GTD ORGANIZER APPLICATION 1.0

Project Report

SWEN 3001 Android App Development Project

BSC_

Joshua Alkins 190908 joshua_alkins@hotmail.com

Ashray Soares 190910 soaresashray@yahoo.com

Contents

1 Introduction

1.1 Problem

The inability to work efficiently and effectively due to a lack of mental clarity and disorganization regarding tasks.

1.2 Solution

The implementation of a the GTD method in the form of a mobile application.

1.3 GTD method description

Our solution to this problem is to implement a system found in the book “Getting Things Done” by David Allen.

The system is designed to free your mind to focus on completing tasks at hand instead of keeping track of various information we all have stored in our brain.

The system is based on:

- Removing information or thoughts that are unnecessary to what you are trying to currently do, by writing down thought and placing it in a trusted system.
- Returning to all the thoughts written down and deciding what is best to do with them and if they are worth your attention in the first place.

1.4 The Process

Collect your random thoughts that are on your mind by writing them out and placing them in the inbox.

Process items in the inbox when you have time by going through each item and deciding what to do with it.

- First if it is not worth keeping, then remove it.
- Next if it is a single doable action, then place it the Actions List.
- If is a goal that requires multiple actions, then place it in the Projects List.
- If the item is to be delegated to someone else, then place it in the Tickler.
- If it is something you are unable to deal with at that point in time but may want to revisit in the future, then place it in the Waiting List.

When you attempting to get work done, simple check the Actions List (usually sorted by where the action can be performed) and intuitively select an item based on where you are, how much time you have, and how much energy you have.

Check over you Projects List as often as you feel is necessary to ensure you are making progress towards your goals by checking that you have actions listed that will bring you closer to completing those projects.

1.5 Traditional Implementation

The traditional implementation of this system uses loose paper, paper trays, file folders, and paper cabinets.

While affective this system uses up large amounts of physical space and materials.

1.6 Digital Implementation

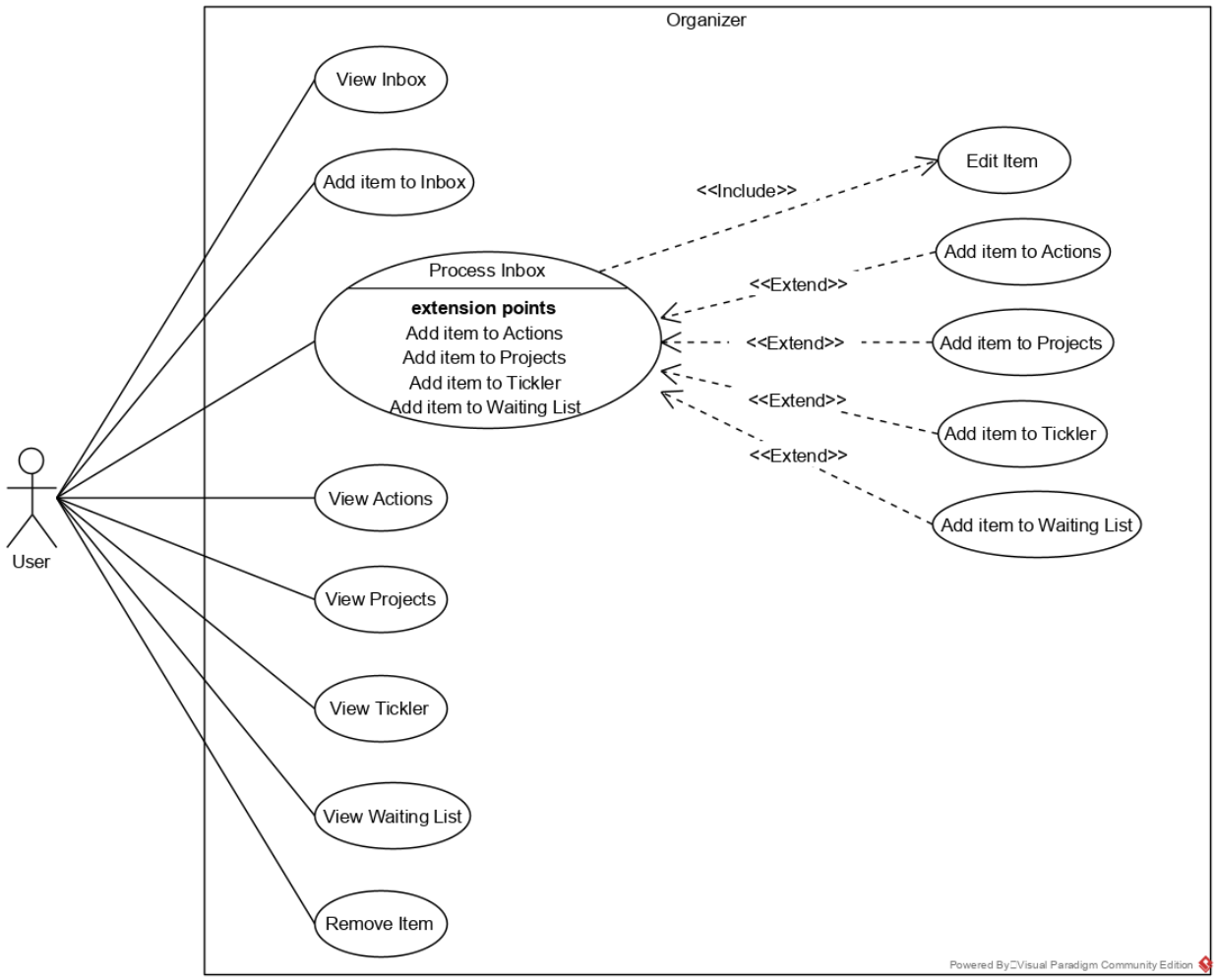
A digital implementation of this system would remove the need for space and materials, as well as reduce the cost start using this system as users would only need to install an application as apposed to by the necessary materials and office supplies.

