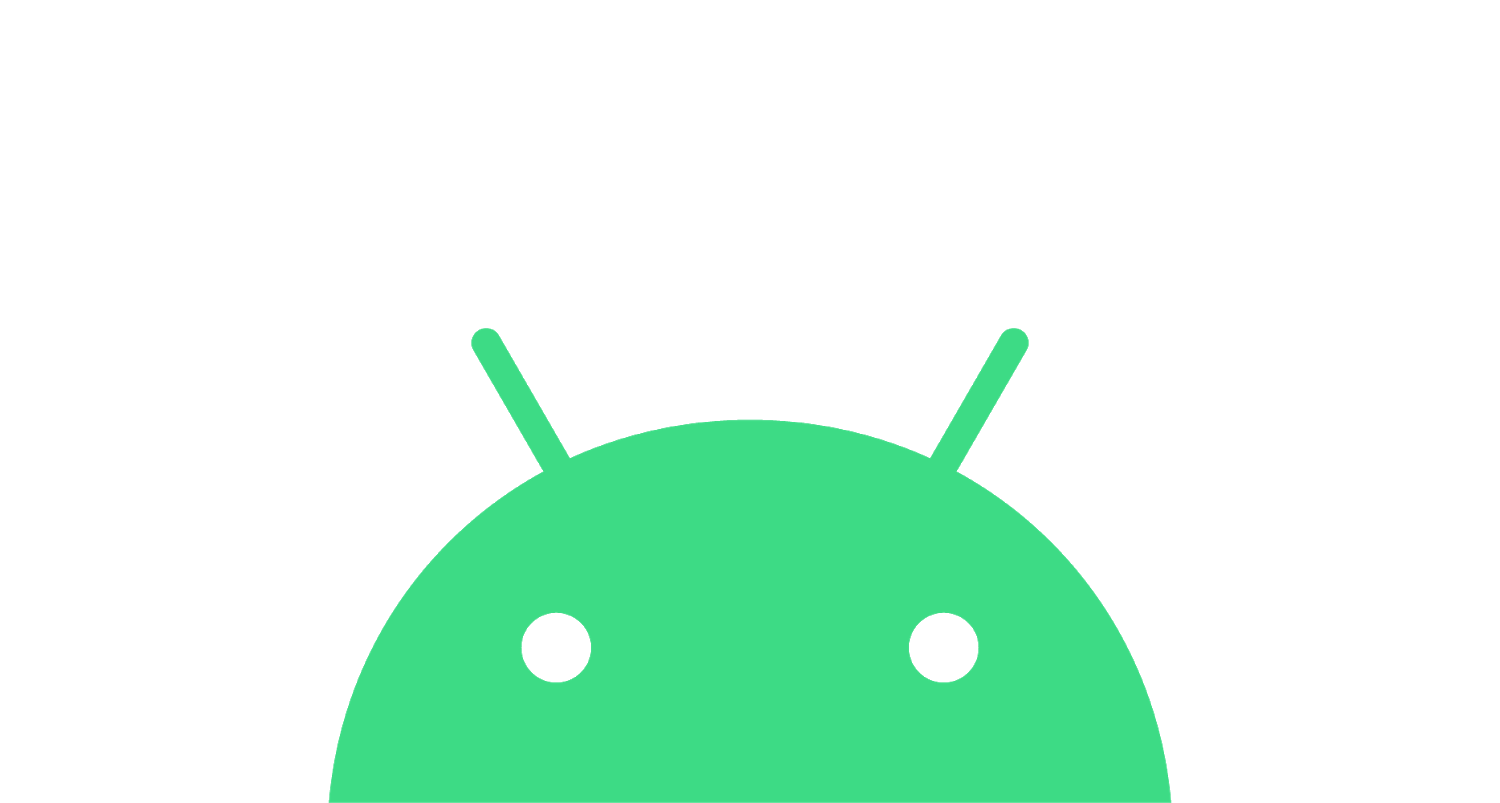
## 



Plant Swap App

SYNOPSIS - ITSMAP Fall 2020

**Group 6**

Group Members

Valeria Polukhina Wellejus au547175 IKT

Niclas Spas au309804 IKT

Trí Nguyen au564065 IKT

Martin An Vo au569472 E

Document version: 1.1

Changelog :

