GitHub Username: alviksar

Orchidarium App

Description

Orchidarium App is online shop that offers you to choose and order the best orchids from around the world anywhere and anytime.

Intended User

Orchidarium App is for everyone who admires the unearthly beauty of exotic flowers and would like to acquire the most beautiful of them. In addition, it provides a tool for authorized users who are store administrators to manage the stock available to order by customers.

Features

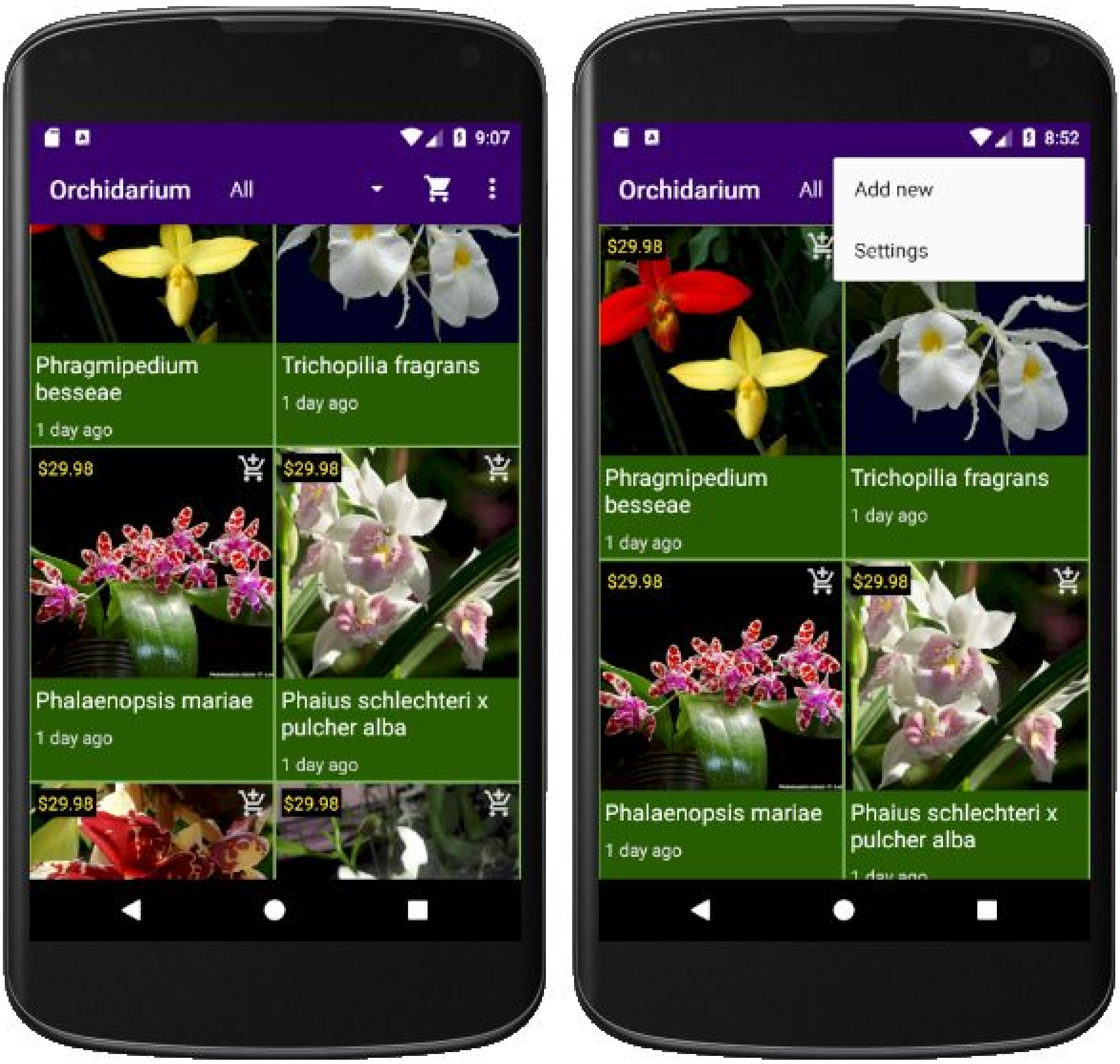
Depending on the role, the application allows users to perform the following functions:

- For the authorized Store Administrators it has activities to add orchid's photos, set the retail price, and then put it up for sale.
- For ordinary users authorization is not required. Customers of the store will be provided with photos of the product with a brief description.
- Users can choose in cart the flowers they like, which will be included in the order for sending by email.
- Customers can subscribe to notifications of new orchids if they want.