In addition, users would be able to use the whole system wherever they are if implemented in the form of a mobile application. This would also potentially allow for new features to be implemented in the future that are difficult or impossible using a paper-based system.

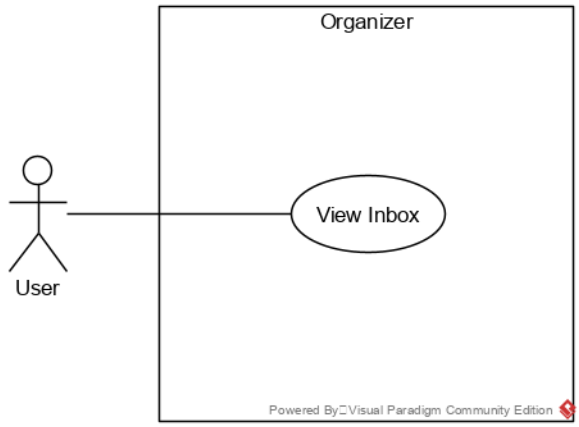
2 App Requirements

The majority of requirements for the system were gathered from the book “Getting Things Done” by David Allen, with the remainder being a result of personal experience with the system.

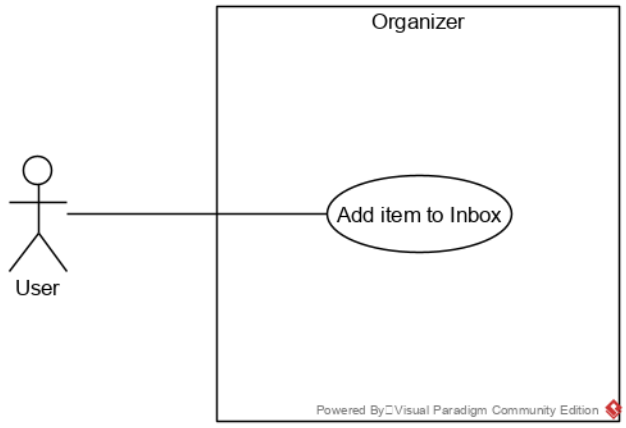
2.1 Use Cases



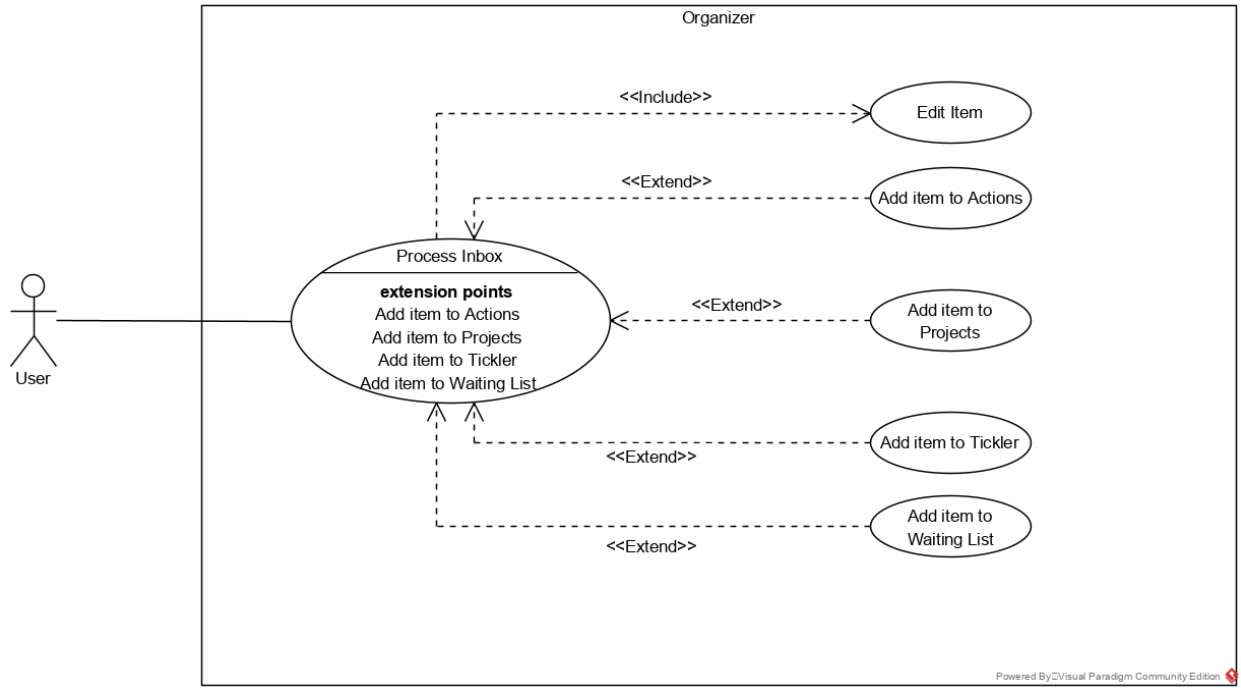
Use case name:	Overview
Actors:	User



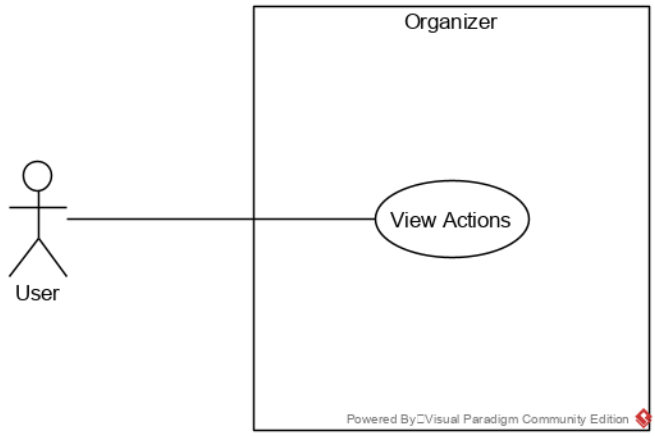
Use case name:	View Inbox
Summary:	User views their inbox.
Actors:	User
Description of main sequence:	User View Inbox System displays Items in Inbox



