

Assignment #4: Designing, Implementing, and Coding A Graphical User Interface

Due Date (Part I): Wednesday, 9/21/2011 by start of class

Due Date (Part II): Friday, 9/23/2011 by start of class

Purpose: This purpose of this exercise is for you to gain experience in designing, implementing, and coding a GUI interface to satisfy functional requirements. Start early – this assignment is non-trivial as you explore laying out a complex interface and getting multiple types of Listeners to work together.

Part I

Step 1: Read the attached abbreviated *functional requirements* list for the City of Winston Salem app. This document contains a list of functional requirements for the search query interface component of the app (the search query interface being the piece of the app that requests the information from the user in order to compose a query). The list of functional requirements has been truncated so we can focus for the moment on the GUI components of the application.

Step 2: Sketch on paper an outline of a design & layout you plan to use for the search query interface. For each part of the interface, draw a labeled arrow from the edge of your paper to the component in your drawing, indicating which type of View you plan to use for that component of the interface.

Step 3: Using the XML layout specification technique, implement to the best of your ability, the design and layout you suggested in Step 2. Feel free to make use of layouts as appropriate as well as any Views we have looked at in class. In addition, add a small TextView at the bottom of the screen. We will remove this later, but it is important for Step 4. Do not worry about any of the other requirements (MapView, ListView, switching screens, etc) you may have seen before - we will cover those later.

Part II

Step 4: When the user performs the action which should initiate the search (that is, the user uses the interface control that satisfies Requirement 1.1.7), write all of the values that were set in the other parts of the interface into the TextView. This basically means you need to implement Listeners for events as appropriate to capture setting values. If the user elects to use the device's GPS coordinates, just write (0,0) as the coordinates in the TextView. There should basically be five settings written here:

Priority: one of [High, Medium, Low]

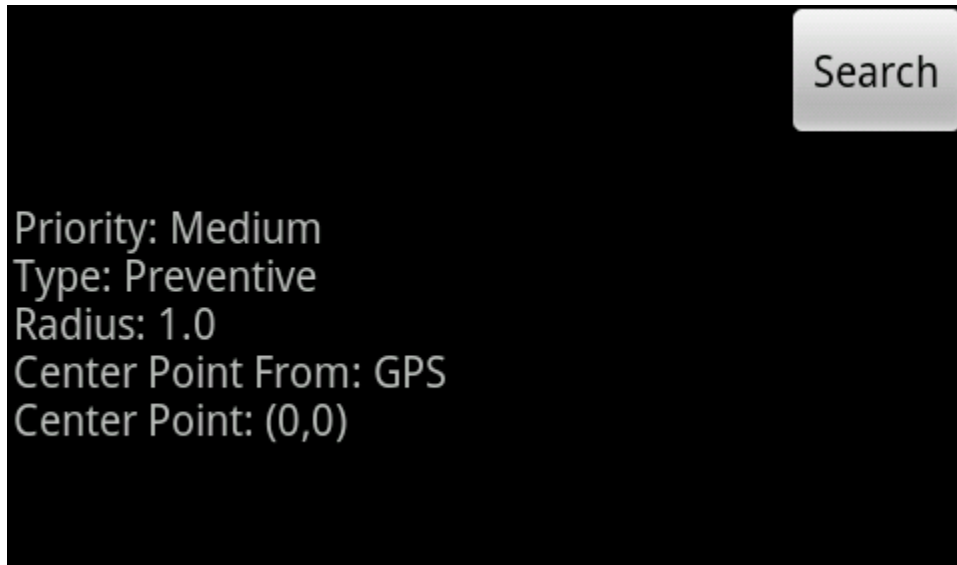
Type: one of [Corrective, Preventive, Inspection]

Radius: an appropriate value here

Center Point From: one of [GPS, Address]

Center Point: one of [(0,0), the actual address entered]

Here is an example of how the temporary TextView should be filled when the Search button is pressed. All of the five values were taken from the interface above the Search button (not shown) which I interacted with to pick these settings.



Functional Requirements (Work Order Query Activity)

- 1.1.1** Allow users to select the work order priority (High[default] Medium, Low).
- 1.1.2** Allow users to select the work order type (Corrective, Preventive, Inspection).
- 1.1.3** Allow users to select the geographical radius for work orders (default 1/2 mile, max 2 miles).
- 1.1.4** [NEW] Allow the user to indicate whether the center-point for the search should be the GPS coordinates of the device or an entered address
- 1.1.5** Allow an address to be entered to be used (to be consumed if the View implementing Feature 1.1.4 indicates an address will be the source of information)
- 1.1.6** [NEW] The user should be able to initiate the search query via the interface.