

[Description](#)
[Intended User](#)
[Features](#)
[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Tasks](#)

[Task 5: Your Next Task](#)

GitHub Username: [agatagruza](#)

Note: below table is based on [support library release tracking](#) and [Picasso](#) and [Timber](#)

Project setup (app will use current and stable versions)	
Android Studio	3.2
Compile SDK	28
Min SDK	16
Gradle	4.4
Libraries	AppCompact 28.0.0
	Design 28.0.0
	ButterKnife 8.8.1
	ButterKnife Compiler 8.8.1
	Picasso 2.71828
	Timber 4.7.1

Discover the cosmos

Description

Discover the cosmos app is a tool that will help uncover the beauty of galaxy and equip user with knowledge needed to tackle looming challenges here on Earth. It's a compendium of daily magnificent pictures of the solar system together with descriptive details written by professional astronomer. User can create its own macrocosms gallery. Moreover, with this app user will be able to locate nearby NASA location and learn more about given photography.

Idea for an app is based on [Astronomy Picture of the Day \(APOD\)](#) as one of the most popular website at NASA. In reality, APOD website is one of the most favorite website across all federal agencies [link](#). Data for an app are pulled from the NASA IPI (<https://api.nasa.gov>).

Intended User

Target users will include students and everyone who want to broaden knowledge of the cosmos, astronomy researchers who invent and solve looming challenges of the world, photographers and people who look for an inspiration and ideas, travelers.

Features

Main features of your app:

- Saves favorite cosmos pictures and detail information of that picture
- Include widget for an easy access
- Finding nearby NASA location based on user input
- Java language will be used for development
- RTL support for all layouts
- Accessibility features
- App will use current and stable versions for all libraries
- All string will be kept in string.xml file

User Interface Mocks

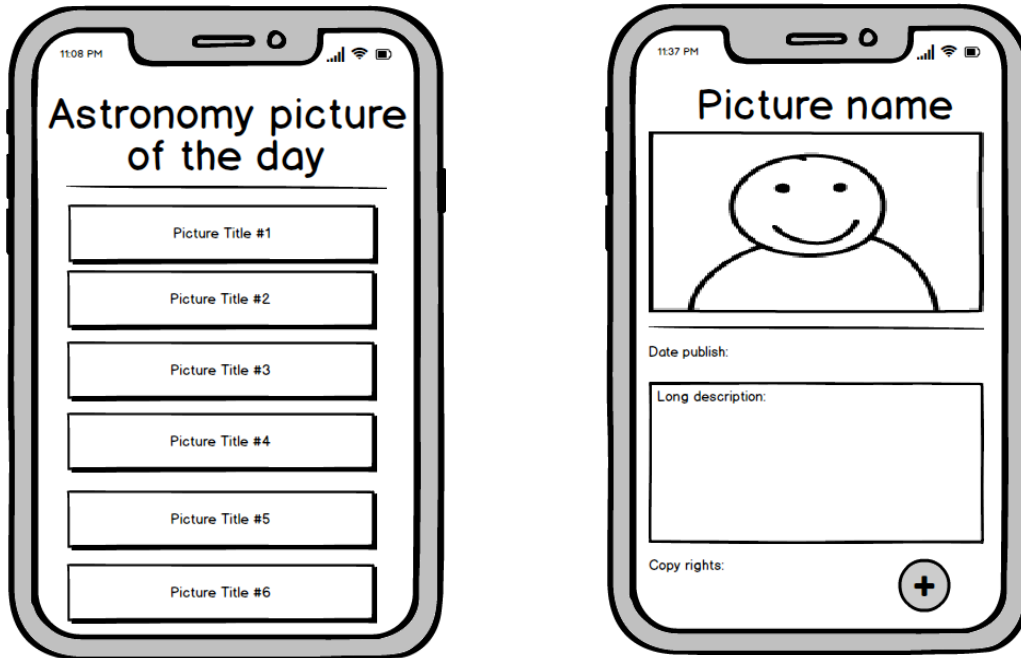
Screen 1 and 2



Screen 1 – Sign Up: When user open the app, Sign Up window will pop up. If she/he is a returning user, then he/she can click on “login here”. After clicking “login here”, Screen 2 will pop up.

