

App Name: Untamed Africa

Description:

Untamed Africa is a mobile application that provide users educational information about African wildlife that can can be typically be sighted at safaris in the continent. The animals are categorised based on their endangerment status and details on each specie are displayed along with videos and images of them in their natural habitat.

This App aims to educate and bring attention to the reality we are usually blind to as humans. This aims to highlight the effects of poaching to the different species, how changes in the environment affects their natural habitat and mostly the conservation efforts that help these species from being completely extinct. This offers a way for ordinary people to get involved efforts efforts to save these animals. This app will also provide statistical data in real-time and the news in this small community.

Intended User:

Anyone that loves nature and wants to contribute in preserving and protecting the animals.

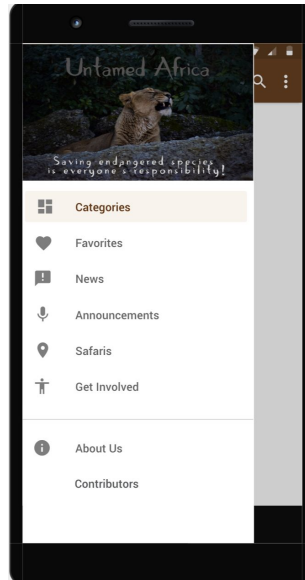
Features:

- **Categories:** These are all animals grouped by their endangerment status and user can select each category see more details of the animal , including facts, images and videos.
- **Favorites:** this is an extension of the categories feature, animals that are part of this feature would be those that the user has favorited and are persisted locally to the device and are available offline.
- **News:** These are real-time news regarding the animals from Rhinos being poached to the increase of their population in the wild.
- **Announcements:** These are user alerts, things that the user can action for example, if there's a fundraising drive the user can see it here and either choose to accept or ignore it.

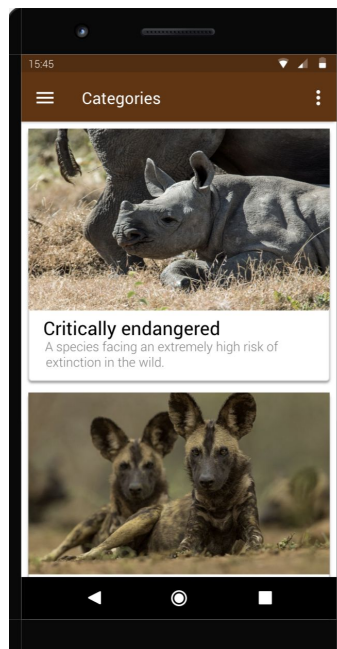
- **Safaris:** This will give you a list of all safaris in the continent where you can spot the animals and it will give the list of the closed safari based on the location.
- **Get Involved:** These are all the ways users can take part in helping conservation effort be it joining a non-profit organisations, donating and spreading the word.

User Interface Mocks:

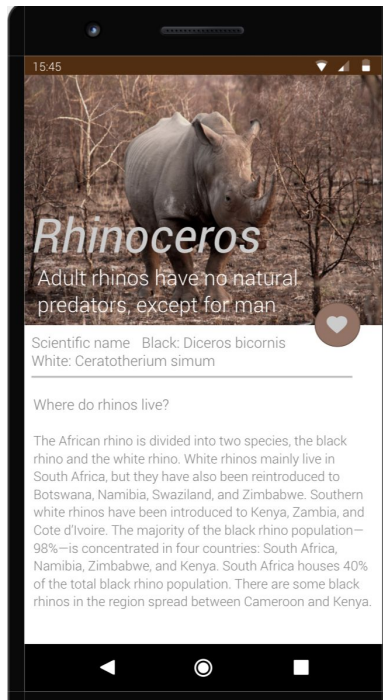
- Navigation drawer showing all the features:



- Categories showing arranged animals by categories:



- Details when animal is selected



- The app widget showing facts of the animal



Key Considerations

Tools Requirements:

- Android Studio 3.1.3
- Java 8 Programming Language
- Gradle version 3.1.3 or higher
- Compile version: 28
- Target SDK version: 28 (Pie)
- Minimum SDK version: 15 (Ice Cream Sandwich)

How will your app handle data persistence?

