Overview

This document lists the inventory of tasks required by the Final Project of the **Android Development for Beginners** course by Galileo.

Proof of completion is provided by screenshots and by code inspection of the source code (in the attached .zip file).

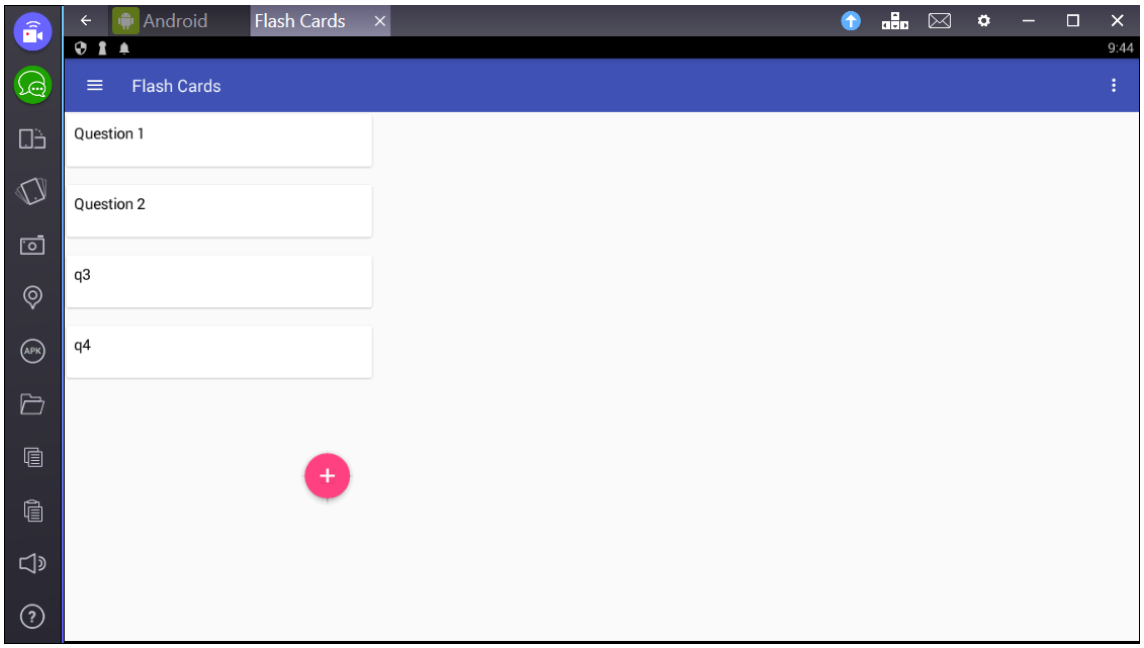
The following is a list of final project tasks with their respective screenshots and a brief explanation.

Task inventory

#	Task	Points	Test Result
1	The application contains at least 2 activities:	**15**	**Pass**
	1.1 MainActivity with a list of questions		**Pass**
	1.2 Activity showing the answer of the selected question		**Pass**
2	The MainActivity should use a recyclerview to display the list of questions (cards)	**10**	**Pass**
3	The MainActivity contains a floating action button that allows adding new cards.	**5**	**Pass**
4	New cards(question and answer) should be added using a dialog	**10**	**Pass**
5	Cards(question and answer) should be stored in a database	**15**	**Pass**
6	The application use a content provider to share information to other apps.	**15**	**Pass**
7	The initial information is read from a JSON file from the resources	**10**	**Pass**
8	The application use Job Scheduler and Notifications to set periodic study reminders	**10**	**Pass**
9	The app contains a widget that display a random question. Not done because I'm using the Bluestacks emulator and Bluestacks doesn't support the display of widgets.	**10**	**Not done**

