Contents

GitHub Username: jordiguzman (https://github.com/jordiguzman)

AstronomyApp

Description

With this astronomy app users will be able to:

- View the APOD page (Astronomy Picture of the Day).
- View a Wikipedia description of all Solar System planets, the Sun and the main satellites.
- Track the ISS (International Space Station) and watch the Earth live via an ISS webcam.
- View the news posted on the Hubble Space Telescope site.
- View Earth pictures from the DSCOVR's Earth Polychromatic Imaging Camera (EPIC), located on the Deep Space Climate Observatory satellite.

Intended User

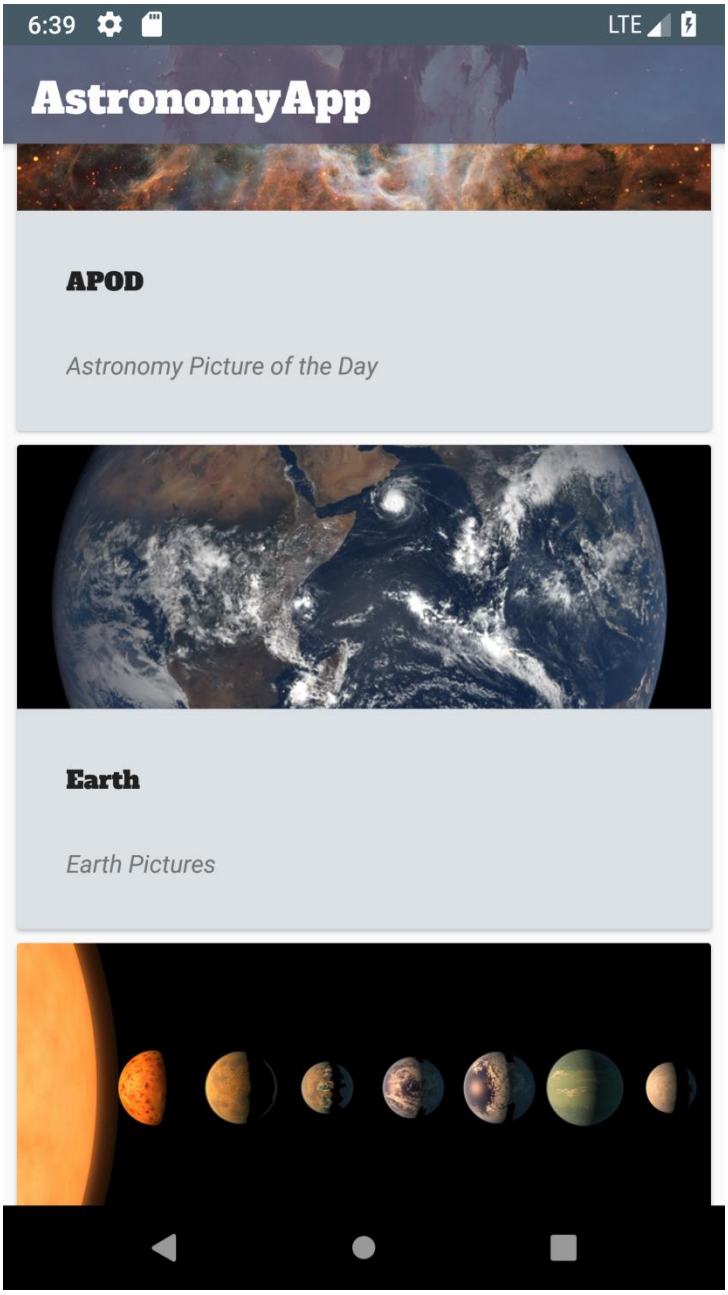
This app is especially suited for enthusiasts of astronomy and astronomy pictures, and is also useful for students of all ages.

Features

- Collects information and pictures via a NASA API and Wikipedia.
- Users can save APOD and Hubble data (implemented); and Earth (to be implemented).
- Users can save and share all pictures that can be enlarged.
- All data backup is done via SQLite.
- Displays Google Maps.
- Displays video via the YouTube API.

