

Implementation Guide

Telstra News Feed Application

(Android Version)

Infosys Limited

Implementation Guide

Change History

Version	Date	Revised By	Description of Change
0.1	20/01/2015	Balajee Raghendra	Initial Version

Reference Document

Document	Document Name	Version
Requirement Document	Androidproficiencyexercisev2.0.pdf	V2.0

Implementation Guide

Contents

Telstra News Feed Application	1
(Android Version)	1
Infosys Limited	1
Change History	2
Reference Document	2
Application Specification	4
Guidelines	4
Application Components	5
Application Screens.....	6
Application Design	9
Test Plan.....	11
Open Source Code/ Examples from Internet.....	12

Application Specification

Create an Android app which:

1. Ingests a json feed from **<https://dl.dropboxusercontent.com/u/746330/facts.json>**.
2. Displays the content in a ListView
 - The title in the ActionBar should be updated from the json data.
 - Each row should be dynamically sized to display its content, no clipping, no extraneous white-space etc.
3. Loads the images lazily.
4. Allows the data/view to be refreshed, via either:
 - A refresh button
 - Pull down to refresh

Guidelines

1. Use Git to manage the source code. A clear history is required.
2. The app should target Android version 4.0. Don't worry about backwards compatibility for this task.
3. The list should scroll smoothly. As much work as possible should be cached.
4. Feel free to use any open-source libraries/examples you need, just be sure to give credit.
5. Comment your code where necessary.
6. Polish your code as much as possible.

