

National College of Ireland
BSc in Computing
2016/2017

Jamie O'Neill
x13437532
jamie.oneill@student.ncirl.ie

Now You're Talking

A mobile application to aid in learning foreign languages
through social interaction and gamification

Supervisor: Dominic Carr



Contents

Executive Summary	3
Introduction.....	5
Background	5
Aims	5
Technologies	6
System	8
Functional requirements.....	8
Data requirements	9
User requirements	11
Use Cases.....	12
Requirement 1: Add Meet up	12
Requirement 2: Join Meet up	14
Fully functional use case	15
Non-Functional Requirements	16
Performance/Response time requirement	16
Availability requirement	16
Disaster Recovery requirement	16
Robustness requirement	16
Security requirement	16
Maintainability requirement	16
Portability requirement	16
Extendibility requirement.....	16
Reusability requirement	17
Resource utilization requirement.....	17
Design and Architecture.....	17
Implementation.....	18
Features and layout.....	23
Navigation Screens	23
Social Screen	23
Learn Screen	24
Profile Screen.....	25

Joining a Meet up	29
Adding and Managing Meet ups.....	31
Messaging	32
Lessons	33
Lesson Feedback	38
Testing.....	40
Evaluation.....	42
Testing feedback.....	42
Development lifecycle	43
Conclusions.....	44
Further development	45
Future features.....	45
Appendix.....	46
References.....	46
Research	46
In app References	46
Meet up search results	47
In app achievements.....	47
Bug Log	47
Project Plan	48
Project Proposal	48
Objectives	48
Background	49
Technical Approach	49
Special resources required	50
Technical Details	50
Evaluation	50
References	51
Monthly Journals.....	51

Executive Summary

Now You're Talking has two major functionalities. Firstly, it will help grow and develop language skills of the user keeping them engaged through the use of gamification. Secondly, the app promotes socializing by offering users a platform in which can schedule meet ups and communicate. This app covers a niche in the market where no other app covers allowing language learning and meet ups. A search on the Google play store for "Meet ups" will show only a handful of apps that allow users to meet and discuss their field of interest, with the vast majority of search results being dating app's (see [Meet up search results](#)). There is certainly room in the market for an application that covers the areas that I intend on covering for users to meet up in an educational environment.

The potential for this app is almost limitless in terms of features so maintaining what features should be demonstrated in this project is an important aspect to give an insight into the overall potential of the app.

This app will show the introductory lessons for German, Spanish and French for an English speaker. These languages have been chosen as they are four of the five most spoken languages in Europe (TNS Opinion & Social, 2012). If there is to be a full release of this application, it will be extended to fully teach all of the most popular languages.

This will be accomplished by creating the app for Android devices and using Google Firebase for backend services such as database hosting. There are several [functional requirements](#) set for this app which we will detail further in this report. The database will consist of 4 key tables handling meet ups both past and future, users, and messages. The app will also consist of an easy to use user interface that is intuitive to users. The app also makes use of API calls for certain services such as maps and translation features.

Testing, reporting and analytics will also be handled through Firebase, this provides an all in one solution to a lot critical information which can be used to improve the application.

Introduction

Background

There are millions of people across the globe learning second or even third languages. As of 2009, within the EU alone, 107 million students are attending an educational institution. 94% of these are learning at least one foreign language (Eurostat, 2015) meaning in this one demographic there is a massive market of potential users of the app without considering non-students and areas outside of Europe.

There has been a shift in the way people learn with the rise of personal technology over the past two decades, the availability of online courses and programs means it is easier than ever before to learn without the use of books. This is especially prevalent when studying a foreign language as stated by Salaberry (2001).

With the high number of people learning second languages and rise in availability of smartphones this influenced me in deciding to create an educational application.

The rise of social media and online forums have also influenced me in choosing to add social functionality. In today's world, so much of our interaction happens across social platforms that it seems somewhat natural at this point that we communicate not only in person but also to communicate on the go electronically should that be an app, website or messaging service.

Even with the rise of electronic communication in our life most people still prefer to speak in person to one another which brought me to the meet up functionality of my app which provides a way to communicate in person.

Aims

There are alternatives for learning a language, there are many applications and software for a range of platforms and devices that attempt to teach users a foreign language in a multitude of different ways. In my research, I have found that systems that use a progression and rewards system has proved more fruitful than traditional learning techniques (Pandey, 2015). As a matter of fact, some companies that have been in this market for years have begun to adapt this approach of teaching. Users are more engaged when they are having fun rather than reading from textbooks hence the gamification approach to the application. As Justin Marquis reports "People playing games also report losing track of time in the real world. This phenomenon is due to the extreme level of engagement that some games can promote in those playing them. This is a sharp contrast to the usual school situation where everyone, often including the teacher, is watching the clock, waiting for the school day to end." (Marquis Ph.D., 2012). From my research I have also found that many people would also like to practice speaking a new language with other new speakers as it is not as intimidating as speaking to someone more proficient in the language and is also viewed as a better way to learn

over practicing by one's self. For example, on meetup.com one of the French learning groups in Dublin alone has over 3000 members to date who schedule regular meet ups. This is what drove me to include socializing as a key aspect of my project.

Technologies

I have decided to develop my application using android studio as it is an easy to use IDE with useful GUI screen when designing an activity view (Figure 1). It was the preview screen, plugin support and my previous experience that led me to choose Android studio over Eclipse or IntelliJ IDEA. It also provides a mobile phone and tablet emulator that can be used for testing an application of multiple devices. As well as that it has built in integration with GitHub allowing me to easily manage version control of the project.

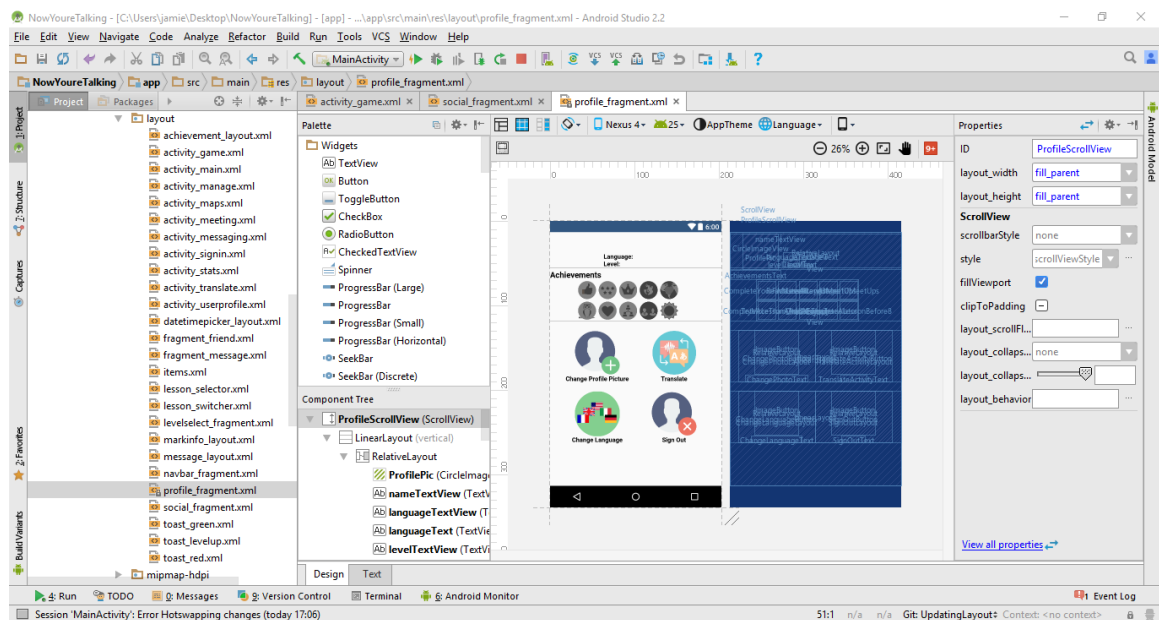


Figure 1: Screen shot of the Android Studio IDE

My app will feature a translation feature to allow users to translate from any of the 4 languages that my app allows. This will be handled by Microsoft's translator API this supports 71 languages allowing for expansion of my app and allows my app to translate 2 million characters a month free of charge. In mass production, this would have to be increased but for a proof of concept is more than enough.

Other options include Google Translate API, Apertium API and LetsMT API. Microsoft was chosen for its pricing and range of languages.

I will be using Google Firebase in my project.

Firebase is a mobile and web application development platform that offers a cloud database, storage, hosting, test labs, online functions, notifications and several other features. There are 3 tiers of pricing. These tiers offer more storage, automatic backups, and higher network traffic depending on the chosen tier. I will be using the free pricing tier as it suits all the needs of my project.

I will be using it for user authentication and profile building, cloud database, cloud storage and in app messaging service. It also allows for data analytics to view certain information that the application owner may want to use to improve their application such as demographic using the application, number of in app purchases and user engagement. Firebase also offers a notification system, crash reporting and test labs among other services that I will be utilizing detailed further in this report.

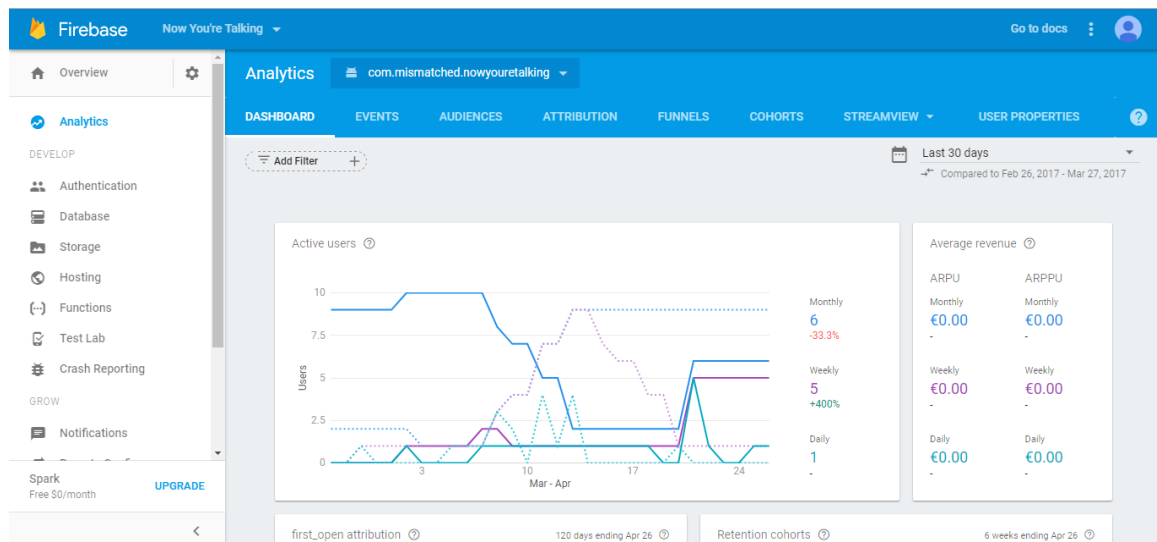


Figure 2: Screen shot of Firebase web console

System

Functional requirements

1. Add Meet up:

All users should have the ability to create a meet up for their selected language of choice. They should also be able to define the location and parameters of the meet up such as date, number of attendees and recommended ability level.

2. Join Meet up:

The user should be able to view meet ups around their physical location on a map starting with a 40km surrounding area and a zoom button bringing the camera to street level. The user can then join multiple meet up's being stored in an online database.

3. Manage Meet up's:

The user should be able to view the details of all meet ups they are currently scheduled to attend and have the function to remove themselves from a meet up or cancel a meet up that they are hosting.

4. Friends system:

Users should have the ability to add one another via their email address which they will then allow them to message one another. The email address system is in place to ensure that there can be no unsolicited friend requests.

5. In App Messaging:

Users should be able to message other users in their friends list. This is to keep in touch outside of meet ups or practice speaking their new language with each other.

6. Learning:

The app will provide the user a platform in which to learn their selected language through the use of gamification.

7. Progression & Achievements:

The user will receive a higher ranking the more they succeed. This ranking will be proof of their accomplishments in the learning sections unlocking new lessons for them to try. As the user accomplishes specific tasks they will be rewarded an achievement. This provides incentive to try a function of the app they might otherwise not and reward long term commitment to learning.

8. User Authentication:

Users must login via their Google account to use the services provided by the app. As all Android users are guaranteed to have a Google account this ensures no user will have to create an account to start using the app.

9. Translation:

The app will provide a feature to allow users to translate words and sentences from a supported language to another supported language.

Data requirements

I am using a NoSQL database to store multiple tables which will be held on Google Firebase as described earlier in this report. Firebase only supports NoSQL databases but NoSQL also allows me to easily pull key value pairs that can be converted to java data objects.

I will be displaying examples for these tables in a JSON format.

The "Users" table (Listing 1) will store information for each user who has signed into the app this will include name, email address, current language selected and their friends list. Users are identified with an id key that is randomly generated by firebase when first uploaded ensuring each user id key is unique.

```
"Users" : {
  "SZ7aVuPh9qPCiwVxRTLiN6u0bF03" : {
    "Email" : "jamieoneill17277@gmail.com",
    "Friends" : {
      "XnRGdk2d7WgtLh3tNra3g9eCsVy2" : "Sinead Downes"
    },
    "Language" : "German",
    "Name" : "Jamie O'Neill"
  },
},
```

Listing 1: Users Table

The “Messages” table (Listing 2) stores the conversations between two users. This is created by combining the two-user’s id’s as a conversation key. Each message is given a unique id and contains Date, To, From and Message text. The app only supports conversations with two users but this can be expanded by adding other parties’ user ids to the conversation key.

```
"Messages" : {
  "RhKUq21q4rW5jQP3FCaMDS4KU4r1SZ7aVuPh9qPCiwVxRTLIn6u0bF03" : {
    "-KXosm7Fjv8rgvmcwnvI" : {
      "date" : "12/03/2017",
      "from" : "SZ7aVuPh9qPCiwVxRTLIn6u0bF03",
      "text" : "hi Dave!!",
      "to" : "RhKUq21q4rW5jQP3FCaMDS4KU4r1"
    },
    "-KXoszqhuAPOoNRk_-dT" : {
      "date" : "12/03/2017",
      "from" : "RhKUq21q4rW5jQP3FCaMDS4KU4r1",
      "text" : "oh hi jim",
      "to" : "SZ7aVuPh9qPCiwVxRTLIn6u0bF03"
    }
  },
}
```

Listing 2: Messages Table

The “Meetings” (Listing 3) table stores all the information for each meet up this is stored by a unique generated id (in this example “-Khd8FZkt9TjcG-bsPPx”) and includes Title, Host, Location, Language, Time, Date, maximum level, minimum level, Note, Number of guests and an inner list of Attendees of the meet up which holds the unique id’s of the users.

```
"Meetings" : {
  "-Khd8FZkt9TjcG-bsPPx" : {
    "Attending" : {
      "-Khd8FZlgq-fK6kCizJH" : "SZ7aVuPh9qPCiwVxRTLIn6u0bF03",
      "-Khd8FZlgq-fK6kCizJI" : "XnRGdk2d7WgtLh3tNra3g9eCsVy2",
      "-Khd8FZlgq-fK6kCizJJ" : "RhKUq21q4rW5jQP3FCaMDS4KU4r1"
    },
    "Host" : "Jamie O'Neill",
    "Language" : "German",
    "Location" : "Dame St, Dublin, Ireland",
    "MaxLevel" : 5,
    "MeetingDate" : "29/7/2017",
    "MeetingTime" : "18:00",
    "MinLevel" : 1,
    "Note" : "For beginners who want to learn together",
    "NumGuests" : 5,
    "Title" : "German for beginners"
  }
},
```

Listing 3: Meetings Table

Once a meeting has passed its meeting date it is moved into a different table called “PastMeetings” this is so the meet up will no longer be accessible for users to view but can still be accessed by the app for statistical purposes and makes viewing the database tables easier from a developer point of view and running queries on the “Meetings” table faster as it has a smaller number of entries.