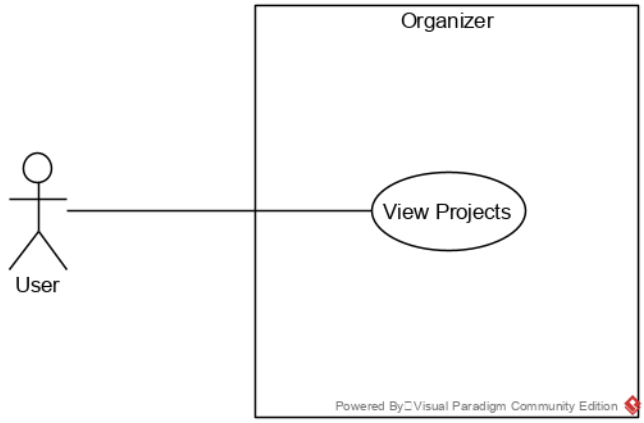
Use case name:	Add item to Inbox
Summary:	User adds an item to their inbox.
Actors:	User
Preconditions:	Inbox is displayed
Description of main sequence:	1. User selects Add item to Inbox 2. User enters Item information 3. System adds Item to Inbox
Postcondition:	New Item has been added to Inbox



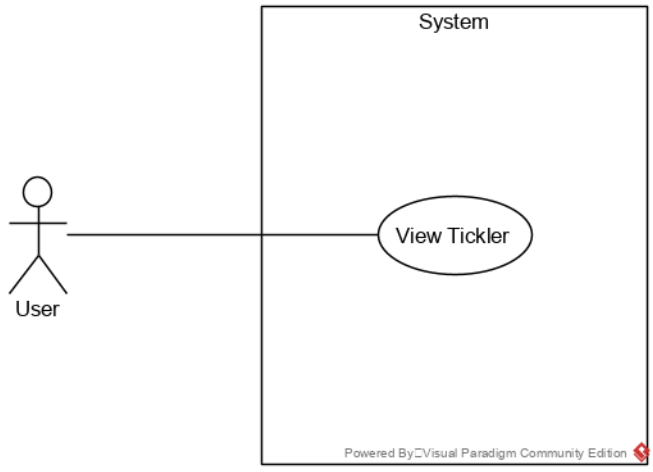
Use case name:	Process Inbox
Summary:	User selects Process Items and is show each item in the inbox, one by one and chooses an action for the item.
Actors:	User
Preconditions:	User has added items to inbox.
Description of main sequence:	1. User selects Process Items 2. System shows User an Item in the Inbox 3. User edits the information in the Item 4. User moves Item to Actions
Alternate sequence:	4b User moves Item to Projects 4c User moves Item to Tickler 4d User moves Item to Waiting List
Postcondition:	1. Shown Items have been placed in chosen locations.



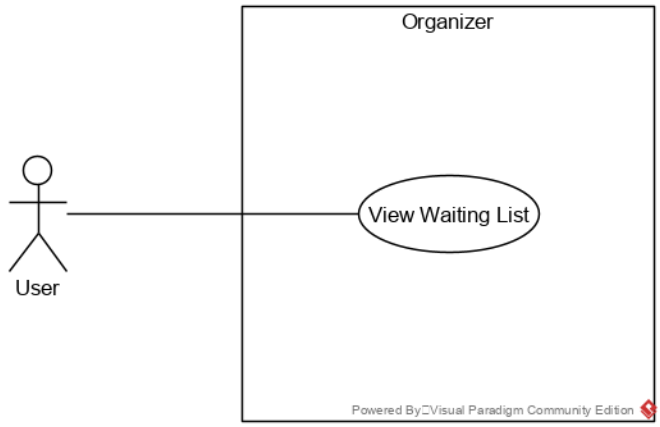
Use case name:	View Actions
Summary:	User views Actions Screen.
Actors:	User
Description of main sequence:	1. User selects View Actions 2. System displays Items in Actions



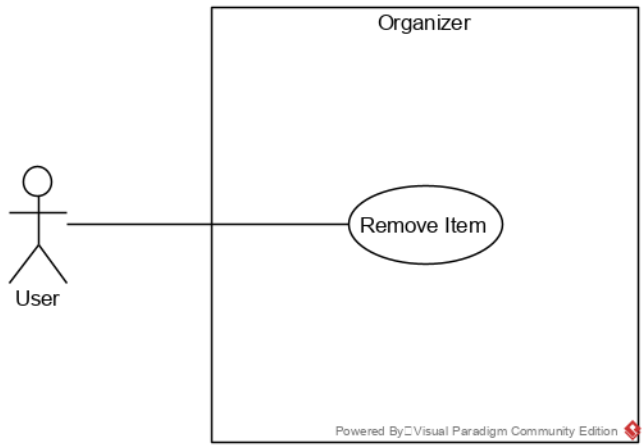
Use case name:	View Projects
Summary:	User views Projects Screen.
Actors:	User
Description of main sequence:	1. User selects View Projects 2. System displays Items in Projects



