Project Report on News App for Android

Department: Computer Science Engineering

Group Members:

Name	Roll Number
Debjoy Bhowal	13
Subhrajit Bhowmick	14
Sampurna Biswas	15
Mainak Bose	16

Index

INTRODUCTION

The project aims to make an application that will display latest news headlines. This application provides several features such as saving articles for future reading, search functionality to search for news articles about any specific topic, location specific headlines and an in app view for displaying the news articles in its entirety.

TECHNOLOGIES INVOLVED

Technologies that are involved are as follows:

Android SDK 5.0 (API level 21) : Several features which were introduced in API Level 21 were used during the development of this app. hence the minimum level required to run this app was determined to be android 5.0 Lollipop.

Google News API: A free version of the Google news API was used. A free API key was generated for the purpose of use in our app. The API provides http links where get requests need to be performed for retrieval of data. The data is received in the form of a JSON. The API provides several options for the http link such as location and specific search queries. All these options were used to implement the features of this app.

JAVA 8: JDK version 8 was used while developing this application.

Volley: Volley is a high level API for handling HTTP requests. We used the getJsonObject() and getJsonArray() method for handling the response received from the news API. Volley operates with the help of a request queue where the exact details of HTTP request are stored and they get executed based on the FIFO nature of the queue data structure itself. Each HTTP request is expected to have 2 methods, one for handling the successful execution of the given http request and the other for any kind of error while

executing the HTTP request. The onResponse() method stores the received data on to realm directly based on if the request is a custom search query or not.

Picasso: Picasso is a high level API for handling image downloads, image cropping, fitting and inflating image view widgets. The headlines list displays a relevant image of the news article. The URL for image was given by the news API and Picasso was used to download the image from the URL, scale the image, fit the image appropriately and inflate the relevant image view in the headlines view.

Realm: Realm is a database management system API for android. it is a NOSQL database which can behave as both relational and non-relational. It is strictly based on OOP concepts. We created a news POJO (Plain Old Java Object) for storing the news we fetched from the news API. Since we are only receiving 10 results for every HTTP get request, we maintain an archive of news articles which we have fetched over time.

Android Widgets used:

- Recycler View
- Image View
- Web View
- Drawer Layout
- Frame Layout
- Progress Bar
- Swipe Refresh Layout
- Navigation View
- Card View
- constraint Layout