User requirements

These requirements have been drawn from a group of testers used to evaluate what aspects are most important to them from a non-functional point of view.

Usability

The most common feedback received was that the application must be user-friendly. All aspects of the application should be intuitive and require almost no learning curve.

Security:

Testers had concerns about security of data when stored on the cloud. This will be addressed by using the latest encryption methods provided by Google.

Cross-platform availability:

Testers expressed an interest in using the application on a multitude of devices such as android, iOS and windows phone. Due to budgetary and time constraints, the first release of my application will be focused on the Android platform. Later releases could support other mobile systems as well as a web platform to reach the maximum audience.

Pricing:

Testers brought up the issue of cost for the service stating a high price would deter from use of the app. This will be a free service, should cash flow become a hindrance in supporting the application it will be made monetized through the use of unobtrusive advertisements or purchasing additional lessons that are optional such as lessons for non-essential speaking phrases.

Language Support:

Some testers stated they had no interest in the available languages and would like more options to select from. The number of supported languages would be expanded upon full release of the app.

Use Cases

Requirement 1: Add Meet up

Description & Priority

Once the user has signed in they will have the ability to create a meet up. The Meet up will store the user who created the meet up, title, location, time, date, minimum recommended level, maximum recommended level, number of guests and an optional note. This meet up will then be added to the database and available for other users to join, this is one of the fundamental requirements of the system.

Use Case 1

Application

Now You're Talking application.

Use Case Diagram

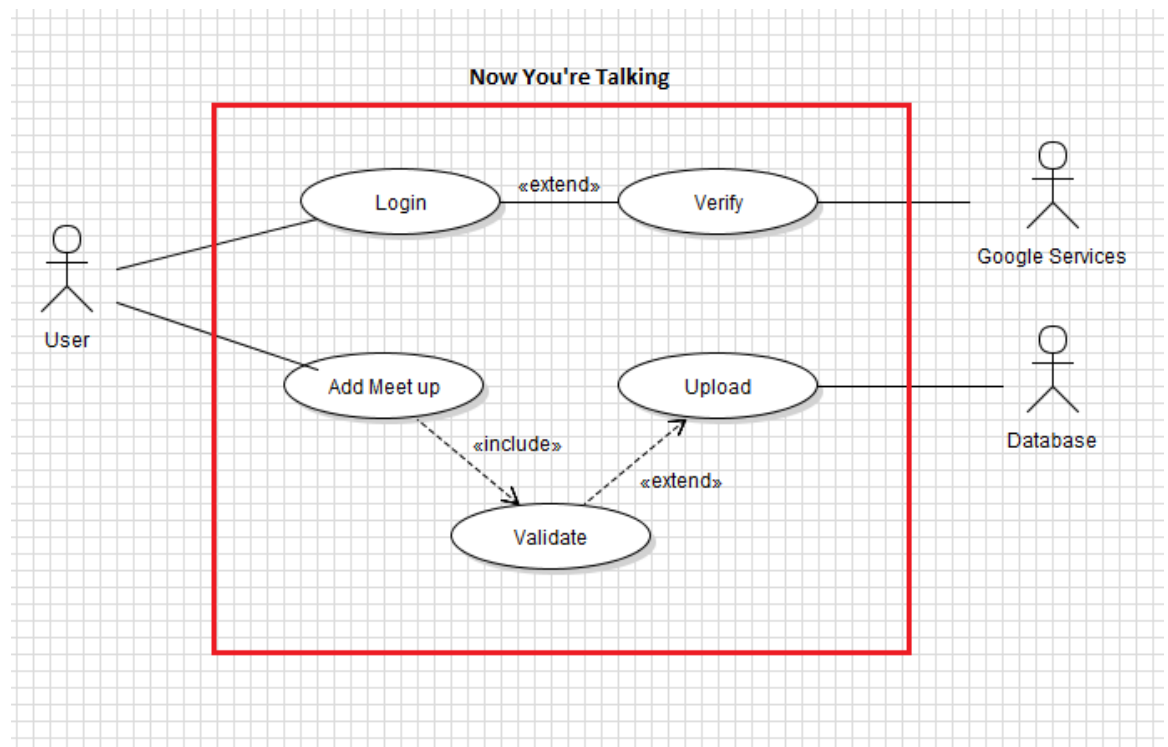


Figure 3: Add meet up use case diagram

Actors

1. User

Flow Description:**Preconditions**

The application must be installed and user logged in.

The user's device must have an internet connection.

Activation

Active from the "Add meet up" button located on the home screen of the application.

Main flow

1. User enters details of Meet up
2. System verifies all necessary information has been entered
3. Information is uploaded to the database

Alternate flow

If there is no internet connection present the upload will take place the next time internet connectivity is present.

Termination

The user cancels the process or loses internet connection before pressing the "Add Meet up" button.

Post condition

The meet up is now available for all other users to view. The application sends the user to the "Manage Meet ups" section to review their meet up.

Requirement 2: Join Meet up

Description & Priority

Users wish to join an existing meet up. These are stored in a cloud database and retrieved asynchronously by the app. The meet ups are displayed on a Google map for the user to select based on location and language.

Use Case 2

Application

Now You're Talking application.

Use Case Diagram

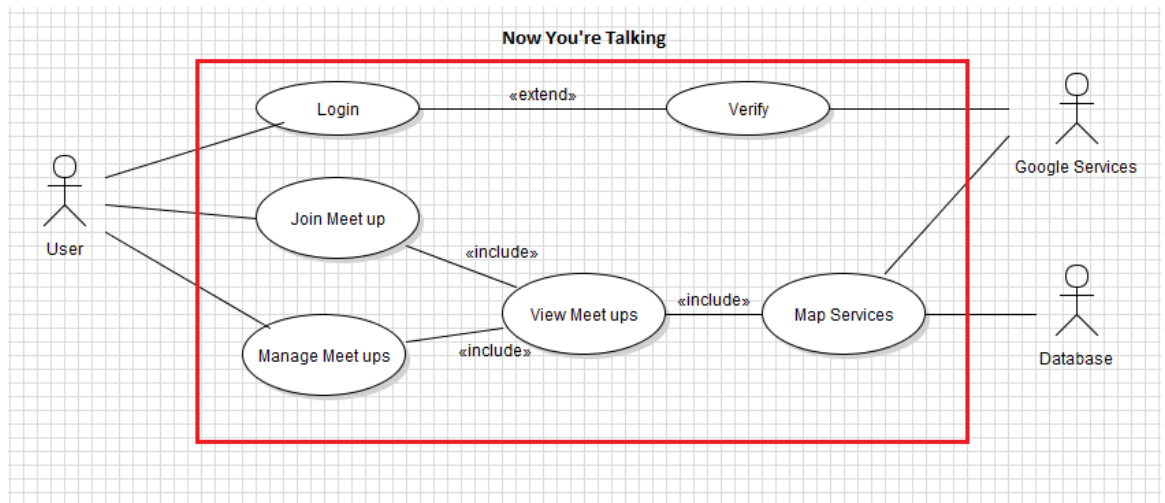


Figure 4: Join meet up use case diagram

Actors

1. User

Flow Description:

Precondition

The application must be installed and user logged in. The user's device must have an internet connection.

Activation

Active when the user selects the "Find a Meet up" button

Main flow

1. Map is displayed on device
2. Current meet ups are loaded onto the map
3. User selects a meet up marker
4. Map displays details of marker selected
5. User selects "Join" button

6. Database is updated
7. User is taken to “Manage Meet up” section

Alternate flow

1. If location services are turned off the user will receive an alert asking if they wish to use these services. Users can choose to turn this on or leave it off and the main flow will then begin.
2. If internet connection is lost at any point after the map has been loaded step 6 is paused and will be restarted once internet connectivity is restored while in the app. In this case step 7 is not completed.

Termination

The user cancels the process or loses internet connection before pressing the “Join” button.

Post condition

The application sends the user to the “Manage Meet ups” section to review the meet up they have joined.

Fully functional use case

The use case shown in figure 3 shows how the user interacts with all of the apps functions and how those functions interact with systems outside of the app. These functions will be explained in detail further in this report.

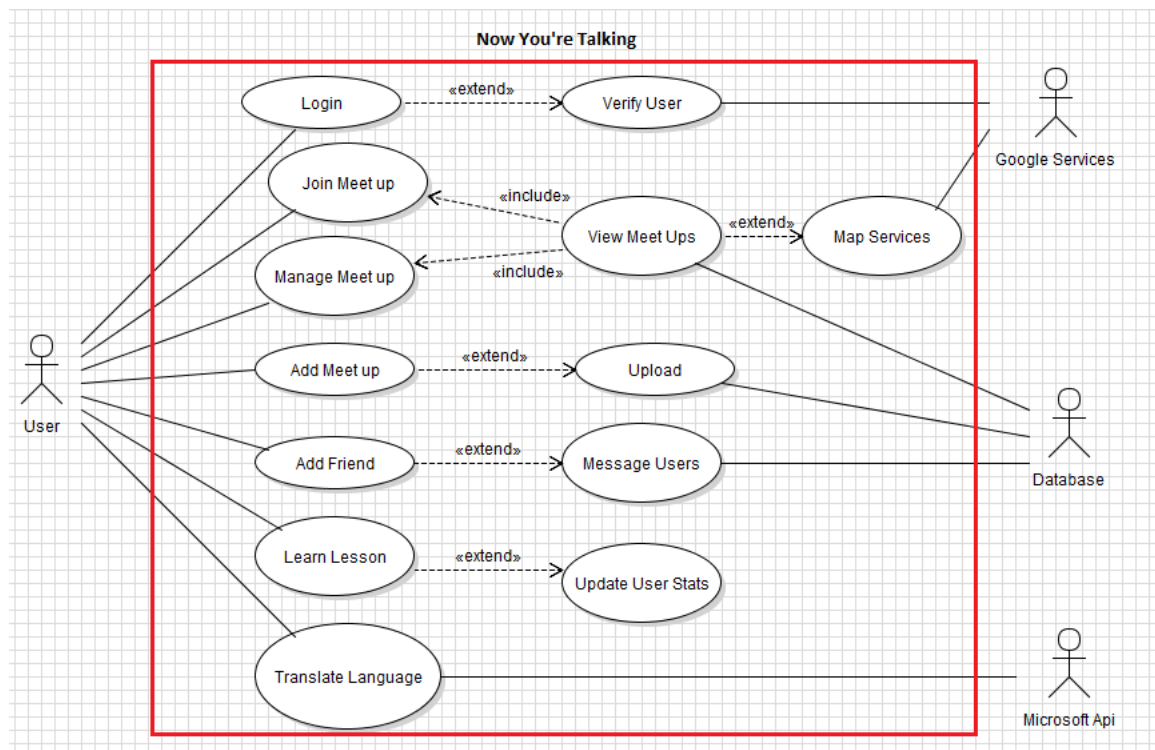


Figure 5: Fully functional use case diagram

Non-Functional Requirements

Performance/Response time requirement

The app must act in real time when displaying information that is stored in the database such as meet up locations or messages. There should be minimal loading times in any part of the app to minimise the impact on the user some functions will happen asynchronously and the UI will be updated accordingly during and after the action.

Availability requirement

The application is planned to have high availability for all users. The only down time that could be expected is in the meet ups and messaging features if Firebase should go down which is out of my control. The learning sections will always be available.

Disaster Recovery requirement

All key data of a user will be stored on a cloud database. As long as the user has access to the Google account they signed in with they will not lose this data.

The application can be updated without loss of previous user data.

Robustness requirement

The application must work on the device unless it is uninstalled by the application's installer or the device does not have an active internet connection. The application must work as marketed allowing users to learn languages and facilitate holding meetings of its users.

Security requirement

The application must store all recorded data and user details securely. This is achieved by using Android shared preferences which is inaccessible to users unless their device is rooted. Encryption is also performed by Google authentication on sign in, this is handled by Google.

Maintainability requirement

The application can easily allow for software updates to be released should that be new features, bug fixes or just improvements in the code to speed up or decrease the size of the application. I also have full access to what users can use the application and the information stored in the database

Portability requirement

The application will be compatible with all versions of Android devices from version 4.0 up.

Extendibility requirement

The application is easily extendable with Android allowing for updates to be built on top of the existing application and pushed to users via the Google Play Store.

Reusability requirement

All data stored in the database has been designed to be reusable across multiple platforms, such as IOS or a web platform, as firebase is compatible with multiple platforms. This allows for easier extension onto those platforms.

Resource utilization requirement

I will try to keep the memory as low as possible but the size of the app will increase as more updates are applied.

Design and Architecture

The application is installed on an android device. The app makes several calls to external api's for certain features such as Google accounts for user profile and Google places for location names. The app also connects to my firebase console which stores the database, image storage. The firebase server can also handle additional features such as notifications

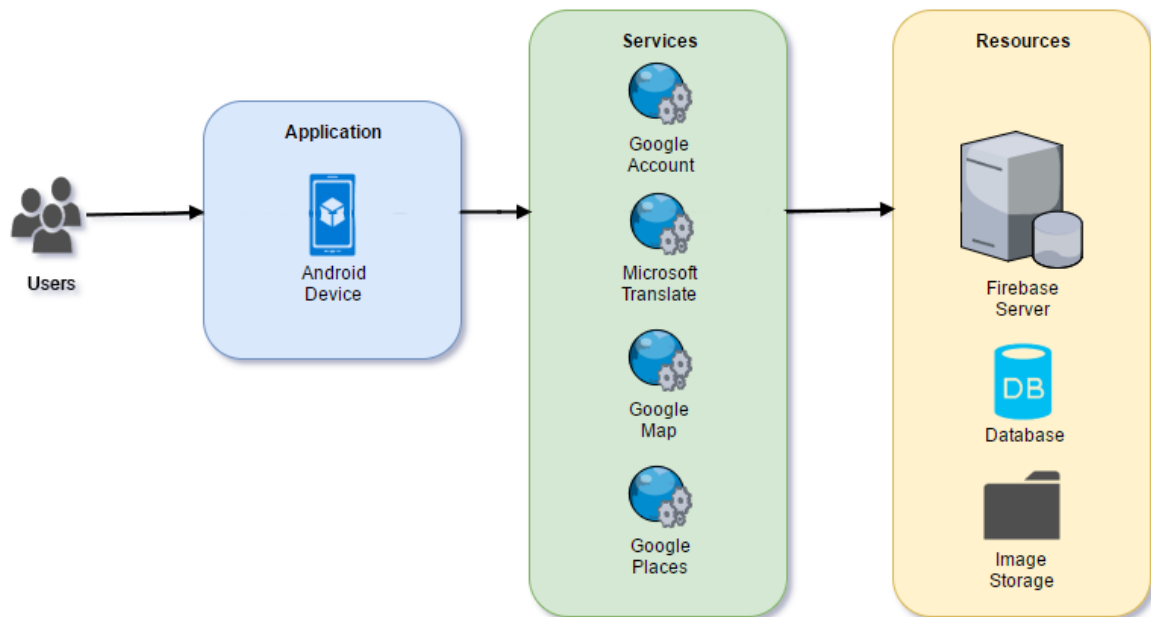


Figure 6: Architecture diagram

Implementation

These class diagrams will show the classes and methods being used in the application and their relationship with one another.

Main Activity:

The first class launched by the app is the SplashActivity, this is used to display an image to the user while the rest of the app loads. Once the app is loaded the SignInActivity is launched, this is where the user signs in with their Google account, if already signed in the MainActivity is loaded. The MainActivity holds the navigation bar, selecting different sections in the navigation bar changes the fragment shown in the MainActivity. The MainActivity displays one of three fragments at a time.

