Software Requirements and Design Document

For

Group <9>

Version 3.0

Authors:

Kevin Tran

Mkhuleko Bemiller

Sean O'Meara

Andres Ponciano

Patrick Zatz

1. Overview

A scheduling management android app targeted towards college students. With additional smart features that help with segmenting study time, class time, and free time. The app will have additional features to promote habits that are beneficial to students such as a study pomodoro timer and notes functionality in a single centralized study app to promote usage. The back end supports user accounts to back up to a firebase database for schedule sharing or to create protected schedule back ups in the event of a physical device failure.

The App will also allow for more options when creating events to reduce the amount of work the user has to do when compared to an app such as Google Calendar. Such options include repeating events that occur in the same week alongside repeating the events throughout different weeks. The app will also be used in combination with the Android Calendar ContentProvider to be able to put events from Procrastiain't onto the calendar.

2. Functional Requirements

0 = Done
1 = High Priority
2 = Medium Priority
3 = Low Priority

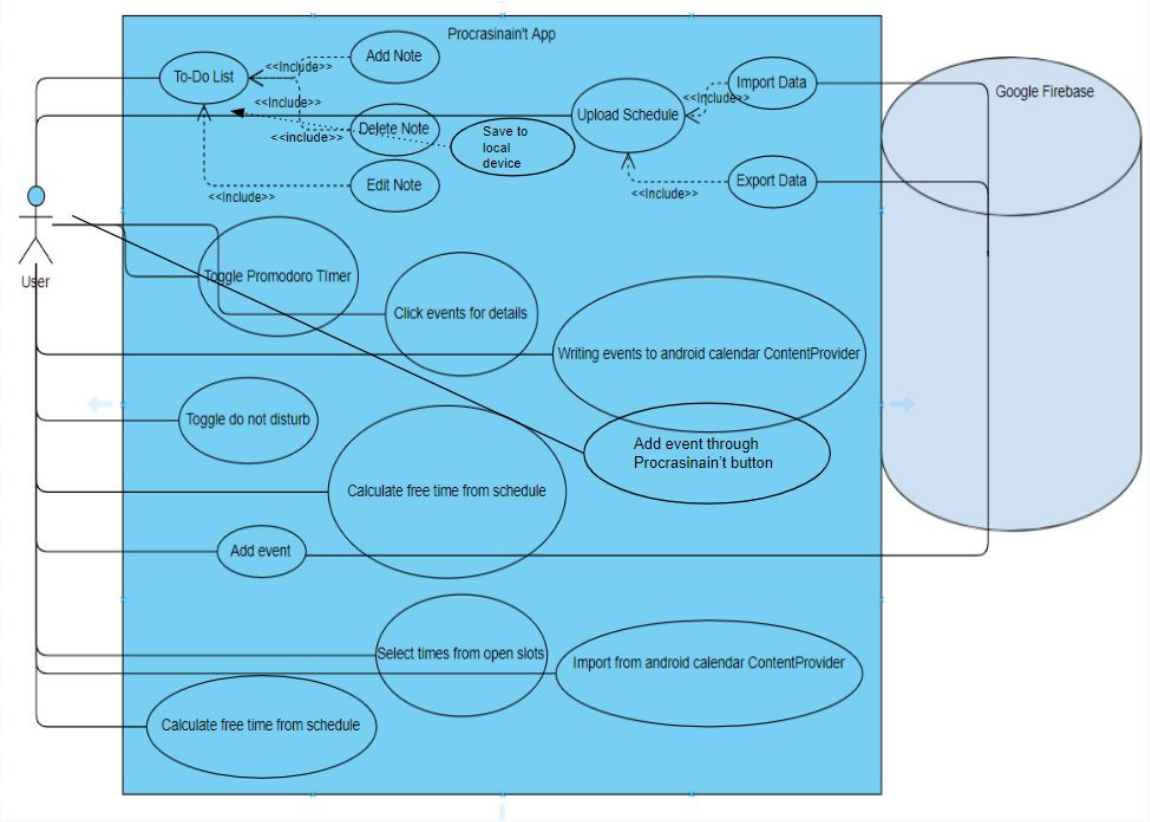
0) The ability to register and log into an account to the firebase.
0) The ability to add notes onto certain events. The user has a notes tab that can be edited to help the user not forget about things. Requires the completion of the Calendar.
0) Adding Events to the ContentProvider. This requires the user to enter in the dates and times of the event and then click the add button. Requires the completion of the Add Event Page.
0) Adding Events to the Calendar. This takes out the information from the ContentProvider and then loads it onto the local calendar for display. Requires the Add Event Page as well as the Calendar page.
0) Reading Events from other Calendars and taking the information given and putting it into the App's ContentProvider to save the new Events. Requires Calendar API.
0) Adding Events to the Firebase. Takes the information that is entered by the user on the Add event page and then uploads them into the Firebase. Requires the completion of the Add Event Page..
0) Adding the Function that helps find time in the schedule for the user to start studying. This is activated when the user decides to find time for hw. It shows the times where they can work on homework. Requires the completion of the Calendar.
0) Adding Notifications to the App. Every day a notification will appear telling the user to make sure to check the app.
0) Displaying more details of an event when clicked on, when a day is clicked it will show the events as well as the details attached to the event. Requires completion of the Calendar.
0) The creation of the Pomodoro timer, which will tell the user when to take breaks when studying. This activates when the user starts to study, it can be disabled and edited for different lengths of breaks.