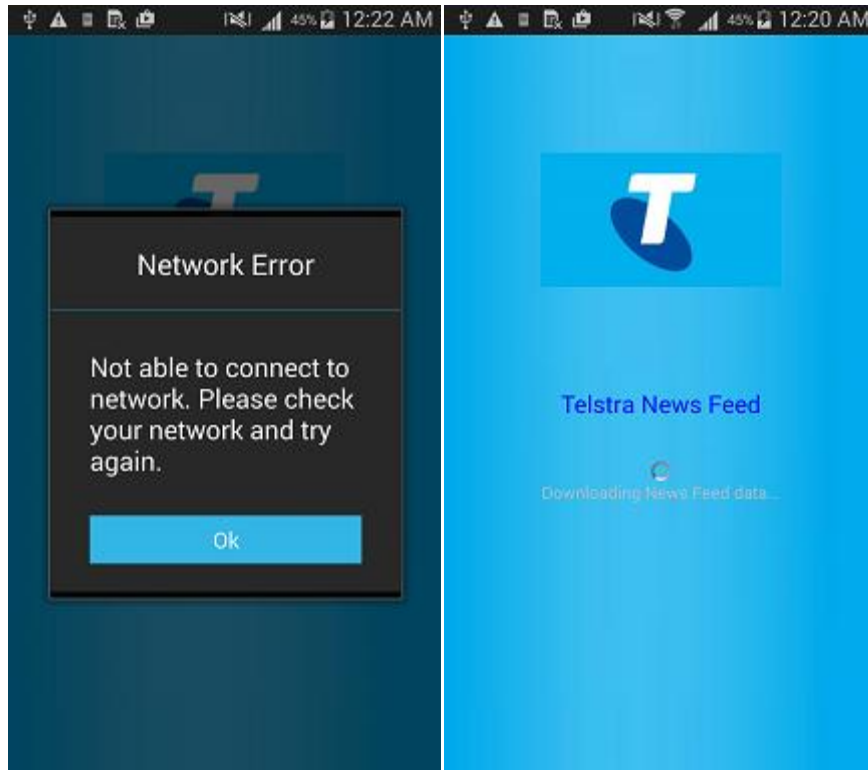
Implementation Guide

Application Components

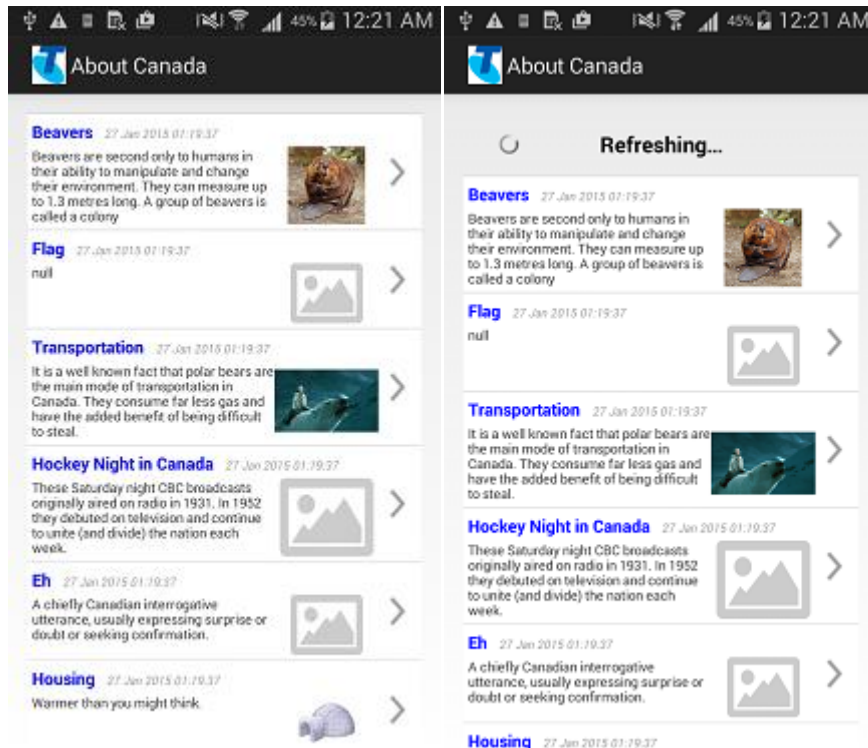
The various components making up the application are explained in below table.

Module Name	Description
Application Controller	It is the main part of the application. Volley Framework library components are initialized here.
Volley Framework[Network ImageView]	NetworkImageView from the volley framework is used to download and cache images asynchronously
JSON Parser	JSON Parser has been used to get json string from the network location and also parse it into POJOs which are used in the application
News Data Pojo	Container for the data downloaded from the network location
PulltoReferesh Listview	This is the component used to present the data on the screen. It has a built in capability for pull to referesh
Splash Screen, Main Activity	Splash screen is the first screen to be shown when the application starts, An async task is stated on this activity to fetch data from the network location. Upon completion of the task the Main activity[NewsFeedActivity] is launched.
AlertDialogFragment	This Fragment is used to show Dialog for network errors and other informations.
NetworkUtil	To check device connection state
SQLite Databasehelper	To create tables and store feed data
Exception Handling	To gracefully handle managed and unmanaged exceptions.

