How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

GitHub Username: StefMa

Choicm

Description

Don’t know what do you want to eat today?
Want to know what the world think will be the next soccer world champion?

It doesn’t matter what you ask! It matters what the world think.
With Choicm you can simply ask questions, provide default answers and let other people around the world decide what the best answer will be.
Example:
Q: Who will be the next soccer world champion?
A: Germany

A: France
A: England
A: Someone else

People around the world can answer by check one of the provided answers and you will immediately know what the world think.

Don’t be shy and ask everything. Feel free to ask if you should eat just a sandwich today or a salad. Be surprised what the world will choose!

Choicm isn’t limited to ask questions. You can of course answer others questions too. Browse over millions of questions and share your opinion with the world. Stay organized by searching for questions or filter them by category.

Intended User

The App is for everyone. It is just kind of fun to ask questions and see how people around the world would answer these.

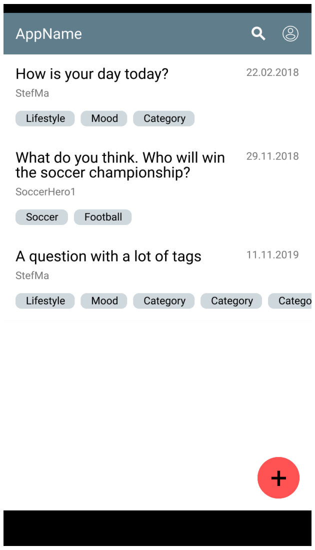
Features

- Will be written in pure Java
- All Strings will be located at the strings.xml and support RTL

- Ask questions
- Answer questions by checking one of the provided answers (multiple choice)
- Search/Filter question
- Create an Account

User Interface Mocks

Screen 1



AppName

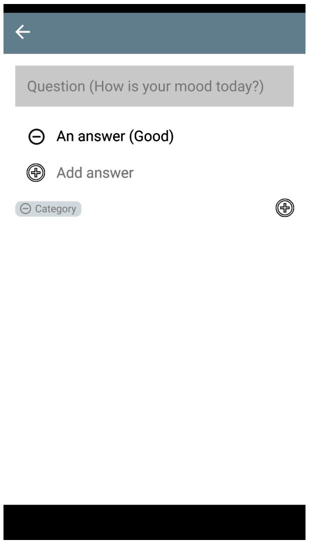
How is your day today? 22.02.2018
StefMa
Lifestyle Mood Category

What do you think. Who will win the soccer championship? 29.11.2018
SoccerHero1
Soccer Football

A question with a lot of tags 11.11.2019
StefMa
Lifestyle Mood Category Category Catego

The App “Overview”. You will see this Screen if you enter the App.
You will see a list of questions together with the question categories, the user who have asked the question and the date.
While scrolling the FAB will disappear.
If you click on the FAB (and logged in) you will go to the “Create question screen” (see below).
The magnifier will start the search (see screen below).
The account icon will start the login (see screen below) or the account screen.
The account screen is not designed but it will just display some information about the user.

Screen 2



Question (How is your mood today?)

⊖ An answer (Good)

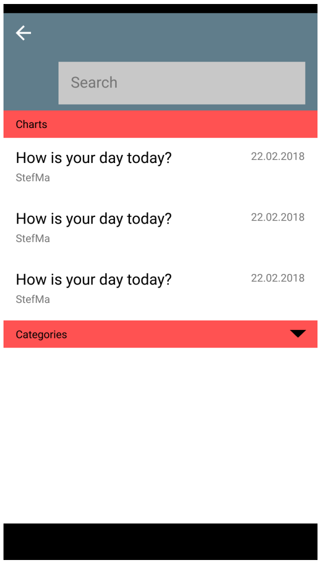
⊕ Add answer

⊖ Category ⊕

In the “Add question screen” you can create a new question.

You can enter the question and provide as many answers as you like.
You can add categories and create new categories.
The “add question” button is missing here but will be in the final product of course.

Screen 3



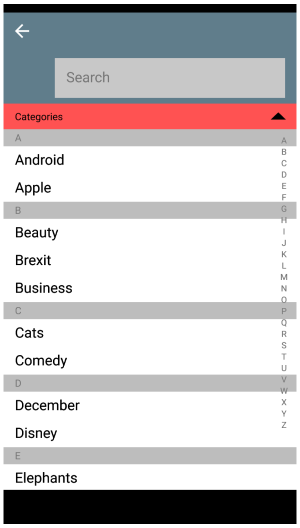
How is your day today? 22.02.2018
StefMa

How is your day today? 22.02.2018
StefMa

How is your day today? 22.02.2018
StefMa

In the Search screen you can either search by some string or by categories (see screen below). This will also display some “chart questions” which are very popular currently.

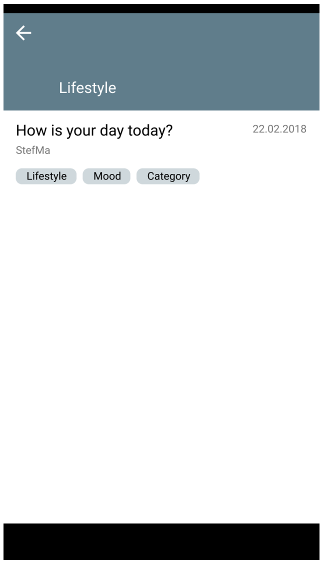
Screen 4



A
Android
Apple
B
Beauty
Brexit
Business
C
Cats
Comedy
D
December
Disney
E
Elephants

If you open the Categories in the search screen the categories got expanded and you can click on a category to see all questions which belongs to these category (see screen below).