METHODOLOGIES INVOLVED

SYSTEM DESIGN

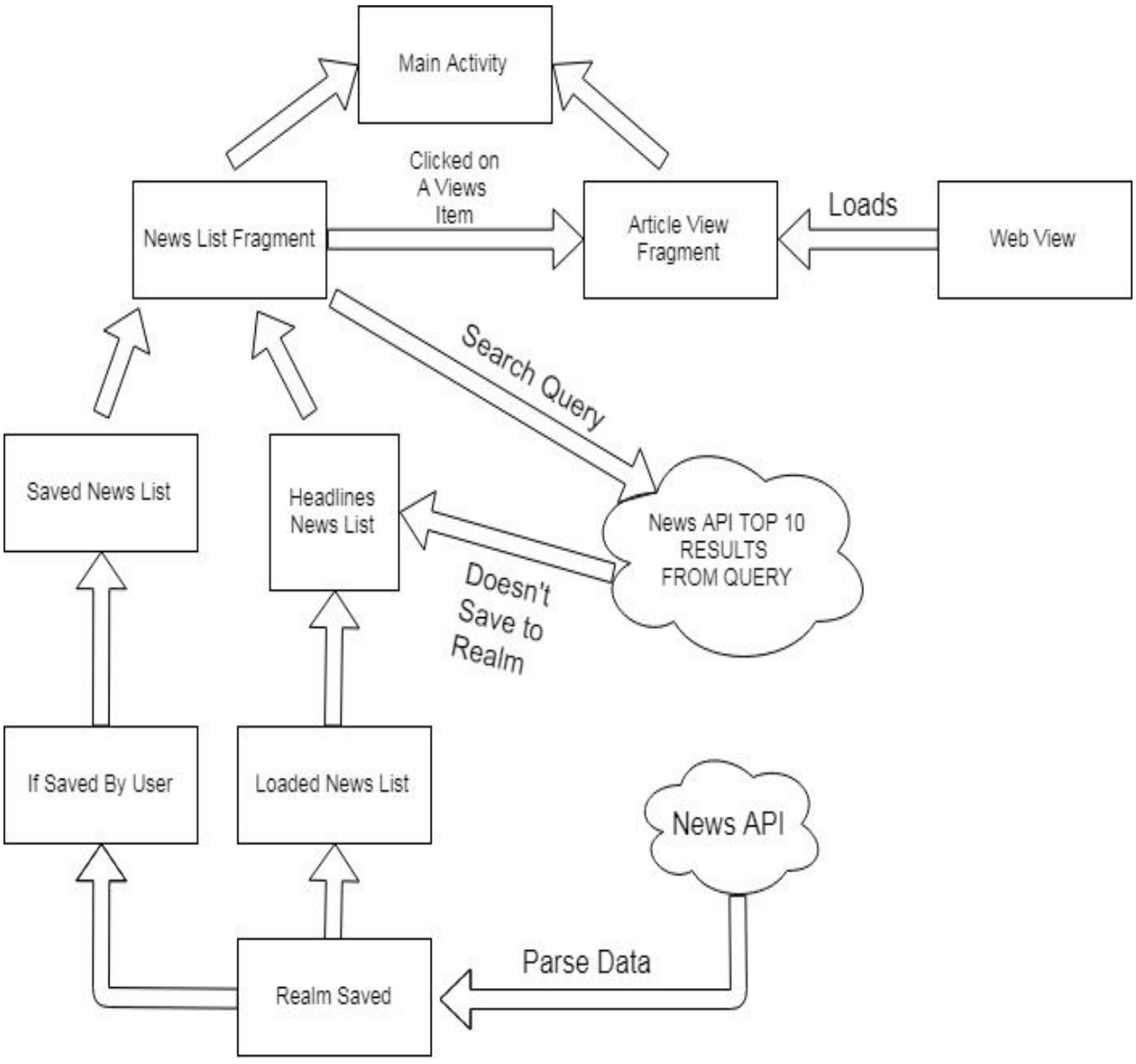

Figure 1: Flow Diagram

Description:

The news app is designed following the material design guidelines as much as possible to ensure a familiar interface for users.

The user is greeted with a welcome splash screen with a logo of the news app when opened followed by the home page. The home page contains the news headlines based on the location selected by the user from the top right button present in the action bar. There is only one main activity inside which fragment for showing news list and the article is swapped as needed. There is only one fragment (news fragment) for showing the list of top headlines and list of saved items.

The list of top headlines fetched from News API and is stored in realm first and then retrieved from realm to show in the fragment with the help of an adapter and RecylerView. There is a boolean flag “saved” for every news item saved in realm to check, if it is marked as saved by the user or not.

When user navigates to saved list, news items are retrieved from realm where the flag “saved” is true thus, retrieving only those news items which were marked saved by the user.

In case of the headlines, an API call is made to news-api.org to get the top headlines with respect to user selected location. These results are stored in Realm and then retrieved again where location is equal to the selected location. The retrieved results are shown in the headlines fragment.

The images which are to be loaded per news article as thumbnails is fetched asynchronously with the help of a library named “Picasso”.

For search, the search query is added in the API call and location is set to global. Unlike above, the results are not stored in the realm for now and are

shown to the user directly. Only if the user saves the article then only the news item is saved in realm.

When a news item is clicked, the news fragment gets replaced with the existing fragment. After the fragment is replaced the news article is loaded with the help of a Webview. When this fragment is active the “+” is added to the top right corner of the action bar for saving the news article, which disappears when pressed back.

