CAPSTONE

SPACE DIG

DILIP KUMAR S

17th June 2017

GitHub Username: dilipkumar4813

Space Dig

Description

The Android application will gather information from NASA gov and provide users information with respect to earth imagery for a specific date, astronomy picture of the day (APOD) images of mars taken using mars rover, search for images and videos of NASA, Techport which will showcase NASA funded development that the public can view and keep track of the development stage and show the close approach data of comets.

The application will make reading information about space easy, store data for offline viewing and get real time information for earthlings.

Intended User

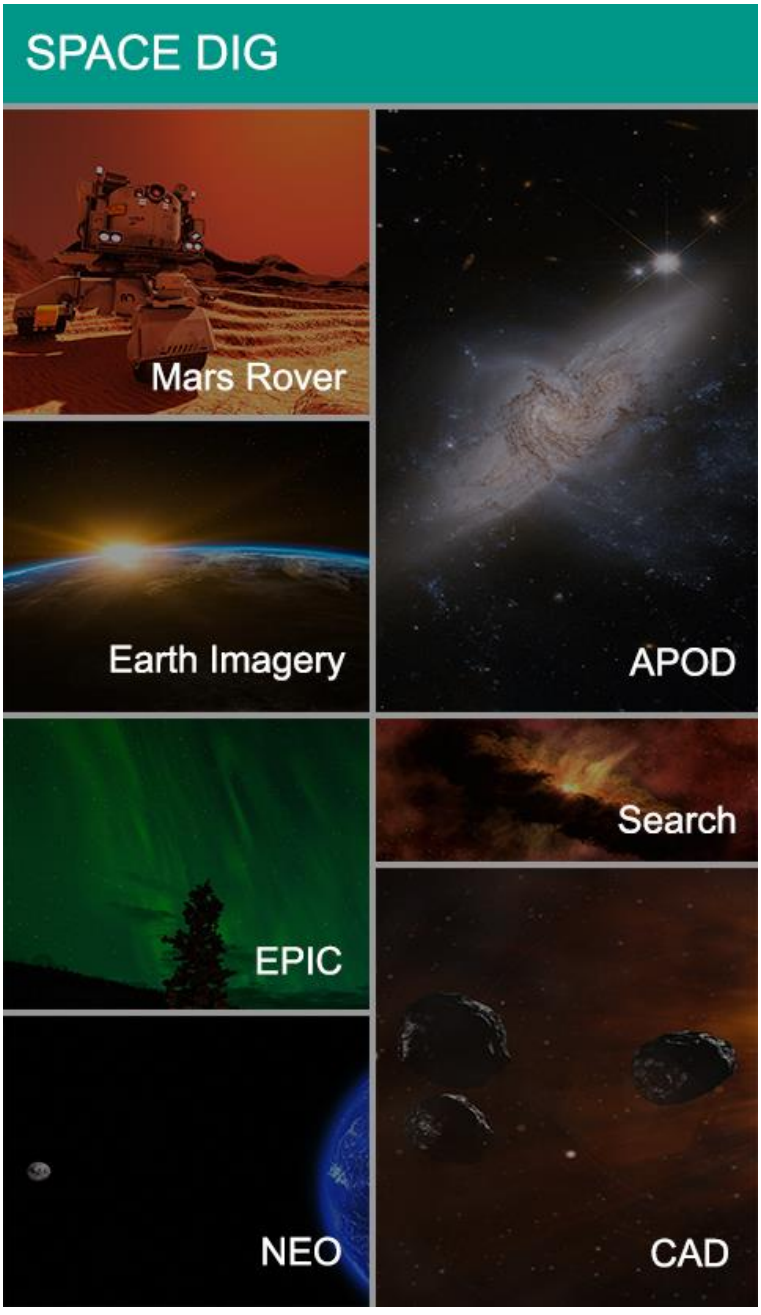
The Android application is intended for researchers, hobbyist and visionaries.

