**NAME:** Rabeea Imran

**ID:** F2023065110

**SUBJECT:** Software Engineering

**SECTION:** W6

**StudyHub: Online Class and Education Management App**

**Project Overview:**

StudyHub is an online platform designed for virtual learning and class management. It allows teachers and students to connect through video calls, share study materials, track attendance, and record sessions. The app is built to support educators in managing online classes efficiently and creating an engaging learning environment for students.

**Problem Statement:**

Many students and teachers face challenges with existing online meeting platforms. These platforms often lack features designed for educational purposes, like tools for interactive learning or classroom management. This makes online learning less effective and harder to manage for educators and students. StudyHub addresses this gap by providing a complete solution for virtual classes, making learning more interactive and organized.

**Objectives:**

* **Interactive Learning:** Provide tools like virtual whiteboards, screen sharing, and real-time quizzes for effective learning.
* **Class Management:** Enable features such as attendance tracking, scheduling, and class recordings for seamless organization.
* **Accessibility:** Make the app easy to use on multiple devices like mobile phones, tablets, and desktops.
* **Collaboration:** Facilitate group discussions and file sharing through in-app chat and collaboration tools.
* **Secure Environment:** Protect user data and ensure a safe and private platform for communication.

**Goals:**

* The main goals of the StudyHub project are:
* **Simplify Online Education:** Offer tools to make virtual learning easy for students and teachers.
* **Improve Engagement:** Provide interactive features like polls, Q&A sessions, and breakout rooms.
* **Enhance Management:** Help educators manage their classes with attendance reports, session recordings, and schedules.
* **Support Scalability:** Build the app to handle large user bases, including schools and colleges.
* **Measure Impact:** Track the effectiveness of the app through user feedback, engagement, and performance metrics.

**Scope:**

* **Video Conferencing:** High-quality video and audio calls for virtual classrooms.
* **Attendance and Reports:** Automatic attendance tracking with detailed reports for teachers.
* **Collaboration Tools:** Virtual whiteboards, chat, and breakout rooms for group work.
* **Resource Sharing:** Upload and share files, study materials, and assignments.
* **Notifications and Alerts:** Reminders for classes, assignments, and deadlines.
* **Class Recording:** Save and access recorded sessions for later review.
* **Integration:** Compatible with platforms like Google Classroom or other LMS systems.
* **Security:** End-to-end encryption to ensure user data privacy.

**Domain Analysis:**

**Customer:**

The primary customers are:

* **Students** who attend online classes.
* **Teachers** who organize and conduct classes.

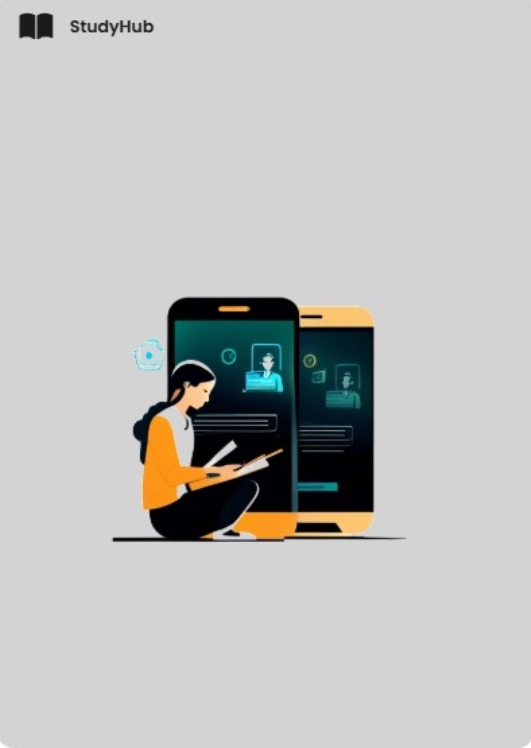
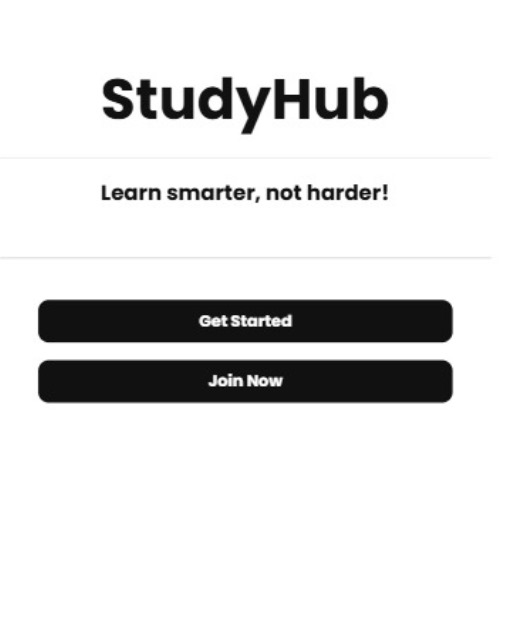
**Stakeholders:**