* **Changes to version 1.0**
  + Added more view design mockups
  + Added considerations
  + Added more diagrams to Wireframe
  + Added Recycler views, View models, Live data, and Fused location provider to Technical Components

# App Vision

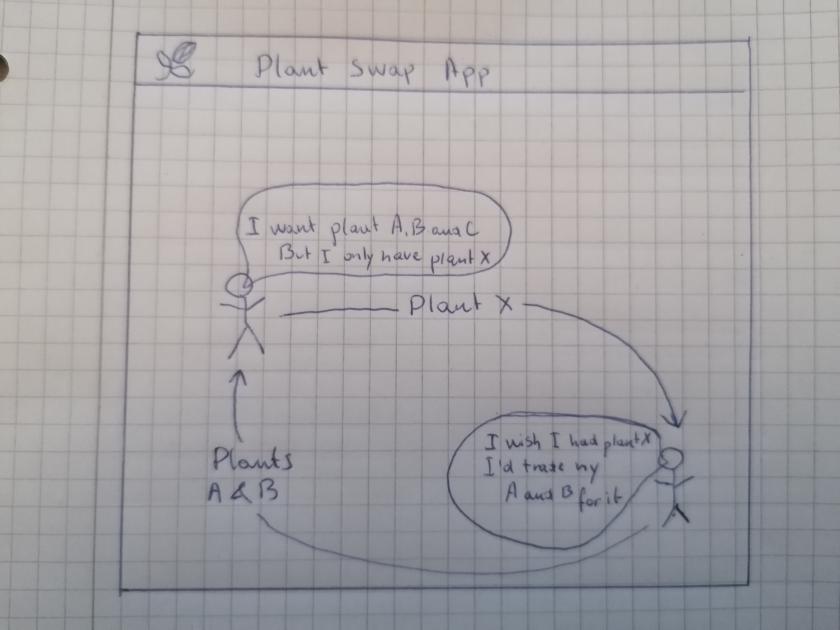
The app has a purpose of making it easier for plant collectors to swap plants. Often when selling buying or trading plants it can be a pain to negotiate a fair swap for both parties and then exchange personal information so the plants can be sent. With this app collectors won’t have to talk to each other or negotiate, they will simply make a wish list when putting something up for swap. If a plant from the wishlist has popped up within a given distance, the user will receive a notification. The user can then see the add and offer a plant from the wishlist for swaps. The other part can decline or accept the proposal. There’s also no need to exchange information as it is already in the system.

# Personal Vision

As a group we wish to learn the aspects of the course that we have not yet used in our previous assignments. That includes Firebase for data storage and location sensors for finding an item with a specific range. We would also like to create a product that will potentially be used by others. Plants are just an example of trade items; it is possible to expand the app with other kinds of items or add payment options to the designed app to broaden the functionality.

# Context

The app is intended as a multi-user app, where users can trade plants with each other. The ‘available’ plants would be fetched from the API <https://trefle.io/>. This will only give users specific plants to choose from and although it hinders the options it also hinders the users from writing gibberish and make spelling errors as one will see on Facebook’s Marketplace. It will integrate with Firebase for data storage and use permissions to access phone’s camera, gallery and location. When a plant is put for swaps it will be added to the database and if the name will match another user’s wishlist choice and the location is within the specified range a user will get a notification about their dream plant being available near them.



