

|  |  |  |
| --- | --- | --- |
| Project Title | | |
| **Student Name**  **Jonatans A de Souza** | **Student ID**  **C14442238** | **Supervisor Name**  **Tanya Thompson** |

**Table of Contents**

1. INTRODUCTION 3

A. Overview of the project and the background behind it.

B. Project Objectives

C. Project Challenges

D. Structure of the document

2. Research 4

a. Background Research 4

b. Alternative Existing Solutions to Your Problem 11

c. Technologies Researched 13

d. Other Relevant Research Done 23

3. Design 29

A. Methodology………………………………………………………………………………………...

B. Technical Architecture Diagram……………………………………………………………

C. Other Design Document…………………………………………………………………………

5. Design 33

a. Technical Architecture Diagram 41

b. Other Design Documents 43

6. Prototyping and Development 48

7. Testing 61

8. Issues and Risks 63

9. Plan and Future Work 63

10. Conclusions 66

# Project Statement

The goal of this project is to develop an English-learning software called VocabNote. It will be designed for those who wish to learn English more efficiently and effectively.

Learners often write down newly learned words into a notebook and typically don't practice those words very regularly to be able to add them to their vocabulary.

Trying to memorize new vocabularies from a notebook is usually ineffective and dull. This project will provide games and Gamification techniques such as badges, leaderboard, perfomance analysis, to motivate and to offer a more entertaining and dynamic learning experience.

This project can be beneficial for English classrooms and students learning from other apps such as Duolingo and Babbel to complement the lesson and allow students to practice their newly learned words more effectively.

Combining games with learning has been proved to be an excellent strategy for learning a language by stimulating the brain’s reward system and releasing the feel-good hormone, which is Dopamine.

# Research

## Background Research

## Learning a foreign language

Learning a foreign language can be an arduous and intimidating process; however, the benefits of it, are far more significant than the effort made by the brain to acquire it.

Think of it as working out in the gym, bicep curls, leg press, among others exercises, they all aid on muscle growth, with practice and repetition, the practitioner will get stronger and agile. The same occurs in our brain when learning a language, it is, in fact, an excellent mental exercise to enhance our cognitive skills and increase intelligence.

Many years ago, experts thought that raising bilingual kids was believed to hinder their development and lower their IQ; however, further studies conducted by Peal, Elizabeth Lambert, Wallace E in 1962proved the contrary. They have discovered that being bilingual does not hinder the overall development, the opposite occurs.

They learned that bilinguals performed better than monolinguals in 15 verbal and nonverbal tests.[[1]](#endnote-1)

Studies conducted by Thomas H. Bak MD, Jack J. Nissan Ph.D., Michael, M. Allerhand Ph.D. and Ian J. Deary proved that speaking more than one language had significantly improved cognitive skills and can also slow down brain aging.[[2]](#endnote-2)

People who speak more than one language are more likely to perform better at math, vocabulary and reading exams,

They have better decision-making skills, creativity, focus, better at memorization skills among others.[[3]](#endnote-3)

## English as a second language

English is recognized as the lingua franca of business and academia.[[4]](#endnote-4)

85% of multinationals use English as their official language.[[5]](#endnote-5)

It is the third most spoken and is considered as the most influential language in the world.[[6]](#endnote-6) It is the most studied language worldwide.[[7]](#endnote-7)

The use of English has been growing substantially; It is used as the primary language for E-Commerce.[[8]](#endnote-8)

According to Babell, 1.5 billion people speak English, only 360 million are native speakers[[9]](#endnote-9).

It is clear that speaking English is very beneficial for having a career advantage. Job opportunities will rise, international communication can be improved, and exploring the world can become more convenient.

There are many ways one can learn English, watching tv shows, listening to podcasts, reading a book, starting a conversation with a native speaker, among others. The key is to find a way where one can be thoroughly immersed in the learning process, away from distractions and having fun to be effective.

The level of comfort dictates how well information is transmitted and stored in the brain. Students who are immersed and motivated tend to learn faster; information is transmitted with ease through the affective filter in the amygdala, resulting in higher level of cognition and decrease the learning curve.[[10]](#endnote-10)

Stress can gradually increase when students are not being motivated and enjoying the subject being taught due to lack of energy and dynamic.

Cognitive Psychology studies have discovered that stress, lack of motivation, anxiety, boredom, and confusion can hinder students from learning.

In stressful conditions, information is blocked from accessing the brain's areas of higher cognitive memory consolidation and storage, which can prevent the learning process.[[11]](#endnote-11)

A gamified approach can be used to aid the learning process.

## Gamification

Gamification is the process of implementing game elements and digital game design techniques to non-game contexts[[12]](#endnote-12), such as education, business, and Healthcare to increase motivation, engagement, entertainment, and learning.

