Pass	
2	Select category calculator	Display calculator section	Pass	
3	Input number 1 Number 1: 3			
4	Input number 2 Number 2: 4			
5	Optional to choose operator			
6	Click convert button	Display result	Pass	

Post-conditions: System allow user to calculate simple calculation and display result

5.3 Chapter Summary and Evaluation

In this chapter, it has provided a detailed overview of the coding and implementation of each function from the developed system. Code snippets are captured using codesnap that provided in the extension of Visual Studio Code. For the testing part, it focuses on the test cases for each function. Each account, study, health and personal module contains several functions that were tested with different types of cases and validations. The expected system responses are all pass after testing.

Chapter 6

Discussions and Conclusion

6 Discussions and Conclusion

This chapter included achievements, contributions, limitations and future improvement and the issues and solutions identified. Readers will gain a comprehensive understanding of all the topics mentioned above in this chapter.

6.1 Summary

This project addresses several challenges faced by university students such as managing multiple web pages and platforms, lack of structured physical activity, difficulty in prioritizing academic tasks, lack of motivation, issues with pronunciation skills, communication gap and so on. The proposed solution is a one-stop student system that integrates various functionalities like GPA grade prediction, BMI and calorie calculators, to-do lists, event reminders, exercise workout schedules, AI chatbot and so on. This system aims to reduce browser clutter, enhance academic performance, improve personal health and streamline access to essential functions, thus reducing stress and increasing efficiency.

Feature-Driven Development (FDD) methodology is chosen for its iterative and incremental approach which enables continuous feedback and adaptation. This methodology allows the project to evolve based on user input, ensuring that functionalities are refined to meet user needs. With its emphasis on quality assurance and regular testing, FDD helps maintain high standards while minimising risks. Moreover, its clear development roadmap enhances predictability, making it easier to manage timelines and avoid delays. Overall, FDD supports efficient, flexible development and focus on delivering high-quality software in manageable increments.

The project is developed using Visual Studio and using ASP.NET web application framework. Several technologies and APIs are utilized to enhance the functionality and user experience of the application. YouTube API is used to searching video for pronunciation function and exercise concern. Google Calendar API is used for adding, editing and deleting reminder in students' Google Calendar. Gmail SMTP is used to send email reminders to students' Gmail. AJAX is used for creating a more dynamic and responsive user interface by allowing asynchronous data retrieval, updates without reloading the entire page and sent and receive data from database. OWIN middleware is used for managing Google authentication and authorization.

During the implementation and testing stage, this system has undergone thorough system testing to ensure that all the results meet the expected outcomes of each of the test case.

6.2 Achievements

The first objective, to allow predict next semester CGPA grade to motivate to achieve better results has been achieved. This can be proved by the CGPA grade calculator function has successfully allows students to predict their grade by inputting current CGPA, target CGPA, total credits hours earned and next semester subject credit. The result can be an indicator for students' coming academic performance and motivate students to work harder in order to achieve good results.

The second objective, to identify who are at health risk at earlier stage has achieved through BMI calculator function which classify students into categories such as obese or overweight. To further solve this health problem, recommendations are provided. Calorie calculator will provide nutrition intake and food portion recommendation of protein, fat and carbohydrate for students as well.

Third objective, to improve pronunciation accuracy and skills is achieved though the video pronunciation function. When students have any pronunciation confused words, students can input the word and the system will search the YouTube and show the relevant pronunciation video.

Forth objective, to simplify the access in functions, reduce navigation and learning time has achieved through this one-stop student system because it has simplified the process from searching functions on the browser one by one by navigation.

This project was completed within the given timeline which show the effective planning and execution process. All the required functionalities were developed and improved based on feedback as well as tested thoroughly. This reflects the successful completion of the project.

The integration of different tools into a single system is a significant strength. It reduces the complexity for users who need multiple browser tabs or applications for different functions. The system's design focuses on simplicity and ease of use, students can easily access the functions with low learning time. However, the system's reliance on third-party APIs like Google and YouTube poses a risk if any of these services change their terms of use, limited access or discontinue support.

6.3 Contributions

This proposed One-Stop Student System represents a creative and innovative solution tailored to address several challenges faced by university students. This system is necessary because it can help students in the study journey using those academic, health and personal support features. This system stands out from market because it integrates a wide range of functionalities such as academic tools, health calculators, exercise workout schedule, event reminder, AI chatbot and so on into a user-friendly platform.