User Interface Mocks

Home Screen

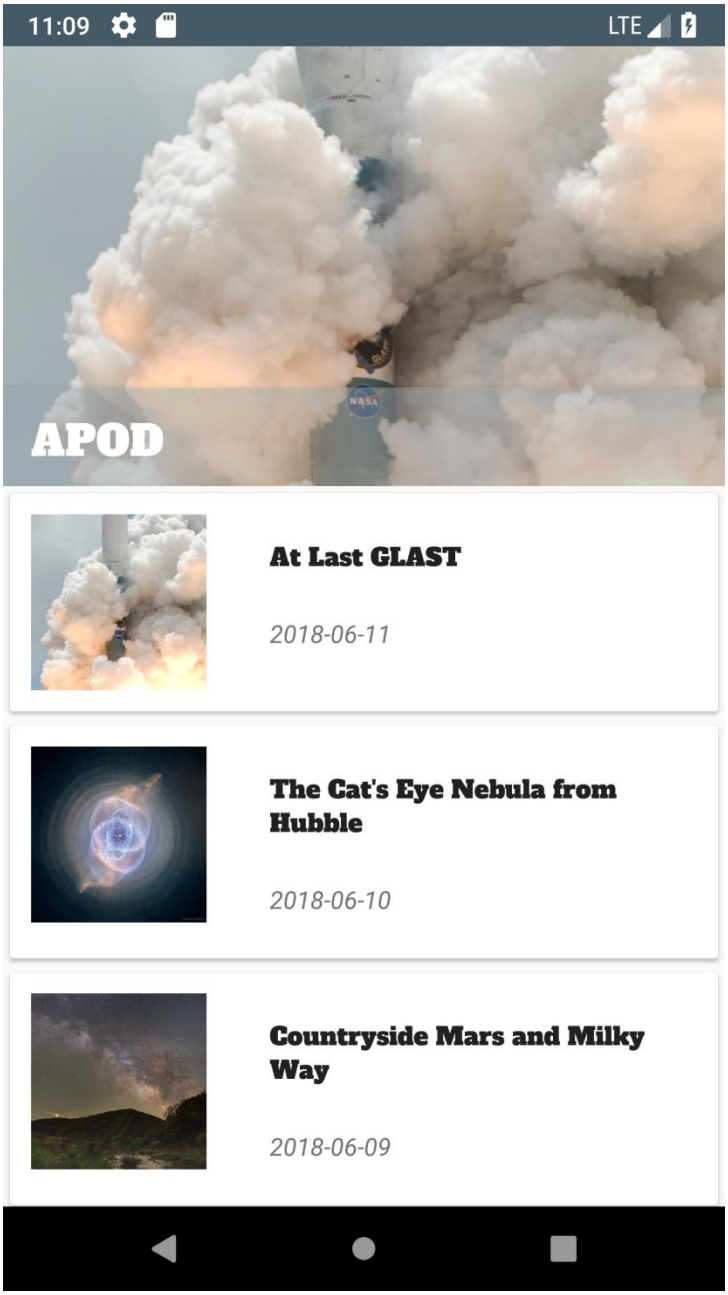


The **Home screen** is a list with pictures, their description and the app name.

List item details:

- Item title and subtitle.
- Picture.

APOD Screen



The **APOD screen** displays their picture of the day.

- List with APOD's last month pictures.
- Picture thumbnail.
- Title and date.

