**Software Requirements Specification**

1. Introduction

CoStudy Cloud is a cloud-based Study Group Organizer designed to facilitate collaborative learning. Leveraging Google’s Cloud technology, this application aims to enable users to seamlessly join study groups, share learning resources, and schedule study sessions. Core functionalities will encompass group creation, session scheduling, and secure document sharing. Of note, scheduled sessions will seamlessly integrate with user’s Google Calander, providing timely notifications before each study session commences.

2. Functional Requirements

1.User Authentication:

1.1. User Authentication

* Users should be able to authenticate using their Google accounts to access the application.
* The signup process involves users providing their Google account credentials during the initial authentication.

1.2. Role Assignment

* During the signup process, if the user is on the whitelist of allowed administrators, they are assigned the System Administrator role. If they are not on the whitelist, they are assigned the default role of Learner.
* System Administrators can promote learners to the Group Administrator role.

1.3. User Profile Management:

* Users should have the ability to update their profile information.
* Users should be able to upload and update their profile picture, securely stored in Google Cloud Storage.

2.Group Creation and Management:

* Group Administrators should be able to create study groups.
* Group creation process involves specifying group name and description.
* Group titles must be unique to ensure each study group only focuses on one area.
* Learners can join existing study group created by group administrators but do not have the ability to create new groups.
* Group Administrators should have the ability to approve the learners who join their group.

3.Session Scheduling:

* Group Administrators should be able to schedule study sessions within their study groups.
* The scheduling process involves specifying details such as session title, summary, details, date, and a list of participants.
* The scheduling system should seamlessly integrate with learners' Google Calendars.
* Learners should receive timely notifications before each scheduled study session.

4.Document Sharing:

* Users should be able to securely share learning resources, such as PDFs, Word documents, and PowerPoint presentations, within their study groups.
* Uploaded study resources should be stored securely in Google Cloud Storage.

3. Nonfunctional Requirements

1.Data Storage:

* Application data, including user information, study groups, and scheduled sessions, should be stored securely in Cloud Spanner.

2.User Interface:

* The user interface should be intuitive and user-friendly, allowing easy navigation and interaction with the application.
* The design should facilitate a smooth user experience for creating groups, scheduling sessions, and sharing documents.

3.Security Measures:

* The application should implement appropriate security measures to protect user data, including secure authentication and authorization mechanisms.

4.Accessibility:

* The application should be accessible to users with a stable internet connection, ensuring worldwide accessibility.

5.Notification System:

* Users should receive notifications for scheduled study sessions, enhancing the overall user experience.

6.Scalability:

* The application should be designed to scale as user activity increases, ensuring performance and responsiveness.

4. Use Case Descriptions

|  |
| --- |
| **Name:** User Authentication |
| **ID:** 01 |
| **Actor:** User, System |
| **Preconditions:** User has an active Google account. |
| **Flow of events:**   1. User initiates the authentication process by selecting the option to log in. 2. System redirects the user to Google’s authentication page. 3. User provides Google account credentials. 4. System verifies the credentials with Google. 5. **IF** authentication succeed **THEN**    1. The user is authenticated   **ELSE**   * 1. The user is prompted to re-enter credentials |
| **Post conditions:** User is authenticated and granted access to the application. |

|  |
| --- |
| **Name:** Role Assignment |
| **ID:** 02 |
| **Actor:** User, System Administrator |
| **Preconditions:** User successfully authenticates and completes the sign-up process. |
| **Flow of events:**   1. **IF** the user is on the whitelist for System Administrator **THEN**   1.1 System assigns the System Administrator role to the user  **ELSE**  1.2 System assigns the Learner role to the user   1. System Administrators can promote Learners to Group Administrators. |
| **Post conditions:** User is assigned the appropriate role (System Administrator, Learner, Group Administrator). |

|  |
| --- |
| **Name:** User Profile Management |
| **ID:** 03 |
| **Actor:** User |
| **Preconditions:** User is logged into the system. |
| **Flow of events:**   1. User navigates to the profile management section. 2. User updates profile information. 3. **IF** user updates the profile picture **THEN**   3.1 The uploaded picture is securely stored in Google Cloud Storage. |
| **Post conditions:** User’s profile information and picture are updated. |