Features

1. Saves information for offline viewing.
2. Shows pictures of mars rover.
3. Search through the wide database of NASA by implementing search.
4. Exoplayer for streaming video.
5. Firebase cloud messaging to maintain user engagement.
6. Firebase analytics for tracking user events.
7. Google maps to fetch and display current location

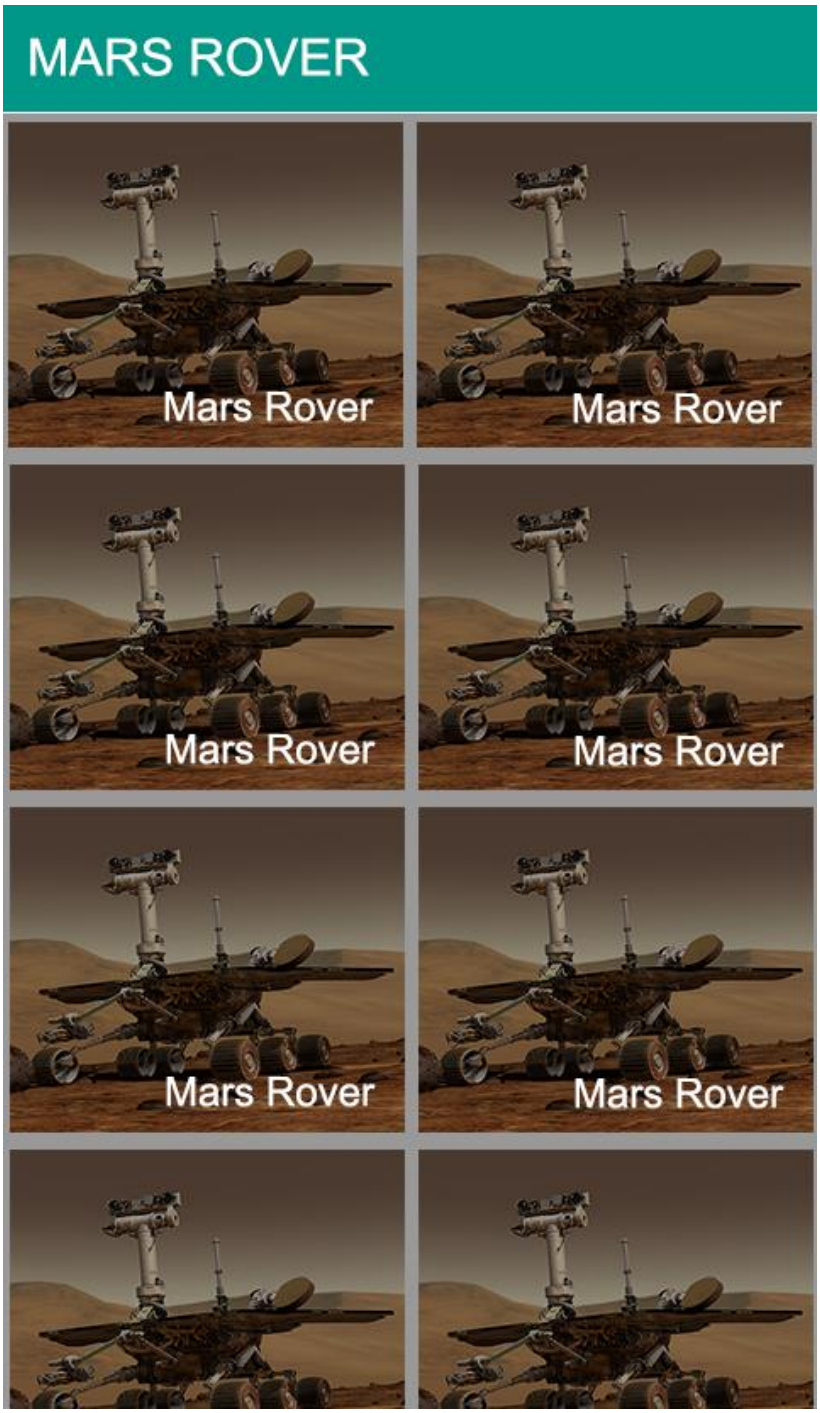
User Interface Mocks

Screen 1



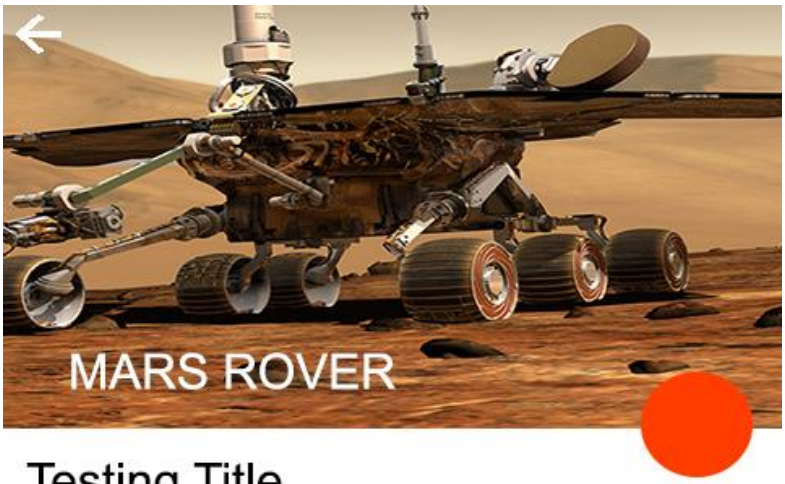
The main screen is a staggered layout which will provide access to the various options provided by the application.

Screen 2



The listing screen will be a grid view, which will contain the image for the item and its relevant data. The data might be just name or may contain other information such as earth date etc. Depending on the category selected the information will be displayed.

