**Simple Scheduler**

Jason Santana, Matthew Pinneo, and Matthew Spitzley

Originally planned, when the user opened the calendar screen, each day that had a task would display the task name with the date. Due to the screen being too small to do this with phones, if a date has a task in the calendar screen it will now be highlighted. Selecting that date specifically will still display all of the information of the task/tasks for that day. If there are too many tasks that go off screen the user will be able to scroll down through the list o f tasks for that day. At the bottom of each list is where the add new task button will be located. There will still be a button available for tasks that have no date named “undated” that will display the information for those tasks. The yearly view for the calendar has not yet been implemented.

There is a working settings button that goes to the settings activity which still has all of the original functions. The only changes made to these functions are what they take in as parameters and what they return, all of the purposes are still the same, as well as notificationType, being split into notificationEmail and notificationPush. The functions that have yet to be implemented are the help, switchScreen, and the googleSignIn. The switchScreen function could bring some errors into the application, as the ability to change the main activity is not something that can be done. What will need to be done is have the default main activity as the task list screen, then based off of the preference setting either do one of two things. The first would be if the user selects the task list screen as the main screen, then the app does nothing new. The second is if the user selects the calendar, if this is the case, when the app starts up it will need to redirect to the calendar activity as soon as possible.

The main page has three new methods: onCreateOptionsMenu, onOptionsItemSelected, and onClick. The onCreateOptionsMenu adds a button to the top right for adding a task. The onOptionsItemSelected method adds the “add or “cancel” option after a task has been input. The onClick method adds a settings button in the shape of a gear cog. This method takes you to a separate page with a list of options for app settings. There is a new field called mTaskList. It is a ListView object and is used for displaying the task list ArrayList. The add a task functionality still needs additional task fields added to the create task menu screen. A UI update method needs to be added to refresh the lask list screen when a new task is added. Along with the settings button a history button needs to be added to take the user to a screen to view a list of completed or canceled tasks. It will display similar to the task list ArrayList and ListView.

The “database.h” class from the old diagram has been removed in favor of the “DBHelper” class. The former class was not correct as SQLite operations and SQL statements are written in methods when developing in Android. Also, the “TaskList” and “CategoryList” classes have been removed, as the DBHelper class contains their functionalities. The DBHelper class has several methods: onCreate, onUpgrade, addCategory, addTask, deleteCategory, deleteTask, getCategoryList, and getTaskList. The onCreate method is called when the SQLite database is first created. The onUpgrade method is called when the database is updated. The addCategory method adds a category to the SQLite database. The addTask method adds a task to the database. The deleteCategory method takes a category and matches its name to a category in the database, then deletes it. The deleteTask method does the same but taking a task and using the task’s name as the identifier. The getCategoryList method returns a list of categories. The getTaskList method returns a list of tasks. All these methods have been implemented, and the local storage side of the database should be complete.

The online synchronization part of the database still needs to be completed. The SQLite database was prioritized because all data storage and manipulation will be done through SQLite. Upon sign-in, a single php script will be used to copy the SQLite database to the MySQL server. So the online aspect of the database is just for synchronization for security of the data. DBSync.php has been added to the class diagram. Also, the Google sign-in functionality has been removed as a class as it will be done in methods in existing classes.

In the Task class, the “recur” field has been changed to an enum rather than a String. The enum “Recur” is now contained in the same java file as the Task class. It is an enum of Strings corresponding to the different times when the user can set a task to recur. They are: Daily, Weekdays, Weekly, Monthly, and Yearly.