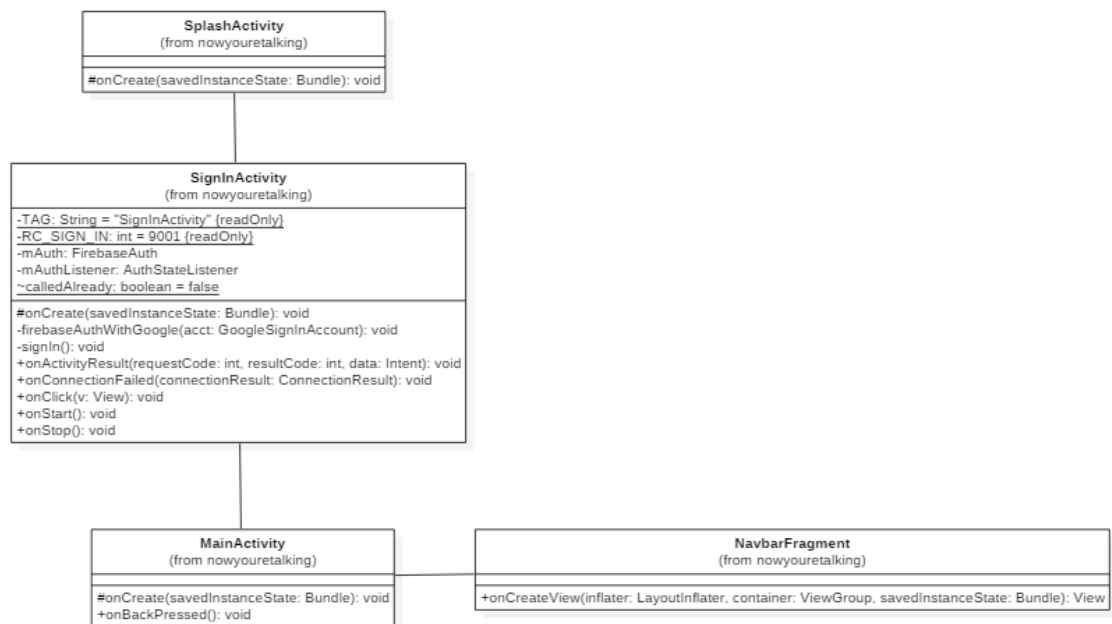


Figure 7: Main Activity diagram

Social Fragment:

The social fragment launches three activities. Figure 8 will show some of the methods related to creating the Google maps view on the MapsActivity while the MeetingActivity shows the method used to upload the new meet up to the database.

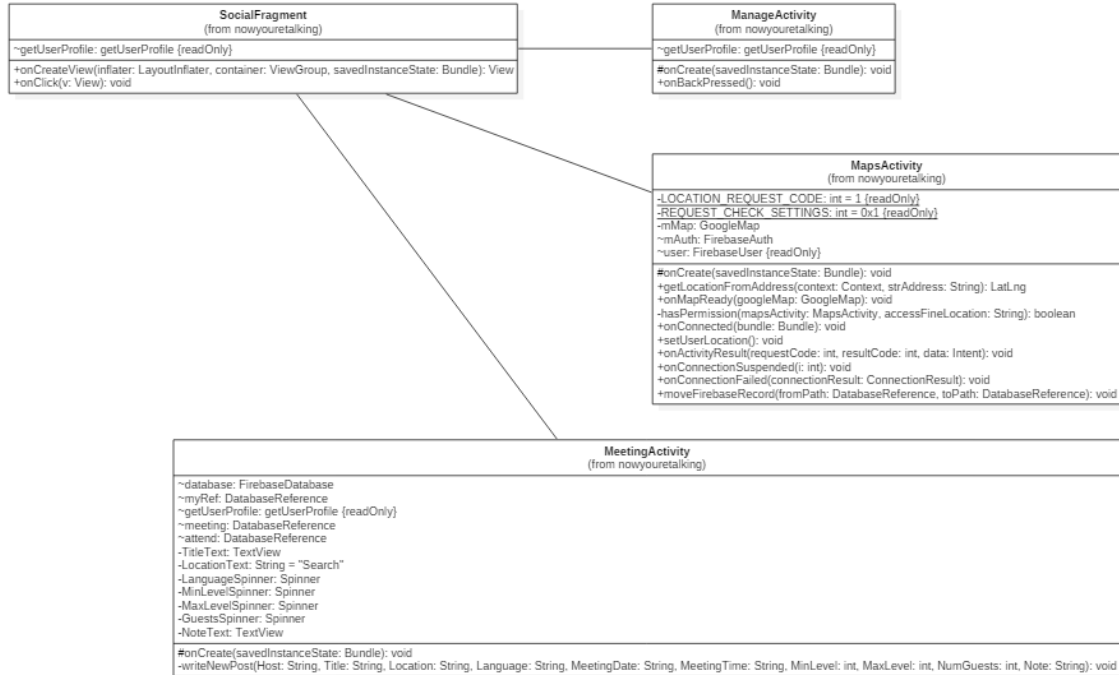


Figure 8: Social Fragment diagram

Learn Fragment:

In the LearnFragment class we set the progress bars on load. The card views that are shown when selecting a lesson are loaded when the relevant activity button is pressed.

The GameActivity is where the lesson takes place. Firstly, the getQuestionList method runs to pull in the questions for that lesson. The loadNextQuestion method is used to cycle through the questions that were loaded. The checkAnswer method compares the user's response to the correct answer for the currently shown question and adds a score to their progress if correct. The hideAllViews method is used to remove the previous questions layout before loading in the next question.

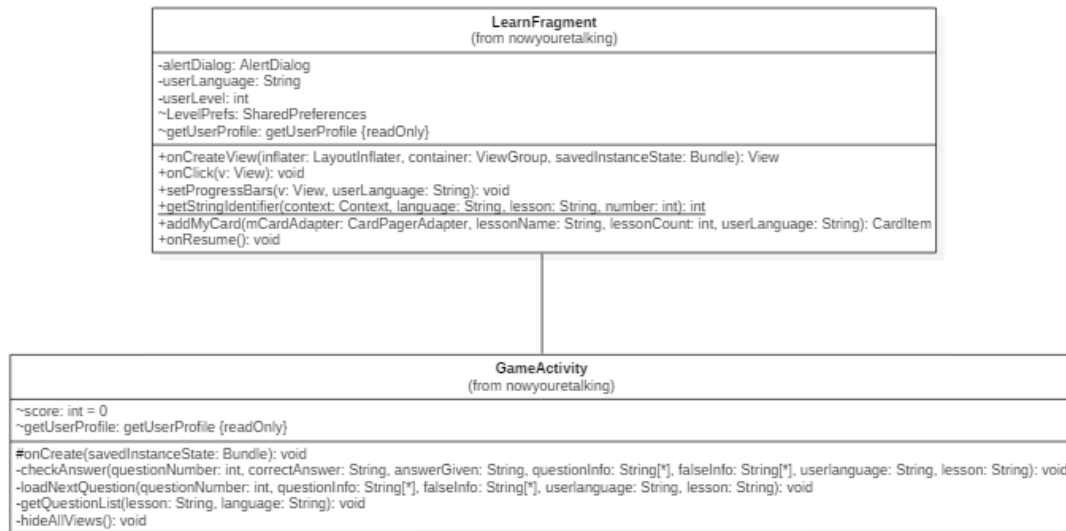


Figure 9: Learn Fragment diagram

Profile Fragment:

The profile fragment holds the methods used for saving and loading images. These are public methods so they can also be called from the messaging activities.

TranslateActivity's methods are used for translating user text, speaking out text and reading in speech to text.



Figure 10: Profile Fragment diagram

Messaging Activity:

The messaging activity holds two fragments, messaging and friend fragments.

The messaging fragment is used to display all of the user's current messages and continue conversations. The friend fragment allows for users to add friends and create new conversations with their friends.

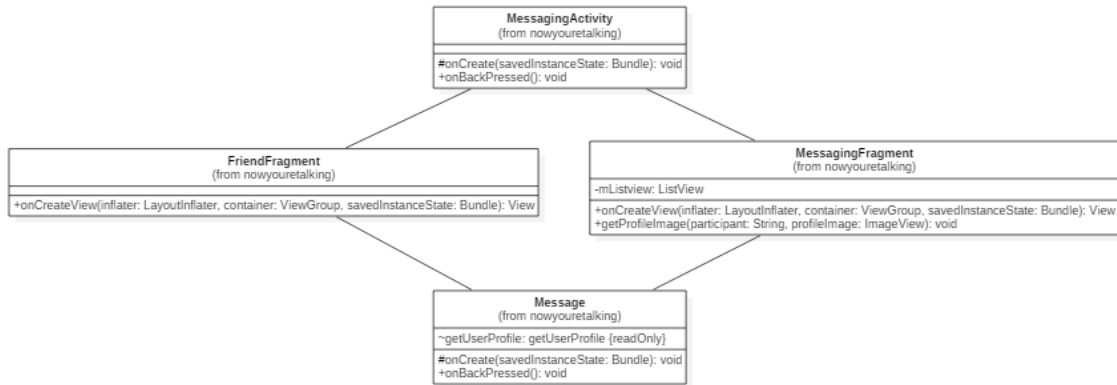


Figure 11: Messaging Activity diagram

Features and layout

This section will describe the features of the application and how they are displayed with practical code accompanying some of the features.

Navigation Screens

There are three navigation screens for this app. These navigation screens display all the activates relating to the selected section. The navigation bar at the bottom of the screen shows what category we are currently in and allows users to move from section to section without loading times. There is a consistent style throughout the application with bright colourful circular images being used for the user to access other areas of the app.

Each of these screens will be examined more closely to explain their features.

Social Screen

When the app is loaded up the first section to open is the social screen (provided the user is signed in). As seen in figure 12, from the social screen the user can get access to all of the meet up functionality available. The “My next meet up” section will only appear when the user has a meet up scheduled to give quick information on the meet up without having to go any deeper within the application.

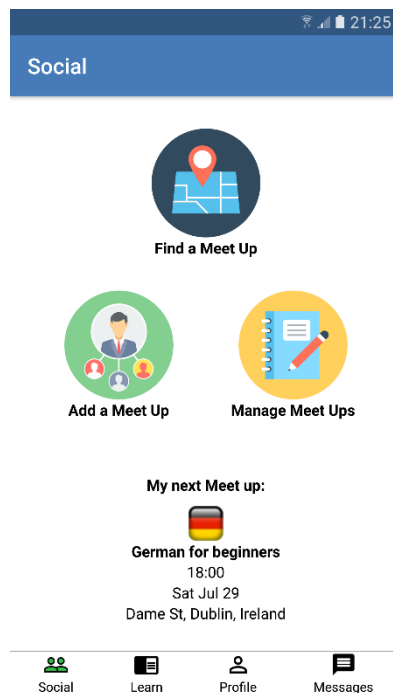


Figure 12: Social screen

Learn Screen

This is where the user selects the lesson to learn. Only one activity is available in the beginning and as the player progresses to new levels more and more activities become unlocked. Each activity contains varying numbers of lessons. Once the user completes a lesson their progress is stored and represented in a progress bar under each lesson as seen in figure 13.2.

Once all the lessons of an activity are complete with 100% correctness the user moves up a level and the next activity is unlocked. The sum of these lesson scores is also represented as the activity's overall progress bar, these progress bars turn gold once completed to indicate to the user they have achieved the highest possible score for the activity.

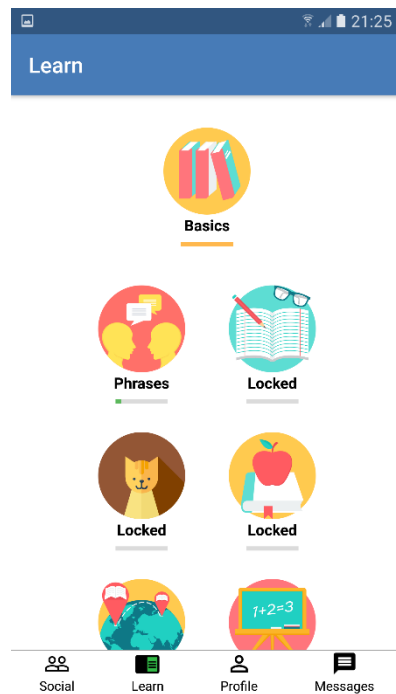


Figure 13.1: Learn screen

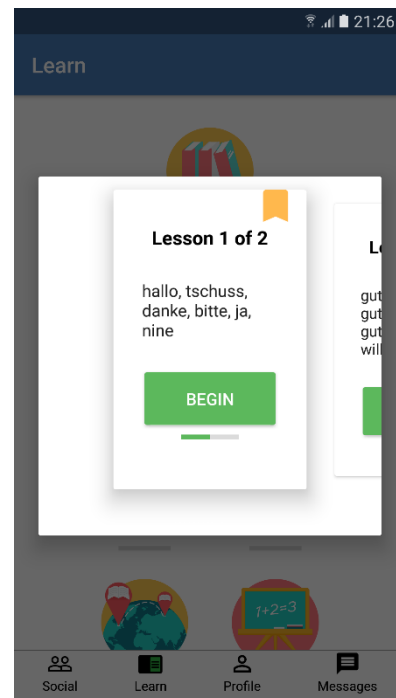


Figure 13.2: Lesson select

Profile Screen

The profile screen displays the user's profile picture, name, current language they are studying and their level in that language.

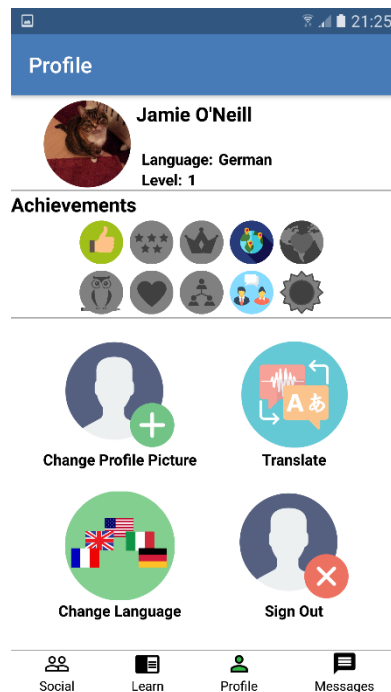


Figure 14.1: Profile

There are five features available from this screen (figure 14.1).

Change language

This allows the user to change the language they wish to study. Progress of each language is stored individually meaning the user can change languages on the fly without losing their level of each language. Changing the language updates the lessons that are shown on the learn screen.

Sign out

Allows the user to sign out of their account and log into another account they may wish to use.

Change profile picture

This allows the user to change the image displayed on their profile. This image saved online for other users to see when they are messaging.

