**Prioritize**

Reminder System that incorporates Priority with time

Joel Wilhelm, Cody Jones, Wajahat Iqbal, Shahrukh Rehman

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.

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 user’s reminders must not be available for viewing by anyone other than the user and then only through the application itself. Ideally, this will be done through an encryption algorithm that has yet to be decided upon. The AES encryption algorithm might be a good place to start looking. It is also possible that the Google Drive API will handle security as well. If time permits, this will be implemented.

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: Notifications consist of all data necessary for their functionality. Date created, date due, priority, a description of the notification, and what type of alarm/alert the user wants.

Reminder Database: Manages the user’s reminders, this database is synced with the cloud so that the user’s other android devices can share the same information.

Priority Algorithm: Sets 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.

Adding a Reminder: Adding a reminder should be quick and easy, as explained above. It should use the Date and Time picker, and when done, will add the reminder to the Reminder Database.

History Queue: Because we’re using google drive for syncing data usage may actually be a slight concern. Probably not, considering the user probably won’t have more than 100 reminders, but we can’t say that for sure. But even if the user has 200 reminders, that 200 lines of text on the drive may be miniscule, BUT I’m not sure how much memory we can use in the first place. It COULD be limited to a number of KB. Will research more, but it’s just something to think about.

Problem and Solution:

Reminder and calendar applications can be complex and oversaturated with usability features and functions resulting in being a pain to use. They require the user to enter too much information which can take too long and can be cumbersome. When an application is annoying to use the users will stop using them. A reminder application needs to have a minimalistic design while retaining maximum functionality, and keep the user from making too many decisions regarding the task they’re trying to remind themselves of. Prioritize seeks to solve this problem with the simple idea of rather than making the user tell the app when and how to be reminded of something, the app decides for them.

The philosophy behind Prioritize is for the user to simply type in the name or description of their reminder, assign a date and/or specific time, and a priority value. The only other options to be concerned about are alert versus alarm, and a simple check mark indi