When back button is pressed and the user is in home screen a toast is shown saying “press back again to exit”. The app closes when pressed again. An exit flag is maintained across the whole application for handling the above functionality.

As the user is using the application for a longer period of time, the size of the app tends to grow more. To counteract that, an option is given in the navigation drawer to clear all data of the app, which was done by just deleting the data folder of the app programmatically.

For showing the loading animation, first a Progressbar view is made full screen, and when the call back for the response is received from different Asynchronous calls, the visibility is set to View.GONE.

When the button for changing the location is selected from the action bar, an alert dialog opens with different locations to select from, if selected by user, the function for loading the headlines is called making the homescreen load again. Once a user selects a location, the name of the location is saved in the phone’s memory using SharedPreferences such that if the user opens the app again he/she will see news from the location he last selected.

Implementation:

Figure 2: Openning Splash Screen with logo for the news app

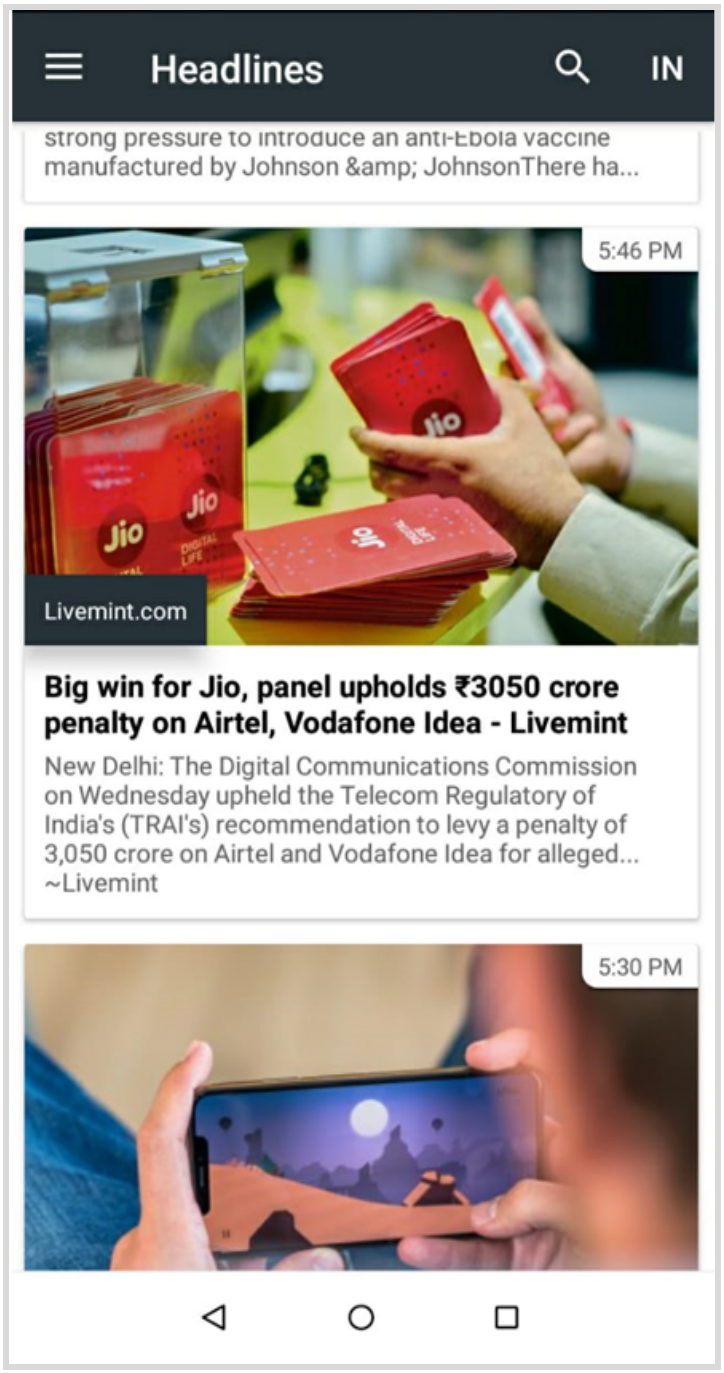

Figure 3: Headlines view with article preview, time of publish of article and relevant image

