

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 – Software Engineering

Lab 5 Deliverables

Lab Group	SCEZ Group 4
Team	ChildConnect
Members	Wee Zi Hao (Huang Zihao) U2323380H
	Sivaguruanathan Keerthivasan U2322668E
	Tan Yi Jun U2322560C
	Edwin Tan Yu Qi U2223581H
	Kumar Advaith U2323503B

Functional Requirements.....	5
Parents:.....	5
1. Landing page.....	5
2. Registration page.....	5
3. Parent Login Page.....	5
4. Forgot Password page.....	6
5. Parent Taskbar.....	6
6. Parent Home page.....	6
7. Parent Children page.....	7
8. Parent Child Detail Page.....	7
9. Parent “Chats” page.....	8
10. Parent Messaging Page.....	8
11. Parent Profile page.....	8
12. Parents Grades Page.....	9
Teachers:.....	9
13. Teacher Home Page.....	9
14. Teacher Classes Page.....	10
15. Teacher Class Details Page.....	10
16. Teacher Student Page.....	10
17. Teacher Homework Assignment Page.....	11
18. Teacher Homework List.....	12
19. Teacher Announcement List Page.....	13
20. Teacher Add Announcement Page.....	13
21. Teacher Student Detail Page.....	14
22. Teacher Taskbar.....	15
23. Teachers “Chats” page.....	15
24. Teachers Messaging Page.....	15
25. Teacher Profile page.....	16
26. Teachers Grades Page.....	16
Child:.....	17
27. Child mode.....	17
Non-Functional Requirements.....	18
1. Usability.....	18
2. Reliability.....	18
4. Security.....	18
5. Maintainability.....	19
6. Scalability.....	19
7. Compatibility.....	19
Data Dictionary.....	19
Use Case Diagrams.....	23
Use Case Diagram 1 (Overall).....	24

Use Case Diagram 2 (Child Academic Interaction System).....	25
Use Case Diagram 3 (Access and Navigation System).....	26
Use Case Diagram 4 (Location Tracking System).....	27
Use Case Diagram 5 (Grading System).....	28
Use Case Descriptions.....	29
Registration System.....	29
Login System.....	30
Forget Password System.....	31
Navigation Bar (Parent).....	32
Navigation Bar (Teacher).....	33
View Homework (Parent).....	34
View documentation (Parent).....	34
Add documentation (Parent).....	35
View attendance (Parent).....	36
View School (Parent).....	37
Child Mode (Parent).....	38
Mark Attendance (Teacher).....	38
Assign Homework (Teacher).....	39
Add Announcement (Teacher).....	40
View Homework List (Teacher).....	41
View Announcement List (Teacher).....	42
Track Location.....	43
Chat.....	44
Editing profile page.....	45
Viewing Homepage (Parent).....	46
Viewing Homepage (Teacher).....	46
Add & View Grades (Teacher).....	47
Sequence Diagrams.....	48
Use Case 1:.....	48
Use Case 2:.....	49
Use Case 11:.....	56
Use Case 13:.....	58
Use Case 14:.....	59
Use Case 15:.....	60
Use Case 16:.....	61
Use Case 17:.....	62
Use case 18:.....	63
Use Case 19:.....	64
Use Case 20:.....	65
Use Case 21:.....	66
Use Case 22:.....	67

Class Diagram.....	68
System Architecture.....	69
State Machine Diagram.....	70
Dialog Map.....	71
Application Skeleton.....	71
Application Testing.....	74
Black Box Testing.....	74
Test Cases and Testing Results.....	75
Sign Up (Register Here).....	75
Parent Registration.....	76
Teacher Registration.....	78
Login (Get Started).....	80
Parent Account.....	81
Children (Select a Child).....	81
Chats.....	82
Home.....	82
Profile.....	83
Child Mode.....	84
Teacher Account.....	84
Home.....	84
Chats.....	85
Classes.....	85
Individual Profile of Student (Classes).....	87
White Box Testing.....	88
Control Flow Testing.....	89
User Account Registration.....	89
User Account Login.....	90
User Account Logout.....	91
Edit Profile.....	92
Change Exit Code.....	93
Add Announcement.....	94
Assign Homework.....	95
Add Announcement.....	96
Add Documentation.....	97
Grading System.....	98

Functional Requirements

Parents:

1. Landing page

- 1.1. The application will be in the English language
- 1.2. When the User clicks on the application, they will be guided to the landing page if they are not signed in
- 1.3. The application must check whether or not the User has provided it with Global Positioning System (GPS) permission
 - 1.3.1. If the GPS permission is not provided, User will be prompted to give the app permission to use GPS
 - 1.3.2. If GPS permission is denied, features that require location access (e.g., child tracking) will be disabled.
 - 1.3.2.1. If the user refuses to accept GPS permission, the app will prompt the user with a message urging them to give access.
 - 1.3.2.2. If the user still refuses, the app will continue to work without location features.

2. Registration page

- 2.1. Users will select between 2 buttons, Parent or Teacher
- 2.2. All input fields shall display placeholder text to guide user input. Validation messages shall appear in red if inputs are invalid
- 2.3. Users must have the option to navigate to a login page if they already have an existing account

3. Parent Login Page

- 3.1. Users with an existing account must be able to login into the application with their respective email address
 - 3.1.1. The input box will be identified by words prompting the user to input their registered email address
 - 3.1.2. The user must click on the text box
 - 3.1.3. If the system is unable to verify the correctness of the email address, it will print an error message “Invalid Email”
- 3.2. Users with an existing account must type in the password created when registering for the account
 - 3.2.1. The input box will be identified by words prompting the user to input the password that they createdF
 - 3.2.2. The user must click on the text box
 - 3.2.3. If the system is unable to verify the correctness of the password, it will print an error message “Wrong Password”

- 3.3. If the both the email address field and password field can be ascertained, the user will be navigated to Home page
- 3.4. The system must also display a “Forgot Password” field to allow User to retrieve their respective account login information
- 3.5. Users must have the option to navigate to the registration page if they do not have an account

4. Forgot Password page

- 4.1. Users will have a input box to type in the email that they registered for an account with
 - 4.1.1. The input box will be identified with words prompting the user to input their email
 - 4.1.2. Users must click on the textbox
- 4.2. User will then click the send button to send a password reset link to their email
 - 4.2.1. If the email entered is not valid, the input box will be highlighted red
 - 4.2.2. If the email entered is not valid, an error message 'Wrong email' will be shown.
- 4.3. Users must be provided with the option to navigate back to the sign in page
 - 4.3.1. They can do this by clicking on the “Sign In” button

5. Parent Taskbar

- 5.1. User will be able to access the task bar at the bottom of the application once they are successfully logged in
- 5.2. User will have 4 mutually exclusive options: “Home”, “Children”, “Chat”, “Profile”
 - 5.2.1. If “Home” is chosen, User must be redirected to the Home page
 - 5.2.2. If “Children” is chosen, User must redirected to the Children page
 - 5.2.3. If “ Chat” is chosen, User must be redirected to the Chat page
 - 5.2.4. If “Profile” is chosen, User must be redirected to the Profile page
 - 5.2.5. User will be able to freely navigate between each of the pages mentioned above by clicking on the relevant page they want to be directed to

6. Parent Home page

- 6.1. Users will be able to view any announcements that the teacher or school has sent out in the announcement box
 - 6.1.1. If there are no new announcements, the announcement box will display “You have no new announcements”
 - 6.1.2. Announcement cards will include the class, teacher who posted the announcement, and which children the announcement is for

7. Parent Children page

- 7.1. Each child registered under the parent is displayed as a button/card.
- 7.2. Each child card displays:
 - 7.2.1.1. Child's full name
 - 7.2.1.2. Child's school name
- 7.3. On Tap: Pressing a child's card navigates to the Child Detail Page

8. Parent Child Detail Page

- 8.1. The top section displays:
 - 8.1.1. Child's name
 - 8.1.2. Child's profile photo
 - 8.1.3. Today's attendance status (e.g., Present, Absent)
 - 8.1.4. Child's assigned teacher which has:
 - 8.1.4.1. Chat Button: Navigates to the private chat with the teacher.
 - 8.1.4.2. Contact Info Button: Navigates to the teacher's profile page with contact details.
- 8.2. The screen has 3 navigation buttons:
 - 8.2.1. View Attendance Records: Opens past attendance data.
 - 8.2.2. View Documentation: Opens certificates, report cards, and other documents.
 - 8.2.3. View Homework List: Opens the list of homework assigned to the child.
- 8.3. The screen has a location tracking feature comprising of:
 - 8.3.1. Activate Tracking Button:
 - 8.3.1.1. Toggles between Activate Tracking and Stop Tracking depending on backend status.
 - 8.3.1.2. Starts/stops polling child's location and notifies the database.
 - 8.3.2. Map View:
 - 8.3.2.1. Displays real-time markers for:
 - 8.3.2.1.1. Child's location
 - 8.3.2.1.2. Parent's location
 - 8.3.2.1.3. School location
 - 8.3.2.2. Map dynamically adjusts to include all three markers.
 - 8.3.2.3. View Fullscreen Map Button: Opens a new screen with the same markers in full-screen view.
- 8.4. The top right of the screen has 2 buttons:
 - 8.4.1. Child Mode button:
 - 8.4.1.1. Which navigates to the child mode screen
 - 8.4.1.2. Change Child Mode Password Button:
 - 8.4.1.2.1. Opens a modal window.
 - 8.4.1.2.2. Allows parents to set or change the PIN required to exit child mode.

- 8.5. Upon entering the child's detail screen for the first time, the user will be prompted to set a password for the child mode screen
- 8.6. The entire screen is scrollable

9. Parent “Chats” page

- 9.1. Displays a list of all ongoing 1-to-1 chats between the parent and teachers.
- 9.2. Each chat item includes:
 - 9.2.1. Teacher's name
 - 9.2.2. Teacher's profile picture (optional)
 - 9.2.3. Last message preview
 - 9.2.4. Timestamp of the last message
- 9.3. On Tap: Tapping a chat item opens the Chat Page for that teacher.
- 9.4. The “New” indicator appears for chats that have been unopened

10. Parent Messaging Page

- 10.1. A typical chat interface with:
 - 10.1.1. Message Input Box: Allows parents to type messages (supports text and emojis).
 - 10.1.1.1. Messages are pressable to change its language into the four national languages of Singapore
 - 10.1.1.2. Messages have an audio button next to them
 - 10.1.1.2.1. Pressing the audio button narrates the message out loud in the language that it is currently in
 - 10.1.2. Send Button: Sends the message to the teacher.
 - 10.1.3. Record Button is available for the parent to record their voice
 - 10.1.3.1. Recorded voice gets translated into text to be able to be sent
 - 10.1.4. Delivered/Read Receipts:
 - 10.1.4.1. Messages show sent, delivered, and read statuses.in the form of ticks
 - 10.1.5. Image Sending:
 - 10.1.5.1. Parents can attach and send images.
 - 10.1.5.2. Image messages should appear inline within the chat.
- 10.2. Chat history is persisted and synced from the backend.

11. Parent Profile page

- 11.1. Users will be able to edit their profile information by clicking on the ‘Edit’ button at the top right hand corner.
 - 11.1.1. If the User clicks on the “Edit” button, information that can be edited such as Phone or Work Number will have a corresponding input box

- 11.1.2. For information that are fixed such as name and email, the container for the information will be greyed out and non-interactable
- 11.2. Users will be able to view their personal information in the profile page
 - 11.2.1. Users will be able to view their profile photo displayed within a circular border
 - 11.2.2. Users registered full name will be displayed in the profile page
 - 11.2.3. Users contact information will be displayed in the profile page
 - 11.2.3.1. Contact information includes details such as the phone number, email, work number and the home address
- 11.3. A logout button for users to log out of the current account

12. Parents Grades Page

- 12.1. Parents will be able to view their children's performance in school
- 12.2. Parents will be able to see one to three different colours in a pie chart indicating the child's overall performance
 - 12.2.1. Green portions of the pie chart indicates the child has done excellent
 - 12.2.2. Yellow portions of the pie chart indicates the child has done average
 - 12.2.3. Red portions of the pie chart indicates the child has done poor
- 12.3. There is a colour legend at the bottom
- 12.4. Parents will be able to see the individual test grades for the child
 - 12.4.1. Name of the test is displayed
 - 12.4.2. Score is displayed
 - 12.4.2.1. Score is displayed in green if the child does well
 - 12.4.2.1.1. "Excellent" is displayed
 - 12.4.2.2. Score is displayed in yellow if the child does average
 - 12.4.2.2.1. "Average" is displayed
 - 12.4.2.3. Score is displayed in red if the child does poorly
 - 12.4.2.3.1. "Poor" is displayed
 - 12.4.3. Class is displayed
 - 12.4.4. Statistics are displayed
 - 12.4.4.1. Average is displayed
 - 12.4.4.2. Median is displayed
 - 12.4.4.3. Mode is displayed
 - 12.4.4.4. Max is displayed
 - 12.4.4.5. Min is displayed

Teachers:

13. Teacher Home Page

- 13.1. Header Section:
 - 13.1.1. Displays the teacher's profile picture at the top of the screen.

- 13.1.2. Profile picture is displayed in a circular container.
- 13.2. Announcements Section:
 - 13.2.1. Shows a list of relevant announcements for the teacher.
 - 13.2.2. Announcement items include:
 - 13.2.2.1. Unread chat messages from parents
- 14. **Teacher Classes Page**
 - 14.1. Displays a scrollable list of all classes assigned to the teacher.
 - 14.2. Each class item/card includes:
 - 14.2.1. Class Name (e.g., "Grade 3A")
 - 14.2.2. Teacher Role:
 - 14.2.2.1. If the teacher is assigned as a form teacher, display "Form Teacher"
 - 14.2.2.2. If the teacher is assigned a subject only, display "Subject Teacher"
 - 14.3. On Tap: Tapping on a class card navigates to the Class Details Page for that class.

15. **Teacher Class Details Page**

- 15.1. Header section with:
 - 15.1.1. Class name, grade, and subject.
 - 15.1.2. Total students (e.g., "15 Students").
 - 15.1.3. Attendance summary (e.g., "12 Arrived, 3 Absent").
- 15.2. Buttons next to the class name:
 - 15.2.1. Assign Homework: Redirects to Homework Assignment Page.
 - 15.2.2. Class Settings: Edit class details (e.g., schedule, room number).
- 15.3. Student list with:
 - 15.3.1. Student names and profile photos.
 - 15.3.2. Current attendance status ("Present," "Absent," "Unconfirmed").
 - 15.3.3. Manage button next to each student to access their Student Page.
- 15.4. Real-time attendance updates (sync with GPS-based check-in/check-out).
- 15.5. Manual override for attendance status (e.g., mark a student "Present" if GPS fails).
- 15.6. Export attendance data for the day/week as PDF/CSV.
- 15.7. "Refresh" button to update attendance statuses.
- 15.8. Highlight students with recurring absences.

16. **Teacher Student Page**

- 16.1. Header Section:
 - 16.1.1. Displays the class name at the top of the screen.
 - 16.1.2. Styled prominently to indicate current context (e.g., "Grade 3A").
- 16.2. Top Action Buttons:
 - 16.2.1. Two horizontally aligned buttons:
 - 16.2.1.1. Homework List Button:

- 16.2.1.1.1. Navigates to the list of assigned homework for the class.
- 16.2.1.2. Announcement List Button:
 - 16.2.1.2.1. Navigates to the list of announcements for the class.
- 16.3. Middle Action Buttons:
 - 16.3.1. Two horizontally aligned buttons:
 - 16.3.1.1. Assign Homework Button:
 - 16.3.1.1.1. Navigates to the homework creation screen.
 - 16.3.1.2. Add Announcement Button:
 - 16.3.1.2.1. Navigates to the announcement creation screen.
- 16.4. Student Search:
 - 16.4.1. A search bar is provided for filtering students by name.
 - 16.4.2. Input updates the displayed student list in real time.
 - 16.4.3. Case-insensitive and partial matches are supported.
- 16.5. Student List:
 - 16.5.1. A list of all students enrolled in the class
 - 16.5.2. Each student is displayed as a card/button that includes:
 - 16.5.2.1. Profile picture
 - 16.5.2.2. Student name.
 - 16.5.2.3. Today's attendance status:
 - 16.5.2.3.1. Present → Green text
 - 16.5.2.3.2. Absent → Red text
 - 16.5.3. Attendance Toggle
 - 16.5.3.1. Tapping on a student card toggles their attendance status:
 - 16.5.3.2. Sends a request to the backend to mark the student present or absent for today.
 - 16.5.3.3. Automatically updates the UI based on the backend response.
 - 16.5.3.4. Status is persisted and synced with backend
 - 16.5.4. Profile Button
 - 16.5.4.1. Each student card includes a “Profile” button.
 - 16.5.4.2. Navigates to the Student Detail Page

17. **Teacher Homework Assignment Page**

- 17.1. Page Overview
 - 17.1.1. Teachers can assign homework to a specific class.
 - 17.1.2. The page is titled “Assign Homework” and is accessible from the class details screen.
- 17.2. Fields to Fill
 - 17.2.1. The form includes the following required inputs:
 - 17.2.2. Title: Text field for the name of the homework (e.g., “Science Practice 2”).

- 17.2.3. Content: Multiline input for homework instructions or tasks (e.g., “Read Chapter 3, Answer Questions 1-5”).
 - 17.2.4. Due Date: Teachers select a due date from a calendar-style date picker.
 - 17.2.5. Subject: Teachers select the subject from a dropdown (e.g., Math, English, Science, Mother Tongue).
- 17.3. Subject Selection
- 17.3.1. Teachers tap to open a dropdown menu.
 - 17.3.2. Predefined subject options are available.
 - 17.3.3. The selected subject is visibly displayed.
- 17.4. Submit Homework
- 17.4.1. Teachers can tap “Submit Homework” to assign the task.
 - 17.4.2. If any field is empty, a warning prompts the teacher to fill all fields.
 - 17.4.3. Upon successful submission:
 - 17.4.3.1. A success message is shown.
 - 17.4.3.2. The teacher is redirected to the previous screen.
 - 17.4.4. If submission fails, an error message is shown to the teacher.

18. Teacher Homework List

- 18.1. Page Overview
- 18.1.1. Displays a list of all homework assignments created for a specific class.
 - 18.1.2. The page is titled “Homework” and accessible from the Class Details Page.
- 18.2. Homework Display
- 18.2.1. Each homework item includes:
 - 18.2.1.1. Title of the homework (e.g., “Math Practice 1”)
 - 18.2.1.2. Content or description (e.g., “Do questions 1-5 on page 12”)
 - 18.2.1.3. Subject (e.g., “English”)
 - 18.2.1.4. Due date (formatted for Singapore locale, e.g., “6 April 2025”)
 - 18.2.1.5. Status:
 - 18.2.1.6. Displayed as “Open” (green text) or “Closed” (red text)
- 18.3. Interaction
- 18.3.1. Tap to Toggle:
 - 18.3.1.1. Tapping a homework item toggles its status between “Open” and “Closed.”
 - 18.3.1.2. This reflects whether the homework is currently active or no longer applicable.
 - 18.3.2. Visual Feedback:
 - 18.3.2.1. Homework cards have clear, distinguishable formatting.
 - 18.3.2.2. Empty states are handled with a message: “No homework assigned to this class.”

19. Teacher Announcement List Page

19.1. Page Overview

- 19.1.1. Displays a list of all announcements created for a specific class.
- 19.1.2. The page is titled “Announcements” and is accessible from the Class Details Page.

19.2. Announcement Display

- 19.2.1. Each announcement includes the following information:
- 19.2.2. Title of the announcement (e.g., “Consent Forms Due Tomorrow”)
- 19.2.3. Content (e.g., “Please submit the signed forms by Friday.”)
- 19.2.4. Created By: The name of the teacher who posted the announcement
- 19.2.5. Date Created: Displayed in Singapore date format (e.g., “6 April 2025”)
- 19.2.6. Status:
 - 19.2.6.1. Displayed as “Open” (green text) or “Closed” (red text)

19.3. Interaction

19.3.1. Tap to Toggle:

- 19.3.1.1. Tapping on an announcement toggles its status between “Open” and “Closed”, indicating whether the announcement is currently active or archived.

19.3.2. Visual Feedback:

- 19.3.2.1. Announcements are shown in clean cards with spacing between them.
- 19.3.2.2. If there are no announcements, display a message:
- 19.3.2.3. “No announcements for this class.”

20. Teacher Add Announcement Page

20.1. Page Overview

- 20.1.1. Allows teachers to create a new announcement for a specific class.
- 20.1.2. The page is titled “Add Announcement” and is accessible from the Class Details Page.

20.2. Input Fields

20.2.1. Title:

- 20.2.1.1. A required text field for the name of the announcement (e.g., “Exam Reminder”).

20.2.2. Content:

- 20.2.2.1. A required multiline text field for the announcement details (e.g., “Science exam on Friday at 9am”).
- 20.2.2.2. Minimum height ensures comfortable typing space for longer messages.

20.3. Submission Flow

- 20.3.1. Submit Announcement Button:
- 20.3.2. Tapping this button submits the announcement to the system.
- 20.3.3. If either the title or content is missing, the teacher is prompted with an alert to complete all fields.

- 20.3.4. Upon successful submission:
 - 20.3.4.1. A confirmation message is shown (e.g., “Announcement added.”).
 - 20.3.4.2. The user is navigated back to the previous screen.
- 20.3.5. If submission fails, an error message is displayed.

21. **Teacher Student Detail Page**

- 21.1. Page Overview:
 - 21.1.1. This page displays detailed information about a specific student.
 - 21.1.2. It is accessed by tapping the “Profile” button beside a student on the Class Details Page.
- 21.2. Student Profile Section
 - 21.2.1. Profile Picture is displayed prominently at the top.
 - 21.2.1.1. If not available, a default avatar is shown.
 - 21.2.2. Student Details:
 - 21.2.2.1. Full name
 - 21.2.2.2. School name
 - 21.2.2.3. Class
 - 21.2.2.4. Grade
- 21.3. Parental Information Section:
 - 21.3.1. Two profile cards are shown — one for each parent (Father and Mother).
 - 21.3.2. Each parent card includes:
 - 21.3.2.1. Profile picture (or default avatar)
 - 21.3.2.2. Full name
 - 21.3.2.3. Role (e.g., Father, Mother)
 - 21.3.2.4. “Chat” button – initiates a private chat between teacher and parent
 - 21.3.2.5. “Contact Info” button – navigates to a page displaying the parent’s contact details
- 21.4. Student-Related Navigation
 - 21.4.1. Three action buttons are available at the bottom of the profile:
 - 21.4.1.1. Attendance Records – Navigates to the student’s historical attendance page
 - 21.4.1.1.1. On this page, the user is able to edit the selected student’s attendance on specific days by:
 - 21.4.1.1.1.1. pressing on the day he/she wishes to edit
 - 21.4.1.1.1.2. Pressing either “Mark Present” or “Mark Absent” button
 - 21.4.1.1.2. The user can also upload a photo of the student as proof of his/her attendance
 - 21.4.1.2. Documentation – Navigates to a page with documents like medical forms, report cards, etc.
 - 21.4.1.2.1. User can also add documentation by tapping on the “Add Documentation” button where they can:
 - 21.4.1.2.1.1. Enter the title and content of the documentation

- 21.4.1.2.1.2. Choose between uploading an image or a pdf
 - 21.4.1.3. Homework List – Opens the list of homework assigned to the student
- 21.5. UI and Feedback
- 21.5.1. While loading, a spinner and loading message are displayed.
 - 21.5.2. The layout is scrollable, responsive, and styled for clarity and accessibility.

22. Teacher Taskbar

- 22.1. User will be able to access the task bar at the bottom of the application once they are successfully logged in
- 22.2. User will have 4 mutually exclusive options: “Home”, “Classes”, “Chat”, “Profile”
 - 22.2.1. If “Home” is chosen, User must be redirected to the Home page
 - 22.2.2. If “Classes” is chosen, User must redirected to the Classes page
 - 22.2.3. If “ Chat” is chosen, User must be redirected to the Chat page
 - 22.2.4. If “Profile” is chosen, User must be redirected to the Profile page
 - 22.2.5. User will be able to freely navigate between each of the pages mentioned above by clicking on the relevant page they want to be directed to

23. Teachers “Chats” page

- 23.1. Displays a list of all ongoing 1-to-1 chats between the parent and teachers.
- 23.2. Each chat item includes:
 - 23.2.1. Teacher’s name
 - 23.2.2. Teacher’s profile picture (optional)
 - 23.2.3. Last message preview
 - 23.2.4. Timestamp of the last message
- 23.3. On Tap: Tapping a chat item opens the Chat Page for that teacher.
- 23.4. The “New” indicator appears for chats that have been unopen

24. Teachers Messaging Page

- 24.1. A typical chat interface with:
 - 24.1.1. Message Input Box: Allows parents to type messages (supports text and emojis).
 - 24.1.1.1. Messages are pressable to change its language into the four national languages of Singapore
 - 24.1.1.2. Messages have an audio button next to them
 - 24.1.1.2.1. Pressing the audio button narrates the message out loud in the language that it is currently in

- 24.1.2. Send Button: Sends the message to the teacher.
- 24.1.3. Record Button is available for the parent to record their voice
 - 24.1.3.1. Recorded voice gets translated into text to be able to be sent
- 24.1.4. Delivered/Read Receipts:
 - 24.1.4.1. Messages show sent, delivered, and read statuses.in the form of ticks
- 24.1.5. Image Sending:
 - 24.1.5.1. Parents can attach and send images.
 - 24.1.5.2. Image messages should appear inline within the chat.
- 24.2. Chat history is persisted and synced from the backend.