Many studies in education and learning contexts have found that gamification has mostly positive effects when used for learning.[[13]](#endnote-13)

One study conducted by Sze Lui, LAM 2011 stated that students favored the use of gamification technologies when learning vocabulary, for the reason that it was more entertaining, and facilitated vocabulary retention. Gamification improved the attitude of students toward learning. [[14]](#endnote-14)

53 percent of technology stakeholders stated that by 2020, the use of gamification will grow extensively, It will change the communications scene, it will be implemented in various ways for work, education, and other aspects of human life.[[15]](#endnote-15)

The global gamification market is expected to be worth $11.10 Billion by 2020.[[16]](#endnote-16)

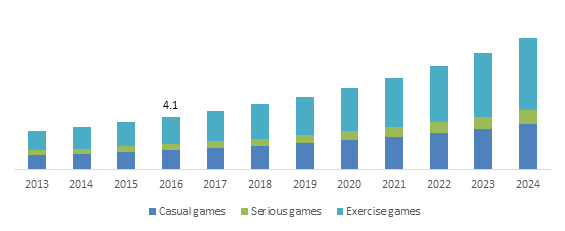


Figure 2: Global Market Insight[[17]](#endnote-17)

Gamification has also become very popular in the Healthcare sector. According to Global Market Insight, Healthcare gamification market is predicted to have a massive growth, due to the popularity of gamification, which is rising. It is expected to be worth $40 billion by 2024.[[18]](#endnote-18)

Disease prevention applications using gamification techniques

are expected to increase, due to the growth of players competing for fitness, supporting the development of many apps which compare performances. [[19]](#endnote-19)

## Gamification techniques

1. Rewards

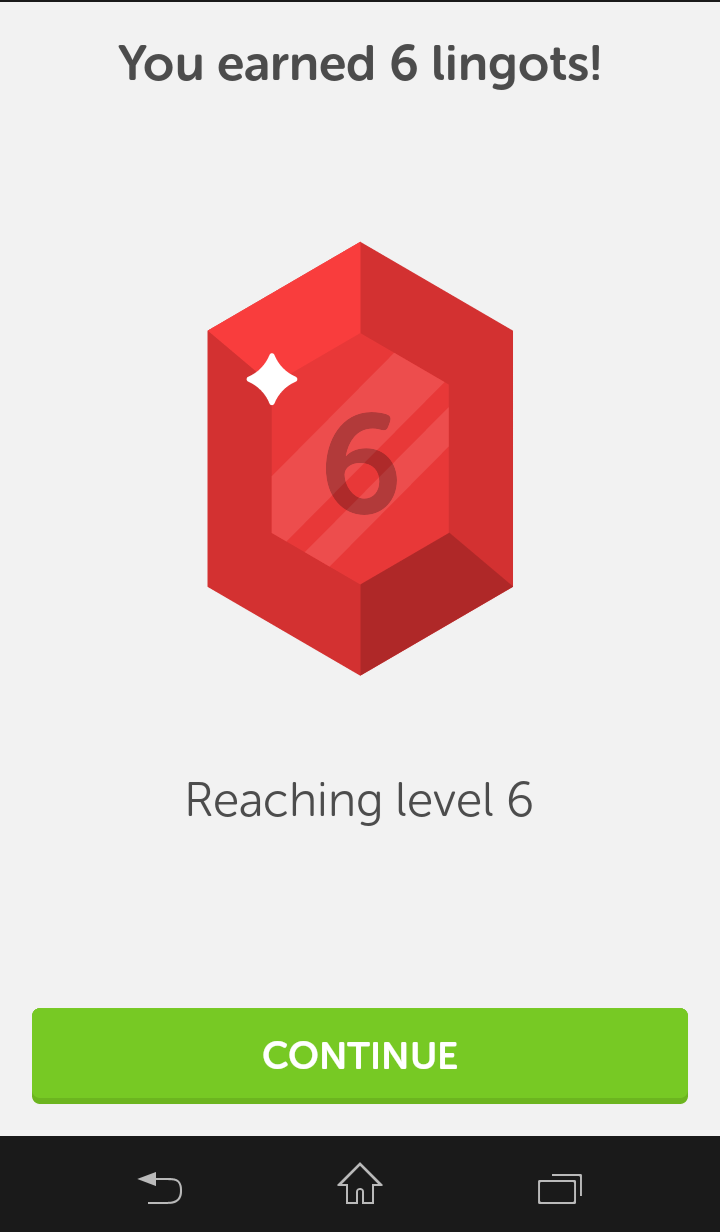
Rewards such as badges, trophies, medals, among others are used by many applications that apply gamification techniques.

Rewards can trigger the release of serotonin which is a hormone that commands our mood and Dopamine.[[20]](#endnote-20)

Duolingo uses a virtual game currency called lingots as a reward, where learners can earn it by finishing lessons, leveling up and having long streaks. Learners earn lingots everytime they finish a lesson.

Lingots can be used on Duolingo store to buy items such as heart refill which permits learners to pass difficult lessons, streak freeze which allows learners to keep their streaks after days of inactivity, among others.

Duolingo uses another form of reward which is badges; learners can earn them when they achieve something within the game.