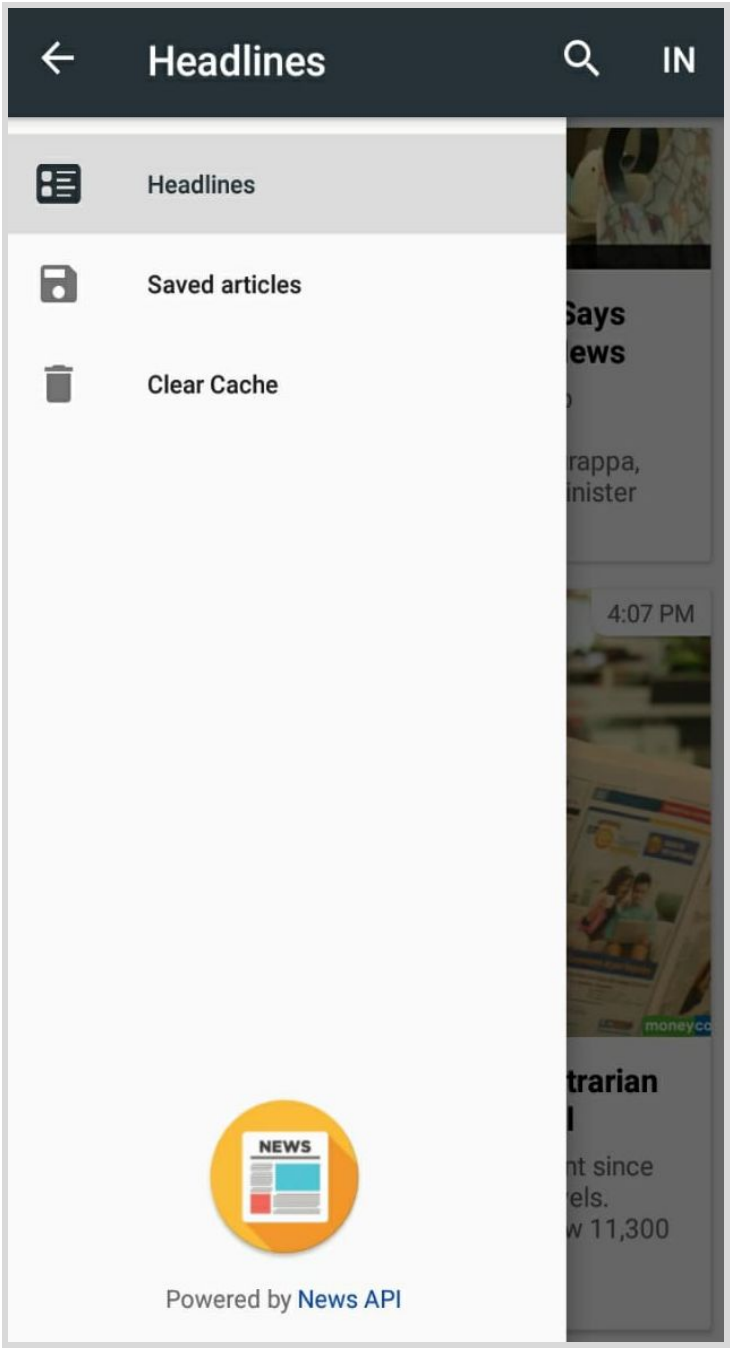

Figure 4: Side Navigation Bar

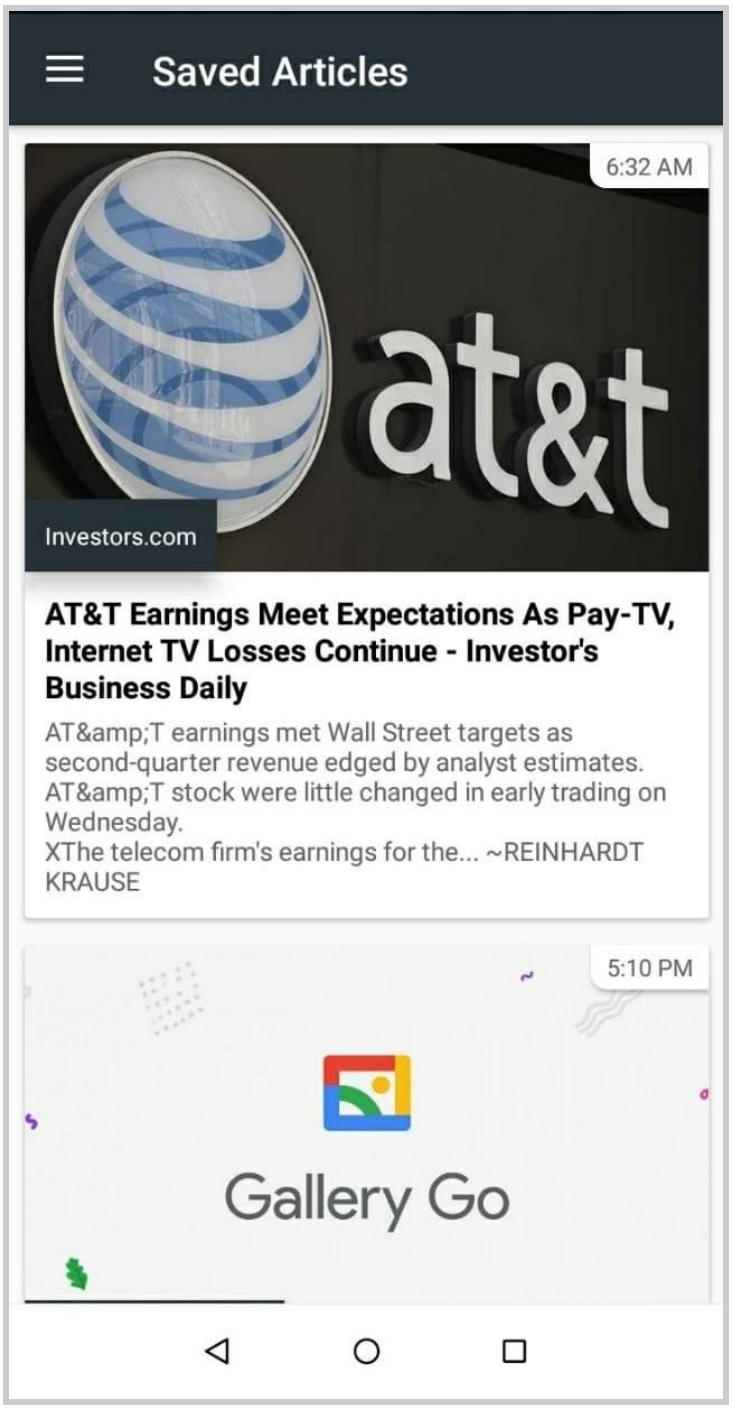