The application will be built in the Java programming language.

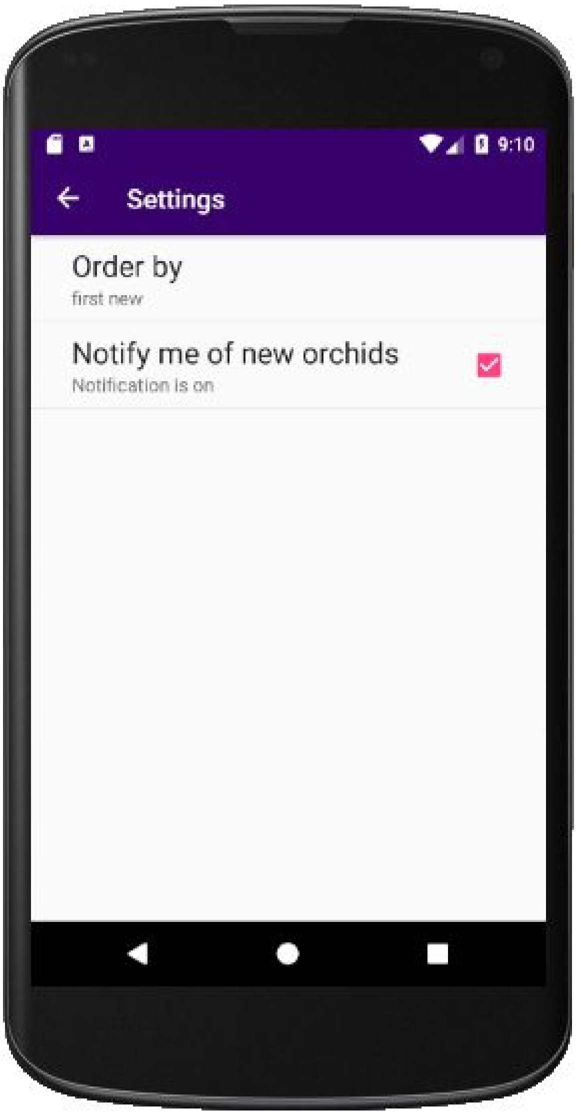
User Interface Mocks

Screen 1 - MainActivity



The main screen displays orchids which available to order. For each of them it shows a price and time when this orchid was set up for sale. There is also main menu that contains items for making order and adding new orchids for sale (visible for admins only).

Screen 2 - SettingsActivity



9:10

Settings

Order by

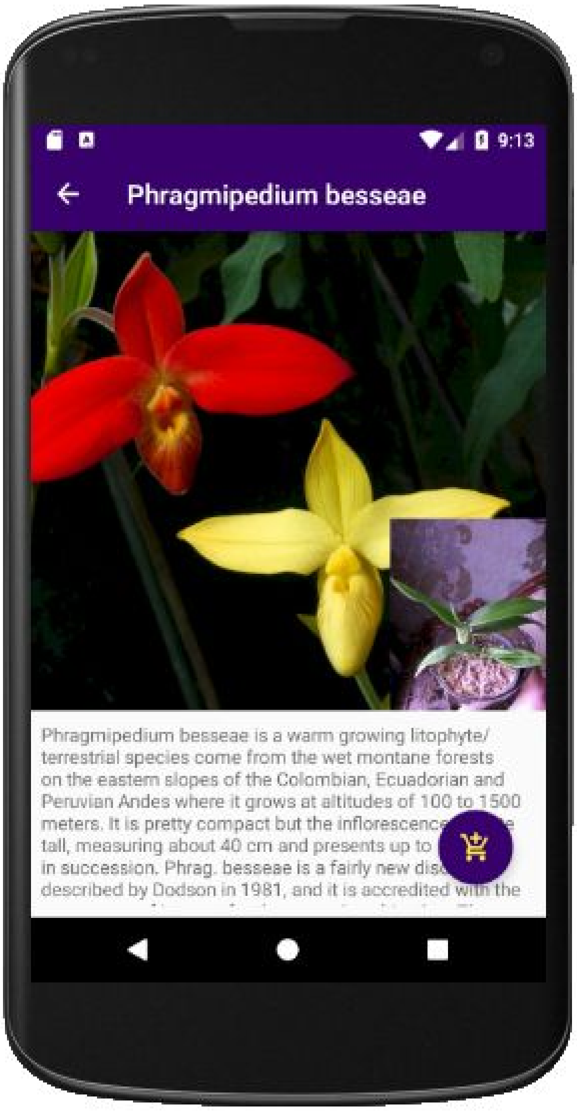
first new

Notify me of new orchids ☑

Notification is on

Settings allows user to choose a sort order in the MainActivity and switch on/off notification of new orchids for sale.