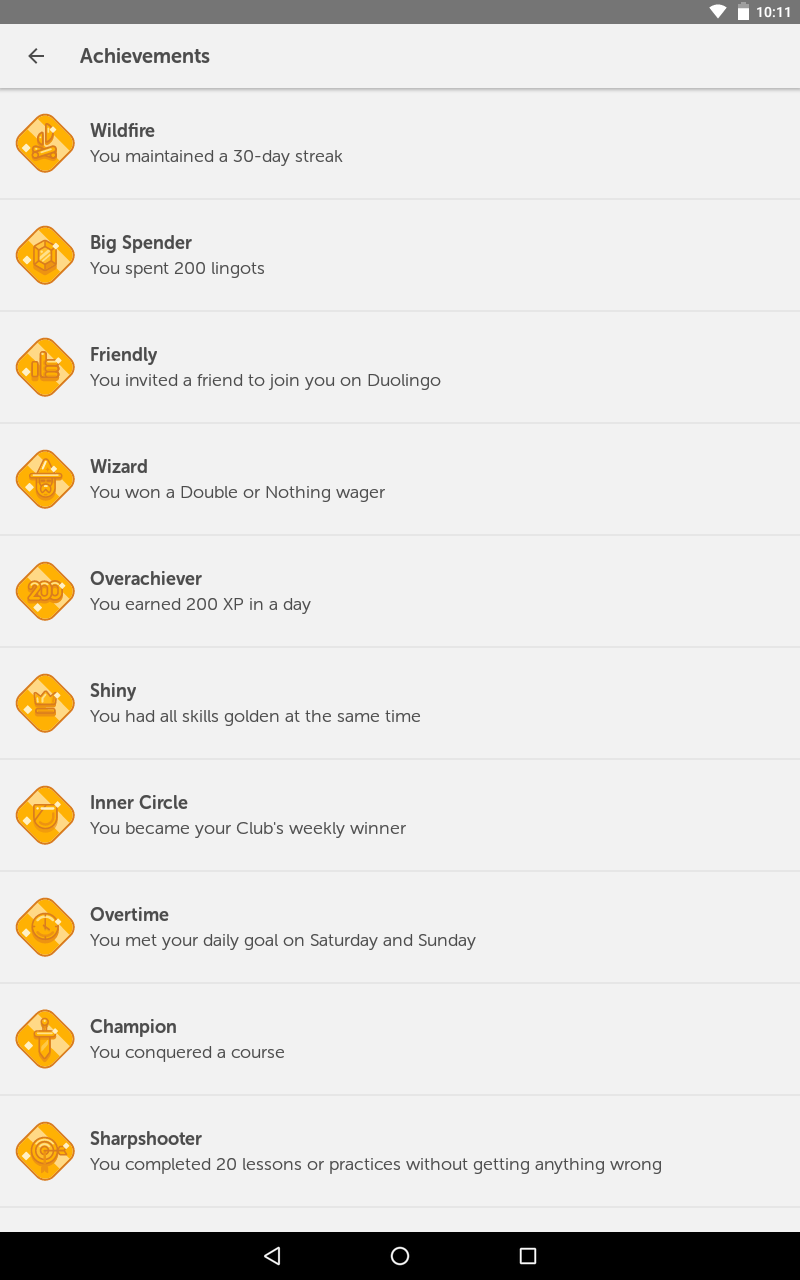


Figure 2.1: Duolingo badges[[21]](#endnote-21) Figure 2.2:Duolingo Lingots[[22]](#endnote-22)

1. Progress bar

Progress bars is an element, which allows players to know their progress and how close they are to achieve something.

Khan Academy uses it to display the progress of its learners when learning a subject.

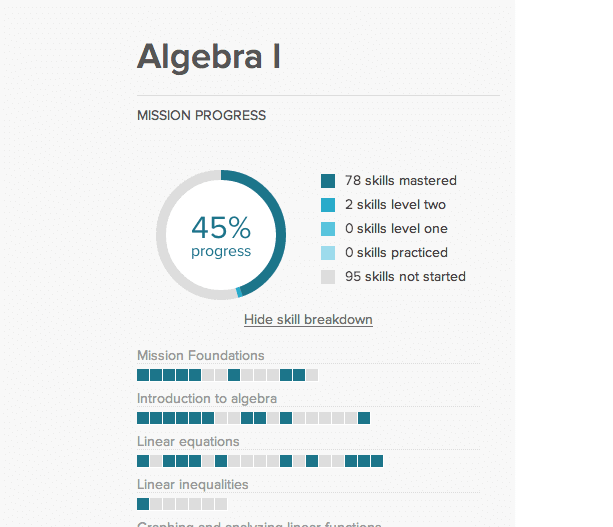


Figure 2.3: KhanAcademy Student’s progress bar.[[23]](#endnote-23)

1. Leveling

Leveling up keeps learners motivated and give them a sense of achievement, it exploits the learner’s desire to reach a new and harder level.

Duolingo applies it everytime learners earn a specific amount of points to level up.



Figure 2.4: Duolingo level up[[24]](#endnote-24)

1. Leaderboard

It is a board where the leading competitors are displayed. It increases motivation and the desire to be on top of the leaderboard.

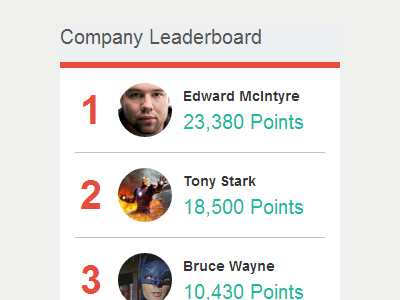


Figure 2.5: Company Leaderboard [[25]](#endnote-25)

1. Quizzes

Quizzes are used to test the learner's knowledge.

Coursera utilizes quizzes at the end of each lesson to check learners understanding of a lesson.

