# StudySphere: Collaborative Study Platform for Android

## Overview

**StudySphere** is an Android application designed to enhance collaborative studying among students. It enables users to create course-specific study spaces or join existing ones using unique course IDs. Within these spaces, users can organize notes, plan study sessions, and communicate via real-time group chat. The app leverages Firebase for backend services and integrates with Google services for calendar functionality, delivering a robust and user-friendly experience.

## Core Features

- **User Authentication**: Secure login and registration system.
- **Dashboard**: A central hub displaying enrolled courses with options to create or join courses.
- **Course Spaces**: Dedicated areas for each course, including:
  - Real-time group chat supporting text, images, and PDFs.
  - Folder-based notes organization for PDFs, images, and text files.
  - Study session planning with RSVP and Google Calendar integration.
- **Course Creation**: Users can create new courses and become their admins.
- **Course Joining**: Users can join existing courses by entering unique course IDs.
- **Logout**: Option to exit the authenticated session.

## Technical Stack

- **Frontend**: Android app developed using Kotlin or Java.

- **Backend**:
    - **Firebase Authentication**: For user login and registration.
    - **Firestore**: For storing course data, notes, and study session details.
    - **Firebase Realtime Database**: For real-time group chat functionality.
    - **Firebase Storage**: For file uploads (notes and chat attachments).
    - **Firebase Cloud Functions**: For secure Google Calendar API integration.
- **Libraries**:
    - Android Navigation Component for screen navigation.
    - ViewModels and LiveData for data management and UI updates.
    - Google Sign-In API for Calendar integration authentication.

# Data Models

## User

- **userId**: String (unique identifier)
- **email**: String
- **enrolledCourses**: List<String> (list of courseIds)

## Course

- **courseId**: String (unique identifier)
- **courseName**: String
- **admin**: String (userId of the creator)
- **members**: List<String> (list of userIds)

## Folder

- **folderId**: String (unique identifier)
- **courseId**: String
- **name**: String

- **parentFolderId**: String (null for top-level folders)

## File

- **fileId**: String (unique identifier)
- **courseId**: String
- **name**: String
- **type**: String (e.g., "pdf", "image", "text")
- **url**: String (Firebase Storage download URL)
- **parentFolderId**: String (links to the containing folder)

## StudySession

- **sessionId**: String (unique identifier)
- **courseId**: String
- **title**: String
- **date**: Timestamp
- **time**: String
- **location**: String
- **description**: String
- **attendees**: Integer
- **rsvps**: Map<String, String> (userId to RSVP status: "yes", "no", "maybe")

## ChatMessage

- **messageId**: String (unique identifier)
- **courseId**: String
- **senderId**: String
- **timestamp**: Timestamp
- **content**: String (text message or file URL)
- **type**: String (e.g., "text", "image", "pdf")

## User Interface (UI) Screens

### 1. Login Screen

- **Components**:
  - Email field (EditText)
  - Password field (EditText)
  - Login button (Button)
  - Register button (Button)
- **Behavior**:
  - On "Login" click: Authenticate with Firebase Authentication; if successful, navigate to Dashboard.
  - On "Register" click: Navigate to Register Screen.

### 2. Register Screen

- **Components**:
  - Email field (EditText)
  - Password field (EditText)
  - Confirm Password field (EditText)
  - Register button (Button)
- **Behavior**:
  - On "Register" click: Validate inputs; create user with Firebase Authentication; if successful, navigate to Dashboard.

### 3. Dashboard (Home Screen)

- **Components**:
  - **BottomNavigationView** with three tabs:
    1. **Courses**: Displays a RecyclerView of enrolled courses.
    2. **Create Course**: Form to create a new course.

