

# Know Your Opinion

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Welcome Screen](#)

[HomeFragment](#)

[QuestionActivity](#)

[HomeFragment \(Tablet\)](#)

[QuestionActivity \(Tablet\)](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Firebase Setup](#)

[Task 2: Project Setup](#)

[Task 3: Implement UI for Each Activity and Fragment](#)

[Task 4: Implement Google Play Services](#)

[Task 5: Implement Content Providers and Loaders](#)

**GitHub Username:** [angads25](#)

## Know Your Opinion

### Description

Know your Opinion is a voting app that allows you to put up your opinion and see what others think about it. Application Users will be provided multiple choice questions. Based upon their personal opinion, they can mark their favourable option. Post submission, they will be allowed to see percentage opinion of others.

### Intended User

App is intended for all users that knows how to operate a phone.

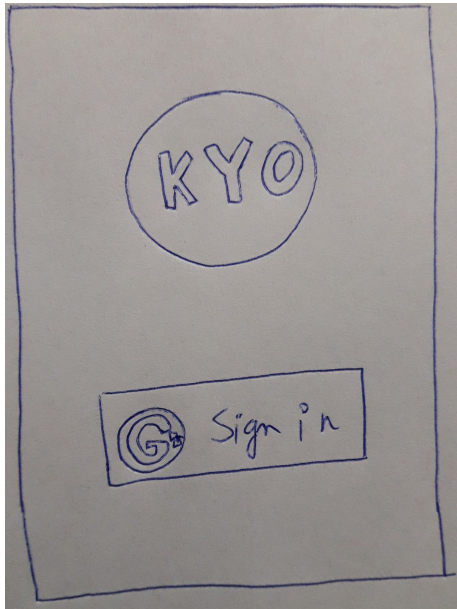
## Features

- Provides Information regarding Global Opinion.
- Provides Real-time data.
- Useful for conducting surveys.

## User Interface Mocks

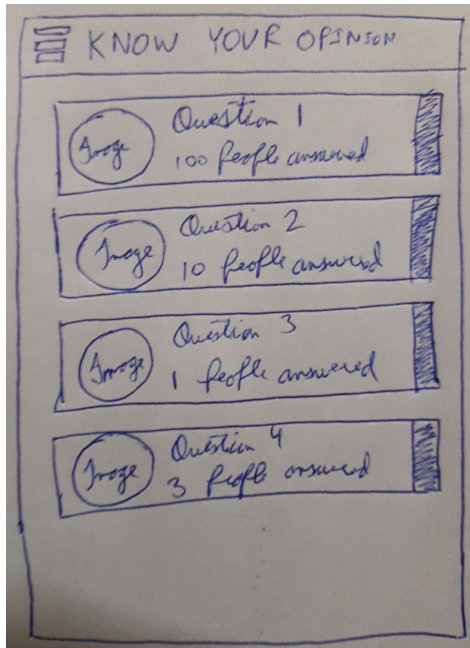
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Welcome Screen



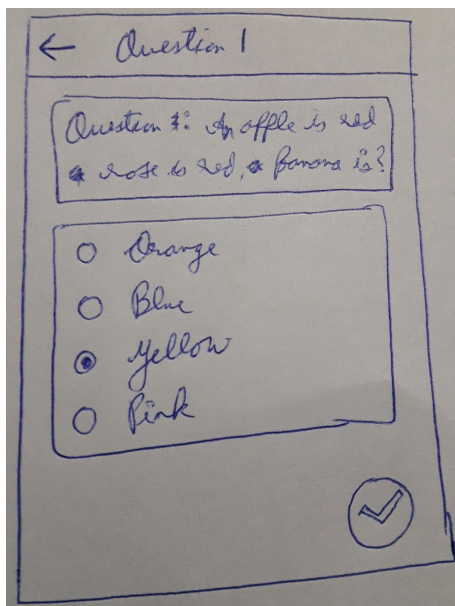
The welcome screen will provide with a Google Plus sign in button and app icon. User has to authenticate himself before moving further.

