**Simple Scheduler**

Jason Santana, Matthew Pinneo, and Matthew Spitzley

**Abstract**

Simple Scheduler is an Android app for time management. Users can schedule and manage their plans quickly and easily, and never miss a due date with customizable reminders. Users can choose to synchronize their data by signing in with Google.

### **Tools and Technologies**

1. Android Studio

This is our primary IDE for development. The front end Java code as well as the interface will be developed in Android Studio

1. Android

The OS Simple Scheduler will run on. It is made Android version 9 or newer.

1. Java

Java is the main language of the front end application.

1. Visual Studio Code

IDE we will use for PHP and SQL.

1. MySQL

The open source database we will use.

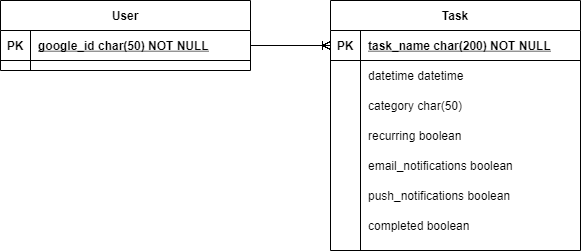
1. PHP

Our language for communication with the MySQL database.

### **Requirements List**

1. Task List Screen
   1. A button that will show a listbox of categories
      1. Displays all of the existing categories as well as an “All” and “Edit” option
         1. Selecting a category will cause the Task List screen to only display the tasks of that category.
         2. Selecting “All” will display all tasks, and selecting “Edit” will display the category edit screen.
   2. A list of tasks will be displayed for the user. Users will be able to read the details of the task: name, time, and date.
      1. When a task is selected it can be edited or marked as complete
      2. Tasks are sorted first by date, then alphabetically.
      3. Each task falls under a header based on the time.
         1. Headers include: Late, No Date, Today, This Week, Next Week, Next Month, and Later.
         2. Late tasks have a color separated date.
   3. A settings button that leads to the settings screen
   4. A new task button that leads to the new task screen
2. Category Edit Screen
   1. Show a list of categories sorted alphabetically.
   2. Selecting a category allows the user to change its name or delete it.
      1. If a category is deleted, the existing tasks with that category will be changed to “no category”
   3. Selecting the add button allows the user to create a new category.
      1. This lets the user name the new category then it is added to the list of categories.
3. New Task Screen
   1. This screen incorporates many text boxes and checkboxes for the user to manually enter information in about their task
      1. Name textbox which will have the text “New Task” by default. This text box will not be allowed to be empty.
         1. An empty text box will result in the application telling the user they cannot make the task until it has a name.
      2. Which category this task falls under
         1. If no category is selected then when the task is made is it marked as “no category”
      3. Date to be completed, this field is optional.
         1. When filling this out the user can choose from a calendar for which day the task is.
         2. The other way to fill this out is to manually enter in a date.
            1. The way the user enters the date manually will depend on how they chose to read dates in the settings.
      4. Time to be completed, this is optional.
         1. This will be filled out by a scroller for the hour and minute of the day.
      5. Once a date is selected, options for recurring and notifications will be displayed.
         1. A setting for whether this task is recurring
            1. If checked, selects when the task should reoccur
         2. A checkbox for whether to get email notifications.
            1. If checked, allow the user to choose when the notification occurs.
         3. A checkbox for whether to get push notifications.
            1. If checked, allow the user to choose when the notification occurs.
4. Edit Task Screen
   1. Tapping a task outside of the completion checkbox allows the user to edit the task offering the exact same options as the new task screen.
5. Calendar Screen
   1. Displays a calendar of the month. In each day that has a task, it shows the name of the task.
   2. Undated tasks are displayed separately from the calendar.
      1. Selecting the undated will show the undated tasks in the same way selecting a day shows the day's tasks.
   3. The category listbox is accessible in the calendar screen as well.
      1. Selecting a category will then display only tasks of that category.
   4. The settings menu is also accessible from the calendar screen.
   5. If the user selects a day with no task it will bring them to the new task screen with the day already filled in for that selected day.
   6. If the user selects a day with a task already in it, a screen will display the tasks for that day.
      1. A new task button will appear with the list.
   7. An option to display the calendar in a yearly view.
      1. This view will not show specific tasks.
      2. Days with tasks will be highlighted.
      3. Each month can be selected to view the more detailed monthly view.
6. History Screen
   1. Accessed by swiping left from the starting screen, the history screen will show completed tasks.
      1. They will be sorted by date, then alphabetically.
   2. Tapping a task on the history screen will “select” it.
   3. A select all button will select every task.
   4. An edit button to select multiple, specific tasks.
   5. Once at least one task is selected, the user can delete the selected task(s) permanently, or restore it as an uncompleted task.
7. Settings Screen
   1. Accessible from the history, task list, and calendar screens.
   2. A “Help” button which will display a brief description of Simple Scheduler’s features.
   3. An option to toggle between 12 hour and 24 hour time.
   4. The ability to change how the user wants to read dates.
      1. Month/Day/Year (2/8/2021)
      2. Day/Month/Year (8/2/2021)
      3. AbbreviatedMonth. Day Year (Feb. 8 2021)
      4. Day AbbreviatedMonth. Year (8 Feb. 2021)
      5. Month Day Year (February 8th 2021)
      6. Day Month Year (8th of February 2021)
   5. An option to change the screen seen on startup to the task list screen or the calendar screen.
   6. An option to select a default time for when a notification occurs.
      1. By default the notification will be 1 hour prior to the event occurring.
      2. If the task is created with less than an hour until it occurs it will immediately send out the notification.
   7. An option to toggle late notifications, which will notify the user they have missed a deadline.
      1. Users can choose these to be push, email, or both.
   8. An option to toggle all push notifications (this will remove them from the task add and edit screens).
   9. An option to sign in with a Google account.
   10. If the user is signed in, an option to synchronize their tasks to the database.
   11. If the user is signed in, an option to toggle all email notifications using the email from the Google account (this will remove them from the task add and edit screens).
8. Database
   1. User table
      1. google\_id
   2. Task table
      1. task\_name
      2. datetime
      3. category
      4. recurring
      5. email\_notifications
      6. push\_notifications
      7. completed

### **Diagram**



### 

### 

### **Updated Timeline**

|  |  |
| --- | --- |
| Week 1 Jan 24 - 30: | Project Proposal |
| Week 2 Jan 31 - Feb 6: | Begin designing the application. As well as begin learning the experience we listed as missing (MySQL and Android) |
| Week 3 Feb 7 - 13: | Continue designing and learning. |
| Week 4 Feb 14 - 20: | Project update 1 - Initial Design |
| Week 5 Feb 21 - 27: | Begin making the application in our respective areas. |
| Week 6 Feb 28 - Mar 6: | Continue working on the application. |
| Week 7 Mar 7 - 13: | Continue working on the application. |
| Week 8 Mar 14 - 20: | Project update 2 - Technical issues |
| Week 9 Mar 21 - 27: | Finalize our individual work on the application. |
| Week 10 Mar 28 - Apr 3: | Ensure the separated work is working correctly when put together. |
| Week 11 Apr 4 - 10: | Repetitively test the application and debug in each area that needs it. |
| Week 12 Apr 11 - 17: | Project update 3 - Implementation and testing |
| Week 13 Apr 18 - 24: | Any final testing and debugging. |
| Week 14 Apr 25 - May 1: | Any final testing and debugging. |
| Week 15 May 2 - 8: | Final Presentation (May 5 8:00-10:00) |