3. Non-functional Requirements

- Cohesive UI design throughout app, matching material design standards
- Consistent database storage of calendar events and firebase to communicate with the built in Android ContentProvider
- A stable non-crashing application

- Ability to run consistently on Android 5.0 and newer phones

4. Use Case Diagram



The use case of our Prcrastinain't app is illustrated above. From this diagram we can see that there is only one actor namely the User who will interact with all use cases.

- 1. Name:To-Do List (Notes)
  2. Participating actors: User
  3. Entry condition:
     - User clicks the Notes button
  4. Exit condition:
     - User clicks the back button
  5. Flow of events:
     1. User opens the application page
     2. User is on the main menu page
     3. User opens the side bar
     4. User Selects notes
-

1. Name:Toggle Promodoro Timer
2. Participating actors: User
3. Entry condition:
    - User clicks the Promodoro Timer button
    - User clicks the start or stop button
4. Exit condition:
    - Timer Ends
    - User clicks the stop button
    - User clicks the back button
5. Flow of events:
    1. User is on the main menu page
    2. User clicks on the Promodoro Timer button
    3. User is brought to the Promodoro Timer page

-

1. Name: Add Event
2. Participating actors: User, Firebase
3. Entry condition:
    - User clicks the Add Event button
4. Exit condition:
    - User clicks the back button
    - User adds the event
5. Flow of events:
    1. User clicks the Add Event button
    2. User fills out Title, Dates, Times, upload to Firebase, recurring events
    3. User clicks the Add Button to upload the Event to the ContentProvider
        a. if the Firebase is selected, then it uploads the event to the Firebase
        b. the user must be logged in to access firebase

-

1. Name: Toggle do not disturb
2. Participating actors: User
3. Entry condition:
    - User clicks the settings buttons
4. Exit condition:
    - User clicks the back button
5. Flow of events:
    1. User clicks the Settings button
    2. User clicks on the Do not Disturb toggle

-

1. Name: Click Event for details
2. Participating actors: User
3. Entry condition:
    - User clicks the Event on the Calendar
4. Exit condition:
    - User clicks the back button
5. Flow of events:
    1. User clicks on a day on the Calendar
    2. User clicks on an event that takes place on the day selected
    3. The Information is displayed to the User

-

1. Name: Upload to Schedule

2. Participating actors: User
3. Entry condition:
    - User clicks the “Exports to schedule while sign in”
4. Exit condition:
    - User clicks the back button
    - User completes the “FireBase Export”
5. Flow of events:
    1. User signs in
    2. User clicks Export To calendar in the slide out menu
    3. The calendar is back up to the firebase
    4. The user is returned to the home calendar view

-
    1. Name: Import from Android Calendar Content Provider
    2. Participating actors: User
    3. Entry Conditions
        a. User Clicks on Read From and Write to Calendar option
    4. Exit conditions
        a. User exits from the option
        b. User saves the imported calendar to local content provider
    5. Flow of events
        a. User clicks Read From and Write to Calendar option
        b. User selects which calendar they want to read events from
        c. User selects Read
        d. User selects Save to save it to local content provider