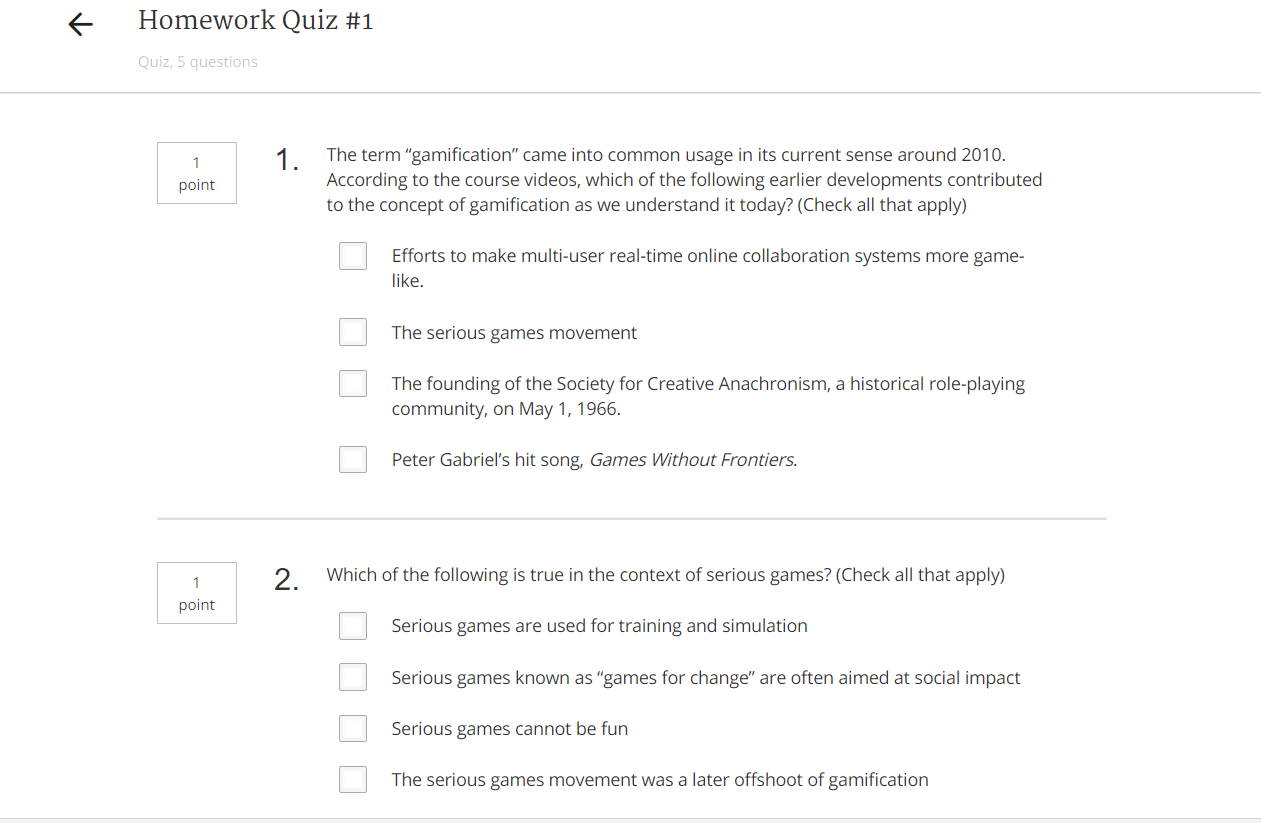


Figure 2.6: Coursera Quizz[[26]](#endnote-26)

1. Challenges

Challenge can be used to maintain the motivation level and push users to achieve something meaningful.

Runkeeper is a fitness tracking app, which allows its users to keep track of their workout. It uses health challenges, which is a fantastic gamification feature to motivate its users.



Figure 2.7: Runkeeper challenges[[27]](#endnote-27)

## Conclusion

Anything can be gamified; it is straightforward, one can gamify efforts such as doing more exercises, losing weight, maintaining habits consistently, the goal is to earn something as a reward such as badges, trophies, being on top of a leaderboard, among others. Gamification taps into our competitive nature thus making players addicted and wanting to play more.

## Chatbot

A chatbot is a software, which can simulate a conversation via text or auditory methods.

The idea of a mechanical consciousness was first conceived in 1872 when the Author Samuel Butler wrote a science fiction novel featuring it.

However, the idea only took off in 1966 when the first form of artificial intelligence was created: Eliza.[[28]](#endnote-28)

Eliza is known as the first chatbot, Joseph Weizenbaum developed it, and its function was to emulate a psychotherapist.

Eliza was able to have basic conversations, if one said to Eliza that he/she is sad, Eliza would reply with "why are you sad?" , it would try to continue a dialog; however, the program was unable to maintain a conversation. [[29]](#endnote-29)

Despite not having real intelligence and only being able to perform pattern matching and substitution methodology, Eliza was considered innovational, which inspired more people to implement chatbots. [[30]](#endnote-30)

Siri

In 2010 Apple acquired an intelligent personal assistant called Siri. SRI International Artificial Intelligence Center invented it.

Siri was developed to allow people to utilize the internet differently, can instead of using a search engine, which gives a list of websites to choose, Siri can have a conversation with its users for example(

-Siri, what is the weather five days forecast for Dublin Ireland?

-It is not looking good through Saturday... down to 5 degrees and raining.)

