

Running Late Co.  
Garrett Quintero  
Clara Tran  
Tracy Karol  
CSCI 448

## Forever ToDo

### Final

#### Using the App

This app is the perfect virtual version of a planner; it allows you to create a todo list that will send you notifications to keep you from missing your due dates. When the app first opens, you are brought to your list of tasks you have yet to finish. If the list is empty, press the plus button located in the top menu bar to add a new task. Here you can set details about the task such as its title, due date and time, priority level, and category. Be sure to hit the save button or else it will not be listed in your todos. Once there are items in your list, you can click on one of the items to be brought to the edit page where you can change any of its details, delete the task, or hit the complete button. Once you have completed your tasks, the button will be disabled, the user will be able to see the task completion time, and the task enters the history list. Back to the Task List page, you are also able to sort your incomplete tasks by your choice of Due Date, Priority, or Category. The UI will automatically update to show the sorted list. The Task List Page will also color code the tasks based on their priority level and/or how close it is to the task's deadline.

The History icon will take you to the list of all of the tasks you have completed. Clicking on one of the tasks will bring you to that task, but you won't be able to edit any details. Scroll through all of your achievements!

The Settings icon will take you to the app's settings, where you can enable notifications, the notification timer, and notification sound/vibration on or off. User will be notified of the deadline for any of the incomplete tasks.

As a surprise, there is an easter egg for a special photo for the user to view. Shake the phone enough and surprise!

#### Backend

On the backend, the Task List is scrollable through the RecyclerView. Each time the user adds a task and saves it, the UI is updated automatically so that when the user is taken back to the Task List page, their newly created task will be there ready for them to see. When the task is saved, the details are persisted in the app by saving the task and its details into the app's internal

database. Each task has a button to mark the task as complete, which is used as a flag. If the task is completed, then the task will be removed from the Task List page and added to the History page. The flag is also used as a check for notifications so that the user won't receive a notification about a passed assignment. In Settings, if the switch for Notifications is enabled, then the boolean that checks if notification is on is set to true and the PollService will be able to push notifications to the user.

## **App Components**

The app has only one activity, which is the Task List Activity. The Task, Task List, History, and Settings are all its own fragments. The app also has a Bottom Navigation View that shows an icon for Task List, History, and Setting. The app is more compact and flexible with having only one activity because it's able to utilize the Navigation View to call and replace the appropriate fragment. Each page that the user can navigate to is its own fragment so that it can inflate its own personalized view since none of them will have the same one. Any changes are also passed between fragments easily because the arguments will pass through one activity. The option to enable notifications gives more flexibility and freedom for the user when they use the app so that they don't have to go into the settings on their phone to disable notifications there.

## **Difference Between Final Design and Original Concept**

Originally, the app would allow user to login if they have a profile for the app. This is so that the user can have their profile across multiple devices. However, due to the time constraint, the difficulty of saving a profile, and ensuring that the profile is saved properly, the concept is scrapped. We also wanted to present a graph of the number of tasks completed on time, before the due date, etc. Again, the concept is scrapped due to time constraint. Other concepts scrapped in the final design were location and images for tasks.

## **Challenges**

Challenges that we faced were saving the due date and time to the database properly. We needed to finish implementing the Date Picker Dialog that the user interacts with to set the deadline to finish adding the Date object to the database. We also needed to change the backend design of having an activity for each fragment to only one activity so that the app is more compatible with the Navigation View and to simplify the implementation of the app. Overall, the biggest challenge is time constraint and finding a time for all of us to meet (because junior year hooray).

Running Late Co.  
Garrett Quintero  
Clara Tran  
Tracy Karol  
CSCI 448

## Forever ToDo

### Proposal

#### **App Description**

Forever ToDo is meant to be a detailed, running ToDo list. Users will be able to create tasks with details such as setting a time and due date, providing a description, recording a time and date of completion, potentially setting a location for location based tasks, setting reminder alarms, setting priority, set warning notifications, selecting or setting a category, and setting a point value or difficulty to the list item. With the tasks they create, the app will sort the tasks to be done by categories, time and date, or priority based on what the user chooses.

When a user completes a task and marks it done in the app, they will earn points towards different achievements within the app and show whenever they reach a new achievement. Once done, tasks will be moved to a different page where users can view each of their completed tasks which can be sorted either by completion date/time or by due date.

To remove clutter, the app can be set to automatically remove completed or overdue tasks after a number of days.

