Prioritize

Joel Wilhelm, Cody Jones, Wajahat Iqbal, Shahrukh Rehman

# Project Goal

Reminder System that incorporates a priority value with time.

# Project Description

Prioritize is an application that aims to organize the user’s life by letting them type in a reminder for something that needs to be done by a certain time, and reminding them in intervals based on priority and the time between creation and the “due-date.” This lets the user periodically be reminded of the event or task over time rather than the user deciding having to decide for themselves. Once reminded, the user can choose to have the reminder be stopped, snoozed to remind the user in an hour or so, or simply be reminded at a later time that is again based on the priority value and the time between the current date and the “due-date.” These reminders will be synced across the user’s android devices allowing them to be reminded at any convenience to them.

# Requirement Specifications

## Functional Requirements:

The user must be able to create a new reminder. When creating a reminder, the user must be able pick a date and have the option of picking a time with that date. The user must also be allowed to pick multiple dates. The user must be able to view the current list of reminders that have been created and have the option of deleting them and editing them. The option of picking a alert versus an alarm must be available, that is a simple notification versus an alarm that won’t go off until user interaction. A reminder must have the option of being repeated. When a reminder is being alerted the notification itself must have the option of “Snooze” which tells the app to remind the user again sooner than later, “Remind me Later” which sets another reminder based on percentage of time remaining via the priority algorithm, and “Stop Reminding Me,” this will cancel the reminder entirely and leave it on the reminder list until the date/time has passed.

## User Interface Requirements:

The UI must implement a minimalistic design. The home page must display a list of the current reminders along with a floating action button at the bottom right for adding a reminder. Swiping to the right must reveal a calendar that has a mark on the dates that have each reminder due date, along with a mark for when that reminder is being alerted. The two of these marks must be displayed in a different style for differentiation.

## Usability Requirements:

The user must be able to create a reminder in a substantially small amount of time. These reminders must be synced automatically with the rest of the users android devices that have the application installed on them under the same Google Account.

## Performance Requirements:

The application must have a minimal response time in order for creating a reminder swiftly, and uninterruptedly. That means punctual app transitions, 0.1 seconds, opting for quick GUI animations rather than slower ones. Syncing across android devices must be as quick as the Google Drive API allows.

## Security Requirements:

The users reminders must not be available for viewing by anyone other than the user and then only through the application itself. The Google Drive API used to syncing android devices uses Google’s account system for security in the cloud.

# System

When user creates a reminder a description is set by the user, a date (or dates) is set by the user, a time is optionally set by the user, a priority value is set via a slider, and the user has options for ignoring the priority system, setting it to be repeatable, and changing from the default alert style to an alarm style notification. Once the reminder is submitted, the priority algorithm takes the priority value, the current date and time, and the date (or date and time) of the reminder and calculates a date and time for the user to be reminded based on the percentage of time between the two dates. The information the user entered and the new priority date is saved in a local device database and is synced with the user’s Google Cloud via the Google Drive API. This allows for the user’s other android devices that have the application installed under the same account to sync with the Cloud and set the reminder on that system as well. The alerts are set via the Android system which lets the application be closed until the time of activation of the event.

# Subsytems

* Home Activity: Fragment system for activity to display the current list of reminders on one slide, while swiping right reveals a calendar with reminders marked on dates.
* Notifications:Reminds the user of the task depending on the priority set by the user through sending a notification via an alert or an alarm. Notifications will offer Snooze, Remind Me Later, and Stop Reminding Me.
* Reminder Database: Manages the user’s reminders, this database is synced with the cloud so that the user’s other android devices can share all of the same information.
* Priority Algorithm: Set’s the date and time for the user to be alerted about a task they have created a reminder for. Takes the time between the date of creation and the “due-date” to set a new date based on the percentage of time between the two.
* GUI: Provides the user with menu, buttons, and calendar for priority, date and time input.
* Adding a Reminder: System for the user to create a reminder to be added to the reminder database. Has field for reminder description, date picker, time picker, repeatable check, alert/alarm switch, and the option for ignoring priority system.

# 

# 

# Timeline