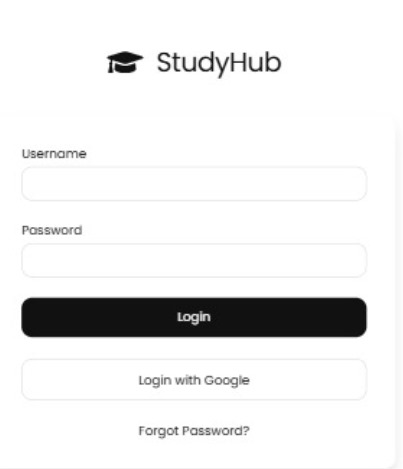
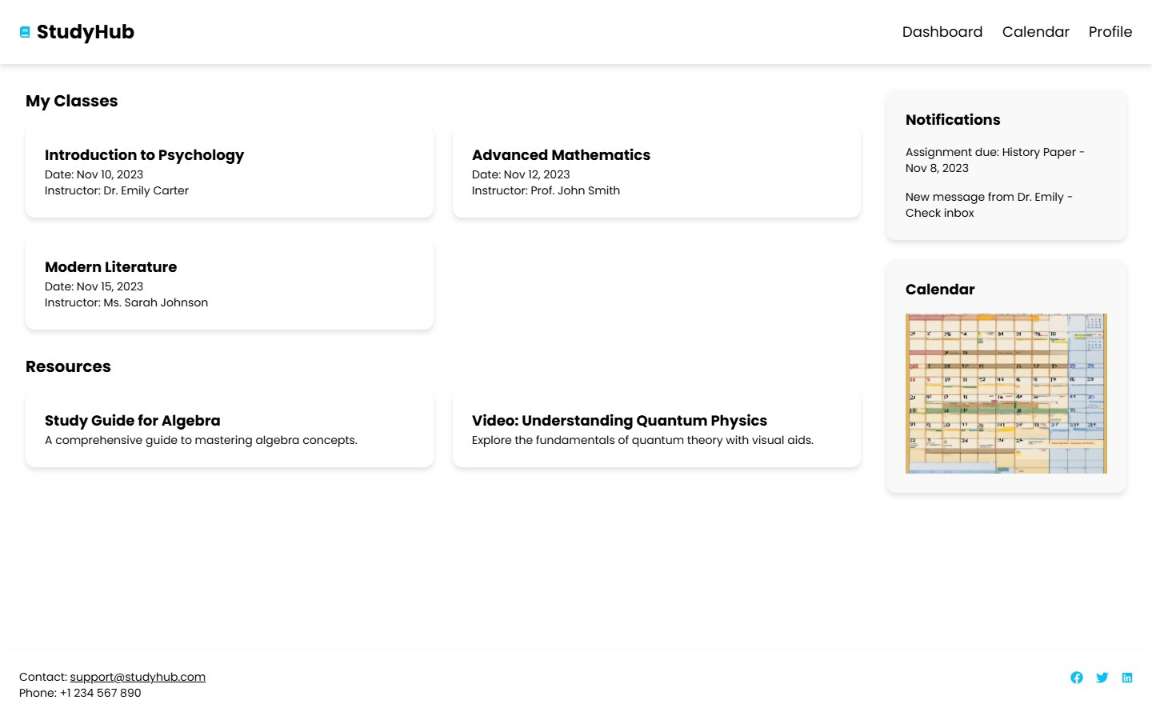
The key stakeholders include:

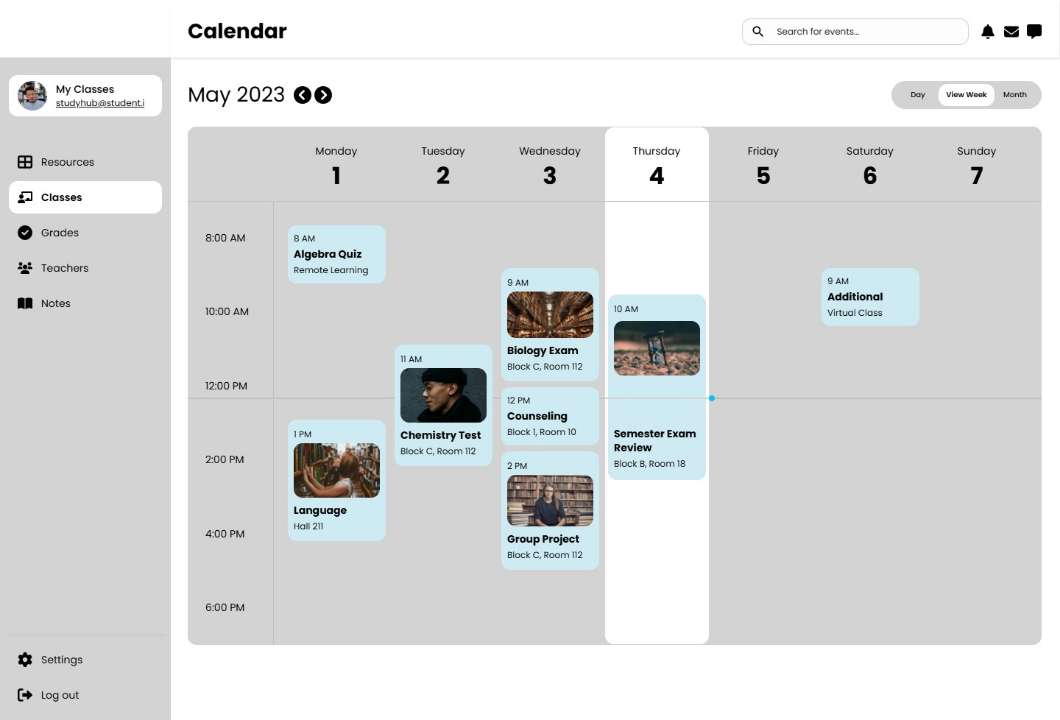
* **Educational Institutions:** Schools, colleges, and coaching centers that use the app for online education.
* **Parents:** Who want their children to have a smooth and effective online learning experience.
* **Developers:** The team responsible for building and maintaining the app.
* **Investors or Sponsors:** Organizations or individuals funding the project.

