

Tech Intellect

Team Members

Smit Saraiya (B00811636, sm252977@dal.ca)

Ravi Zala (B00805073, rv901764@dal.ca)

Yash Modi (B00799125, ymodi@dal.ca)

Haritha Desikan (B00801722, haritha.desikan@dal.ca)

Aravind Govindarajan (B00803336, aravind.govindarajan@dal.ca)

The project is available at the following location: [Tech Intellect](#)

Project Summary

Tech Intellect is a technical quiz application that tests the users on some interesting topics like Natural Language Processing, Computer Science Acronyms, R Programming, and Google go. The quiz application has three levels of difficulty namely easy, medium and hard for the users to select according to their expertise in that particular topic. The app delivers the quiz to users in two different modes namely Timed and Endless. Timed mode forces users to finish the quiz within a stipulated time depending on the difficulty level and Endless mode allows the users to take the quiz and learn on their own pace.

Libraries

Following are the list of external libraries used for this project:

Better Spinner : Library for creating materialistic drop-down spinner. Source [here](#)

Firebase database library: This library helps to interact with firebase database . Data is synced across all clients in realtime, and remains available even when the app goes offline. Data is stored as JSON and synchronized in realtime to every connected client.

Firebase auth library: Library provides authentication of user with Firebase. When user sign into app, it authenticates user using credentials. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Backend services of Firebase will then verify those credentials and return a response to the client. After a successful sign in, one can access the user's basic profile information, and control the user access to the data stored in other Firebase products.

Installation Notes

1. Install Android Studio (<https://developer.android.com/studio/>)
2. Download or clone (<https://git.cs.dal.ca/modi/Tech-Intellect.git>) our repository from the GitLab
3. Select "Open existing project" option in android studio
4. Select our directory to open the project

5. Provide Java SDK to run the application
6. If you want to run the application in a virtual device then create any virtual device with API level 23 or higher and select it as the deployment target. Alternatively, you can also deploy the application in any connected Android device.

Code Examples

While developing the project, we faced quite a few setbacks. But as a team, we worked together to get the optimal solution for all the problems.

Problem 1: We needed a method to check if the device is connected to internet

Since our quiz applications requires network connection, we had to check if the user has an active internet connection while using the app. This was essential because our app kept crashing when there is no active connection. We were then able to solve the code by implementing the below code snippet.

```
//Method to check if there is active internet connection
public boolean isConnectedToInternet(){
    ConnectivityManager connectivity =
    (ConnectivityManager)getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);
    if (connectivity != null)
    {
        NetworkInfo[] info = connectivity.getAllNetworkInfo();
        if (info != null)
            for (int i = 0; i < info.length; i++)
                if (info[i].getState() == NetworkInfo.State.CONNECTED)
                {
                    return true;
                }
    }
    return false;
}
```

Problem 2: We found it difficult to implement the dynamic allocation of card layout in dashboard.

The problem we faced was that the cards were overwritten instead of getting appended to the previous card. The following code snippet solved the problem.

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View root = inflater.inflate(R.layout.fragment_tab2, container, false);
    RecyclerView recyclerView = (RecyclerView)root.findViewById(R.id.recyclerview);
    recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));

    //According to listsize cards will be formed dynamically
    recyclerView.setAdapter(new RecyclerViewAdapter(getContext(),Tab1.flist));
    return root;
}
```

Problem 3: Disabling dashboard functionality for guest users using intent was quite difficult.

We have used a tabbed interface for the home screen. We have dashboard in one of the tabbed screens. For the guest users, we are supposed to disable the dashboard tab. Disabling a particular tab alone was bit difficult. Later we came up with the below code that helped us to overcome this problem.

```
private Button btnGuest;

//Guest flag is false in the beginning
public static boolean signInGuestflag = false;
btnGuest.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent intent = new Intent(getApplicationContext(), HomeScreen.class);
        signInGuestflag = true;
        startActivity(intent);