Practical code:

The following code is used to load a user's profile picture from the online storage. Three variables, `userToLoad`, `viewToFill` and `activity`, are passed to the method as seen in the first line of the code (Listing 4). This is a public method meaning I can call this piece of code from anywhere in the application wherever it is needed. If the user has an image in the online storage it is pulled in and saved locally to the device, via another method, so that later it will not need to be downloaded again but just taken from the device's memory making the app faster to load. If the user has yet to select a profile picture the default image is loaded instead. Then some styling is put on the image to round the edges to make it look more appealing on the eye.

```
public String loadProfileImage (final String userToLoad, final ImageView viewToFill, final Activity activity){
    // Create storage reference
    final StorageReference storageRef = storage.getReferenceFromUrl("gs://now-youre-talking.appspot.com/ProfilePictures/");
    //set image based on user id
    StorageReference myProfilePic = storageRef.child(userToLoad);
    //set max image download size
    final long ONE_MEGABYTE = 10000 * 10000;
    myProfilePic.getBytes(ONE_MEGABYTE).addOnSuccessListener(new OnSuccessListener<byte[]>() {
        @Override
        public void onSuccess(byte[] bytes) {
            //decode image
            Bitmap userImage = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
            //save this user image to local device
            saveToInternalStorage(userImage, userToLoad, activity);
            //set circle style
            RoundedBitmapDrawable roundedImage = RoundedBitmapDrawableFactory.create(activity.getResources(), userImage);
            roundedImage.setCornerRadius(Math.max(userImage.getWidth(), userImage.getHeight()) / 2.0f);
            ImageView imgView = viewToFill;
            imgView.setImageDrawable(roundedImage);
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
            //reset to default image if no image is selected
            StorageReference myProfilePic = storageRef.child("defaultImage");
            myProfilePic.getBytes(ONE_MEGABYTE).addOnSuccessListener(new OnSuccessListener<byte[]>(){
                @Override
                public void onSuccess(byte[] bytes) {
                    Bitmap userImage = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
                    RoundedBitmapDrawable roundedImage = RoundedBitmapDrawableFactory.create(activity.getResources(),userImage);
                    roundedImage.setCornerRadius(Math.max(userImage.getWidth(), userImage.getHeight()) / 2.0f);
                    ImageView imgView = viewToFill;
                    imgView.setImageDrawable(roundedImage);
                }
            });
        }
    });
    return null;
}
```

Listing 4: loadProfileImage code

Achievements

Achievements are in place to give the user a sense of accomplishment. There are currently 10 achievements available ([In app achievements](#)) but more can be added to keep users engaged. The achievements are designed in 3 ways. The first set of achievements are easily unlockable for the user such as complete a lesson or complete a lesson after 10 pm. These are in place to get the user interested in the achievement hunting by starting them off with simple tasks. The second set of achievements are designed to get the user to use features they may otherwise not and engage in all of the apps features such as using the translator or adding a friend. The final set of achievements are rewarded to players for accomplishing long term goals such as reaching level 50 of a language or attending 10 meet ups, this will encourage the users to stay active in the app and provide long term rewards.

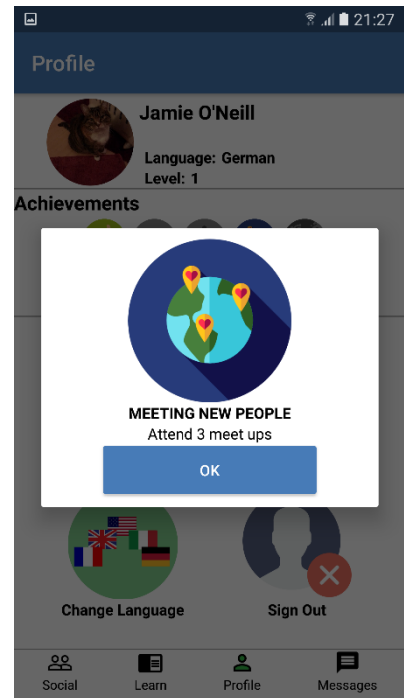


Figure 14.2: Achievements

Practical code:

Achievements are handled in the AchievementHelper class.

Throughout the app, achievements are placed in the code in their relevant sections where the achievement should be unlocked. This is done by passing the title of the achievement to the achievement helper class

```
//check Achievements
AchievementHelper AchievementHelperClass = new AchievementHelper();
AchievementHelperClass.UnlockAchievement("UseTheTranslator", TranslateActivity.this);
```

The achievement is then unlocked if it has not been already.

```
public void UnlockAchievement(String unlock, Activity activity){
    //get Achievements
    SharedPreferences Achievements = activity.getSharedPreferences("Achievements",
        MODE_PRIVATE);
    Boolean thisAchievement = Achievements.getBoolean(unlock, false);
    //if it hasn't been unlocked yet unlock it
    if(!thisAchievement){
        SharedPreferences.Editor editor = Achievements.edit();
        editor.putBoolean(unlock, true);
        editor.apply();
        //toast
        Toast.makeText(activity, activity.getString(R.string.achievementUnlocked),
            Toast.LENGTH_SHORT).show();
    }
}
```

Listing 5: Achievement code

Translate

The translation feature allows for users to translate to and from all available languages the app supports. This can be done by entering text or using the speech to text feature. The output is displayed in the selected language which can be read out via the text to speech feature.

Practical code:

Firstly, we need to import the dependencies from Microsoft

```
import com.memetix.mst.language.Language;
import com.memetix.mst.translate.Translate;
```

Then define our Translate API key

```
//set authentication
Translate.setClientId("NowYoureTalking");
Translate.setClientSecret("nowyourtalkingsecret");
```

Connecting to Microsoft's translate API requires an internet connection so firstly we need to insure there is a connection if so we can then translate.

```
//check for internet
ConnectivityManager connectivityManager =
    (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();

if(activeNetworkInfo != null){
    //do the translation
}
```

Once text has been entered and the translate button is pressed the translation is done. This is done with the Translate.execute() method. This method takes in three parameters. The text you want translated, the language of the text entered and the output language.

```
//return translation
String translatedText =
    Translate.execute(originalText, Language.valueOf(fromLanguage), Language.valueOf(toLanguage));
//set result text
ResultLabel.setText(translatedText);
```

Now that we have the translated text we can use Android's text to speech feature to speak out the translated words with a native accent by first setting the language then passing in the text as the "input" parameter.

```
//speak out the text
tts.speak(input, TextToSpeech.QUEUE_FLUSH, null);
```

Listing 5: Translate code

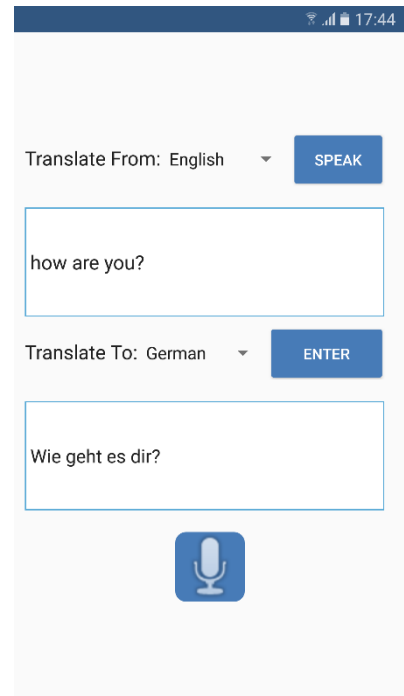


Figure 15: Translation

Joining a Meet up

When the user taps the “Find a meet up” button from the social screen a map is loaded. The map uses the user’s location to pinpoint and display all meet ups in the user’s area that still have available spaces for other users to join. The user’s own meet ups are shown but greyed out on the map. When a user tap on a meet up a dialog box is shown displaying the information about that meet up with an option to join or get directions to the location.

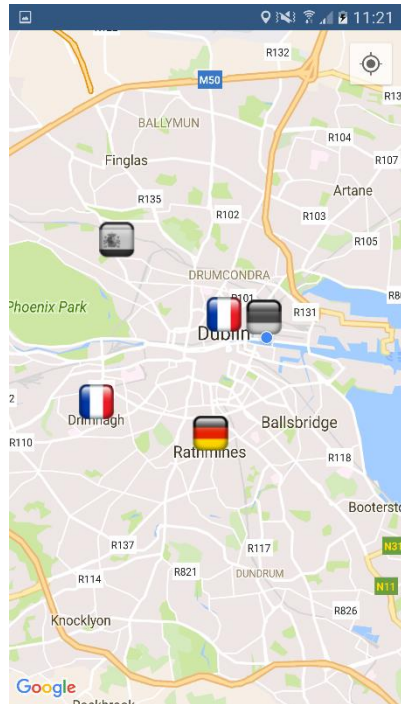


Figure 16.1: Map of meet ups

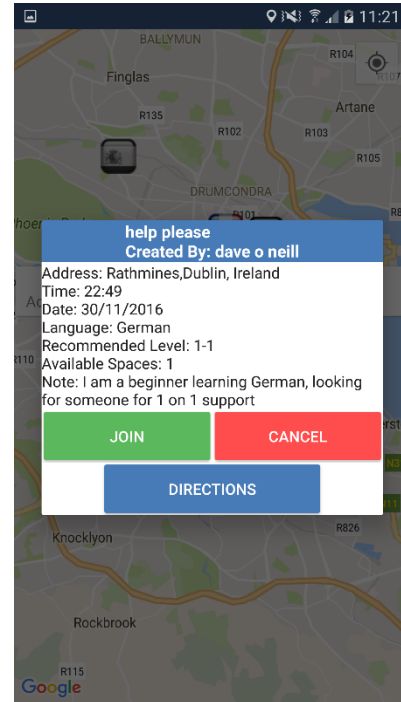


Figure 16.2: Joining a meet up

Practical code:

This code will show how the app reads in from the database and the key value naming convention then displays this as a marker.

When the map is ready it runs through each entry in the “Meetings” table defined previously in this report. Firstly, it checks how many users are currently attending the meet up and will only display it if there is room for another attendee. If there is room for another guest it pulls back all the data for the meet up.

```
// Read from the database
myRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot child : dataSnapshot.getChildren()) {
            //show if user is not already attending
            String Guests = child.child("Attending").getValue().toString();
            //find out if spaces are available
            int NumGuests = child.child("NumGuests").getValue(int.class);
            int Attending = (int) child.child("Attending").getChildrenCount();
            int Spaces = NumGuests - Attending;
            //if there is a space get info and display
            if (Spaces >= 1) {
                //get info and display
                //get info from DB
                String Host = child.child("Host").getValue(String.class);
                String Titles = child.child("Title").getValue(String.class);
                String Locations = child.child("Location").getValue(String.class);
                String MeetingTime = child.child("MeetingTime").getValue(String.class);
                String MeetingDate = child.child("MeetingDate").getValue(String.class);
                String Language = child.child("Language").getValue(String.class);
                int MinLevel = child.child("MinLevel").getValue(int.class);
                int MaxLevel = child.child("MaxLevel").getValue(int.class);
                String Note = child.child("Note").getValue(String.class);
                //convert address
                LatLng newaddress = getLocationFromAddress(MapsActivity.this, Locations);
```

It then checks what language the meet up is in and changes the marker icon accordingly.

```
// marker icon holder
BitmapDescriptor markerIcon;
//add maker here and add the database key
if (Language.equals("French")) {
    if (Guests.contains(user.getUid())){
        markerIcon = FrenchDark;
    }else{
        markerIcon = French;
    }
}
```

All this data is then passed to a map marker as well as a tag of the key of the database entry. This key will be used later when a user selects to join the meet up so we know which meet up the user wants to join. This can be seen later in Figure 11.1.

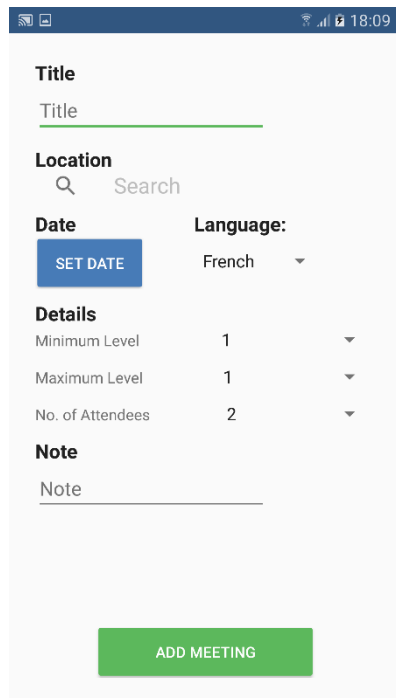
```
Marker marker = mMap.addMarker(new MarkerOptions().position(newaddress).title(Titles + "\nCreated By: " +
Host).snippet("Address: " + Locations + "\nTime: " + MeetingTime + "\nDate: " + MeetingDate + "\nLanguage: " + Language +
"\nRecommended Level: " + MinLevel + "-" + MaxLevel + "\nAvailable Spaces: " + Spaces + "\nNote: " + Note).icon(markerIcon));
marker.setTag(child.getKey());
}
```

Listing 6: Map marker code

Adding and Managing Meet ups

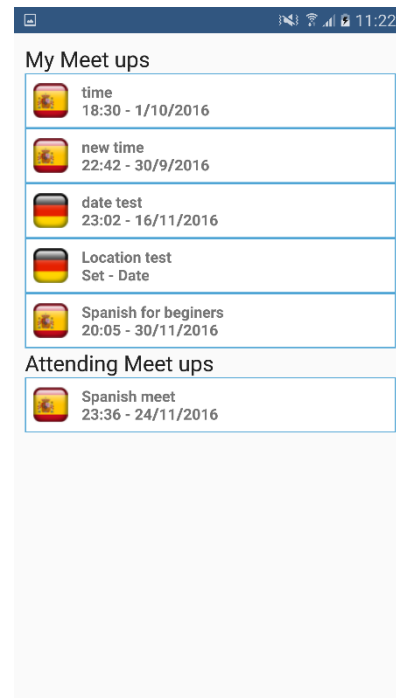
When adding a meet up the user must fill all details about the meet up except for an optional note. Each piece of information has an appropriate input type depending if it is text, a number or a date. These fields are validated before being uploaded ensuring all necessary text is filled and the date has not already passed.

The manage meet ups section displays all meet ups a user is currently attending which is divided into the meet ups they own or the meet ups others own and they are attending. When a meet up is tapped, the user has the option to remove themselves from said meet up.



The screenshot shows a mobile app interface for adding a new meeting. The form is divided into several sections: 'Title' with a text input field, 'Location' with a search bar, 'Date' with a 'SET DATE' button, and 'Language:' with a dropdown menu currently set to 'French'. Below these is a 'Details' section with three rows: 'Minimum Level' with a value of 1, 'Maximum Level' with a value of 1, and 'No. of Attendees' with a value of 2. Each row has a dropdown arrow on the right. At the bottom is a 'Note' section with a text input field. A large green 'ADD MEETING' button is at the very bottom.

Figure 17.1: Adding a meet up



The screenshot shows two sections of a mobile app. The top section, 'My Meet ups', contains a list of five items, each with a Spanish flag icon, a title, and a date: 'time' (18:30 - 1/10/2016), 'new time' (22:42 - 30/9/2016), 'date test' (23:02 - 16/11/2016), 'Location test' (Set - Date), and 'Spanish for beginners' (20:05 - 30/11/2016). The bottom section, 'Attending Meet ups', contains one item with a Spanish flag icon, the title 'Spanish meet', and the date '23:36 - 24/11/2016'.

Figure 17.2: Managing meet ups

Messaging

The messaging section shows a list of users you are currently have conversations with, their user profile image, the last message received and date. Users can add friends and start new conversations from the “Friends” tab. From the friends tab a user can add new friends via their email address.

When a user taps on one of their conversations the message screen is shown allowing users to communicate in real time to one another.

