4.1 Describe the Fragment lifecycle

Attach the fragment to an activity.

Initial creation of the fragment.

Bring a fragment up to a resumed state

Initial creation of fragment’s activity

makes the fragment visible to the user

makes the fragment interact with the user

makes the fragment no longer interacting with user

makes the fragment no longer visible to user

Destroy the view of fragment

final cleanup of the fragment’s state

makes the fragment no longer attached to its activity

4.2 List the set of Fragment callback methods used by a Fragment.

|  |  |
| --- | --- |
| Same callback methods as an Activity | Additional callbacks |
| onCreate() | onCreateView() |
| onStart() | onInflate() |
| onResume() | onActivityCreated(); |
| onPause() | onAttach() |
| onSaveInstanceState | onDestroyView() |
| onStop() | onDetach() |
| onDestroy() |  |

4.3 Explain when onInflate() and onActivityCreated() are called.

onInflate(): Called every time the fragment is inflated

onActivityCreated(): Called when the fragment’s activity has been created and this fragment’s view hierarchy is instantiated. Final initializations, such as restoring a state or retrieving a view, are often implemented here.

4.4 Describe the features of a typical action bar.

Application Icon: The application icon can be used as a unique identification for an application. To the right of the icon is the View control, the title element that can specify the application name or the activity the user is currently using.

Action Items: Action items are buttons for the most used action of the application. To provide quick and easy access, it is also typical to include action buttons for the important actions that require accessibility and prominence. If all of the specified action buttons cannot be displayed on the screen, those that do not fit will automatically be moved to the action overflow menu.

Action Overflow: The action overflow menu is a drop-down menu list that is often used for actions performed less frequently. This is also the location of actions that do not fit on the main action bar.

4.7 How does adaptive design differ from responsive design?

Responsive design is used for heavy data-driven content. It uses screen grouping techniques and complex navigation to present data content in a more intuitive and device-sensitive way. Adaptive design is primarily used for the rearrangement of fixed user interface elements in an application

4.8 Describe the master/detail design pattern.

In this design pattern, the user is provided with a list of items, which is referred to as the master list. Upon selecting one of the items, additional information relating to that item is then presented to the user within an area of the screen called a details panel.

4.9 How are Adapters used in a master/detail design?

The list items are inserted into the master list using an Adapter that collects the data content from a source, such as an array or database query, and converts each item result into a view that is placed into the list. The Adapter retrieves the data, then turns it into a view, and then adds to the details panel.