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.

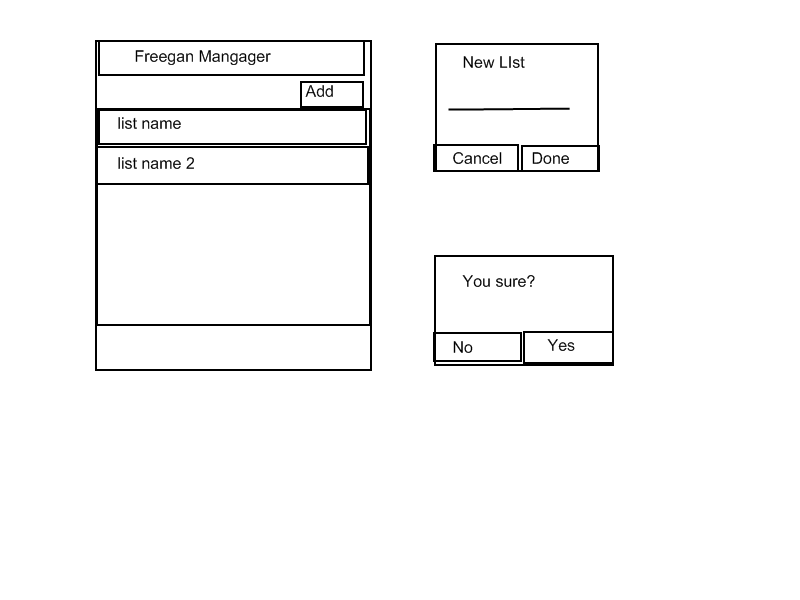
**Manager Mode**

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 other option is the show location option on the long press. If the manager decides to show the locations, it will start the show Locations activity. This will take all items in the hunt and display just their locations and their names. There could possibly be an option to show them on a map. On the back button, it will then go back to the manager activity. 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.

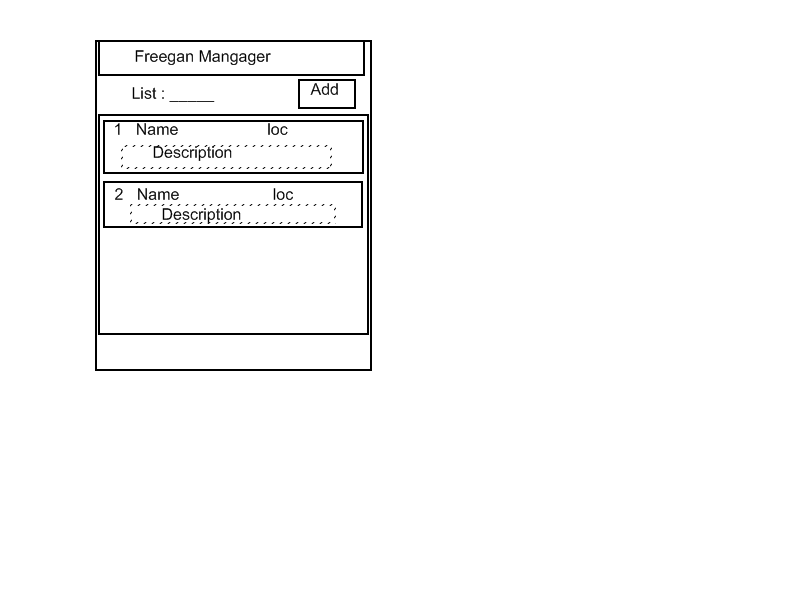
If the manager performs a tap on a hunt, the app will go into the add hunt items activity. It will pass in the hunt name. This activity will display a list of hunt items/objects, and Add button, and the name of the hunt. The display of the items/objects will have its name, description and location. If 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.

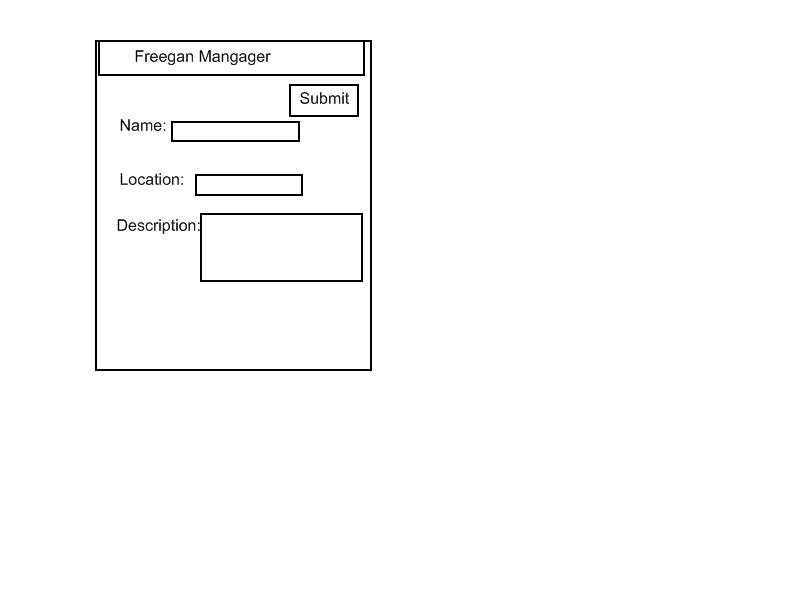
The Manager Activity will look like this:



