

Index for QuizTime

[Index for QuizTime](#)

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Screen 7](#)

[Key Considerations](#)

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

[Describe any edge or 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 or other external services.](#)

[How you will support accessibility?](#)

[How resources will be stored in the project including colors, strings, and themes?](#)

[How the application implements one or more of the following SyncAdapter/JobDispatcher or IntentService or AsyncTask for backend communication?](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Code the Adapter for trending and saved recyclerview.](#)

[Task 4: Create a new activity called quiz-activity.](#)

[Task 5: Firebase Auth](#)

[Task 6: Insert Quiz](#)

[Task 7: Create Widget](#)

[REST Server Endpoints](#)

QuizTime

Description

Create, Share and Test your knowledge by attempting quizzes created by other users. Showcase your knowledge by creating quizzes.

Intended User

Intended for everyone, from students to friends and family for taking a trivia quiz.

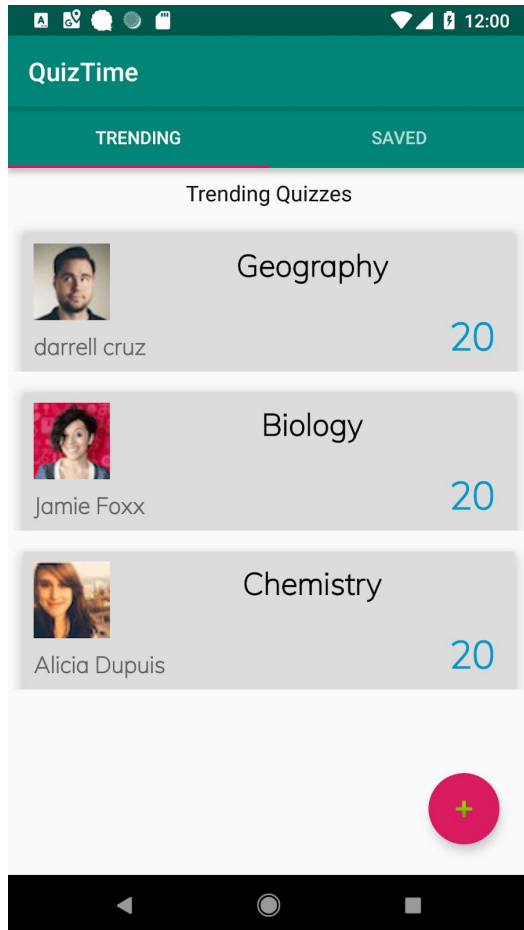
Features

List the main features of your app. For example:

- Saves Quiz for offline use
- Create own Quiz
- Preview the Quiz
- Publish the Quiz
- Share the Quiz
- Attempt the Quiz multiple time till they are thorough