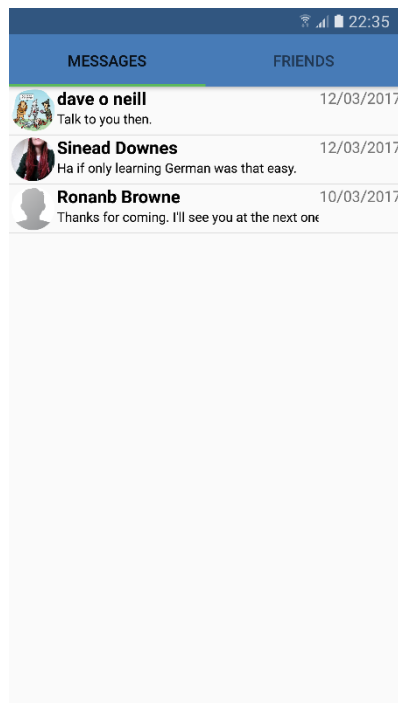


Figure 18.1: Messages list

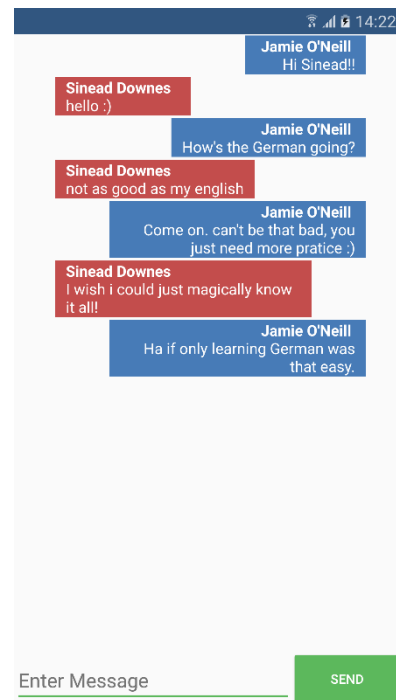


Figure 18.2: A user conversation

As shown in Listing 2 ([Messages Table](#)) each conversation is stored in a database. This data is pulled back from the database and styled differently depending on whether the user sent or received the message resulting in a conversation screen as shown in Figure 18.2.

Lessons

From the Learn screen once a user selects an activity and a lesson (figure 13.1 & 13.2) they will begin the lesson. A lesson consists of 10 questions, although this can be expanded upon, where the user must answer all questions correctly in order to complete the lesson. These questions are presented in a game format with each question providing a different game design. There are currently 6 types of questions in the app to begin with but again more question types would be added if this app is to move past the proof of concept stage.

Even though each lesson has designated words it aims to teach, some words are introduced before its designated lesson to gently ease the users into learning words indirectly through either visuals or process of elimination thus making the subsequent lessons slightly easier to learn. The Lesson shows a progress bar along the top of the screen to indicate how far they are through the lesson.

Each type of question will be explained in more detail to show their difference in presentation and what purpose the question type serves in terms of learning.

Image Question:

This question type is used as the introductory question using visuals to show the user not only the word they are trying to learn but so other words that will be used in latter lessons.

Once the user selects an image the “Check” button becomes enabled and validates the users answer

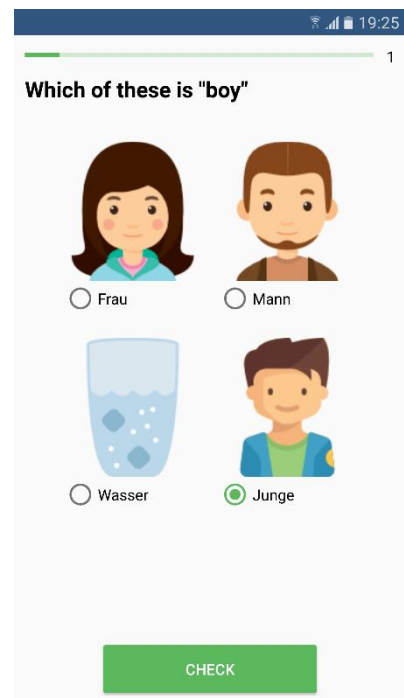
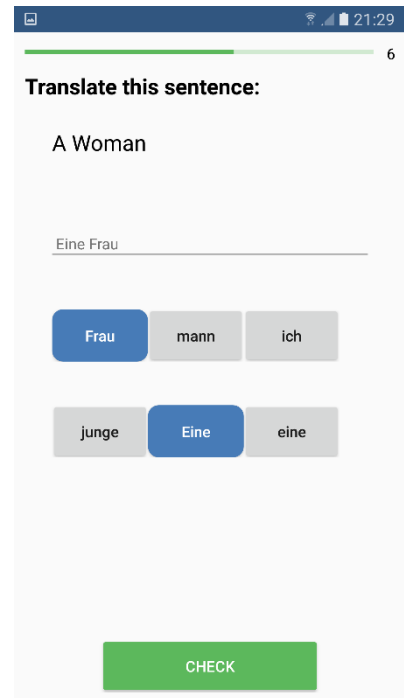


Figure 19: Image question

Build the sentence

The user is tasked with translating a sentence that is in either the language they are learning or their own language. This is done by building the sentence word by word by tapping on the words shown on screen. When a word is added to the sentence it is displayed on the line above the words, this will also highlight the word to show it has been selected. When a highlighted word is tapped again it is removed from the sentence. This is in place so a user can correct any mistakes they may make.

Once at least one word has been selected the “Check” button becomes available.



The screenshot shows a mobile app interface for building a sentence. At the top, there is a status bar with a battery icon, signal strength, and the time 21:29. Below the status bar is a green progress bar and the number 6. The main heading is "Translate this sentence:". Below this, the text "A Woman" is displayed. Underneath, the text "Eine Frau" is shown in a light blue box. Below this, there are two rows of word buttons. The first row contains "Frau" (highlighted in blue), "mann", and "ich". The second row contains "junge", "Eine" (highlighted in blue), and "eine". At the bottom, there is a green "CHECK" button.

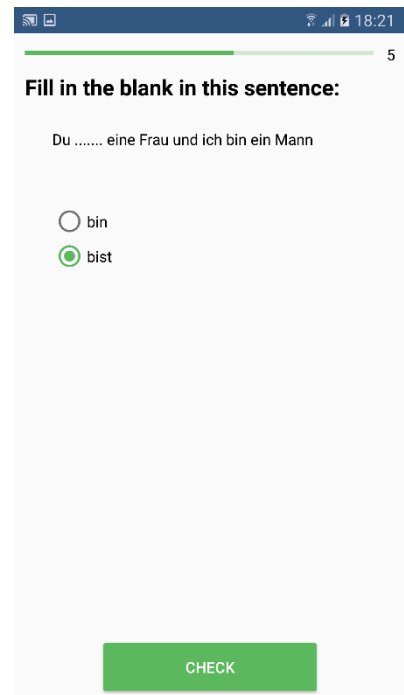
Figure 20: Build the sentence

Fill in the blank

The user is presented a sentence with part of the sentence missing. It is up to them to determine which of the options available is correct.

The number of options can vary between two and four.

This is one of the more difficult questions and requires the user to apply what they have learned in previous questions in order to be able to understand the sentence as a whole.



The screenshot shows a mobile app interface for filling in the blank. At the top, there is a status bar with a battery icon, signal strength, and the time 18:21. Below the status bar is a green progress bar and the number 5. The main heading is "Fill in the blank in this sentence:". Below this, the text "Du eine Frau und ich bin ein Mann" is displayed. Underneath, there are two radio button options: "bin" and "bist". The "bist" option is selected, indicated by a green dot. At the bottom, there is a green "CHECK" button.

Figure 21: Fill the blank

Translate Question

The user is given a sentence and asked to write the correct translation into the textbox provided.

The sentence can be to or from their selected language.

This question is usually preceded by an easier type of question that uses the words the user is asked to translate. This creates a flow of learning designed to get the user to understand the meaning and spelling of words through repetition.

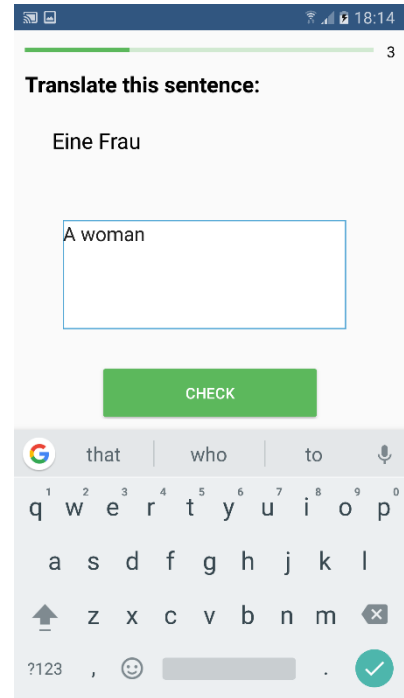


Figure 22: Translate sentence

Translate Image

Using a combination of previous question types the user must translate a word with the help of a visual aid. This question is asked using words they have learned earlier in the current lesson or from a previous lesson.

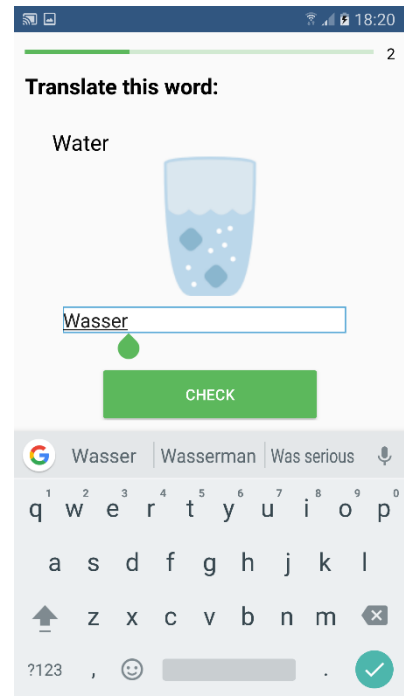


Figure 23: Translate image

Match the words

This is a matching game aimed to help the user associate words correctly.

First the user taps a word which then becomes highlighted then they tap the matching word. If correct, the user is given feedback and the words are disabled. The question continues until all the words are matched.

This is the only question in which the user cannot get wrong. This is to ensure the user is correctly identifying the words before moving on.

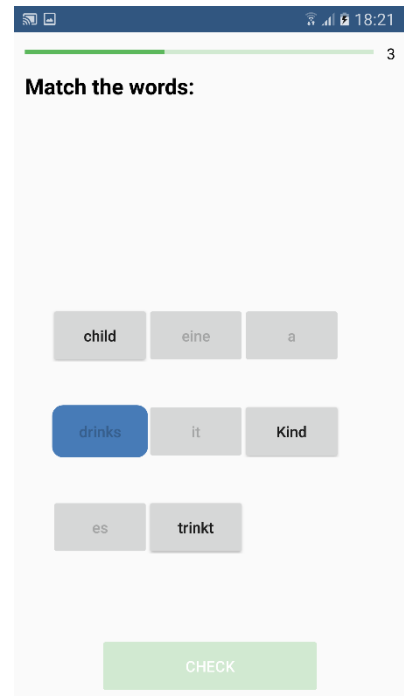


Figure 24: Matching

Practical code:

Questions are stored in xml files. Each language has its own file to read questions from. In the language file the questions are sorted by activity and lesson number. In Listing 7 we can see these questions are for the German Basics activity lesson number 3.

The questions are stored in the structure of Question type, Question, Answer.

```
<string-array name="German_Basics3_questions_array">
  R.array.German_basics_questions_array
  <item>ImageQuestion,Which of these is \"water\",Wasser</item>
  <item>TranslateQuestion,Brot und wasser,Bread and water</item>
  <item>ButtonQuestion,Ein Madchen trinkt, A girl is drinking</item>
  <item>TranslateImageQuestion,Water,Wasser</item>
  <item>MatchingQuestion,none,Well done</item>
  <item>MatchingQuestion,none,Well done</item>
  <item>MissingQuestion,Du ..... eine Frau und ich bin ein
  Mann,bist</item>
  <item>ButtonQuestion,Sie ist eine Frau, She is a woman</item>
  <item>TranslateImageQuestion,Bread,Brot</item>
  <item>MissingQuestion,Du ..... ein Junge,bist</item>
</string-array>
```

Listing 7: German questions

Listing 8 shows the other options to be populated with the question when it is asked in the lesson. Figures 19 – 24 shows examples of how these options are used within the lesson.

```
<string-array name="German_Basics3_false_array">
    R.array.German_basics_questions_array
    <item>Mann,Brot,Frau</item>
    <item>none,none,none</item>
    <item>is,girl,it,are,water,drinking,A</item>
    <item>none,none,none</item>
    <item>1eine,2a,3Kind,4child,5es,6it,7drinks,8trinkt</item>
    <item>1I,2ich,3er,4he,5bist,6are,7sie,8she</item>
    <item>bist,bin,none,none</item>
    <item>She,is,a,woman,and,boy,girl,child</item>
    <item>none,none,none</item>
    <item>bist,bin,none,none</item>
</string-array>
```

Listing 8: False options

These question arrays are expandable to fit any number of questions and question types.

When the lesson is started, it will load the question list dependant on the user's selected language, activity and lesson number.

The app uses the `getQuestionList` method (Listing 9) to get all the question details. A switch statement is then used to check the question type and the layout is changed according to the question type.

```
private void getQuestionList(String lesson, String language) {
    //set array used for game
    String languageLessonArray = language + "_" + lesson + "_questions_array";
    //get id of array
    int resId = GameActivity.this.getResources().getIdentifier(languageLessonArray, "array",
        GameActivity.this.getPackageName());
    //pull back questions for this game
    String[] questionInfo = getBaseContext().getResources().getStringArray(resId);

    //set wrong questions for game
    String languageFalseArray = language + "_" + lesson + "_false_array";
    //get id of array
    int resId2 = GameActivity.this.getResources().getIdentifier(languageFalseArray, "array",
        GameActivity.this.getPackageName());
    //pull back false answers for this game
    String[] wrongAnswers = getBaseContext().getResources().getStringArray(resId2);

    //load the first question which is 0 in array
    loadNextQuestion(0, questionInfo, wrongAnswers, language, lesson);
}
```

Listing 9: `getQuestionList` method

While adding new question types requires coding, adding new languages and activities with the existing question types is only a matter of creating a question list as shown in Listing 7.

Lesson Feedback

Checking answer

When the user validates a question by pressing the “Check” button at the bottom of each question screen, the answer is checked to see if it was correct or incorrect. The feedback is presented in either green or red with a check or a cross for either a right or wrong answer. The correct answer is always show after the question so if the user was wrong they can see the correct answer.

As the feedback is presented the next question is loaded to make the lesson streamlined.

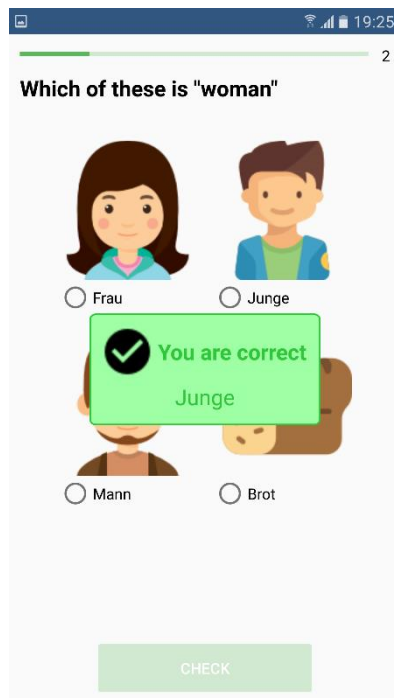


Figure 25.1: Correct



Figure 25.2: Incorrect

End of lesson

The end of lesson screen will display different images and messages based on the score the user receives. A higher score will display a happier image and congratulatory message while a lower score displays a sadder image with a more encouraging message.

The user is given the option to retry the lesson or return to the lesson selection screen.