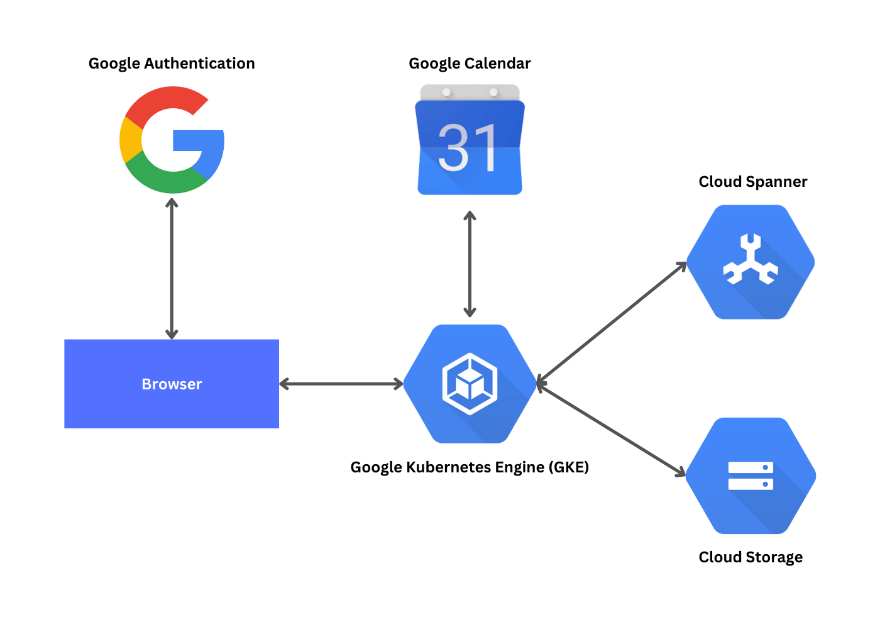
|  |
| --- |
| **Name:** Group Creation and Management |
| **ID:** 04 |
| **Actor:** Group Administrator, Learner |
| **Preconditions:** User is logged into the system. |
| **Flow of events:**   1. Group Administrator initiates the group creation process. 2. Group Administrator specifies group title and description. 3. System ensures the uniqueness of group titles. 4. Learners can join existing groups with Group Administrators’ approval. |
| **Post conditions:** Study group is created, and learners can join with approval. |

|  |
| --- |
| **Name:** Session Scheduling |
| **ID:** 05 |
| **Actor:** Group Administrator, Learner |
| **Preconditions:** User is part of a study group. |
| **Flow of events:**   1. Group Administrator initiates session scheduling within the study group. 2. Group Administrator specifies session details (title, summary, date, participants). 3. Application integrates with Google Calander. 4. The session is added to the Google Calander of all selected participants (learners). 5. Learners receive timely notifications for scheduled study sessions. |
| **Post conditions:** Study session is scheduled, and participants receive notifications. |

|  |
| --- |
| **Name:** Document Sharing |
| **ID:** 06 |
| **Actor:** Group Administrator, Learner |
| **Preconditions:** User is part of a study group. |
| **Flow of events:**   1. User uploads learning resources (PDFs, Word docs, Presentations) to the group. 2. Uploaded resources are securely stored in Google Cloud Storage. |
| **Post conditions:** Learning resources are shared and stored securely. |