        3. **Join Course**: Form to join a course by ID.
    - **Courses Tab**:
        - RecyclerView listing course cards (showing courseName).
        - Clicking a course navigates to its Course Screen.
    - **Create Course Tab**:
        - Form with:
            - Course Name field (EditText)
            - Create button (Button)
    - **Join Course Tab**:
        - Form with:
            - Course ID field (EditText)
            - Join button (Button)
- **Behavior**:
    - **Courses Tab**: Fetches enrolledCourses from Firestore and populates the RecyclerView.
    - **Create Course Tab**: On "Create" click, store new course in Firestore, set user as admin, add courseId to user’s enrolledCourses, refresh Courses tab.
    - **Join Course Tab**: On "Join" click, validate course ID, check if course exists and user isn’t already a member, add user to course members and course to enrolledCourses, refresh Courses tab.

## 4. Course Screen

- **Components**:
    - Toolbar with courseName.
    - **BottomNavigationView** with three tabs:
        1. **Chat**: Real-time group chat interface.
        2. **Notes**: Folder-based notes organization system.

    3. **Meets**: List of study sessions.

  - **Chat Tab**:
    - RecyclerView for messages (sender, timestamp, content).
    - Message input field (EditText) and Send button (Button).
    - Attach button (for images/PDFs).
  - **Notes Tab**:
    - RecyclerView listing folders and files.
    - Buttons: New Folder, Upload File.
    - Clicking a folder navigates deeper; clicking a file opens it.
    - Search bar (EditText) for filtering notes.
  - **Meets Tab**:
    - RecyclerView listing study sessions (title, date, time, location, attendees, RSVP status).
    - Create Session button.
    - RSVP buttons per session (Yes, No, Maybe).
    - For "Yes" RSVP: Option to add to Google Calendar.
- **Behavior**:
  - Chat updates in real-time via Firebase Realtime Database.
  - Notes fetches data from Firestore; supports navigation and uploads.
  - Meets fetches sessions from Firestore; updates RSVPs and triggers Calendar integration.

## 5. Create Study Session Screen

- **Components**:
  - Form with:
    - Title field (EditText)
    - Date field (DatePicker or EditText)

        - Time field (TimePicker or EditText)
        - Location field (EditText)
        - Description field (EditText)
        - Cancel button (Button)
        - Save button (Button)
- **Behavior**:
    - On "Save" click: Store session in Firestore; return to Meets tab.

### 6. Menu/Profile Section

- **Components**:
    - Logout option (MenuItem).
- **Behavior**:
    - On "Logout" click: Sign out from Firebase Authentication; navigate to Login Screen.

## Key Functionalities

### Authentication

- **Login**: Validate credentials with Firebase Authentication.
- **Register**: Create user with email/password in Firebase Authentication.

### Course Management

- **Create Course**:
    - User inputs course name, creates course in Firestore.
    - User becomes admin; courseId added to enrolledCourses.
- **Join Course by ID**:
    - User enters course ID in Join Course tab.
    - System validates ID (exists, user not already a member).

    - If valid, adds userId to course members and courseId to enrolledCourses.

### Real-time Chat

- **Send Message**: Write ChatMessage to Firebase Realtime Database.
- **File Sharing**: Upload file to Firebase Storage; share URL in ChatMessage.
- **Receive Messages**: Listen for updates; update RecyclerView.

### Notes Organization

- **Folder Navigation**: Fetch contents from Firestore based on parentFolderId.
- **File Upload**: Store file in Firebase Storage; save metadata in Firestore.
- **Search**: Filter contents by name.
- **File Viewing**: Open files with appropriate viewers.

### Study Sessions

- **Create Session**: Save StudySession to Firestore.
- **RSVP**: Update rsvps in Firestore; trigger Calendar integration for "yes."
- **Calendar Integration**: Use Google Sign-In and Cloud Functions to create events.

## Additional Details

- **Error Handling**: Show loading indicators, success/failure messages, and handle network issues.
- **UI/UX**: Inspired by Canvas (dashboard), Discord (chat), and Notion (notes).
- **Performance**: Optimize Firestore queries and RecyclerView usage.
- **Testing**: Test with sample data for course creation, joining, and feature usage.

---