Siri allows a user to book reservations at restaurants, schedule events, transcribe text messages by utilizing speech-to-text. [[31]](#endnote-31)

Alexa

Alexa is a personal assistant developed by Amazon. It performs the same actions as Siri does; however, it is run as a service in the cloud, which allows Amazon developers to improve it continuously by the millions of users utilizing this product. [[32]](#endnote-32)

There are two primary methods to develop a chatbot.

Retrieval-based model has a set of predefined questions and responses which are stored in a repository. Based on the input and context, the bot chooses an appropriate response to it users, however it only outputs responses that are in the repository.

Generative model is a more complicated approach, which uses "Deep learning". It does not use a predefined repository like the retrieval-based model.

The main advantage of using a retrieval-based model is that it will not make a grammatical mistake, however, it is hard to scale, for the reason that new set of responses will have to be input to the repository.

The main advantage of generative model is that it is more scalable for the reason that it automatically outputs a response based on the data that was fed to the Deep learning model; however, there is a chance of a very inaccurate response thus making it impractical at times.

Generative model is extremely complex to perfect it with the current knowledge available about Deep learning algorithms. Chatbot using Deep learning can get us only so far. However, it is far from perfect.

With current technology, it is impossible to develop a perfect chatbot based on the generative model.

Sequence to Sequence model which is a Deep learning architecture, is uniquely suited for a generative model; however, it is still at an early stages of building an accurate chatbot.

Chatbots in production are more likely to use a combination of retrieval-based model and Deep learning currently.[[33]](#endnote-33)

## Natural language processing

Natural language processing(NLP) is the area of artificial intelligence which aids computers to comprehend, manipulate and interpret human language. Its primary objective is to fill the gap between computer understanding and human communication.

There are various techniques used in NLP to interpret human languages such as ruled-based and algorithmic approaches, machine learning, and statistical methods. The vast approach to process natural language is needed for the reason that voice-based data and text varies.

The main tasks used in NLP are part of speech tagging, tokenization, lemmatization/stemming, language detection and identification of semantic relationships. These tasks are used to enable the processes such as:

Machine translation- Allows for automatic translation of text or speech from one language to another such as English to French.

Speech-to-text and text-to-speech conversion - enables the conversion of voice command into written text and vice versa.

Document summarization- Automatically generate synopses of large bodies of text.[[34]](#endnote-34)

## Alternative Existing Solutions

## Duolingo

Duolingo is a language learning app free of charge, which offers courses on many languages, it currently has 32 languages courses for English speakers and much more for speakers from other languages such as French, Portuguese, German, Italian among others.

Duolingo uses many gamification principles such as badges, points, streaks, lingots which is a virtual currency, progress bar, a shop and so on.

Duolingo is divided into modules, each module there are specific lessons to help learners to achieve their goals. For each correct answer, one can earn points to be able to progress to a new lesson.

In each lesson, one can practice their speaking, listening and writing in a very entertaining way.

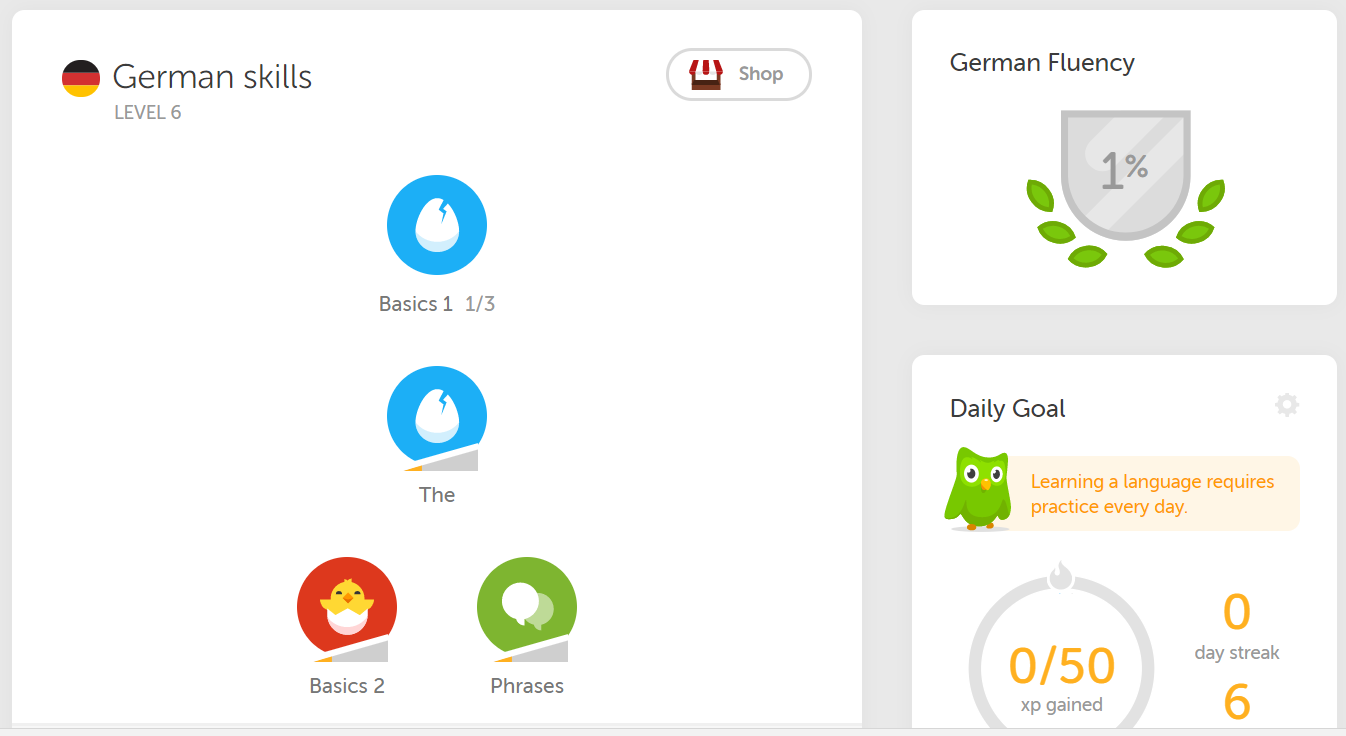


Figure 2.8 Duolingo.[[35]](#endnote-35)