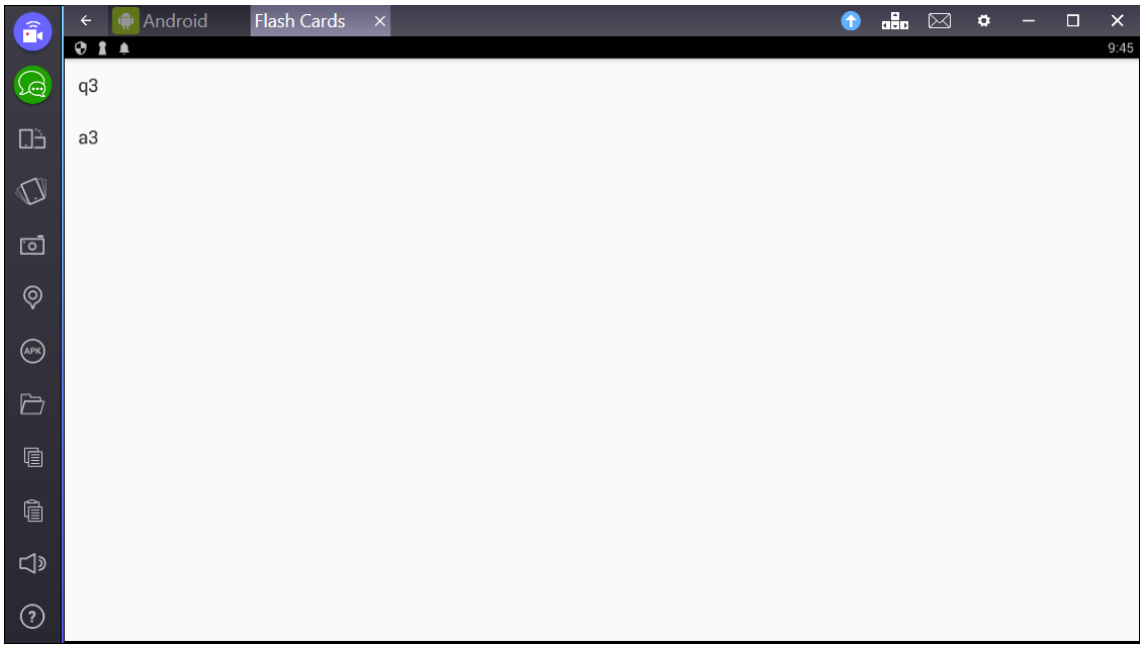
Screenshots of each task tested follow.

1.1 MainActivity with a list of questions

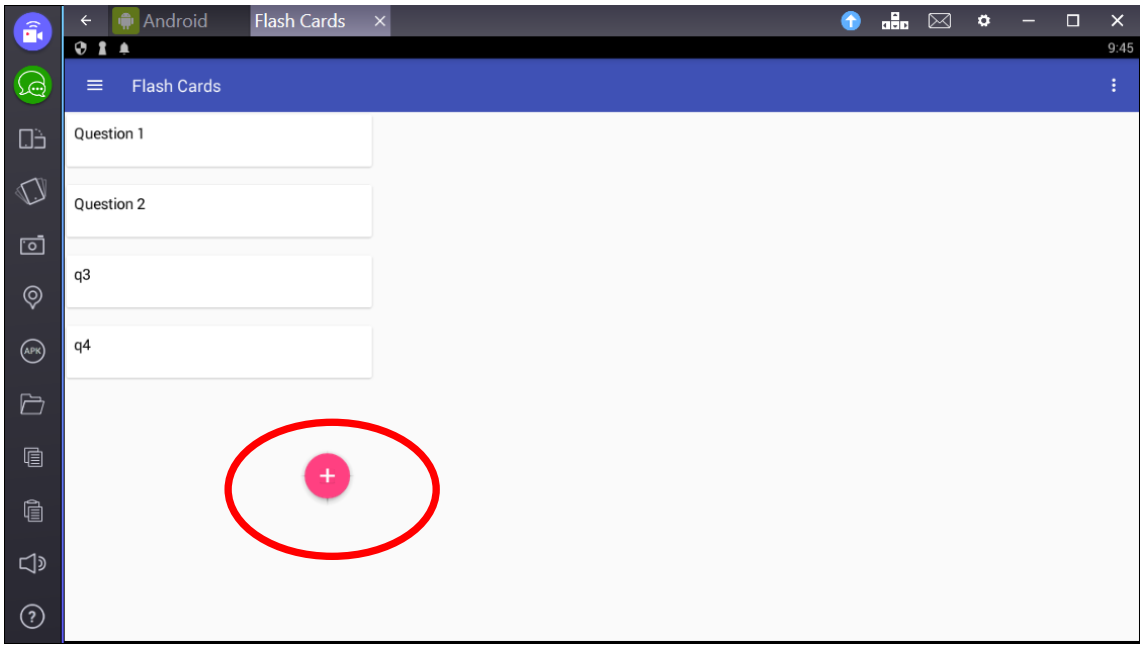


This is done with a Recycler view.