25. Teacher Profile page

- 25.1. Users will be able to edit their profile information by clicking on the ‘Edit’ button at the top right hand corner.
 - 25.1.1. If the User clicks on the “Edit” button, information that can be edited such as Phone or Work Number will have a corresponding input box
 - 25.1.2. For information that are fixed such as name and email, the container for the information will be greyed out and non-interactable
- 25.2. Users will be able to view their personal information in the profile page
 - 25.2.1. Users will be able to view their profile photo displayed within a circular border
 - 25.2.2. Users registered full name will be displayed in the profile page
 - 25.2.3. Users contact information will be displayed in the profile page
 - 25.2.3.1. Contact information includes details such as the phone number, email, work number and the home address
- 25.3. A logout button for users to log out of the current account

26. Teachers Grades Page

- 26.1. Teachers will be able to view their a list of test that the students in their class have taken
- 26.2. Teachers will be able to click into each individual test to view information about each test
 - 26.2.1. The name of the test is displayed
 - 26.2.2. The class is displayed
 - 26.2.3. Teachers will be presented with an overview of the test’s class statistics
 - 26.2.3.1. Average is displayed
 - 26.2.3.2. Median is displayed
 - 26.2.3.3. Mode is displayed
 - 26.2.3.4. Max is displayed
 - 26.2.3.5. Min is displayed

- 26.2.4. A bar chart showing the student's performance is displayed
- 26.2.5. A line chart showing the student's performance is displayed
- 26.2.6. A scrollable list of the children in the class is displayed. For each children,
 - 26.2.6.1. Name is displayed
 - 26.2.6.2. Score is displayed
 - 26.2.6.3. A slider representing the score against the total is displayed
- 26.3. Teachers will be able to click the "add new test" button to upload a new test scores
 - 26.3.1. New Test is displayed
 - 26.3.2. Class is displayed
 - 26.3.3. The field "Test Name" is displayed, allowing the teacher to type in the name of the test
 - 26.3.4. A scrollable list of children in the class is displayed. For each children,
 - 26.3.4.1. Name is displayed
 - 26.3.4.2. Score is displayed through a numeric value that can be edited
 - 26.3.4.3. Score is displayed through a slider that can be dragged
 - 26.3.5. Teachers will be able to click the "Save Test" button to upload the test scores onto the database for parents to view

Child:

27. Child mode

- 27.1. Page Overview
 - 27.1.1. Activated when a parent presses the top right button from the child's detail screen
 - 27.1.2. Provides a limited and safe interface specifically designed for children.
 - 27.1.3. Prevents navigation gestures to exit the screen unintentionally.
- 27.2. Homework Display
 - 27.2.1. A scrollable list of the child's currently assigned homework is shown.
 - 27.2.2. Each homework card displays:
 - 27.2.2.1. Homework Title
 - 27.2.2.2. Description
 - 27.2.2.3. Due Date
- 27.3. Location Tracking
 - 27.3.1. "I'm Leaving Home/School" Button:
 - 27.3.1.1. If location tracking is not already active, this button starts real-time foreground tracking.
 - 27.3.1.2. Triggers location updates to be sent to the backend.
 - 27.3.2. If tracking is currently active, display a visible banner
- 27.4. Exit Child Mode
 - 27.4.1. "Exit Child Mode" Button:
 - 27.4.1.1. Opens a modal requiring the user to enter a 4-digit PIN (exit code).
 - 27.4.1.2. PIN is validated against the one set by the parent.
 - 27.4.1.3. If correct:

- 27.4.1.3.1. Child Mode ends.
- 27.4.1.3.2. Location tracking is stopped.
- 27.4.1.3.3. User is navigated back to the parent view.
- 27.4.1.4. If incorrect:
 - 27.4.1.4.1. An error alert is shown: "Incorrect Code"
- 27.5. UX Safeguards
 - 27.5.1. Navigation gestures are disabled while Child Mode is active to prevent accidental exit.
 - 27.5.2. Child mode password is fetched securely from the backend upon entry.

Non-Functional Requirements

1. Usability

- 1.1. The application shall provide a touch-friendly user interface for mobile devices.
- 1.2. The application shall provide mobile-friendly layouts for all user roles.
- 1.3. Each screen shall follow a consistent layout structure.
- 1.4. The application shall allow new users to use core features without prior training.
- 1.5. The system shall display error messages in simple and understandable language.
- 1.6. The application shall adapt to different screen sizes including phones and tablets.
- 1.7. The application shall ensure a minimum color contrast ratio of 4.5:1 for text.
- 1.8. The application shall provide a logical and accessible navigation order.
- 1.9. All non-decorative images and icons shall include alternative text.

2. Reliability

- 2.1. The system shall achieve an uptime of at least 99.9% during operational hours.
- 2.2. The system shall restore full functionality within 5 minutes following an unplanned system reboot or mobile OS update.
- 2.3. The system shall deliver automated notifications for check-ins and check-outs with a success rate of at least 98%.
- 2.4. The application shall continue operating with reduced functionality when a non-critical component fails.

3. Performance

- 3.1. Each screen transition shall occur within 2 seconds under normal network conditions.
- 3.2. Each chat message shall be sent and received within 2 seconds.
- 3.3. Each location update shall be reflected in the system within 10 seconds.
- 3.4. The application shall render the map screen with all markers within 3 seconds.
- 3.5. Each profile image shall load within 2 seconds.
- 3.6. Each backend API request shall return a response within 1 second.

4. Security

- 4.1. All user data shall be transmitted over HTTPS.
- 4.2. The system shall enforce role-based access control for parent, teacher, and child roles.

- 4.3. Each user shall authenticate using Firebase Authentication.
- 4.4. The system shall require passwords with alphanumeric characters.
- 4.5. The application shall store child location data securely using Firestore security rules.
- 4.6. The system shall verify all API requests with Firebase authentication tokens.
- 4.7. The system shall avoid collecting unnecessary personal data.
- 4.8. The application shall comply with relevant data protection regulations such as GDPR and PDPA.

5. Maintainability

- 5.1. The frontend codebase shall follow component-based structure with reusable UI components.
- 5.2. The backend codebase shall follow modular design using FastAPI routers.
- 5.3. The project shall use Git for source control.
- 5.4. All critical functions shall include unit tests.
- 5.5. The codebase shall include inline comments for complex logic.

6. Scalability

- 6.1. The application shall support up to 500 concurrent users.
- 6.2. The system shall support growing data by using subcollections and indexes in Firestore.
- 6.3. The application shall support deployment to cloud platforms such as Firebase Hosting or Render.
- 6.4. The system shall allow adding new user roles or modules without impacting existing features.

7. Compatibility

- 7.1. The application shall be compatible with major mobile operating systems, including iOS and Android.
- 7.2. The application shall support small to large screen sizes, including phones and tablets.
- 7.3. The application shall use platform-native components that behave consistently across iOS and Android.

Data Dictionary

Term	Definition
Relevant Authorities	Includes emergency services such as police and ambulance, as well as school administrators and teachers who receive notifications in cases of emergency or unauthorized location changes.

Check-In/Check-Out	A system-generated confirmation when a child reaches or leaves a designated location (e.g., school, child care center). Notifications are sent to parents and relevant authorities.
Real-Time Tracking	A feature that continuously monitors the child's location in small intervals, ensuring they remain in approved locations and generating alerts if they deviate. Uses GPS-based location verification.
User Interface (UI)	The interactive and visual components of the application, used by parents, teachers, and administrators to navigate features such as messaging, tracking, and attendance.
Notifications	Alerts or updates sent to users regarding child check-ins/check-outs, emergency alerts, homework, announcements, or system updates. Notifications must be delivered within one minute.
Emergency Contact Integration	A system allowing children to quickly contact parents, emergency services, or school staff in case of distress. Dedicated buttons enable instant police, ambulance, or parental contact within 10 to 30 seconds.
GPS-Based Location Verification	A feature that confirms whether the child is within a designated safe zone or an expected route, ensuring real-time tracking reliability.
Digital Consent Forms	A secure platform for parents to sign and submit digital forms related to school activities, medical authorizations, and field trips. Automated reminders are sent for pending submissions.

Secure Communication	A one-on-one encrypted messaging system between parents and teachers. Group messaging is not allowed to maintain privacy and security. Teachers can send group announcements instead.
Attendance Tracking	A feature allowing teachers and parents to monitor a child's school presence, with automated check-in notifications and the ability to manually update attendance if GPS tracking fails.
Announcement System	A structured communication system where teachers and school staff can send group updates regarding school events, safety alerts, and assignments. Parents can filter, search, and mark announcements as read.
Emergency Alerts	Critical notifications sent in cases of injury, unauthorized location changes, or other emergencies. Measured in seconds.
Child Mode	A restricted version of the app for child users, allowing them to view homework, use emergency call buttons, and interact with essential features under parental control. To exit Child Mode, a security PIN is required.
Taskbar Navigation	A persistent bottom menu that allows users to access key pages: Home, Children, Chat, School, and Profile (for parents) or Home, Classes, Chat, School, and Profile (for teachers).
Locale Settings	User-configurable preferences for language and region-based settings. The app automatically detects the preferred language based on device settings.
Uptime	The percentage of time the system remains operational.

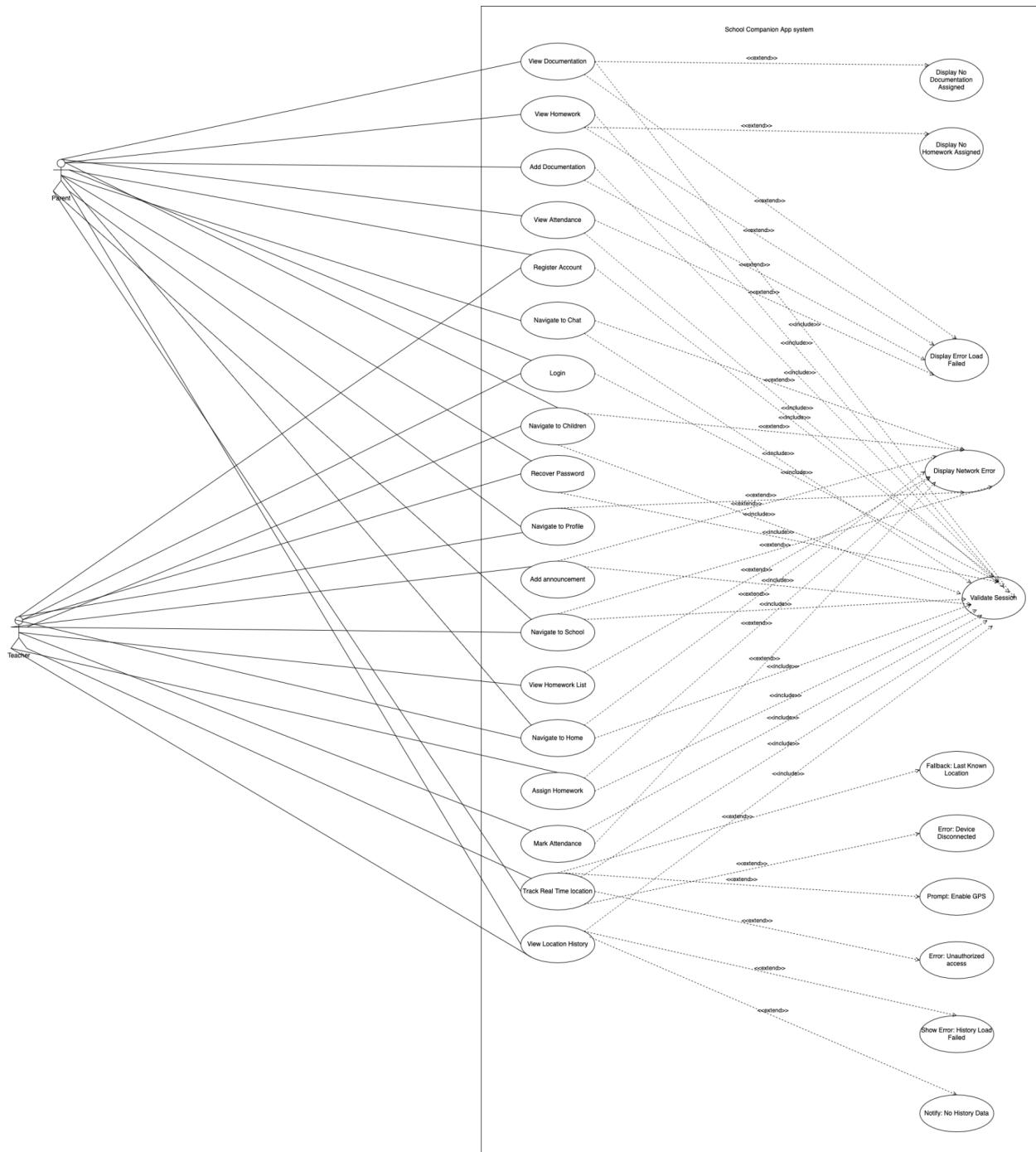
System Reboot	The process of restarting the application or server due to an unplanned failure or update.
Automated Notifications	System-generated alerts for check-ins, check-outs, announcements, and emergency events.
Check-in/Check-out Request	A user action logging a child's arrival or departure in the system, triggering automatic notifications to parents and teachers.
SQL (Structured Query Language)	The language used to manage and query the relational database of the application.
Third-Party Integration	The ability to connect the system with external applications such as school management platforms, emergency contact services, Google Maps, and weather APIs.
User	Any individual using the application, including parents, teachers, administrators, and emergency responders.
Usability	The app provides a touch-friendly user interface, multi-language support, tooltips, and accessibility features such as screen readers and high-contrast mode.
Reliability	The amount of time the system remains fully functional with automated notifications.
Performance	The speed at which the system processes check-in and check-out requests, verify GPS-based location and deliver notifications without compromising resource usage.

Supportability	The system must support seamless updates, minimal downtime for feature enhancements, database compatibility with SQL, and integration with emergency and school management services.
Scalability	The system's capability to handle a huge amount of concurrent users, maintain performance benchmarks as more childcare centers are integrated, and ensure a modular architecture for future feature expansion.
MOE	Ministry of Education Singapore
Four national languages of Singapore	English, Chinese, Malay, Tamil

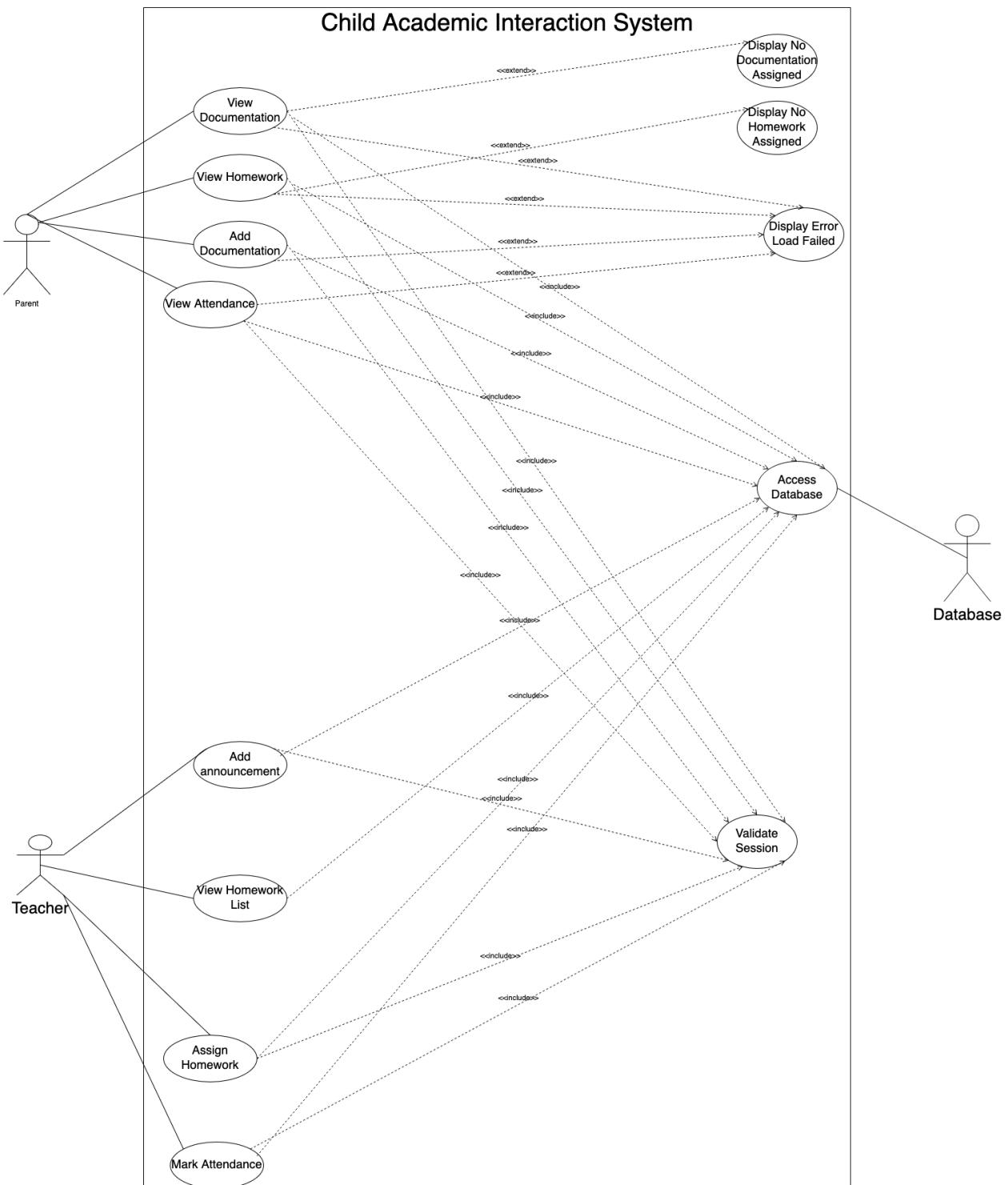
Use Case Diagrams

For ease of readability, we have split our use case diagrams into smaller sub components to represent the smaller interactions

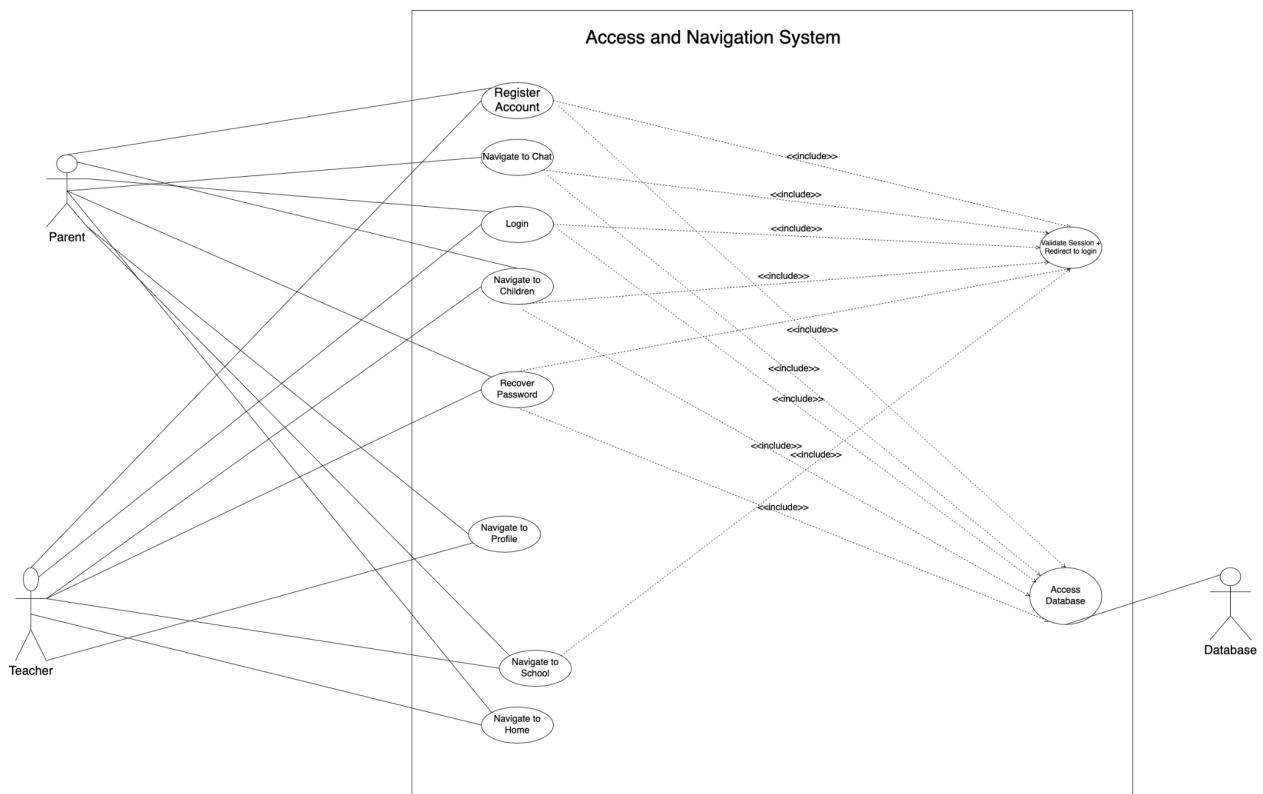
Use Case Diagram 1 (Overall)



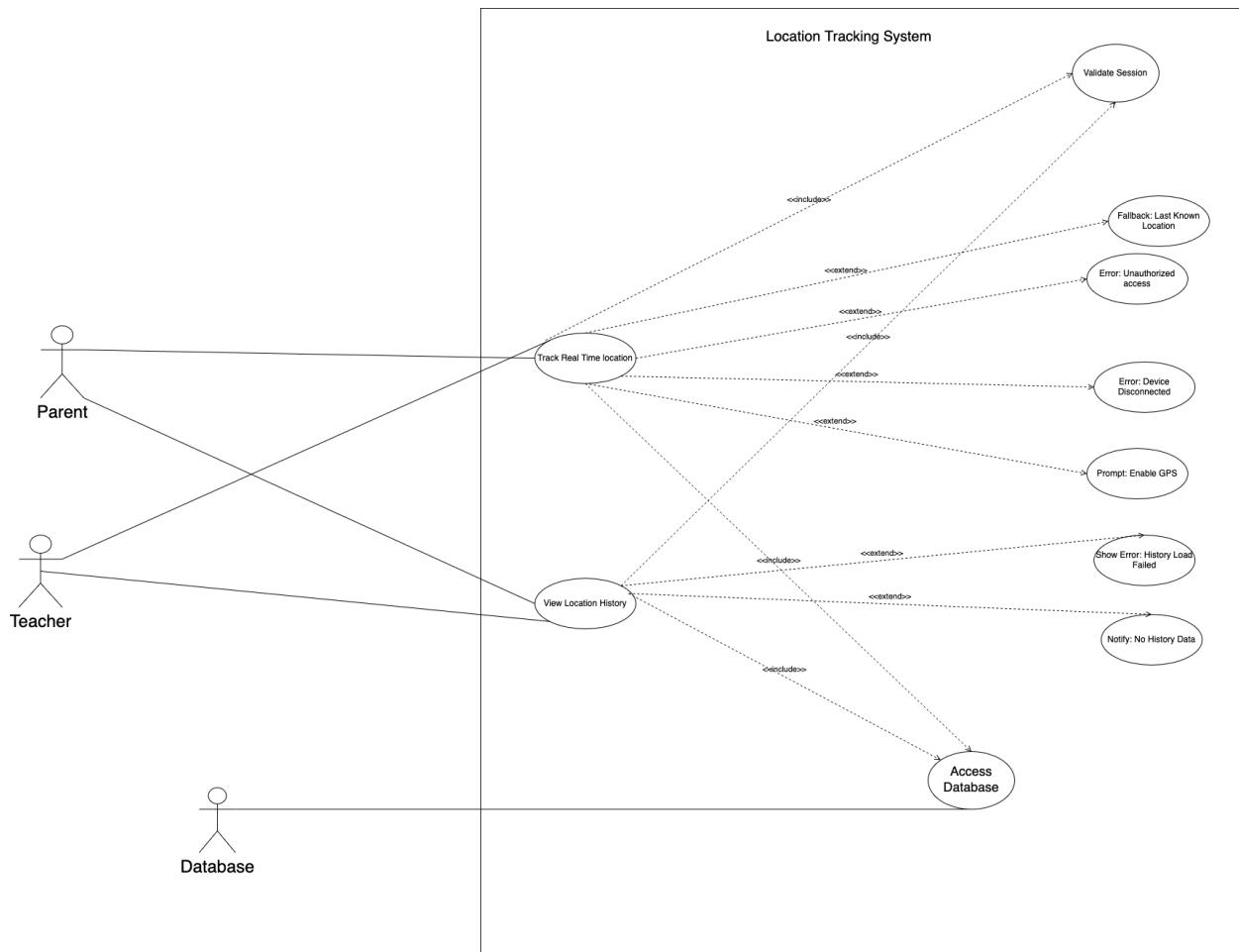
Use Case Diagram 2 (Child Academic Interaction System)



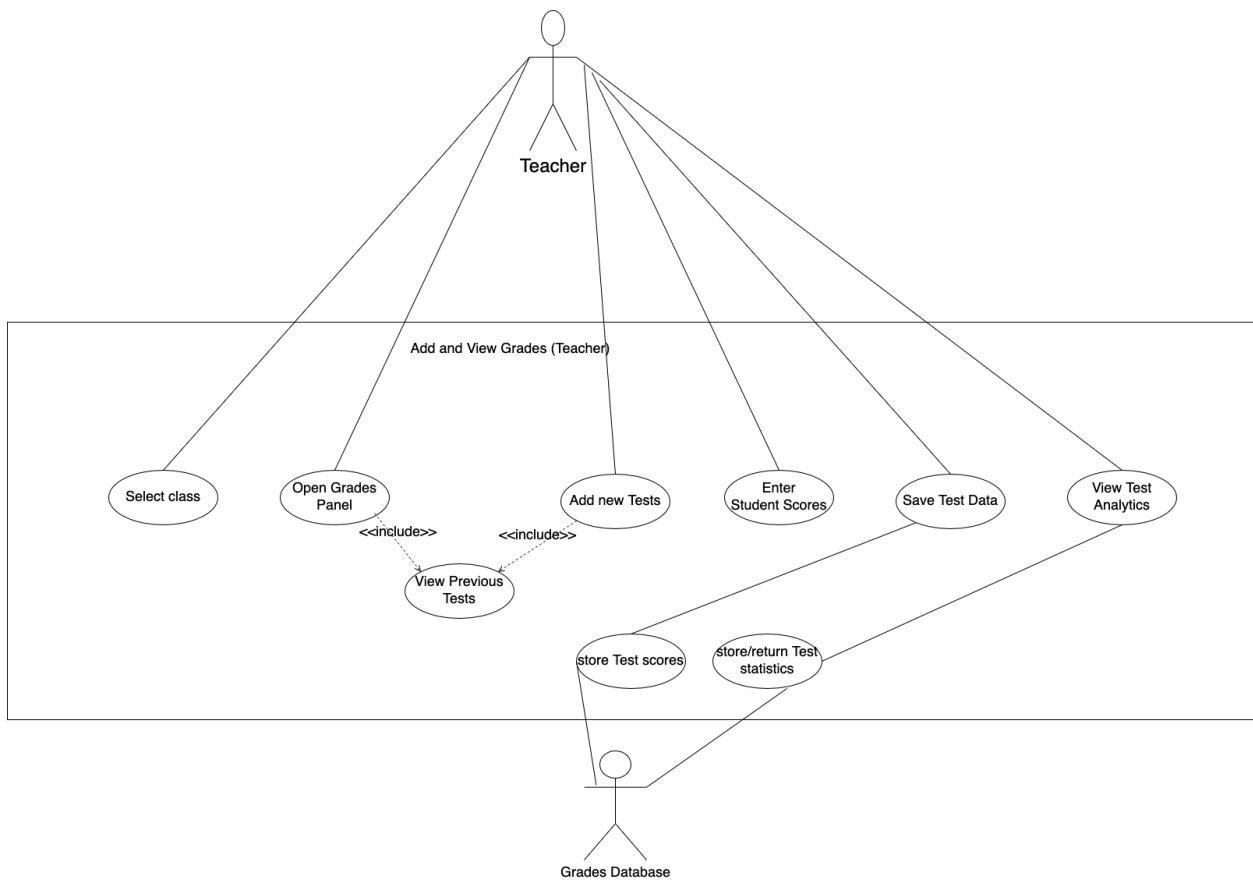
Use Case Diagram 3 (Access and Navigation System)



Use Case Diagram 4 (Location Tracking System)



Use Case Diagram 5 (Grading System)



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Registration System		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Kumar Advaith
Date Created:	05/02/2025	Date Last Updated:	05/02/2025

Actor:	Parent, Teacher, Database
Description:	Registration System: Allows new users (parents or teachers) to register for an account in the ChildConnect system. The registration requires personal details and password confirmation for security.
Preconditions:	None
Postconditions:	<ol style="list-style-type: none"> 1. The user is registered into the database 2. The user is logged in
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Get Started" button on the landing page of the app. 2. The user will be taken to the registration page where the full name, email and password will be entered. Password is confirmed. 3. The user will finalise the details and press the "Register" button. 4. The app verifies the registration information with the database, ensuring the password and confirm password fields match 5. If the registration information is verified, the database will save the user information, and the app proceeds to display the home page.
Alternative Flows:	<p>AF-S4: If the registration information is invalid</p> <ol style="list-style-type: none"> 1. The app displays the message "Invalid registration details, Please try again" 2. The app returns to step 2 <p>AF-S5: If the password and confirm password do not match:</p> <ol style="list-style-type: none"> 1. The app displays the message "Passwords do not match, Please try again." 2. The app returns to step 2.
Exceptions:	<p>EX-1: If the registration information is already in the database</p> <ol style="list-style-type: none"> 1. The app displays the message "User already registered. Please login instead". 2. The app takes the user to the login page instead.