To maintain motivation, Duolingo records how many days in a row learners spend learning, and send notifications to them when it notices they have not logged in for a period.[[36]](#endnote-36)

Duolingo is an excellent example of successful application of gamification for language learning.

A study conducted in 2012, has proved the effectiveness of Duolingo, researchers have discovered that the average improvement ability was 91.4 points which were statistically significant. Participants gained 8.1 points per one hour of study using Duolingo.[[37]](#endnote-37)

## Babbel

Babbell is a paid language learning app, unlike Duolingo, Babbel is a paid app and has fewer gamification techniques.

Here is the price range.

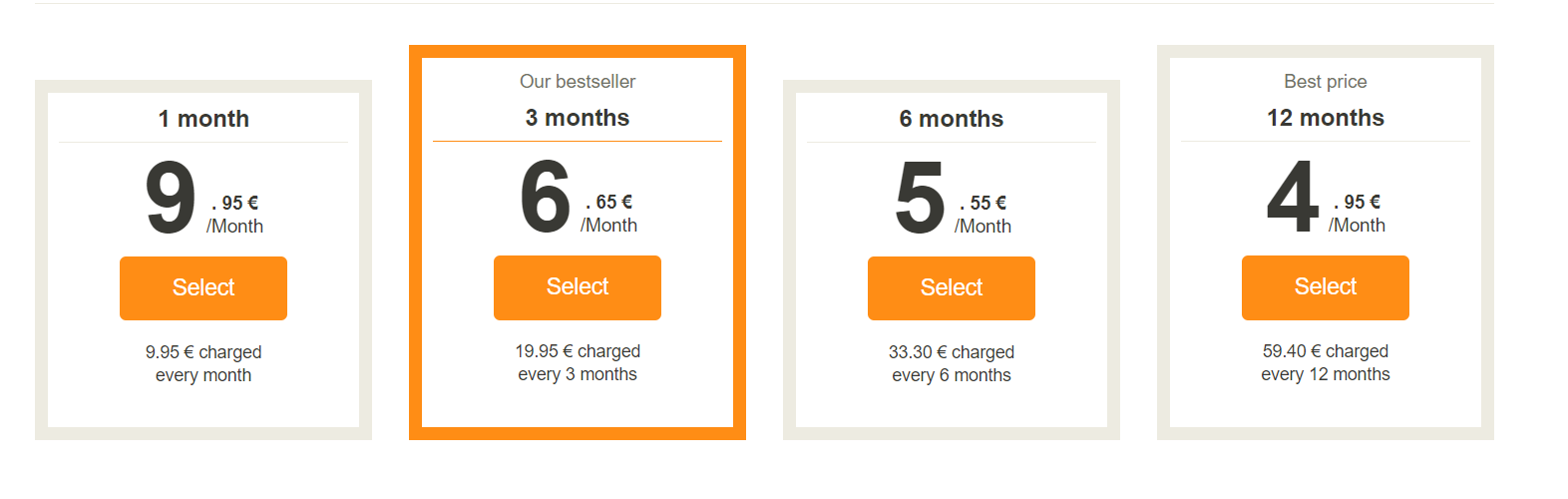


Figure 2.8 Babbel price range[[38]](#endnote-38)

Babbel's structure to teaching a language is different from Duolingo methods of teaching.

Babbel is broken down into course catalogs such as.

* Beginner course

One can be introduced to the language, and learn the basics.

* Grammar

This course is focused on the grammar of a chosen language. It offers many exercises and explanations.

* Countries and traditions
* In this course, one can learn not only the language chosen, but more about the culture of countries where the language is spoken.
* Special

Here students can revise about idioms, pronunciation, Words, and sentence[[39]](#endnote-39)

## Conclusion

Vocabnote will not compete directly with Babbel and Duolingo, they both teach English, the goal of this app is to allow students to have more options to practice vocabulary learned on those apps or classroom.

## Technologies Researched

# Mobile phone Operating system

## Android and IOS

Android and IOS account for 99.7 percent of the mobile Operating System(OS)market share.[[40]](#endnote-40) This is the main reason for researching on them

Android powers hundreds of millions of gadgets around the world.

An ordinary person mainly chooses android mobile phones because of the variety of smartphones which use Android as their platform, and they are cheaper compared to IOS gadgets.

Figure 2.9 illustrates the global mobile OS market share. In the first quarter of 2017 Android accounted for 85 percent and IOS 14.7 percent of the market.