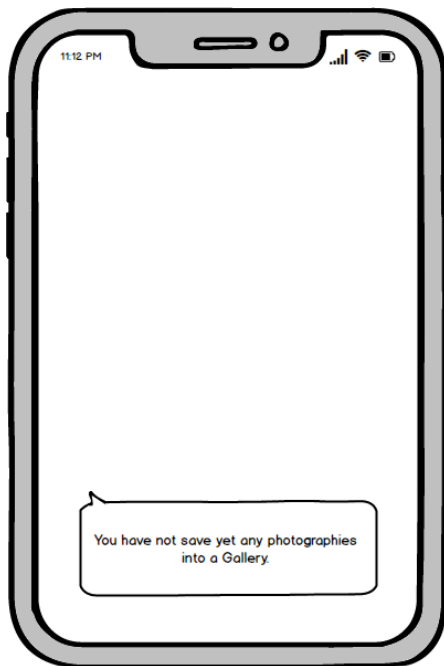
Screen 2 – Login Activity: If user already has an account, he/she can directly login.

Screen 3 and 4



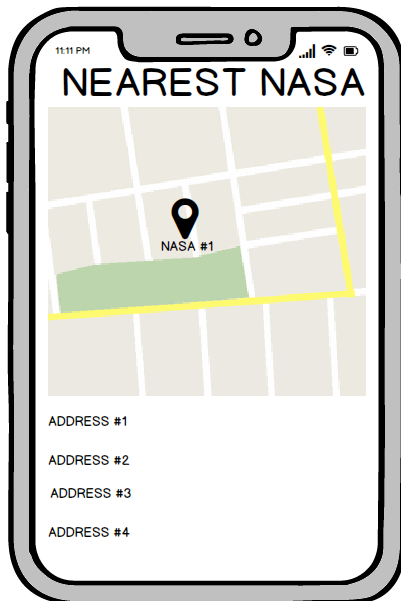
Screen 3 – Main Activity: After login in user will see list of most current astronomy pictures. After clicking on the picture, DetailActivity will show up (Screen 4) with detail information about chosen photography. User by clicking “+” will have ability to add photography to favorites.

Screen 5



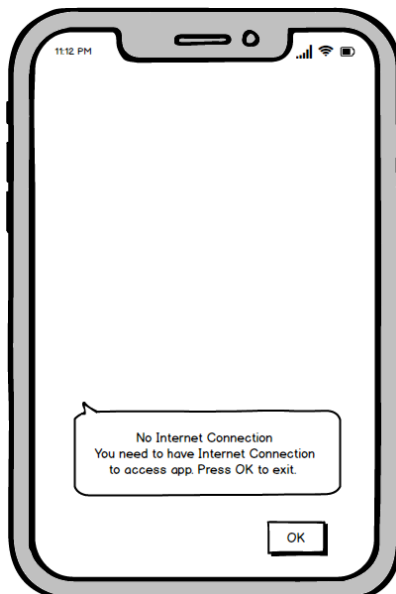
Screen 5 – Gallery: Initially there will be no photography in Gallery (favorite cosmos pictures and detail information about pictures).

Screen 6



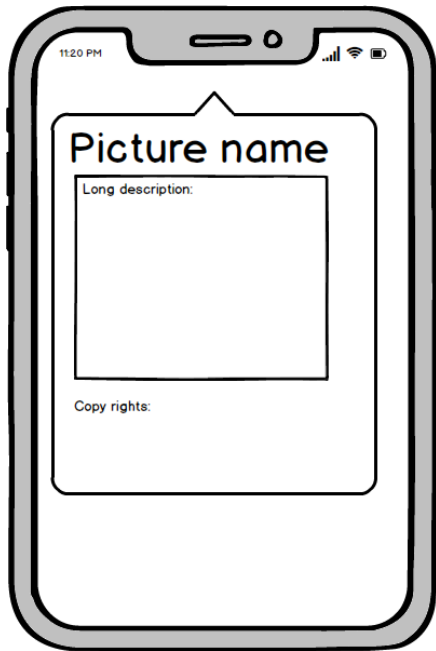
Screen 6 – NASA Location: based on user physical location, app will be able to navigate to nearest NASA location where user can learn more about given photography.