	EX-2: The user presses the “Sign In” button on the registration page 1. The app takes the user to the login page instead.
Includes:	Login System, Database
Special Requirements:	The system must verify that the "Password" and "Confirm Password" fields match before allowing registration.
Assumptions:	Referring to just the term “user” can mean both Parent user and Teacher user.
Notes and Issues:	None

Use Case ID:	2		
Use Case Name:	Login System		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Kumar Advaith
Date Created:	05/02/2025	Date Last Updated:	05/02/2025

Actor:	Parent user, Teacher user, Database
Description:	Login System
Preconditions:	None
Postconditions:	The user is logged in
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the “Sign in” button on the landing page 2. The user will be taken to the login page where the email and password will be entered. 3. The user will enter the details and press the “Login” button. 4. The app will verify the login information with the database. 5. If the login information is verified, the app will proceed to display the homepage.
Alternative Flows:	<p>AF-S4: If the login information is invalid</p> <ol style="list-style-type: none"> 1. The app displays the message “Invalid login details, Please try again” 2. The app returns to step 2
Exceptions:	<p>EX-1: The user presses the “Forget Password” button on the login page</p> <ol style="list-style-type: none"> 1. The app takes the user to the forget password page instead.
Includes:	Database
Special Requirements:	The system must support account lockout mechanisms for multiple failed login attempts.
Assumptions:	Referring to just the term “user” can mean both Parent user and Teacher user.
Notes and Issues:	None

Use Case ID:	3		
Use Case Name:	Forget Password System		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Kumar Advaith
Date Created:	05/02/2025	Date Last Updated:	05/02/2025

Actor:	Parent user, Teacher user, Database
Description:	Allows users who forgot their password to reset it through an email verification process.
Preconditions:	The user is in the login page
Postconditions:	<ul style="list-style-type: none"> 1. The user receives a password reset email. 2. The user can reset their password using the provided link.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the “Forget Password” button on the login page. 2. The user will be taken to a forget password page where the email can be entered. 3. The user enters the email. 4. The app will verify if it is a valid email with the database. 5. The database will send a forget password email to the user’s email address. 6. The app will display the notification “A reset email has been sent”.
Alternative Flows:	<p>AF-S4: If the email information is invalid</p> <ol style="list-style-type: none"> 1. The app displays the message “Invalid email, Please try again” 2. The app returns to step 2
Exceptions:	None
Includes:	Database
Special Requirements:	The reset email link should be valid for only a limited period for security.
Assumptions:	<p>Referring to just the term “user” can mean both Parent user and Teacher user.</p> <p>Users will check their registered email for the reset link.</p>
Notes and Issues:	None

Use Case ID:	4
Use Case Name:	Navigation Bar (Parent)

Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Sivaguruanthan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	08/04/2025

Actor:	Parent
Description:	Provides a navigation interface for parents to easily switch between different sections of the application, including HomeUI, ChildrenUI, ChatUI, and ProfileUI.
Preconditions:	Parent User needs to logged into the account
Postconditions:	The user is redirected to the selected section.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. If users is not in the homeUI page and they click on the home button, they will be navigated to the homeUI page 2. If users are not in the childrenUI page and they click on the children button, they will be navigated to the childrenUI page 3. If users are not in the chatUI page and they click on the chat button, they will be navigated to their chatUI page 4. If users are not in the profileUI page and they click on the profile button, they will be navigated to the profileUI button
Alternative Flows:	None
Exceptions:	None
Includes:	Login System, Database
Special Requirements:	The navigation system must provide smooth transitions between screens.
Assumptions:	The term "users" here can only mean parents
Notes and Issues:	None

Use Case ID:	5		
Use Case Name:	Navigation Bar (Teacher)		
Created By:	Tan Yi Jun	Last Updated By:	Tan Yi Jun
Date Created:	05/02/2025	Date Last Updated:	05/02/2025

Actor:	
Description:	Provides a navigation interface for teachers to easily switch between different sections of the application, including HomeUI, ClassUI, ChatUI, and ProfileUI.
Preconditions:	Teacher User needs to logged into the account
Postconditions:	The user is redirected to the selected section.

Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. If users are not in the homeUI page and they click on the home button, they will be navigated to the homeUI page 2. If users are not in the classUI page and they click on the class button, they will be navigated to the classUI page 3. If users are not in the chat page and they click on the chatUI button, they will be navigated to their chatUI page 4. If users are not in the profileUI page and they click on the profile button, they will be navigated to the profileUI button
Alternative Flows:	None
Exceptions:	None
Includes:	Login System, Database
Special Requirements:	The navigation system must provide smooth transitions between screens.
Assumptions:	The term "users" here refers only to teachers.
Notes and Issues:	

Use Case ID:	6		
Use Case Name:	View Homework (Parent)		
Created By:	Tan Yi Jun	Last Updated By:	Sivaguruathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Parent, Database
Description:	Allows user to view any homework that is assigned to their child by their teacher
Preconditions:	<ol style="list-style-type: none"> 1) The user must be logged in as a parent. 2) The user must be in the childrenUI page
Postconditions:	User successfully views details of the child's assigned homework
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user will click on the name of the child 2. The user will enter the ChildProfileUI 3. The user will click on the "Homework List" button 4. The user will be navigated to the HomeworkListUI

	<p>5. The Homework Controller retrieves homework data from the database</p> <p>6. The user can view the details of the homework assigned for that particular child</p>
Alternative Flows:	AF-S3: System will display the message: "No homework assigned for this class"
Exceptions:	EX-1: The system displays an error message: "Failed to load homework. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> The homework should be sorted by the most recent date first.
Assumptions:	Homework is relevant and updated frequently.
Notes and Issues:	<p>ChildrenUI is different from ChildUI:</p> <ul style="list-style-type: none"> ChildrenUI is a brief list of all the children ChildUI is the specifics for a particular child

Use Case ID:	7		
Use Case Name:	View documentation (Parent)		
Created By:	Tan Yi Jun	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Parent, Database
Description:	Allows user to view any documentation that they have previously uploaded
Preconditions:	<ol style="list-style-type: none"> 1) The user must be logged in as a parent. 2) The user must be in the childUI
Postconditions:	User successfully views details of the documentation history
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user will click on the name of the child 2. The user will enter ChildProfileUI 3. The user will click on the "Documentation" button 4. The user will be navigated to the DocumentationUI 5. The system retrieves documentation data from the database 6. The user can view the details of the documentation that they have previously uploaded
Alternative Flows:	AF-S3: System will display the message: "No documentation found for this child"
Exceptions:	EX-1: The system displays an error message: "Failed to load documentation. Please try again later."
Includes:	Database

Special Requirements:	<ul style="list-style-type: none"> The documentation should be sorted by the most recent date first.
Assumptions:	Documentations are relevant and updated frequently.
Notes and Issues:	<p>ChildrenUI is different from ChildUI:</p> <ul style="list-style-type: none"> - ChildrenUI is a brief list of all the children - ChildUI is the specifics for a particular child

Use Case ID:	8		
Use Case Name:	Add documentation (Parent)		
Created By:	Tan Yi Jun	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Parent, Database
Description:	Allows user to add any new documentations into the system
Preconditions:	3) The user must be logged in as a parent. 4) The user must be in the childUI
Postconditions:	User gets the “documentation has been added” message
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user will click on the name of the child 2. The user will enter the ChildProfileUI 3. The user will click on the “Documentation” button 4. The user will be navigated to the DocumentationUI 5. The user clicks on the add documentation button 6. The user will be navigated to the DocumentationFormUI 7. The user keys in the information or attachments they want to add in the UI 8. The system adds documentation data to the database 9. It displays a message to state that the documentation has been added
Alternative Flows:	AF- 5: If the user tries to add documentation without adding any information, there will be an error
Exceptions:	EX-1: If the documentation is not saved in the database, there will be an error message: “Failed to upload the documents”
Includes:	Database
Special Requirements:	1. File types allowed to upload include PDF, JPG and PNG
Assumptions:	Documentations are relevant and updated frequently.
Notes and Issues:	<p>ChildrenUI is different from ChildUI:</p> <ul style="list-style-type: none"> - ChildrenUI is a brief list of all the children - ChildUI is the specifics for a particular child

Use Case ID:	9		
Use Case Name:	View attendance (Parent)		
Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Wee Zi Hao (Huang Zihao)
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Parent, Database
Description:	Allows user to view attendance of a child
Preconditions:	1) The user must be logged in as a parent. 2) The user must be in the childUI
Postconditions:	User successfully views details of the child's attendance
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user will click on the name of the child 2. The user will enter ChildProfileUI 3. The user will click on the "Attendance Record" button 4. The user will be navigated to the AttendanceUI 5. The system retrieves attendance data from the database 6. The user views the attendance records displayed in calendar format
Alternative Flows:	NIL
Exceptions:	EX-1: The system displays an error message: "Failed to load attendance. Please try again later."
Includes:	Database
Special Requirements:	The attendance is showed in the form of a calendar: - Dates attended are marked clearly with a green circle - Dates absent are marked clearly with a red circle - Date where attendance is not taken have neither
Assumptions:	Attendance records are updated regularly and accurately
Notes and Issues:	ChildrenUI is different from ChildUI: - ChildrenUI is a brief list of all the children - ChildUI is the specifics for a particular child

Use Case ID:	10		
Use Case Name:	View School (Parent)		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Parent, Database
--------	------------------

Description:	Allows users to view information about the school
Preconditions:	1) The user must be logged in as a parent 2) The user must be in the ChildUI
Postconditions:	User successfully views the school information
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user will click on the name of the child 2. The user will enter ChildProfileUI 3. The user will click on the button with the school name 4. The user will be navigated to the SchoolInformationUI 5. The system retrieves school data from the database 6. The user views information about the school
Alternative Flows:	NIL
Exceptions:	EX-1: If the database retrieval fails. The system displays an error message: "Failed to load school information. Please try again later."
Includes:	Database, government API integration
Special Requirements:	The school information should be fetched from the government API
Assumptions:	The government API providing school information is reliable and update regularly
Notes and Issues:	ChildrenUI is different from ChildUI: - ChildrenUI is a brief list of all the children - ChildUI is the specifics for a particular child

Use Case ID:	11		
Use Case Name:	Child Mode (Parent)		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	31/03/2025

Actor:	Child, Parent, Database
Description:	Enter Child Mode
Preconditions:	1) The user is logged in as a parent 2) The user must be in the ChildUI
Postconditions:	The app goes into Child Mode
Priority:	High
Frequency of Use:	High
Flow of Events:	<u>Entry:</u> 1. The user will click on the name of the child 2. The user will click the child mode button on the top right of the app 3. The app is not in child mode

	<p><u>Exit:</u></p> <ol style="list-style-type: none"> 1. The user will click on the exit child mode button 2. The user will be prompted to type the pin number to exit child mode 3. This number will be validated with the backend 4. The user is back to the Child UI if the correct pin numbers are keyed in
Alternative Flows:	<p>AF exit -3: If the user keys in the wrong pin number</p> <ol style="list-style-type: none"> 1. There will be an error message: "The exit code entered is incorrect" 2. The user will need to try keying in the exit code again
Exceptions:	NIL
Includes:	Login System, Database
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	12		
Use Case Name:	Mark Attendance (Teacher)		
Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Teacher, Database
Description:	This user can mark attendance for their classes by updating the attendance status of students as present or absent
Preconditions:	<ul style="list-style-type: none"> • The user must be logged in as a teacher. • The user must be in the classesUI page
Postconditions:	Attendance records are updated in the database.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user will click in the name of the classes that he needs to take the attendance for 2. The user can change the status of the child to present if the child is present 3. This change in status will be updated in the database
Alternative Flows:	AF-S2: The user can change the status of the child to absent if they accidentally mark the child as present
Exceptions:	<p>EX-1: If the database update fails:</p> <ol style="list-style-type: none"> 1. The system displays an error message: "Failed to save attendance. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> • The system should allow attendance to be modified from present to absent and the other way round too • Attendance data must be saved securely and accurately.
Assumptions:	None

Notes and Issues:	None
-------------------	------

Use Case ID:	13		
Use Case Name:	Assign Homework (Teacher)		
Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Teacher, Database
Description:	Allows the user to assign homework to their students
Preconditions:	<ul style="list-style-type: none"> • The user must be logged in as a teacher. • The user must be in the classesUI page
Postconditions:	The Homework detail is updated in the database and visible in the HomeworkListUI
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> 1. The user will click in the name of the class that he needs to assign the homework 2. The user will click on the assign homework button to navigate to the AssignHomeworkUI 3. The user will key in the details of homework 4. The user will click the submit homework button 5. The homework details will be added in the database 6. This homework detail will be reflected in the HomeworkListUI
Alternative Flows:	<p>AF-S4: If the user fails to fill up all the homework details:</p> <ol style="list-style-type: none"> 1. System will display an error message:"Missing Fields, please fill in all fields."
Exceptions:	<p>EX-1: If the database fails to save the homework:</p> <ol style="list-style-type: none"> 1. System will display an error message: "Failed to save the homework. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> • Homework data must be securely stored and reliably synced between teacher and student views
Assumptions:	None
Notes and Issues:	None

Use Case ID:	14		
Use Case Name:	Add Announcement (Teacher)		
Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Sivagurunathan Keerthivasan

Date Created:	05/02/2025	Date Last Updated:	03/04/2025
---------------	------------	--------------------	------------

Actor:	Teacher
Description:	Allows the user to add announcements for the parents/students
Preconditions:	<ul style="list-style-type: none"> • The user must be logged in as a teacher. • The user must be in the classesUI page
Postconditions:	The announcement detail is updated in the database and visible in the AnnouncementListUI
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> 1. The user will click in the name of the class that he wants to send the assignment out to 2. The user will click on the add announcement button to navigate to the AddAnnouncementUI 3. The user will key in the details of the announcement 4. The user will click the submit announcement button 5. The announcement details will be added in the database 6. This announcement detail will be reflected in the AnnouncementListUI
Alternative Flows:	AF-S4: If the user fails to fill up all the announcement details: 2. System will display an error message:"Missing Fields, please fill in all fields."
Exceptions:	EX-1: If the database fails to save the announcement: 2. System will display an error message: "Failed to save the announcement. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> • Announcement data must be securely stored and reliably synced between teacher and student views
Assumptions:	None
Notes and Issues:	None

Use Case ID:	15		
Use Case Name:	View Homework List (Teacher)		
Created By:	Sivagurunathan Keerthivasan	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Teacher, Database
Description:	Allows users to view the homework that they have already assigned and change the status of the homework from open to closed

Preconditions:	<ul style="list-style-type: none"> The user must be logged in as a teacher. The user must be in the classesUI page
Postconditions:	The homework list detail is visible in the HomeworkListUI
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> The user will click in the name of the class that he wants to check the homework list for The user will click on the homework list button to navigate to the HomeworkListUI The HomeworkListUI will fetch the homework data from the database and reflect it on the UI <p><u>Status:open</u></p> <ol style="list-style-type: none"> If the status for the homework is open, it is visible on the parent's homeworkListUI <p><u>Status: closed</u></p> <ol style="list-style-type: none"> If the status for the homework is closed, it is not visible on the parent's homeworkListUI
Alternative Flows:	AF-S2: If there are no homework assigned by the user yet: <ol style="list-style-type: none"> There will be a message to say that there is no homework assigned yet.
Exceptions:	EX-1: If the database retrieval fails: <ol style="list-style-type: none"> The system displays an error message: "Failed to retrieve homework list. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> Homework data must be securely stored and reliably synced between teacher and student views
Assumptions:	None
Notes and Issues:	None

Use Case ID:	16		
Use Case Name:	View Announcement List (Teacher)		
Created By:	Kumar Advaith	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Teacher, Database		
Description:	Allows users to view the announcements that they have already sent out and change the status of the announcement from open to closed		
Preconditions:	<ul style="list-style-type: none"> The user must be logged in as a teacher. The user must be in the classesUI page 		
Postconditions:	The announcement list detail is visible in the AnnouncementListUI		

Priority:	High
Frequency of Use:	Regular
Flow of Events:	<p>1. The user will click in the name of the class that he wants to check the announcement list for</p> <p>2. The user will click on the announcement list button to navigate to the AnnouncementListUI</p> <p>3. The AnnouncementListUI will fetch the announcement data from the database and reflect it on the UI</p> <p><u>Status:open</u></p> <p>2. If the status for the announcement is open, it is visible on the parent's HomepageUI</p> <p><u>Status: closed</u></p> <p>2. If the status for the announcement is closed, it is not visible on the parent's HomepageUI</p>
Alternative Flows:	AF-S2: If there are no announcement assigned by the user yet: 1. There will be a message to say that there is no announcement assigned yet.
Exceptions:	EX-1: If the database retrieval fails: 1. The system displays an error message: "Failed to retrieve announcement list. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> Announcement data must be securely stored and reliably synced between teacher and student views
Assumptions:	None
Notes and Issues:	None

Use Case ID:	17		
Use Case Name:	Track Location		
Created By:	Tan Yi Jun	Last Updated By:	Tan Yi Jun
Date Created:	05/02/2025	Date Last Updated:	05/02/2025

Actor:	Parent, Child, GPS Device Sensor, database		
Description:	Allows parents to track their child's location within the system if permission for the app to use GPS is enabled		
Preconditions:	<p>1. All users must be logged in as a parent.</p> <p><u>Parent:</u></p> <p>1. The user must be in the classesUI page.</p> <p><u>Child:</u></p> <p>1. The user must be in child mode</p>		
Postconditions:	<p>1. The child's location tracking is stopped</p> <p>2. The database ensures no further location is updated in the database</p>		

Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The parent user will set the child user's device in child mode 2. The parent user selects the child user he is tracking in the childUI. 3. Then he will click activate tracking to start tracking the child user's location. 4. The GPS device begins polling the child user's location every 10-16 seconds 5. The backend updates the child's location in real time using the child user's location data. 6. The parent user can view the child user's current location and last updated timestamp along with the school location and their location. 7. The parent user can press the "stop tracking" button to stop the tracking process
Alternative Flows:	<p>AF-S3: If location permissions are not granted, the system displays an alert and tracking cannot be initiated.</p> <p>AF-S3: If tracking is already active, the system prevents duplicate tracking sessions</p> <p>AF-S3: The child user can press the "I have left school" button to start the tracking system on the child user side</p>
Exceptions:	EX-1: If the child's device is offline or unable to provide location updates, the system displays an error message to the parents
Includes:	Database, GPS device sensor
Special Requirements:	<ol style="list-style-type: none"> 1. There must be real-time updates on the map dynamically 2. There must be no tracking of location when the stop tracking button is pressed
Assumptions:	1. The child carries a GPS-enabled device.
Notes and Issues:	None

Use Case ID:	18		
Use Case Name:	Chat		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Wee Zi Hao (Huang Zihao)
Date Created:	04/04/2025	Date Last Updated:	04/04/2025

Actor:	Parent, Teacher
Description:	Parents and Teachers are allowed to communicate with each other through our application. Parents/Teachers are not allowed to directly create a chat with any Parent/Teacher but rather go through our application's interface through the student profile.

Preconditions:	The user is logged in as a parent or teacher.
Postconditions:	Messages are sent from our application to our database successfully. Messages are fetched from our database to be read on the application successfully.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects a child's profile and navigates to the child's parent's/teacher's information. 2. The user selects the "chat" button on the user profile of the parent/teacher he or she would like to communicate with. 3. The application brings the user to the private chat between the user and the parent/teacher. 4. The user types the message that he or she would like to send and presses the send button when done. 5. The message gets sent to the backend and appears on the chat log for both users in the chat
Alternative Flows:	<p>AF-S3</p> <ol style="list-style-type: none"> 1. If an existing chat does not exist between the two individuals yet, a new one will be created. <p>AF-S4</p> <ol style="list-style-type: none"> 1. A user can press the photo icon to upload an image as part of the message. 2. The image will be uploaded as part of the message.
Exceptions:	None
Includes:	Database
Special Requirements:	<p>The system supports pull to refresh so that messages can be seen immediately</p> <p>The system supports</p> <ol style="list-style-type: none"> 1. A single grey tick to represent an unread message 2. Two grey ticks to represent a read message <p>The system supports displaying the "New" tag when a chat is unread.</p>
Assumptions:	The user you are trying to communicate with has an account on our platform.
Notes and Issues:	None

Use Case ID:	19		
Use Case Name:	Editing profile page		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Parent, Teacher Database		
Description:	Allows a logged-in parent to update their profile information, such as contact number, or profile picture.		

Preconditions:	<ul style="list-style-type: none"> The user must be logged in as a parent or teacher
Postconditions:	The user gets a message saying that his information has been updated
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> The user navigated to the ProfileUI The user clicks on the edit button The user edits his information in one or more fields The user clicks the save button This information is updated in the backend and the changes are reflected in the profileUI The user gets a message saying that his information has been updated
Alternative Flows:	AF-S2: If the user accidentally clicks the edit button, he can click the cancel button on the page where he changes his information
Exceptions:	EX-1: If the database update fails: <ol style="list-style-type: none"> The system displays an error message: "Failed to update profile. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> All personal data must be stored securely and comply with data protection standards
Assumptions:	None
Notes and Issues:	None

Use Case ID:	20		
Use Case Name:	Viewing Homepage (Parent)		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Parent, Database
Description:	Displays any new announcements
Preconditions:	<ul style="list-style-type: none"> The user must be logged in as a Teacher
Postconditions:	All unseen announcements and messages are displayed on the home page
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> The user logs into the system The announcement controller retrieves all announcements that the parent has not viewed yet from the database Home page displays a list of recent announcements
Alternative Flows:	AF-S3: If there are no new announcements <ul style="list-style-type: none"> A message such as "No new announcements at this time" is shown.

Exceptions:	EX-1: If the database retrieval fails: - The system displays an error message: "Failed to retrieve announcements. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> Announcements should be time stamped and shown by recency All data must be securely retrieved and displayed from the database
Assumptions:	None
Notes and Issues:	None

Use Case ID:	21		
Use Case Name:	Viewing Homepage (Teacher)		
Created By:	Wee Zi Hao (Huang Zihao)	Last Updated By:	Sivagurunathan Keerthivasan
Date Created:	05/02/2025	Date Last Updated:	03/04/2025

Actor:	Teacher, Database
Description:	Displays any new announcements
Preconditions:	<ul style="list-style-type: none"> The user must be logged in as a parent
Postconditions:	All unseen announcements are displayed on the home page
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> The user logs into the system The message controller and announcement controller retrieves all announcements that the teacher has not viewed yet from the database Home page displays a list of recent announcements
Alternative Flows:	<p>AF-S3: If there are no new announcements</p> <ul style="list-style-type: none"> A message such as "No new announcements or messages at this time" is shown.
Exceptions:	EX-1: If the database retrieval fails: <ul style="list-style-type: none"> The system displays an error message: "Failed to retrieve announcements and messages. Please try again later."
Includes:	Database
Special Requirements:	<ul style="list-style-type: none"> Announcements and messages should be time stamped and shown by recency All data must be securely retrieved and displayed from the database
Assumptions:	None
Notes and Issues:	None

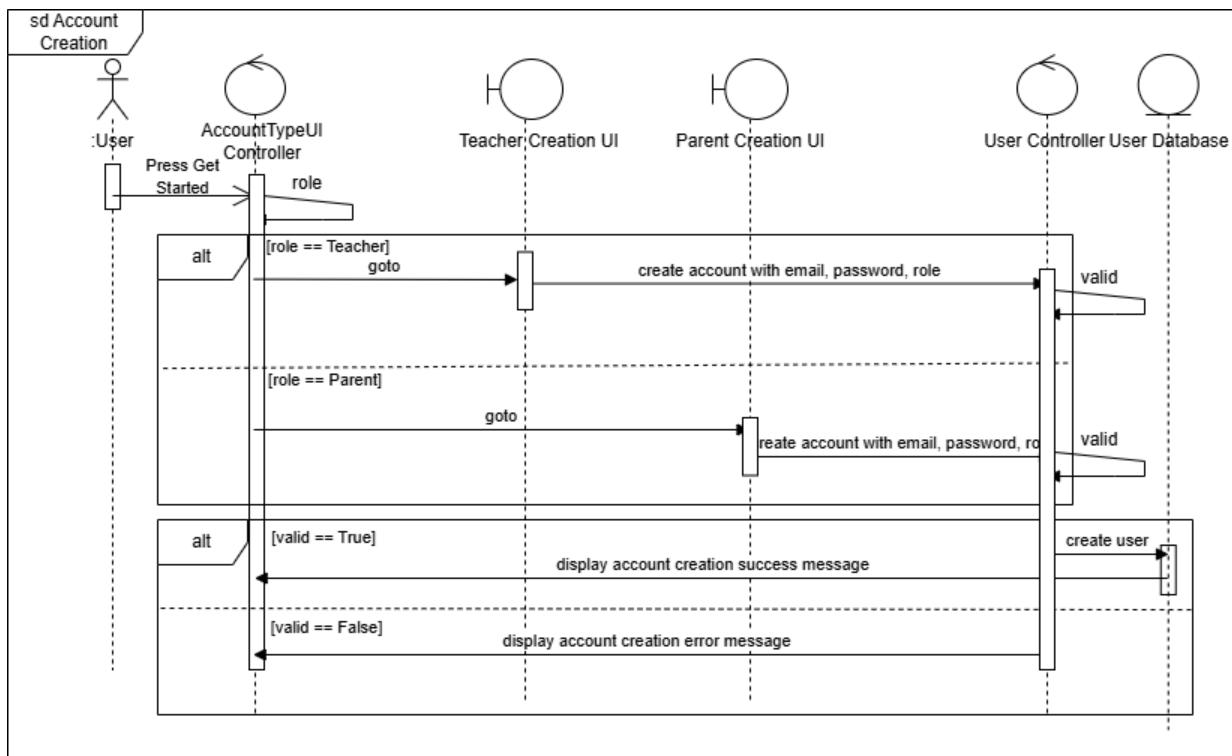
Use Case ID:	22
--------------	----

Use Case Name:	Add & View Grades (Teacher)		
Created By:	Kumar Advaith	Last Updated By:	
Date Created:	12/04/2025	Date Last Updated:	13/04/2025