Figure 5: Saved Articles containing the list articles saved by the user

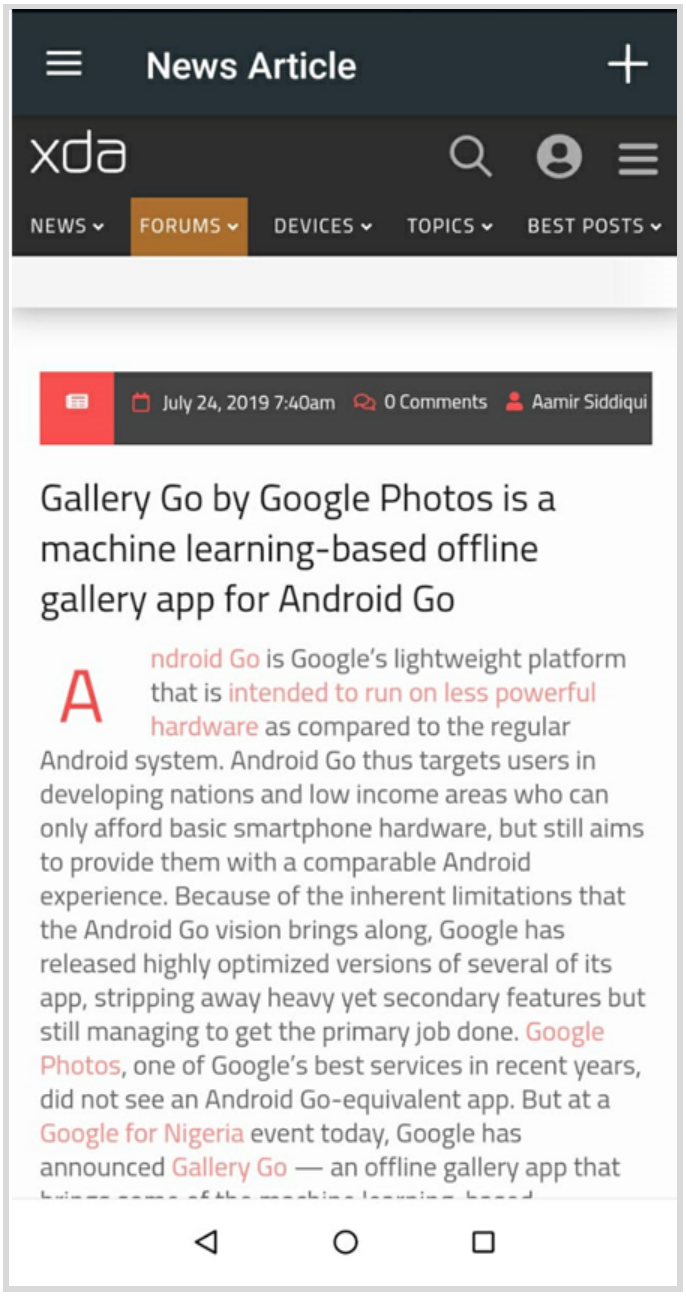

Figure 6: A News Article being displayed with the plus sign in the top right being used to save the article