The system eliminates the need for students to navigate between different websites and applications for certain functions. This simplifies the process is innovative as it saves time and reduces stress from opening many browser tabs which is crucial for students managing hectic schedules. The AI chatbot that is capable of answering TAR UMT related queries to new students is a new feature. It offers real-time assistance support whenever students have doubts about the university.

By including the BMI and calorie calculators along with exercise workout schedules, the system encourages a balanced lifestyle which promotes physical well-being to aid in academic success. This focus is rarely seen in traditional student systems that only focus on academic part. Besides, the reminders feature ensures that students remain aware of the events, assignment deadlines and exercise schedules. This feature is important in fostering time management skills and reducing the likelihood of missed assignments or event activities.

The marketability of the One-Stop Student System has huge potential. It provides the demand for a comprehensive student tool including study, health and personal categories in market. Universities could adopt this system to enhance students' support services with minor modification of AI chatbot to cater to answer various university questions. Moreover, the application could be marketed directly to students who seek a one stop solution to manage the academic and personal lives efficiently. In this digital world, this system becomes even more valuable to provide support in students' study journey.

6.4 Limitations and Future Improvements

The first limitation is dependence on internet connectivity. Since the system relies on online services and APIs like YouTube and Google Calendar, students without stable internet access may face challenges in fully utilizing the system. Besides, the chatbot is developed to handle common queries regarding TAR UMT, if there is any outsider students use this system will not get the desire responses due to scope of the chatbot.

To enhance the one-stop student system for university students, several future improvements can be considered. Adding a web game module such as math quizzes, can help students relax while also training the logical thinking skills. This provides a fun and educational break from students' studies. Besides, implementing a chatroom allows students to interact, seek academic help from seniors especially in the situation that students don't have a way to know about senior in respective course. The chatroom needs to ban sensitive words is to ensure a peace and safe environment. Moreover, integrating VR technology enables students to explore the campus environment virtually, which is particularly beneficial for new students to familiarize themselves with the campus layout and facilities before physically arriving. An anonymous feedback feature allows students to voice their concerns or dissatisfaction, give comments or rate relevant feedback posts. This can help the university top management to understand and address student issues better due to its anonymous feature. Furthermore, a lost and found feature helps students recover lost items by posting post based on different item category and this can reduce the stress and inconvenience of students who lost item. Additionally, including a time and money tracker with expense tracking and budgeting features helps students manage the finances effectively. This feature can record future bills and track spending across various categories such as food, shopping and daily supplies. By providing a clear overview of their financial situation, it helps students make informed decisions, maintain emergency funds and reduce reliance on family financial support to be independent adult. This not only reduce financial burdens on students' families but also equips students with essential financial management skills, preparing for independent living and future societal responsibilities. Furthermore, a sleep tracker can help students monitor their sleep patterns and provide tips for improving sleep quality because good sleep is crucial for academic performance and overall health. All these future improvements are outside the project scope which can provide comprehensive support to university students in managing their academic, health and personal lives more effectively.

6.5 Issues and Solutions

One of the challenges faces is integrating multiple APIs which are YouTube and Google Calendar API with the ASP.NET framework. This required understanding each API's mechanism and data handling formats. To tackle this, I have studied about how to integrate the API in the system and implemented error handling to manage any integration failures.

Moreover, I have face challenge in using Google account to login to this system. I found the solution, which is navigate to the Google Developer Console, create a project, create credentials (OAuth 2.0 Client IDs), added the URL of the website in Authorised redirect URIs sections, configure OAuth Consent Screen which tells students the application requesting access to their data, what kind of data are asked for and the terms that apply and fill up the necessary fields like name, logo, support email and so on. Google Calendar API need to be enabled in this current project as well. Client ID and Client secret is needed to give this system the ability to request consent of a student to access their data.

Besides, I have written the chatbot python code in Jupyter Notebook but face difficulties running it in ASP.NET webform. After finding some online solution, I found that the code needs to be downloaded into python file (.py file) first and then need to create a batch file (.bat file) to run the python scripts from ASP.NET webform. While in the webform, it needs to use the System.Diagnostics namespace to run the batch file.