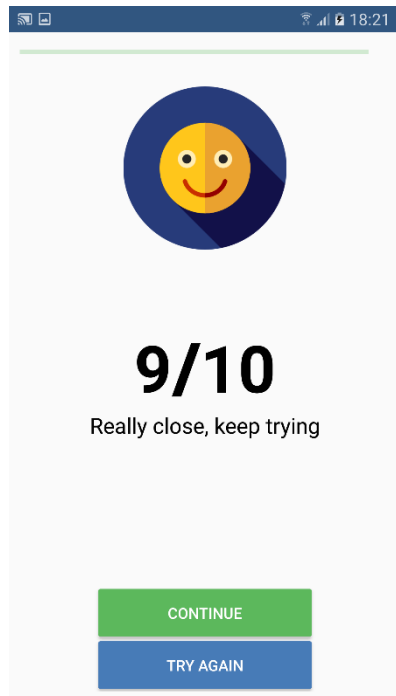


Figure 26.1: High score

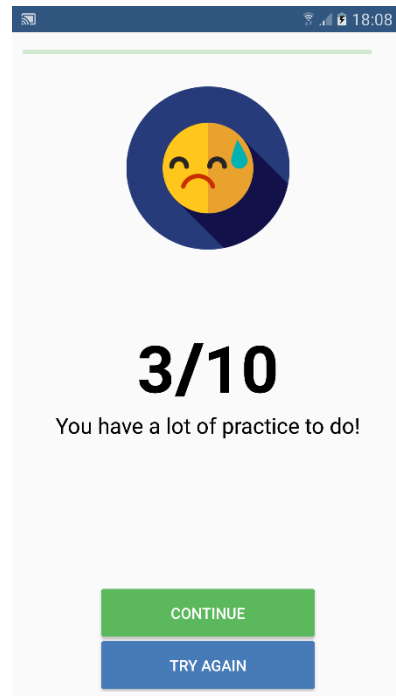


Figure 26.1: Low score

Testing

Testing was carried out using the Firebase Test Lab. This allows me to upload my application's apk and have it tested automatically for errors on multiple devices and multiple versions of android.

This provides a log of warnings or error reports should it fail any testing. In figure 27 we can see an activity map of the steps taken in a test of signing in, selecting a language and adding a new meet up performed by the automated testing.

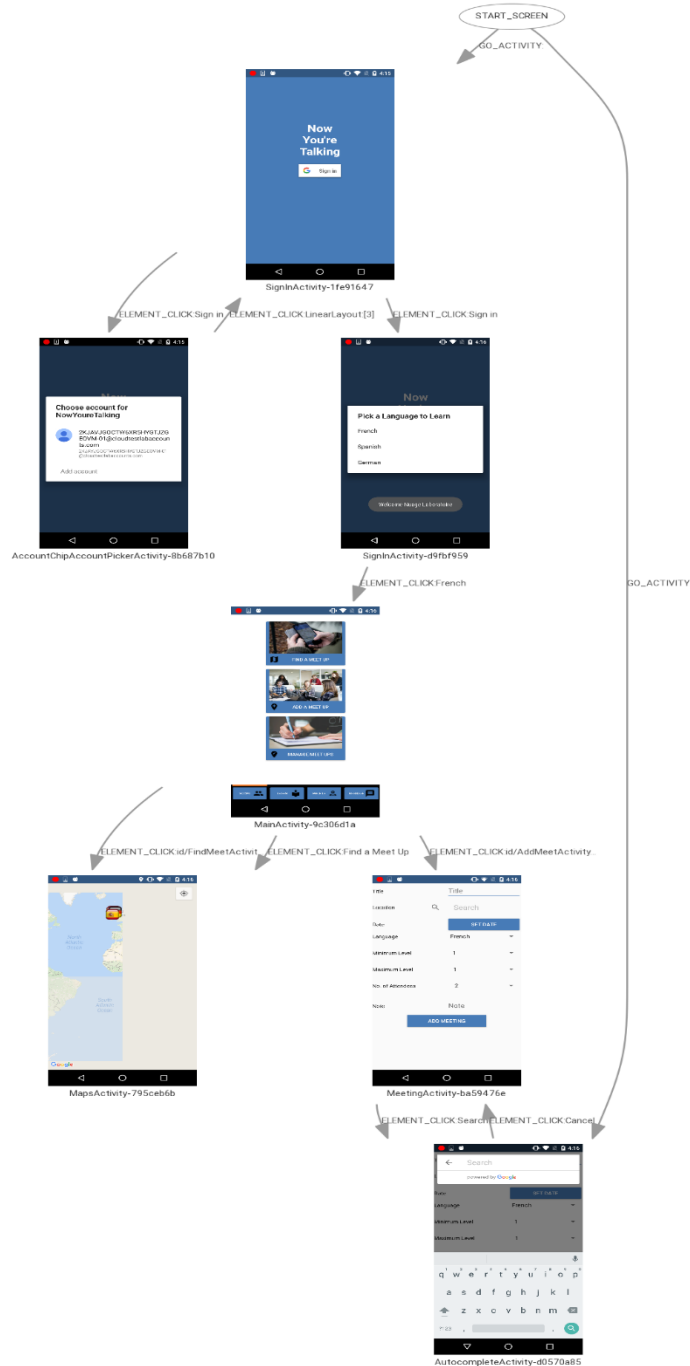


Figure 27: Test Lab

Once a test has completed I can view a video file of the test being performed and pinpoint any failures.

If a failure occurs I am provided a log file to identify the error (Listing 10).

Exception java.lang.IndexOutOfBoundsException: Invalid index 0, size is 0

```
java.util.ArrayList.throwIndexOutOfBoundsException (ArrayList.java:255)
java.util.ArrayList.get (ArrayList.java:308)
com.mismatched.nowyouretalking.CardPagerAdapter.getCardViewAt
(CardPagerAdapter.java:49)
com.mismatched.nowyouretalking.ShadowTransformer.enableScaling
(ShadowTransformer.java:33)
com.mismatched.nowyouretalking.LearnFragment.onClick (LearnFragment.java:166)
android.view.View.performClick (View.java:5702)
android.view.View$PerformClick.run (View.java:22541)
android.os.Handler.handleCallback (Handler.java:739)
android.os.Handler.dispatchMessage (Handler.java:95)
android.os.Looper.loop (Looper.java:158)
android.app.ActivityThread.main (ActivityThread.java:7229)
java.lang.reflect.Method.invoke (Method.java)
com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run
(ZygoteInit.java:1230)
com.android.internal.os.ZygoteInit.main (ZygoteInit.java:1120)
```

Listing 10: Testing error log

Evaluation

My application has been evaluated through testing via emulation on multiple devices including smartphones and tablet virtual devices. A log of current bugs is kept ([Bug Log](#)) to keep track of sections that will need to be focused on for future testing to ensure the application is free of known bugs.

The testing has been done manually for the most part, initially by myself, then by group of testers following a large update, that way all behaviour was tested rather than automated where real life actions are not taken into account.

The user interface was put to testers who gave feedback and recommendations based on their experience allowing for greater easy of use in terms of how the layout and navigation is presented.

Testing feedback

Figure 28.1 shows a prototype interface for one of the screens while figure 28.2 shows the final version of the same screen. Based on feedback from testers many changes were implemented such as business like images being replaced by more colourful and friendly images, updated navigation bar, a header bar showing the current section and immediate access to information on the next meet up.

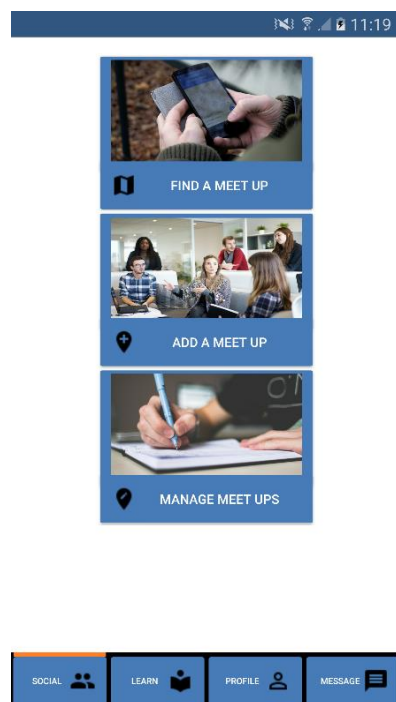


Figure 28.1: Prototype screen

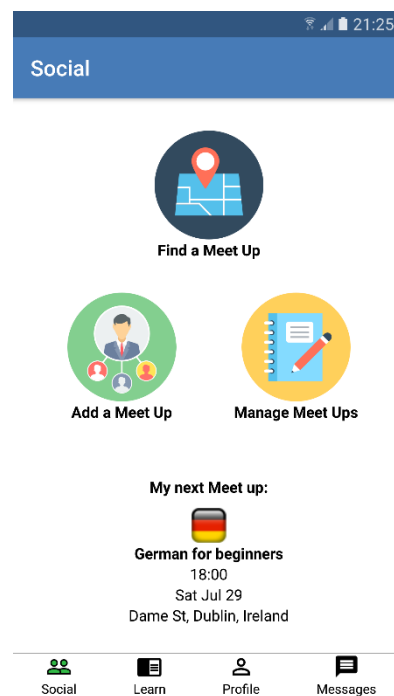


Figure 28.2: Final screen

Several other areas of the app were also changed based on user testing the most important being the lessons, this helped identify areas in which users struggled and correct some wrong translations.

Development lifecycle

My development cycle follows a build, test, deploy and enhance process. As a feature is added it is put through a brief testing phase to ensure correct functionality and continually built upon until nearing the end of the development cycle where the full testing phase will begin with users and test labs.

Throughout the applications lifecycle Firebase also offers crash reporting. This shows how many crashes happen on a given day and the number of users impacted by those crashes as well as a stack trace of the errors that happen. This helped to identify errors and crashes testers were experiencing.

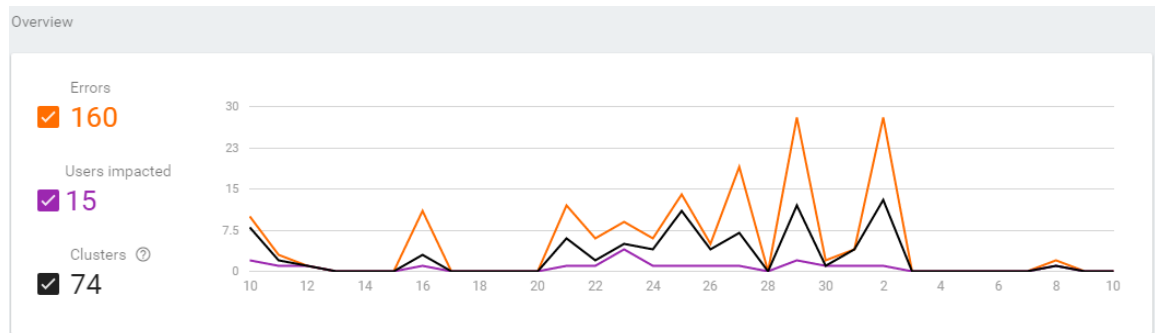


Figure 29: Crash reporting

I am also using Firebase's analytics that will show me information such as number of active users, device models and locations. This will help me see if I am reaching my target audience or if there is an unusual market that has an interest in the app that I can begin to target also.

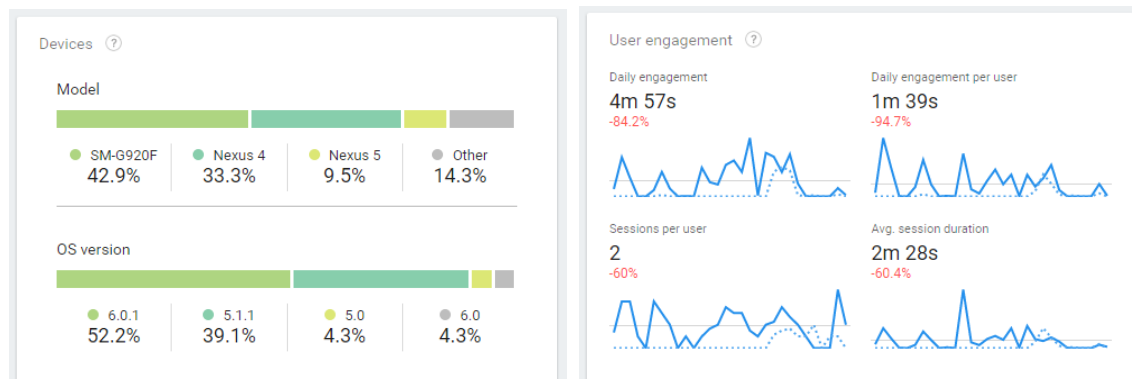


Figure 30: Analytics

Conclusions

The strongest advantage that this project has is its uniqueness in being a learning and meet up tool. Although there are several language learning apps available on different platforms I found that few have taken on a gamification approach, especially those marketed towards adults. The unique aspect of my project is the meet up and social functionality. Throughout the research process while looking to see if a similar product existed there were no language teaching providers found that also offers users a way to meet up and communicate in the real world. Group meet ups is a common occurrence for people learning a new language with many happening in both cities and rural areas, having the convenience of arranging these from the same platform they are learning on saves time and ensures that all users are experiencing the same progression levels.

However, it will be difficult to compete with well-known language learning companies as they have a brand, massive amounts of the target audience and the resources to continue to improve and expand their service.

Further development

With more time, I think this project has potential to be a global success. It is a unique idea with high potential. Additional help in areas such as design and marketing could improve the apps chances to be highly successful in the market place.

Future features

This is a list of functions that have not been implemented. Some of which have been in mind since the start of development and others which have been realised through the course of development and user testing but due to time constraints and availability of technology could not be added during the course of this project. These features would be included in the applications full release and give the app even more depth.

1. Voice over:
Voice overs will be done in the learning section to help the user pronounce the words they are learning. E.g. Counting to ten in the language selected.
2. Seasonal themes
During occasions such as Christmas and Halloween extra lessons will become available corresponding to the occasion.
3. Mascots:
Mascots will be used to provide information, in the form of pop up tooltips, on functionality, provide help, tutorials and show results for tests in the learning sections. This will also create a face of the brand and help with marketing the app.
4. Groups:
Groups / clubs will be added to allow users to easily invite friends to a meet up's they would be interested in. A user will be able to join multiple clubs for various languages and club admins will be able to send meet up invites to the members of the club.
5. Challenge / Versus game mode:
Users will be able to challenge one another to a lesson to test their skills against each other.
6. Nearby invitations:
To add the fear of missing out element users will be notified if friends are attending meet ups in the area.

Appendix

References

Research

Eurostat (2015). *Foreign language skills statistics - Statistics Explained*. [online]

Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/Foreign_language_skills_statistics [Accessed 18 Apr. 2017].

Marquis Ph.D., J. (2012). Game-Based vs Traditional Learning – What’s the Difference?. [Blog]

Available at: <http://www.onlineuniversities.com/blog/2012/08/game-based-vs-traditional-learning-whats-difference/> [Accessed 5 May 2017].

Salaberry, S. (2001). *The use of technology for second language learning and teaching: A retrospective*. [online]

Available at: http://salaberry.blogs.rice.edu/files/2013/10/Salaberry_2001_MLJ.pdf [Accessed 19 Apr. 2017].

Pandey, A. (2015). Top 6 Benefits of Gamification in eLearning - eLearning Industry. [online]

Available at: <https://elearningindustry.com/top-6-benefits-of-gamification-in-elearning> [Accessed 13 Oct. 2016].

TNS Opinion & Social (2012). *EUROPEANS AND THEIR LANGUAGES*. [online]

Available at: http://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs_386_en.pdf [Accessed 5 May 2017].