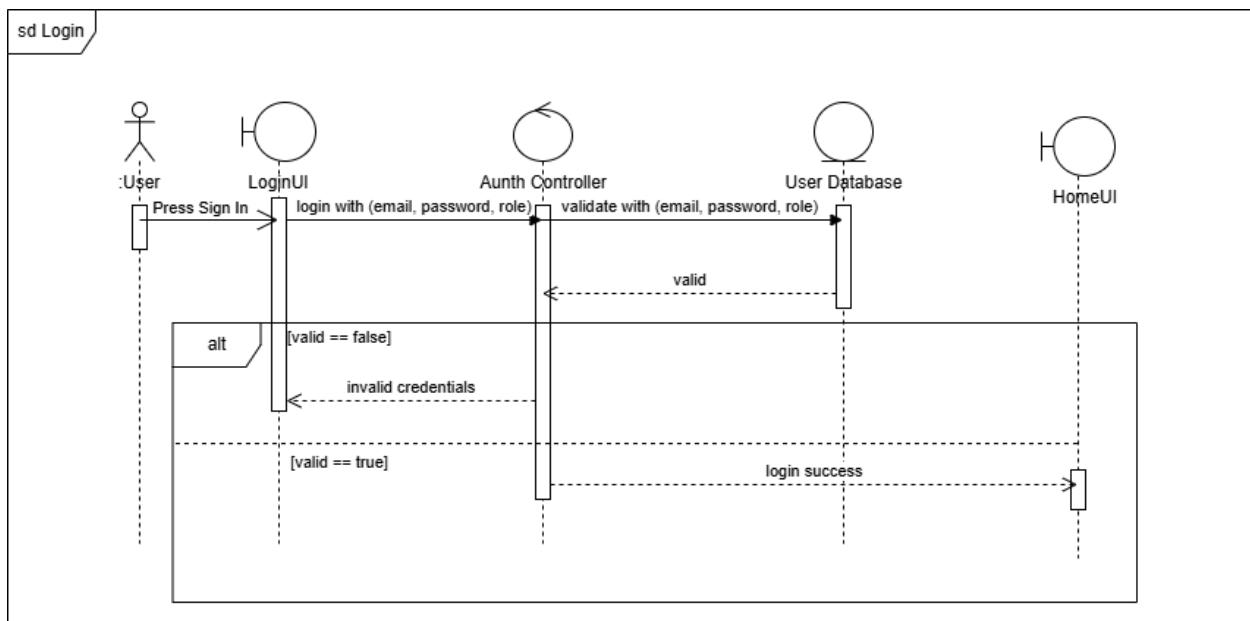
Actor:	Teacher, Grades Database
Description:	Teacher adds scores for a new test and views student performance analytics.
Preconditions:	<ul style="list-style-type: none"> ● Teacher must be logged in. ● A class must be selected.
Postconditions:	<ul style="list-style-type: none"> ● Test is saved with scores. ● Charts are updated with test statistics.
Priority:	High
Frequency of Use:	Regular
Flow of Events:	<ol style="list-style-type: none"> 1. Teacher selects the Classes tab. 2. Clicks on a class (e.g., "1A"). 3. Clicks Grades. 4. Views previous test scores. 5. Clicks Add New Test. 6. Inputs test name and enters scores per student. 7. Clicks Save Test. 8. System stores data and displays success confirmation. 9. Teacher views updated test statistics and charts.
Alternative Flows:	<p>AF-1: Invalid Score Entry - If scores are non-numeric or exceed 100, the system blocks save.</p> <p>AF-2: Teacher Cancels Input - Teacher can leave the test creation screen without saving.</p>
Exceptions:	<p>EX-1: Database Failure - A message "Failed to save test scores. Try again later." is shown.</p>
Includes:	Database
Special Requirements:	Score range: 0–100 only. Instant UI feedback on success or failure. Responsive layout (mobile & tablet)
Assumptions:	Teachers already have access to the class roster.
Notes and Issues:	Add analytics export in future versions

Sequence Diagrams

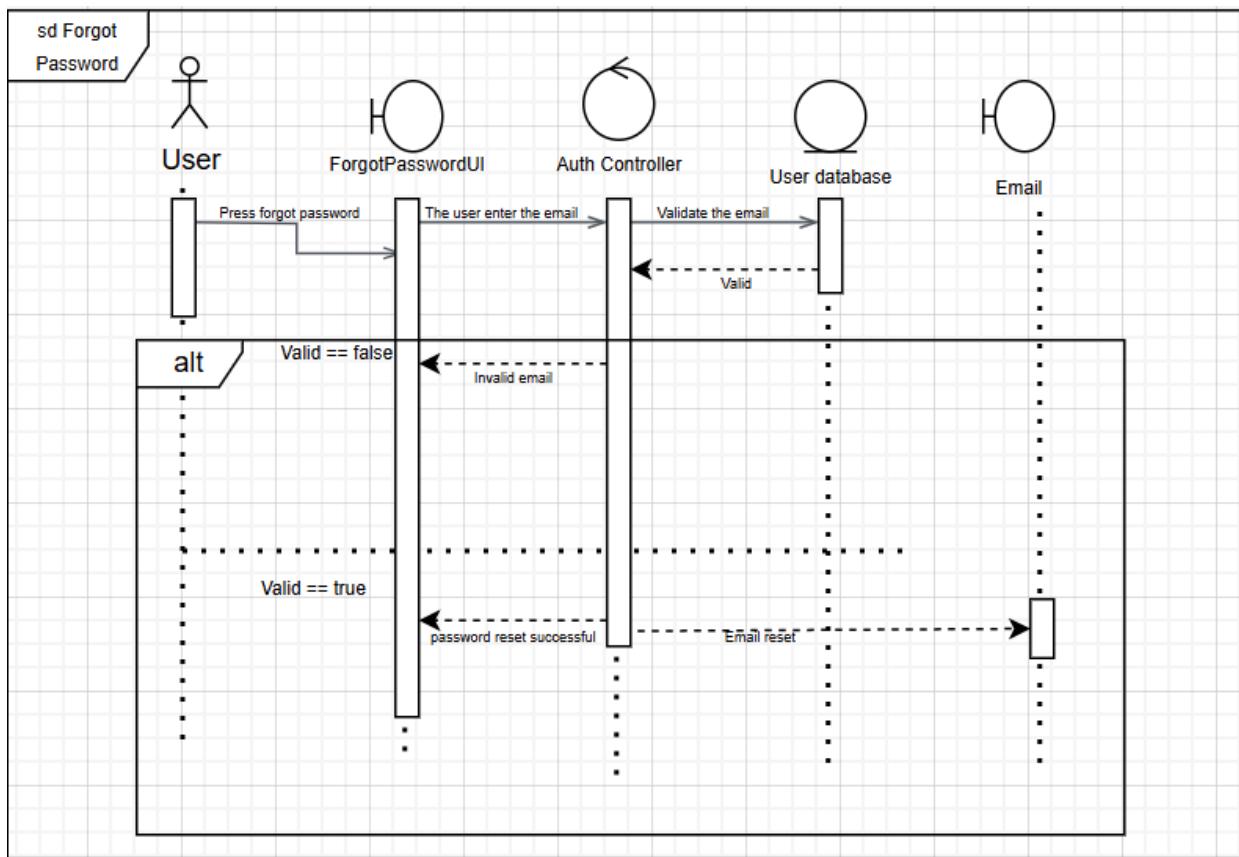
Use Case 1:



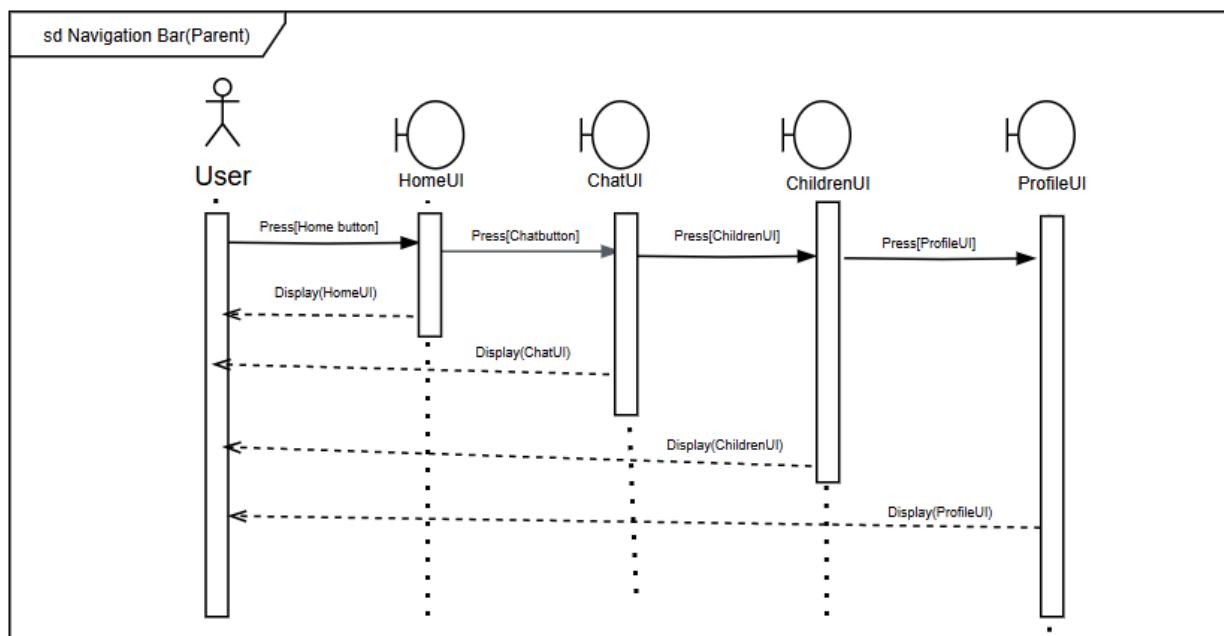
Use Case 2:



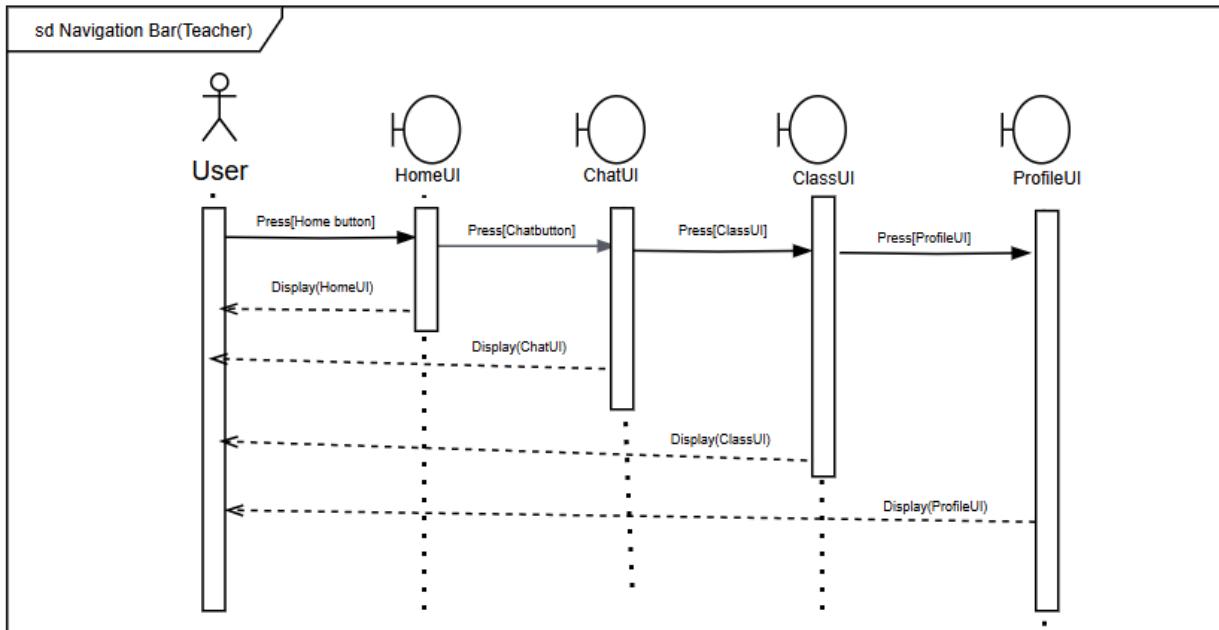
Use Case 3:



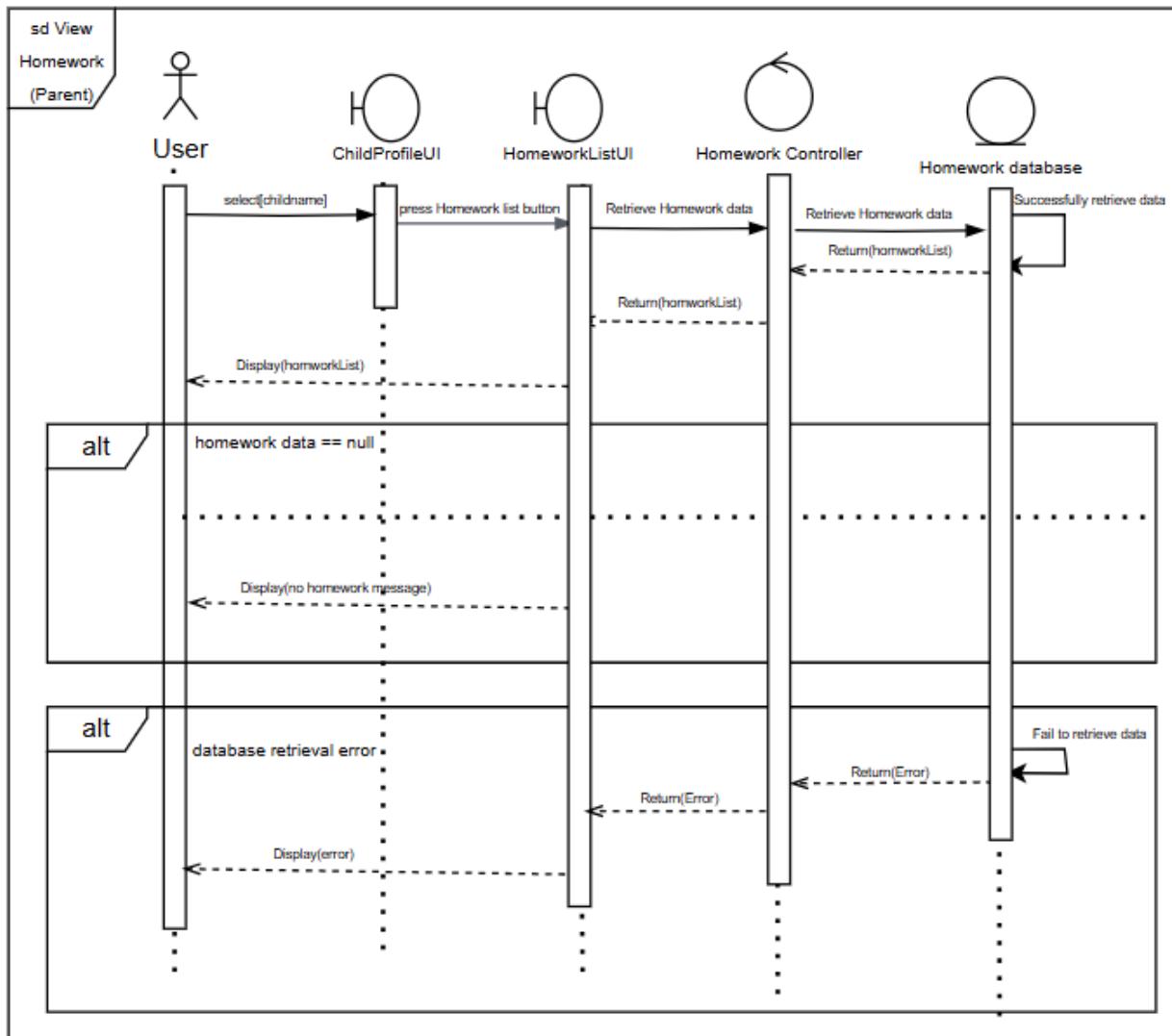
Use Case 4:



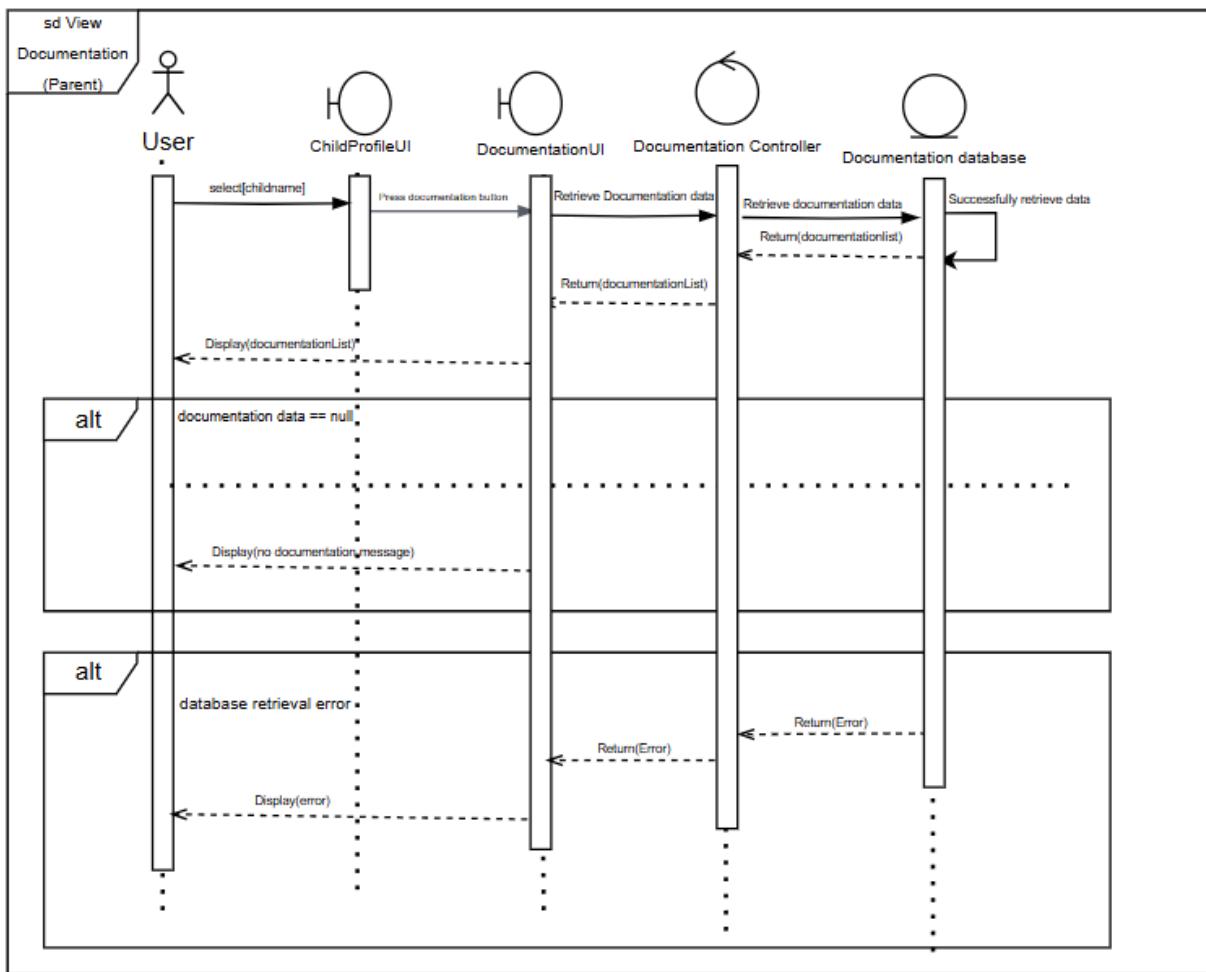
Use Case 5:



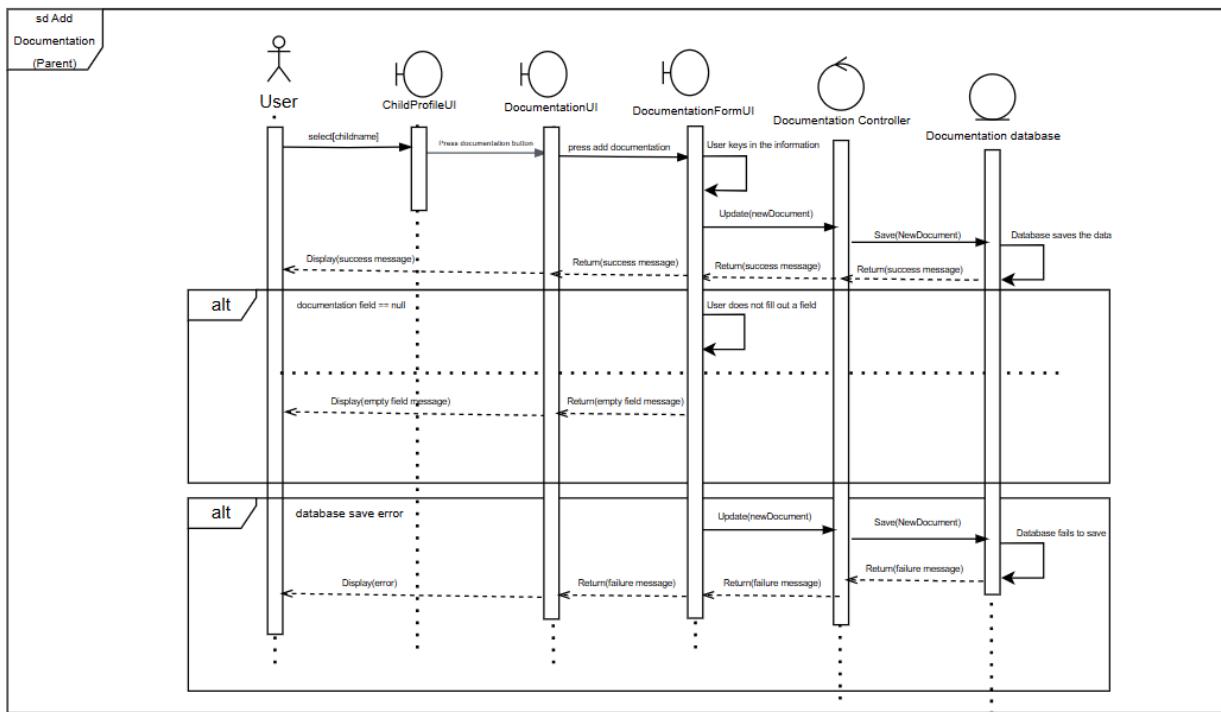
Use Case 6:



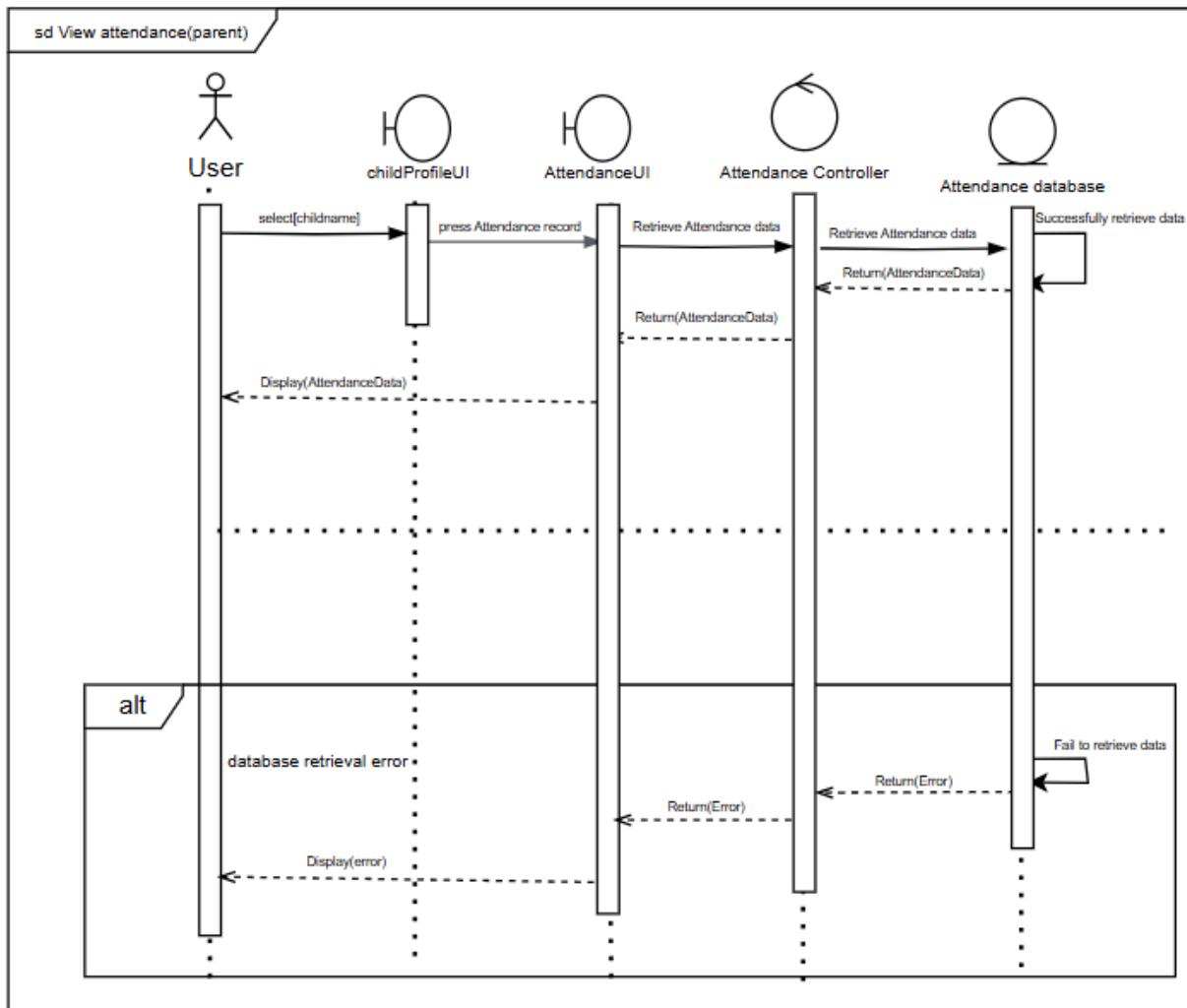
User case 7:



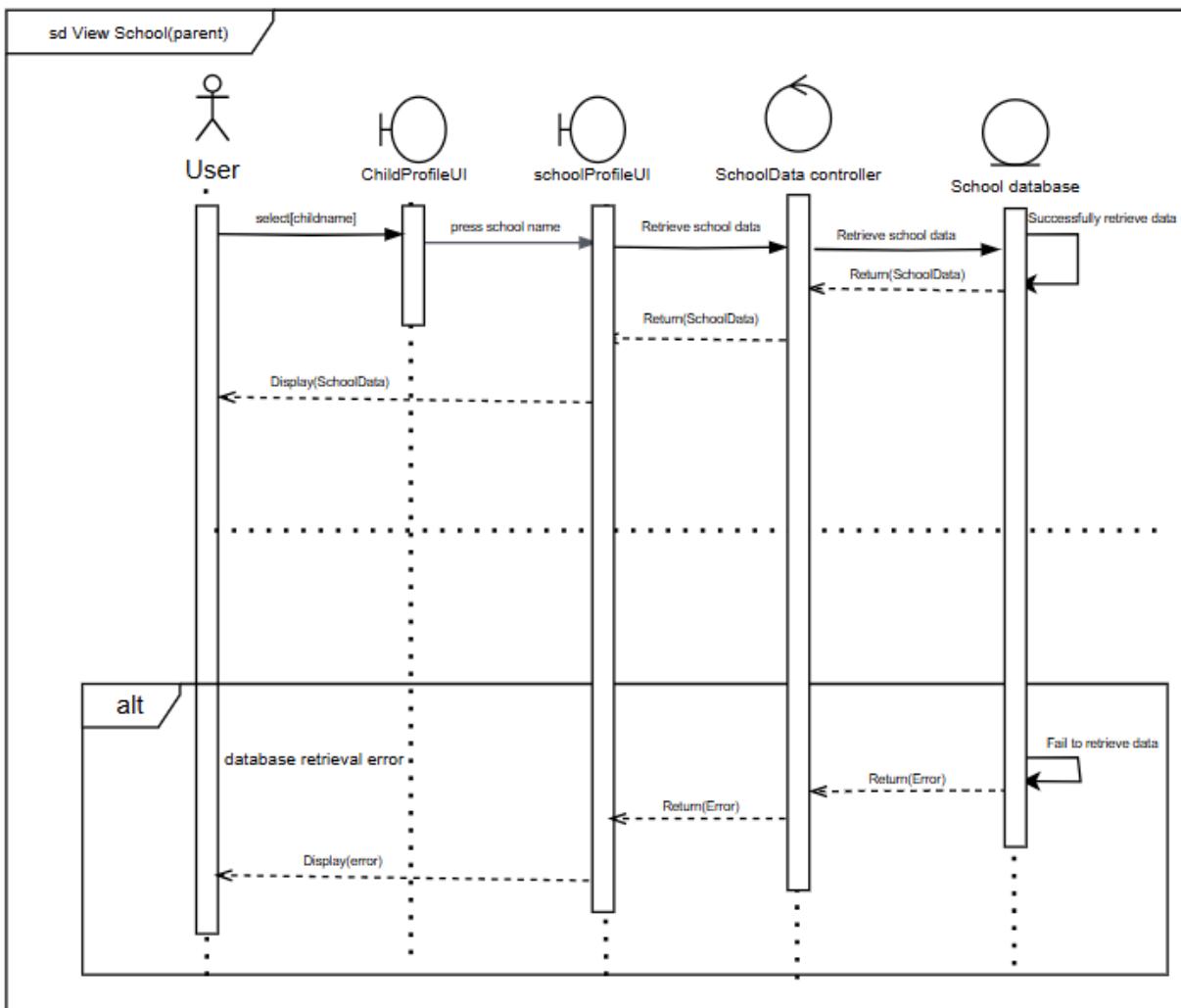
Use case 8:



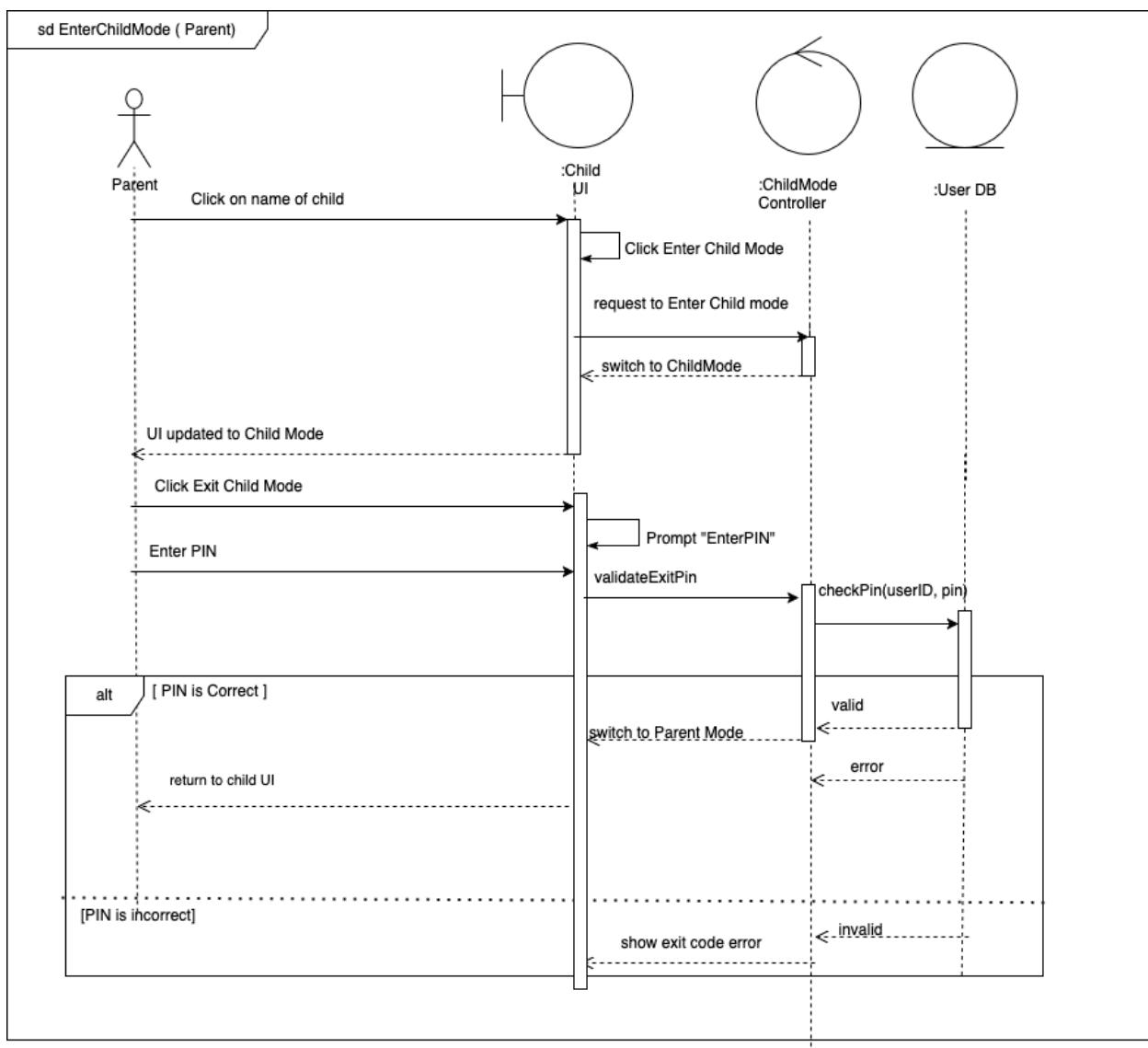
Use case 9:



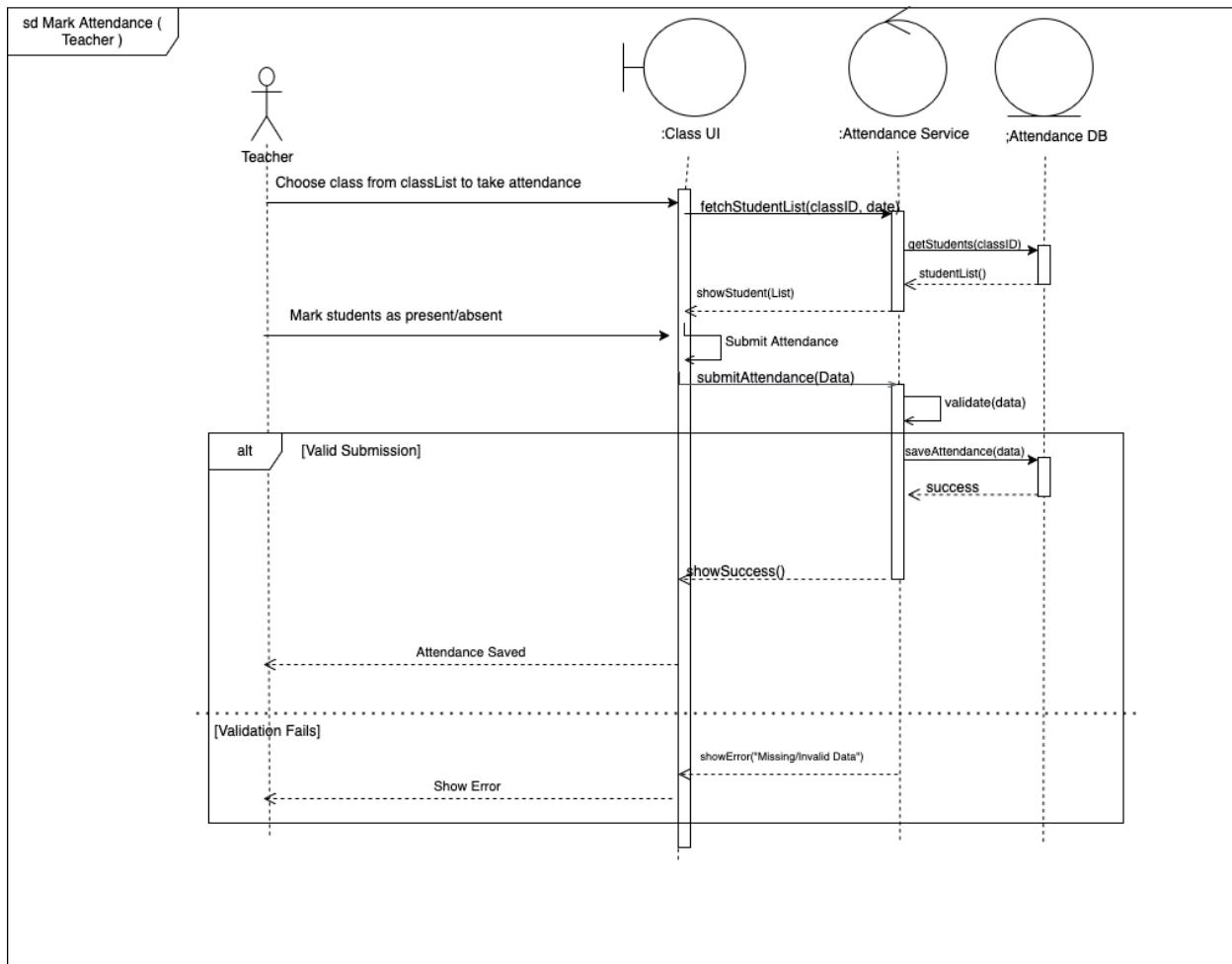
Use case 10:



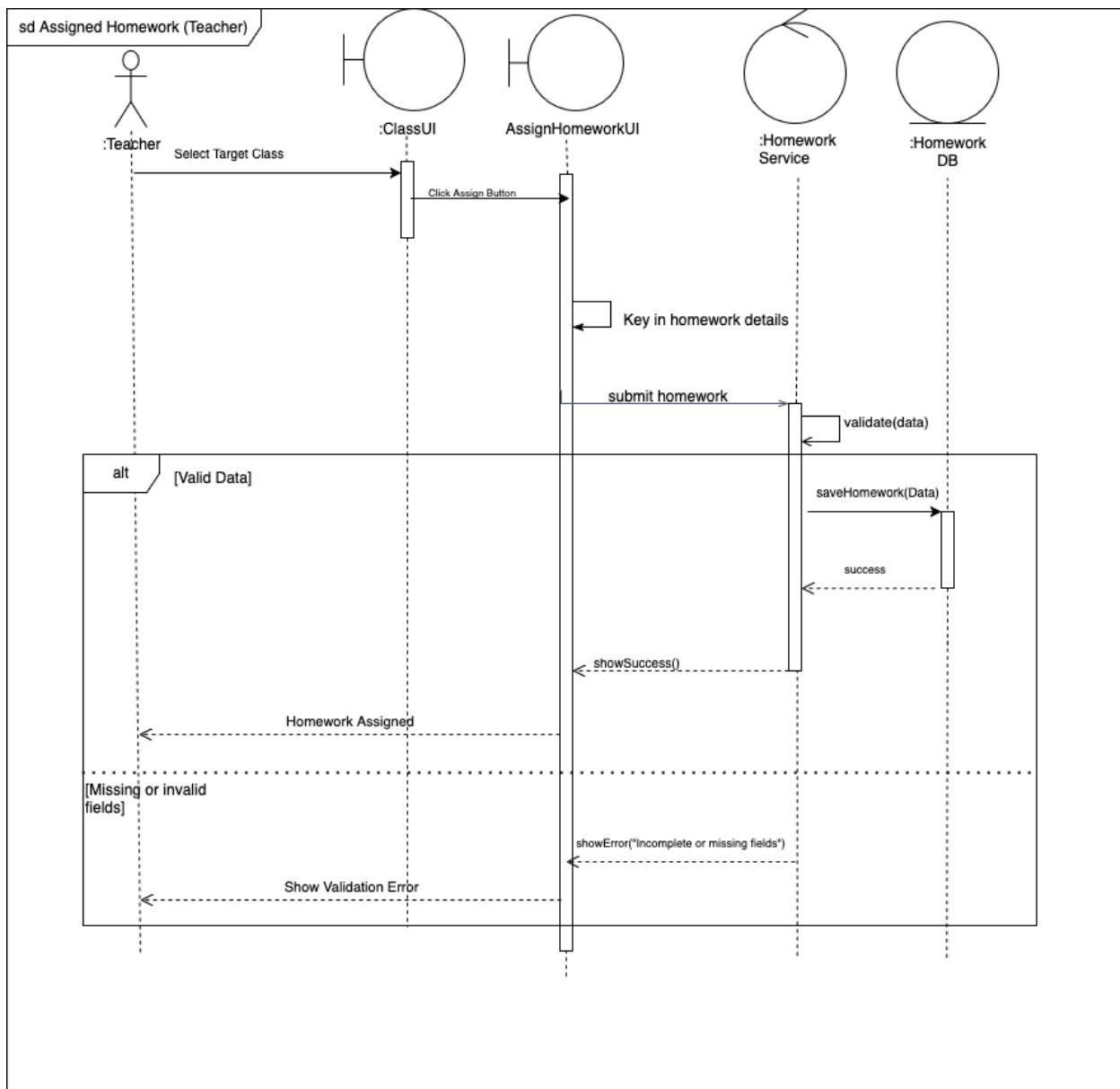
Use Case 11:



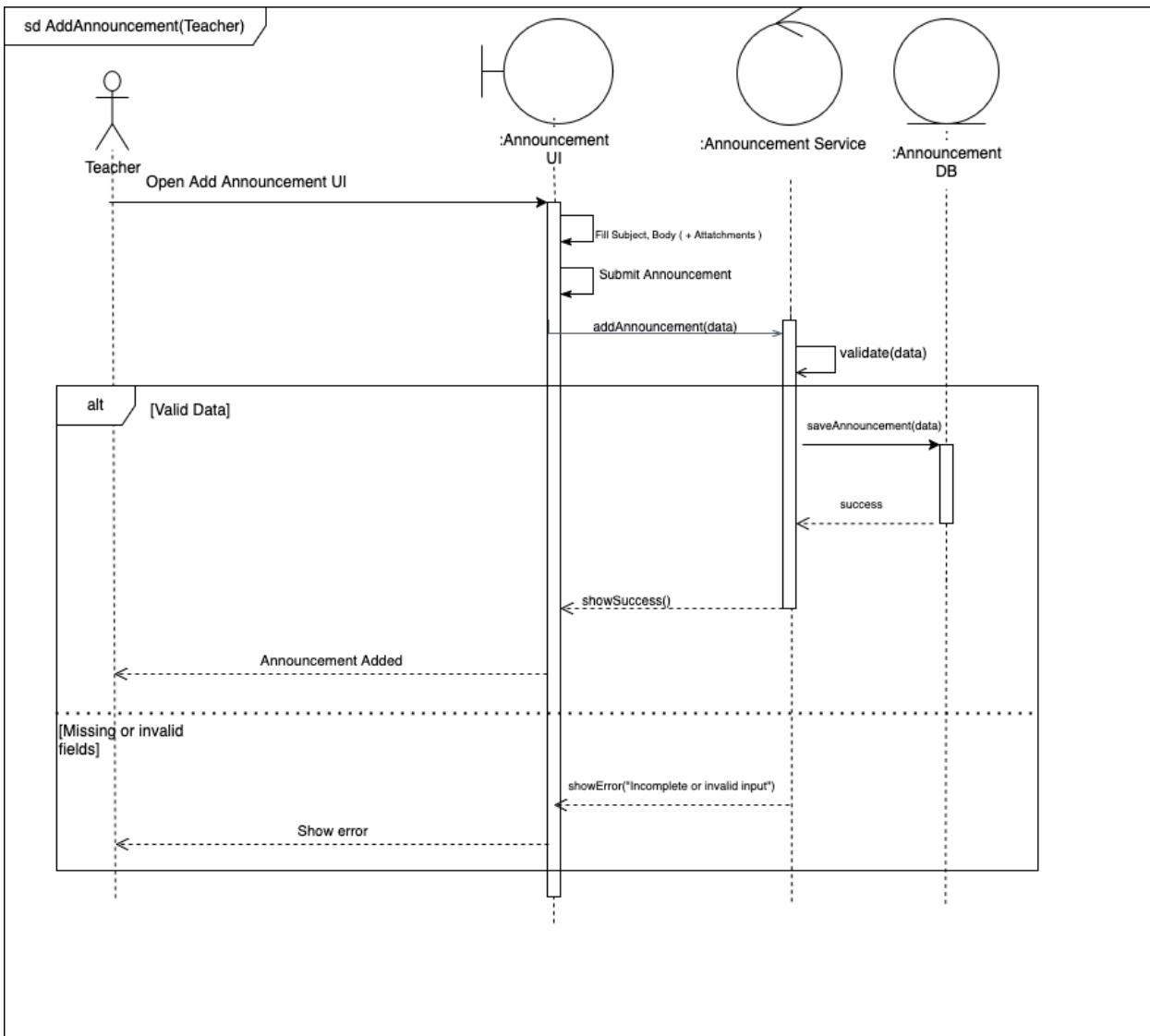
Use Case 12:



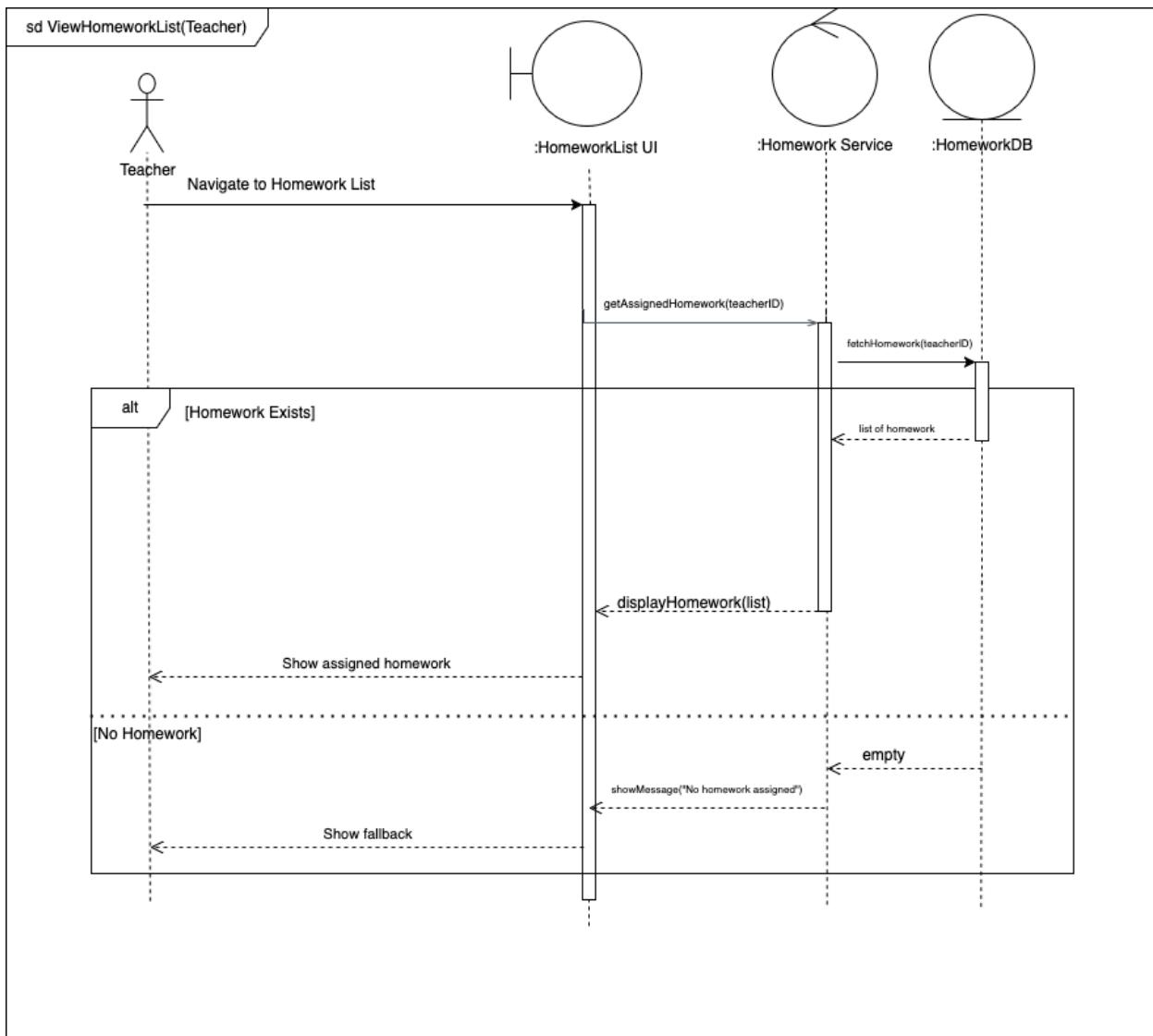
Use Case 13:



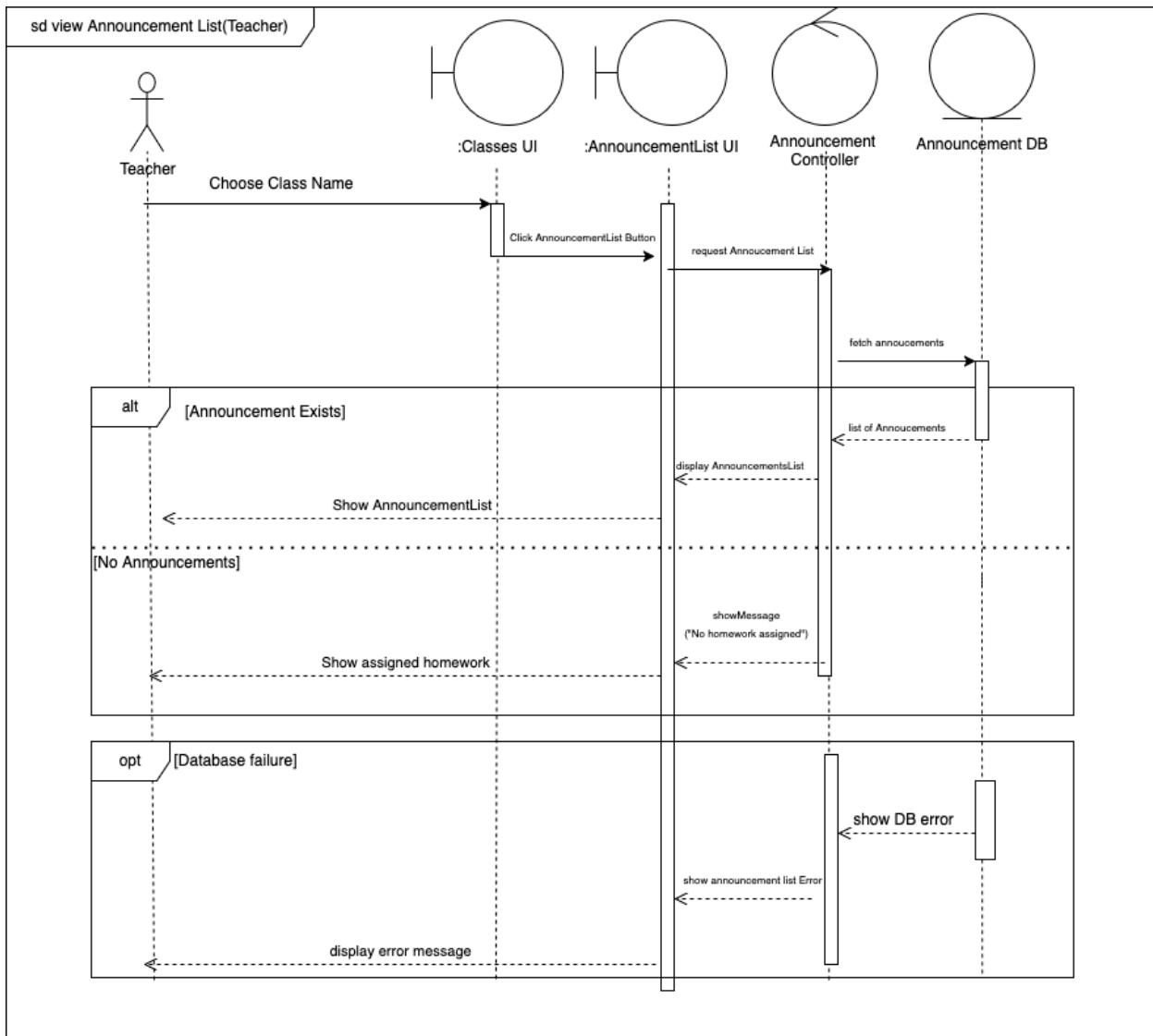
Use Case 14:



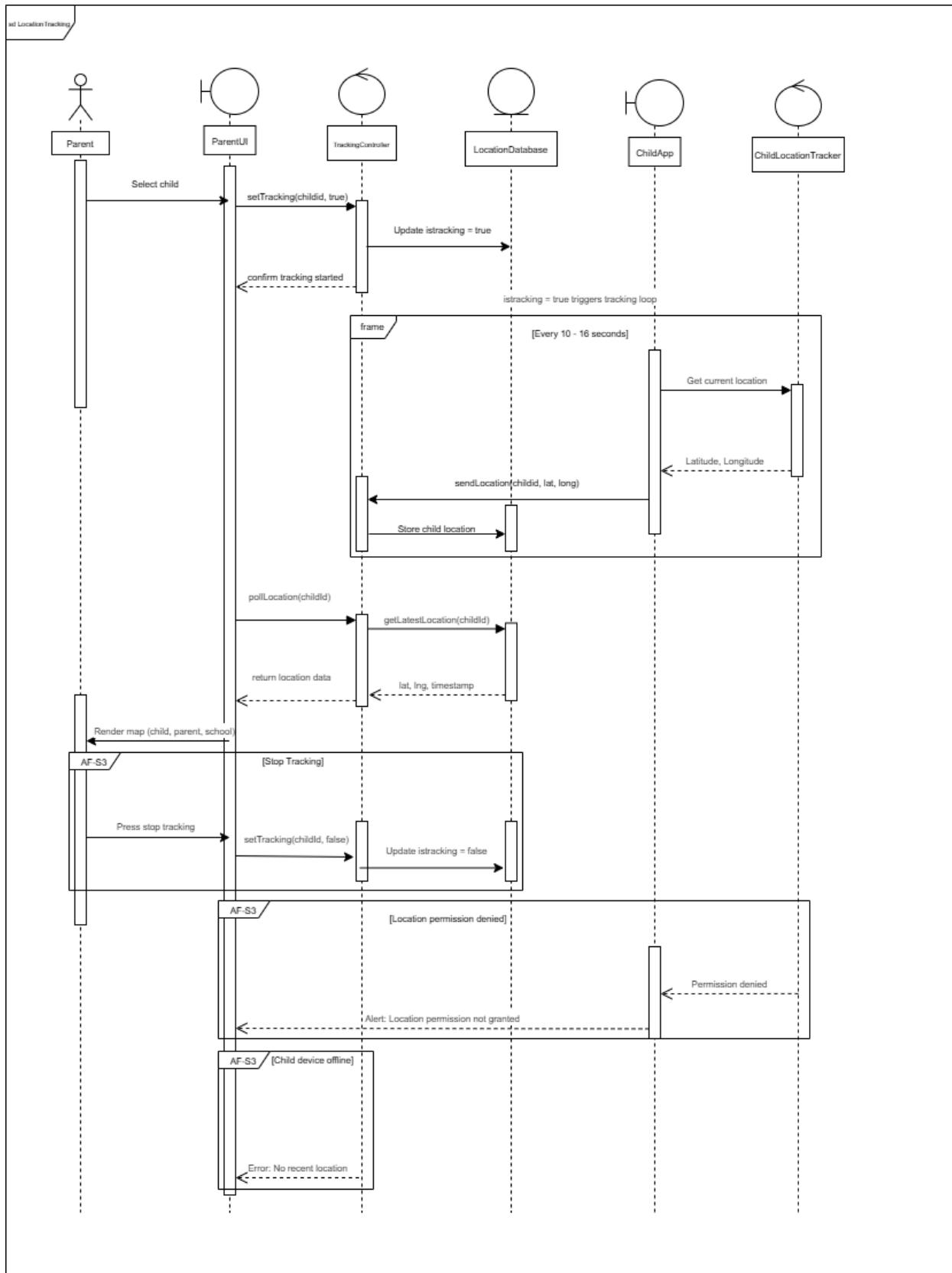
Use Case 15:



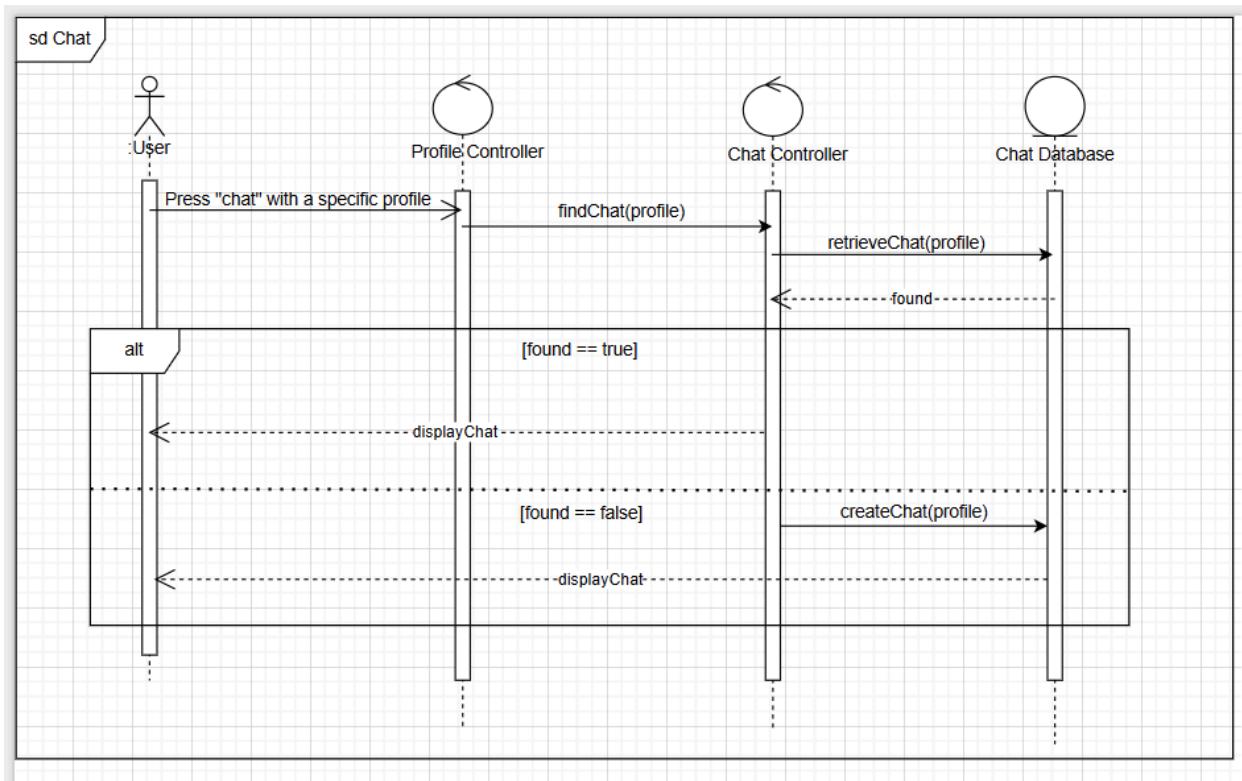
Use Case 16:



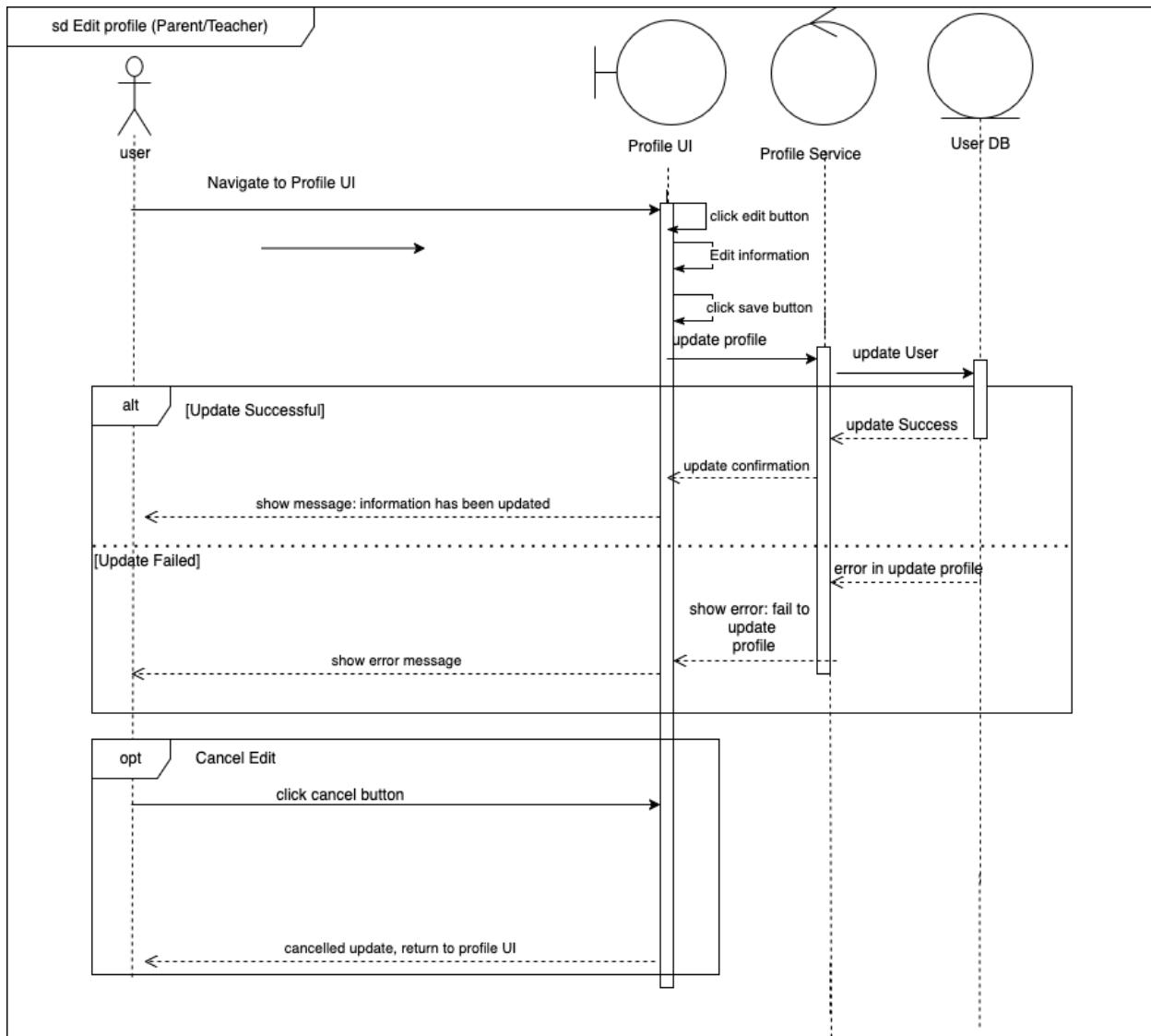
Use Case 17:



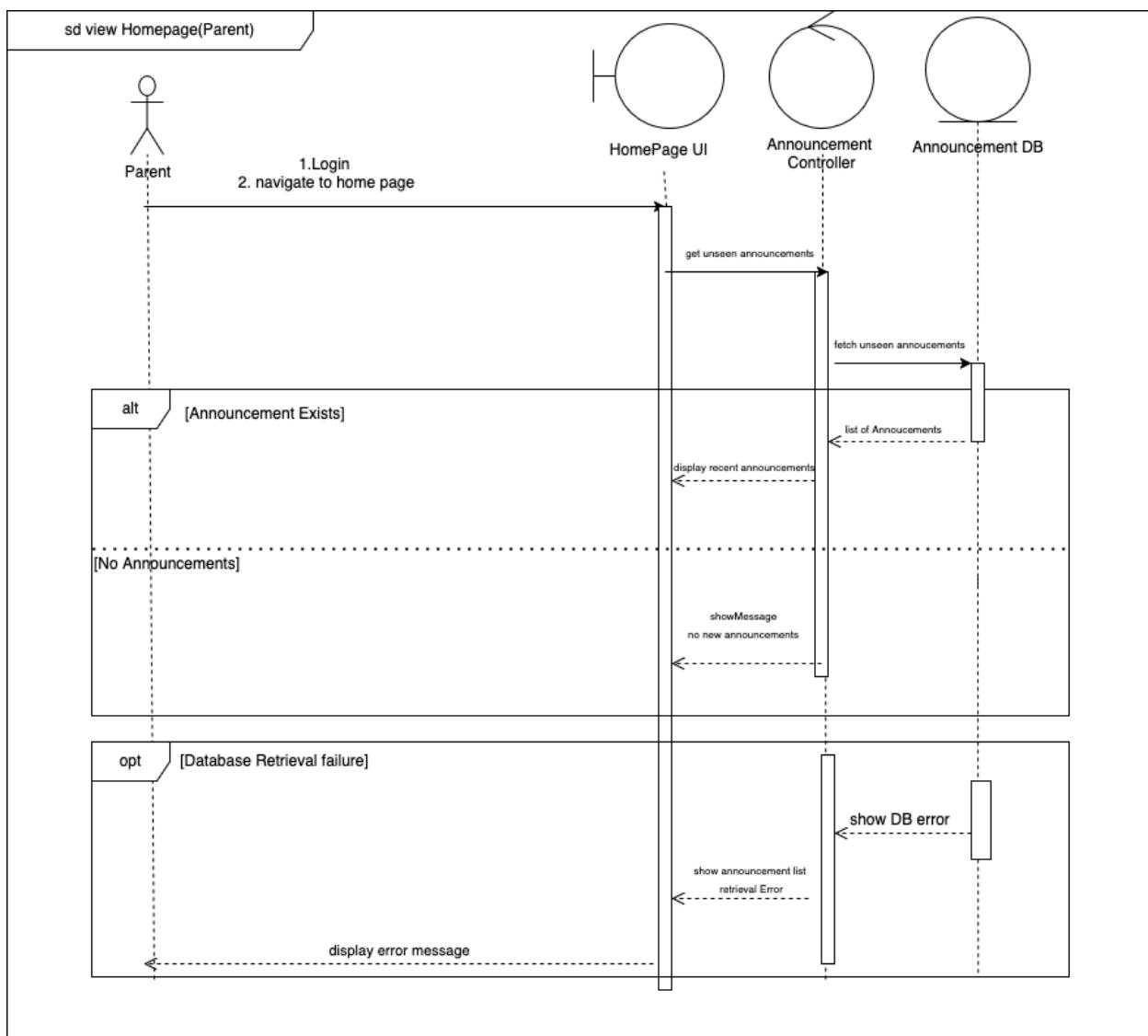
Use case 18:



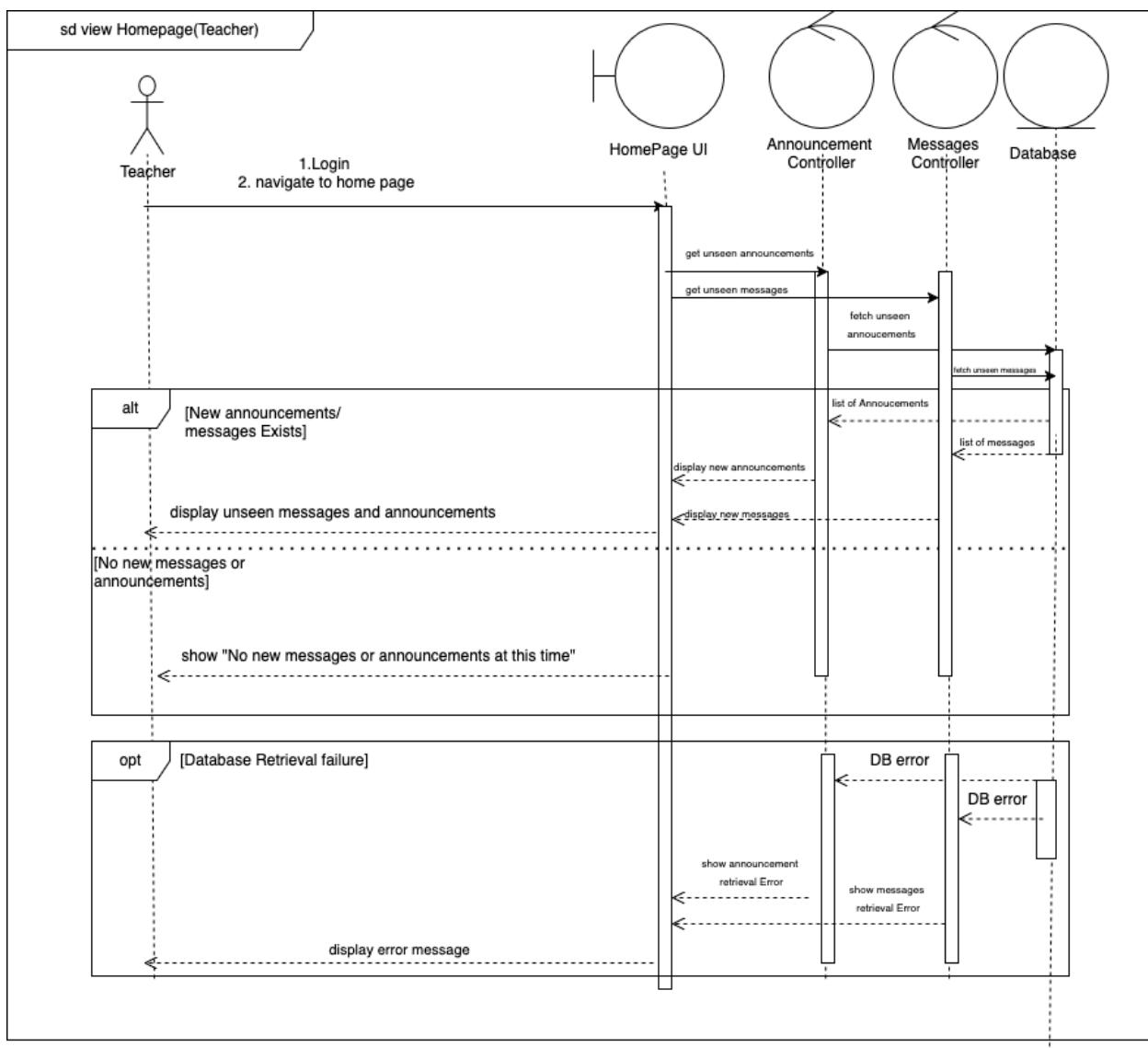
Use Case 19:



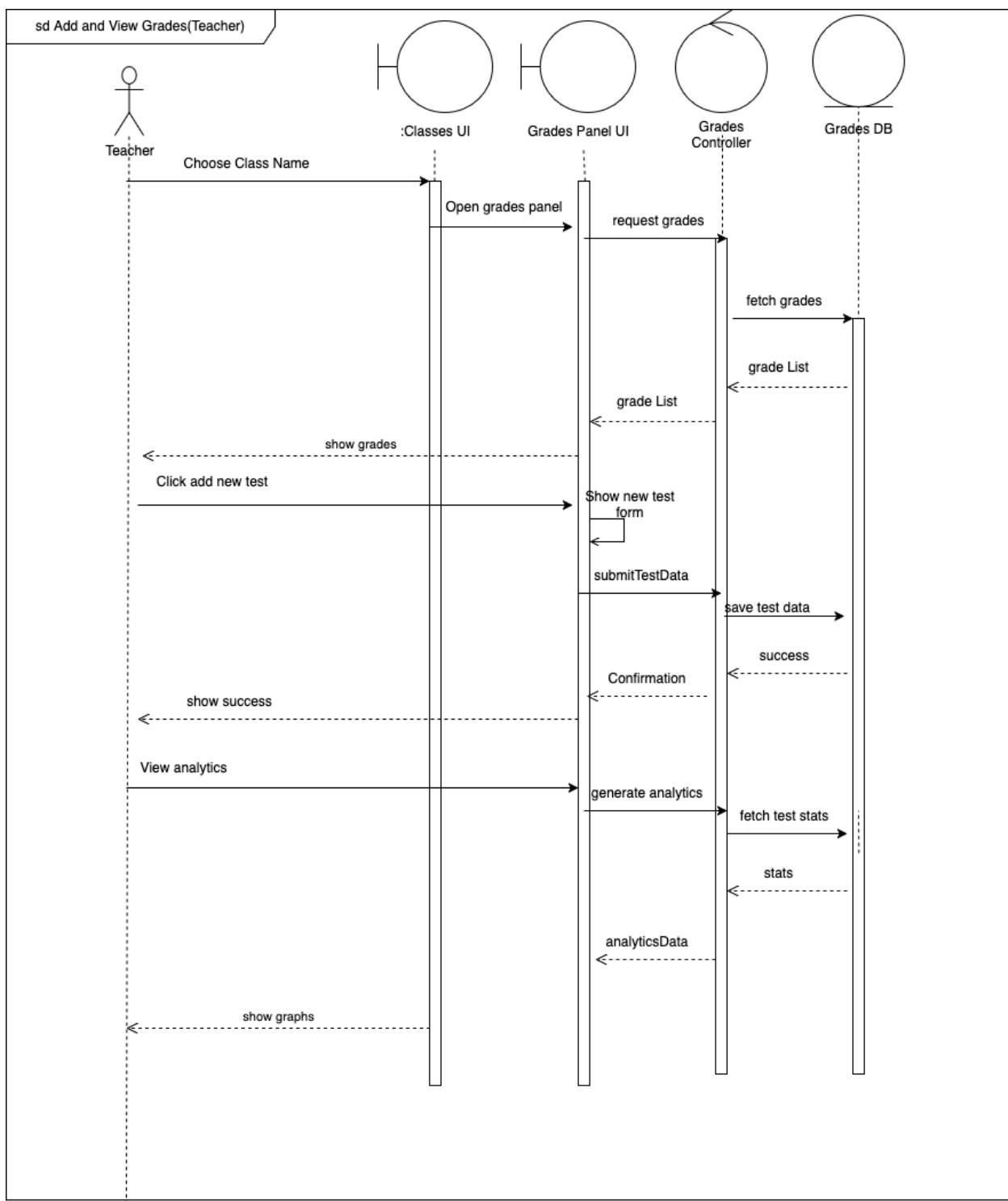
Use Case 20:



Use Case 21:

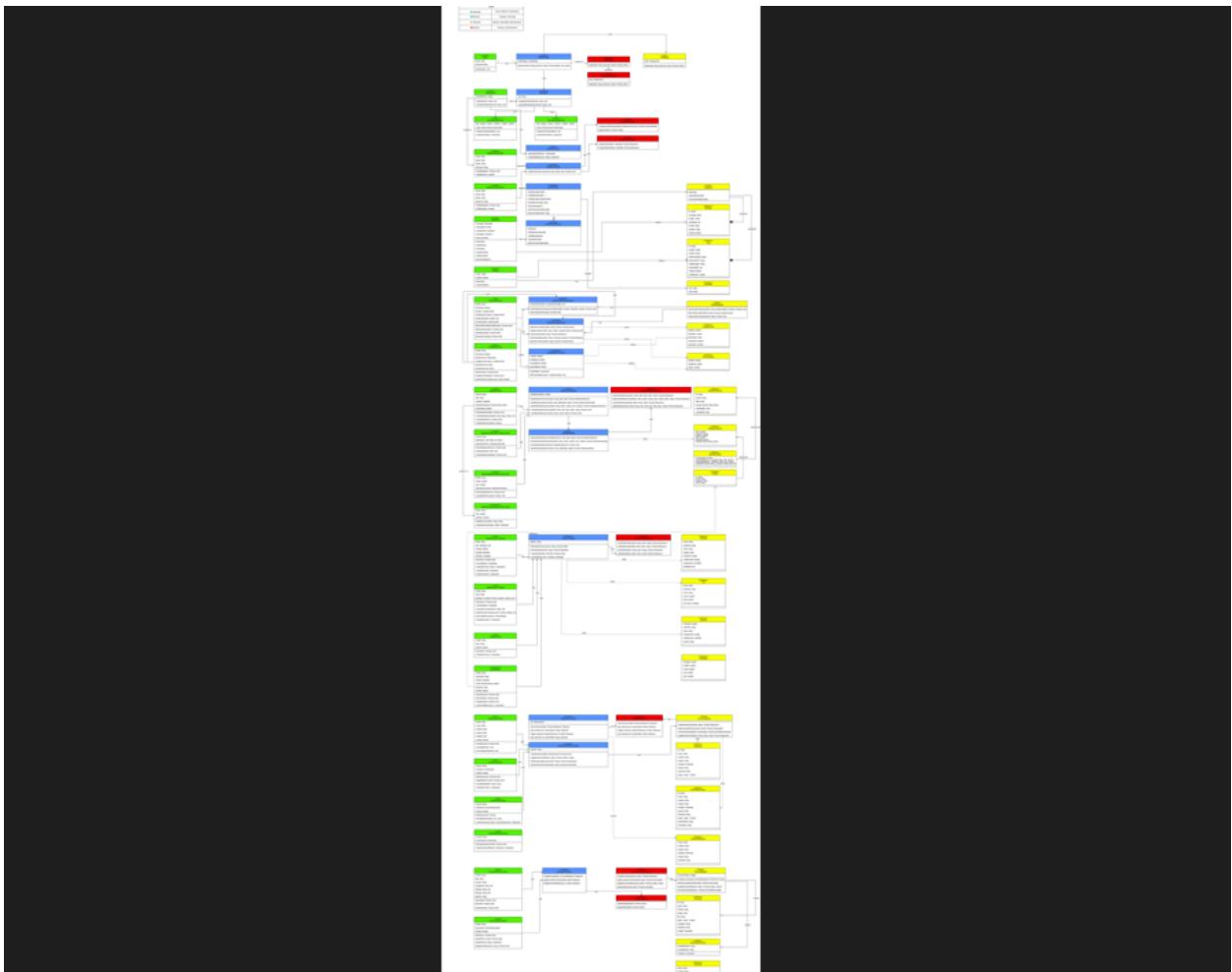


Use Case 22:

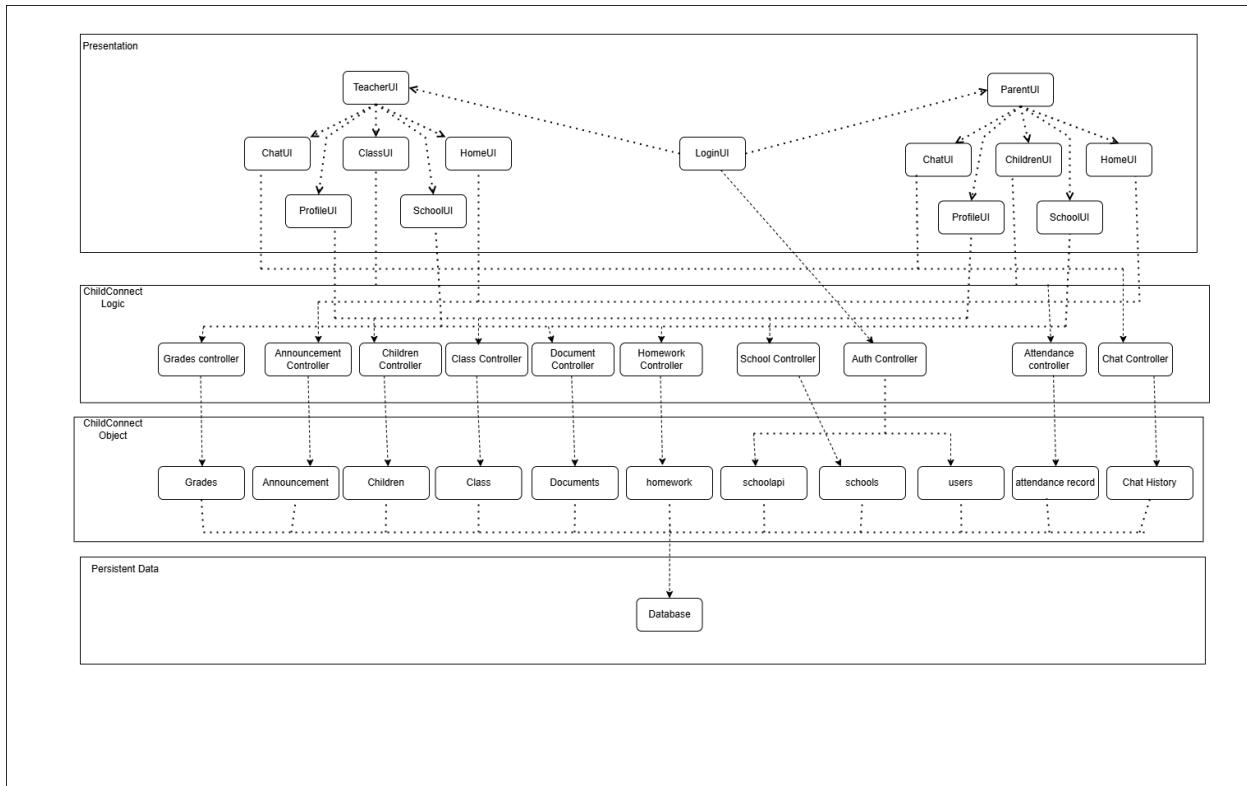


Class Diagram

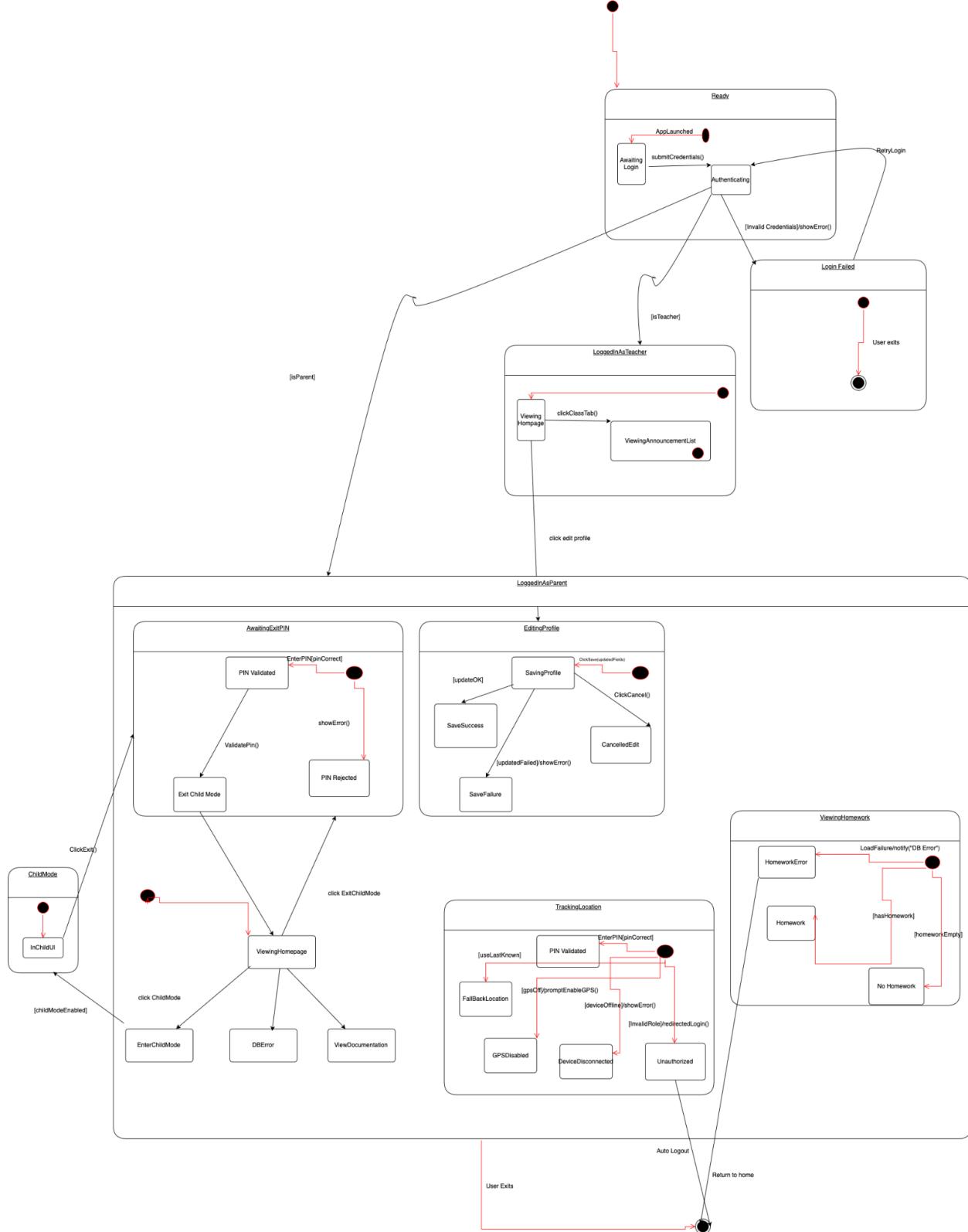
Will be placed in a separate file ClassDiagram.png due to its size. A FULL class diagram including Model, View and Controller classes is **available in ClassDiagram.png**. Shown below is only a snippet of what the full thing looks like.