Screen 3 - DetailActivity

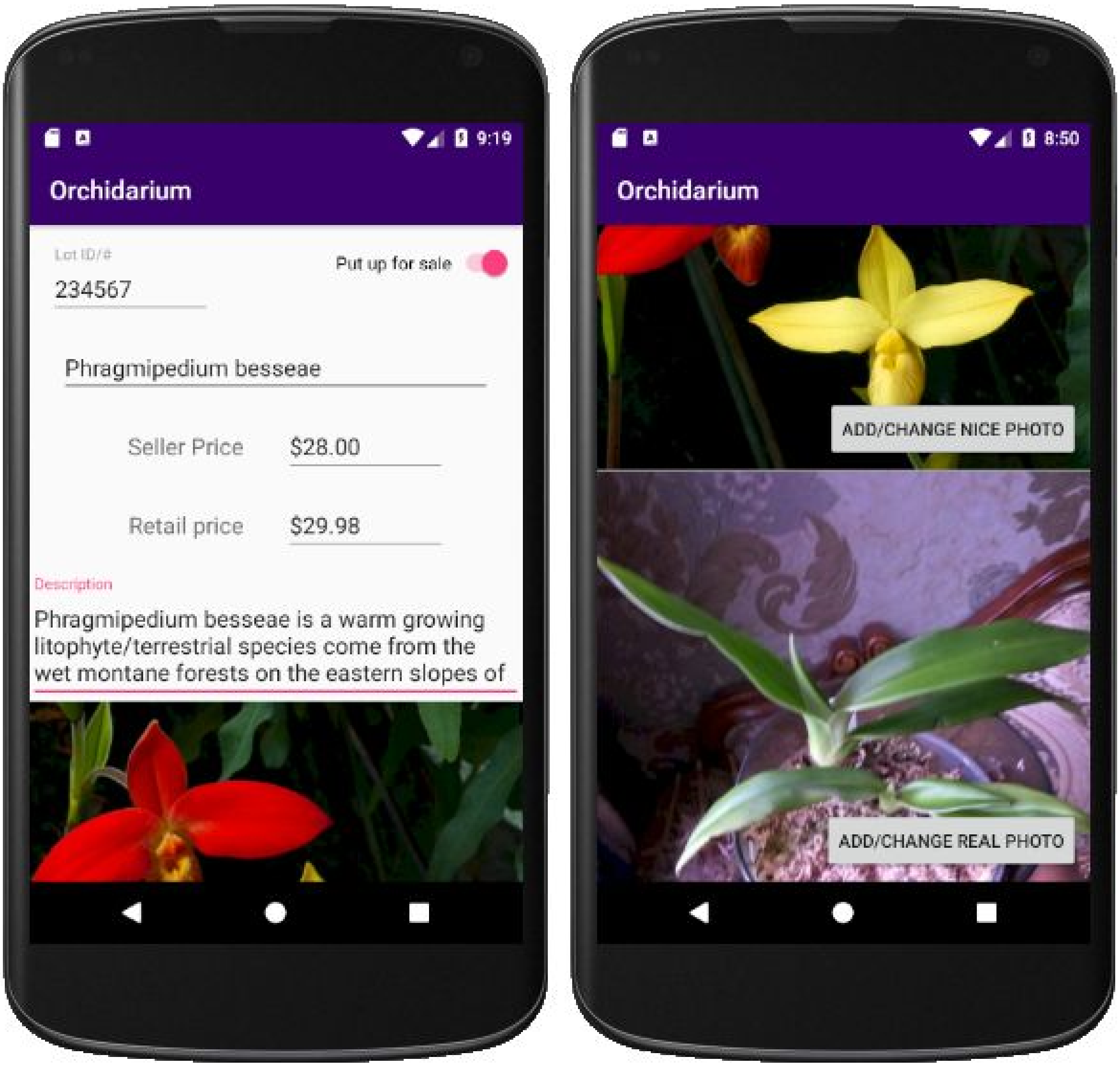


Phragmipedium besseae

Phragmipedium besseae is a warm growing litophyte/ terrestrial species come from the wet montane forests on the eastern slopes of the Colombian, Ecuadorian and Peruvian Andes where it grows at altitudes of 100 to 1500 meters. It is pretty compact but the inflorescence [illegible] tall, measuring about 40 cm and presents up to [illegible] in succession. Phrag. besseae is a fairly new dis[illegible] described by Dodson in 1981, and it is accredited with the