    }
});
if(!MainActivity.signInGuestflag)
{
    tabLayout.addTab(tabLayout.newTab().setText("Dashboard"));
    MainActivity.signInGuestflag=false;
}
}
```

Feature Section

The main features of the application are listed below.

Registration: Allows new users to either register with the quiz application by providing their name, email address, and password.

Login: Allows existing users to log into the quiz application using their email address and password which was created at the time of registration.

Guest login: Users who don't wish to register with the application, but would like to get a flavour of the app, can use this feature to login as a guest and take the quiz.

Difficulty levels: Tests the users on the expertise level they have opted. The quiz application supports three level of difficulties, namely easy, medium and hard.

Timed mode: Forces the users to finish the quiz within a stipulated time depending on the difficulty level.

Endless mode: Allows the users to take the quiz and learn at their own phase.

Dashboard: Enables the users to visit their past learning history. Once a quiz is successfully completed, the corresponding quiz will be created dynamically under the dashboard tab. On clicking it, users can go through the questions and answers that was asked in the quiz. Note that this feature is not available for guest users.

Buzzer and vibration: The app has buzzer sounds for wrong and right answers. This can be controlled under the settings menu. Also, the app vibrates for every wrong answer. Just like the buzzer sounds, vibration can also be turned on/off in the settings menu.

Forgot password: Helps users to reset their password. The app sends an email to the users' registered email address. The link provided in the email can be used to set the new password.

Change password: Helps users to change their old password and set a new password.

Final Project Status

We have successfully implemented the major functionalities of our application.

The next step of our project is to implement a repetition algorithm that would repeat the wrong questions during the quiz. This can help the users to rectify their mistakes and learn from it. We also look forward to add more questions for the topics provided.

Minimum Functionality

A learning quiz application with login and registration module. The application will allow users to gain knowledge on interesting topics like computer science acronyms, natural language processing, Google Go and R programming language *[Completed]*.

Expected Functionality

1. *Guest login* feature for the users who don't wish to register with the application, but would like to get a flavour of the app *[Completed]*.
2. The quiz application will have *three levels of difficulty* namely *easy, medium and hard* for the users to select according to their expertise in that particular topic *[Completed]*.
3. The application will deliver the quiz to users in *two different modes* namely *Timed and Endless*. Timed mode will force users to finish the quiz within a stipulated time depending on the difficulty level and Endless mode will let users to take the quiz and learn on their own pace *[Completed]*.
4. The *dashboard tab* on the home screen of the application will display the user's learning history. Guest users will not be able to use the dashboard feature *[Completed 90%]*.
5. The application will alert the users with a *vibration* when a wrong answer has been submitted. This vibration feature can be configured by the user in the Settings menu *[Completed]*.
6. The app will play *buzzer sounds* for right and wrong answers. The buzzer feature can be configured by the user in the Settings menu *[Completed]*.

Bonus Functionality

1. Implementation of *change password* and *forgot password* functionalities for user convenience *[Completed]*.
2. The application will repeat the wrongly answered questions to users by using a repetition algorithm. In this way, users can learn from their mistakes easily *[Not implemented]*.

Sources

[1] "MediaPlayer overview | Android Developers", *Android Developers*, 2018. [Online]. Available: <https://developer.android.com/guide/topics/media/mediaplayer#java>.

[2] G. Training, "9.2: Adding Settings to an App · Android Developer Fundamentals Course – Practicals", *Google-developer-training.gitbooks.io*, 2018. [Online]. Available: https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-practicals/content/en/Unit%204/92_p_adding_settings_to_an_app.html.

- [3] H. Android?, "How to display a Yes/No dialog box on Android?", *Stack Overflow*, 2018. [Online]. Available: <https://stackoverflow.com/questions/2478517/how-to-display-a-yes-no-dialog-box-on-android>.
- [4] "List of computing and IT abbreviations", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/List_of_computing_and_IT_abbreviations.
- [5] M. Gupt, "Android Tabs Example - With Fragments and ViewPager - Truicon", *Truicon*, 2018. [Online]. Available: <https://www.truicon.com/2015/06/android-tabs-example-fragments-viewpager/>.
- [6] a. app, "android pressing back button should exit the app", *Stack Overflow*, 2018. [Online]. Available: <https://stackoverflow.com/questions/2354336/android-pressing-back-button-should-exit-the-app>.
- [7] "R Programming Multiple Choice Questions and Answers - Sanfoundry", *Sanfoundry*, 2018. [Online]. Available: <https://www.sanfoundry.com/r-programming-multiple-choice-questions-answers/>.
- [8] "Natural Language Processing - AI Questions and Answers - Sanfoundry", *Sanfoundry*, 2018. [Online]. Available: <https://www.sanfoundry.com/artificial-intelligence-mcqs-natural-language-processing-1/>.
- [9] "Frequently Asked Questions (FAQ) - The Go Programming Language", *Golang.org*, 2018. [Online]. Available: <https://golang.org/doc/faq>.
- [10] "Go Tutorial", *www.tutorialspoint.com*, 2018. [Online]. Available: <https://www.tutorialspoint.com/go/index.htm>.
- [11] "Create a Card-Based Layout | Android Developers", *Android Developers*, 2018. [Online]. Available: <https://developer.android.com/guide/topics/ui/layout/cardview>.
- [12] A. Hathibelagal, "Getting Started With RecyclerView and CardView on Android", *Code Envato Tuts+*, 2018. [Online]. Available: <https://code.tutsplus.com/tutorials/getting-started-with-recyclerview-and-cardview-on-android--cms-23465>.
- [13] "Create a List with RecyclerView | Android Developers", *Android Developers*, 2018. [Online]. Available: <https://developer.android.com/guide/topics/ui/layout/recyclerview>.
- [14] "Vibrator | Android Developers", *Android Developers*, 2018. [Online]. Available: <https://developer.android.com/reference/android/os/Vibrator>.