Screen 3

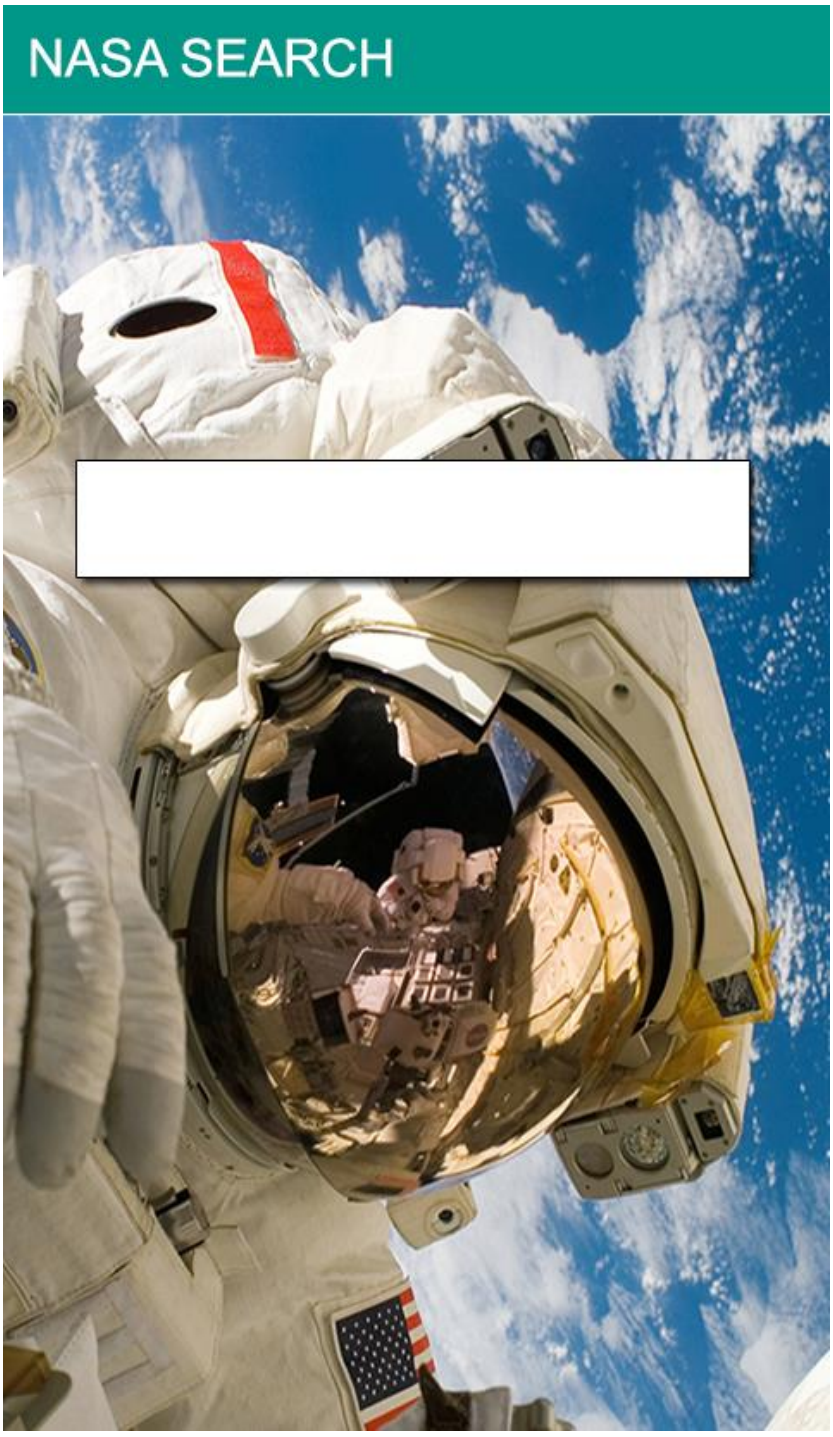


Testing Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante.

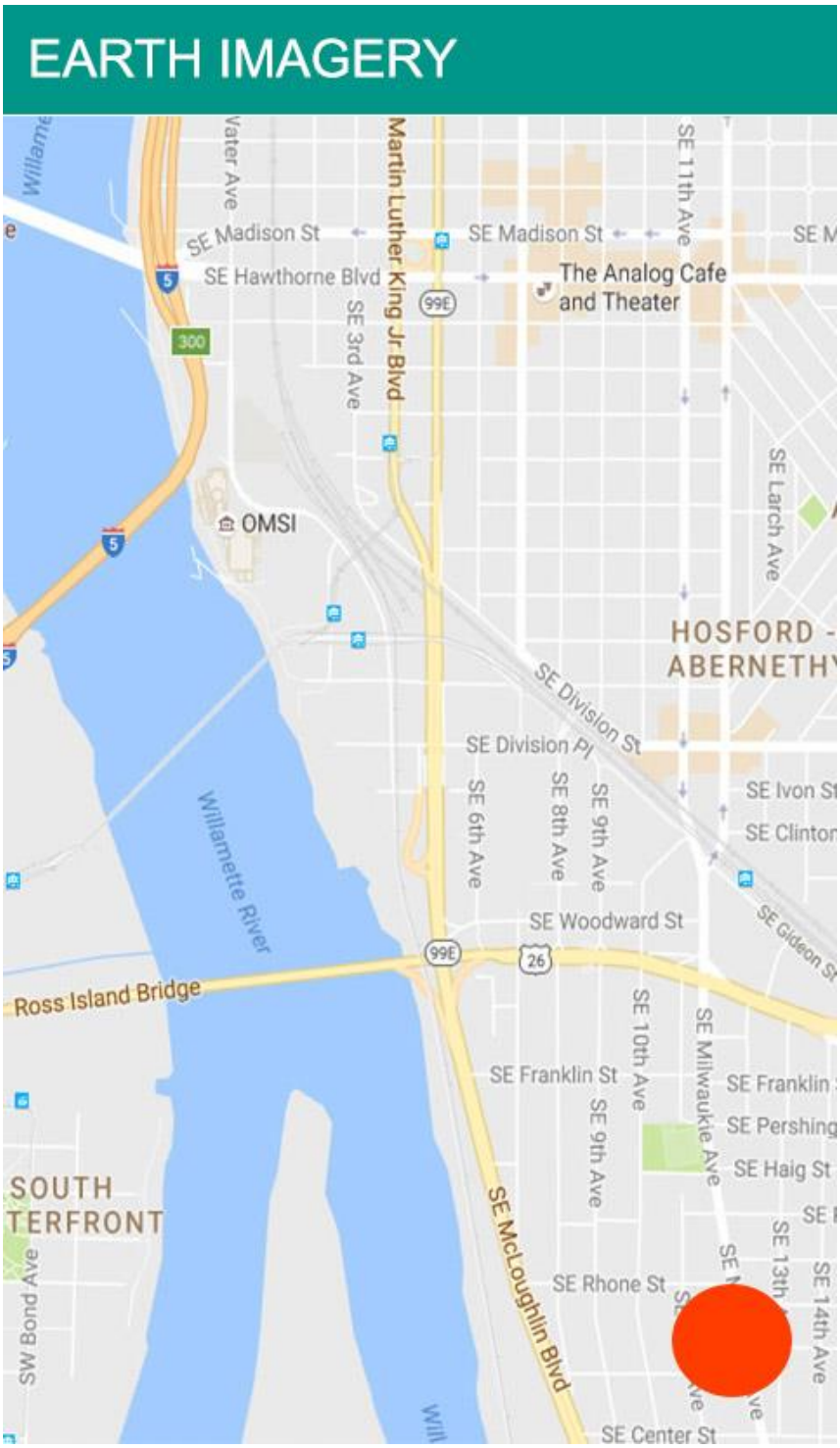
The details screen will load the information with respect to each item, If the screen contains a video then exoplayer will be included. If the user changes the orientation to landscape and video is present then the video player would become full screen. Share option is provided in the screen to allow them to post the data in any social networking platform.