#### **Features/Components**

The app will include many of the features that will be addressed in this class. Buttons and text boxes will be highly used in the layout of the app as well as toasts and animations that will appear with action from the user. For example, when a task is completed, an animated check mark will play on the screen. The app will make use of notifications to let the user know that their deadline is approaching, they have missed the deadline, etc. and will also have an alarm feature that the user will be able to set when creating a task. When first entering the app there will be a main menu that will also include buttons and text boxes and may have other settings for the app available to the user. We plan to create an “Achievement” tracker that will be used with a statistical graph that can show the user’s history of their ability to finish their tasks.

On the back end side of the app, we will be making use of Cloud database/SQLite along with Google APIs to store all the data of the tasks with their individual settings and the user history. We are hoping to add a location tracker feature for tasks that are location based so that the app

can notify a user information about travel time or let them know if they are close to a location where they could complete one of their tasks. One more component we would like to add is a syncing mechanism that will allow the app to use information from various apps such as Google Maps or Google Calendar in addition to being able to sync with other devices.

### **Planned Distribution of Labor**

Clara will be responsible for creating toasts, notifications and alarms as well as helping Garrett with the overall design of the app including default colors, color schemes, and the app layout. Garrett will be focusing on the database elements of this app including the research on Google APIs and using the database software listed prior. Tracy will be programming the menu, animations, and task creation while also handling the syncing feature with assistance from Garrett.

### **Road Map**

Our first step in developing Forever ToDo will be the design. The app layout will be established with text boxes and buttons as a starting place to add functionality to the app. It will then have a potential for nice details (colors, images, other styling) to be added as well. Basic components such as toasts and active listeners on the buttons or other areas will be done next. At this point, the task creation and animations will be working. Once the app is in a functional state, the notification component will be included.

After the front end is mostly finalized, back end work will be the priority. Setting up a database will be the next task in addition to making use of an API if possible. Our last goal will be to enable the syncing feature. Given the time constraints, we are not sure if this will end up in the app. Thorough testing will be performed throughout this process of development to ensure that the final product is clean and works efficiently.