*Rich picture showing the core functionality*

# Requirements

## User stories

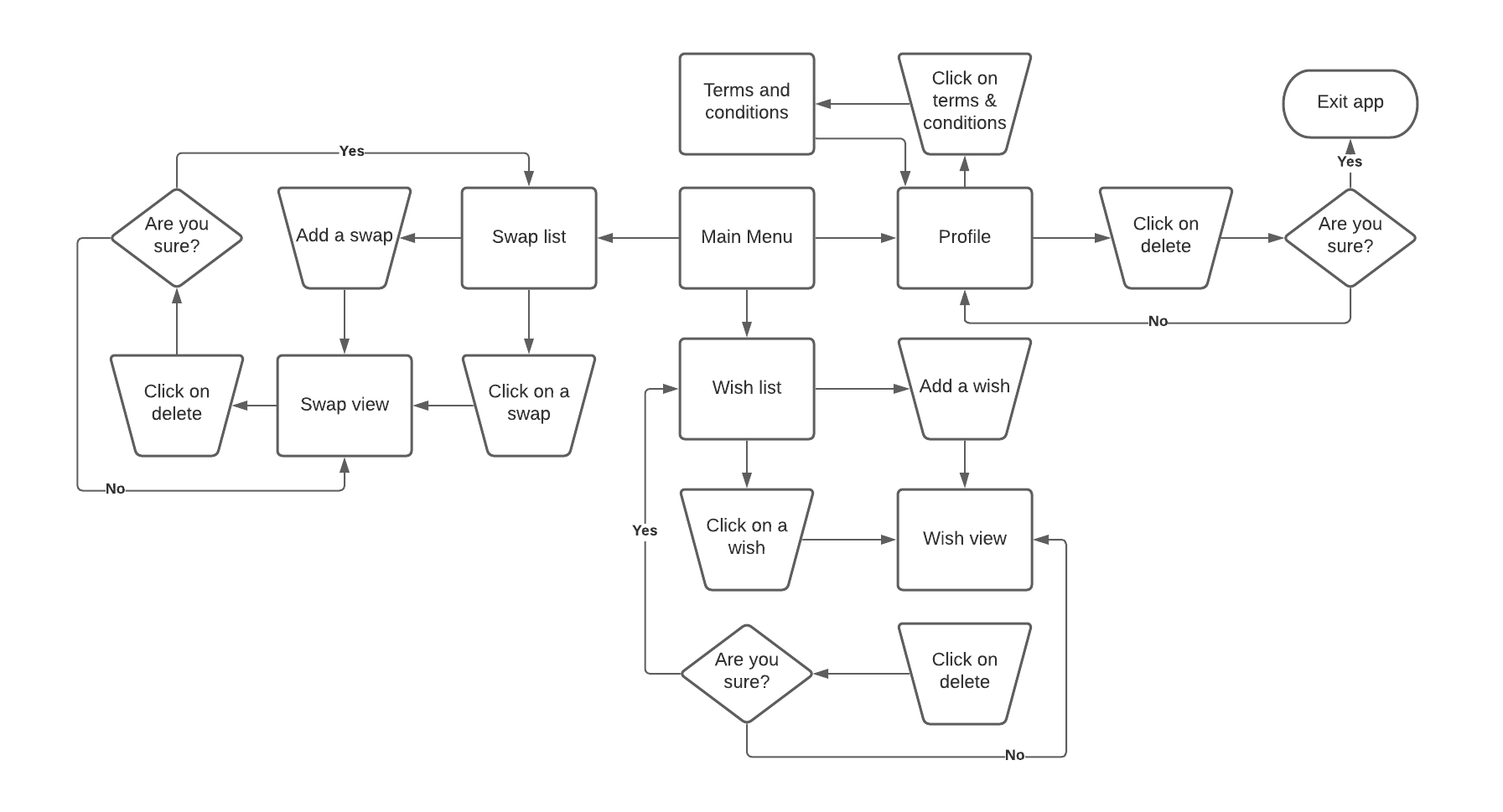
1. As a user I can register with my contact information
2. As a user I can get a notification whenever an interesting item is within a specified range
3. As a user I can edit my notification settings (broaden or shorten range for a plant or add a new plant)
4. As a user I can put a plant for swap
5. As a user I can add, change or delete a plant from my swap list
6. As a user I can either accept or decline a swap proposal
7. As a user I can propose to swap plants I get notified about
8. As a user I cannot involve the money in the deal