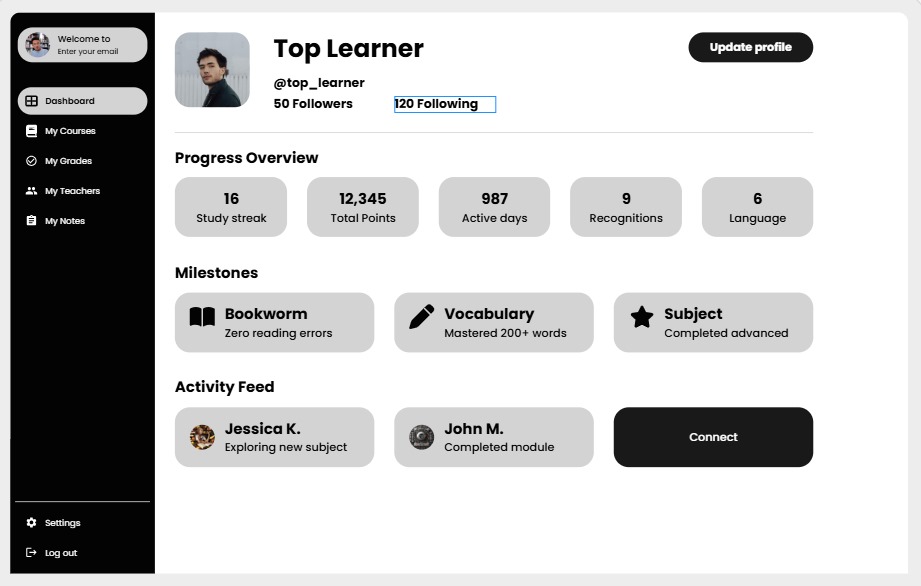
**Deployment:**

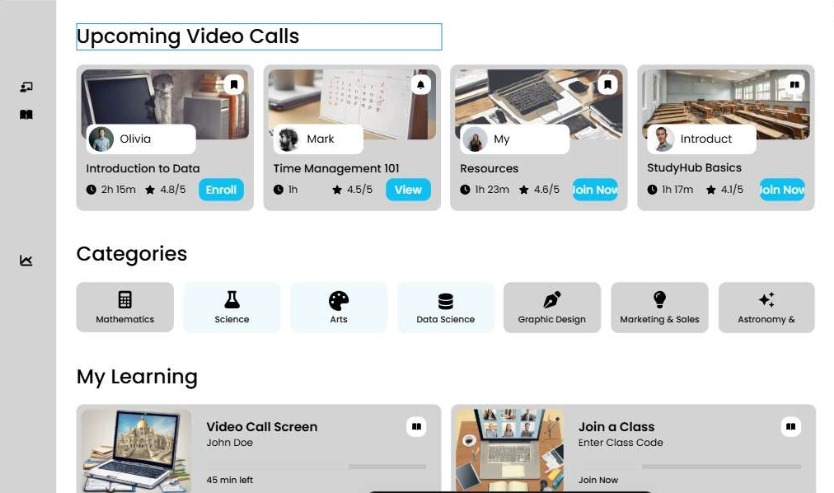
* **App Stores:** Publish the app on Google Play Store and Apple App Store for global accessibility.
* **Web Version:** Provide a web-based interface for users who prefer desktop access.
* **Beta Testing:** Release a beta version to gather user feedback and fix any issues before full deployment.
* **Monitoring and Maintenance:** Use tools like Google Analytics to track app performance and implement regular updates.
* **Updates:** Regularly release new features, performance improvements, and bug fixes.

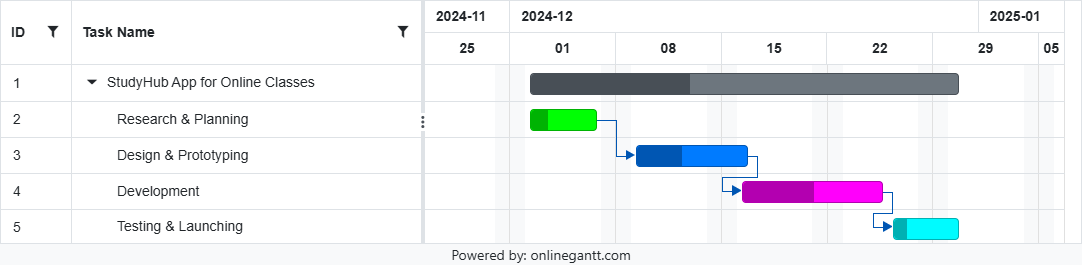










**Gang Chart:**

**Functional Requirements**

**1. User Authentication**

* **Secure Login**: Implement secure login via email, social media accounts, and institutional accounts.
* **Password Recovery & Two-Factor Authentication**: Provide password recovery options and enable two-factor authentication.

**2. Video Conferencing**

* **High-Quality Video and Audio**: Support high-definition video and audio calls with stable connections.
* **Participant Limit**: Set scalable limits on participants per session.

**3. Meeting Scheduling**

* **Schedule Sessions**: Allow users to schedule study sessions and classes in advance.
* **Calendar Reminders**: Integrate with external calendars and send automated reminders.

**4. Screen Sharing**

* **Screen Sharing**: Enable full and partial screen sharing.
* **Annotation Tools**: Provide tools for annotating shared screens.

**5. Recording**

* **Session Recording**: Allow hosts to record sessions and store them in the cloud.
* **Download Option**: Provide an option for participants to download recorded sessions.

**6. Real-Time Chat**

* **Text Messaging**: Facilitate text-based communication during sessions.
* **File & Link Sharing**: Allow sharing of files and links within the chat.

**7. Participant Management**

* **Host Controls**: Enable hosts to manage participants and assign roles.
* **Attendance & Breakout Rooms**: Track attendance and provide breakout room functionality.

**8. Virtual Hand Raise**

* **Raise Hand Feature**: Allow participants to signal when they want to speak.
* **Host Notifications**: Notify hosts when a participant raises their hand.

**9. Notifications & Alerts**

* **Session Notifications**: Notify users about upcoming sessions and important updates.
* **Message Alerts**: Alert users about new messages with customizable preferences.

**10. Multi-Device Support**

* **Cross-Platform Compatibility**: Ensure the app works seamlessly on all devices.
* **Responsive Design**: Implement responsive design and provide native apps for major platforms.

**11. File Sharing**

* **Upload and Share Files**: Enable users to upload and share study materials.
* **File Storage & Types**: Provide cloud storage for common file types with access controls.

**12. Feedback Mechanism**

* **Feedback Forms**: Offer feedback forms at the end of each session.
* **Anonymous Feedback & Analysis**: Allow anonymous feedback and analyze it to improve content.

