

# Feature Design Document (User/Functional Reqs.)

## Google Authentication Account

### **Purpose:**

Users want to start using GoogleKeep through a simple authentication process.

### **Database dependencies:**

To authenticate a user, the database needs to register the user associated to his/her email .

### **Technical Note:**

Users need an existing email to create their login. They cannot use a registered email already in GoogleKeep database to create a new login.

### **Feature Result:**

User can use an email to create login credential to be able to utilize GoogleKeep's functionality.

## User Account Login/Logout

### **Purpose:**

Users need a login and log out function to keep others from manipulating their account details and notes.

### **Database dependencies:**

The database finds a match to the user's email in the users list. Then, it retrieves and secures notes associated with the user's ID and their login email.

**Technical Note:**

GoogleKeep needs to keep the user signed in on the browser unless the logout function is called.

**Feature Result:**

Users can log into GoogleKeep using their credentials and log out to hide their notes and account data from each other.

**Homepage****Purpose:**

Users need a homepage where they can have readily access to GoogleKeep's functions.

**Database Dependencies:**

User inputs are stored as data in different components in the database associated with the registered user.

**Technical Note:**

Notes should be displayed on the homepage where it can be organized by alphabetical and recency filters.

**Feature result:**

Users have a homepage where they can login, register, logout, create, delete, tag, archive, pin and share notes (UI).

**Add/Delete Note****Purpose:**

Users need to keep memory of information in a digital note that they can access.

### **Database dependencies:**

The database stores the created note with the associated user ID and login email.

The database removes the note selected by the user to be deleted and dissociates it with the user's ID and login email.

### **Technical Note:**

The note must include a subject and description. If a previously created note is no longer needed, I want to be able to discard unwanted digital notes.

### **Feature result:**

The user can successfully create a new note which shows up on the homepage. The user can successfully delete an existing note and it no longer shows up on their homepage.

## **Share note**

### **Purpose:**

Users want to be able to share digital notes with each other.

### **Database dependencies:**

GoogleKeep parses the user and recipient's email input and directs to the database.

The shared note's ID is now referenced to both the user and the recipient's ID.

### **Technical Note:**

If a shared note is deleted, the note is removed for both the sender and recipient.

### **Feature result:**

The user is able to share and view notes with recipients through email.

## **Pin note**

### **Purpose:**

Users want to keep their most important notes at the top of the page.

### **Database dependencies:**

-----

### **Technical Note:**

-----

### **Feature result:**

-----

## **Edit**

### **Purpose:**

As a user who wants to revisit and modify a note, I want to make a change to a note that already exists.

### **Database dependencies:**

The database needs to reflect the new changes that the user has made to a note.

When a user chooses to edit a note, the database needs to find it in the user's associated notes and update its contents.

### **Technical Note:**

The note must exist and once I make the changes I want, the database should update the note and store those changes.

**Feature result:**

The existing note's old content is replaced with the edited content that the user wanted and is reflected on the users homepage.

## **Archive**

**Purpose:**

User wants to temporarily set a digital note aside from the main page.

**Database dependencies:**

An event is triggered by the user in which references to targeted note's ID in the database. Using the note's ID, the system updates the archive state of the note and sends it to the archive sub database.

**Technical Note:**

Archived notes need to be able to be restored back into the homepage.

**Feature result:**

The user can view archived notes in another page and restore it back to the homepage by clicking the archive button.

## **Upload Image**

**Purpose:**

If necessary, users want to be able to add displaying images in my digital notes for effective note taking.

**Database dependencies:**

-----

**Technical Note:**

-----

**Feature result:**

The user can upload images from a finder and display them on the selected note.

**Search By Tag**

**Purpose:**

Users want to identify and locate their notes by adding search tags.

**Database dependencies:**

When a user adds tags, the database needs to find the particular note and store the associated tags with the specific note.

**Technical Note:**

The tags need to be stored as an element of the note so that the specific note can be filtered using the tags.

**Feature result:**

The user can view their notes grouped by the specific tags.