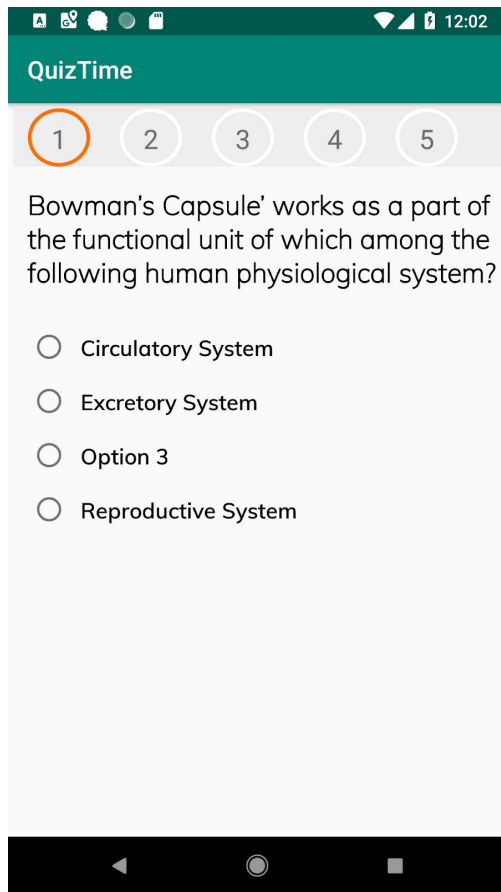
User Interface Mocks

Screen 1



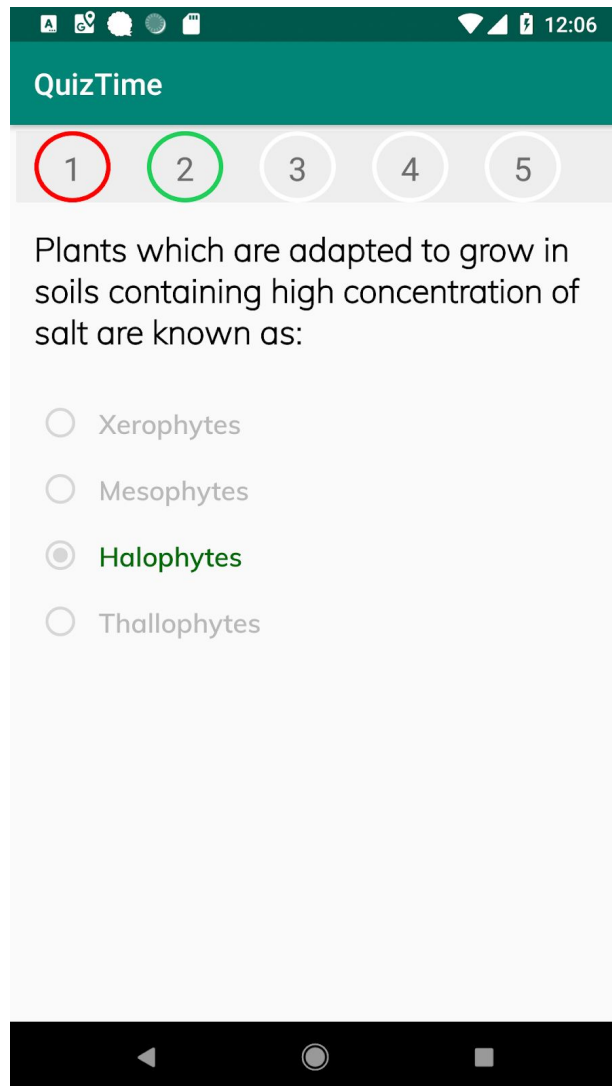
The main landing screen of the app. It shows two tabs Trending and Saved. Below tabs it displays the list of quizzes which are currently trending. A Floating Action Button is also seen which is used to create a quiz. More on it in SCREEN 5.

Screen 2



When clicked on any of the quiz on SCREEN 1 it shows the quiz. The quiz gets saved to room database automatically and is then available under Saved tab of Screen 1 The tab items on top display the number of questions in the quiz. When you click on that number, the respective question is shown. When clicked on the radio button, it changes color. Green if correct, Red if wrong. The tab also changes color according to the answer. More on it in Screen 3.

Screen 3



The screenshot shows a mobile application interface for a quiz. At the top, there is a status bar with icons for location, signal, and battery, and the time 12:06. Below this is a teal header bar with the text "QuizTime". Under the header, there is a row of five circular tabs numbered 1 to 5. Tab 1 is outlined in red, and tab 2 is outlined in green. Below the tabs, the question text reads: "Plants which are adapted to grow in soils containing high concentration of salt are known as:". Below the question, there are four radio button options: "Xerophytes", "Mesophytes", "Halophytes", and "Thallophytes". The "Halophytes" option is selected, indicated by a green dot in the center of the radio button and the text "Halophytes" being green. At the bottom of the screen, there is a black navigation bar with three icons: a back arrow, a home circle, and a recent apps square.

