

# SC2006 Software Engineering



## Lab#1 Deliverables

Lab Group: SCEC E3

Group Name: Mission: Passable

Lab Supervisor: Mr Goh Tong Hai

Group Members:

CHUA QIN DI  
NICHOLAS CHANG CHIA KUAN  
REDDIPALLI SAI DHARSHAK SATHAVAHANA  
SHARMA HIMANSHU  
VONA TANISHA  
WU YIQING

## **Table of Content**

Table of Contents .....	2
1. Purpose .....	3
2. Documentation of Functional and Non-functional Requirements .....	3
A. Functional Requirements .....	3
B. Non-functional Requirements .....	5
3. Data Dictionary.....	6
4. Initial Use Case Model, Consisting of Use Case Diagram and Use Case Descriptions.....	7

## 1. Purpose

Our team, Mission: Passable, aims to empower students to manage their academic progress and their peers' through an application, StudyBuddy. StudyBuddy provides an interactive platform where students can add tasks into a personalized schedule, monitor their progress, and support each other through collaborative study sessions. StudyBuddy is designed to make studying more engaging, organized, and productive for students to stay on track and achieve academic success together.

## 2A. Functional Requirements

2.1 The app requires users to register for a new account.

2.1.1 The system must request for the user's username, password and email.

2.1.2 The system must validate all information provided by the user for user registration.

2.1.2.1 The system must validate that the username provided is not already registered.

2.1.2.2 The system must validate that the email provided is valid and not registered.

2.1.2.3 The system must validate that the password provided must meet the following requirements.

2.1.2.3.1 The password provided must be at least 8 characters long, containing 1 capital letter and 1 number.

2.1.3 The system must save the user's information to the database.

2.2 The app requires users to login to their account to use the app.

2.2.1 The app must request the user for their username and password for verification.

2.2.1.1 Username and Password must be registered in the database.

2.2.2 The app must return to home page with the user logged in upon successful verification.

2.3 The app allows users to manage their profile.

2.3.1 The system must be able to display user's information on their profile such as name, password, email, and courses.

2.3.2 Users shall be able to update their information on their profile.

2.3.2.1 The system must validate all information provided by the user for profile update.

2.3.2.2 The system must save the user's updated information to the database.

2.4 The app allows users to manage their friend list.

2.4.1 The system must be able to display the user's friend list.

2.4.2 Users shall be able to add friends by sending a friend request.

2.4.2.1 Username must exist in the database in order to send a friend request.

- 2.4.3 Users must be able to accept or reject friend requests.
- 2.5 The app allows users to manage courses in their profile.
  - 2.5.1 Users shall be able to add courses.
  - 2.5.2 The system must be able to display the user's courses.
    - 2.5.2.1 Courses must be registered into the database via Add Course in order for users to view them.
  - 2.5.3 Users shall be able to remove their own courses.
    - 2.5.3.1 Course requested to be removed by the user must be deleted from database.
  - 2.5.4 The system shall allow users to view their friend's courses.
    - 2.5.4.1 User must be registered as friends with another user in the database in order for users to view another user's courses.
- 2.6 The app allows users to manage the status of tasks in their timetable.
  - 2.6.1 Users shall be able to add tasks.
  - 2.6.2 The system must be able to display the user's tasks.
    - 2.6.2.1 Tasks must be register into the database via Add Task in order for users to view them.
  - 2.6.3 Users shall be able to remove tasks.
    - 2.6.3.1 Tasks requested to be removed by the user must be deleted from database
  - 2.6.4 Users shall be able to mark tasks as completed.
- 2.7 The app allows users to manage study groups.
  - 2.7.1 Users shall be able to create a study group.
    - 2.7.1.2 Users can only add friends into the study group.
  - 2.7.2 The system must be able to display the user's study groups.
    - 2.7.2.1 Study groups must be registered into the database via Create Study Group for users to view them.
  - 2.7.3 The app allows users to disband the study group.
- 2.8 The app allows users to manage timetable
  - 2.8.1 The app allows users to view their personal timetable.
  - 2.8.2 The app allows users to view their friend's timetable.
  - 2.8.3 Timetable should display tasks to be completed in the corresponding time slots.
- 2.9 The app allows users to set and receive notifications of their tasks.