APOD Detail Screen

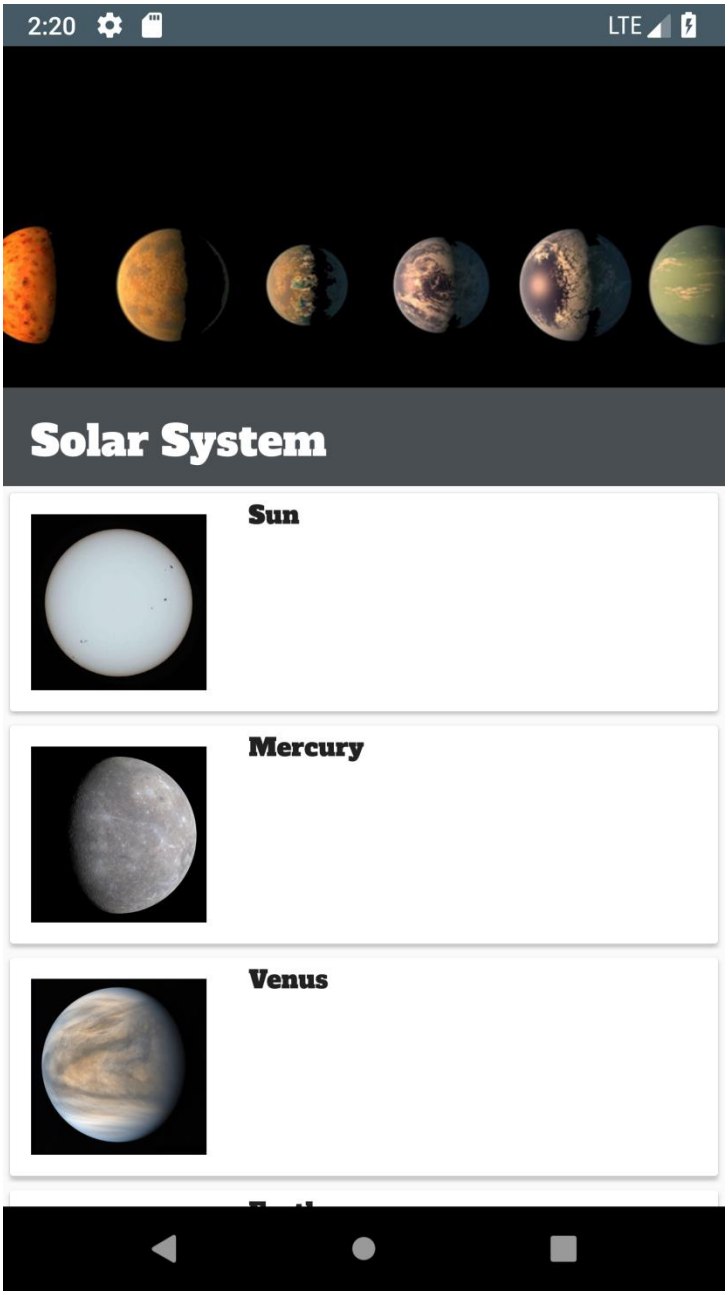
APOD Detail screen

- Users can click a picture to go to a new activity. Pictures can be videos which autoplay. Static pictures can be zoomed in, saved and shared.
- Picture title, date and copyright. If there is no copyright “No data” is displayed.
- Description.
- Items can be scrolled horizontally with a ViewPager.
- FLoatingButton to save the aforementioned information in SQLite. The APOD main screen will have an ImageButton to access favorites (to be implemented).

Earth Screen

To be implemented. It will be like the APOD section.