The app will be using Firebase Realtime Database to store data remotely. This will allow the app to sync data while online and persist it to disk, making it available offline, even when the user or operating system restarts the app. The app will work as it would online by using the local data stored in the cache.

Describe any edge or corner cases in the UX.

When the app is launched the user will land to the categories screen, the user can use navigation drawer to switch between various features of the app. Selecting a category will open a different screen with a list of all the species that are in that classification and when a list item is selected the a more detailed view is launched with images, videos and details such as habitat, diet, etc.

When on the categories screen the app closes when the users taps back but when on other screens the app defaults to the categories feature and then app closes should the user tap back again.

When the user selects an image of video, the app will open that media files in full screen mode. When user taps back it goes back to the list of media files associated to the selected animal.

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

1. [Glide v4](#) (Version 4.9.0) - Glide is a fast and efficient image loading library for Android.
2. Exoplayer (version 2.9.0) - It provides an alternative to Android MediaPlayer API for playing audio and video both locally and over the Internet.
3. Dagger 2 (version 2.16) - for dependency injection.
4. Mockito (version 2.1.3) - mocking dependencies for unit testing.

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

1. Firebase Database - for real time data storage
`com.google.firebase:firebase-database:17.0.0`
2. Firebase Storage – to store the images
`com.google.firebase:firebase-storage:17.0.0`
3. Firebase Analytics – for app usage reporting
`com.google.firebase:firebase-analytics:16.5.0`
4. Firebase Crash – to report on crashes
`com.google.firebase:firebase-crash:16.2.1`
5. Google Maps - to show the location of the safaris
`com.google.android.gms:play-services-maps:16.1.0`

Required Tasks

Task 1: Project Setup

- Create Project.
- Configure Libraries and add all the dependencies.
- Setup the Dagger 2 and add the base dependency injection.
- Setup packaging and arrange packages by features using the MVVM design pattern:
 - Service
 - Repository
 - Dependency
 - Firebase
 - Utils
 - Model
 - View

- ViewModel
- Configure product flavors
 - Release
 - Development

Task 2: Implement UI for Each Activity and Fragment

- Navigation Drawer:
 - Add the main activity that holds the navigation drawer and contains a view where the different feature fragments will be inflated.
- Implementation of Categories feature:
 - Create a CategoriesActivity
 - Build UI for CategoriesFragment
 - Build UI for SelectedCategoryFragment
 - Build UI for SelectedAnimalFragment
- Implementation of Favorites feature:
 - Create a FavoritesActivity
 - Build UI for FavoritesFragment
- Implementation of News feature:
 - Build UI for NewsActivity
- Implementation of Safaris feature:
 - Build UI for SafarisListActivity
 - Build UI for SelectedSafariFragment
- Implementation of Announcements feature:
 - Build UI for AnnouncementsListActivity
 - Build UI for SelectedAnnouncementFragment
- Implementation of Get involved feature:
 - Build UI for GetInvolvedActivity

Task 3: Firebase Setup

- Create Firebase project on the Google developer console
- Add Firebase to the app
- Store animal media files in Firebase storage
- Create entities in Firebase
- Create POJO in Android Studio to match the Firebase entities
- Setup Firebase dependency injection module

Task 4: Implementation of the Business Logic

- categories :
 - No network call, navigate to each feature by selecting the items on the navigation drawer.
- Implement the viewModels for all the stated in task 2.
- The viewModels must handle all the data transfer logic using asyncTasks and provide the response to the view by injection.
- For favorites use Room database to save the favorites locally and use a AsyncTask to facilitate that process on the background.

Task 5: Implement the App Widget Service

- Implement a Remote View Service for the app widget

Task 6: Test the App