I have developed my time management skills throughout this project. Balancing project development with academic responsibilities has been a significant challenge. However, weekly meetings have been significantly helpful in addressing this issue because weekly meeting provide motivation to complete the project on time and allow for gradual improvement based on feedback.

References

1. Awotunde, J. B., Farhaoui, Y., Adigun, I. P., Imoize, A. L., & Awotunde, J. B. (2024). Prediction of student's academic performance using learning analytics. In Lecture notes in networks and systems (pp. 314–325). https://doi.org/10.1007/978-3-031-48465-0_41
2. Woods, N., & Siponen, M. T. (2024). How memory anxiety can influence password security behavior. Computers & Security, 137, 103589. <https://doi.org/10.1016/j.cose.2023.103589>
3. Seng, A. (2024, March 12). How to Lose Weight Sustainably (5 kg in 10 Weeks): Weight Loss Planner | NOVI Health. NOVI Health. <https://novi-health.com/library/free-weight-loss-planner-how-to-lose-05-kg-per-week-sustainably>
4. Calonia, J. T., Pagente, D. P., Jane, D., Capio, R. T., Tembrevilla, A. P., Guzman, C. A., & Jay, A. (2023). Time Management and Academic Achievement: Examining the Roles of Prioritization, Procrastination and Socialization. <https://doi.org/10.5281/zenodo.8115965>
5. Dinh, H. M., & Tran, T. K. (2023). EduChat: an AI-Based chatbot for University-Related information using a hybrid approach. Applied Sciences, 13(22), 12446. <https://doi.org/10.3390/app132212446>
6. Maksymov, A., & Tryus, Y. V. (2023). Combined Method of Solving Time Management Tasks and Its Implementation in the Decision Support System. Lecture Notes on Data Engineering and Communications Technologies, 131–146. https://doi.org/10.1007/978-3-031-35467-0_9
7. Soares, A. B., Ribeiro, R., Alves, P., De Melo Jardim, M. E., & De Medeiros, C. a. C. (2023). Time management: what do University students think about it? Revista De Estudios E InvestigacióN En Psicología Y EducacióN, 10(1), 1–14. <https://doi.org/10.17979/reipe.2023.10.1.9468>
8. Ma, R., Lassila, H., Leysan Nurgalieva, & Janne Lindqvist. (2023). When Browsing Gets Cluttered: Exploring and Modeling Interactions of Browsing Clutter, Browsing Habits, and Coping. <https://doi.org/10.1145/3544548.3580690>
9. TrustRadius. (2023, September 11). What's Todoist? [Video]. TrustRadius. <https://www.trustradius.com/products/todoist/reviews?qs=pros-and-cons#product-details>
10. Louise, J., Olleras, C., Dagwayan, M., Aliann, M., Dejacto, J., Rose, L., Mangay, M., Ebarsabal, Dinah, J., Diaz, C., Puti-An, A., Lendio, J., Nadera, J., Taneo, C., Cabello, A., & Minyamin. (2022). The Life of the Laters: Students Procrastination in Accomplishing Academic Deadlines in Online Learning. <https://doi.org/10.5281/zenodo.6791776>
11. Khanna, D., Peltzer, C., Kahar, P., & Parmar, M. (2022). Body Mass Index (BMI): a screening tool analysis. Cureus. <https://doi.org/10.7759/cureus.22119>
12. Osilla, E. V., Safadi, A. O., & Sharma, S. (2022, September 12). Calories. StatPearls - NCBI Bookshelf. <https://www.ncbi.nlm.nih.gov/books/NBK499909/>