## 2B. Non-functional Requirements

Usability	<ul style="list-style-type: none"><li>• The app must be simple and intuitive for users.</li></ul>
Reliability	<ul style="list-style-type: none"><li>• User data must be encrypted and secured.</li><li>• Data should be backed up once per week.</li><li>• The app must be able to handle at least 10,000 concurrent users without any performance issues.</li><li>• Users should be able to maintain at least 30 tasks without any performance issues.</li><li>• Users should be able to maintain at least 200 friends without any performance issue.</li><li>• The system must be upgradable as user count increases.</li><li>• The system should have 99.9% uptime.</li></ul>
Performance	<ul style="list-style-type: none"><li>• The app should retrieve and display data in less than 3 seconds to users.</li><li>• Once a task/course/friend is added, the system must register the data into the database and display it to the users within 2 seconds.</li></ul>
Supportability	<ul style="list-style-type: none"><li>• The app must be functional on all mobile Operating Systems: Apple and Android.</li></ul>

### 3. Data Dictionary

Term	Definition
Course	A course is a basic unit of teaching. A course must be either compulsory or elective. A course must include lectures and tutorials . Some courses may have labs.
Task	A piece of work to be done or undertaken.
Timetable	A chart showing the times at which events are scheduled to take place or when tasks are scheduled to be completed.
Notifications	An alert (typically a pop-up or other message) generated by an application to notify the user of a new message or update.
Friend (Online)	A person met and communicated with through the internet.
Friend Request	An online invitation to allow a user to become a Friend of the sender. Receiver can accept or reject the Friend Request
Friend List	A list of people that a user is connected to on a social networking site.
Study Group	A group of people who gather virtually to collaboratively study without needing to be physically present in the same location.
Username	Unique username chosen by the user
Password	Encrypted password for authentication Requires at least 8 characters long, containing 1 capital letter and 1 number
Email	User's email address (must be unique)
Add Course	A function allowing users to add a course associated to the user
Add Task	A function allowing users to add a task associated to the user
Create Study Group	A function allowing users to create a Study Group by adding Friends into the group

#### 4. Use Case Descriptions

Refer to UseCaseDiagram.pdf for the Use Case Diagram

Use Case ID:	DB1		
Use Case Name:	Register		
Created By:	Chua Qin Di	Last Updated By:	Chua Qin Di
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Register</b> allows new users to create an account in our system database. To create an account, users must input Username, Email, Password
Preconditions:	<ol style="list-style-type: none"><li>1. Username must not be taken</li><li>2. Email must not be linked to existing account</li><li>3. Passwords must meet requirements: must be at least 8 characters long, containing 1 capital letter and 1 number.</li></ol>
Postconditions:	
Priority:	High
Frequency of Use:	Once
Flow of Events:	<ol style="list-style-type: none"><li>1. User inputs required information: Username, Email, Password.</li><li>2. System validates the information.</li><li>3. System creates the account.</li><li>4. System updates account information in database.</li><li>5. System directs the user to the homepage.</li></ol>
Alternative Flows:	<p>Username taken</p> <ol style="list-style-type: none"><li>1. System displays "User account already exists"</li><li>2. System prompts user to input a different username</li><li>3. Continue from main flow step 1</li></ol> <p>Email linked to existing account</p> <ol style="list-style-type: none"><li>1. System displays "Email linked to existing account"</li><li>2. System prompts user to log in to existing account or register account</li><li>3. If log in, System directs the user to the Login page</li><li>4. If register account, continue from main flow step 1</li></ol> <p>Password does not meet requirements</p> <ol style="list-style-type: none"><li>1. System displays "Password does not meet requirements"</li><li>2. System prompts user to re-enter their password.</li><li>3. Continue from main flow Step 1.</li></ol>
Exceptions:	

