CSC 191 – Advanced Android Development
Assignment 2 Answers
Mapping GUI Components to Functional Requirements

1.1.1 Allow users to select the work order priority (High[default] Medium, Low).

A RadioButton group or spinner is probably the best choice here, under the assumption that the user can only pick one priority to view. Both controls provide the ability to select 1-of-N and to set a default value to be pre-selected. The spinner is better for saving overall screen space.

If more than one priority should be allowed to be shown at the same time, then a checkbox setup would be more appropriate.

1.1.2 Allow users to select the work order type (Types TBD).

This requirement is very much like 1.1.1. A RadioButton group or spinner is probably the best choice here, under the assumption that a user can only pick one type to view. Both provide the ability to select 1-of-N and to set a default value to be pre-selected. The spinner is better for saving overall screen space.

If more than one type should be allowed to be shown at the same time, then a checkbox setup would be more appropriate.

(A TabHost could also work, with the Tabs containing all work orders of each different type.)

1.1.3 Allow users to select the geographical radius for work orders (default 1/2 mile).

It seems like the number of radius choices here may be fairly large (more than 2 or 3 options). There also has to be a default setting. Appropriate choices could be: a Spinner, set by default to 0.5 and with the ability to choose other settings based on the Spinner drop down box; an EditText, if testing was done to make sure only valid numbers were entered; and a SeekBar (a slider) combined with a TextView to show the current value of the SeekBar.

1.1.4 Display map with markers representing open work orders of the specified type.

A MapView is the best choice here, as it provides direct access to a GoogleMap. It supports Overlays which could be used to mark the location of the work orders.

(One might be able to use a WebView, which could be presented with the web content in the view set to an appropriate GoogleMaps page – using this approach, getting the overlays (pins) set in the WebView would require significant interaction with the GoogleMaps server)

1.1.5 Display list of work orders of specified type.

A ListView is likely the best choice here, as it provides the ability to represent a list, where each list item is textual information. It should support scrolling.

1.1.6 Allow users to select whether to show work orders as list or on a map.

A number of options should work here: a ViewFlipper, a TabHost, RadioButton with two choices, or two Buttons. The ViewFlipper would allow a gesture (like a swipe) to move to a different screen (though the Google Map ,

which also responds to swipes, could interfere with this). A RadioButton could cause the screen to switch when its state changed. The two Buttons would cause a screen switch so the screen relating to the last Button pressed is shown. Finally a TabHost would allow the ability to switch between tabs by touching the desired tab.

1.1.7 Given selection of a particular work order, display the details of the work order.

A TextView (or multiple TextViews laid out appropriately) is the best choice here, as the details are likely mostly textual in information.

1.1.8 Allow the user to select the center point for the search radius by entering an address.

An EditText control allows the user to physically provide text input to the system, using a physical or virtual keyboard.