13. Kennedy, D. R., & Porter, A. L. (2022). The illusion of urgency. *The American Journal of Pharmaceutical Education*, 86(7), 8914. <https://doi.org/10.5688/ajpe8914>
 14. K.R.C Bandara, D.D.T.D Dureksha, S.C Pinidiya, R.M.G.H Amarasinghe, Thelijagoda, S., & Kishara, J. (2022). Healthy Heart – Heart Risk Prediction System on Personalized Guidance for Heart Patients. 2022 IEEE 7th International Conference for Convergence in Technology (I2CT). <https://doi.org/10.1109/i2ct54291.2022.9824170>
 15. Kamal, M., & Baqui, G. A. (2022). A STUDY ON STUDENT MOTIVATION AT UNIVERSITY LEVEL. Khulna University Studies, 203–212. <https://doi.org/10.53808/kus.2013.11and12.1208-s>
 16. Katsoulis, M., Lai, A. G., Diaz-Ordaz, K., Gomes, M., Pasea, L., Banerjee, A., Denaxas, S., Tsilidis, K., Λάγιου, Π., Misirli, G., Bhaskaran, K., Wannamethee, G., Dobson, R., Batterham, R. L., Kipourou, D. K., Lumbers, R. T., Li, W., Wareham, N., Langenberg, C., & Hemingway, H. (2021). Identifying adults at high-risk for change in weight and BMI in England: a longitudinal, large-scale, population-based cohort study using electronic health records. *The Lancet Diabetes & Endocrinology*, 9(10), 681–694. [https://doi.org/10.1016/s2213-8587\(21\)00207-2](https://doi.org/10.1016/s2213-8587(21)00207-2)
 17. Chang, J. C., Hahn, N., Kim, Y., Coupland, J., Breneisen, B., Kim, H. S., Hwong, J., & Kittur, A. (2021). When the Tab Comes Due: Challenges in the Cost Structure of Browser Tab Usage. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3411764.3445585>
 18. Mehmood, I., Zahoor, A., & Shah, A. H. (2021). Operative use of sentence connectors in English writing skills: An experimental study based on the scaffolding technique. *Asian ESP Journal*, 17(2), 231-248.
 19. Golubnitschaja, O., Liskova, A., Koklesova, L., Samec, M., Biringer, K., Büsselberg, D., Podbielska, H., Kunin, A. A., Evseyeva, M. E., Shapira, N., Paul, F., Erb, C., Dietrich, D. E., Felbel, D., Karabatsiakis, A., Bubnov, R., Polivka, J., Polivka, J., Birkenbihl, C., & Fröhlich, H. (2021). Caution, “normal” BMI: health risks associated with potentially masked individual underweight—EPMA Position Paper 2021. *EPMA Journal*, 12(3). <https://doi.org/10.1007/s13167-021-00251-4>
 20. Panchali (2021). Eisenhower Box for Prioritising Waiting List of Orthodontic Patients. [https://www.researchgate.net/publication/352005944 Eisenhower Box for Prioritising Waiting List of Orthodontic Patients](https://www.researchgate.net/publication/352005944_Eisenhower_Box_for_Prioritising_Waiting_List_of_Orthodontic_Patients)
 21. K., H. K., Palakurthi, A. K., Putnala, V., & Kumar K., A. (2020). Smart College Chatbot using ML and Python. 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). <https://doi.org/10.1109/icscan49426.2020.9262426>
 22. Sokolova, K., & Pérez, C. (2021). You follow fitness influencers on YouTube. But do you actually exercise? How parasocial relationships, and watching fitness influencers, relate to
-

- intentions to exercise. *Journal of Retailing and Consumer Services*, 58, 102276. <https://doi.org/10.1016/j.jretconser.2020.102276>
23. Castro, Ó., Bennie, J. A., Vergeer, I., Bosselut, G., & Biddle, S. J. H. (2020). How sedentary are university students? A Systematic Review and Meta-Analysis. *Prevention Science*, 21(3), 332–343. <https://doi.org/10.1007/s11121-020-01093-8>
24. Toçi, A. (2020). Problems with Pronunciation Among Students of English Language and Literature-Seeu. *SEEU Review*, 15(2), 113–125. <https://doi.org/10.2478/seeur-2020-0020>
25. Wang, M. (2020). Research on physical exercise of college students based on the data analysis of Scale. 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). <https://doi.org/10.1109/aemcse50948.2020.00117>
26. Rachmawati, R., & Cahyani, F. (2020). The use of YouTube videos in improving Non-English Department students' pronunciation skills. *Alsuna*, 3(2), 83–95. <https://doi.org/10.31538/alsuna.v3i2.916>
27. Clark, D., Gill, D., Prowse, V. L., & Rush, M. (2020). Using goals to Motivate college students: Theory and evidence from field experiments. *The Review of Economics and Statistics*, 102(4), 648–663. https://doi.org/10.1162/rest_a_00864
28. Antaris, I., & Omolu, F. A. (2020). FACTORS AFFECTING PRONUNCIATION DIFFICULTIES OF 8th GRADE STUDENTS OF MTSN PALU BARAT. *Journal of Foreign Language and Educational Research*, 2(2), 10-19. <https://doi.org/10.31934/jofler.v2i2.1349>
29. Kaur, H., Singh, T., Arya, Y. K., & Mittal, S. (2020). Physical Fitness and Exercise during the COVID-19 Pandemic: A Qualitative Enquiry. *Frontiers in Psychology*, 11. <https://doi.org/10.3389/fpsyg.2020.590172>
30. Mauliya, I., Relianisa, R. Z., & Rokhyati, U. (2020). Lack of motivation factors creating poor academic performance in the context of graduate English Department students. *Linguistics*, 6(2), 73. <https://doi.org/10.29300/ling.v6i2.3604>
31. Adams, R. V., & Blair, E. (2019). Impact of Time Management Behaviors on Undergraduate Engineering Students' Performance. *SAGE Open*, 9(1). <https://doi.org/10.1177/2158244018824506>
32. Misra, A., & Dhurandhar, N. V. (2019). Current formula for calculating body mass index is applicable to Asian populations. *Nutrition & Diabetes*, 9(1). <https://doi.org/10.1038/s41387-018-0070-9>
33. 5. Prevention or Delay of Type 2 Diabetes: Standards of Medical Care in Diabetes—2018. (2018). *Diabetes Care*, 41(Supplement 1), S51–S54. <https://doi.org/10.2337/dc18-s005>
34. OuYang, G. (2018, May 1). Design and Implementation of Body Mass Index Calculator Based on Android Technology. *IEEE Xplore*. <https://doi.org/10.1109/IMCEC.2018.8469304>