# App Design and System Overview

## Wireframe

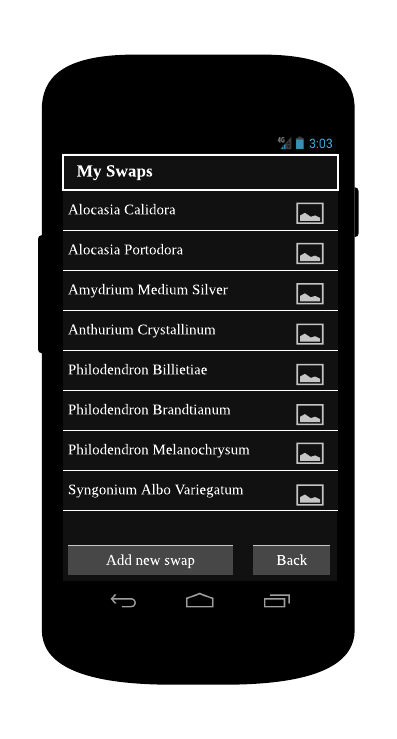
The flow depends on whether the user is registered or not. If the user is not registered;

* Profile 🡪 Terms & conditions 🡪 Profile 🡪 Add a wish 🡪 Add a swap 🡪 Main Menu

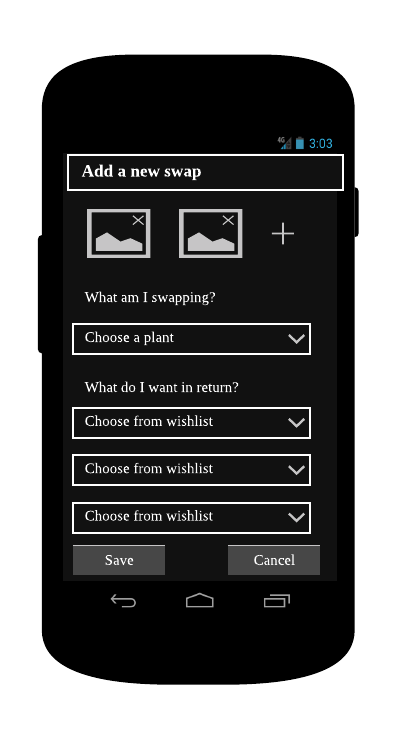
If the used is registered the flowchart shows possible navigation in the app. The content of the swap view and wish view depend on whether the user clicked ‘Add a swap/wish’ or clicked on an existing item.

*Flow of the app*

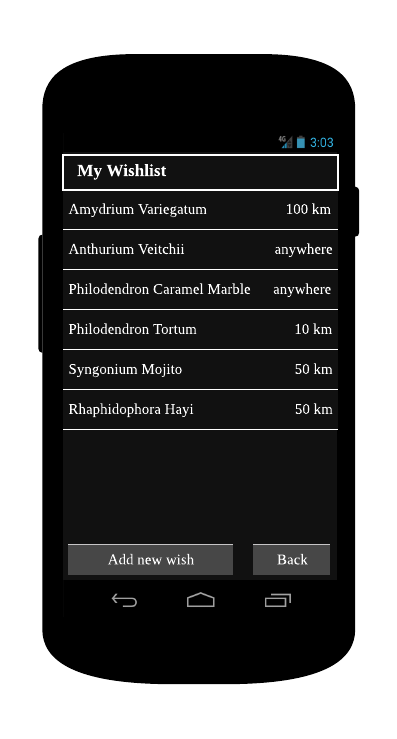
Example of the different views is shown below