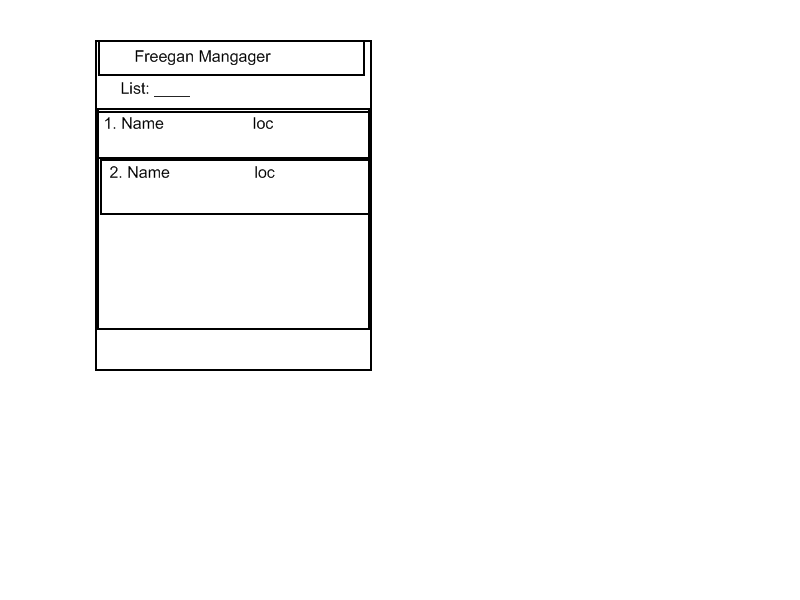
The add items activity will look like this:



The add object activity will look like this:



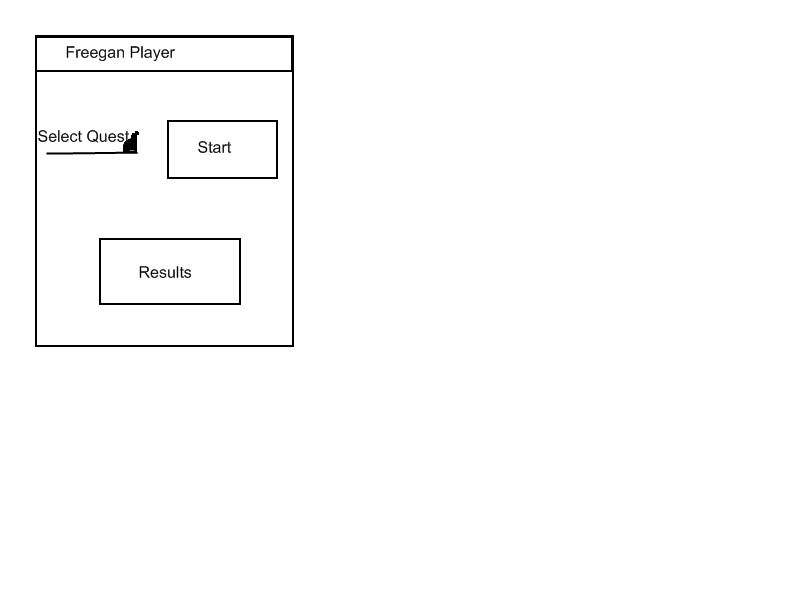
The show locations activity will look like this:



**Player Navigation**

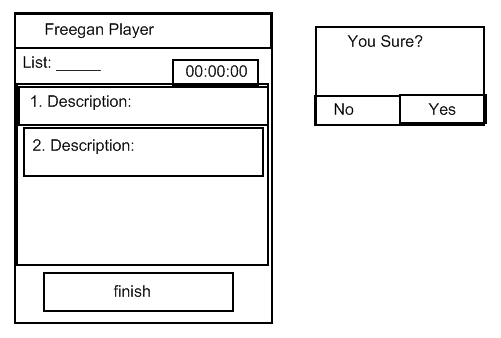
When the app first opens the Main Activity will display the player’s options. In this activity the player will be able to select a scavenger hunt, start the hunt, and view the results of a hunt. In order to start a hunt or see the results of a hunt the player will need to select a hunt from the dropdown of hunts. If the player doesn’t select one, the start/result button will be grayed out and now be clickable. In order to add a hunt, the player will need to enter manager mode (see above). If the player is able to and does select a hunt to play, the start and results buttons will be clickable. When the player clicks the “Start” button, the app will start the Hunt Activity which displays the list of locations/objects that the player needs to find. If the hunt has already been completed or is in progress, the player will not be able to start the hunt because it will not be displayed in the select a hunt drop down. When the player clicks the “Results” button, the app will start the Results Activity which displays the list of locations/objects, the player input from the hunt, and the total time it took to complete the hunt. If the hunt has not finished, the results button will be greyed out and not clickable.