## HomeFragment in MainActivity

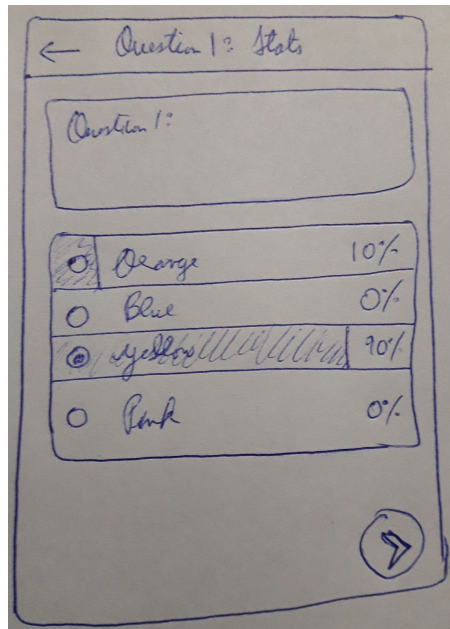


HomeFragment with a list of questions. Content of the list is an image related to question, question itself and no of responses recorded.

## QuestionActivity

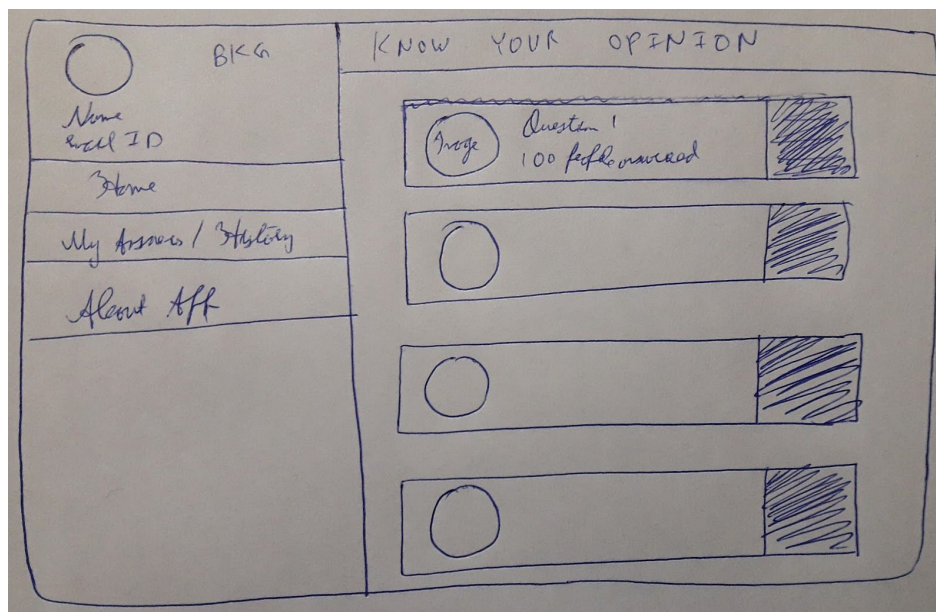


QuestionActivity contains the question and its possible answers. On selecting an answer, Floating Action Button will be made visible to continue the flow.



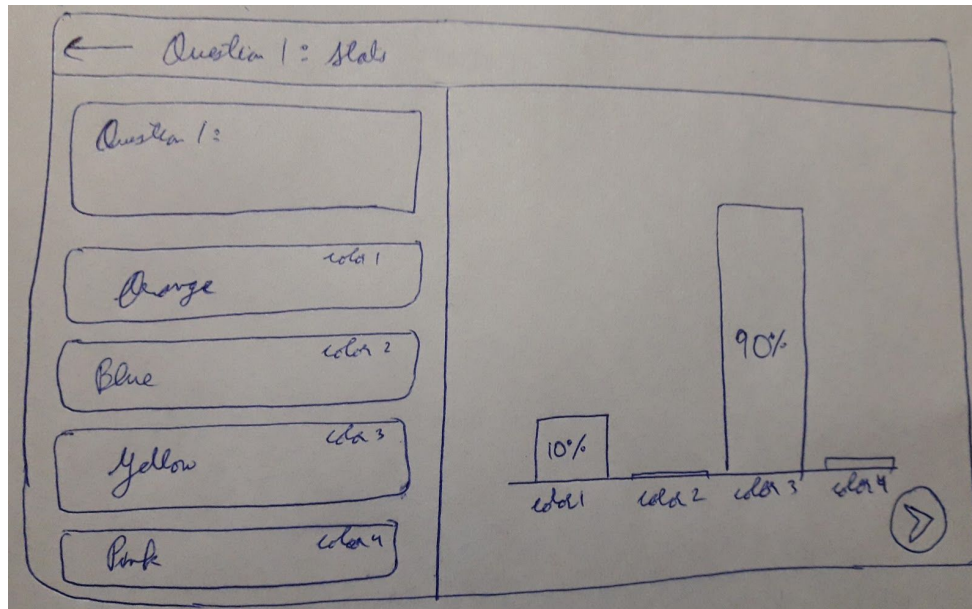
On Submission, the options will be updated with overall percentage response of the question.