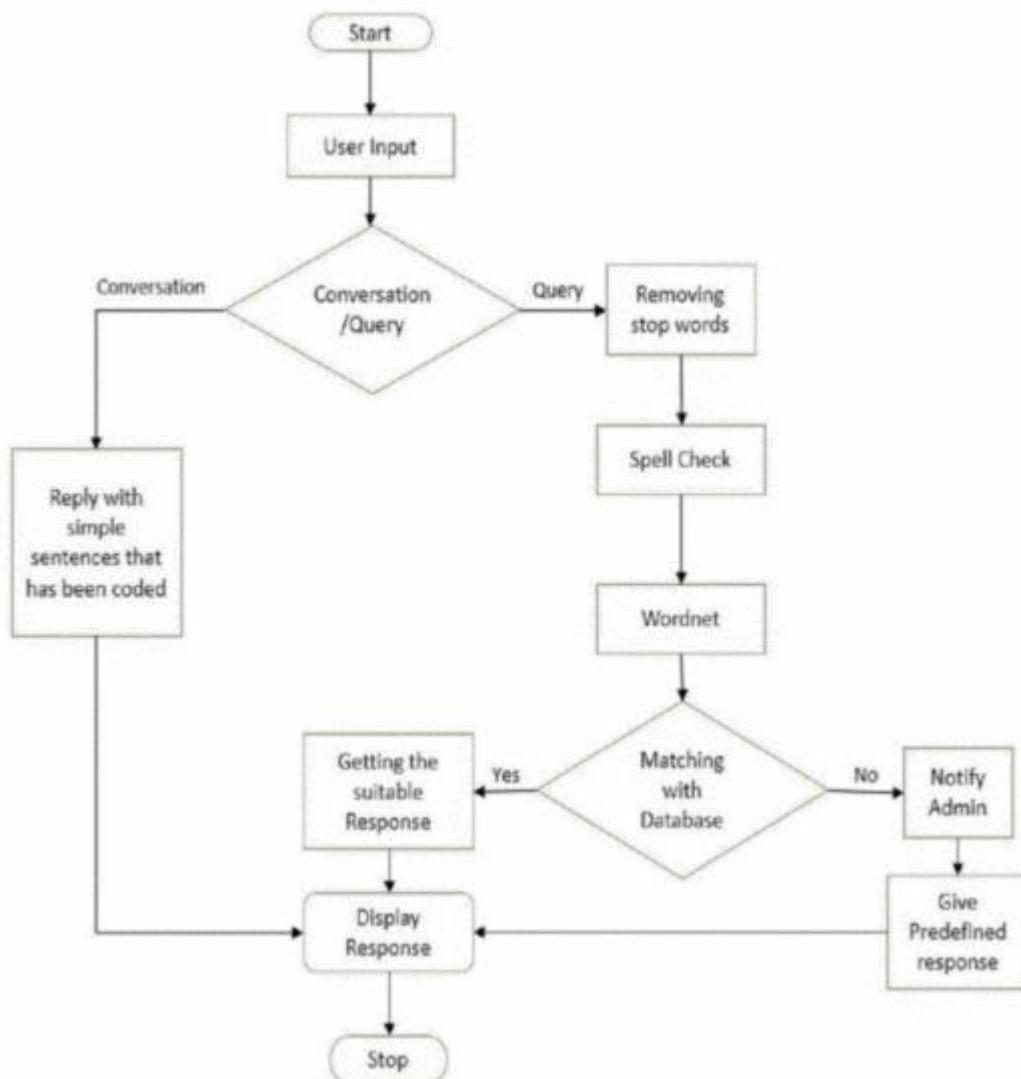
35. Nicklin, J. M., Meachon, E. J., & McNall, L. A. (2018). Balancing Work, School, and Personal Life among Graduate Students: a Positive Psychology Approach. *Applied Research in Quality of Life*, 14(5), 1265–1286. <https://doi.org/10.1007/s11482-018-9650-z>
36. Azubuike Ezenwoke, Ofure Aigbefo, Sarkis, A., & Odukoya, J. (2018). A PREFERENCE-BASED GRADE RECOMMENDER TOWARDS THE ATTAINMENT OF A TARGET GRADE POINT AVERAGE (GPA).
37. Simanullang, M. (2018, November 1). The effect of applying video on the Students' English Pronunciation Accuracy at the Fifth Semester Students at the English Study Program of the Teacher's Training and Education Faculty the University of Sisingamangaraja XII Tapanuli in Academic Year 2018/2019. Nelite. <https://www.nelite.com/publications/268296/the-effect-of-applying-video-on-the-students-english-pronunciation-accuracy-at-t>
38. Clark, D. A., Gill, D., Prowse, V. L., & Rush, M. G. (2017). Using Goals to Motivate College Students: Theory and Evidence from Field Experiments. <https://doi.org/10.3386/w23638>
39. Moulin, M. S., & Irwin, J. D. (2017). An assessment of sedentary time among undergraduate students at a Canadian university. *International Journal of Exercise Science*, 10, 1116–1129.