QuizTime

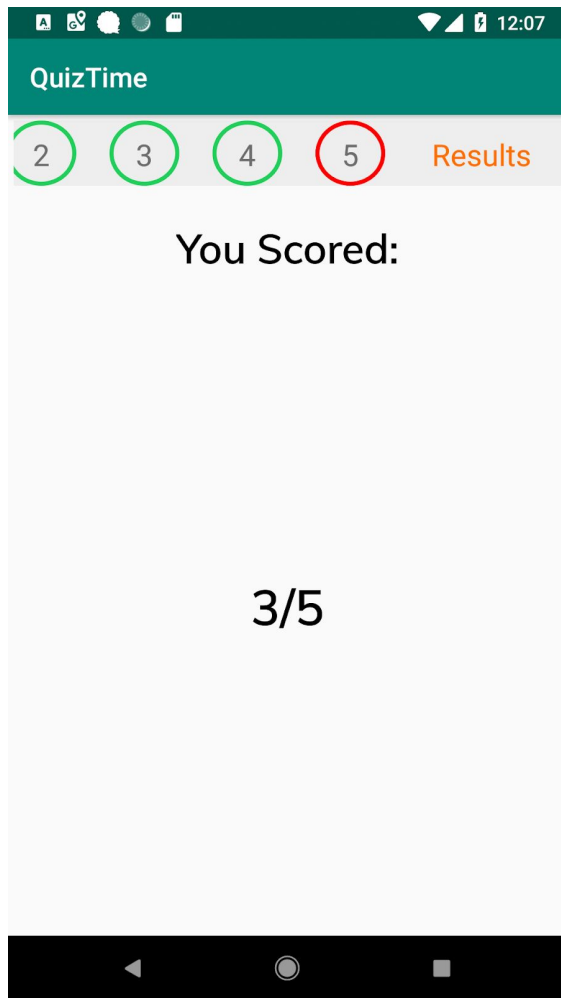
1 2 3 4 5

Plants which are adapted to grow in soils containing high concentration of salt are known as:

- ☐ Xerophytes
- ☐ Mesophytes
- ☒ Halophytes
- ☐ Thallophytes

When Clicked on Any of the option, if the option is correct,the radio button and tab with question number turns green.

Screen 4



The last tab shows the Results. Note the color change of the tab which displays question number.

Screen 5

The screenshot shows the 'QuizTime' app interface. At the top is a dark green header with the app name 'QuizTime' and a floating action button (FAB) with an upward arrow. Below the header, the 'Title of Quiz' is 'General Knowledge'. The 'Quiz Question' is 'What is the Capital of India?'. There are four options: 'Option A: Delhi', 'Option B: Mumbai', 'Option C: Agra', and 'Option D: Chennai'. The 'Select Correct Answer' dropdown is set to 'B'. At the bottom, there is a green button labeled 'ADD ANOTHER' and a light green bar at the very bottom with the text 'PREVIEW - General Knowledge'. The Android status bar at the top shows the time as 12:01 and various icons.

QuizTime

Title of Quiz

General Knowledge

Quiz Question

What is the Capital of India?