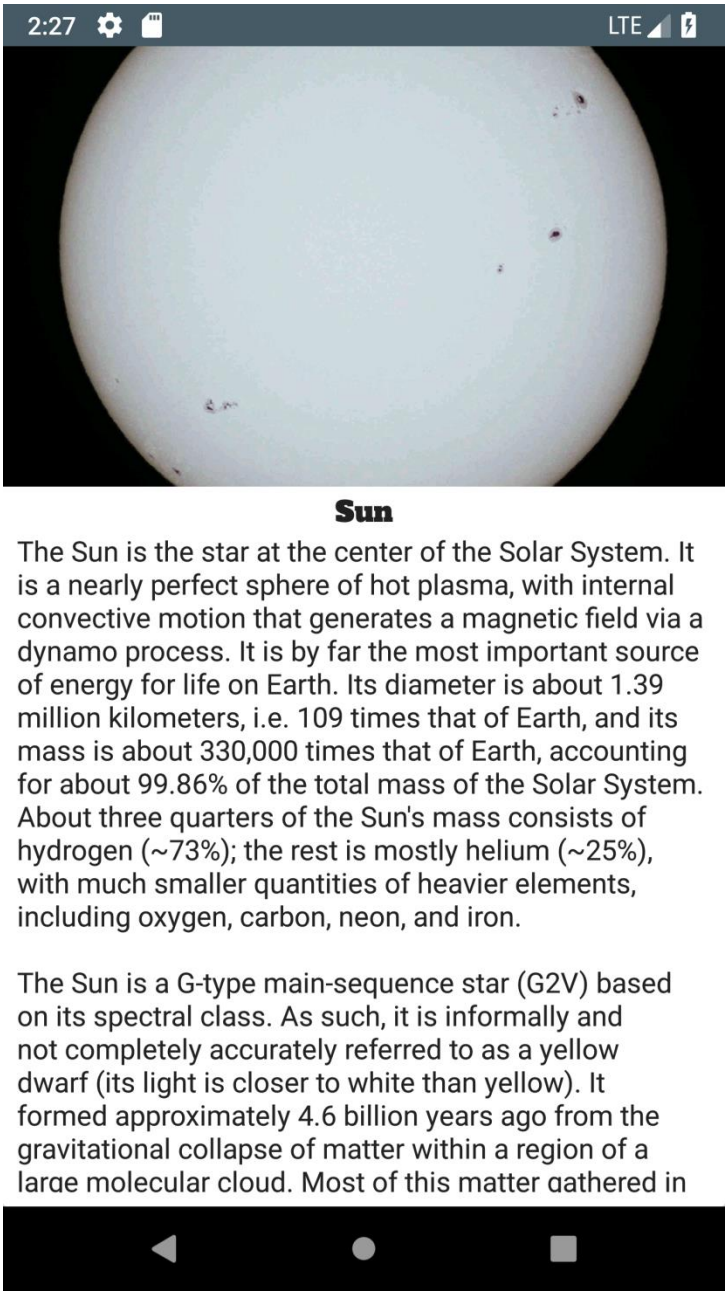
Solar System Screen



Solar System screen

- Lists the Sun, all Solar System planets, Pluto, the Galilean moons (Io, Ganymede, Callisto and Europa) and two Saturn satellites (Titan and Enceladus).
- Picture thumbnail.
- Title and description (to be implemented).

Solar System Detail Screen



Solar System Detail screen

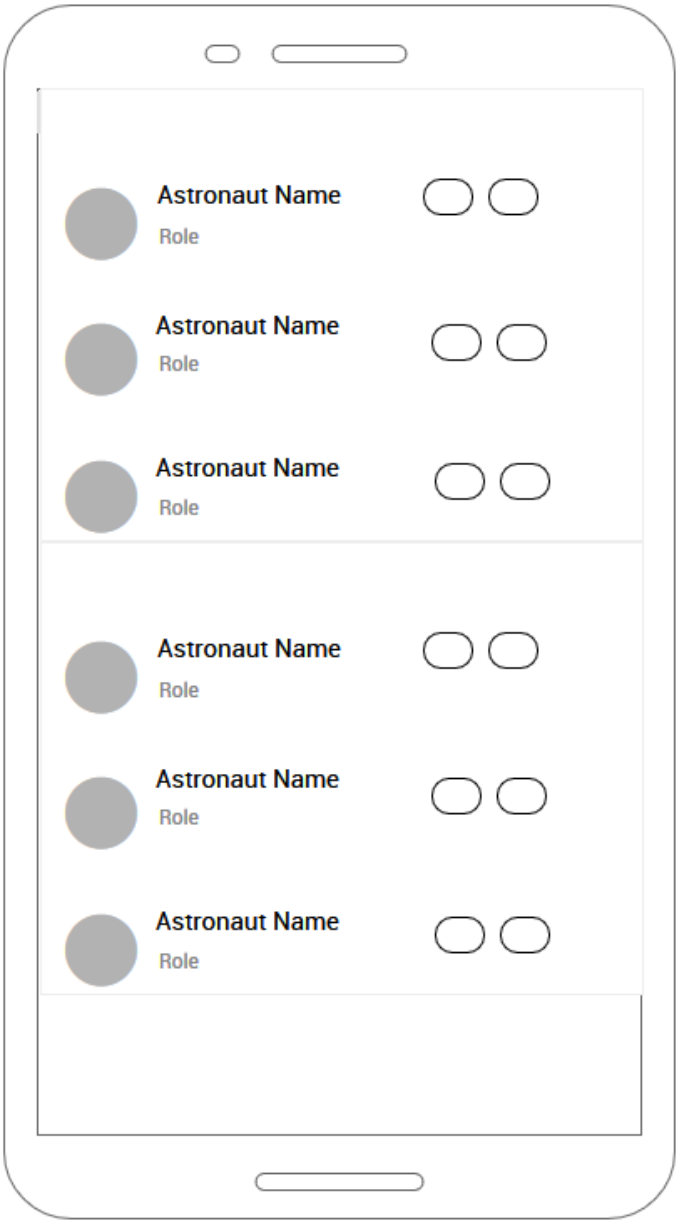
- Users can click a picture to go to a new activity. Pictures can be zoomed in, saved and shared.
- Title and excerpt. The excerpt is the Wikipedia entry summary.
- Items can be scrolled horizontally with a ViewPager.