**Non-Functional Requirements:**

1. **Performance**:

Ensure the platform can handle many concurrent users. Maintain low latency for real-time interactions.

1. **Security**:

Use end-to-end encryption for communications and file transfers. Ensure compliance with data protection regulations like GDPR and FERPA.

1. **Usability**:

Design an intuitive and easy-to-navigate interface. Ensure accessibility features for users with disabilities.

1. **Reliability**:

Aim for high uptime to ensure availability. Implement regular data backups and disaster recovery plans.

1. **Maintainability**:

Use a modular architecture for easy updates and maintenance. Provide comprehensive documentation for users and developers.

1. **Compatibility**:

Ensure compatibility with major web browsers. Support major operating systems (Windows, macOS, Linux, iOS, Android).

**List of Actors:**

**Primary Actors:**

1. **Students**
   1. Participate in study sessions and classes.
   2. Share files and links during sessions.
   3. Use real-time chat for communication.
   4. Provide feedback on sessions.
2. **Educators/Teachers**
   1. Host and schedule study sessions and classes.
   2. Share screens and record sessions.
   3. Manage participants and assign roles.
   4. Collect and analyze feedback to improve content.

**Secondary Actors:**

1. **Administrators**
   1. Manage user accounts and authentication.
   2. Monitor platform usage and performance.
   3. Ensure compliance with security and data protection regulations.
2. **IT Support**
   1. Provide technical assistance to users.
   2. Maintain and update the platform.
   3. Handle backup and recovery processes.

**Tertiary Actors:**

1. **External Calendar Services (e.g., Google Calendar, Outlook)**
   1. Sync scheduled sessions with users' calendars.
   2. Send reminders and notifications.
2. **Cloud Storage Providers**
   1. Store session recordings and shared files.
   2. Ensure secure access and data integrity.

**Use Cases for StudyHub:**

1. **Log In**
   * Authenticate user credentials to access the platform.
   * Enable secure login through various methods.
2. **Host a Session**
   * Schedule and start a virtual class or study session.
   * Set session parameters like participant limits and roles.
3. **Join a Session**
   * Allow learners and guest users to enter scheduled sessions.
   * Provide options for direct links or codes for session access.
4. **Manage Participants**
   * Host can mute/unmute participants, assign roles, or remove attendees.
   * Track participant behavior and manage breakout rooms.
5. **Screen Sharing**
   * Share entire screens or specific applications during a session.
   * Use annotation tools for interactive presentations.
6. **Record Sessions**
   * Record class content for future playback.
   * Store recordings securely and provide download options.
7. **Send Notifications**
   * Alert users about upcoming sessions or updates.
   * Customize notification preferences for users.
8. **Upload and Share Files**
   * Upload study materials and share them with participants.
   * Support various file types and provide cloud storage.
9. **Raise Hand**
   * Learners can indicate a question or request to speak.
   * Notify the host when a hand is raised.
10. **Playback Recordings**
    * Access and review session recordings.
    * Provide playback controls and download options.
11. **Track Attendance**
    * Monitor learner participation during sessions.
    * Generate attendance reports for hosts.
12. **Log Out**
    * Exit the platform securely.
    * Ensure all user data is cleared upon logout.

**System use case diagram:**

**StudyHub Management System**

Instructor/Tutor

Learner

**Login Use Case**



**Learner**

**Tutor**

**Use Case Table for Login Process**

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Use Case Name** | Login |
| **Created By** | Rabeea Imran |
| **Date Created** | 24-05-24 |
| **Last Updated By** | 03-06-24 |
| **Last Revision Date** | 24-06-24 |
| **Actors** | |  | | --- | | **Primary:** Student, Tutor |  |  |  | | --- | --- | |  | **Secondary:** System, Administrator | |
| **Description** | This use case allows a user to log in to the StudyHub app using their credentials (email and password). If the user forgets their password, they can reset it through the system’s password recovery process. The system verifies the email and password, granting access to the user’s personal dashboard or home page. |
| **Trigger** | |  | | --- | |  |  |  | | --- | | The user attempts to log in, register, or reset their password | |
| **Preconditions** | 1. The user must have a valid email for registration. 2. The user must be logged in or attempt to reset their password if they forgot it. |
| **Post conditions** | 1. The user is successfully logged in or redirected to the password reset page. 2. If password reset is triggered, the user receives a password reset link. |
| **Normal Flow** | 1. The user navigates to the login page of the StudyHub app.  2. The user enters their email and password.  3. The system verifies the credentials (email and password).  4. If credentials are valid, the user is granted access to the **dashboard** or home page.  5. If credentials are invalid, the system displays an error message prompting the user to re-enter credentials. |
| **Alternative Flows** | 1. **Forgot Password**: The user clicks "Forgot Password", enters their email, and receives a reset link. 2. **Re-Login**: If the user logs out and returns later, they enter their email and password to log back in. |
| **Exceptions** | 1. If the email/password is invalid, the system displays an error message. 2. If the email is not registered, the system prompts the user to check their credentials or register. |
| **Includes** | |  | | --- | |  |  |  | | --- | | 1. Register Account (for first-time users). 2. Forgot Password (if user cannot remember password) | |
| **Frequency of Use** | |  | | --- | | **High:** Frequent use by students and tutors logging in and out of the system. |  |  | | --- | |  | |
| **Special Requirements** | 1. Password must meet security criteria (e.g., minimum length, complexity). 2. Email must be unique for each registration. |
| **Assumptions** | 1. The user has internet access. 2. The email entered by the user is valid. |
| **Notes and Issues** | 1. The system must securely hash passwords before storing them.  2. Ensure email verification for registration and password reset to avoid fraudulent activity.  3. Role-based login access (Student/Tutor) could be considered. |
| **Use Case ID** | UC-1 |