Use case name:	View Tickler
Summary:	User views Tickler Screen.
Actors:	User
Preconditions:	
Description of main sequence:	1. User selects View Tickler 2. System displays Items in Tickler



Use case name:	View Waiting List
Summary:	User views Waiting List Screen.
Actors:	User
Description of main sequence:	1. User selects View Waiting List 2. System displays Items in Waiting List



Use case name:	Remove Item
Summary:	User views their inbox.
Actors:	User
Preconditions:	User has added an Item to a List
Description of main sequence:	1. User selects an Item 2. User selects remove Item 3. System deletes Item
Postcondition:	Item is deleted

3 App Design

3.1 Class Diagram

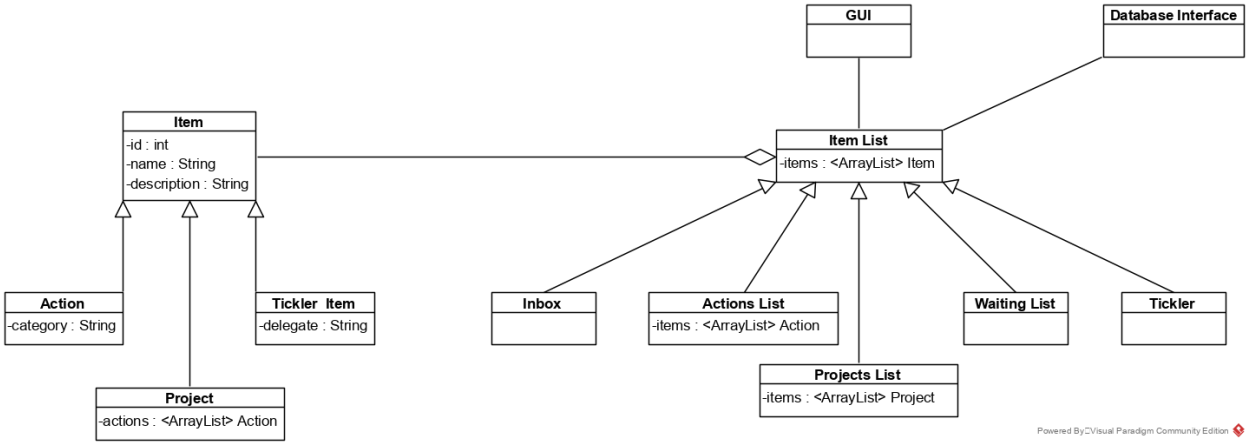


Figure 1 System Overview Class Diagram

3.2 Main Design Decisions

Navigation Drawer – A Navigation Draw will be used as they allow for a larger quantity of views to be displayed comfortably as opposed to Bottom Navigation.

Fragments for the main views – Common tactic to display main views of an app quickly as opposed to switching between activities.

Recycler View– Each Item in each category/list will be displayed with a recycler view as it is the most efficient way to neatly display information in a recurring format.

Fragments for displaying item details – When an Item in the recycler view is selected, a new fragment will appear to display the item information.

SQL-Lite Database – A local SQL-Lite database will be used for storing items between sessions as it allows for data of greater quantity and complexity to be stored as opposed to shared preferences.

4 Current Status of App

Inbox was implemented to store basic items for the time being.

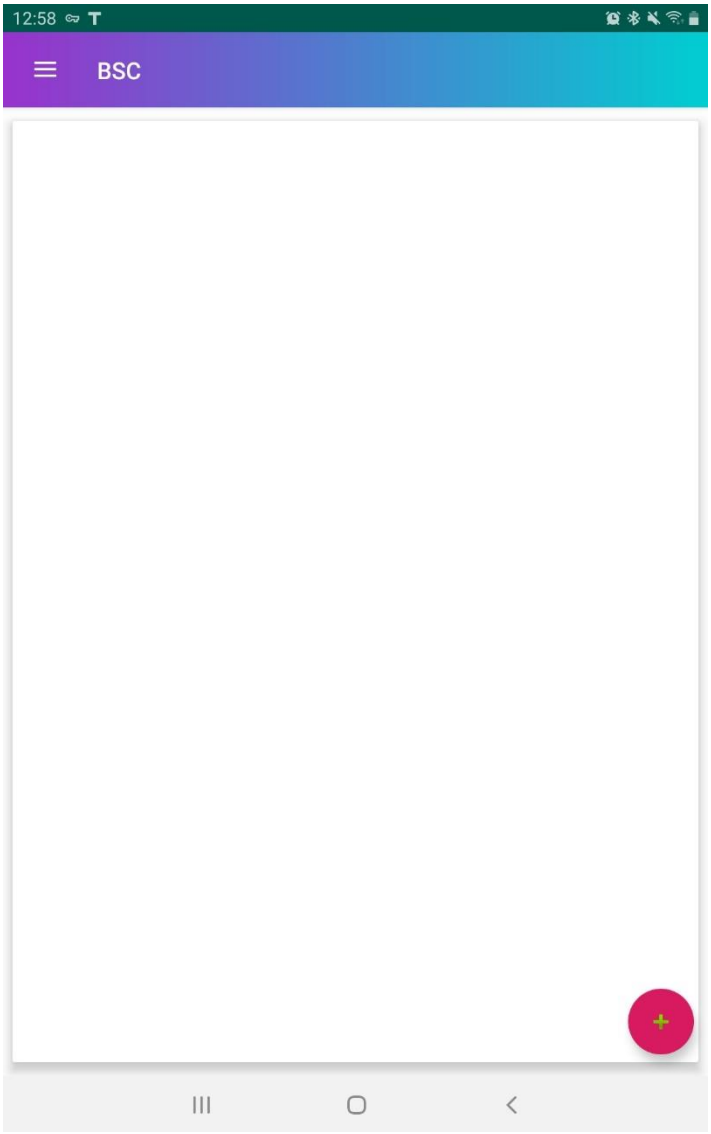


Figure 2 Inbox Empty

Add Item Screen was implemented to add the basic Item to the Inbox.