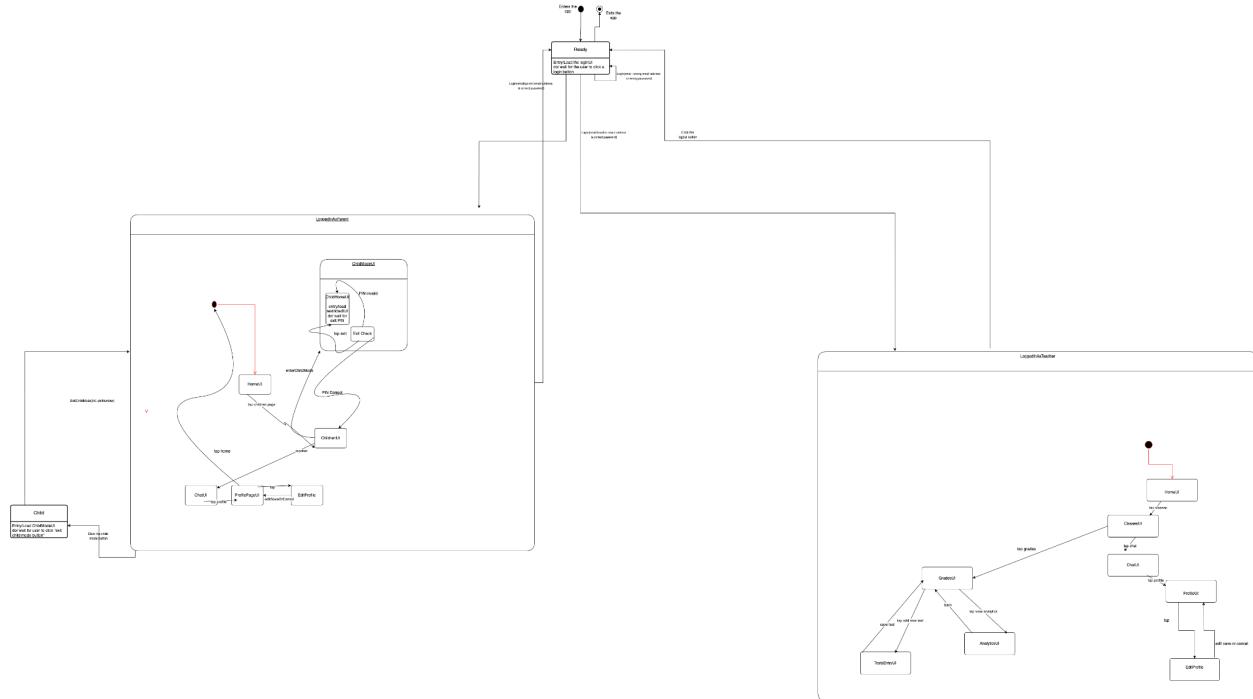
System Architecture



State Machine Diagram



Dialog Map



Application Skeleton

ChildConnect/

```
|- app/
  |- (parent)/          # Frontend app screens
    |- (tabs)/          # Parent role screens
      |- home.tsx       # Parent tabs structure
      |- children.tsx    # Parent home screen
      |- chatlist.tsx    # Children list screen
      |- profile.tsx     # Parent chat list
                         # Parent profile

    |- children/          # Child-related screens
      |- [id].tsx         # Child detail screen
      |- childmode.tsx    # Child mode setup
      |- childinfo.tsx    # Child info screen

    |- chat/              # Chat screens
      |- [id].tsx         # Chat detail screen

  |- profile/            # Profile-related screens
    |- attendance.tsx    # Child attendance view
```

```

    └── childhomework.tsx      # Child homework view
    └── documentationlist.tsx # Documentation list
    └── testResult.tsx        # Test results view

    └── (teacher)/           # Teacher role screens
        └── (tabs)/          # Teacher tabs structure
            └── home.tsx       # Teacher home screen
            └── classes.tsx     # Classes list screen
            └── chatlist.tsx    # Teacher chat list
            └── profile.tsx      # Teacher profile

        └── classes/          # Class management screens
            └── [id].tsx        # Class details
            └── addHomework.tsx   # Add homework form
            └── classdetails.tsx  # Class details

        └── student/          # Student management
            └── [id].tsx        # Student details

        └── chat/              # Chat screens
            └── [id].tsx        # Chat detail screen

        └── profile/           # Profile-related screens
            └── homeworkList.tsx # Homework list
            └── attendance.tsx   # Mark attendance
            └── attendanceRecord.tsx # Attendance records
            └── newTest.tsx       # Create new test
            └── gradeList.tsx     # Grade list view
            └── childhomework.tsx # Student homework view

    └── (childmode)/         # Child mode screens
        └── [id].tsx          # Child mode interface

    └── register/            # Registration flows
        └── registerParent.tsx # Parent registration
        └── registerTeacher.tsx # Teacher registration

    └── login.tsx            # Login screen
    └── roleselection.tsx    # Role selection screen
    └── forget.tsx           # Forgot password screen
    └── globals.css          # Global styles
    └── index.tsx            # Entry/Onboarding screen

    └── assets/              # App assets

```

```

  └── images/          # Images and SVGs
      ├── onboardingimg.svg    # Onboarding image
      ├── profilepic.svg      # Default profile picture
      └── legoat.jpg          # Sample profile image

  └── components/      # Reusable UI components
      └── data/            # Static mock data
          └── childData.ts   # Child mock data

      └── ChildLocationTracker.tsx  # Location tracking component
      └── routebutton.tsx        # Navigation buttons
      └── PasswordChangeModal.tsx # Password change modal

  └── factories/        # Factory pattern implementations
      ├── profileImageFactory.tsx  # Profile image creation
      └── roleViewFactory.tsx     # Role-specific UI generation

  └── utils/            # Utility functions
      └── server_ip.json       # Backend server IP configuration

  └── backend/          # FastAPI backend server
      └── controllers/        # API route controllers
          ├── auth_controller.py  # Authentication
          ├── profile_controller.py # Profile management
          ├── child_controller.py  # Child management
          ├── class_controller.py  # Class management
          ├── chat_controller.py  # Chat functionality
          ├── location_controller.py # Location tracking
          ├── attendance_controller.py # Attendance tracking
          ├── homework_controller.py # Homework management
          ├── document_controller.py # Document management
          ├── announcement_controller.py # Announcements
          ├── school_controller.py  # School information
          ├── childmode_controller.py # Child mode management
          ├── tts_controller.py     # Text-to-speech
          ├── stt_controller.py     # Speech-to-text
          └── translation_controller.py # Translation services

  └── models/           # Backend data models
      └── parent/            # Parent-related models
          ├── ChildrenModel.ts  # Children data model
          ├── ChildDetailModel.ts # Child detail model
          ├── ChatModel.ts       # Chat data model
          └── ParentModel.ts     # Parent data model

```

```
|- static/          # Static files served by backend
|- venv/           # Python virtual environment
|- requirements.txt # Python dependencies
└── main.py        # FastAPI entry point

node_modules/      # Node.js dependencies

firebaseConfig.js  # Firebase configuration
serviceAccountKey.json # Firebase service account key
speech-to-text-key.json # Google Cloud Speech API key
.env               # Environment variables
.gitignore         # Git ignore configuration
package.json       # NPM package configuration
app.json           # Expo app configuration
babel.config.js   # Babel configuration
tailwind.config.js # Tailwind CSS configuration
tsconfig.json      # TypeScript configuration
README.md          # Project documentation
```

Application Testing

White box testing and black box testing is being utilized when executing application testing.

Black Box Testing

During the account sign up phase, the user has to key in their email, passwords and name. After signing up, the account user's details and particulars will be stored in the database. This will allow users to login successfully afterwards.

During the user login phase, the user has to key in the same email and password used when they first signed up for an account. When the email and password matches an existing user which is stored in the database, the user will be directed into the home screen depending on whether it is a parent account or teacher account.

EquivalenceClassTesting

Equivalence Class Testing and State Based Testing will be used instead of Boundary Value Testing. Equivalence Class testing is due to the login and signup process requiring discrete values as their inputs. State-Based Testing is used to ensure that the application behaves accurately in every possible state and transition.

Login Function:

Valid Equivalence Class: Valid username and password. Username and password matches an existing user.

Invalid Equivalence Class: Invalid username and password. Username and password does not match any existing user.

Sign Up Function:

Valid Equivalence Class: Email, Password and Full Name are valid. No overlapping of email addresses with an existing user.

Invalid Equivalence Class: Email, Password and Full Name are invalid(format). Overlapping of email addresses with an existing user.

Test Cases and Testing Results

Sign Up (Register Here)

Test Case Name	Test Input	Expected Output	Actual Output	Test Result
Register-1	Click “Register Here”	Display of user interface asking if the user is a parent or a teacher	Display of user interface asking if the user is a parent or a teacher	Pass
Register-2	Click “I'm a Parent!”	Display of registration page	Display of registration page	Pass
Register-3	Click “I'm a Teacher!”	Display of registration page	Display of registration page	Pass

Parent Registration

Input parameters: Name, Email, Phone Number, Password

Test Case Name	Test Input	Expected Output	Actual Output	Test Result
Register-1	(Valid) Full Name:"Bob Tan" (Valid) Email: "bobtan@gmail.com" (Valid) Phone Number: 90302581 (Valid) Password:"parent"	Success Account created! OK	Success Account created! OK	Pass
Register-2	(Invalid) Full Name:"" (Invalid) Email:"" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-3	(valid) Full Name:"Bob Tan"	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields.	Pass

	(Invalid) Email:"" (Invalid) Phone Number:"" (Invalid) Password:""		OK	
Register-4	(Invalid) Full Name:"" (Valid) Email: "bobtan@gmail.com" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-5	(Invalid) Full Name:"" (Invalid) Email:"" (Valid) Phone Number:"90302581" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-6	(Invalid) Full Name:"" (Invalid) Email:"bobtanatgmaildotcom" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-7	(Valid) Full Name:"Bob Tan" (Invalid) Email:"bobtanatgmaildotcom"	Registration failed Firebase: Error (auth/invalid-email) OK	Registration failed Firebase: Error (auth/invalid-email) OK	Pass

	(Valid) Phone Number:"90302581" (Valid) Password:"parent"			
Register-8	(Valid) Full Name:"Bob Tan" (Existing Email) Email:"exist@gmail.com" (Valid) Phone Number:"90302581" (Valid) Password:"parent"	The email is already registered. OK	The email is already registered. OK	Pass

Teacher Registration

Input parameters: Name, Email, Phone Number, Password

Test Case Name	Test Input	Expected Output	Actual Output	Test Result
Register-1	(Valid) Full Name:"Bob Tan" (Valid) Email: bobtan@gmail.com (Valid) Phone Number: 90302581 (Valid) Password:"parent"	Success Account created! OK	Success Account created! OK	Pass
Register-2	(Invalid) Full Name:"" (Invalid) Email:"" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass

Register-3	(valid) Full Name:"Bob Tan" (Invalid) Email:"" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-4	(Invalid) Full Name:"" (Valid) Email:" bobtan@gmail.com " (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-5	(Invalid) Full Name:"" (Invalid) Email:"" (Valid) Phone Number:"90302581" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-6	(Invalid) Full Name:"" (Invalid) Email:"bobtanatgmaildotcom" (Invalid) Phone Number:"" (Invalid) Password:""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields. OK	Pass
Register-7	(Invalid) Full Name:"Bob Tan"	Registration failed Firebase: Error	Registration failed Firebase: Error	Pass

	(Invalid) Email:"bobtanatgmaildotcom" (Valid) Phone Number:"90302581" (Valid) Password:"parent"	(auth/invalid-email) OK	(auth/invalid-email) OK	
Register-8	(Valid) Full Name:"Bob Tan" (Existing Email) Email:"exist@gmail.com" (Valid) Phone Number:"90302581" (Valid) Password:"parent"	The email is already registered. OK	The email is already registered. OK	Pass

Login (Get Started)

Input parameters: Emails, Password

Test Case Name	Test Input	Expected Output	Actual Output	Test Result
Login-1	(Valid) Email:" bobtan@gmail.com " (Valid) Password:"parent"	Login Successful Bob Tan	Login Successful Bob Tan	Pass
Login-2	(Valid) Email:" bobtan@gmail.com " (Invalid) Password:"wrongpassword"	Login Failed Firebase: Error (auth/invalid-credential). OK	Login Failed Firebase: Error (auth/invalid-credential). OK	Pass
Login-3	(Valid) Email:" bobtan@gmail.com " (Invalid) Password:""	Login Failed Firebase: Error (auth/missing-password).	Login Failed Firebase: Error (auth/missing-password).	Pass

		OK	OK	
Login-4	(Invalid) Email:"" (Valid) Password:"parent"	Login Failed Firebase: Error (auth/invalid-e mail). OK	Login Failed Firebase: Error (auth/invalid-e mail). OK	Pass
Login-5	(Invalid) Email:" fakeuser@gmail.com " (Valid) Password:"parent"	Login Failed Firebase: Error (auth/invalid-c redential). OK	Login Failed Firebase: Error (auth/invalid-c redential). OK	Pass
Login-6	(Invalid) Email:"bobtanatgmailedotcom" (Valid) Password:"parent"	Login Failed Firebase: Error (auth/invalid-e mail). OK	Login Failed Firebase: Error (auth/invalid-e mail). OK	Pass

Parent Account

Children (Select a Child)

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Children-1	Select a child's name (Peter Tan)	Request for activation of location	Request for activation of location	Pass
Children-2	Clicking into attendance records	Details of child and calendar dates(Highlighted/non-h ighlighted)	Details of child and calendar dates(Highlighted/non-h ighlighted)	Pass
Children-3	Clicking into non-highlighted date	Attendance DD/MM/YYYY No data recorded for this day Contact a teacher if you believe this is an error	Attendance DD/MM/YYYY No data recorded for this day Contact a teacher if you believe this is an error	Pass
Children-4	Click into highlighted date	Absent/Present	Absent/Present	Pass

Children-5	Click add Documentation	Title: Content: Upload image Submit	Title: Content: Upload image Submit	Pass
Children-6	Click Submit	Success Documentation added! OK	Success Documentation added! OK	Pass
Children-7	Click Activate Tracking	Map shows child and school location	Map shows child and school location	Pass
Children-8	Click View Fullscreen Map	Display full picture of map with icons marking the locations of child and school	Display full picture of map with icons marking the locations of child and school	Pass

Chats

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Chats-1	Click Chats	Chats  Search for a chat... Class 1E4 2h ago Class 2K2 4h ago	Chats  Search for a chat... Class 1E4 2h ago Class 2K2 4h ago	Pass
Chats-2	Click into conversation (e.g Class 1E4)	Display individual conversations with a message bar user interface	Display individual conversations with a message bar user interface	Pass

Home

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Home-1	Click Home	Displays picture icon of account “Welcome Back!” Announcement Unread chats	Displays picture icon of account “Welcome Back!” Announcement Unread chats	Pass

Profile

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Profile-1	Click Profile	Display Edit Button at top right corner. Display profile picture, name, email, phone number, work phone number and address respectively. Display Child Mode button Display code change button Display Logout button at the end.	Display edit button at top right corner. Display profile picture, name, email, phone number, work phone number and address respectively. Display Child Mode button Display code change button Display Logout button at the end.	Pass
Profile-2	Click Edit Button	Same layout Changing of profile picture is enabled Editing of phone, work phone and address are enabled	Same layout Changing of profile picture is enabled Editing of phone, work phone and address are enabled	Pass
Profile-3	Click Logout Button	Redirect to Login Page	Redirect to Login Page	Pass
Profile-4	Click Code Change button	Change Password Save Cancel	Change Password Save Cancel	Pass
Profile-5	Enter code lesser than 4 digits. E.g “01”	Error Password must be 4 digits OK	Error Password must be 4 digits OK	Pass
Profile-6	Click Save Button (Changing of code)	Success Password updated OK	Success Password updated OK	Pass
Profile-7	Click Cancel button (Changing of code)	Return back to child profile page	Return back to child profile page	Pass

Child Mode

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Child-1	Click Child Button	Child Mode Enabled This device is in child mode. Below is the list of homework assigned. Math Homework Math TYS page 2 Due: 27 Mar 2025 “I’m leaving home/school” Button “Exit Child Mode” Button	Child Mode Enabled This device is in child mode. Below is the list of homework assigned. Math Homework Math TYS page 2 Due: 27 Mar 2025 “I’m leaving home/school” Button “Exit Child Mode” Button	Pass
Child-2	Click “I’m leaving home/school” Button	Tracking started Your location is now being tracked OK	Tracking started Your location is now being tracked OK	Pass
Child-3	Click Exit Child Mode	Display of notification “Enter Exit Code”	Display of notification “Enter Exit Code”	Pass
Child-4	Invalid Exit Code	Incorrect Code The exit code entered is incorrect. OK	Incorrect Code The exit code entered is incorrect. OK	Pass
Child -5	Valid Exit Code	Return to child details page	Return to child details page	Pass
Child-6	Cancel Exit Code	Return to Child Mode	Return Child Mode	Pass

Teacher Account

Home

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Home-1	Click Home	Displays picture icon of account “Welcome Back!” Announcement Unread chats	Displays picture icon of account “Welcome Back!” Announcement Unread chats	Pass

Chats

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Chats-1	Click Chats	Chats Chatting with Bob Tan Hi 7/4/2025, 11:09:13 AM	Chats Chatting with Bob Tan Hi 7/4/2025, 11:09:13 AM	Pass
Chats-2	Click into conversation	Display individual conversations with a message bar user interface, File attachment button	Display individual conversations with a message bar user interface, file attachment button	Pass
Chats-3	Send a message	Message is being sent in a form of text bubble Double tick symbol(Message read) Single tick symbol(Message unread)	Message is being sent in a form of text bubble Double tick symbol(Message read) Single tick symbol(Message unread)	Pass
Chats-4	Attach an image	Popup notification of requesting access to image file, image file accessed	Popup notification of requesting access to image file, image file accessed	Pass

Classes

Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Class-1	Click Classes	My Classes 1E4 Role: Form Teacher	My Classes 1E4 Role: Form Teacher	Pass
Class-2	Click into "1E4"	Homework List Announcement List Assign Homework Add Announcement Search Bar Display of student name and attendance Individual profile button for each student	Homework List Announcement List Assign Homework Add Announcement Search Bar Display of student name and attendance Individual profile button for each student	Pass
Class-3	Click Homework List	Math Homework Math TYS page 2 Subject: Math Due: 27 March 2025	Math Homework Math TYS page 2 Subject: Math Due: 27 March 2025	Pass

		Status: open	Status: open	
Class-4	Click Announcement List	Announcements Exam Reminder Science exam this Friday at 9am. By Jane Tan on 23 March 2025 Status: open	Announcements Exam Reminder Science exam this Friday at 9am. By Jane Tan on 23 March 2025 Status: open	Pass
Class-5 (Homework)	Click Assign Homework	Assign Homework Title: Content: Due Date: Subject:	Assign Homework Title: Content: Due Date: Subject:	Pass
Class-6 (Homework)	(Empty inputs) Title: "" Content: ""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields OK	Pass
Class-7 (Homework)	Title: "Math" (Empty) Content: ""	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields OK	Pass
Class-8 (Homework)	(Empty) Title: "" Content: "Do chapter 1"	Missing Fields Please fill in all fields. OK	Missing Fields Please fill in all fields OK	Pass
Class-9 (Announcement)	Click Add Announcement	Title: Content: Submit Announcement	Title: Content: Submit Announcement	Pass
Class-10 (Announcement)	(Empty inputs) Title: "" Content: ""	Missing Fields Please enter both name and content OK	Missing Fields Please enter both name and content OK	Pass
Class-11 (Announcement)	Title: "Math" (Empty) Content: ""	Missing Fields Please enter both name and content OK	Missing Fields Please enter both name and content OK	Pass
Class-12 (Announcement)	(Empty) Title: "" Content: "Do chapter 1"	Missing Fields Please enter both name and content OK	Missing Fields Please enter both name and content OK	Pass
Class-13 (Announcement)	Click "Attendance: Absent/Present"	Toggling of attendance between absent and present	Toggling of attendance between absent and present	Pass

Individual Profile of Student (Classes)

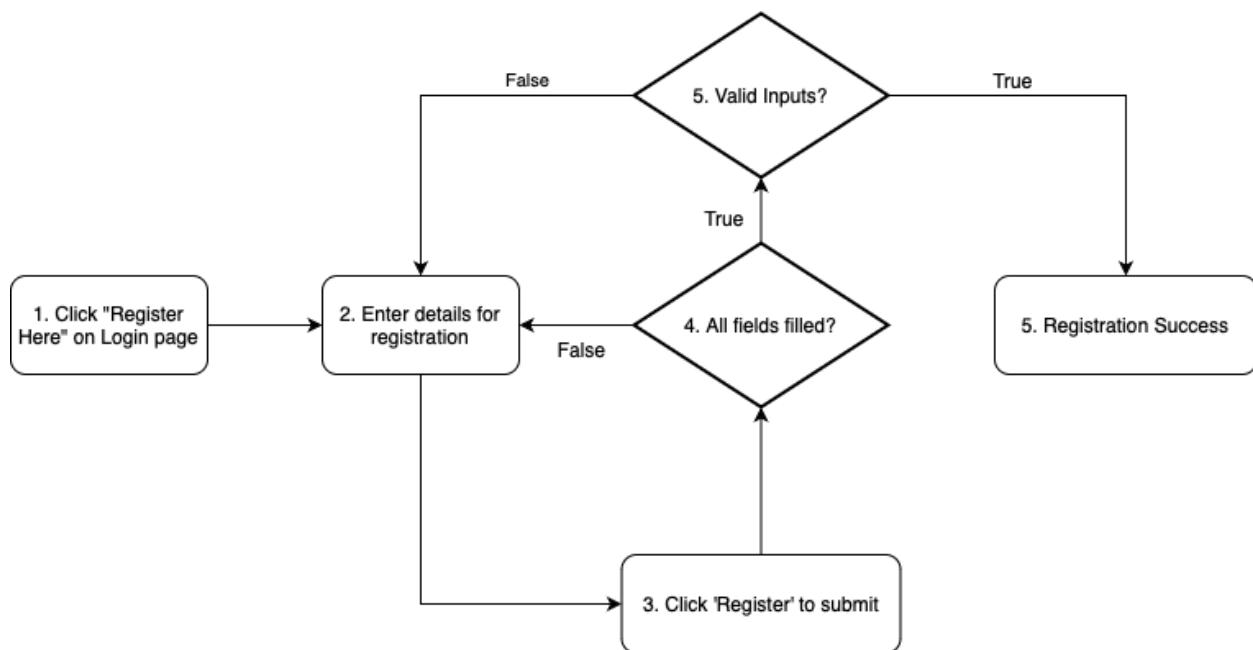
Test Case Name	Test Scenarios	Expected Output	Actual Output	Test Result
Profile-1	Click the 'Profile' button next to a student's name	Details of the student Chat and contact information of Mother and Father of the student Attendance Records Documentation Homework List	Details of the student Chat and contact information of Mother and Father of the student Attendance Records Documentation Homework List	Pass
Profile-2	Click Chat(Mother/Father)	Display of chat page between parents	Display of chat page between parents	Pass
Profile-3	Click Contact Info(Mother/Father)	Display of parents details(Name, Email, Phone)	Display of parents details(Name, Email, Phone)	Pass
Profile-4 (Attendance record)	Click Attendance Records	Display of student details Display of calendar (Red highlights mean absent/ Green highlights mean present)	Display of student details Display of calendar (Red highlights mean absent/ Green highlights mean present)	Pass
Profile-5 (Attendance record)	Click on any date in the calendar	Attendance DD/MM/YYYY Details of student Status of student(Present/Absent) Upload Photo Mark Present Mark Absent	Attendance DD/MM/YYYY Details of student Status of student(Present/Absent) Upload Photo Mark Present Mark Absent	Pass
Profile-6 (Attendance record)	Click Upload Photo	Access to image file to upload photo	Access to image file to upload photo	Pass
Profile-7 (Attendance record)	Click Mark Present	Success Marked Present OK	Success Marked Present OK	Pass
Profile-8 (Attendance record)	Click Mark Absent	Success Marked Absent OK	Success Marked Absent OK	Pass
Profile-9 (Documentation)	Click Documentation	Display the list of documentation being uploaded Uploaded pictures with respect to each document	Display the list of documentation being uploaded Uploaded pictures with respect to each document	Pass