**Session Use Case**

**Learner**

**Tutor**

| **Field** | **Details** |
| --- | --- |
| **Use Case Name** | **Schedule Meeting** |
| **Created By** | **Rabeea Imran** |
| **Date Created** | **09-01-25** |
| **Last Updated By** | **09-01-25** |
| **Last Revision Date** | **09-01-25** |
| **Actors** | **Primary:** Student, Tutor |
|  | **Secondary:** System, Administrator |
| **Description** | This use case allows a user (student or tutor) to schedule a meeting within the StudyHub app. The system enables the user to select a date, time, and duration for the meeting, and provides options to invite other participants or tutors. The system handles scheduling, availability checking, and sends notifications to attendees. |
| **Trigger** | The user attempts to schedule a meeting |
| **Preconditions** | 1. The user must be logged in to schedule a meeting.  2. The user must have a valid account and role (student/tutor).  3. The user must have internet access. |
| **Postconditions** | 1. The meeting is successfully scheduled and added to the user’s calendar.  2. Invitations are sent to participants, and the meeting details are updated in the system. |
| **Normal Flow** | 1. The user logs into the StudyHub app.  2. The user navigates to the meeting scheduling page.  3. The user selects the date, time, and duration for the meeting.  4. The system checks the availability of the user and other participants.  5. If available, the system confirms the meeting and sends invitations.  6. If not available, the system suggests alternative times. |
| **Alternative Flows** | 1. **Meeting Reschedule:** If a conflict arises, the user reschedules the meeting by selecting an available time.  2. **Cancelled Meeting:** If the user cancels the meeting, notifications are sent to participants, and the meeting is removed from the system. |
| **Exceptions** | 1. If the user’s selected time is unavailable, the system displays an error message and suggests alternative times.  2. If the user does not have permission to schedule a meeting, the system prompts the user to contact an administrator. |
| **Includes** | 1. **Check Availability:** To ensure the user and participants are free at the selected time.  2. **Send Invitations:** To notify participants of the scheduled meeting. |
| **Frequency of Use** | **Moderate to High:** Frequently used by students and tutors scheduling meetings for classes or discussions. |
| **Special Requirements** | 1. The system must be able to handle time zone differences for users in various regions.  2. The system should provide notifications for meeting reminders. |
| **Assumptions** | 1. The user has internet access.  2. The system can check real-time availability of participants. |
| **Notes and Issues** | 1. Ensure that scheduling conflicts are handled smoothly, and alternative times are suggested to the user.  2. Ensure notifications are sent in real-time to all meeting participants. |
| **Use Case ID** | **UC-2** |

**Screen Share Use Case**



**Learner**

**Tutor/Host**

| **Field** | **Details** |
| --- | --- |
| **Use Case Name** | Screen Sharing |
| **Created By** | Rabeea Imran |
| **Date Created** | 09-01-25 |
| **Last Updated By** | 09-01-25 |
| **Last Revision Date** | 09-01-25 |
| **Actors** | **Primary**: Student, Tutor |
|  | **Secondary**: System, Administrator |
| **Description** | This use case allows a user (student or tutor) to share their screen with other participants during a live meeting or session on the StudyHub app. The system facilitates the sharing of the user’s screen, enabling collaboration, presentation, and sharing of materials. |
| **Trigger** | The user selects the option to share their screen during a meeting |
| **Preconditions** | 1. The user must be logged into the StudyHub app.  2. The user must be in an active meeting or session.  3. The user must have a screen sharing option enabled by the system. |
| **Postconditions** | 1. The user’s screen is successfully shared with other meeting participants.  2. The system ensures that all participants can view the screen in real-time. |
| **Normal Flow** | 1. The user logs into the StudyHub app.  2. The user enters an active meeting or session.  3. The user clicks the "Share Screen" button.  4. The system activates screen sharing and shares the user’s screen with other participants.  5. The user stops screen sharing when done.  6. The system stops the screen sharing and notifies participants. |
| **Alternative Flows** | 1. **No Permission to Share Screen:** If the user doesn’t have permission to share their screen, the system notifies them and suggests contacting the administrator.  2. **Screen Share Resumed:** If the user accidentally stops sharing their screen, they can click "Resume Screen Sharing" to continue. |
| **Exceptions** | 1. If the system detects an issue with the screen-sharing feature (e.g., network connectivity problems), the system notifies the user of the failure.  2. If the user leaves the meeting, the screen sharing is automatically stopped. |
| **Includes** | 1. **Activate Screen Sharing:** Ensures the screen is successfully shared.  2. **Stop Screen Sharing:** Allows the user to end the sharing session. |
| **Frequency of Use** | Moderate: Used during collaborative meetings and presentations. |
| **Special Requirements** | 1. The system must ensure that only the intended screen or window is shared.  2. Screen sharing should be compatible across devices (PC, mobile). |
| **Assumptions** | 1. The user has internet access.  2. The user has the necessary permissions to share the screen.  3. The system supports screen sharing functionality on various devices. |
| **Notes and Issues** | 1. Ensure that sensitive information is not inadvertently shared by the user.  2. Implement security measures to prevent unauthorized screen sharing. |
| **Use Case ID** | UC-3 |