ISS Screen

ISS screen

- Current ISS location. Line showing the ISS path in its last orbit (to be implemented).
- Latitude and longitude of its current position.
- Information about speed, altitude, GMT and whether it is on a night time area.
- Menu (to be implemented). It will open another activity (see below) that will display astronauts currently at ISS and a video from the YouTube API showing a live stream from ISS.

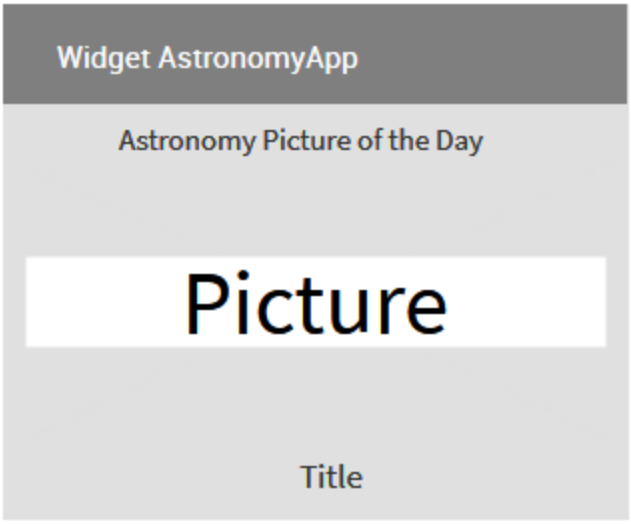
ISS Screen Detail



Astronaut Detail screen

- Picture.
- Name and role.
- Items go to their Wikipedia entry/personal site and buttons go to their Twitter and Facebook.

Widget Screen



- Widget name.
- APOD picture, updated daily.
- Picture title.
- Users can click the widget to open the app.

Comments

The UI needs finishing as regards graphics, text and general look & feel. There will be animations between activities and menus.

Key Considerations

How will your app handle data persistence?

All data will be stored on SQLite databases with a ContentProvider. Video position will be saved using savedInstanceState if the screen is rotated.

Describe any edge or corner cases in the UX.

All activities and fragments will go back to the parent when clicking the back button.

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

- Glide: To show pictures from a URL.
- Jake Wharton's Butterknife: To read code more easily.
- AppCompat, CardView and Design: For UI support and compatibility.
- Gson, Retrofit2 and Httpclient: To fetch data received from web APIs.
- Chris Banes PhotoView: To zoom in on pictures.

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

GSM Play Services will be used to display Google Maps, by implementing the OnMapReadyCallback interface and its methods.

Next Steps: Required Tasks

Task 1: Project Setup

- Create an Android Studio Project.
- Configure libraries in build.gradle (as the need arises).
- The app is written solely in the Java Programming Language.
- All strings used in this project are in a strings.xml file and RTL layout switching is enabled on all layouts.
- All the pictures include an adapted general content description, navigation using a D-pad and no audio.
- The app includes a widget to provide relevant information to the user on the home screen.
- The app will implement all the Core App Quality Guidelines specifications.

Task 2: Implement UI for Splash Screen

- Build UI for SplashActivity to preload pictures displayed on the Home screen, saved on a Firebase project.

Task 3: Implement UI for Home Screen

- Create UI layout whit CoordinatorLayout and CollapsingToolbarLayout.
- Create RecyclerView and its Adapter for the main list.

Task 4: Implement UI for APOD, Earth, Solar System and Hubble

This UI will be used for all the above screens.

- Create UI layout whit CoordinatorLayout and CollapsingToolbarLayout.
- Create RecyclerView and its Adapter for lists.

Task 5: Implement UI for ISS

- Create UI layout to display Google Maps and menu.

Task 6: Implement Activities and Fragments

- All activities and fragments in Task 4 are similar.
- RecyclerView for the UI and its Adapter.
- In APOD and Hubble, items can be saved on a database.
- Click item going to a fragment with ViewPager.
- Click item on a fragment going to an activity to view an enlarged picture with PhotoView or YouTube.
- In AndroidManifest, Provider will be implemented for APOD and Hubble. Uses Internet, Network State, Write External Storage, Access Coarse Location and Access Fine Location permissions.

Task 7: API Services

- ISS activity uses AsyncTask because the on-demand request takes only one second.
- APOD, EARTH and Hubble use SyncAdapter.
- Solar System uses IntentService.

Task 8: API Widget

- The app includes a widget that provides the APOD picture and title, which are updated daily.

Task 9: Optimize UI for Landscape and Tablet

- Create new layouts with optimized sizes.