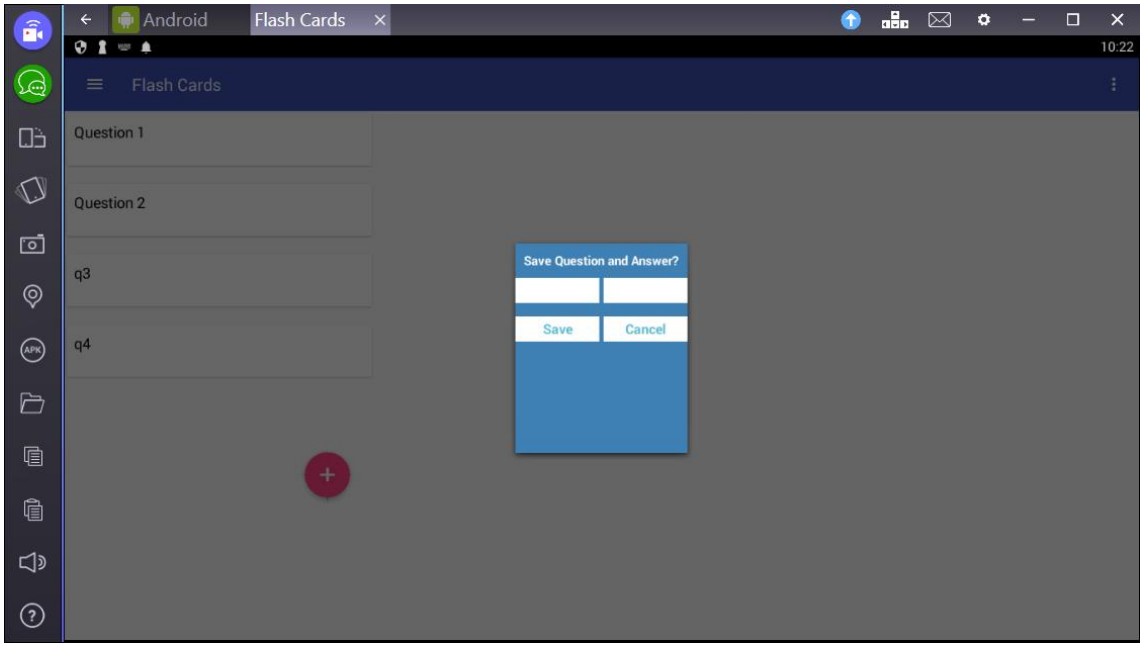
1.2 Activity showing the answer of the selected question



3. The MainActivity contains a floating action button that allows adding new cards.



4. New cards (question and answer) should be added using a dialog



This dialog allows configuring the frequency of notifications (in seconds) and whether notifications are activated or not. Notification configuration values are stored in Shared Preferences. Configuration values take effect after clicking the **Save** button and on application start.