The detail activity displays photos of orchids and a brief description. It has a floating action button that marks this orchid for inclusion in the order list. There will be an edit button for administrators that provides a link to the StoreAdminActivity to modify the data.

Screen 4 - StoreAdminActivity



Lot ID/#
234567

Put up for sale

Phragmipedium besseae

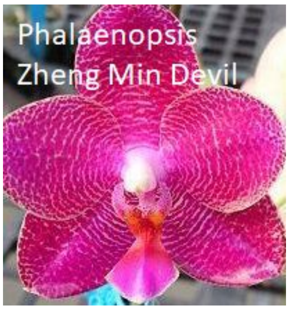
Seller Price $28.00

Retail price $29.98

Description
Phragmipedium besseae is a warm growing litophyte/terrestrial species come from the wet montane forests on the eastern slopes of

This activity is for changing data and adding new orchids to the application database. Application assigns an unique number for each newly added orchid. In addition, the administrator should write a name and a brief description. "Nice photo" shows orchid in all its glory at the best possible time. "Real photo" shows the plant as as it is now in fact.

Screen 5 - App's widget



App’s widget displays a new orchids in store or random of orchid images.

Key Considerations

How will your app handle data persistence?

Orchidarium App will use Firebase Realtime Database for store simple data and Firebase Storage feature for store images. https://firebase.google.com/products/

The application will use the Preferences to store the set of orchid ids that an user puts in a cart.

Describe any edge or corner cases in the UX.

- The store administrator can upload images of flowers from the gallery and also take photos with the camera.
- The detail activity will be animated. If the user tap on a smaller image, it will swap places with a large one.
- User can put orchid in cart by tap on the FAB in Detail Activity. The application saves information about the selected products locally and checks the current availability of each position at the time of order making.
- The application will use common implicit intent for sending an order by email https://developer.android.com/guide/components/intents-common#Email
- The application will provide a widget which shows a random or new of orchids. The widget will use an IntentService for choosing and downloading images.
- App keeps all strings in a strings.xml file.

- App enables RTL layout switching on all layouts.
- App includes support for accessibility. That includes content descriptions and navigation using a D-pad. There are not any audio cues, so non-audio versions of them is not applicable.

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

Application will using following libraries:

- Glide to handle the loading and caching of images.
- Butter Knife to find and automatically cast the corresponding view in your layout.
- Timber to implement different logging logic for Debug and Release version of application.

Library Name	**Version**	**URL**
Glide	4.7.1	https://github.com/bumptech/glide
Butter Knife	8.8.1	http://jakewharton.github.io/butterknife/
Timber	4.7.1	https://github.com/JakeWharton/timber

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

Application will using following Firebase services:

- Realtime Database for storage of data about goods and authorized users in database JSON format.
- Firebase Storage feature for store images.
- Authentication service for authentication and distinction of shop administrators.
- Cloud Messaging for notification users about new orchids coming.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Include in gradle and configure libraries
- Create Firebase Project.
- Add Firebase to Orchidarium App
  https://firebase.google.com/docs/android/setup?authuser=0

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for SettingsFragment and SettingsActivity
- Build UI for DetailActivity
- Build UI for StoreAdminActivity

Task 3: Implement data workflow

- Implement functions for adding new data, changing and saving it in Firebase Realtime Database.
- Load image in Firebase Storage and save link to it in database.
- Implement a delete data and images function.
- Retrieve data from Firebase Realtime Database for different sort and selection modes.
- Mark chosen items and save list of chosen items locally.
- Implement making an order.

Task 4: Create user roles

- Implement authentication.
- Change UI for admins and customers.

Task 5: Implement additional features

- Implement notification with Cloud Messaging.

- Make a app’s widget and related IntentService for refreshing its image.

Task 6: Implement responsive design features

- Create detail layouts for landscape mode .
- Change UI for tablet mode.

Task 7: Final test and release application

- Run integration tests.
- Handle error cases.
- Create build variants.
- Sign the application release.