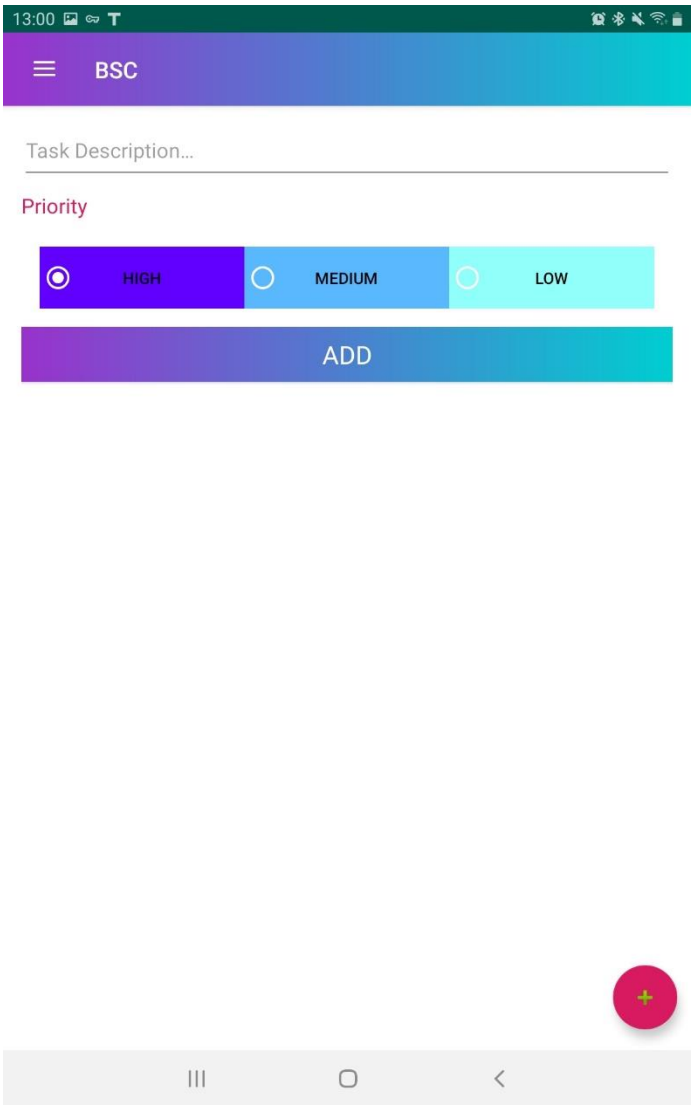


Figure 3 Add Item Screen Empty

Items currently have fields for task description and priority, though this can be subject to change.