40. Zollanvari, A., Kızılırmak, R. Ç., Kho, Y. H., & Hernández-Torrano, D. (2017). Predicting students' GPA and developing intervention strategies based on Self-Regulatory learning behaviors. *IEEE Access*, 5, 23792–23802. <https://doi.org/10.1109/access.2017.2740980>
41. Ismail, N. Z., Aziz, A., Shafiq, M. E., Razak, M., Ramli, N., Aziz, A., & Norafizal, M. (2016). The characteristic of GPA and CGPA forecast calculator (GCFC). *KONAKA Konferensi Akademik 2016*, 21-25.
42. Jslim. (2011). GitHub - jslim89/tarc-cgpa-calc: A CGPA calculator for TARCollege written in Java. This is my diploma project. GitHub. <https://github.com/jslim89/tarc-cgpa-calc>
43. Apply to CSUSB | Admissions | CSUSB. (n.d.). <https://www.csusb.edu/admissions/apply-csusb>
44. BMI calculator. (n.d.). <https://www.calculator.net/bmi-calculator.html>
45. Calorie calculator. (n.d.). <https://www.calculator.net/calorie-calculator.html>
46. Unit converter. (n.d.). <https://www.unitconverters.net/>
47. Calorie Intake and Weight Loss Calculator - NASM Bodyweight Tool. (n.d.). NASM. <https://www.nasm.org/resources/calorie-calculator>
48. Studious - apps on Google Play. (n.d.). <https://play.google.com/store/apps/details?id=leslie3141.android.studious&hl=en&gl=US>
49. IStudiez Pro for Windows – Best App for Students. (n.d.). <https://istudentpro.com/>
50. Home Page | Notion Template. (n.d.). Notion. <https://www.notion.so/product/notion-for-education>