		Add Documentation Button	Add Documentation Button	
Profile-10 (Documentation)	Click 'Add Documentation'	Title Content Upload Image Upload PDF Submit	Title Content Upload Image Upload PDF Submit	Pass
Profile-11 (Documentation)	Title: "Math" Content: "Do 2 pages"	Success Documentation added! OK	Success Documentation added! OK	Pass
Profile-12 (Documentation)	Title: "Math" Content: ""	Success Documentation added! OK	Success Documentation added! OK	Pass
Profile-13 (Documentation)	Title: "" Content: ""	Error Please enter a title. OK	Error Please enter a title. OK	Pass
Profile-13 (Documentation)	Title: "" Content: "Do 2 pages"	Error Please enter a title. OK	Error Please enter a title. OK	Pass
Profile-14 (Documentation)	Click "Upload image"	Access image files Selected image is shown	Access image files Selected image is shown	Pass
Profile-15 (Documentation)	Click "Change image"	Access image files Selected image is shown	Access image files Selected image is shown	Pass
Profile-16 (Documentation)	Click "Upload PDF"	Access files PDF file attached	Access files PDF file attached	Pass
Profile-17 (Documentation)	Click "Submit"	Documentation will be uploaded into the documentation list	Documentation will be uploaded into the documentation list	Pass
Profile-18	Click "Homework List"	Display the list of homeworks	Display the list of homeworks	Pass

White Box Testing

Control Flow Testing

User Account Registration

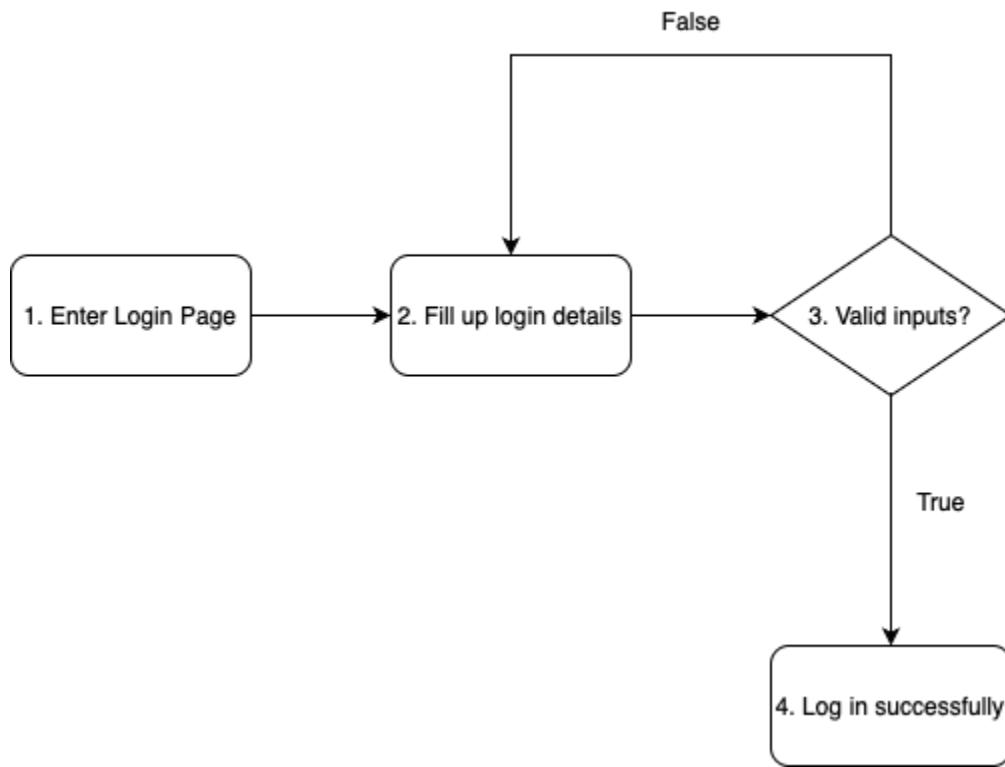


Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4,5,6 (Baseline)	valid_input = True, fields_filled = True	Toast: "Success Account created! OK" Redirected to login page	Toast: "Success Account created! OK" Redirected to login page	Pass
1,2,3,4,2,3,4,5,6	valid_input = False, fields_filled = True valid_input = True, fields_filled = True	Toast: "Registration failed Firebase: Error (auth/invalid-email) OK" Toast: "Success Account created!"	Toast: "Registration failed Firebase: Error (auth/invalid-email) OK" Toast: "Success Account created!"	Pass

		OK" Redirected to login page	OK" Redirected to login page	
1,2,3,4,2,3,4,5,2,3,4,5,6	valid_input = False, fields_filled = False, valid_input = True, fields_filled = True	Toast: "Missing Fields Please fill in all fields. OK" Toast: "Success Account created! OK" Redirected to login page	Toast: "Missing Fields Please fill in all fields. OK" Toast: "Success Account created! OK" Redirected to login page	Pass

CC Value = 1 + 2 = 3

User Account Login

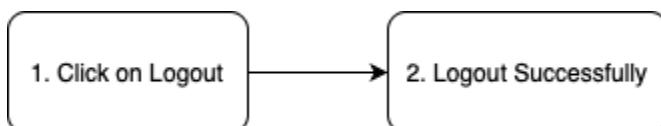


Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4	valid_input = True,	Toast: "Login	Toast: "Login	Pass

(Baseline)		Successful Bob Tan" Redirected to Home Page	Successful Bob Tan" Redirected to Home Page	
1,2,3,2,3,4	valid_input = False, valid_input = True	Toast: "Login Failed Firebase: Error (auth/invalid-cre dential). OK" Toast: "Login Successful Bob Tan" Redirected to Home Page	Toast: "Login Failed Firebase: Error (auth/invalid-cre dential). OK" Toast: "Login Successful Bob Tan" Redirected to Home Page	Pass

CC Value = 1 + 1 = 2

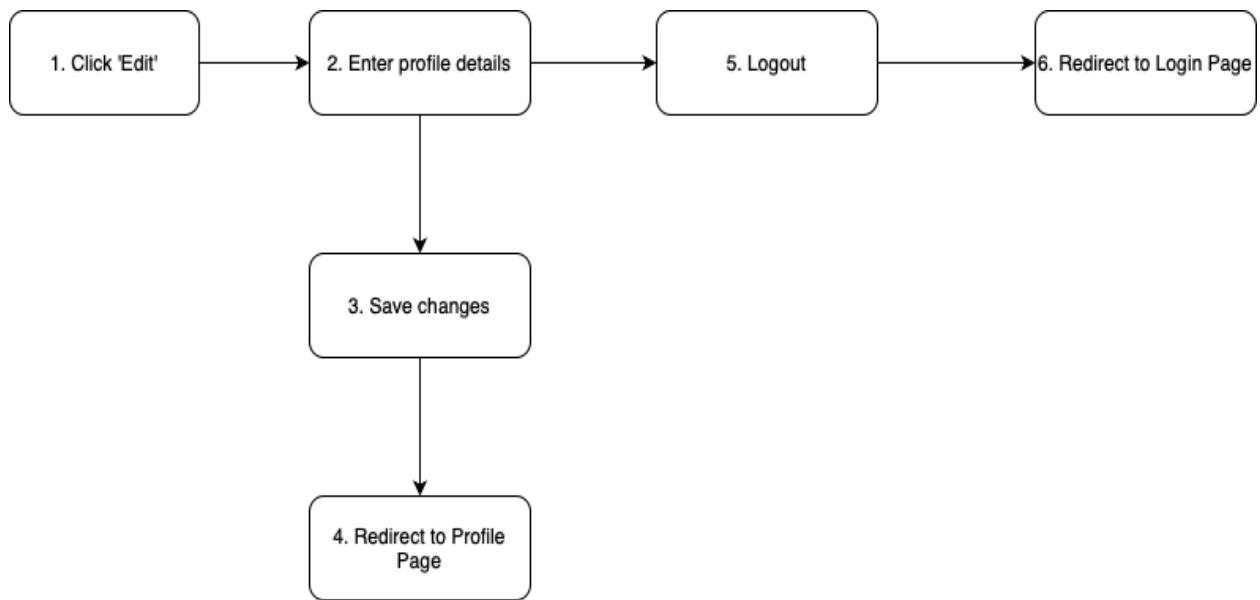
User Account Logout



Path Taken	Test Cases	Expected	Actual	Passed
1,2 (Baseline)	logged_Out = True	Toast: "Confirm Logout Are you sure you want to log out? Cancel Logout" Redirect to login page	Toast: "Confirm Logout Are you sure you want to log out? Cancel Logout" Redirect to login page	Pass

CC Value = 1

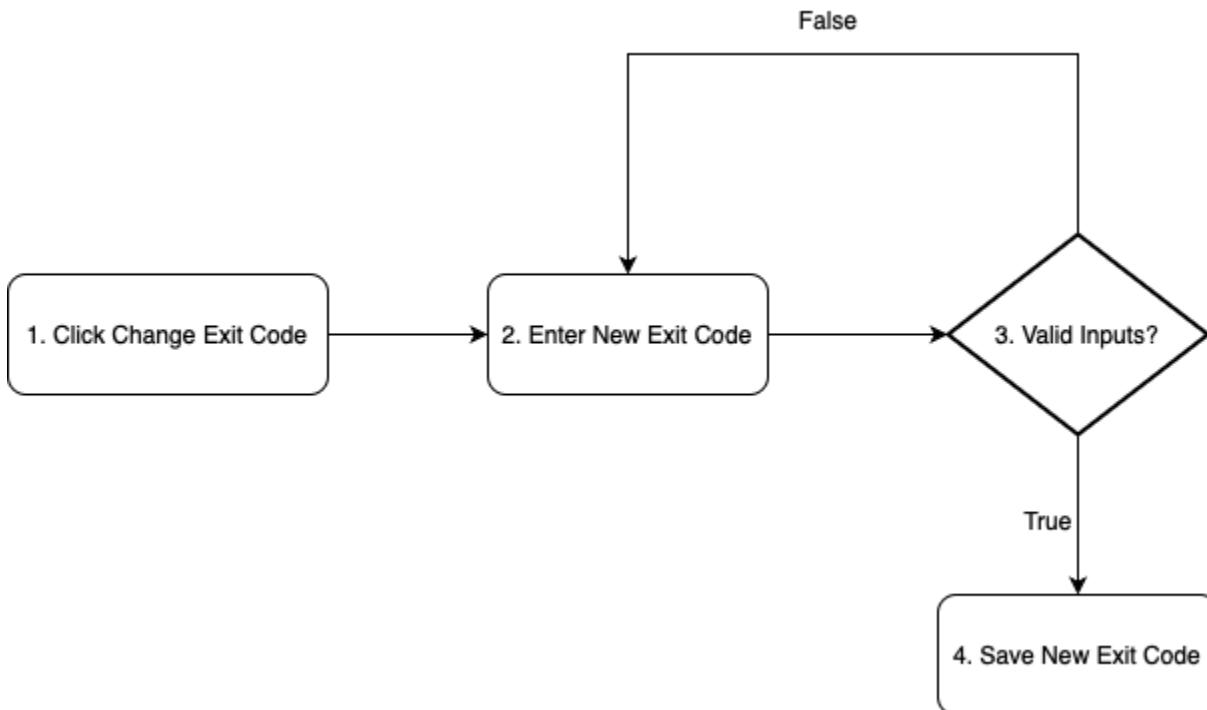
Edit Profile



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4 (Baseline)	save_Changes = True	Toast: "Success Profile updated "	Toast: "Success Profile updated "	Pass
1,2,5,6	logged_Out = True	Toast: "Confirm Logout Are you sure you want to log out? Cancel Logout" Redirect to login page	Toast: "Confirm Logout Are you sure you want to log out? Cancel Logout" Redirect to login page	Pass

CC Value = 1 + 1 = 2

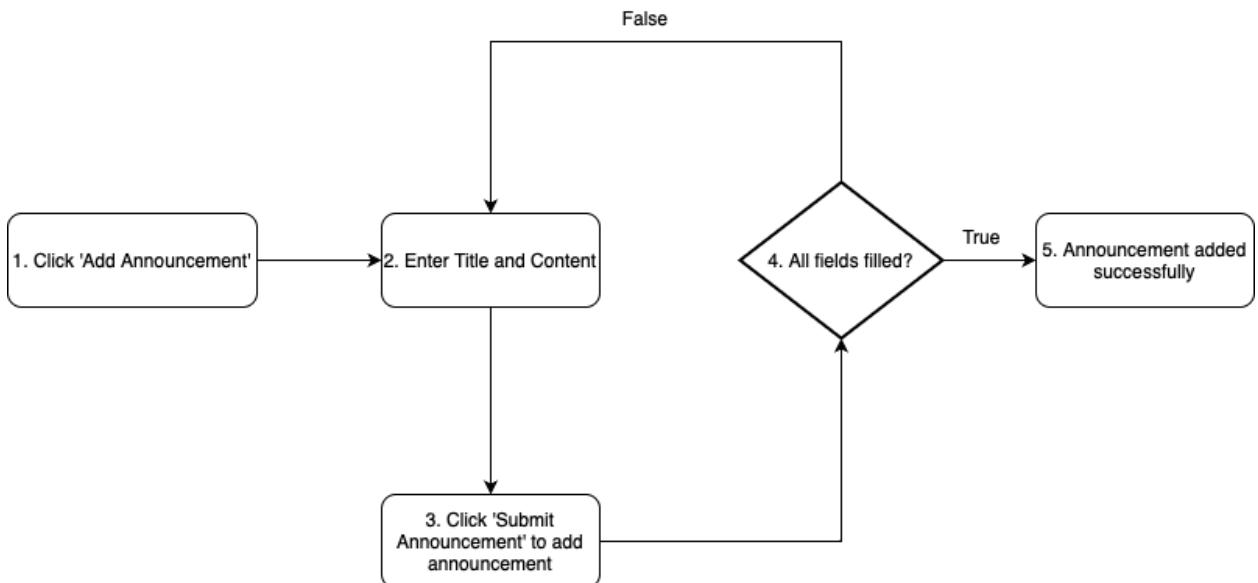
Change Exit Code



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4 (Baseline)	valid_input = True,	Toast: "Success Password updated. OK" Return to child details page	Toast: "Success Password updated. OK" Return to child details page	Pass
1,2,3,2,3,4	valid_input = False, valid_input = True	Toast: "Incorrect Code The exit code entered is incorrect. OK" Toast: "Success Password updated. OK" Return to child details page	Toast: "Incorrect Code The exit code entered is incorrect. OK" Toast: "Success Password updated. OK" Return to child details page	Pass

CC Value = 1 + 1 = 2

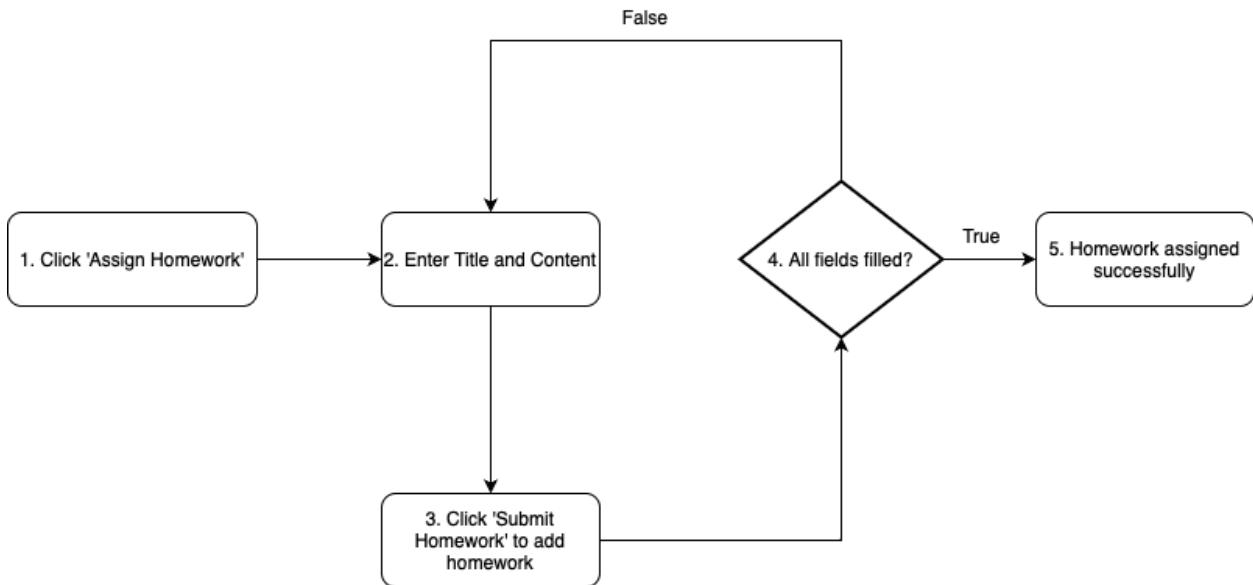
Add Announcement



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4,5 (Baseline)	fields_filled = True,	Toast: "Announcement added successfully" Redirected to class page	Toast: "Announcement added successfully" Redirected to class page	Pass
1,2,3,4,2,3,4,5	fields_filled = False, fields_filled = True	Toast:"Missing Fields Please enter both name and content OK" Toast: "Announcement added successfully" Redirected to class page	Toast:"Missing Fields Please enter both name and content OK" Toast: "Announcement added successfully" Redirected to class page	Pass

CC Value = 1 + 1 = 2

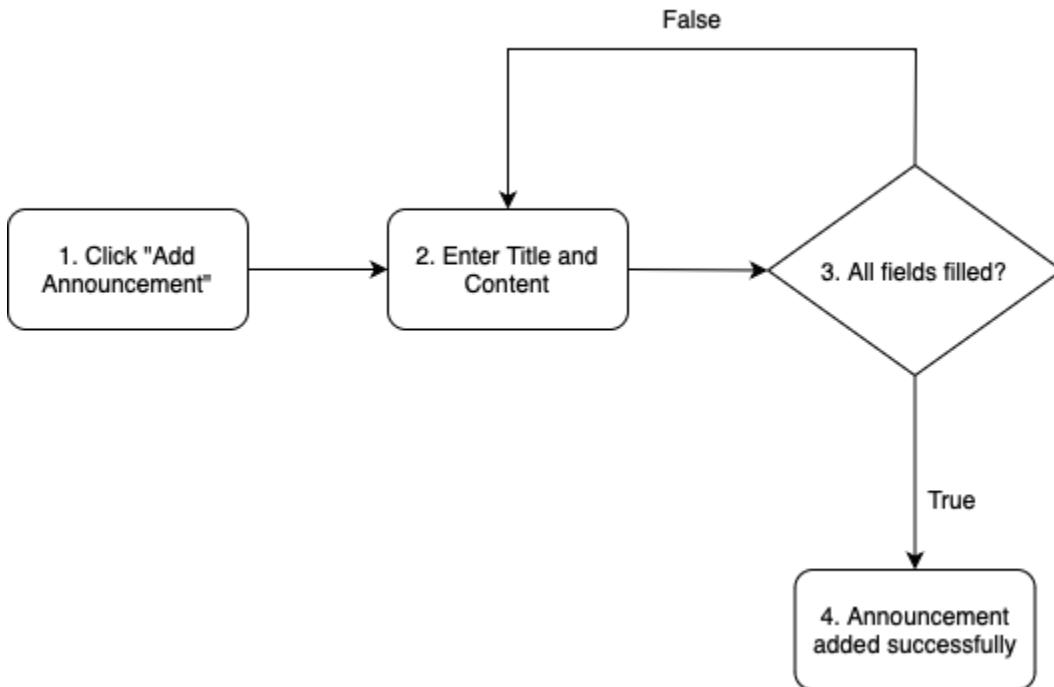
Assign Homework



Path Taken	Test Cases	Expected	Actual	Passee d
1,2,3,4,5 (Baseline)	fields_filled = True,	Toast: "Homework assigned successfully OK" Redirect to class page	Toast: "Homework assigned successfully OK" Redirect to class page	Pass
1,2,3,4,2,3,4,5	fields_filled = False, fields_filled = True,	Toast: "Missing Fields Please fill in all fields. OK" Toast: "Homework assigned successfully OK" Redirect to class page	Toast: "Missing Fields Please fill in all fields. OK" Toast: "Homework assigned successfully OK" Redirect to class page	Pass

CC Value = 1 + 1 = 2

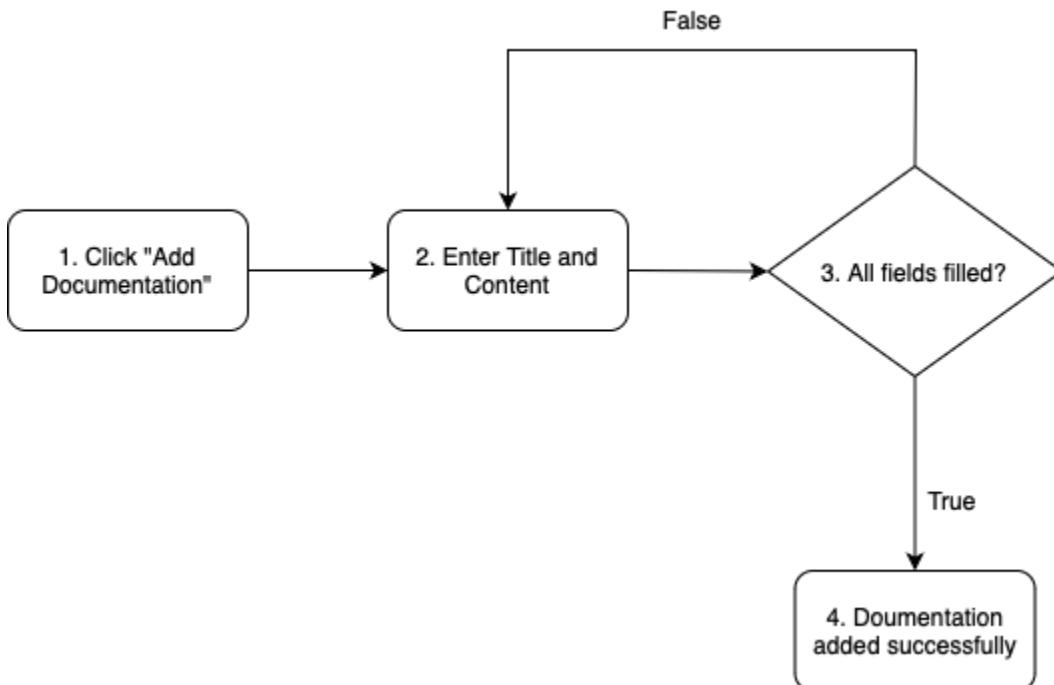
Add Announcement



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4 (Baseline)	fields_filled = True	Toast: "Success Announcement added. OK"	Toast: "Success Announcement added. OK"	Pass
1,2,3,2,3,4	fields_filled = False fields_filled = True	Toast: "Missing Fields Please enter both name and content OK" Toast: "Success Announcement added. OK"	Toast: "Missing Fields Please enter both name and content OK" Toast: "Success Announcement added. OK"	Pass

CC Value = 1 + 1 = 2

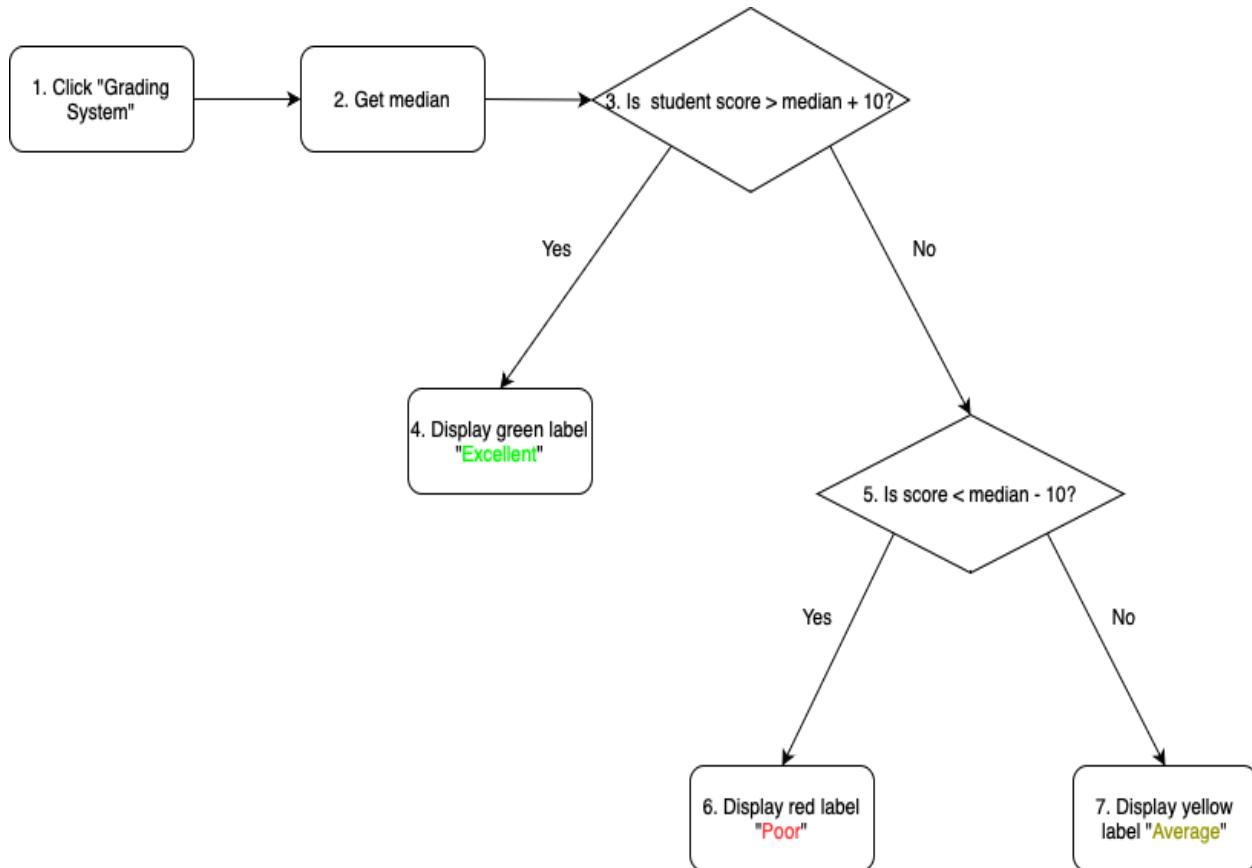
Add Documentation



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,4 (Baseline)	fields_filled = True	Toast: "Success Documentation added! OK"	Toast: "Success Documentation added! OK"	Pass
1,2,3,2,3,4	fields_filled = False fields_filled = True	Toast: "Error Please enter a title. OK" Toast: "Success Documentation added! OK"	Toast: "Error Please enter a title. OK" Toast: "Success Documentation added! OK"	Pass

CC Value = 1 + 1 = 2

Grading System



Path Taken	Test Cases	Expected	Actual	Passed
1,2,3,5,7 (Baseline)	Student score within ±10 range of median (Baseline)	Display yellow label "Average"	Display yellow label "Average"	Pass
1,2,3,4	Student score > median + 10	Display green label "Excellent"	Display green label "Excellent"	Pass
1,2,3,5,6	Student score < median - 10	Display red label "Poor"	Display red label "Poor"	Pass

$$\text{CC Value} = 7 \cdot 7 + 2(1) = 2$$