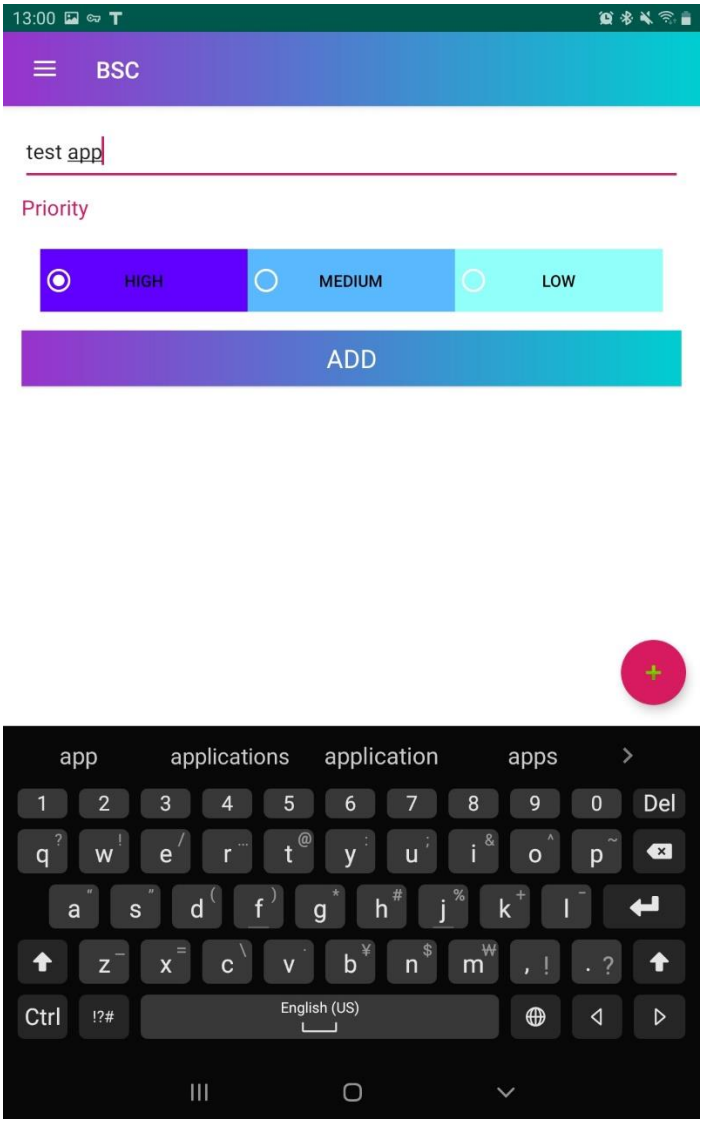


Figure 4 Add Item Screen filled out 2

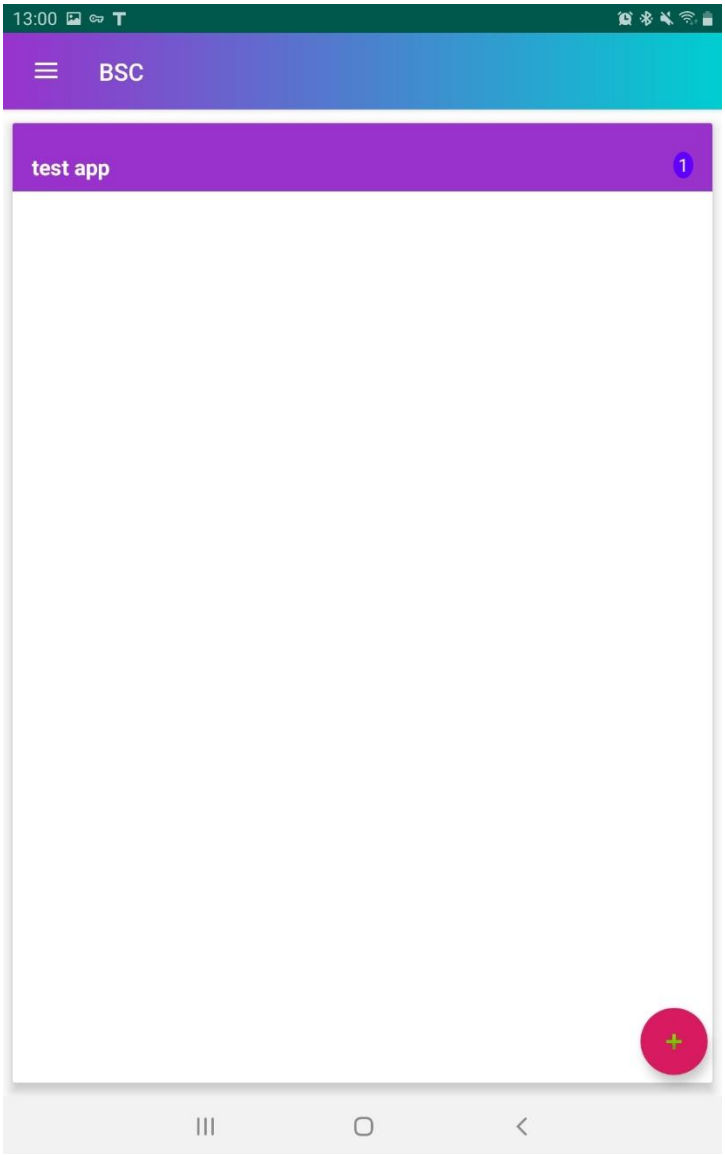


Figure 5 Inbox with Item

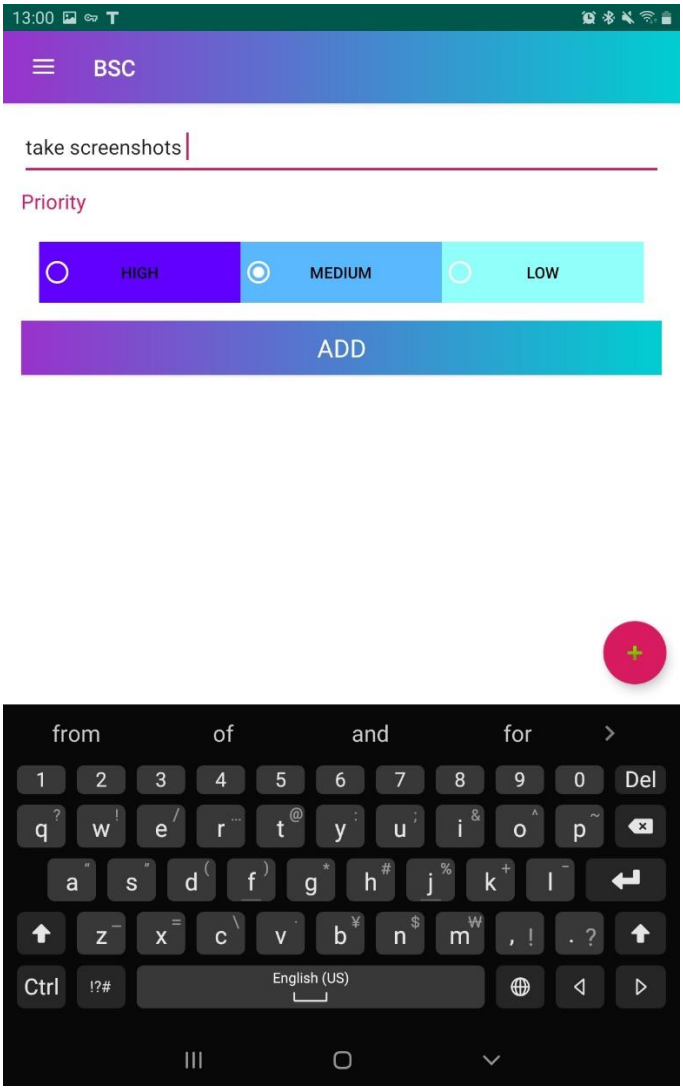


Figure 6 Add Item Screen filled out 2

Toasts are display whenever Items are successfully added, to provide additional feed back to the user and confirm the action has happened.