Option A

Delhi

Option B

Mumbai

Option C

Agra

Option D

Chennai

Select Correct Answer

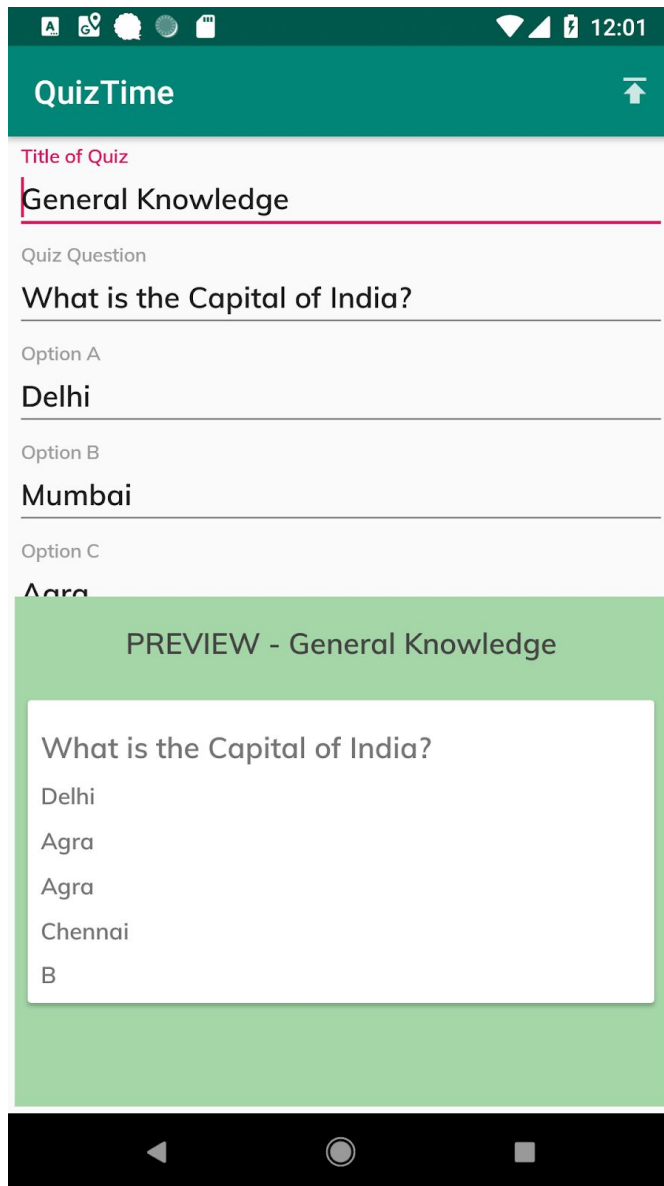
B

ADD ANOTHER

PREVIEW - General Knowledge

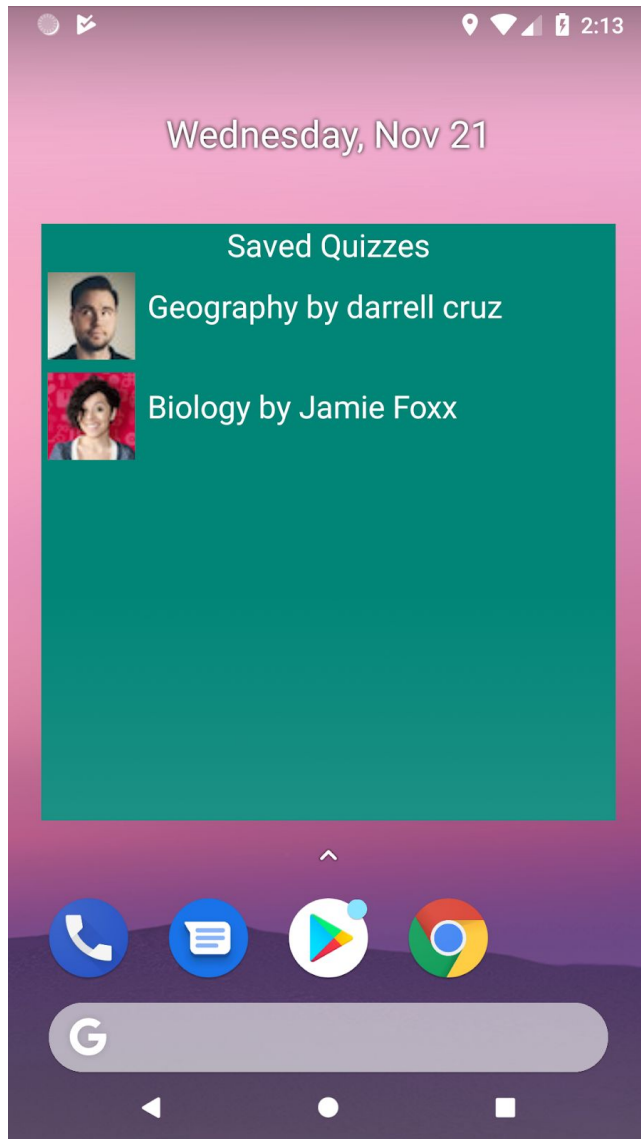
When the FAB is pressed on Screen 1, it animates to this page. This page is used to publish the quiz. After entering all the details press Add another, the Bottom Tab changes text and title of the quiz is shown.

Screen 6



When the Bottom sheet is pulled up, it displays the current quiz. These cards are swipped to be removed. When the user is satisfied, using the menu option, the quiz gets published and a dialog to share the quiz is shown. When share is pressed, normal text with link is shared.

Screen 7



Widget must show the saved quizzes along with the author and image of the author.

Key Considerations

How will your app handle data persistence?

On device data persistence will be handled by Room Database.

Describe any edge or corner cases in the UX.