Appendices

TELOS framework

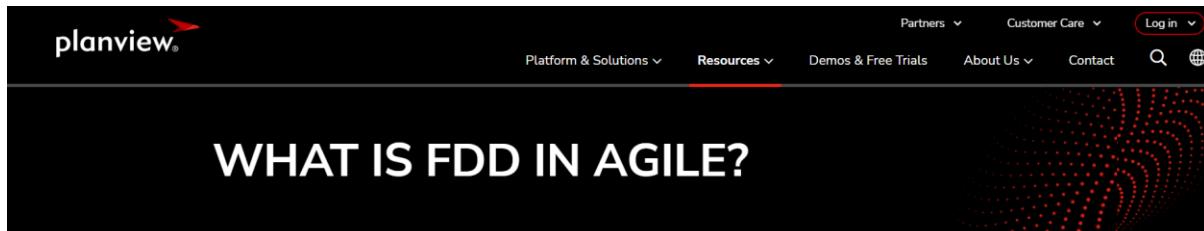


Editor. (2023, April 26). Technical feasibility study in software Engineering: Things to consider before development starts. AltexSoft. <https://www.altexsoft.com/blog/technical-feasibility/>



Smart College Chatbot using ML and Python. (n.d.). IEEE Conference Publication | IEEE Xplore.

<https://ieeexplore.ieee.org/abstract/document/9262426/metrics#metrics>



Lynn, R. (2021, May 21). What is FDD in Agile? Planview.

<https://www.planview.com/resources/articles/fdd-agile/>

The screenshot shows the Wrike Agile Guide website. The top navigation bar includes links for Why Wrike?, Features, Resources, Enterprise, Pricing, Contact Sales, EN, and Log in. A green button on the right says 'Try Wrike for free'. The main content area has a sidebar with a search bar and a 'Guide overview' list. The main text area discusses the iterative nature of FDD, domain object modeling, and the Scaled Agile Framework (SAFe). Below this is a section titled 'The five steps in FDD' with a note about five key activities.

What is FDD in Agile? | Wrike Agile Guide. (n.d.). <https://www.wrike.com/agile-guide/faq/what-is-fdd-in-agile/>

Link reference for FDD:

- <https://www.linkedin.com/pulse/fdd-have-you-heard-feature-driven-development-maxime-oriol/>
- <https://www.zentao.pm/blog/Introduction-to-Feature-Driven-Development-1461.html#:~:text=Improved%20quality%3A%20FDD%20emphasizes%20quality,the%20needs%20of%20end%2Dusers.>
- <https://www.bobstanke.com/blog/feature-driven-development>
- <https://adaptmethodology.com/blog/feature-driven-development/#:~:text=What%20are%20some%20advantages%20of,focus%20throughout%20the%20development%20process.>
- <https://www.planview.com/resources/articles/fdd-agile/#:~:text=An%20Agile%20methodology%20for%20developing,to%20track%20progress%20and%20results.>

Questionnaires

One-Stop Student System Survey

Hi, I am a student from Bachelor of Software Engineering (Honours) in Year 2, Semester 3.

I am proposing a system targeted for university students with function from study, health, personal categories.

This survey will take around 2 minutes to complete. Thanks for your participation!

seowhc-wm22@student.tarc.edu.my [Switch account](#)



Not shared

* Indicates required question

Age *

- 17-20
- 21-25
- 26-30
- 31-35
- 36-40
- Other:

What education level are you in? *

- Foundation/Pre-U (UEC, O Level, A Level etc)
- Diploma
- Bachelor Degree
- Master Degree
- PhD

What faculty are you from? *

- Faculty of Accountancy, Finance and Business (FAFB)
- Faculty of Applied Science (FOAS)
- Faculty of Computing and Information Technology (FOCS)
- Faculty of Built Environment (FOBE)
- Faculty of Engineering and Technology (FOET)
- Faculty of Communication and Creative Industries (FCCI)
- Faculty of Social Science and Humanities (FSSH)

Do you use any applications to ease your study, health and personal plaining? *

- iStudies
- Notion
- No
- Other: _____

How was your satisfaction in using it to help you during your university life?

1 2 3 4 5

Not satisfy

Satisfy

Do you need a function to predict your next semester grades in achieving target CGPA? *

Yes

No

Do you face any issue of mispronunciation? *

Yes

No

Do you currently using any BMI or calorie tracker app? *

Yes

No

Do you use the to-do list function to plan your tasks? *

- Yes
 No

Do you set any goal to achieve your aim? *

- Yes
 No

Do you face any queries about school when you are a freshie? *

- Yes
 No

According to previous question, what are the doubts/problems you encountered?

Your answer

Submit

Clear form