Screen 7



Screen 7 – No Internet: In case there is no Internet connection, app will not crash. Instead it will show up Toast message.

Screen 8



Screen 8 – WIDGET: User can add last viewed photography to the widget. From the widget user will see detailed information about lastly seen photography.

Key Considerations

How will your app handle data persistence?

Data persistence will be handled in the app by building Content Provider and connecting to SQLite database.

Describe any edge or corner cases in the UX.

Internet connection is needed to retrieve data from NASA API. In a situation when there is no network connection, the app will not crash. Instead, it will show an appropriate message informing the user of the situation.

There is a limit of 1,000 requests per hour per given API key (NASA restriction). Exceeding these limits will result in temporarily blocking the API key from making further requests. It will be lifted after one hour.

Input from the server and user will be validated. If there is no data or if there is a wrong format, the app will not crash. Instead, it will log that incident.

By adding a back arrow on the toolbar, the user will be able to go back one tab at the time.

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

Discover the cosmos app will take advantage of

- ButterKnife lib for databinding. It makes code look cleaner and saves time to write repetitive lines of code
- Picasso lib for handling the loading and caching of images from NASA API into respective ViewHolders
- Timber lib for enhancing logging experience

Describe how you will implement Google Play Services or other external services.

There are three main Google Play Services that will be used in Discover the cosmos app.

- Location – will provide user current location
- Maps – will provide nearby NASA location based on user current location
- AdMob – from time to time small advertising banner will pop up to provide user with advertising experience
- (If time permits) Good to have Analytics on Firebase. Follow instructions from <https://firebase.google.com/docs/analytics/android/start/>

Next Steps: Required Tasks

Task 1: Project Setup

- Use Java as a solely developing language in Discovering the cosmos app
- Use one of the template in Android Studio to create project
 - As a MainActivity add a Bottom Navigation Activity
- Configure libraries
 - Add ButterKnife dependency for databinding
 - Add Picasso dependency for performing image loading
 - Add Timber dependency for gaining logging experience
- Add into manifest Internet, Network and Google Play Services
- Enable RTL layout switching in application structure
- Create string.xml file to keep all string values in one designated place

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for Gallery(favorite cosmos pictures and detail information about pictures)
- Build UI for finding nearest NASA location
- Add map placeholder to find nearest NASA location
- Add AppBar to Main Activity and corresponding navigation components

Task 3: Retrieve and load data from external API

- Design network utilities that will be used to build URLs and make network requests and transfer data
- Extend AsyncTask class to retrieving data from NASA IPI. The reason behind that is to offload long running task to the list of processes running in the background

Task 4: Use SQLite database for Data Persistence

- Enforce SQLite database Helper and Content Provider
 - Connect to your database
 - Implement database management functionality required to interact with saved gallery (favorite cosmos pictures and detail information about pictures)
- Enforce Loader class for adding data from SQLite into Views

Task 4: Implement Google Play Services

- Place AdMob banner into designated placeholder View
- Enforce Location functionality to get user current location
- Enforce Maps functionality into designated placeholder View. User will be able to find nearest NASA location based on user input Location

Task 5: Implement Accessibility Feature

- Follow <https://developer.android.com/guide/topics/ui/accessibility/apps> to support content description and D-pad navigation (touch points are at least 48dp, colors will be used with sufficient contrast)

Task 6: Implement Widget

- Create home screen widget that displays detailed information for lastly seen photography