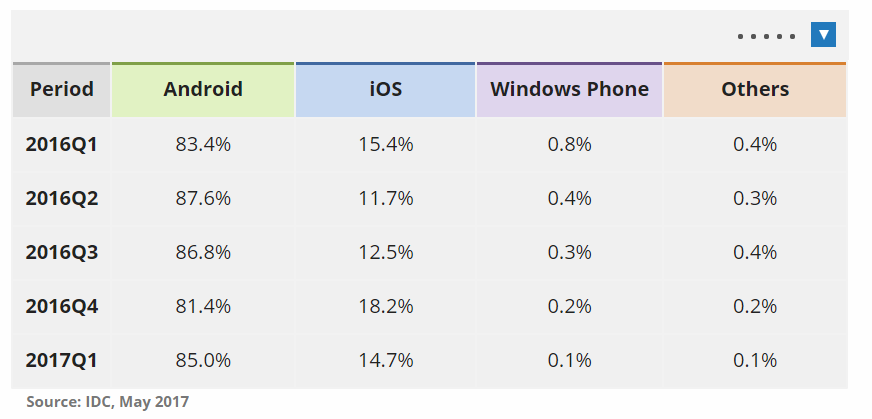


Figure 2.9: **Global mobile OS market share**[[41]](#endnote-41)

Android and IOS have dominated the market share for many years.

As shown in figure 2.9, Android has grown 3.6 percent from the first quarter of 2016.

Financially, Android is also better for developers, regarding app submission.

To submit an app to Google play store, one has to pay only $25, unlike Apple store which is an annual fee of $99.

## Conclusion

The platform chosen for this project is Android because:

* It is the most popular gadget operating system
* Potential to have more users
* Cheaper to submit an app to Google play store

## Hybrid vs Native app

At the beginning of this project, choosing between developing a hybrid or native app was a very hard task, both ways of development have its pros and cons, which is cited below.

A native application is a mobile application that is developed to be used on a particular platform, for example, Android, IOS, etcetera. It is implemented with its native programming languages such as java for android and Objective-C or swift for IOS.[[42]](#endnote-42)

A Hybrid application is a web application which runs on a native platform. It is developed using web technologies like HTML5, CSS, and JavaScript and deployed using Cordova, which is a mobile application framework.

Native app Hybrid app

|  |  |
| --- | --- |
| Advantages  * Great performance * Better UI/UX design * Safer  Disadvantages  * High development and maintenance cost * Cannot be deployed on other platforms | Advantages  * Cheap development and maintenance cost * Fast development speed * Efficient scaling * Can be Deployed on many platforms  Disadvantages  * Slower performance |

## Conclusion

The native approach was chosen for this project, for the reason that having to learn how to develop a Hybrid app would require a lot of time.

# Programming languages

## Java and Kotlin

Java has kept itself stable as the most popular programming, Kotlin has been recently introduced by Google as a programming language to develop Android applications, these are the reasons why Java and Kotlin were researched.

Java is an object-oriented programming language, which is general-purpose.

Sun Microsystems developed Java in 1995.

Figure 2.10 illustrates the top ten most popular languages worldwide. Java leads as the most popular with C coming in second.

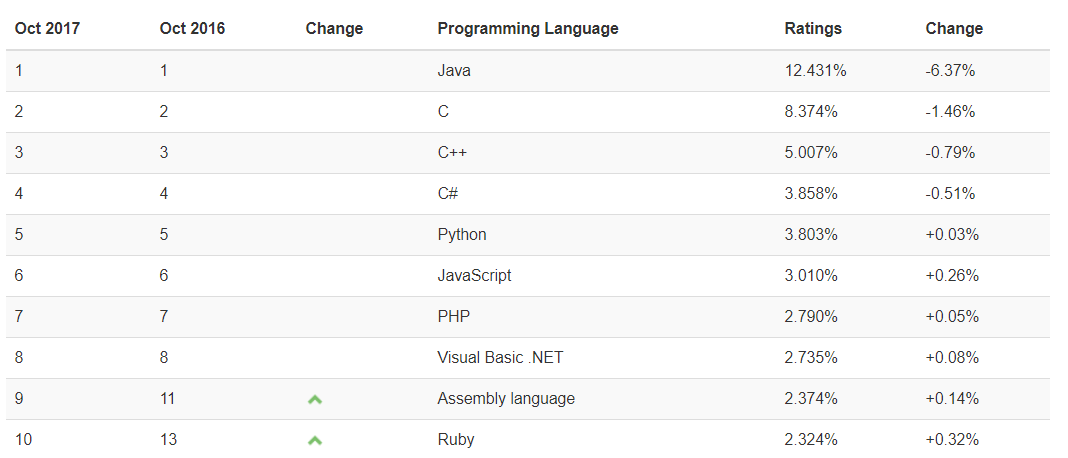


Figure 2.10: Most popular programming language index[[43]](#endnote-43)

Most of the tutorials and apps online regarding Android development are developed in Java. However, Java is not the only language, which one can develop an android app, Any language that runs on Java virtual machine can be used.

Google has announced Kotlin as an official language on Android.

Kotlin is a statically typed programming language which runs on java virtual machine.[[44]](#endnote-44)

Android studio 3.0 completely supports Kotlin.