5. Cards(question and answer) should be stored in a database

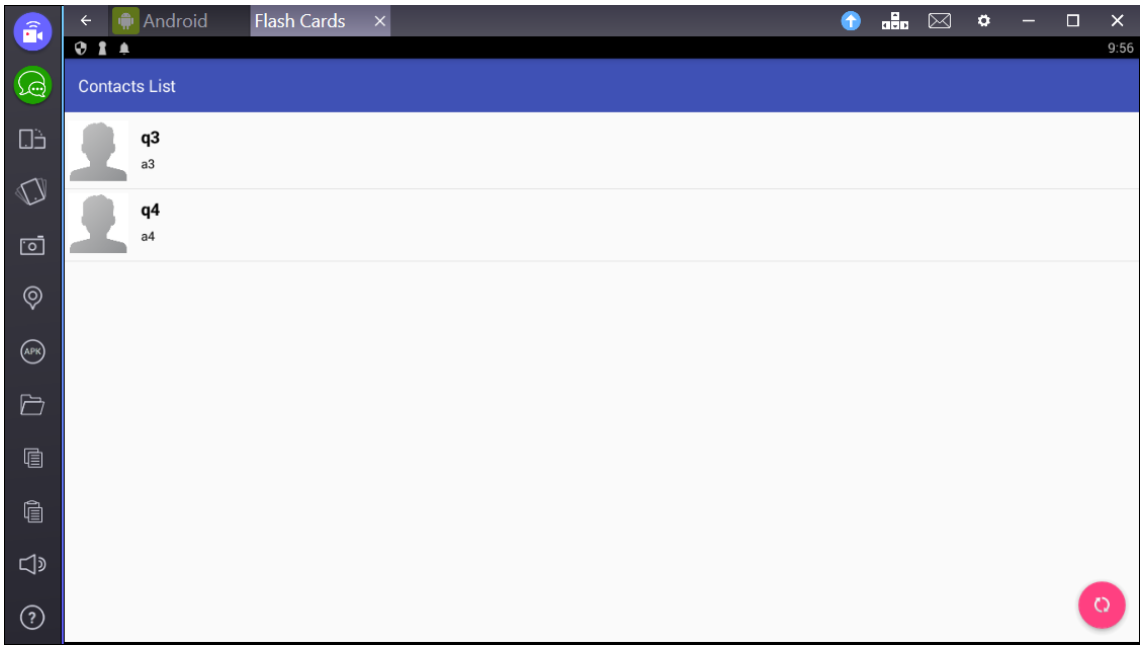
Note: the first two records in the Main screen are **not** stored in the database. These two rows are read from a Json file. The third row onwards is stored in a SQLite database.

6. The application use a content provider to share information to other apps.

The content provider may be tested with the project in the folder named **ContactsListApp** under the **Final Project** folder.

Note: the first two records in the Main screen are **not** stored in the database. These two rows are read from a Json file. The **ContactsListApp** will only read from the third row onwards.

Note: If you don't add new rows using the Questions Dialog the **ContactsListApp** will not display any rows.

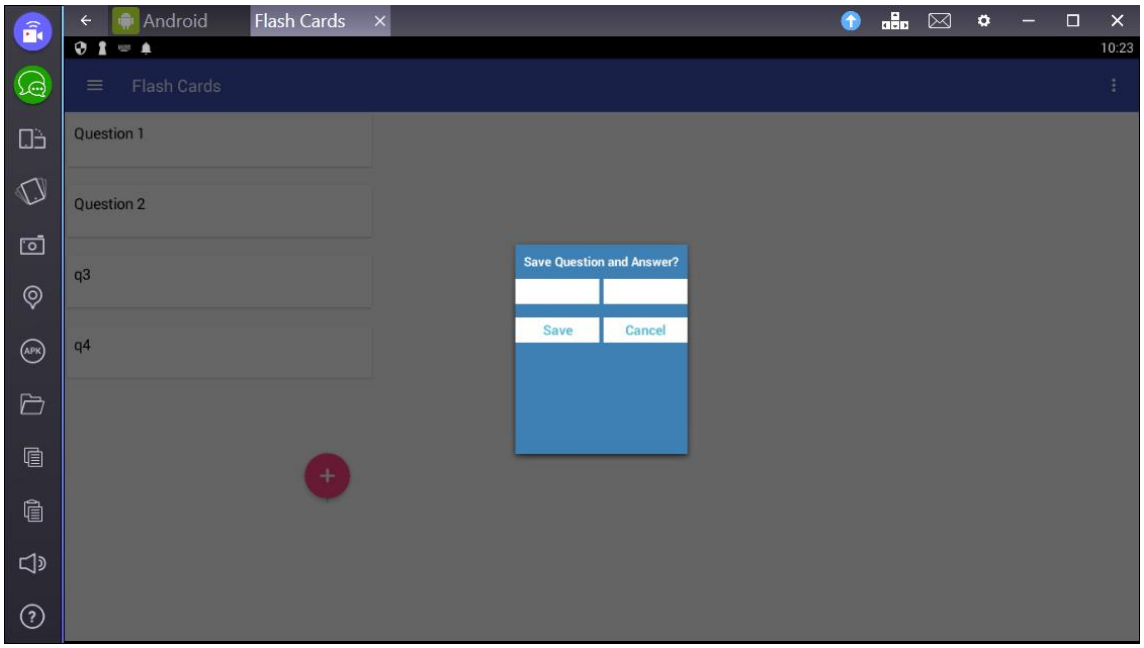


7. The initial information is read from a JSON file from the resources

The first two records in the Main screen are **not** stored in the database. These two rows are read from a Json file.

