Prioritize

Progress Report 2  
Cody Jones, Joel Wilhelm, Wajahat Iqbal, Shahrukh Rehman

**Problem and Solution**

Most calendar applications nowadays are complex and oversaturated with different usability features and functions, and therefore can be a pain to use. They require the user to enter too much information, which takes too long, and is cumbersome. When an application is too annoying to use, users will stop using them. There does not exist an application that is minimalistic in design, and yet, retains maximum functionality. This is the problem that Prioritize aims to solve. Rather than the user telling the app when to be reminded of something, the app should decide when to remind the user.

While other applications require the user to enter information about every reminder, such as when to be reminded, and how, this adds to the complexity and time to use. Prioritize seeks to differentiate itself from other applications by utilizing a priority system that, which when enabled, will simplify the whole process, and hopefully streamline the process of using a personal assistant application.

**Model 1 - System Design and Approach**

Prioritize takes a due date or time, more simplistically, a date or time that the user wants a task to be done by, and a designated priority value. Given these inputs, it calculates a new reminder date and time based on the time remaining between the time of creation and the final due date. If a reminder is created at Monday at 2:00 PM for task that needs to be completed by Wednesday at 2:00 PM, and the user has picked a five out of ten priority value, Prioritize will alert the user on Tuesday at 2:00 PM. Since the priority level for this reminder was five out of ten, this is 50% of the time between the two dates. This is the basic premise of the priority algorithm, a percentage of the time remaining until the due date and time. If the user selected a priority level of seven out of ten, then Prioritize will remind the user after 30% of the time has passed.

Prioritize will be an android application, and will not be supported for iOS or web development. It will, however, be able to sync with other android devices the user owns. There will be two databases, a local database, and a cloud-based database. If the user has an internet connection, they will be able to sync the local database to the cloud database in a secure and encrypted manner. If they do not wish to connect or sync, they will be able to use the application without worry.

**Model 2 - System Design and Approach**

Priority values take a slightly different meaning in this model. Instead of ten levels of priority, there are only five. For each level of priority, the app will alert the user a single day in advance, every morning until that that reminder is told to stop or the event is completed. Rather than scheduling alerts based on a percentage of time, the user is given an alert only every day with a maximum of five days in advance along with an alert at the time of the event. The higher the priority the higher it appears on the list of reminders being alerted to the user on that day. This results in Prioritize simply giving the user a list of things to do every day.

Model 2 will also be an android application, and can sync the same as Model 1.

**Comparison of Models**

Model 2 handles priority in a way that makes the user decide when to be reminded of

something rather than the app making the decision. This breaks away from the original idea of setting a reminder, setting a time and a priority value and letting the app choose how soon to remind the user. When the user tells the app that he/she has an activity to be done 30 days later, model two doesn’t account for that time and only reminds the user X days in advance. Model 1 takes into account the importance of that task and will let the user know about it until it’s completed or until they no longer want to be reminded.

**Subsystems and Requirements**

**Design Choices (and feasibility)**