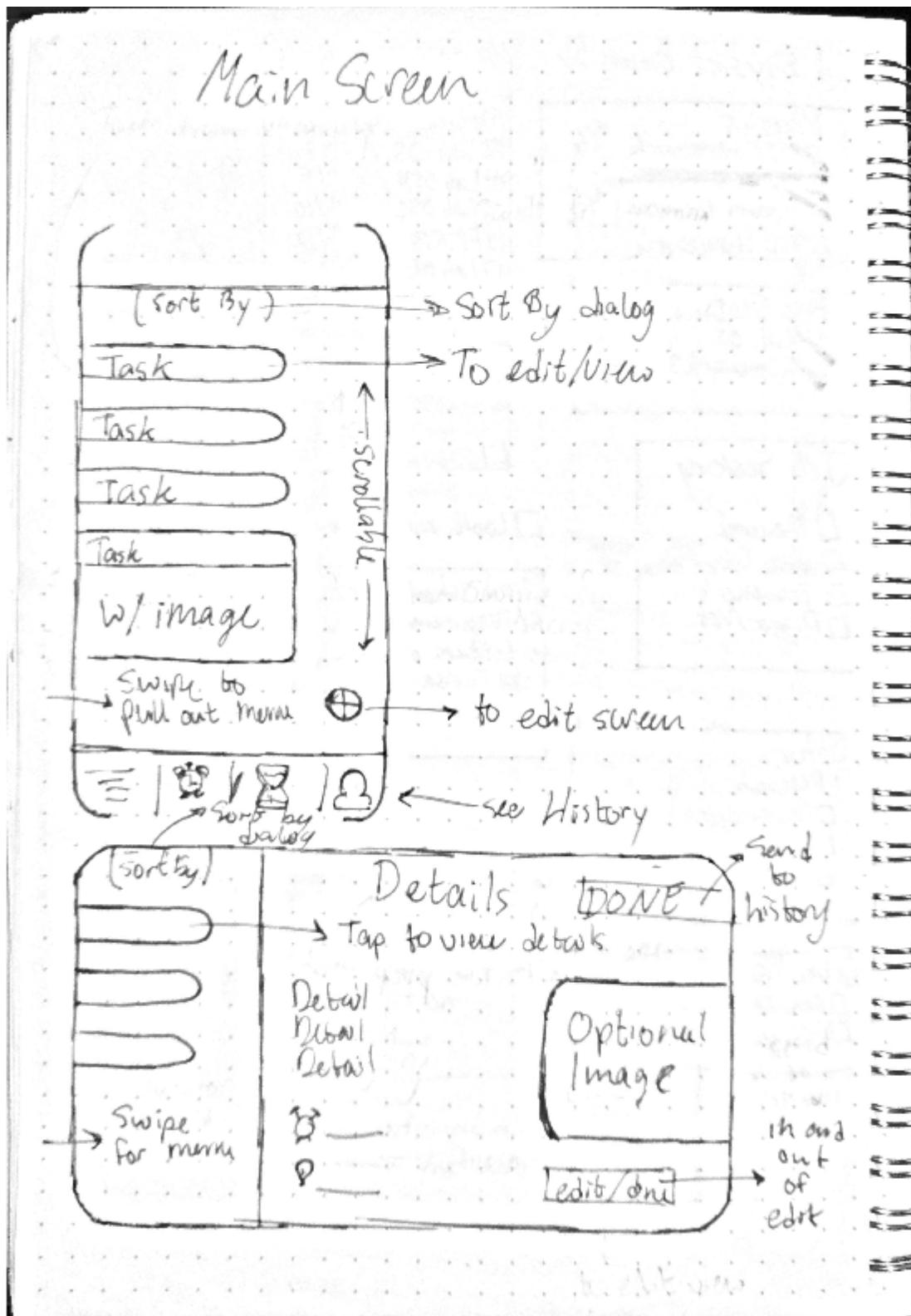
### **Anticipated Challenges**

If we try to use a cloud database or syncing system within the app, we'll have to research and find the proper APIs to use, which may be difficult for our first try. Connecting the back end elements with the front end will be challenging as well as debugging and testing. Incorporating the location tracker may be an issue as well since it will need to be active when the app is potentially in a destroyed state.

### **Sources**

We may potentially use some of Google's API if we try to implement syncing between multiple devices and using Google Maps for the location tracker.

# Story Board



# Login Screen



Username

Password

Sign In  
sign up

To List View

To Sign Up Page

# Password Reset Page

Reset Password

Enter Old Password

---

Enter New Password

---

Confirm New Password

---

Submit

Return  
to  
Profile/  
Settings  
Page

# Profile / Settings Page

edit profile picture



Username

Settings  
Change Password  
Notifications  
Alarms  
Sign Out

See History



Toggle  
Settings

To Login Page

To Password  
Reset Page

# Sign Up Page

Welcome!

Username

Password

Confirm Password

Email

Phone (Optional)

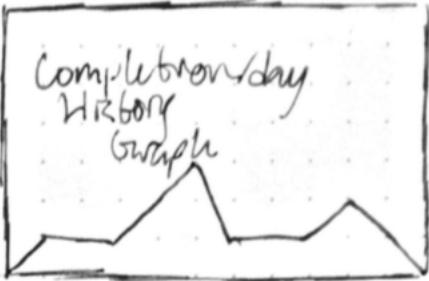
Agree to Terms & Conditions

Submit

To List View

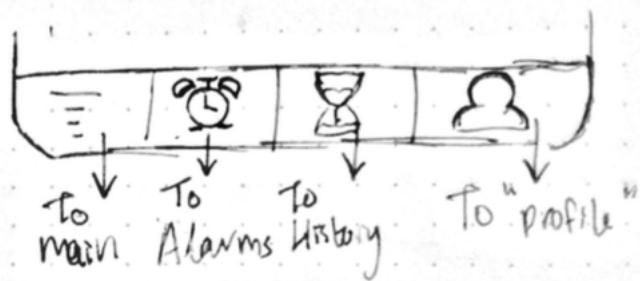
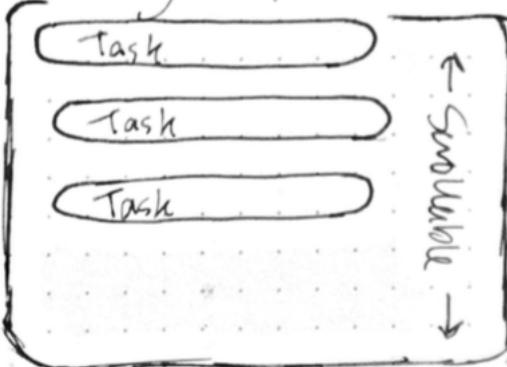
# History

