

**GitHub Username:** luisfnicolau

# Asquui

## Description

Asquui make it easier to know opinions and points of view of your friends. YOu can post questions that your followers will answer or you can answer their questions.

## Intended User

FOr general user, for everyone that want to ask a question to people around.

## Features

- Allow post of questions
- Organize friends questions in a timeline
- Save location for discover purposes
- Allow to post images with questions

## User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, [www.ninjamock.com](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

### Screen 1

Replace the above image with your own mock [ click on the above image, then navigate to Insert → Image... ]

Provide descriptive text for each screen



Main Page of app, show posts of users that user follow. On each post you can answer, comment, see extra images and choose extra options like follow or unfollow.

## Screen 2



This screen you can navigate through posts of users to discover ad it has a search bar to search for specifics posts or users.

### Screen 3



This screen you can see your profile and posts, edit them or go to comments list

## Screen 4



Some Nice comment

Reply



Some Nice comment

Reply



Some Nice comment

Reply



Some Nice comment

Reply



Some Nice comment

Reply



Some Nice comment

Reply



Some Nice comment

Reply

Comments list of a post

## Screen 5

Asquui

Your question here

☒ One option question +

☒ Option I

☐ Multiple answer question +

☒ Option I

☐ Text answer

Preview:

Your quwstion here

☒ Tomato

☐ Salad

☐ Juice

Here you can create a post and see a preview before send it.

## Key Considerations

**How will your app handle data persistence?**

Use Firebase Realtime Database to handle data persistence.

**Describe any edge or corner cases in the UX.**

Return to correct screen when accessing comments from Screen 1 or Screen 3.

Return to correct post in the list when back from comments list.

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

Picasso to handling load of images that users can post.

Retrofit2 to handle http requests.

FirestoreUI and Firebase Auth to handle Authentication.

Firebase Storage to store images that users will post.

Transitions Everywhere to make easy to do smooth animations.

Otto to handle Bus events

ButterKnife to make easy bind activities and views.

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

Use Google Play Location Services to discover users that are next to another.

## 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

- Initiate a project with a bottom navigation bar.
- Configure libraries that will be used

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Post Layout
- Build UI for comment text

### Task 3: Implement Models to use in Database

### Task 4: Implement Adapters

- Implement Posts list adapter
- Implement Comments list adapter

### Task 4: Setup Firebase Realtime Database

- Get Posts to Post List
- Get comment list

## Task 5: Implement Authentication with FirebaseUI

- Set app icon and name in FirebaseUi Auth.
- Prompt to a username.
- Update UI