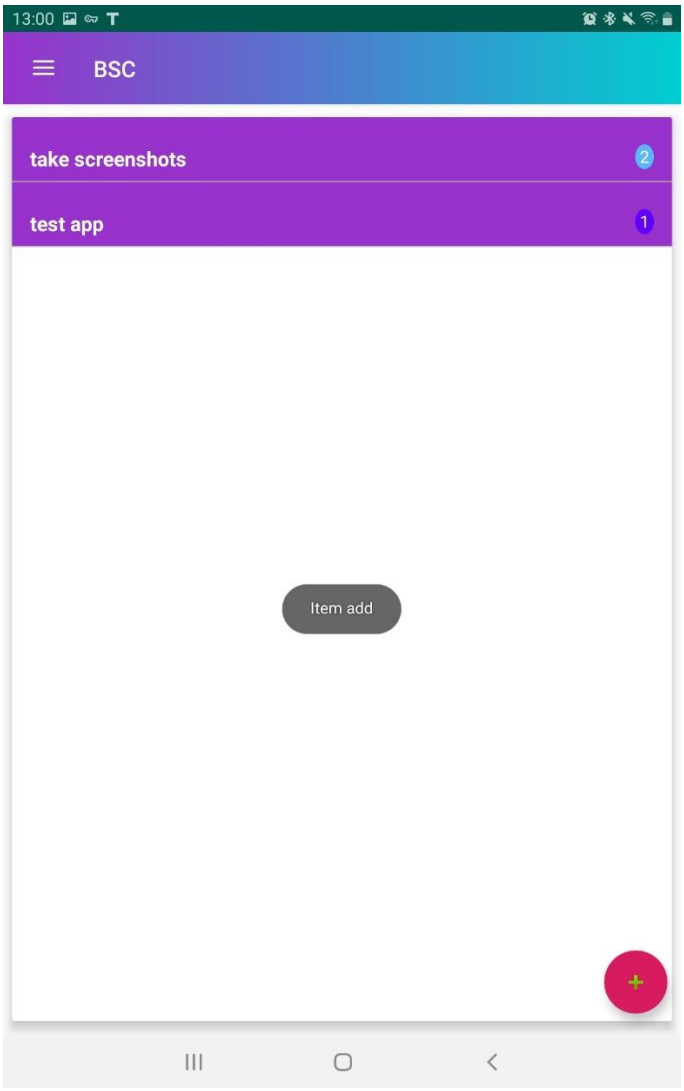


Figure 7 Inbox with multiple Items and Toast

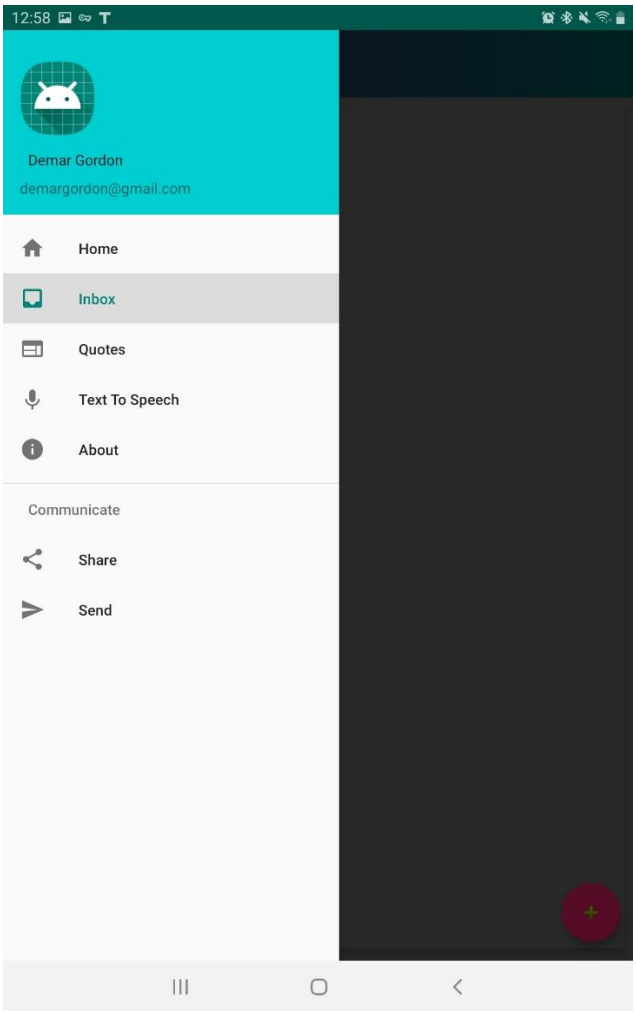


Figure 8 Navigation Drawer

WebView was implemented to display motivational quotes from ” https://www.brainyquote.com/topics/motivational-quotes”.