Note: If you don't add new rows using the Questions Dialog the **ContactsListApp** will not display any rows.

The Questions Dialog appears below.

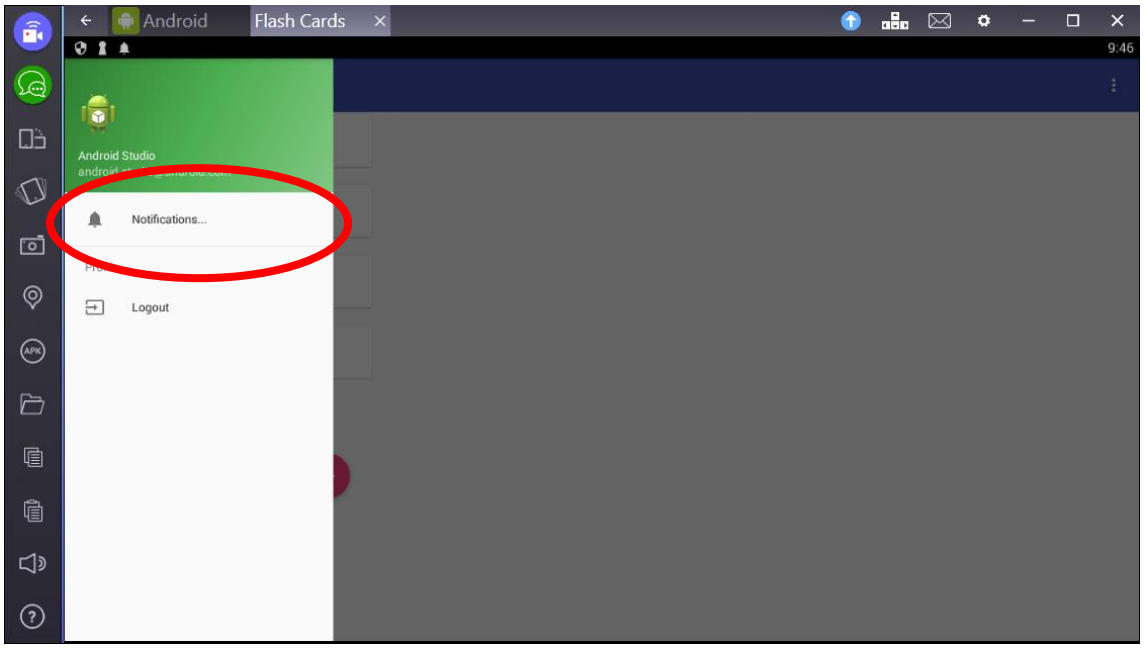


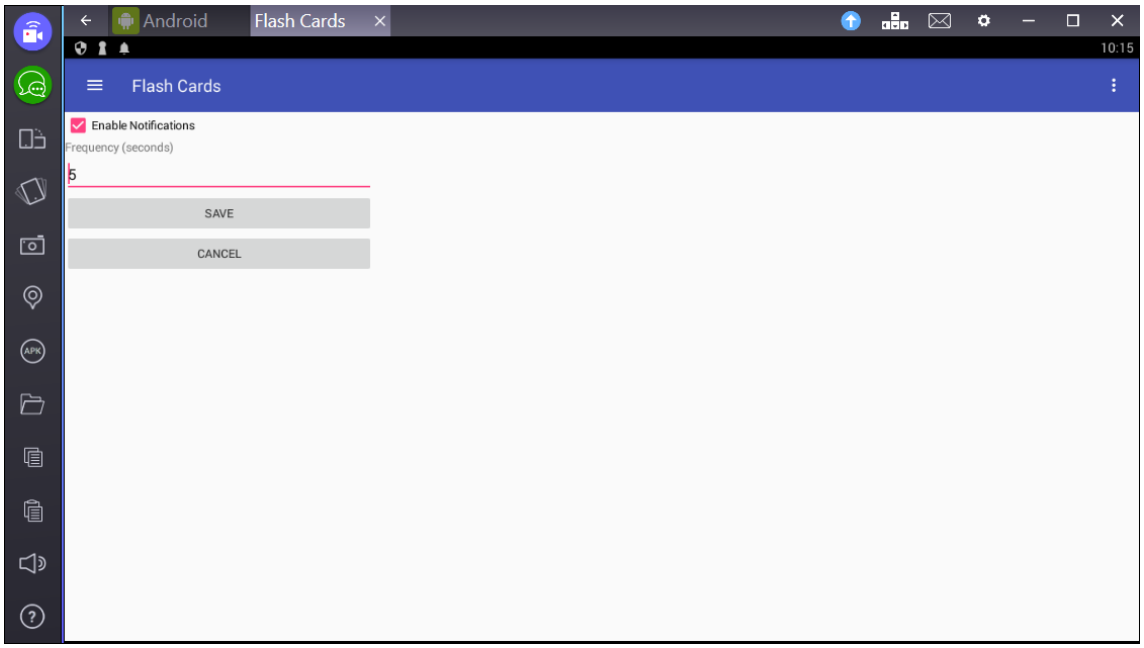
8. The application uses Job Scheduler and Notifications to set periodic study reminders

