Craig Soto, Ben Casey

Android App Proposal

Project 2 App: The Freegan Quest (Scavenger Hunt)

October 30, 2013

**Introduction**

The proposed Android Application is called the Freegan Quest (synonymous to “Scavenger Hunt”), which will achieve two aspects to the scavenger hunt, managing and playing. The managing of the hunt will consist of making a list of locations/objects to be found. This includes the name and the description of the location/object. Once the hunt is set up, a player may now participate in a hunt, from a list of hunts, and find the location/object based on the description and verify having gone to it by entering a keyword or image of the location/object. The normal features the app will have are: 9 activities, edit text fields, and buttons. These will be implemented to allow the user to manage lists of hunts and a list of locations/objects, and the user to start, play, finish and see the results of a hunt. The advanced features the app we will have are: database interactions, possible GPS locating, camera features, timer, and fragments. The database interactions will be implemented to store, retrieve, and update each hunt and the locations/objects within it by the manager or player so that the app can store this information after the app has been closed, stopped or interrupted. The GPS locating app will be used to verify the location of a player to check it against what the manager intended if the manager chooses that option. The camera feature will be used to take a picture of the locations/object, if the manager chooses that option. The timer will be used to time the players from start to finish of the hunt. The fragments will be implemented for inserting a new hunt and for easy viewing of the list of locations/objects in a specific hunt. Also, fragments will be used for confirmation purposes when finishing a hunt or deleting a location/object/hunt. The new features for App 3 we will incorporate are the three menu options (settings, help, about) and a Manager Mode action button for the persistent item.

* Scavenger Player
  + Manager mode
    - Add new hunt
      * Dialog – enter name of hunt
      * Confirm
        + Add to DB
      * Cancel
        + Go back to Add new hunt
    - Tap item in list
      * Add hunt item
        + Add item

Add Item Activity

Add name

Add Location

Add Description

Submit

Add to DB

Return to Add hunt item

* + - * + Long press

Delete

* + - Long press
      * Delete
        + confirmation
  + Hunt Drop Down
    - Select hunt from drop down
    - Already finished
      * Can’t restart
      * If time restart hunt
  + Start
    - Go to list of locations in hunt
    - Tap item
      * Enter in work/picture
    - Finish
      * Confirmation
        + Go back to player start menu
    - Timer
      * No user altering, continual ticking.
  + Results
    - View only
      * Description, player input, time

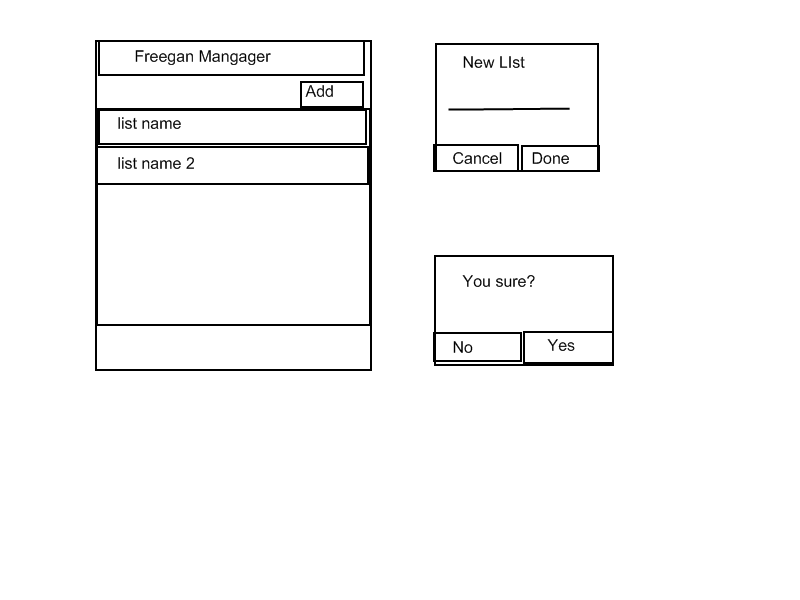
**Manager Mode**

When the manager mode starts the first time, the Manager Activity will display an empty list of hunts and an “Add” button. When the manager clicks “Add”, an input dialog will appear that will allow the manager to type in the name of the hunt they want to add to the list. After typing in the name of the hunt and clicking “Ok” the app will add the hunt to the database, the dialog will disappear, and the Manager Activity will be visible with the name(s) of the hunt(s) in the list. If the manager performs a long press on the hunt name, the app will display a list of options. Among those options will be edit and delete, which will allow the app to interact with the database accordingly. If the manager decides to edit the name of a hunt the activity will pass the key to the hunt name in the database to the dialog. This is so that the dialog will be able to retrieve and edit the information before updating it in the database. For the deletion of a hunt, it will have a confirmation to make sure the manager wants to delete it. The manager will be allowed to add as many hunts as they need (in theory) but for our app we will make sure the app can handle at least 7 with no issues.