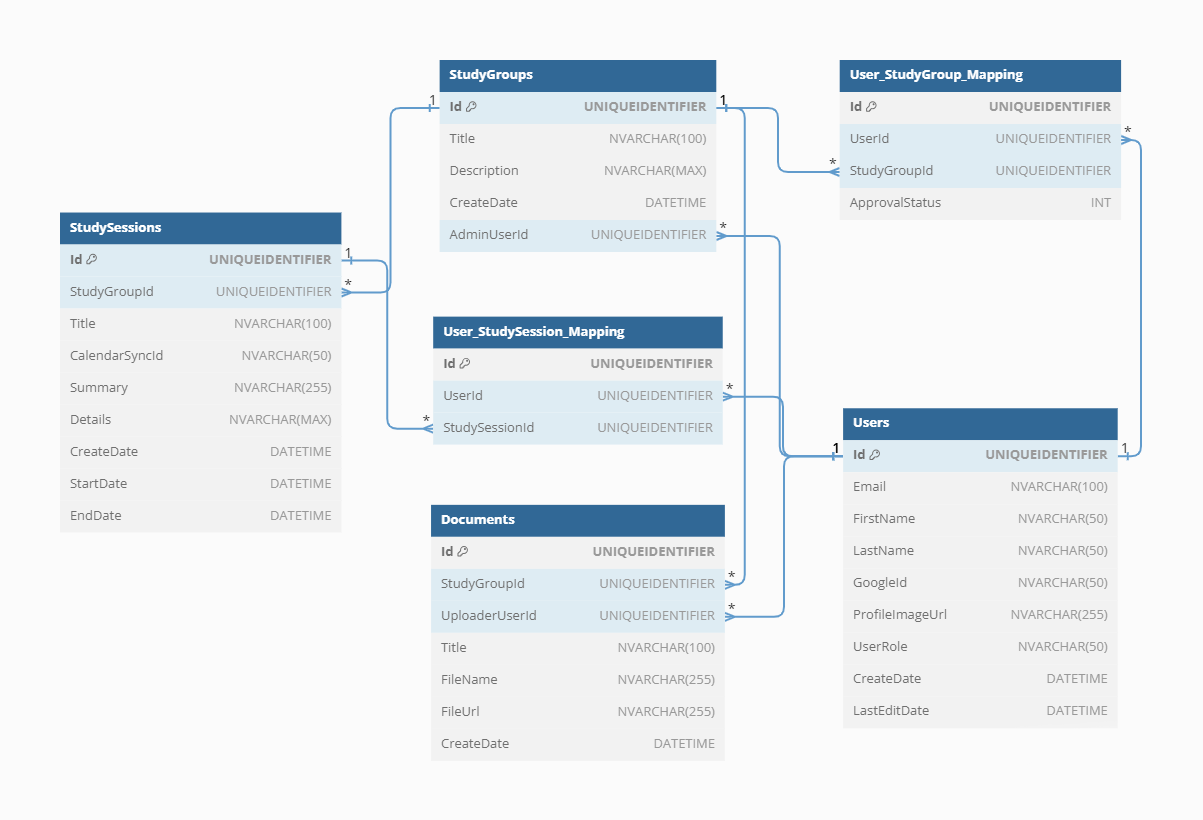
1. System Architecture

The graphical representation of the System Components is as follows:



1. Data Model

Database schema for the application is as follows:



1. Solution Structure and Deployment

7.1 Architecture Overview:

The application follows the Model-View-Controller (MVC) architectural pattern, promoting a modular and organized structure. This separation of concerns allows for maintainability and scalability.

7.2 Design Patterns:

The Repository design pattern will be leveraged to enhance the organization and efficiency of the data access layer. Repositories will handle the interactions with the database, providing a centralized mechanism for data access and manipulation. This pattern promotes separation of concerns and improves maintainability by abstracting the data access logic from the rest of the application.

7.3 Testing Strategies:

To ensure the reliability and correctness of the application, a comprehensive testing strategy will be employed. This includes both unit testing and integration testing. Unit tests validate individual components in isolation, while integration tests ensure seamless collaboration between these components.

7.4 Deployment Approach:

The deployment process is designed to be efficient and scalable. The application will be packaged into a Docker container, ensuring consistency across various environments. This Docker image is then deployed and orchestrated using Google Kubernetes Engine (GKE), providing a robust and scalable container orchestration solution.