## ToDo History

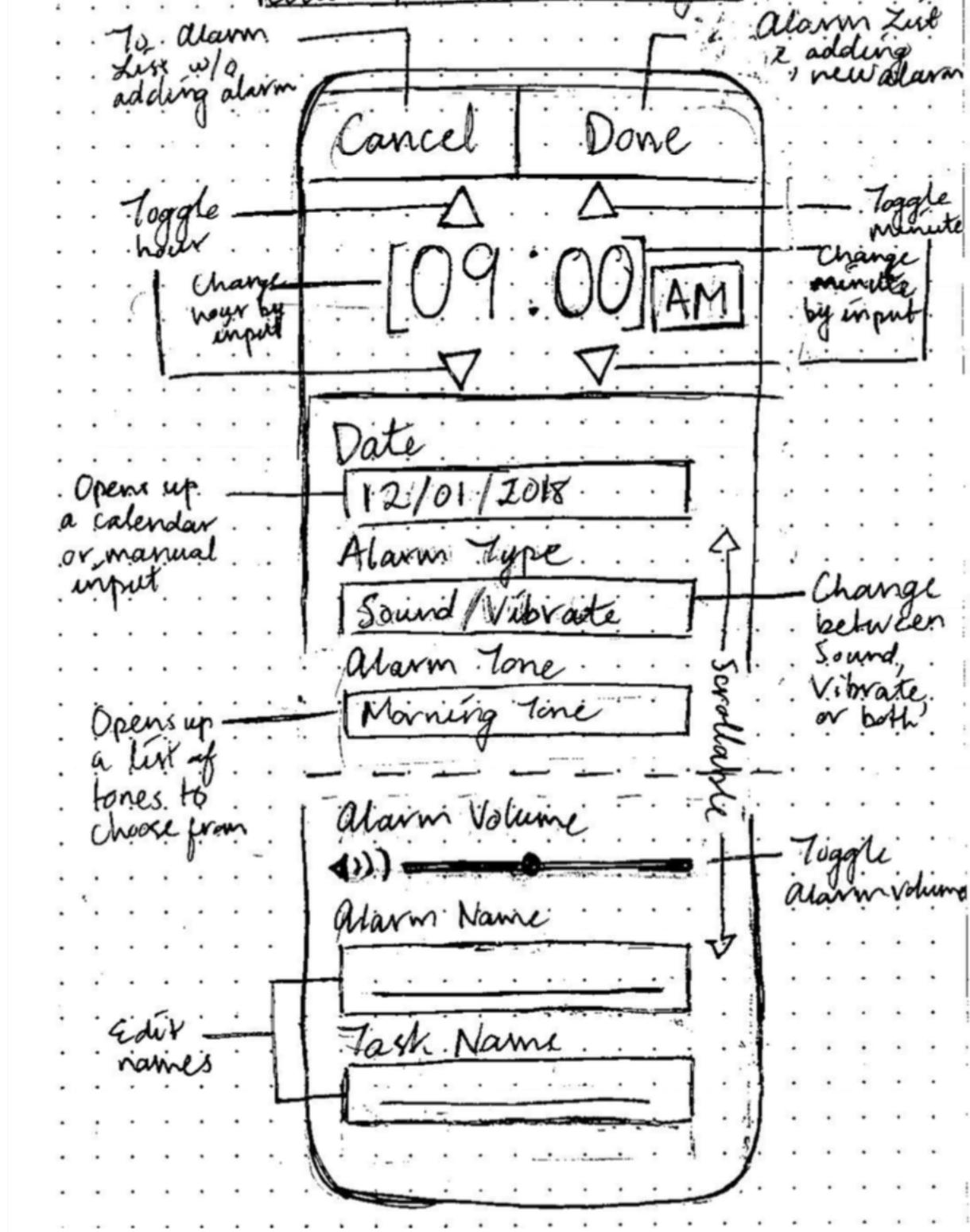


% On Time Completion  
% Early Completion  
% Late Completion

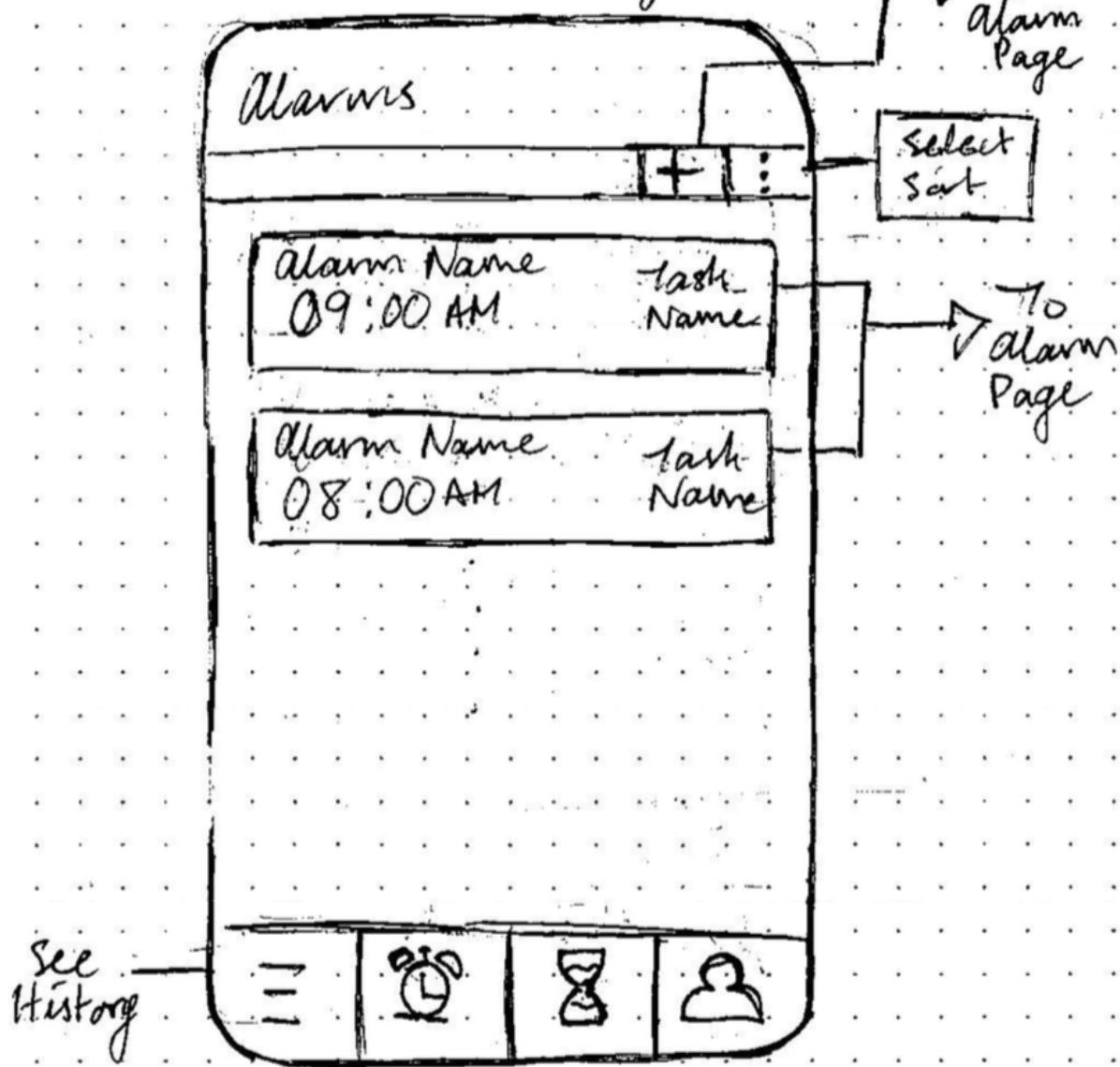
## Recently Completed



## Alarm/Edit Alarm Page



## Alarm List Page



## Edit/New Task Page

Task Name

Due Date  
12/01/2018

Time  
09:00 AM

Description

Priority Level

Category  
School

Notification  
12/01/2018 07:00 AM

Alarm  
09:00 AM

Edit Text

opens up calendar or manual input

opens up alarm option (not page!)

Toggle Level

open list of options

To Alarm Page

Scollable

# **Alpha Release Notes**

App: ForeverToDo

## Usage Instructions:

The app will bring up the task list view on launch with two buttons at the bottom that navigates to the other pages. "To Profile" will take you to the profile page, which will have a default image and username. To go back to the task list view, simply press the up arrow. "To History" will take you to the history page, which will also have a default image for the graph and a list of recent tasks completed or past its due date. Click on one of the tasks to view its details where all of its items are disabled until the Edit item is clicked. The up arrow will take you back to the task list page. To test the "Force Crash!" button, you must first press the sign out link in the preset profile, which will then take you to the Login page where the button is located.

## Known Bugs:

- "To Profile" will take you to a preset profile instead of the Login Page since the implementation to check if user is logged in or not is not set up yet
- Due date, time, notifications, and alarms are set up as EditText since the PickerDialogs are not set up yet
- Keyboard automatically pulls up when you click on a task to see its details

## Specific areas for feedback:

- profile picture
- overall flow of the app, ignoring the profile bug
- task list design (e.g. if borders are needed, colors, etc)
- history page's setup with list being the only thing scrollable or to have the whole page scrollable

# **Final Release Notes**

Authors:

Tracy Karol

Clara Tran

Garrett Quintero

Usage:

Navigation through menu bars and creating tasks is intuitive for the user.

Bugs:

None