**Recording Use Case**



**Learner**

**Tutor**

| **Field** | **Details** |
| --- | --- |
| **Use Case Name** | Recording Lecture |
| **Created By** | Rabeea Imran |
| **Date Created** | 09-01-25 |
| **Last Updated By** | 09-01-25 |
| **Last Revision Date** | 09-01-25 |
| **Actors** | **Primary**: Tutor, Student |
|  | **Secondary**: System, Administrator |
| **Description** | This use case allows a tutor to record a lecture during a live session on the StudyHub app. The system facilitates the recording process and provides options to save, pause, or stop the recording. |
| **Trigger** | The tutor clicks on the "Record" button during the lecture session. |
| **Preconditions** | 1. The tutor must be logged into the StudyHub app.  2. The tutor must be in an active session.  3. The system must allow recording functionality. |
| **Postconditions** | 1. The lecture is successfully recorded.  2. The system saves the recording and provides a link for later viewing. |
| **Normal Flow** | 1. The tutor logs into the StudyHub app.  2. The tutor starts a live session.  3. The tutor clicks the "Record" button to begin recording.  4. The system starts recording the session.  5. The tutor clicks "Stop Recording" when done.  6. The system saves the recording and provides a link for viewing. |
| **Alternative Flows** | 1. **Recording Paused:** If the tutor pauses the recording, the system saves the current segment and allows the tutor to resume recording.  2. **Recording Failed:** If there is an issue with recording (e.g., system error), the system notifies the tutor. |
| **Exceptions** | 1. If the tutor clicks "Stop Recording" but the system detects a problem (e.g., storage full), the system displays an error message.  2. If the tutor leaves the session early, the system stops the recording automatically. |
| **Includes** | 1. **Pause Recording:** Allows the tutor to pause and resume the lecture recording.  2. **Stop Recording:** Finalizes the lecture recording and saves it. |
| **Frequency of Use** | Moderate: Used by tutors to record lectures and provide access to students for later review. |
| **Special Requirements** | 1. The system must ensure the recorded lecture is clear and free of technical issues.  2. The recording feature should be available across devices (PC, mobile). |
| **Assumptions** | 1. The tutor has internet access.  2. The system can handle the recording and saving process without technical failures. |
| **Notes and Issues** | 1. Ensure that recordings are stored securely.  2. Implement features to protect privacy and prevent unauthorized access to the recorded lectures. |
| **Use Case ID** | UC-4 |

**Message Use case**



**Learner**

**Tutor/Host**

| **Field** | **Details** |
| --- | --- |
| **Use Case Name** | Messages During Lecture |
| **Created By** | Rabeea Imran |
| **Date Created** | 09-01-25 |
| **Last Updated By** | 09-01-25 |
| **Last Revision Date** | 09-01-25 |
| **Actors** | **Primary**: Student, Tutor |
|  | **Secondary**: System, Administrator |
| **Description** | This use case allows students and tutors to send and receive messages during a live lecture session on the StudyHub app. The system provides real-time messaging functionality. |
| **Trigger** | A user (student or tutor) sends a message during a lecture session. |
| **Preconditions** | 1. The user must be logged into the StudyHub app.  2. The user must be in an active lecture session. |
| **Postconditions** | 1. The message is successfully sent and displayed to all participants.  2. The system stores the message for later review if needed. |
| **Normal Flow** | 1. The user logs into the StudyHub app.  2. The user joins an active lecture session.  3. The user types a message in the chat window.  4. The system sends the message and displays it in the chat.  5. All participants in the lecture can view the message in real-time. |
| **Alternative Flows** | 1. **Private Message:** If the user sends a private message, the system ensures that only the intended recipient can view it.  2. **Message Edited:** If a user edits a sent message, the system updates the message in the chat. |
| **Exceptions** | 1. If the system detects a connection issue, the message is queued and sent once the connection is restored.  2. If the user leaves the session, they can no longer send or receive messages. |
| **Includes** | 1. **Send Message:** Allows users to send messages during the lecture.  2. **Edit/Delete Message:** Allows users to modify or delete their sent messages. |
| **Frequency of Use** | High: Frequently used by students and tutors during lectures for communication and clarification. |
| **Special Requirements** | 1. The system must support real-time messaging with minimal latency.  2. The system should allow message moderation by the tutor or administrator. |
| **Assumptions** | 1. The user has internet access.  2. The system supports seamless real-time communication. |
| **Notes and Issues** | 1. Ensure message content is stored securely and respects user privacy.  2. Implement features to prevent spam and inappropriate messaging during sessions. |
| **Use Case ID** | UC-5 |

**Attendance Use Case**



**Learner**

**Tutor**

| **Field** | **Details** |
| --- | --- |
| **Use Case Name** | Attendance |
| **Created By** | Rabeea Imran |
| **Date Created** | 09-01-25 |
| **Last Updated By** | 09-01-25 |
| **Last Revision Date** | 09-01-25 |
| **Actors** | **Primary**: Student, Tutor |
|  | **Secondary**: System, Administrator |
| **Description** | This use case allows the host (tutor or administrator) to track the attendance of participants in a session. The system records attendance as participants join or leave the session and allows the host to view the list of attendees.. |
| **Trigger** | The user (tutor/administrator) starts the session and tracks participants joining or leaving. |
| **Preconditions** | 1. The user must be logged in to the StudyHub app.  2. The session must be in progress for attendance tracking.  3. The user must have permission to track attendance (e.g., tutor or administrator). |
| **Postconditions** | 1. The attendance list is updated in real-time with participants joining and leaving the session.  2. The attendance record is saved and available for review after the session ends. |
| **Normal Flow** | 1. The user starts the session as a tutor or administrator.  2. Participants join the session.  3. The system automatically tracks the presence of participants as they join or leave.  4. The tutor or administrator can view the attendance list.  5. The attendance list is saved after the session ends. |
| **Alternative Flows** | 1. **Manual Attendance Marking:** If the system fails to track a participant automatically, the host can manually mark attendance for that participant. |
| **Exceptions** | 1. If the system fails to track attendance due to a technical issue, the host is notified and asked to manually update the attendance list. |
| **Includes** | 1. **Track Join/Leave Events:** To record when participants join or leave the session.  2. **View Attendance List:** To allow the tutor/administrator to view the real-time list of attendees. |
| **Frequency of Use** | High: Frequently used during study sessions or classes to monitor attendance. |
| **Special Requirements** | 1. The system must ensure accurate tracking of attendance even with intermittent connectivity.  2. The attendance list should be available for review and export (e.g., CSV) after the session ends. |
| **Assumptions** | 1. The system is capable of detecting participant join/leave events in real-time.  2. The tutor or administrator has sufficient privileges to view and manage attendance. |
| **Notes and Issues** | 1. Ensure that the attendance tracking is accurate, even if there are connectivity issues.  2. Consider providing an option for exporting the attendance list for record-keeping or reporting. |
| **Use Case ID** | UC-6 |