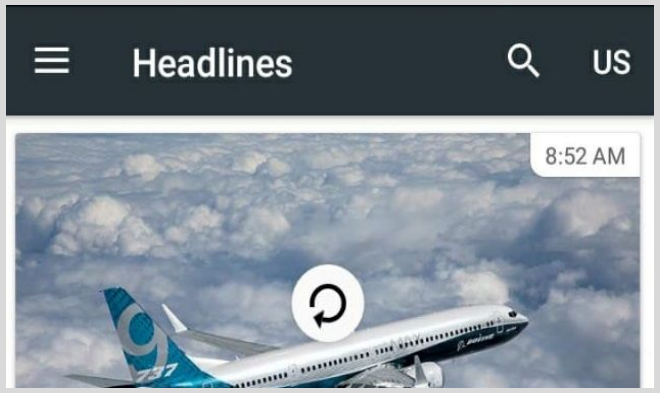

Figure 7: Swipe down to refresh

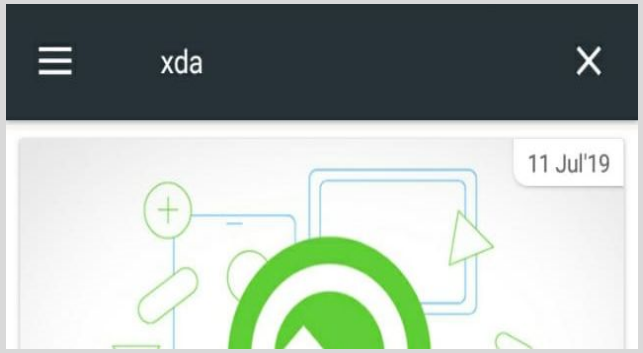

Figure 8: Search bar being used to search for articles related to xda

Figure 10 : Loading Animation

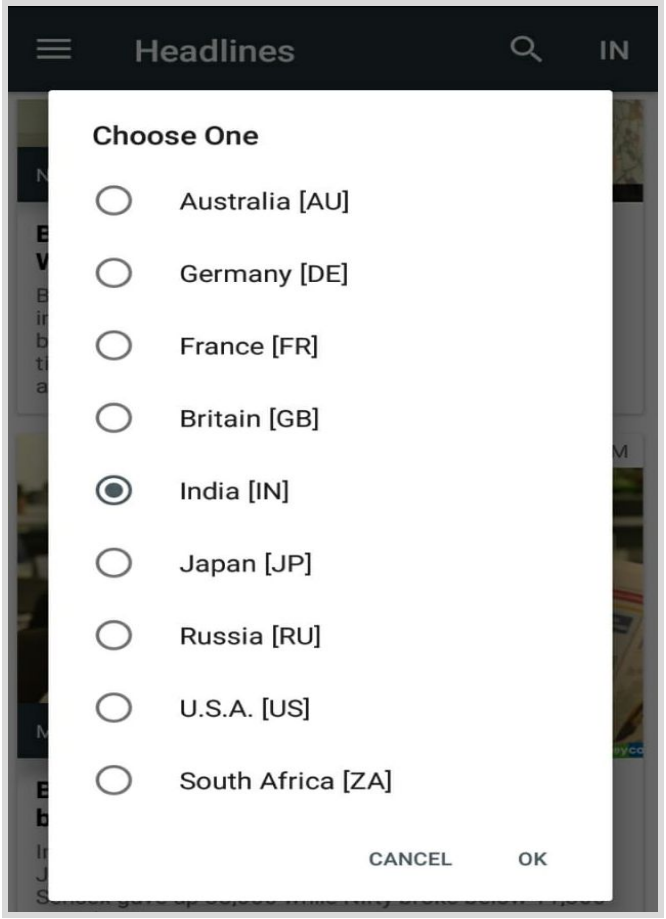

Figure 9: Location Picker

Figure 11: No articles saved prompt

Individual Contribution:

Debjoy Bhowal(13) : Worked on UI design, added loading animation in different places of the application, added a splash welcome screen when the application is opened, added a search option in actionbar and made it work, added option to change location, addition and working of an option to clear all data. Added credits to “News-api.org”.

Shubhrajit Bhowmik(14): Handled the back press for the application, avoided extra addition to backstack and the double press of back button to exit from the application by maintaining an exit flag throughout the application.

Sampurna Biswas(15): Worked on the Saved list page. Added a message as text if there is nothing saved. Worked with the existing Pojo class to add attributes and get the saved item by the user. Used the add symbol in the action bar for saving and deleting.

Mainak Bose (16): Design of the basic structure of the project. Fetching data from the API and parsing the data into recylerview, loading of the webpage, handled the fragment transactions. Addition and working for drawerlayout, working of the saved button on actionbar, asynchronously loaded the thumbnails using Picasso.

Conclusion:

The app is currently in a stable state without any bugs. We were able to fix several bugs such as sudden crashes while operating the app, UI bugs due to the mismanagement of the fragment transactions as well as long loading times while displaying an article. The app currently performs every action it is supposed to without any problems. There are several features which can be added in the future such as translation of every headline and news article to any desired language selected by the user, offline loading of saved articles, a more customized webview for displaying the news articles and optimizing internet data usage.