
GitHub Username: vohsalab

Stargazer

Description

Since the mankind was born people were always looking at the stars. Nowadays you can raise your eyes to the sky and see International Space Station - satellite made by people of Earth. You just need to know when you should do that and this app will help you. Choose the date from provided list, walk outdoors and raise your eyes. Want to know where it is now? You can see its location on a map. Want more? Check the live video feed from station's cameras. And last but not least: NASA's "Astronomy picture of the day": breath-taking pictures plus captivating descriptions.

Intended User

This app is for astronomy and space exploration enthusiasts and other curious people

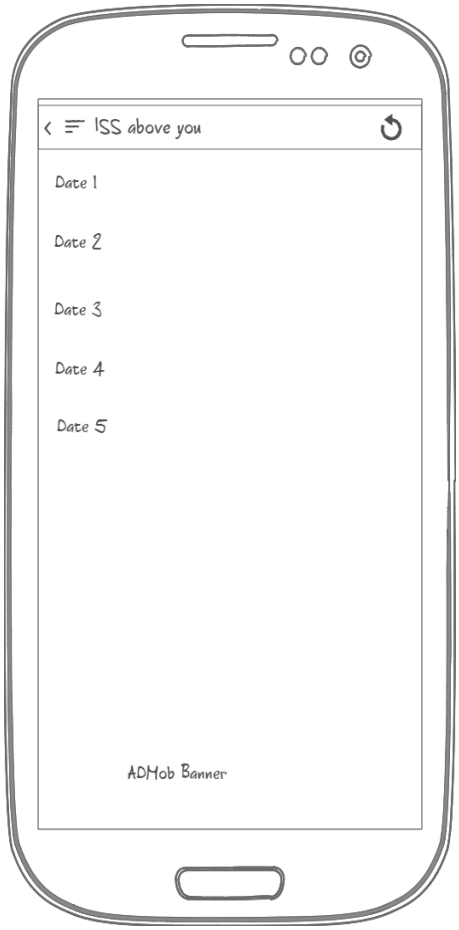
Features

- Shows date and time when you can see ISS in the skies at your location
- Shows current location of ISS on the map
- Provides live video feed from ISS cameras
- Shows NASA's "astronomy picture of the day" with it's description

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



This screen will provide the list of the five closest dates when ISS will fly over your location. Refresh button allows to refresh it based on current location

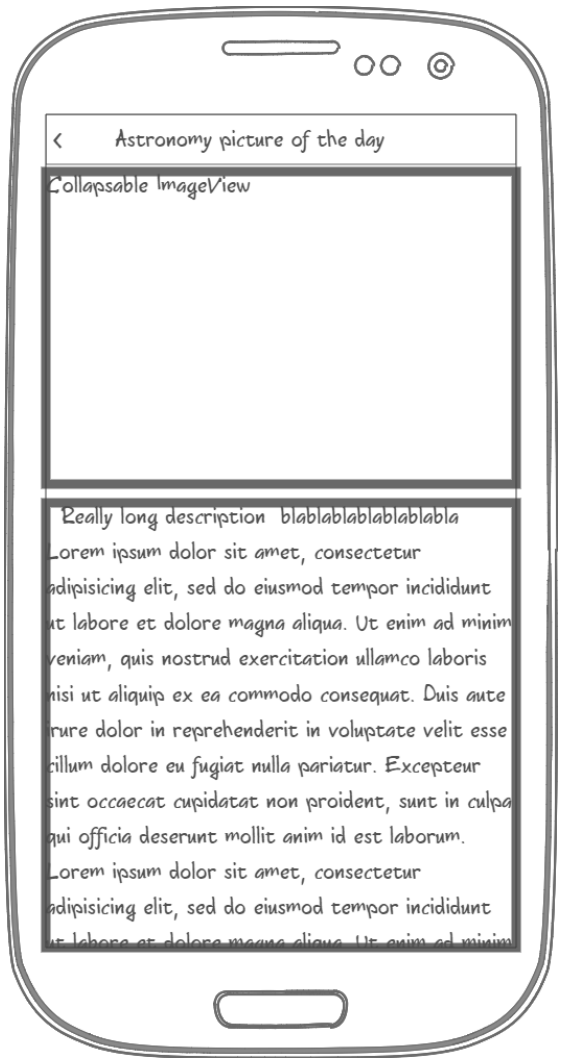
Screen 2

At the “Details” screen you can set up the 15-minute notification ar alarm reminder

Screen 3

On the “ISS Now” screen you can see ISS current location on the map and live videofeed from ISS cameras

Screen 4



“Astronomy picture of the day” allows user to see picture and information based on the NASA’s page

Key Considerations

How will your app handle data persistence?

ISS data will be loaded from external API and stored in local database

Describe any corner cases in the UX.

Navigation Drawer will be used for navigation between activities. Within the Astronomy Picture of the Day activity ImageView will be collapsable.

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

Retrofit for network calls, Picasso for loading images, Butterknife for reducing amount of boilerplate code, Ustream Player SDK for the videofeed

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Install required libraries
- Research required APIs

Task 2: Implement UI for Each Activity and Fragment

- Build UI for "ISS location" Activity
- Build UI for Location Details Activity
- Build UI for "ISS now" Activity
- Build UI for "Astronomy picture of the day" Activity

Task 3: Build ISS Location Activity

- Implement network call
- Implement content provider
- Populate list with available dates
- Add ADMob banner

Task 4: Build Location Details Activity

- Add notification setup. User can choose between notification and alarm

Task 5: Build ISS Now activity

- Implement MapView showing current ISS location with custom map marker
- Implement custom view with live video feed from ISS based on Ustream SDK

Task 5: Build Astronomy Picture of the Day activity

- Load picture and description from REST Service