The Main Activity will look like this:



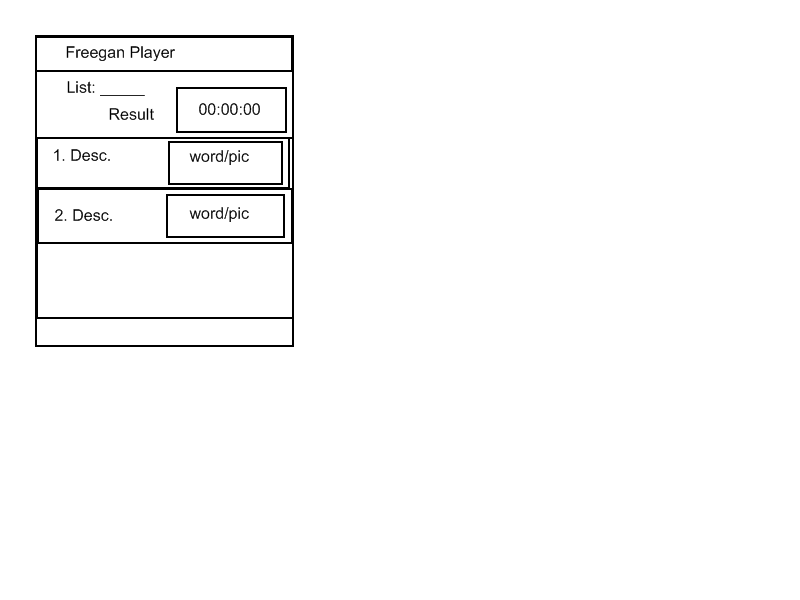
When the player is in the Hunt Activity the player will only be able to update the keyword or take a picture (based on what the manager of the hunt decides) and finish the hunt. When the player selects a particular object in the hunt, two possibilities will occur. If the manager has decided to use a keyword a fragment dialog will appear and the player will be able to enter in a keyword. The player can edit this keyword as long as the hunt hasn’t finished. The other option is for the user to take a picture of the location/object. The app will use the camera on the phone in order to take the picture. When the player takes the picture the app will return to the Hunt Activity with the location/object list item displaying a small thumbnail image in the list item. Similar to the keyword, the player will be able to update the picture by clicking the location/object and take a new picture. Whenever the player updates a list item, the app will update the database with the new information. The player will only be able to finish the hunt if all locations/objects have a keyword or picture inserted by the user, until that criteria is met the “Finish” button will be greyed out and not clickable. When the player fills in all the keywords/pictures for the scavenger hunt items and the player clicks the “Finish” button, the timer will stop, the database will be updated with the hunt time, start the Main Activity, and finish() the Hunt Activity. To prevent the player from cheating, the timer will continue to run until the hunt is finished.

The Result Activity will look like this:



The Result Activity is only going to be used for viewing the results of a hunt. This activity is mainly used by the manager to verify the player has the correct pictures or keywords, or the fastest time. The list locations/objects will contain the description and the keyword/picture similar to the Hunt Activity. The user will be able to click the list item and view the player input.

The Result Activity will look like this:



**Life Cycle**

Upon starting any of the app activities, the activities will initialize its variables in the onCreate() method. 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 variable that will be saved in the paused and stopped methods will be the timer. Since the timer will continue to run regardless if the app is interrupted, closed, or in any other activity, until the hunt is finished, the timer will to continue to run. When the user goes back to the activity, the restarted and resumed methods will update the timer variable to be displayed to the user. The activities will use the onDestroy() method only when the user leaves the activity and doesn’t need to go back to it, like when the player clicks the “Finish” button in the Hunt Activity.

**Submissions**

For the intermediate submission, we will have manager mode complete plus the Main player screen. A manager will be able to click the “Add” button for hunts and items. The app will be able to navigate to the manager mode. There, the app can add, edit, and deletes hunts along with show locations. On show locations, it will navigate to the show locations activity and shows all the locations of that hunt. With tapping on a hunt, it will navigate to the Add Items Activity where the manager can add, edit, and delete items. On the Add button, it can navigate to Add Object Activity where the manager can add info in the fields and submit it to 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 player to select and start a hunt. In the start the hunt, it will navigate to the hunts list. On tapping of the item it will go to the Enter activity where it will either be word or picture capturing. When all items have been entered, the finish button will be able to be clicked and a confirmation will appear. It will then go back to the main screen. When the results button gets it, it will bring up the view only Results Activity. Functionality for the landscape view will also be completed, so that the user can rotate the device as needed.