In app References

All images taken from flaticon:

<http://www.flaticon.com>

Maps and directions supported by Google:

<https://developers.google.com/maps/>

Translations being handled by Microsoft translate API:

<https://www.microsoft.com/en-us/translator>

CardView taken from - Rúben Sousa:

<https://github.com/rubensousa/ViewPagerCards>

CircleImageView taken from - Henning Dodenhof:

<https://github.com/hdodenhof/CircleImageView>

Meet up search results

In Ireland, when searching for meet up applications on google play store here is a list of the top ten results:

Rank	Name	Context
1	Meetup	Meet up
2	MeetMe: Chat & Meet New People	Messaging
3	Party with a Local – Meetup	Meet up
4	Mush - meet local mom friends	Mothers only
5	Badoo - Free Chat & Dating App	Dating
6	Tinder	Dating
7	Farmers Match Dating & Meet Up	Dating
8	Sex.Meet	Dating
9	POF Free Dating App	Dating
10	Hookup App!!! Hot Adult Dating	Dating

In app achievements

Number	Title	Goal	Difficulty
1	Beginner	Complete your first lesson	Easy
2	On my way	Reach Level 5	Medium
3	The Master	Master a language	Hard
4	Meeting new people	Attend 3 meet ups	Medium
5	Enjoying the company	Attend 10 meet ups	Hard
6	Night Owl	Complete a lesson after 10pm	Easy
7	Mix it up	Try more than one language	Easy
8	Getting social	Add a friend	Easy
9	Speak to me	Use the translator	Easy
10	Early riser	Complete a lesson before 8am	Easy

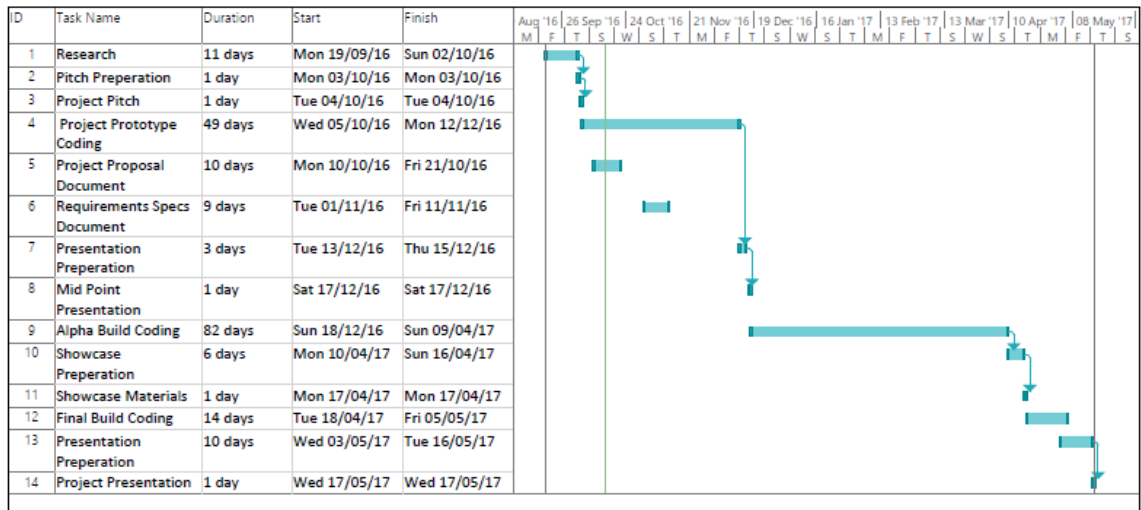
Bug Log

This is a list of known bugs in the app that are not application breaking but require work on for future releases

Number	Location	Description	Notes
1	ButtonQuestion	first word is not rewritten when tapped a second time	Issue is due to how 0 element in array is handled
2	MessagingFragment	Messages are not being sorted by date received	
3	Messages	Displaying user images cause a memory leak as they are loaded too often.	Not currently implemented due to bug

Project Plan

My project plan is based around deliverables for the project. This way I can easily manage not just the development of my application but also the documentation that goes with it.



Project Proposal

Objectives

Overview

This app will have two major functionalities. Firstly, it will help grow and develop language skills of the user keeping them engaged through the use of gamification. Secondly the app will promote socializing by offering users a platform in which can schedule meet ups and communicate.

Target Group

My target audience for this project will be people interested in learning a new language which is an area that will always have consumers, Marian (2016) provided a study showing the percentage of people learning a second language by countries in Europe. Ireland alone has 10% of the population learning a second language which is also one of the worst in Europe but this still translates as a potential audience of 400,00 users in a single country. If we look at a country that has a high rate of learners such as Sweden, 26% of the population are learning a second language which is the equivalent of just under 2.5 million potential users.

Motives

There are alternatives to learning a language, there are many applications and software for a range of platforms and devices that attempt to teach users a foreign language in a multitude of different ways. In my research (Pandey, 2015) I have found that systems that use a progression and rewards system has proved more fruitful than traditional learning techniques. As a matter of fact, some companies that have been in this market for years have begun to adapt to this approach of teaching. Users are more engaged when they are having fun rather than reading from textbooks hence my gamification approach to my project.

From my research I have also found that many people would also like to practice speaking a new

language with other new speakers as it is not as intimidating as speaking to someone more proficient in the language and is also viewed as a better way to learn over practicing by one's self. For example, on meetup.com one of the French learning groups in Dublin alone has over 3000 members to date who schedule regular meet ups. This is what drove me to include socializing as a key aspect of my project.

Unique

Although there are several language learning apps available on different platforms I found that few have taken on a gamification approach with Duolingo being the obvious exception as they were one of the first to take advantage of this method. Leaving a gap in the market for a competitor. The truly unique aspect of my project is the meet up and social functionality, there is little to no language teaching providers also offering users a way to meet up and communicate in the real world. Group meet ups is a common occurrence for people learning a new language with many happening in both cities and rural areas, having the convenience of arranging these from the same platform they are learning on saves time and ensures that all users are experiencing the same progression levels.

Challenge - Why is it challenging

Personally, this project provides many challenges to be overcome, previously I have developed only one android application in the past over a year ago so I am a bit rusty at android programming. There are also aspects I must take into account with this project that I previously wouldn't have such as real world data analytics and testing. Using github for version control is also something that I have no experience in doing. Presentations also carry a lot of marks for this project which is something I struggle with so will be a challenge for me to overcome my weaknesses in this area.

Background

As mentioned earlier in this report there are a large number of people across the globe learning second or even third languages. There has been a shift in the way people learn with the rise of personal technology over the past two decades, the availability of online courses and programs mean it is easier than ever before to learn without the use of books. This influenced me in choosing to create an educational application over another field.

The rise of social media and online forums have also influenced me in choosing to add social functionality. In today's world so much of our interaction happens across social platforms that it seems somewhat natural at this point that we communicate not only in person but through some kind of buffer should that be an app, website or messaging service.

Even with these buffers in our life most people still prefer to speak in person to one another which brought me to the meet up functionality of my app which provides a way to communicate in person.

Technical Approach

For my research I looked into many areas such as how people learn to speak a new language, why gamification promotes learning and how to implement it, how people communicate with each other online, what makes an easy to use interface.

I found most if not all of this information online and it has helped me get a more focused approach to how I will go about creating this application.

I chose my requirements based on the research I did as well as my own initiative to what will be the best fit for the project. Following my project pitch to a group of lecturers they provide some more input as to what would add some more complexity to my project, I have decided to implement some of the features that were discussed at the project pitch,

As for implementation, I have decided to develop my application using android studio as it is an easy to use IDE with useful GUI screen when designing an activity view. It also provides a mobile phone and tablet emulator that can be used for testing an application of multiple devices. As well as that it has built in integration with Github allowing me to easily manage version control of the project.

I will be using Google Firebase in my project as it provides several features and is offered for free. I will be using it for user authentication and profile building, cloud database storage and possibly an in app messaging service. It also allows for data analytics to view certain information that the application owner may want to use to improve their application such as demographic using the application, number of in app purchases and user engagement. Firebase also offers a notification system, crash reporting and test labs among other services.

Special resources required

This project will require users to use an android device with minimum android version 4.00 operating system and internet access for use of all features.

Potentially the project could be scaled to iPhone, iPad, Windows phone devices and web browsers. I will not be doing this during this project but I plan on developing the application with this scalability in mind.

Technical Details

The application will be developed using Java and native android. I will be using google api's such as the sign in api, authentication api, maps api and firebase api.

Evaluation

My application will be evaluated through rigours testing through emulation on multiple devices including smartphone and tablet virtual devices. I will keep a log of know bugs to keep track of sections that will need to be focused on for future testing to ensure the application is bug free. The testing will be done manually that way all behaviour will be tested rather than automated where real life actions are not taken into account.

The user interface will be put to testers who will give feedback and recommendations based on their experience allowing not only a nice to look at platform but a more streamline app.

However, testing will not just happen once the app is nearing completion. My development cycle will follow a build, test, deploy, enhance process as a feature is added it is put through a brief testing phase to ensure correct functionality and continually built upon until nearing the end of the development cycle where the full testing phase will begin.

References

Marian, J. (2016). *Number of people learning a second language by country*. [online] Jakubmarian.com. Available at: <https://jakubmarian.com/number-of-people-learning-a-second-language-by-country/> [Accessed 13 Oct. 2016].

Pandey, A. (2015). Top 6 Benefits of Gamification in eLearning - eLearning Industry. [online] eLearning Industry. Available at: <https://elearningindustry.com/top-6-benefits-of-gamification-in-elearning> [Accessed 13 Oct. 2016].

Monthly Journals

Reflective Journal - September

Student name: Jamie O'Neill

Programme (e.g., BSc in Computing): BSHC

Month: September

My Achievements

During the first week of back into college I came up with the initial idea for my project. My project will be a mobile application for android devices used to learn and develop language skills as well as scheduling meet ups with fellow learners to create a cooperative learning platform.

I began by writing down rough ideas and features of the application, rough sketches of what the screens might look like, what will be stored in the database, where pieces of data will be stored (Internally / Externally), the scope of the project and ideas of the user interface.

I created a wireframe of the initial flow of the application, this will most likely change in some ways as the project develops but I am hoping to keep the core of it as a template for the work I do here on out.

I looked into Google Firebase as a tool to assist with handling database and data analytical processing. Once I did a tutorial and developed a simple project using Firebase I realised that it is capable of a lot of features that I had not originally intended my application to do but with the firebase tool I may incorporate them into my project to add an additional layer of depth. I am fairly certain I will use this tool in my project and will decide which of the features I want to include before I do my project pitch to the lecturers.

My Reflection

I Feel I am off to a good start for the first week back in college as I was trying to think of project ideas throughout the summer and have finally decided upon one that I feel will be useful for users and fun for me to make.

I haven't touch on android coding for about a year so I am really looking forward to coding once more. I think in terms of challenge this project is a good test for me as I want to include many elements and features android is capable of that I have yet to use.

I realise that the project will consume a lot of time thankfully I really enjoy working on projects such as this that I can choose the subject matter and work individually without relying on other students to complete the work.

Even work that I might have found tedious before, such as documentation and reports, I am looking forward to doing as the project is completely my own and I will hopefully have something that I am proud to say I built from the ground up

One fear I have of the project is my ability to present the project as I would like. I am not the best public speaker and always feel intimidated presenting, should that be to an entire room or just to 3 lecturers. I am hoping that my passion for this project will give me a little bit extra confidence and make me more comfortable in presenting my ideas.

Intended Changes

With the project pitch scheduled for the 5th of October I am hesitant to begin coding the project until I receive the green light. I will begin preparations for the pitch and try to come up with a way to best get my project across to the lecturers.

In the meantime, I intend to do more research on my ideas to see what has been done before and what new elements I can add. I also will do a few tutorials using GIT as I have not used online repositories before. Finally, I plan of learning more of Google Firebase as I feel that can really improve my project.

Supervisor Meetings

Supervisor meetings will not take place until the project idea has been approved following the pitch.

Reflective Journal – October

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: October

My Achievements

At the start of the month I gave my project pitch to 3 of the lectures in the college. My idea was approved with revision. Even though my project was approved my fears of not being able to present myself well were justified. The lecturers had liked the concept but worried that the project may not be complex enough and made a few suggestions to give it that something extra which I will take on board and try to incorporate should they be fitting. However, I feel they judged my presentation very harshly as I couldn't get my idea across which I realise is my own fault but I feel the scope they were aiming for is unrealistic due to my finite time and managing of other projects.

Before I started coding my app I created a github repository that I will use throughout its development for version control. I also created a Firebase project through the firebase developer console as I have decided I will use this for user accounts and cloud database storage.

I began coding my application firstly by implementing a google sign in feature then connecting that to my Firebase console this allows me to view all users of the application and send group emails or push notifications to the users or a subset of users. Once that was working correctly I created a user profile activity which pulls in the user's name and email address for now I will flesh this out and add more features to this section at a later point. I also added in database read and write functionality, right now this is only for testing purposes but be important as I begin work on the other features.

I also finished and submitted my project proposal document on the 16th, five days before the deadline. It took me about 4 days to complete. I am somewhat happy with the document but I believe we can make future changes to it before it is marked which I certainly will do to make it more in line with my final project.

For that document I also created a Gantt chart which should prove to be very useful and I plan on referring to it to keep me on track to complete the project and all documentation on time.

I have begun coding the functionality for handling the in app meet ups, I now have a meet up being created and added to the database, I will add a user input to get this information and will later display these on a map for all users to see.

I added the user interface for adding a meet up and it is working correctly with the database, right now it is still a rather ugly user interface but I will clean it up towards the end of the project.

I have added a map activity that is reading in from the database. The map updates in the background when there is a change in the database in real time. Currently the map pulls the address the user entered when setting a meeting location then converts this to latitude and longitude then places a marker on the map. Right now it only displays the location and title of the meet ups happening but eventually each marker will display more information about each meet up when it's respected marker is clicked and give the option to join that meet up.

I have been looking into using the Google Places API but after testing it in my app I decided the functionality does not quite fit into how I want my app to be so I will not be moving forward with using that API.

All markers on map now show the information for each meet up. I have customized each marker colour depending on what language the meet up is for, I think I will at some point change the default marker altogether to be something like a country flag to make it look a bit nicer.

I also added a "join" button that does nothing right now but this will add the user to the meet up selected.

The user attending function has been changed to allow multiple users to attend. If the amount of users attending has been maxed the meet up is no longer displayed on the map.

I have added a calendar view and time picker to the meeting activity. Meeting time now reads into to database and displays on map.

Meetings now must have title, location, date and time if they are to be added to database.

I have upgraded the build tools to version 25.0.0, I also upgraded google play services and firebase plugins to version 9.6.1. I did this because the newer version of the plugin gives me the

option to add objects to the map markers, which is how I handle the “Join” button functionality. The “Join” button on the maps activity now marks the user as attending for the meet up selected. The “Join” button will only add users who are not already attending so there can be no duplicate users in one meet up.

Markers now display the country flag of the language selected instead of the default google maps market. This just adds more ascetic appeal.

The maps activity now asks for GPS location permission and sets the map view based on the user’s physical location if they agree to allow location permission.

My Reflection

I believe I have made great progress in my project over the last month. I have set up my Firebase and Google consoles as well as a Github account. In my app its self I have added authentication, user profiling, database reading and writing, meetup functionality, maps with correct data. All in all, I consider that to be a good month.

Not only have I added these features but I am also learning as I go so it’s not as if I can easily throw in every functionality I want. I must spend time researching, viewing documentation and guides then work out how to implement these features to suit my own needs which is not a fast process.

I feel that if I can continue the rest of my project at my current rate I should have a very good project by the end of the year.

Intended Changes

Because I am still very early in development I cannot really list intended changes as there is so much to do. I am still working my way through adding all my intended functionality.

I have decided to focus on my app in 4 stages:

1. Social functions
2. Learning functions
3. User interface & design
4. Final testing

When a stage is approaching completion I will evaluate it and list any intended changes in this section of future journals.

Supervisor Meetings

I have contacted my supervisor, Dominic Carr, and will have my first meeting with him after reading week at the start of next month.

Reflective Journal - November

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: November

My Achievements

I have moved on from my maps activity and will focus on my user management of meet ups. I started by removed testing buttons from main activity as they were no longer needed. I then added a manage activity to view all meetings the user is attending. Currently this displays all the meet ups the use has either created or have pressed join in the maps activity. I added dialog box on click of meetup text view right now this is displaying the meet up information but I will change the look and functionality of the dialog box later.