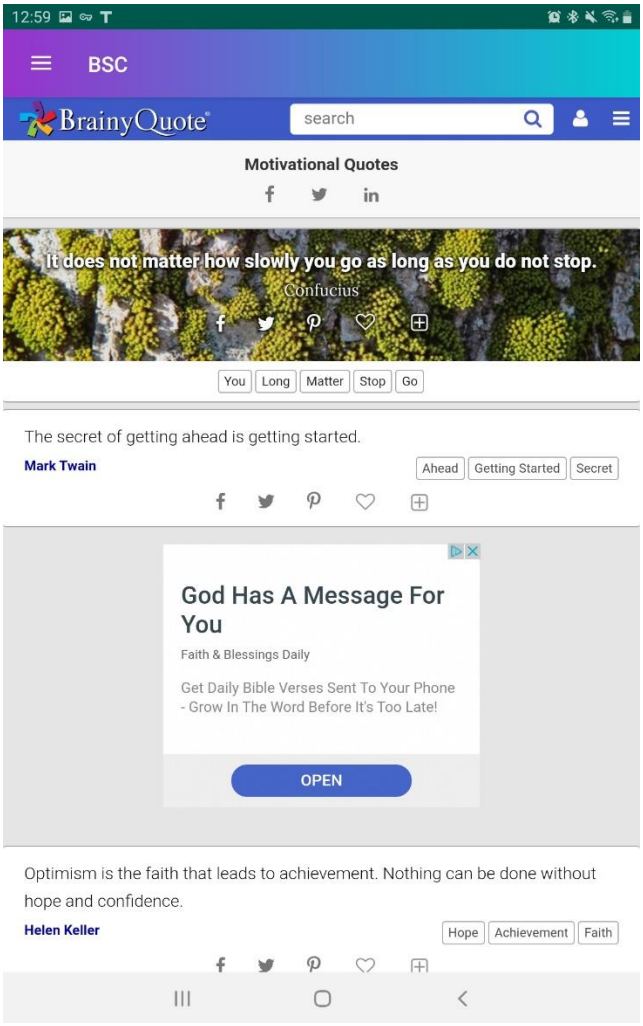


Figure 9 Web Viewer

Text to speech was added with the hopes of being used for built in accessibility options, for the visually impaired, in the future, however the feature is currently limited to single screen.

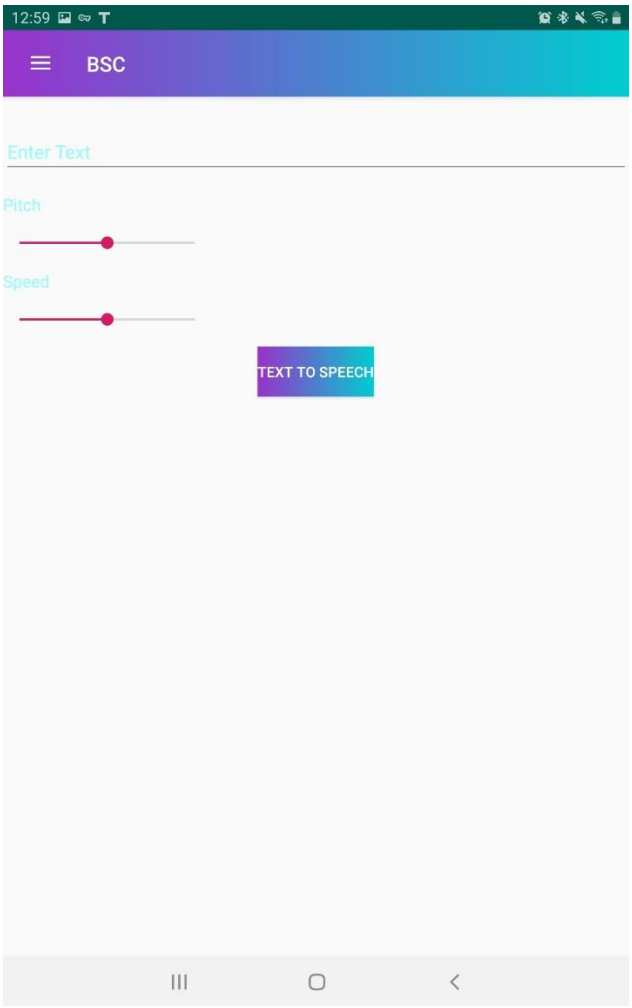


Figure 10 Text to Speech

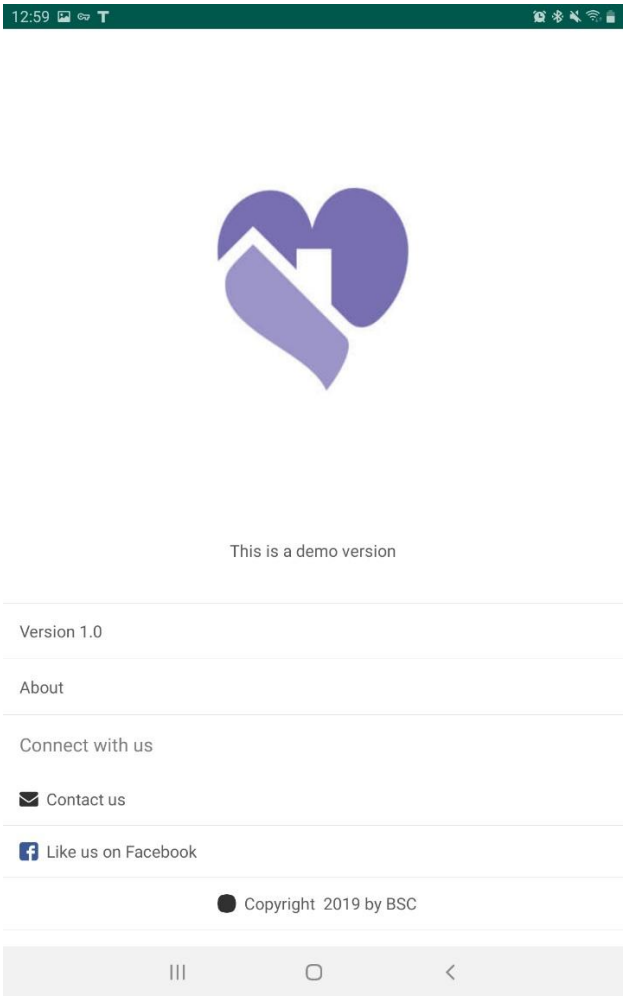


Figure 11 About Page

5 Conclusion

Though most of the features were not able to be implemented, we were able to create the foundation for the design.

Currently implemented features include:

- Navigation Drawer
- Inbox
- Inbox Items
- Addition of Inbox Items to Inbox
- Removal of Inbox Items
- SQL-Lite Database
- WebView
- Recycler View
- Text to Speech

In the future the remaining functionality can be implemented.