The app uses Firebase Job Service, Notifications and Shared Preferences. Notifications are issued according to Configuration Settings which are stored in Shared Preferences.

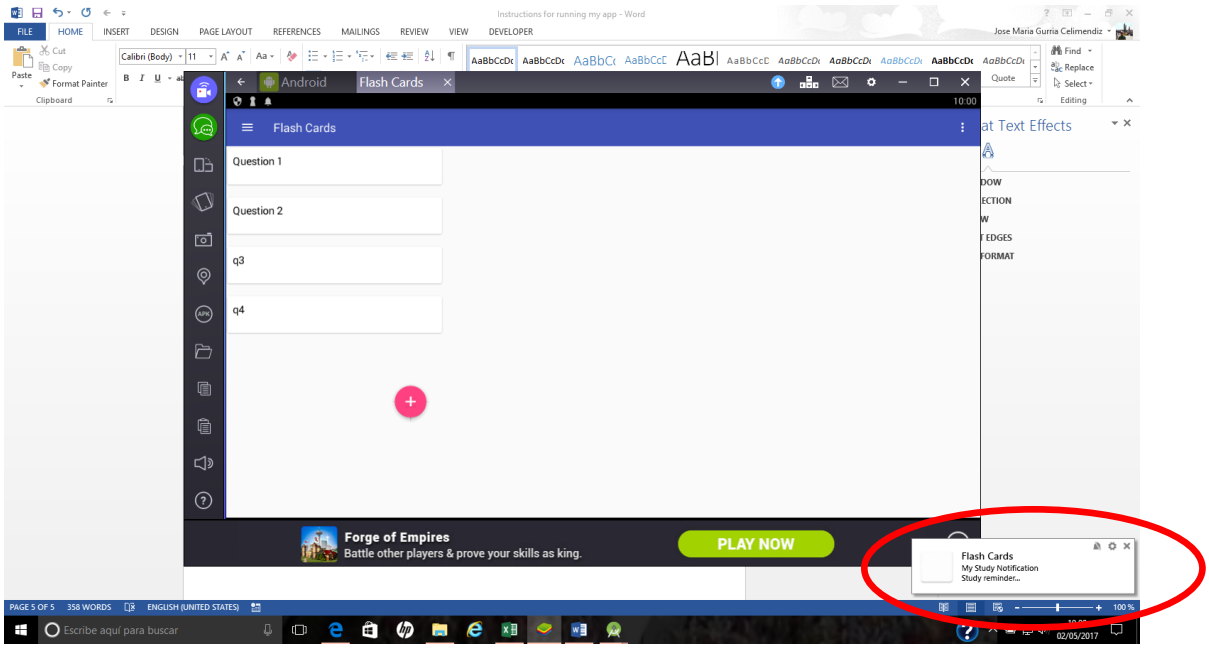
Notifications are issued even after the app is closed (after initially starting the app on Bluestacks reboot).

It is possible to set periodic study reminders, using Job Scheduler, Notifications and Shared Preferences. The user can configure Notification with a **Configuration Dialog** and store settings in Shared Preferences. The frequency can be defined in the Notifications Configuration Dialog (appears below).





The notifications appear according to the configured settings.



9. The app contains a widget that display a random question

This part of the Final Project wasn't completed because development was done using the Bluestacks emulator and Bluestacks doesn't support displaying widgets. I was unable to test widget code with the Bluestacks emulator.