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	



Use Case ID:	DB2		
Use Case Name:	Login		
Created By:	Chua Qin Di	Last Updated By:	Chua Qin Di
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Login</b> allows users to enter the app and use the features intended for registered users of the app. To login to the system, the user must input their username and the corresponding password.
Preconditions:	User must have an existing account in the database
Postconditions:	System displays the main homepage
Priority:	High
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. User enters username and password on the login page.</li> <li>2. System validates the user's username and password.</li> <li>3. System shows the main homepage.</li> </ol>
Alternative Flows:	<p>Invalid Username</p> <ol style="list-style-type: none"> <li>1. System cannot find the username in the database.</li> <li>2. System displays an error message.</li> <li>3. System prompts the user to input username and password or register an account.</li> <li>4. Use case resumes at main flow step 1 or Register account page</li> </ol> <p>Invalid Password</p> <ol style="list-style-type: none"> <li>1. Password does not match for the username.</li> <li>2. System displays an error message.</li> <li>3. System prompts user to input username and password.</li> <li>4. Use case resumes at main flow step 1.</li> </ol>
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	PF1		
Use Case Name:	Manage Profile		
Created By:	Wu Yiqing	Last Updated By:	Wu Yiqing
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Manage Profile</b> allows users to view and update personal information.
Preconditions:	<ol style="list-style-type: none"> <li>1. User must have an existing account in the database.</li> <li>2. The user must have logged into the app.</li> </ol>
Postconditions:	
Priority:	High
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. User navigates to profile settings.</li> <li>2. User selects the option to update profile.</li> <li>3. User inputs updated information. (e.g., name, password and email)</li> <li>4. User submits updated information.</li> <li>5. System validates the updated information.</li> <li>6. System updates the new profile into database.</li> <li>7. System displays new profile for user.</li> </ol>
Alternative Flows:	
Exceptions:	<p>Invalid Data Entry</p> <ol style="list-style-type: none"> <li>1. User enters an invalid information format.</li> <li>2. System displays error message and rejects update profile request.</li> <li>3. User must rectify the invalid format before proceeding.</li> </ol> <p>User cancels update profile request.</p> <ol style="list-style-type: none"> <li>1. User chooses to exit the update profile page.</li> <li>2. App returns to home page.</li> </ol>
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	FR1		
Use Case Name:	Manage Friends		
Created By:	Vona Tanisha	Last Updated By:	Vona Tanisha
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Manage Friends</b> allows users to add existing users as friends or allows users to accept another user's Friend Request or remove friends from their friend list.
Preconditions:	Friend must be an existing user in the database
Postconditions:	Request will be sent to friend, pending his approval Friend shows up on Friend List
Priority:	Medium
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. User inputs another existing user's username to add as friend</li> <li>2. System displays "Friend request sent"</li> <li>3. User chooses to press Accept or Reject Friend Request</li> <li>4. User removes friend from Friend List</li> </ol>
Alternative Flows:	<p>Cannot add yourself as Friend</p> <ol style="list-style-type: none"> <li>1. System displays error message and prompt user to input another username</li> </ol> <p>Username does not exist</p> <ol style="list-style-type: none"> <li>1. System displays error message and prompt user to input another username</li> </ol> <p>Already your friend</p> <ol style="list-style-type: none"> <li>1. System displays error message and prompt user to input another username</li> </ol>
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	CO1		
Use Case Name:	Manage Course		
Created By:	Chua Qin Di	Last Updated By:	Chua Qin Di
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Manage Course</b> allows users to add, remove or view their current courses into the database.
Preconditions:	
Postconditions:	
Priority:	Medium
Frequency of Use:	Occasional
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects Add Course, View Course or Remove Course <ol style="list-style-type: none"> <li>a. Add Course allows users to add their courses into their course list by inputting course ID</li> <li>b. View Course allows users to view the information of the selected course</li> <li>c. Remove Course allows users to remove the course from their course list</li> </ol> </li> </ol>