Screen 5



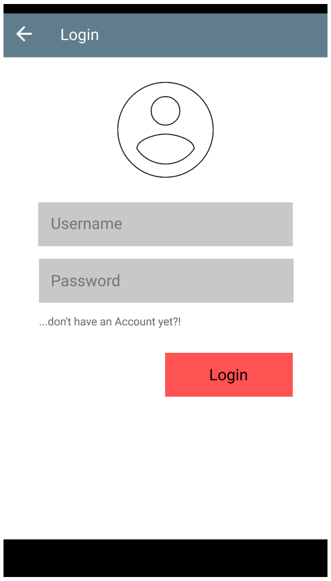
How is your day today?

22.02.2018

StefMa

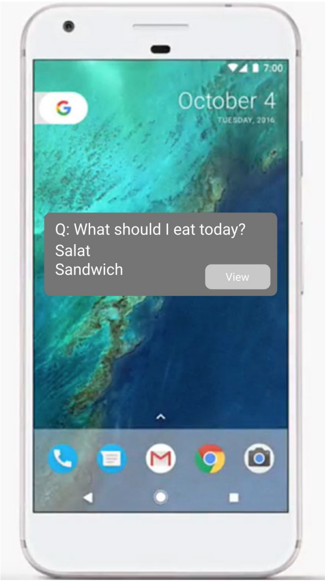
If you click on a category you will see this screen. This will display all the questions which belongs to a category.

Screen 6



If you want to add a question and are not logged or you click on the account icon (while not logged in) you will go to this screen. You can login into the app at this screen.
If you don't have an account yet you can create a new one.

Screen 7



Q: What should I eat today?
Salat
Sandwich

The Widget shows each 30 minutes a new question with some answers.
The user can click on the widget (or the button) to view the question and answer them.
To update the Widget I'll implement a IntentService which will fetch the data to display them.

Key Considerations

How will your app handle data persistence?

I will use Firebase Realtime Database for persistence.

Describe any edge or corner cases in the UX.

One edge case could be if an user creates **a lot of** answers. Imagine you have a question like "How would be the winner of the soccer world championship?". They must provide 32 possible answers. That may look bad. For now I would go with that by adding a ScrollView to the UI. But maybe I can improve the UI better **or** limit the possible answers in the future.

Another edge case could be if the user opens the AnswerActivity via the Widget. What happen if the user press the back button? Should the App close? Should the App go back and start the Main(List)Activity?
I think I will apply the default navigation here. Means the App should close if the user press back. If the user press the up button the Main(List)Activity should start.

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

Library/Tool	Version
Android Studio	3.1.3
Gradle	4.8.1
AppCompat-v7, Design, CardView, RecyclerView…*	27.1.1
Firebase Realtime Database, Analytics, Auth	16.0.1
RxJava	2.1.16
Timber	4.7.1
ThirtInch	0.8.5

* I’ll include “as many as needed” Android-support libraries.

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

I will implement Firebase Realtime Database to save the questions/answers.
I will use Firebase Authentication to implement the login/registration. It is only planned to implement the Mail/Password login. The other logins (like G+, Facebook or Twitter) aren’t planned to integrate!
I’ll also implement Firebase Analytics to track some default user interactions within the app.

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

To setup the project I’ll do the following steps:

- Create a new AS project

- Add the basic libraries I'll use (Android support, RxJava, Firebase Auth/Database/Analytics etc)

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main(List)Activity
- Build UI for the Login/Registration Activity
- Build UI for the “Add questions” Activity
- Build UI for the “See questions and answer them” Activity
- Build UI for the Search Activity

Task 3: Implement Basics for the Main(List)Activity

- Create the UI for the Main(List)Activity
  - Toolbar (with their Actions)
  - Create the RecyclerView with their Items
  - Add FAB and scrolling behaviour for them
- No other services/real logic will be implemented in this step.

Task 4: Implement Login/Registration

- Create the UI for the Login/Registration Activity
  - Toolbar
  - Other views
- Implementation Firebase Authentication to register/login Users

Task 5: Implementation Add Questions

- Create the UI for the Add Questions Activity
  - Toolbar
  - Question EditText
  - Answer-CustomView
- Implement Firebase Realtime Database to save the questions (and answers)
- No implementation of the categories yet

Task 6: Update Main(List)Activity

- Implement the Firebase Realtime Database in the Main(List)Activity to display real questions from the “backend”
- Implement that a click on an item will open the detail page

Task 7: Implementation Answer Question

- Create the UI for the Answer Questions Activity
  - Toolbar
  - Question TextView
  - Answer views
- User can pick an answer which will be saved in the Realtime Database

Task 8: Implementation Widget

- Create the UI for the Widget
- Display a random question each 30 minutes which opens the question directly

Optional: Task 9 - Task 11

Task 9 : Implementation Search

- Create the UI for the Search Activity
  - Toolbar
  - Display charts
  - No categories yet!
- Let users search for a given string

Task 10: Implementation Categories

- Implement the categories in the Main(List)Activity
- Implement the categories in the AskQuestionActivity
- Implement the categories in the AnswerQuestionActivity
- Implement the categories in the Search

Task 11: Implementation Analytics

- Implement Firebase Analytics on all places where it make sense

Add as many tasks as you need to complete your app.

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”