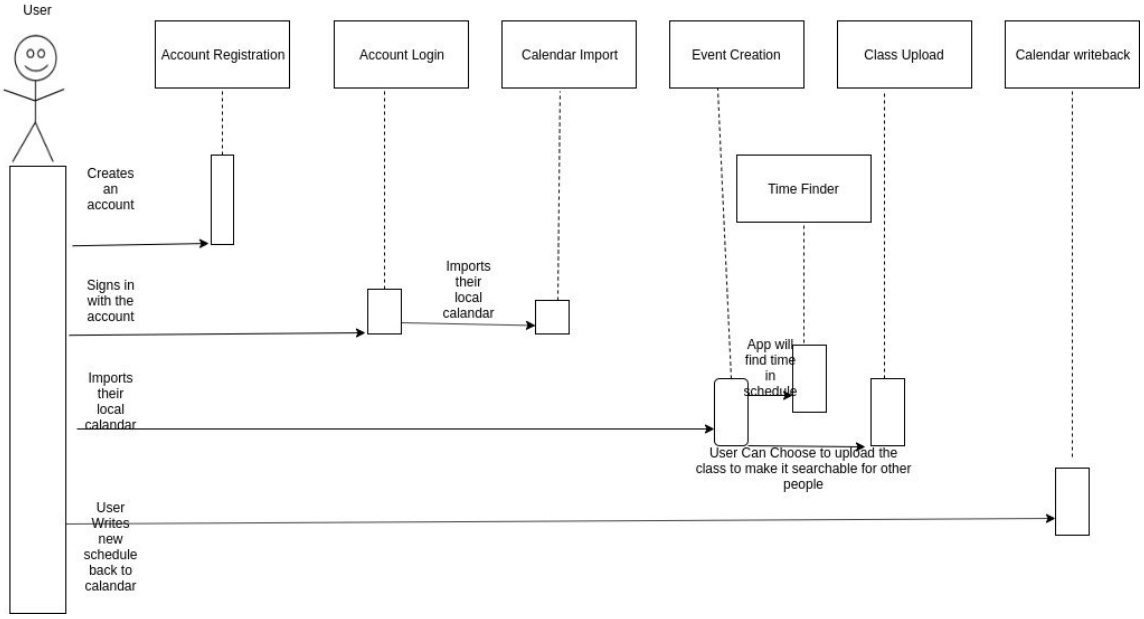
-
    1. Name: Writing Events to Android Calendar Content Provider
    2. Participating actors: User
    3. Entry Conditions
        a. User Clicks on Read From and Write to Calendar option
    4. Exit Conditions
        a. User exits from the option
        b. User writes the local event to local content provider
    5. Flow of events
        a. User clicks Read From and Write to Calendar option
        b. User selects which repeating events they want to write into the calendar
        c. User selects writes to Android ContentProvider Calendar
        d. User Exits from option

1. Name: Calculate free time from schedule
2. Participating actors: User
3. Entry condition:
    - User clicks the calculate free time from schedule button
4. Exit condition:
    - Click back
    - Finishes the calculation
5. Flow of events:
    5. User is on the main menu page
    6. User clicks the calculate free time from schedule button

- 1.

    1. Name: Add event with Procrastiain't
    2. Participating actors: User
    3. Entry condition
        - Clicks the Procrastiain't button on the menu
    4. Exit condition:
        - Clicks the back button
        - The user adds an event
    5. Flow of Events:
        - User clicks the Procrastiain't button
        - User fills up the information needed for the event
        - User clicks next
        - The user then selects the time period that the event will be in
        - The next dialog is filled out to specifiy when in that time period the event starts.
        - The user confirms and adds the event.

5. Class Diagram and/or Sequence Diagrams



6. Operating Environment

Any Android device using Android 5.0 or higher, has an internet connection, a local calendar app, and has a valid google account/ email.

7. Assumptions and Dependencies

The project depends on the Firebase in order to login and register their account as well as the ability to upload and download user created events.