## Android Fundamentals Project Self-Evaluation

**Instructions:** Once you’ve completed your Final Project, please respond to the questions below. This is a chance for you to briefly explain to the grader your thought-process during development. Once you are done, include this with the source code and accompanying files you are submitting. Then, give yourself a pat on the back for making a great app!

# Questions about Required Components

## Permissions

**Please elaborate on why you chose the permissions in your app.**

|  |
| --- |
| *<!-- Permissions required by the google cloud system adapter in order to receive notifications -->*  *<!—Permissions internet required by the app to function -->*  *<!—Permissions storage required to store the images -->*  *<!—Permissions main mail account required to know what is the main email account so the user can type less - can be dismissed in the future when the login will work differently>*  *<!-- Permissions to read contacts required by the system to know who the users’ friends -->*  *<!-- Permissions required by wake on boot by the notification service – so the user will get update when the phone starts – this is only a backup so every once in a while the app will still pull data>*  *<!-- Permissions phone state required to know what is the id of the user -->*  *<!-- Permissions required by the sync adapter -->* |

## Content Provider

**What is the name of your Content Provider, and how is it backed? (For example, Sunshine’s Content Provider is named WeatherProvider backed by an SQLite database, with two tables: weather and location.)**

|  |
| --- |
| ProductsContract backed by sqllite . tables : Users .Products. Messages. Subscriptions |

**What backend does it talk to? (For example, Sunshine talks to the OpenWeatherMap API.)**

|  |
| --- |
| It talks to my own Web application and api <http://ymarq.azurewebsites.net>. |

**If your app uses a SyncAdapter, what is it called? What mechanism is used to actually talk over the network? (For example, Sunshine uses HttpURLConnection to talk to the network, but your app may use a third-party library to do the talking.)**

|  |
| --- |
| YmarqSyncAdapter uses **HttpURLConnection.** |

**What loaders/adapters are used?**

|  |
| --- |
| CursorLoaders are used for products (ProductsBuyerFragment2  And ProductsSellerFragment  Another loader is in message fragment for the details of the product.  There are also adapters for the subscriptions and contacts |

## User/App State

**Please elaborate on how/where your app correctly preserves and restores user or app state. (See rubric for examples on this question)**

|  |
| --- |
| The app stores in the database the user data and after the first login, the user will no longer see the login activity.  The app stores products subscriptions and messages so the user will be able to see data even if he is offline. He will not be able to push data (publish product or write message )though.  There are also options stored in the settings / shared preferences |

# Questions about Optional Components

Answer the questions that are applicable to your final project

## Notifications

**Please elaborate on how/where you implemented Notifications in your app:**

|  |
| --- |
| When a product is received – from a friend OR when a new product matches one of the users subscriptions a notification is received  A notification is also received when a message is received.  The server send a google cloud message that is intercepted and the app starts pulling the latest notifications ( notificationservice) |

## ShareActionProvider

**Please elaborate on how/where you implemented ShareActionProvider:**

|  |
| --- |
| I’ve implemented shareProvider as in Sunshine. I just shared a picture of the product |

## Broadcast Events

**Please elaborate on how/where you implemented Broadcast Events:**

|  |
| --- |
| There are couple of BroadcastReceivers :  One intercepts google cloud messages and starts the notification service to pull data from the server  There is another one on the chat fragment that receives the new message flag and refresh the UI.  There is another one that starts on each boot or on the first login and ensures that as a backup , once every period (it can be couple of hours) the app will pull notifications from the server |

## Custom Views

**Please elaborate on how/where you implemented Custom Views:**

|  |
| --- |
| Currently there is no custom view. |