### HomeFragment in MainActivity (Tablet)



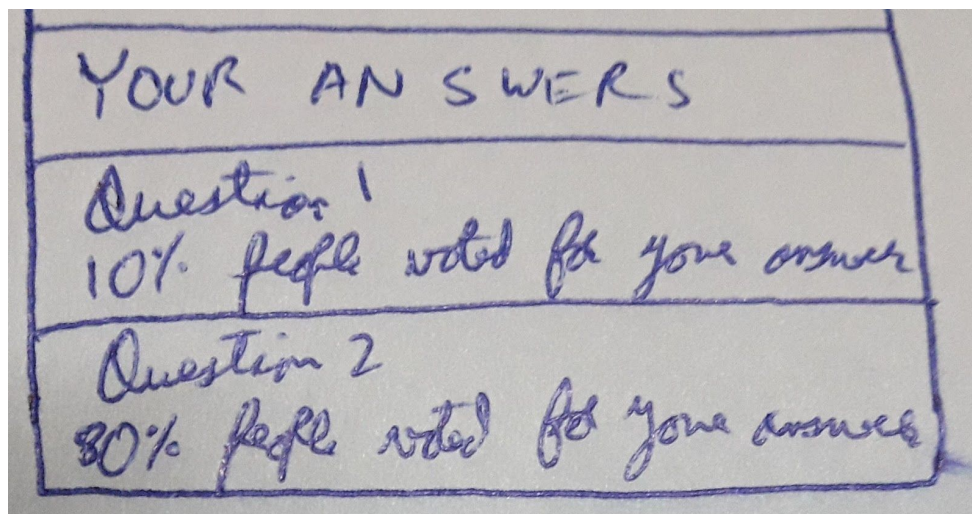
Content of navigation drawer being replaced by a separate fragment.

## QuestionActivity (Tablet)



Extra space is utilized by graphical representation of percentage.

## Widget UI



The widget will show questions those were answered. Locally saved data is used for this.

## Key Considerations



### How will your app handle data persistence?

For keeping/viewing historical data, content providers is used. However for unanswered questions, data won't be cached so as to take advantage of firebase real-time database.

### Describe any corner cases in the UX.

User may see the question however may not mark his answer. For overcoming this issue a FAB is used for confirmation.

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

Picasso/Glide to handle the loading and caching of images.

### Describe how you will implement Google Play Services.

1. Using firebase for Google+ Sign In and Database.
2. Admob ads for interstitial ads.

## Required Tasks

### Task 1: Firebase Setup

- Create firebase project.
- Create real-time database.
- Enable Google Plus Sign In.

### Task 2: Project Setup

- Creating studio project.
- Integrating firebase to Android Application.
- Adding Project Dependencies.

### Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI QuestionActivity

- Build UI for MainActivity Fragments.
- Build Multipane UI for tablets.
- Adding RecyclerView and AppBar.

#### **Task 4: Implement Google Play Services**

- Adding firebase auth and database.
- Implement Admob Interstitial Ads.

#### **Task 5: Implement Content Providers and Loaders**

- Create Local SQLite database and helper classes.
- Create content provider for keeping answered questions locally.
- Implement Loaders in HistoryFragment.
- Implement SyncAdapter for updating Local Data periodically.