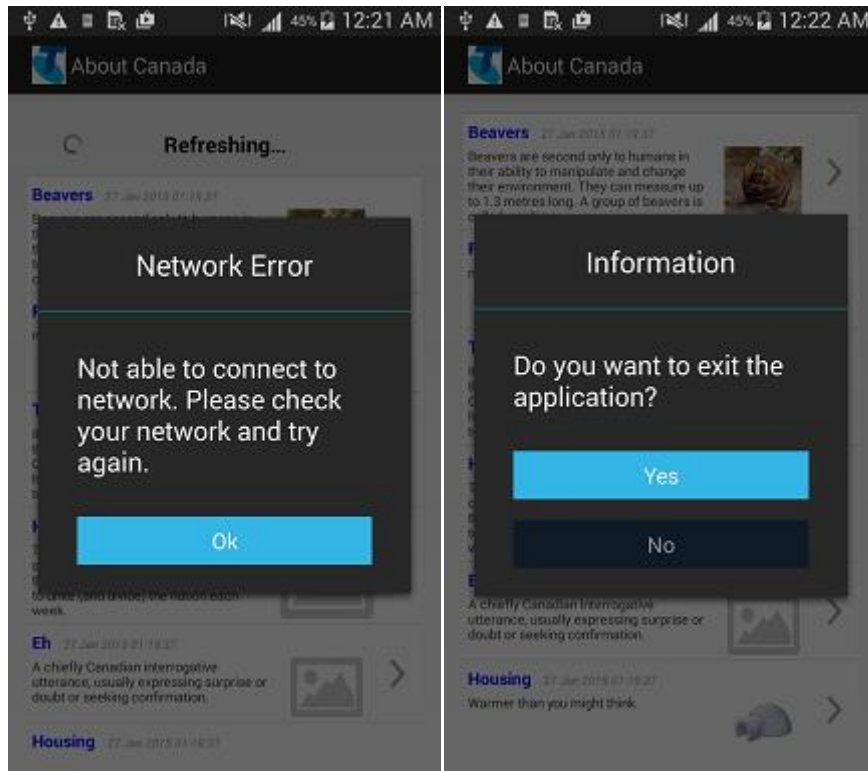
Application Screens



Implementation Guide



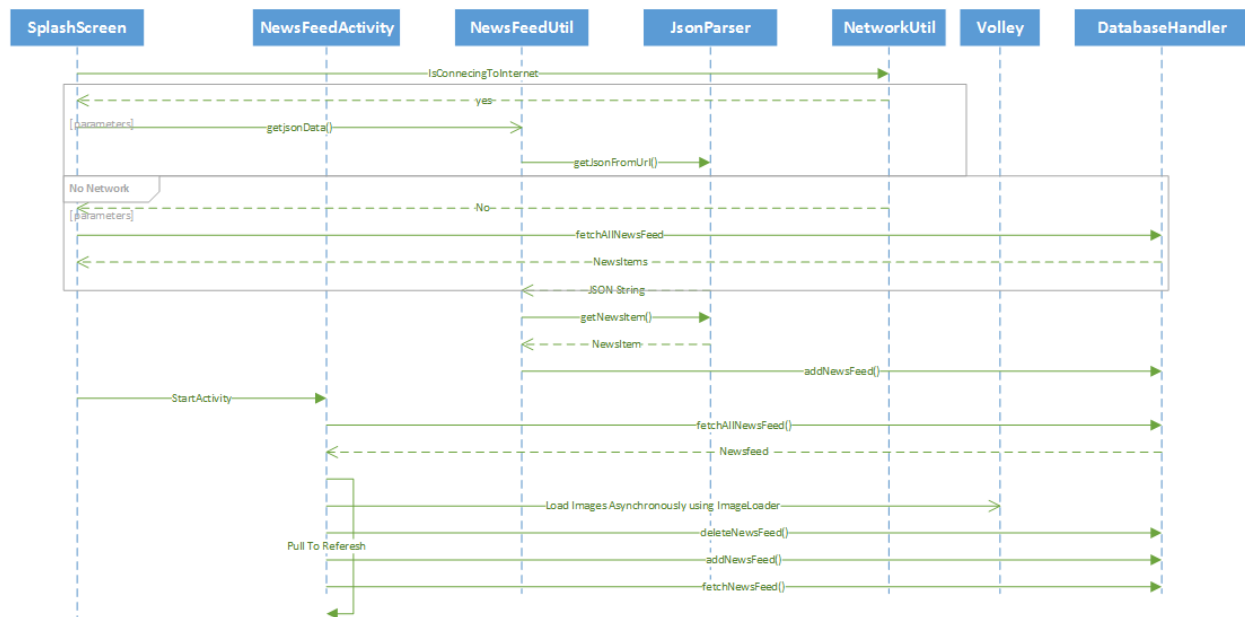
Implementation Guide



Implementation Guide

Application Design

The Application design can be understood with the help of the below diagram



Implementation Guide

S.No.	Application Component	Use Case Mapping	Rationale of use
1.	Splash Screen	To show an application start screen to the user and download the backend data in the background. [Though this use case is not mentioned explicitly, but it is a good idea.	To keep the app responsive.
2.	SQLITE database along with database handler	To make the app usable under no-network condition. [Again not mentioned explicitly but a nice to have feature].	SQLite allowed us to cache the feed data in the database and load it onto the listview under no-network condition. For it to work, the data must be downloaded atleast once. Also, the url of the images was hashed to create a primary key. "SHA-256" incoding was used.
3.	Volley NetworkImageView	Images should load lazily, so that listview can be easily scrollable.	Volley controls provides this functionality in its image control. It is free to use.
4.	PulltoReferesh Listview	Application should support pull to referesh for data updates.	It is a re-used component taken from github.
5.	Exception handling framework	Application should be able to handle exceptions gracefully.	

Implementation Guide

Test Plan

S.No.	Test Case	Result
1.	Application should show splash screen on start	Pass
2.	Application should inform the user under No Network Condition	Pass
3.	Application should download data in the background while the splash screen is shown	Pass
4.	Application should download the image asynchronously and should be able to cache it.	Pass
5.	Application should Sync the listview from the network datasource upon pull to referesh	Pass
6.	In case of issues during pull to referesh, application should show proper error message	Pass
7.	List view rows should expand appropriately as per the data	Pass
8.	Application should update Title Bar text from the Json data	Pass
9.	Application should show a confirmation dialog to User before exiting	Pass
10.	Application should easily be toggled between background and foreground	Pass
11.	Application should not loose state in case the device is locked on the app.	Pass
12.	Application should close gracefully in case of any uncaught exceptions.	Pass
13.	In case of network failure, application should load already saved data and should give a prompt appropriately.	Pass

Open Source Code/ Examples from Internet

Following sites were referred for guidelines and help for creating the application

1. Splash Screen + Async Task
<http://www.androidhive.info/2013/07/how-to-implement-android-splash-screen-2/>
2. Volley Framework + Image Loader
<http://www.androidhive.info/2014/07/android-custom-listview-with-image-and-text-using-volley/>
3. JSON Parser
<http://www.androidhive.info/>
4. Pulltorefersh ListView
<https://github.com/erikwt/PullToRefresh-ListView>