Screen 4



The search will be used to search through the database and load the results in a grid view format as in screen 2. On clicking each item the details will be loaded into the next screen.

Screen 5



When the user wishes to get the earth imagery of the location, they can move the map to their location and click on the fab button to fetch the earth imagery for the current day.

Key Considerations

How will your app handle data persistence?

The application will handle data persistence by using Content provider and Shared preferences.

Describe any corner cases in the UX.

- While video streaming if the user has played the video, the time stamp is recorded and the user will be provided with a prompt to check if they would like to resume or start from the beginning.

Describe any libraries you'll be using and share your reasoning for including them.

- Glide for loading images from the web and caching.
- Retrofit 2 to handle API calls.
- RxJava and RxAndroid for handling threading.
- Schematic for Content provider.
- Butterknife for loading views.
- Exoplayer to facilitate streaming of videos.
- Google maps for loading maps
- Firebase analytics for logging user events
- Firebase cloud messaging to send notifications.

Describe how you will implement Google Play Services.

Maps using Google Play Services to fetch users current location and plot of the map.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Register with NASA and get API key
- Configure Gradle to include API key
- Create the folder structure for the project

Task 2: Implement UI for Each Activity and Fragment

- Build UI for main staggered view activity
- Build UI for Mars Rover, CAD, EPIC, NEO and Search
- Build UI for details page
- Build UI for Search main activity
- Build UI for Google map plotting
- Build UI for Earth Imagery
- Build UI for About the app

Task 3: Google Maps

- Create new project in Google console
- Enable Google Maps API
- Register Android app by providing SHA1 and package name for the project
- Add Google map functionality to the application
- Fetch current location
- Fetch location information based on movement of map

Task 4: Firebase

- Import project created in Google console to firebase
- Enable analytics and FCM
- Generate JSON file and include in the project
- Build the FCM module
- Add Firebase analytics log event for various actions

Task 5: Network Calls

- Build interface for the various API's involved in the project
- Create POJO for the API's
- Create an interceptor that will work only during debug version
- Connect API for APOD
- Connect API for mars rover module
- Connect API for CAD
- Connect API for search
- Connect API for Neo
- Connect API for earth imagery
- Connect API for EPIC

Task 6: Google Maps and Earth Imagery

- Pass Google map set location to Earth Imagery API
- Fetch the result and render the view

Task 7: Content Provider

- Create content provider for the entire application
- Insert data when an API call is made for offline viewing purpose
- Unless the API contains pagination delete and insert values for the remaining API's when network is available
- Reduce number of API calls

Task 8: Unit Test

- Build unit tests for the following modules
- APOD
- CAD
- NEO
- Earth Imagery
- Search
- Mars Rover
- EPIC

Task 9: Functional Test

- Build unit tests for the following modules
- APOD
- CAD
- NEO
- Earth Imagery
- Search
- Mars Rover
- EPIC

Task 10: Documentation

Document the entire project in Wiki pages

Task 11: Image Assets

- Create the image assets necessary for Google play store.

Task 12: Release

- Test run the application on a tablet and phone
- Run through Google play store checklist
- Release the application

Task 13: Promotion

- Promote the application
- Track user events
- Engage the users through notifications
- Have fun