The "Remove" button on the manage meet up activity now removes the user from selected meet up. This was extremely frustrating as I spend over 4 hours on this one function when it turned out to be as simple as adding another for loop to go through the currently attending users, getting their unique key and matching that key with their database entry.

I updated some of the button names around the app to more meaningful names while before they were just labelled like button1 for testing purposes. I have changed the default colour scheme in the app to a light shade of blue for the primary, dark blue for secondary, and a light red as third. I also changed the background of the screens to the new primary colour but I think it does not look good on the eye so I will change it back next time I am working of the code.

I have completed my requirements specifications document a week before the due date, I am fairly happy with it and so is my supervisor who I sent it on to for review. I have been informed that we can come back to this at a later date but mine already includes functionality, use cases and wireframes to name a few so I feel not much more will need to be added.

I have reverted background colours back to white as I felt blue just didn't work. I also removed the logo.png from the sign in screen to be replace by text. The sign in now welcomes user by name when they login. User profile information can now be read from any activity without duplicate code this is just a general thing I should have done from the start but didn't for testing purposes. Strings are now all stored in strings.xml as is the industry standard. Turning on GPS location is now handled in app rather than starting a settings intent this speeds up the application and ascetically looks better. Updated map animation if GPS is turned on, the map now zooms and snaps to user location rather than automatically being at the location I feel it makes the map look more dynamic.

Added a toast to tell the user to click zoom button if they turn on GPS from the request dialog. This was the simplest way to achieve zoom after turning on GPS in this was as I was struggling to get it done programmatically. I have added images used for animation to the project.

An animation of the app's mascots now plays on home screen. These mascots will be used throughout the app to serve as helpers and give information on functionality.

I have changed the time and date pickers to show in a single dialog box when adding a new meet up. This is done to make it easier for the user to select both at the same time and also looks much better than before.

Buttons are now all the same style using my primary app colour as a background and white text, this gives a constant look and feel to the app. I have also added red button style that is used for cancel buttons. I have cleaned up redundant code on maps and meeting as I can now call the methods I need from another class.

Map marker now shows new dialog box when clicked instead of info window, I feel this is a massive improvement visually.

The map will not show meet ups if user created them as they do not need to see their own meet ups here when they are already attending. This speeds up load times on the map.

Manage meet ups now split meet ups based on if the user created it or not so the user can easily see which they are the owner of and which they are attending. Users can now delete a meetup entirely from the database if they created it.

I have changed the dialog box in manage activity to allow for more information to be shown about the meet ups.

Location field is now filled out from auto complete google places API when adding a meet up. This is a vast improvement over the previous way as it now only auto completes but assures the location will be added to the map.

Changed format of time/date picker to use location default formatting to fix bug where app would crash on any android device below version 6.0.

I have added a API version check to time/date picker that completely resolves previous crashing on devices below 6.0. Joining meet up send the user to manage activity to view results. added back button control on main and manage activity for more fluid control for the user.

I have reverted the map back to display all markers, the markers user is attending now show in grey to signify they are attending these while view the map. Removed unused buttons from map also.

With that I have finished my clean-up of the first section and will be moving on to adding more functionality.

I have added a translation activity that allows the user to input text and translate it from one of the languages available to any of the other languages. I did this using Microsoft's translation API that allows my app to translate 2 million characters per month.

I've Added text to speech functions on translate activity, this uses the built in android text to speech functionality, all languages supported now have individual speech voices to adjust for accents. I also added a circular button style that I am using to round the edges around my play speech button, I may use this style for multiple items around the app at some stage. Added catch to make sure location is entered on add a meet up as this was threw an error if nothing had been entered, this bug only occurred after I added the auto complete function.

Added directions functionality to map makers and manage meet ups, when a user clicks the directions button it loads the google maps application with the location of the meet up they have selected pre-set. I changed "join" meet up button colour to green to make it more obvious to users.

Added new navigation buttons with icons to main activity, this makes the app easier to navigate and more in line with material design. This makes the main activity use fragments to display different sections of app. Added percentage support dependency to allow the navigation bar support all screen widths. Added slider and animations to main activity to highlight current fragment view on the navigation bar.

Removed all mascot animations as I felt they were too cartoonish for my app as I want a slicker feel to the design. Removed title from home screen as it seemed redundant. Added new images and icons for social fragment view. Changed navbar to be fragment this will allow it to be used in any activity if I want to add it to different sections.

Added messaging activity and message class. Users can now send messages in real time from one user to another if set up in database first. Added a message button to navbar with new icon to easily launch the messaging function. Changed layout and animation on navbar to accommodate the new message button.

My Reflection

I am very annoyed over the length of time the remove from meet up function took me to figure out. Due to Google Firebase still being relatively new the support for it is not the same as higher profile platforms. On one hand this does make for a more challenging experience and a greater reward for figuring a problem out on my own but on the other hand I have wasted a large portion of the night on just one issue.

I have also been playing around with logo's but have not decided on anything yet and may just use plain text possibly with a mascot of some kind that could be animated in certain sections of the app giving the user hints and tips. I like this idea but feel it could take up a lot of my time as I know nothing about animation but I will consider it and may come back to it at a later point in the project if time permits and if it fits the rest of the app's style.

I am pleased with my progress for this month I added plenty of new features, fixed a lot of bugs and cleaned up a good chunk of the user interface. I also feel my grasp of coding has improved a great amount since I have started the project, I find myself not looking up how to accomplish tasks as much as I can generally figure it out for myself. I think I am well ahead of completing the application before the due date if I can keep up my current pace and provided other assignments don't take up too much of my time.

Intended Changes

I need to add a way of users to be able to start a chat in app without being set up in the database first. I also have some more user interface work to do before I'll be happy with the design.

I would like to wrap up all of the social aspects of the app before the mid-point presentation in mid-December if I can then move on to the learning sections after the presentation.

Supervisor Meetings

I have met with Dominic Carr and gave him a run through of what my app is currently capable of. He was pleased with my progress so far and happy to let me work away on it.

I will consult Dominic if and when I get stuck on something as well as when I add a new significant feature for him to view and offer opinions on how to improve it.

Reflective Journal - December

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: December

My Achievements

After adding the messaging feature last month, I had some bugs and layout issues to fix with these activities'. Such as text clearing after a message is send, the view going to bottom after sending message and changed margins of messages displayed.

I have added the ability for users to upload profile pictures. The profile picture is displayed on profile fragment of the main activity. There is a default image displayed if no image has been uploaded by the user.

Added image save to and load from device methods in userprofile fragment, this allows me to store images on my backed storage as well as the devices catch memory. User profile images now show on MessagingActivity next to user's who have an active conversation.

Messages now go to bottom on start-up and when new message comes in. Added message sender's name to message text. updated message text layout to be more agile on multiple devices.

I Updated the launcher icons. set portrait Orientation throughout app as coding landscape mode is not necessary for my application. Changed social fragment images and layout of social fragment to make it more appealing. I also decreased the height of the navbar and did a general small code clean up. As well as rescaling images on social fragment for faster load times and removed unused images from the project.

My Reflection

I think I have decided on the final layout and style for the application. I have only implemented in in one section of the app so far but it is comforting to know the further design for the rest of the application.

I did not spend much time coding this month as I was pre occupied with the midpoint presentation.

The presentation went well I was able to clearly demonstrate the current state of my project and received feedback that I can incorporate further on down the road.

Intended Changes

My current list for changes so far consists of:

- add continuous check for internet connection.
- add ability to start message.
- add set image on first sign in.
- set nice design on messaging activity.
- add next scheduled meet up on social fragment.
- add speech to text on translate activity.
- move old meet ups to "Past meet ups" table
- add statistics about past meet ups attended.
- add badges/achievements.

Supervisor Meetings

My supervisor gave me feedback during the midpoint presentation that I will add to the application at a later time.

Reflective Journal - January

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: January

My Achievements

Due to Christmas holiday break and having exams in mid-January I was not able to get as much practical project work done as I would have like I instead mainly worked on documentation after the exams.

My Reflection

I have been thinking of ways to implement features listed in my previous journal entry and have come up with a solution to certain features which I will implement during February.

Overall I am disappointed at the lack of practical work done for this month but studying for exams took priority over project work. I plan on getting back on track for the next month

Intended Changes

No changes from last report

Supervisor Meetings

No meeting due to break and exams.

Reflective Journal - February

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: February

My Achievements

This month I was able to begin coding of the learning section. Following my design document, I could implement it almost identically to the documentation.

I began by set up images for selecting a lesson and the initial creation of the game class. I also moved the translate button to user profile fragment as it was not relevant to the lessons.

I set up the game xml for 3 types of question: imageQuestion, buttonQuestion and translateQuestion. Then I set up game activity to read in questions based on lesson and language selected.

The first set of questions I got running was the image questions. I then added all the questions for the German basic lesson as this would be my starting and testing point till the game activity was working for all questions.

Next I added button and translate questions to the lesson and set the check answer button to disabled until an answer is selected. I also made the position of answers on screen random each time the lesson is loaded.

I set the keyboard to hide after translate question has been entered. I included a result screen after each question that will show if the user was correct or wrong and added an end of lesson screen but this will require more design work.

I added an imageHelper to get images for imageQuestions regardless of what language the user has selected. Once the lesson ends I added a try again or continue option after game.

I also set the lesson score to save locally on the device after game this will be used later to determine how far the user has progressed in each lesson. Progress bars now display when selecting a lesson to see how much of each lesson is complete. added star beside the lesson name if the lesson is completed 100%.

I enabled data persistence across the app. This means that once the user has signed in with an internet connection they can they continue to use the device off line and updates to the database will happen once the user regains an internet connection.

Now that the lesson is working I added "basic" questions for Spanish and French as well which all work perfectly for the "basic" lesson. Each language now saves its own lesson score at the end of the lesson meaning a user can now learn multiple languages on one device.

My Reflection

I have made great progress this month to compensate for the lack of work during January. I am very happy to have implemented most of the learning section of that app and to do it in such a way that it is scalable to multiple types of questions and languages.

One problem that I still consider is my user interface I am still not happy with it but I am stuck in a rut in terms of knowing what to do with it. I will take my app to some testers to determine how they would want it to look and navigate.

Intended Changes

To do list:

- add continuous check for internet connection. (may not be needed)
- add ability to start message.
- add set image on first sign in.
- set nice design on messaging activity.
- add next scheduled meet up on social fragment.
- add in friend feature
- add sign out button
- make end of quiz look nicer
- disable buttons on learn screen, unlock as level goes up
- add Q's for 2 more lessons
- add 2 more game types to lessons
- change game to take 3 lessons (possibly)

Supervisor Meetings

I met with my supervisor to get advice in terms of scope of my lessons. It was very helpful and we agreed to keep it relatively limited, perhaps 3 lessons and 4 or 5 game types in the lessons for 3 of the selected languages. We discussed different types of game modes such as sound or video questions but what affect that will have on accessibility may be drastic. There could be a work around this such as disabling those types of question which is something I will come back to once all lessons and game types are in place.

Reflective Journal - March

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: March

My Achievements

This month I focused on tidying code, fixing bugs, performance improvements and upgrades to current features.

I have updated the user experience by allowing the user to sign out of their account at any time if they wish to change accounts. I also allowed the user to change the language they wish to learn from the menus rather than needing separate accounts.

I updated the user interface for the messaging section adding a tabbed view with messages and friends separated. Users can now also add friends through their email address. I also added a

way for users to start a new conversation with friends which was always the goal but I just haven't gotten around to adding until now.

Fixed a bug when adding a new meet up the maps activity which was trying to read in the "Attending" field which is empty because it is added after all the other fields. This caused the app to crash. To fix this I changed firebase `addValueEventListener` to `addListenerForSingleValueEvent` in maps activity.

Added speech to text functionality to translate allowing users to speak into the phone to get the translation. Added check for internet on maps and translate as these features need an active internet connection to function correctly.

Past meet ups no longer show on the map and old meet ups now move to their own database reference. I then added a `getMeetingStats` method which returns stats for the user some of these stats will include information on past meet ups hence why I store the past meet ups in a separate database location.

Next meet up now shows on social fragment so the user can quickly see when the next scheduled meet to is and its location.

Added stats activity that reads in the number of previous meetups this will hold more stats at a later point.

I added splash screen when the app is loading as previously it was dull and looked like nothing was happening. This is only a temporary image in place for now and will be updated down the road.

My Reflection

I am happy to finally catch up on some of the items on my to do list. I managed to check off 10 items from last months to do list although while doing so I have added 6 new items to the list that will need to be done.

Overall progress on the app continues to steam ahead. As for next month, I will change focus back to the learning section at least to begin with as I want to get that part tied off then will continue to work off the to do list.

Intended Changes

To do list:

- add set image on first sign in.
- make end of quiz look nicer
- disable buttons on learn screen, unlock as level goes up
- add Q's for 2 more lessons
- add 2 more game types to lessons (match the words, fill the blank, unjumble the sentence, identify the misspelled word)
- change game to take 3 lessons
- sort messages by date received
- add statistics about past meet ups attended. (working on, need more stats)
- add badges/achievements.

- change translate game to accept all text case's.
 - Show level progress on app start.
 - change splash screen image / get new launcher icon
-

Reflective Journal - April

Student name: Jamie O'Neill

Programme: BSc in Computing

Month: April

My Achievements

April was an extremely important month for this project as I needed to add the largest chunk of the learning section, freshen up the layout, add in achievements and make and final changes before the project is due in May.

I started by changing the app icon and splash screen then added initial matching words game. Next I added toasts to matching game and added new images for toasts as well as changing background layouts for toasts.

I added card views when a lesson is selected allowing for multiple lessons in one topic. Changes to naming convention for lesson strings and removed unused layout files.

Added new lessons to the basics games. Updated image helper and strings for lessons. Buttons now update in button and matching games.

I added fill in the blank game and translate image game. There were bug fixes to some games and I added all German basic questions. I went on to add all basic questions for French and Spanish as well.

Progress now shows on lesson selection screen. Bookmark updates when user 100% a lesson. Level select now uses the sum of the lessons to fill the progress bars.

Updated images on level selection screen. language levels are now individual. Level is set automatically for different languages. added facebook.stetho to view shared preferences. Lessons now unlock based on user level. Level now increases if users 100% 3 lessons. Added toast after levelling up to show user. Added shadow to card of the current lesson selected.

Updated cardview to have deeper shadow. added credits to Rúben Sousa's code. Changed conditions that unlocked lessons are displayed at. Added "phrases" questions for all languages. That was the majority of the gaming section completed and I could now move on to implement the achievements.

Updated images on level select screen. Added AchievementHelper class to display achievements on profile page. Added code to unlock all achievements apart from "master" achievement which cannot be unlocked at this point.

I added an app bar, changed layout of social fragment. next meet up now hides if there is none. removed stars from learn screen.

Fixed some wrong answers for French lessons. result of question is now in a toast. Removed border from next meet up. Language is now stored locally (speeds up load times). Added import to round out profile images. Fix for translate achievement not unlocking. Adding images to git not previously included.

Added images on result screen. Layout changes to add meet up screen. Small changes to timepicker layout. Added language to profile screen. Changed game to read local language for improved speeds.

Updated entire navbar including images. removed scrolling animation.

My Reflection

I am very happy with the progress made in the final month of my project. I added many of the features that were intended from the offset. As well as all of the coding progress I also finished off my showcase poster.

I consider the app as I envisioned it when I first came up with the idea. Although throughout the course of the project I have come up with several new features I think would improve the app but wouldn't be able to implement due to the tight timing I had already set for the features I had from the start.

Intended Changes

None, end of project.