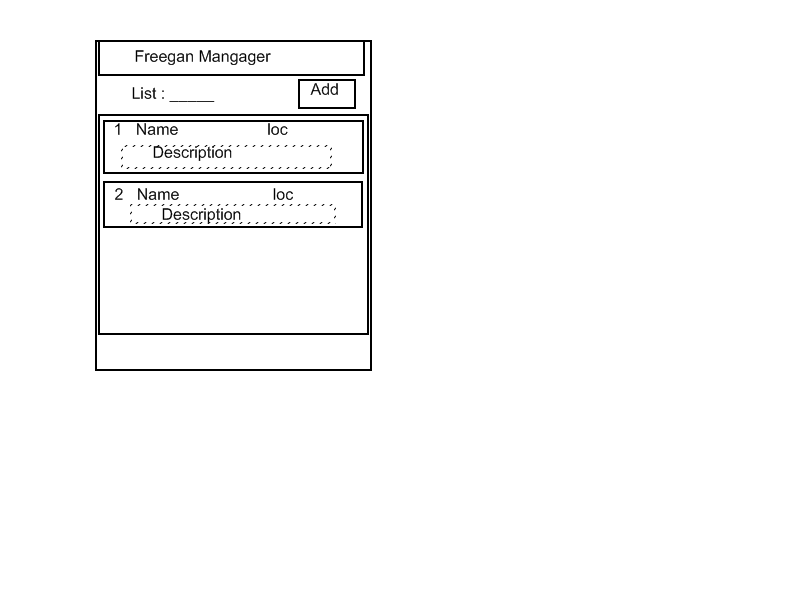
If the manager performs a tap on a hunt, the app will go into the add hunt items activity. It will pass in the hunt name. This activity will display a list of hunt items/objects, and Add button, and the name of the hunt. The display of the items/objects will have its name, description and location. If the manager performs a long press on an item, it will bring up a list of options which will be delete or edit. On a delete, the app will just go ahead and delete the corresponding information from the database. On an edit, it will bring up the add item activity with the fields filled out from the information stored in the database.

If the manager clicks on the “Add” button in the add items activity, it will bring up the add object activity. It will have a name text field, location text field, a description text area, and a submit button. The name text field is used to input what the player will be trying to find. The location text field is used to input the location of the object using gps coordinates. The description text area is used to input the clever description of how to find the object. The description is what the player will see to find the object. When the manager hits the submit button, it will send all the information from the text fields and add them to the database. The app will then send it back to the add items activity with the list updated.

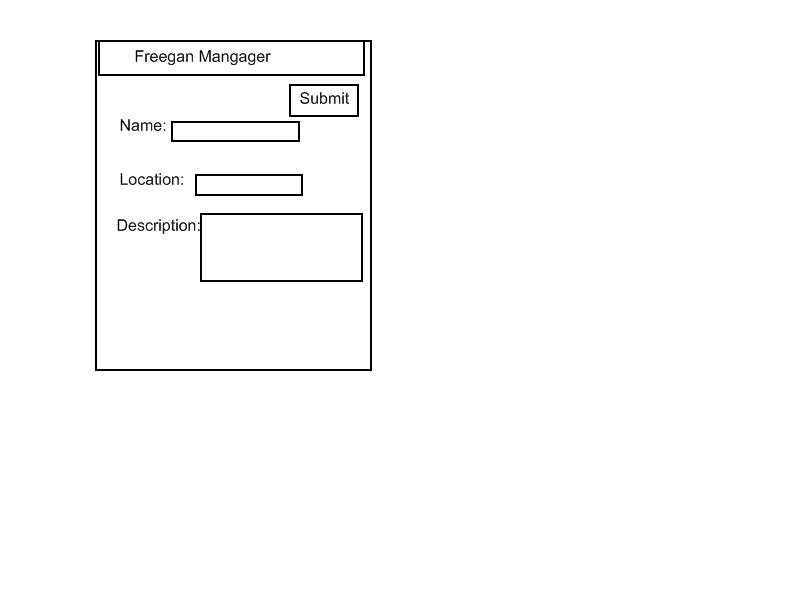
The Manager Activity will look like this:



The add items activity will look like this:



The add object activity will look like this:



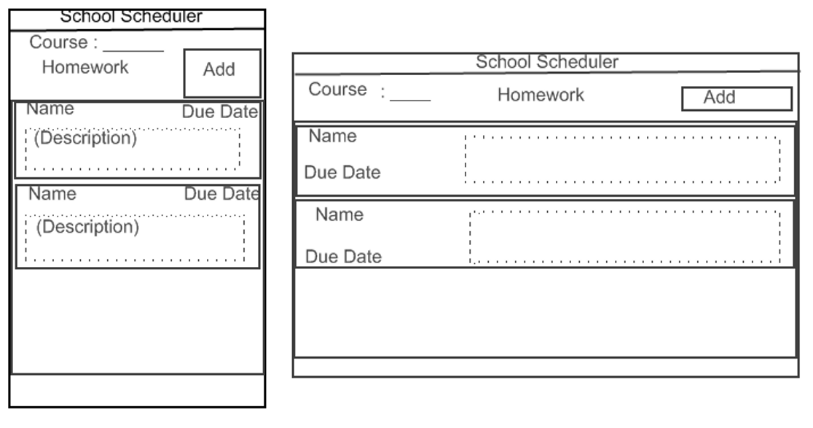
**Add/Edit/Delete Homework**

Once there is a least one course in the list of courses, the user has the ability to add homework to that course. To do that, the user will have to click on the course. This will pop up a fragment that displays an empty list of homework (initially) and an “Add” button. When the user clicks the “Add” button, the app will start the Homework Activity, which will be used to add homework and its information to the database. The Main Activity will pass the name of the course to this activity so that when the activity adds the homework to the database, it can associate that homework with the course.

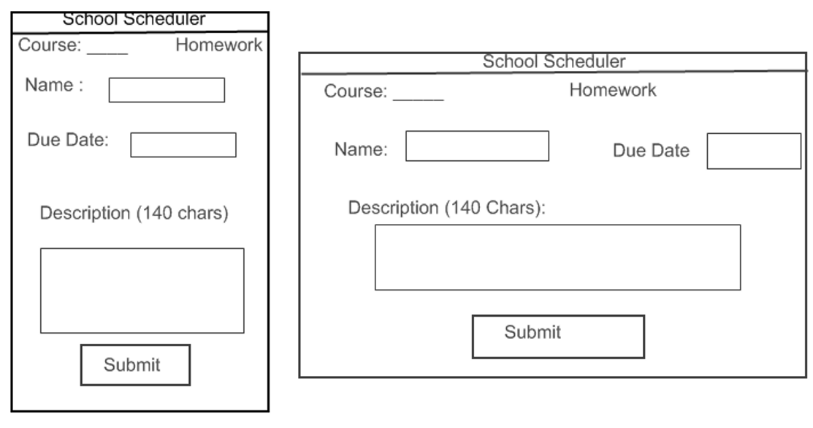
The activity will display the name of the course, have edit text areas for the name of the homework and a description of the homework, a date selector field (to select when the homework is due), and a “Submit” button. The description will be limited to 140 characters so that when it appears in the list of homework for the course, the user can see a quick description of that homework. After the user has filled in all the criteria and click submit the activity will add this information to the database and return back to Main Activity with the fragment list of homework visible. If the user doesn’t enter in a name or a date the app will enter in generic input (“Untitled”, today’s date) and add this information to the database. If the description is empty it will remain empty when added to the database. If the user clicks the back button the app will return back to Main Activity with the fragment list of homework visible. No information will be passed from the Homework Activity to the Main Activity/fragment because the fragment will need to show the updated list of homework and will retrieve that from the database.

Similar to the list of courses, if the user performs a long press on the homework, the app will display a list of options. Among those options will be edit and delete, which will allow the app to interact with the database accordingly. If the user decides to edit homework information the Main Activity will pass the key to the homework information in the database to the Homework Activity. This is so that the activity will be able to retrieve and edit the information before updating it in the database. The user will be able to see all information needed for the homework (name, date, description). This will make it so that the user won’t have to navigate to a new activity to see the full information, because it is already visible. The user will be allowed to as many homework assignments as they need (in theory) but for our app we will make sure the app can add at least 10 with no issues.

The fragment containing a list of homework assignments will look like this:



And the Homework Activity will look like this:



**Life Cycle**

Upon starting the Android app, the app will initialize the variables used throughout the app in the onCreate() method. These variables will be used in the onStart() and onResume() methods to set up the app layout. As the user interacts with the app, these variables will be updated and saved incase the app is paused, stopped, restarted and resumed. This will help the user resume activity with the app in the case a phone call occurs or the user needs to exit the app. The variables that will need to be saved will be any variable not yet saved to the database, such as: the name of a course in the open dialog, the name of a homework, due date, and description in the Homework Activity. When in the Homework Activity, if the user submits or cancels (click back button), it will use the onDestroy() method and clear all the variables that were saved for the homework information.

**Submissions**

For the intermediate submission, we will have activity and fragment navigation complete. A user will be able to click the “Add” button for courses and homework assignments. This means that the course dialog and homework fragment will appear but won’t be able to view any information added by the user because the database interaction with the app won’t be implemented. The app will be able to navigate to the Homework Activity, insert information in the editable areas, and navigate back to the Main Activity. Again, no information edited here will be visible in the fragments because it will not be inserted into the database. This will all be done for the portrait view only.

For the final submission, we will have all functionality complete for the app. In particular, the app will allow the user to properly add/edit/delete courses and homework assignments and have a working database that handles adding/editing/deleting courses and homework assignments. Limitations will be enforced and special homework cases will be handled (empty name or date). Functionality for the landscape view will also be completed, so that the user can rotate the device as needed.