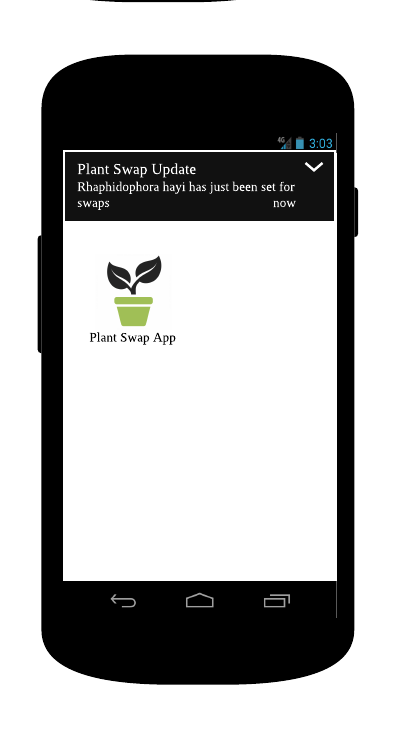
*Main menu Swap list*



*Add a swap Change a swap*



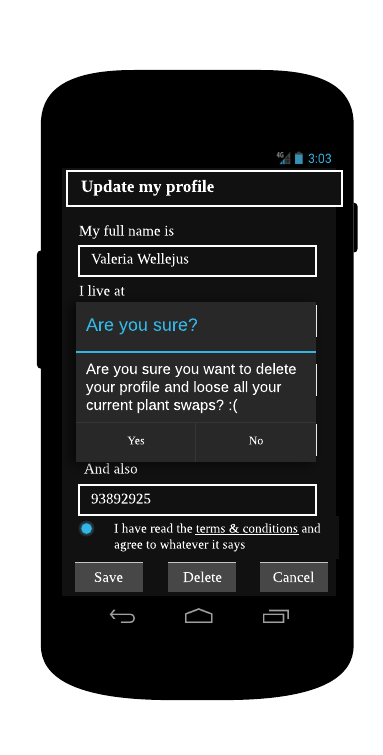
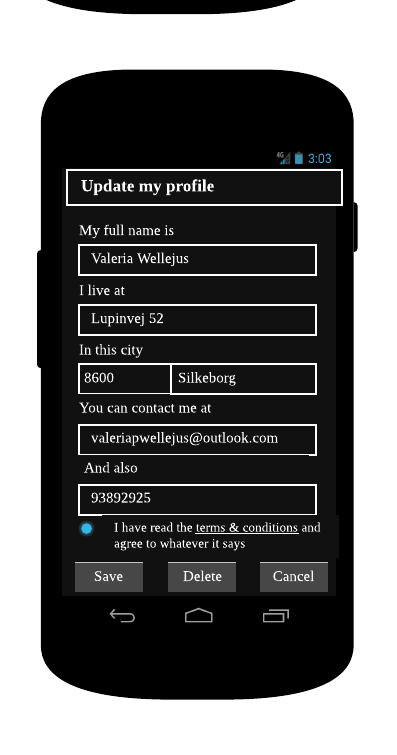
*Wish list Add a wish*



*Edit a wish Show notification*



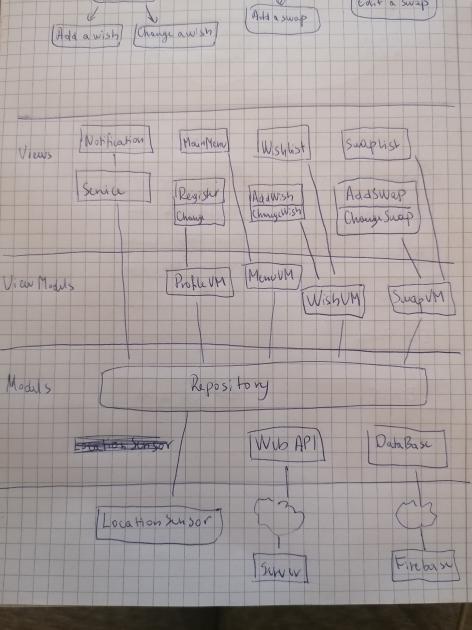
*Register site Terms and conditions*



*Edit profile Dialog box when ’Delete’ is clicked*

## Architecture overview

Connection between different components:



Technical Components

List of all Android SDK and external libraries/dependencies:

* Activities
* Services
* RecyclerView
* ViewModels
* LiveData
* Firebase
* Fused Location Provider
* Volley
* Glide
* Notifications
* Permissions (camera and location)

# Other considerations

Possible problems with GDPR when filling out the register form. The solution to that is to write about the issue in ‘terms and conditions’.

Possibility to add machine learning for plant recognition when taking a photo of it.