**Virtual Hand Raise Use Case**



**Learner**

**Tutor/Host**

**Notification Use Case**

**Learner**

**Tutor**

**Feedback UseCase**



**Learner**

**Tutor/Host**

**Extend use case:**

<<extend>>

<<extend>>

<<extend>>

<<extend>>

**Instructor/Tutor**

<<extend>>

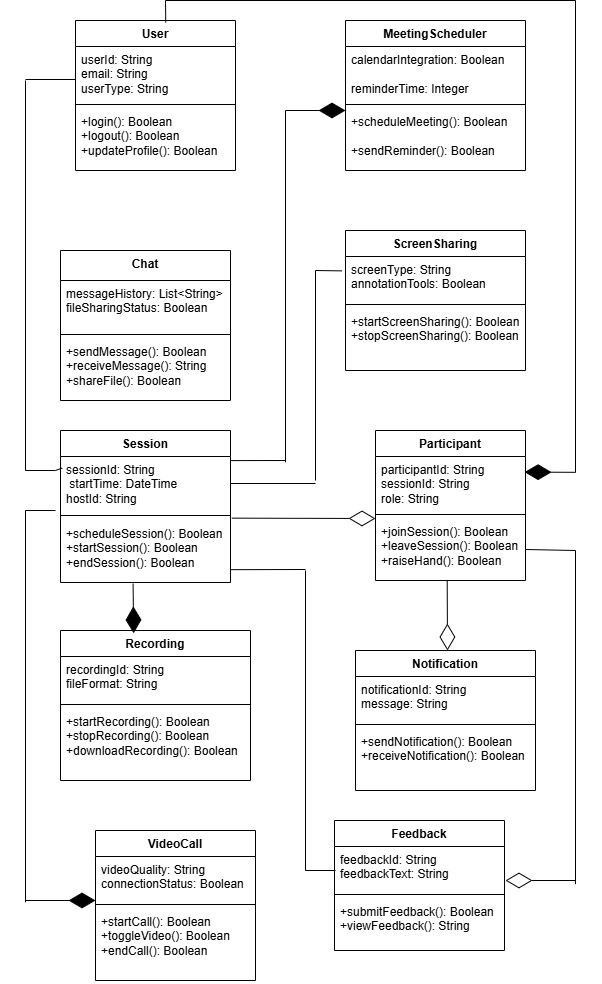
<<extend>>