- There is no edit quiz option when the user tries to add a quiz. The user will have to remove the quiz and re-enter all the details.
- Unstable or missed network connection: the application must not crash in that cases
- Device orientation change: the application must handle all long-running operations correctly considering possible configuration changes
- UI freezes: the application must not use the main thread for any resource consuming operations

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

1. Picasso - 2.71828 - For image loading and caching,
2. Retrofit - 2.4.0 - For network queries
3. Firebase Auth - 16.0.5 - For Authentication
4. Firebase Dynamic Links - 16.1.3 - For sharing links
5. Android Architecture Components(Room, LiveData, Viewmodel, Paging, etc)
6. Gradle - gradle-4.6
7. Gradle Android Build tools - com.android.tools.build:gradle:-3.2.1
8. Androidx.appcompat:appcompat:-1.0.2
9. Androidx.constraintlayout:constraintlayout:-1.1.3
10. Androidx.recyclerview:recyclerview:-1.0.0
11. Androidx.legacy:legacy-support-v4:-1.0.0
12. Androidx.cardview:cardview:-1.0.0
13. Androidx.room-2.0.0-rc01
14. com.google.android.material:material:1.1.0-alpha01

Android Studio 3.2.1

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

Link the app to firebase via android studio, tools -> Firebase -> Connect to firebase.

How you will support accessibility?

By adding contentDescription in all the relevant places to help the user navigate through the app. Using Android lint to check for any missing content description. Along with it proper color use with enough contrast to make them visible.

How resources will be stored in the project including colors, strings, and themes?

Colors will be stored in colors.xml
Strings will be stored in strings.xml
Themes will be stored in styles.xml

How the application implements one or more of the following SyncAdapter/JobDispatcher or IntentService or AsyncTask for backend communication?

IntentService will be used for sending the Auth data to the server.

The Android application will be written solely in the Java Programming Language.

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

Create a new Project in Android Studio with an empty activity called Main Activity

Create 3 fragments:

- Trending - to display trending quizzes
- Saved - to display saved quizzes

In main activity add the two fragments to the tab. Create a view pager adapter and link it to show the two tabs.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity with viewpager and tablayout
- Build UI for trending fragment and model
- Build UI for saved fragment and model
- Build UI for individual item to be displayed inside the recyclerview of these two fragments.

Task 3: Code the Adapter for trending and saved recyclerview.

Implement Paging Library from AAC and link it to the trending fragment recyclerview.
Implement Room Database to show all the saved quizzes.

Make sure to start a new activity when any of the item in the recyclerview is clicked passing the quiz-id as intent.

Task 4: Create a new activity called quiz-activity.

Implement deep-link as this will be the activity which gets displayed by Dynamic Links.

Once created complete the below steps:

- Create fragment Quiz Fragment - Create a viewpager to display the quiz, along with change the color of the tabs which are shown in screen 3 and screen 4, This fragment fetches the quiz from the network.
- Page Fragment - which actually displays the individual quiz data in the viewpager. This fragment is attached to the viewpager of the Quiz Fragment,
- Complete the UI for the activity and both the fragment.

Task 5: Firebase Auth

Use firebase authentication to get the user profile and image and send it to server. Add all the relevant methods to the MainActivity for this. When the fab is clicked if the user is logged in grant him/her access to the Insert Quiz Section else ask for authentication/signup. Send the data got from firebase auth to the server. IntentService will be used to send the Auth data to the server.

Task 6: Insert Quiz

Create a fragment which will be used to publish the quiz. It should contain a Bottom Sheet for previewing the quiz before publishing. This fragment send the quiz to the server via network. A spinner must be used to select the correct answer. This fragment also handles the dynamic link generation, shortening and sharing to others.

Task 7: Create Widget

Create a widget which should show all the saved quiz in gridview and must mention the name and author of the quiz.

REST Server Endpoints

BASE URL = <https://comsec.co.in>

FETCH ALL QUIZ (GET REQUEST)

BASE_URL + /quiz/

Accepted/Required Parameters: **page**

FETCH ONE QUIZ (GET REQUEST)

BASE_URL + /quiz/

Accepted/Required Parameters: **t**

SEND QUIZ (POST REQUEST)

BASE_URL + /quiz/index2.php

Accepted/Required Parameters: **title**

SEND USER DATA (POST REQUEST)

BASE_URL + /quiz/loguser.php

Accepted/Required Parameters: **name, email, photourl, emailVerified, uid**