Alternative Flows:	
Exceptions:	
Includes:	Add Course, View Course, View Friend's Course, Remove Course
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	TS1		
Use Case Name:	Manage Task		
Created By:	Chua Qin Di	Last Updated By:	Chua Qin Di
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Manage Task</b> allows users to add, view or remove their tasks on the database
Preconditions:	
Postconditions:	
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects Add Task, View Task or Remove Task <ol style="list-style-type: none"> <li>a. Add Task allows users to add tasks to complete onto the their task list by inputting the description of the task and the deadline</li> <li>b. View Task allows users to view the information of the selected task</li> <li>c. Remove Task allows users to remove the task from the task list</li> </ol> </li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User can mark task as completed</li> </ol>
Exceptions:	
Includes:	Add Task, View Task, View Friend's Task, Remove Task
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	SG1		
Use Case Name:	Manage Study Group		
Created By:	Vona Tanisha	Last Updated By:	Vona Tanisha
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>Manage Study Group</b> allows users to create study group or disband it, and also send notifications to group members
Preconditions:	Study group members must be friends first. Only the creator of the Study Group can disband it.
Postconditions:	
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects create study group or disband study group <ol style="list-style-type: none"> <li>a. Create a study group by inputting the usernames of the group members</li> <li>b. Disband the study group will remove all members and delete the group from the database</li> </ol> </li> <li>2. Users of an existing group can send notifications to their group members</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. If user is not connected to the internet <ol style="list-style-type: none"> <li>a. User will be notified to try again when the internet connection is restored.</li> </ol> </li> </ol>
Exceptions:	
Includes:	Create Study Group, Disband, Notify Friends
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	TB1		
Use Case Name:	View Timetable		
Created By:	Vona Tanisha	Last Updated By:	Vona Tanisha
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>View Timetable</b> allows users to view their personal timetable and their Friends timetables
Preconditions:	Must be friends to view other users timetable
Postconditions:	
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> <li>1. User can Choose to view personal timetable</li> <li>2. User can choose a Friends timetable to view</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. If the user is not connected to the internet <ol style="list-style-type: none"> <li>a. The system notifies the user that they will not be able to view a friend's timetable till internet connection is restored.</li> </ol> </li> </ol>
Exceptions:	
Includes:	View personal timetable, View Friends Timetable
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	NT1		
Use Case Name:	Notifications		
Created By:	Wu Yiqing	Last Updated By:	Wu Yiqing
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User, Database
Description:	<b>Notifications</b> are alerts received by users when the system detects upcoming deadlines for tasks, and messages sent to the user from other users.
Preconditions:	<ol style="list-style-type: none"> <li>1. Upcoming deadlines for task</li> <li>2. Incoming friend requests</li> <li>3. Friend notifies user</li> </ol>
Postconditions:	
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> <li>1. System detects that a message is sent to the user</li> <li>2. The app displays an alert on screen for the user with the information</li> </ol>
Alternative Flows:	E1. User is not connected to internet <ol style="list-style-type: none"> <li>1. System delays notification until user is connected to the internet</li> </ol>
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	



Use Case ID:	PG1		
Use Case Name:	ProgressBar		
Created By:	Chua Qin Di	Last Updated By:	Chua Qin Di
Date Created:	3 Feb 2025	Date Last Updated:	3 Feb 2025

Actor:	User
Description:	<b>ProgressBar</b> is a graphical UI to show users the completion rate of their current tasks
Preconditions:	1. Have at least one current task
Postconditions:	
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	1. The app displays a graphical progress bar to show the percentage of tasks to be completed
Alternative Flows:	No ongoing tasks 1. The app will display a message indicating that there are no tasks to be completed at the moment
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	