<<extend>>

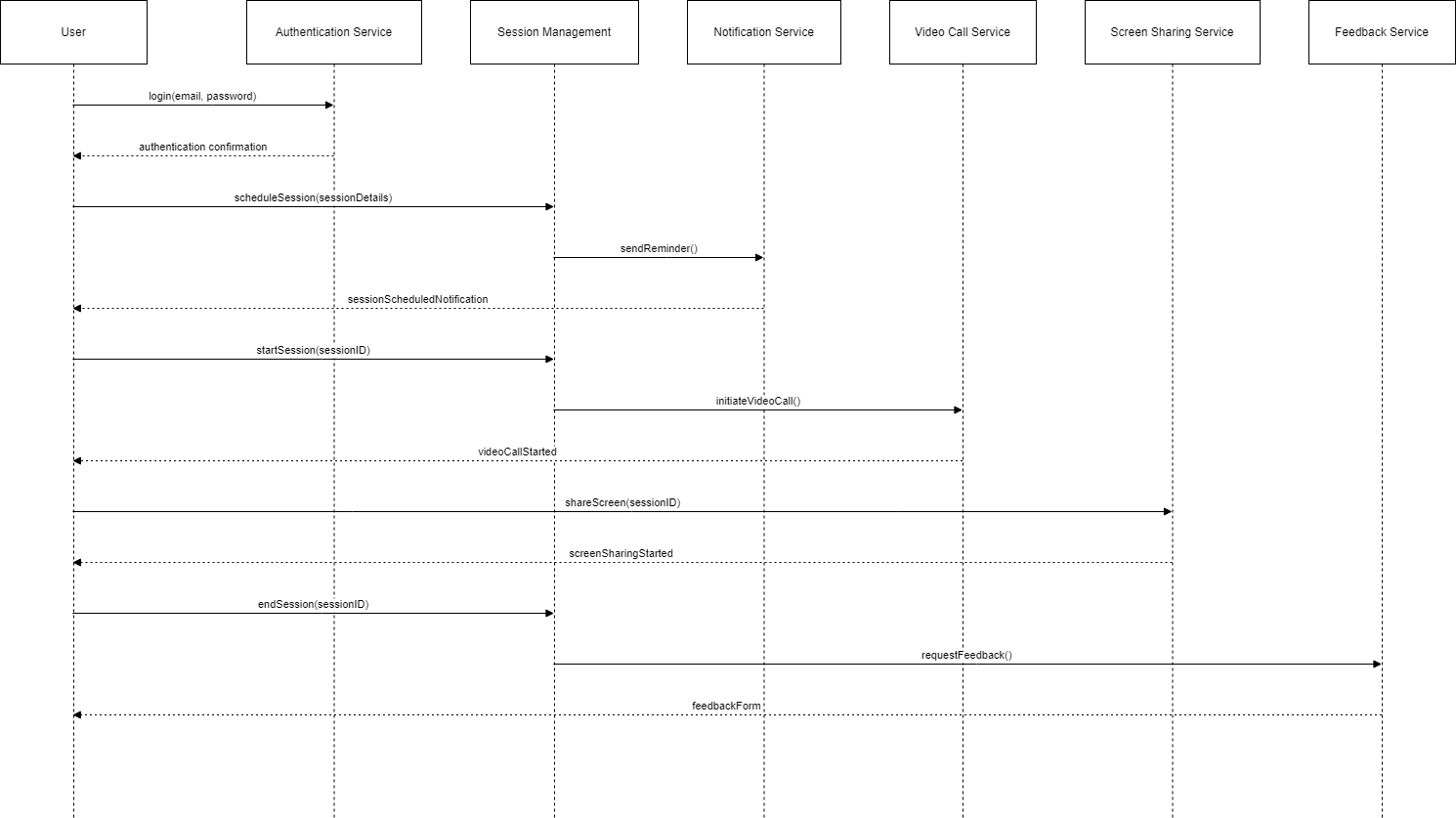
**Learner/Student**

<<extend>>

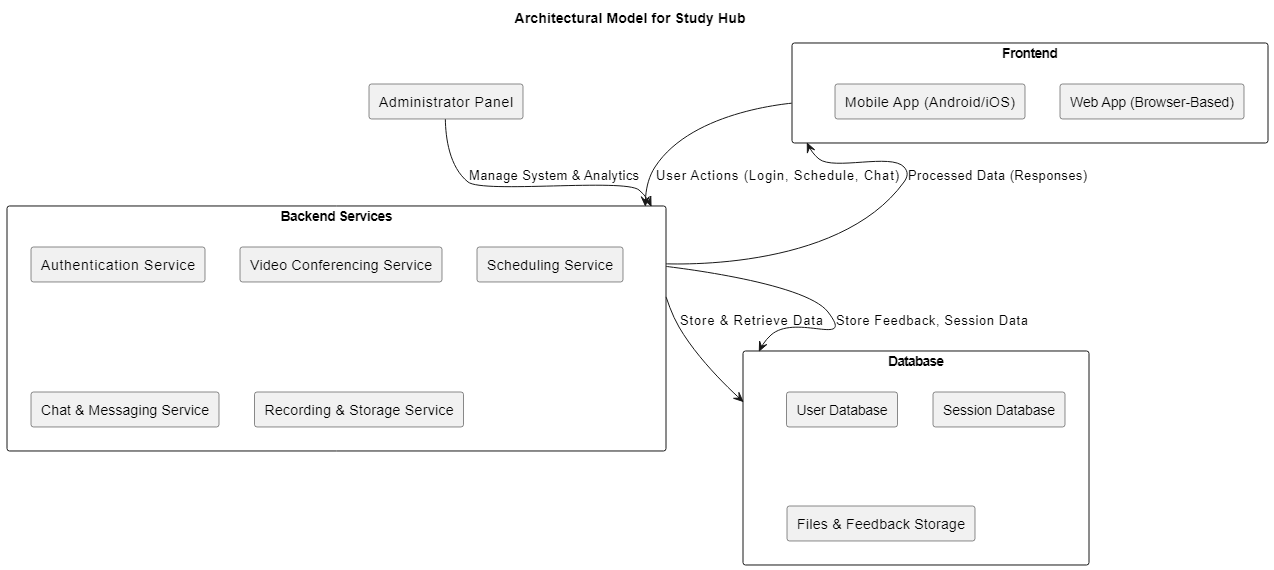
<<extend>>

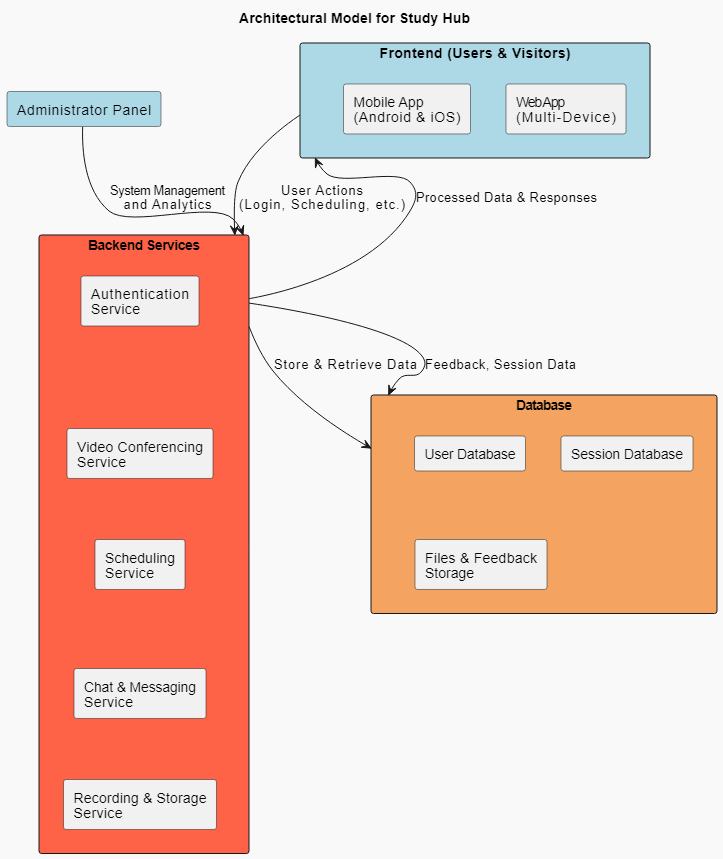


**Sequence Diagram:**

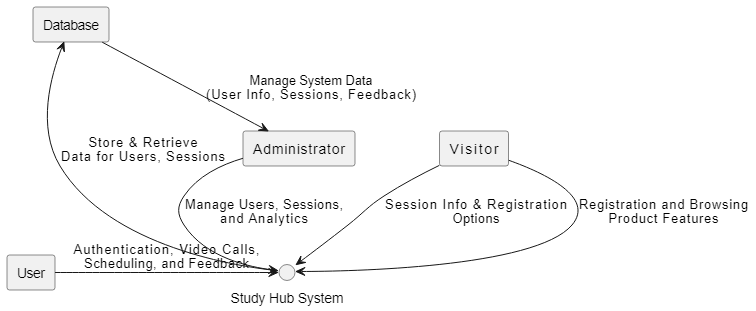


**Architecture Diagram:**





**DFD LEVEL 0:**



**DFD LEVEL 1:**