Kotlin can work alongside another language such as Java on Android; one can use Java and Kotlin on the same project interchangeably, It is interoperable with other languages used for Android development.[[45]](#endnote-45)

Java Kotlin

|  |  |
| --- | --- |
| Advantages  * The most popular language * Better community support  Disadvantages  * Requires more line of codes to develop a behavior | Advantages  * More concise code * Compatible with existing code  Disadvantages  * New programming language * Less community support |

## Conclusion

Due to having experience with java in the past, it will be the primary language for this project.

## Python and R

Python and R are the most preferred languages for data scientists to develop machine learning and deep learning model, to automate computers and make predictions using data. These two languages were researched for the reason that a ChatBot using a deep learning model will be implemented.

Python is a high-level, general-purpose, interactive and object-oriented programming language used worldwide, Guido van Rossum was the creator of it.

Python is an interpreted language; one does not need to compile an application before executing it, it processed at runtime by the interpreter.

Python has a smaller and clear defined Syntax which can be beneficial for novices who wants to learn how to programme, it uses English keywords frequently, which makes it easily readable compared to other languages.

It is a portable language which can run on many hardware platforms with the same Interface.

It offers an extensive standard library with a wide range of facilities which aid programmers not to reinvent the wheel.[[46]](#endnote-46)

R is a programming language for statistical computing and graphics. It was developed at Bell Laboratories.

R has been an excellent choice for those in the data science field because it provides a vast range of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering,) and graphical techniques.[[47]](#endnote-47)

The popularity of R in the Data Science field has been growing extensively. Data science contains many predictive modeling techniques, which can be difficult to understand and apply.

Many novices and data scientists looking to improve their skills choose Python because of its simplicity and a shorter learning curve compared to many other languages. It has many libraries such as Theano, Keras, and TensorFlow, which aid developers in the creation of machine learning and deep learning models. [[48]](#endnote-48)

Python R

|  |  |
| --- | --- |
| Advantages  * Easy to learn * Supports many libraries for Deep learning * Excellent community support * Syntax similar to English  Disadvantages  * Smaller community support compared to R | Advantages  * Vast range of Statistical techniques * Excellent community support * Better for data visualization  Disadvantages  * Steeper learning curve |

## Conclusion

Python was the language chosen to develop and train a Deep learning model to implement a Chatbot because it is renowned for being an easy programming language to learn with a straightforward syntax.

# Middle tier languages

Django and Laravel are among the top most popular middle-tier frameworks. However, they are written in two completely different languages. Both of them were researched to compare which one was the most suitable for this project.

## Django

Django is a highly popular and open source Python web framework which uses Model, View, Template (MVT) architecture similar to MVC. Django allows for a rapid and efficient development of a web application.Django integrates user authentication, content administration, sitemaps, RSS feeds, form database management and much more. It helps to avoid SQL injection and many other security mistakes made by developers.

## Laravel

Laravel is a PHP web framework; its creator is Taylor Otwell, and in 2011 it was released. It is considered the most popular PHP framework.

Laravel provides many functionalities to allow for rapid and enjoyable implementations, which decreases development overhead such as authentication, validation, session, pagination, caching and so on.

With Laravel, one can connect to a database and run queries with ease and concise code. It supports four database systems such as MySQL, SQLite, SQL server, and PostgreSQL[[49]](#endnote-49)

Here are the advantages and disadvantages of both.[[50]](#endnote-50)

Django Laravel

|  |  |
| --- | --- |
| Advantages  * Provides an admin panel * Prototype development can be very fast * Great documentation and community support  Disadvantages  * Template fails are hard to spot because it does not show an error message | Advantages  * Good documentation * Web app with authentication can be written easily * Powerful template called Blade  Disadvantages  * Poor performance |

## Conclusion

Django is the ideal framework for this project, mainly because of its powerful libraries, its convenience to aid in the development of the backend, creation, and serialization of a web-browsable API which is used to allow the android app to fetch, manipulate and store data, and it is a python framework. Python is the leading programming language for Data science, and it has a robust library called TensorFlow which will be used to aid in the creation of a Deep learning model to implement a Chatbot[[51]](#endnote-51)

# Databases

## Postgresql

PostgreSQL is a robust, open source object-relational database system. The PostgreSQL Global Development Group created PostgreSQL back in 1996.

It provides most of the SQL 2008 data types, such as INTEGER, NUMERIC, BOOLEAN, VARCHAR, INTERVAL, TIMESTAMP AND DATE.

It supports all major operating systems, for example, Windows, Linux, Unix and so on.

PostgreSQL offers a very detailed documentation and native programming interfaces for Perl,Net, Python Java, C/C++ and several more, it also provides a sophisticated Multi-Version Concurrency Control (MVCC) which handles data consistency when multiple processes are accessing the same table.[[52]](#endnote-52)

## MySQL

MySQL is an open source relational database system, developed, distributed and supported by Oracle.

It is the most popular open source relational database worldwide.

It is used by many well-known web applications such as YouTube, Twitter, Facebook, among others. It is renowned for being easy to use, reliable, and scalable and provides excellent performance.

MySQL offers many features such as

* Online Schema change and performance Schema to manage business requirement and monitor user and application level performance and resource consumption
* SQL and NoSQL access to perform complicated Queries,
* Big Data Interoperability using MySQL
* Flexibility to develop and deploy on various operating system
* It includes many advanced features such as MySQL Enterprise Monitor, MySQL Enterprise Backup.[[53]](#endnote-53)

PostgreSQL MySQL

|  |  |
| --- | --- |
| Advantages  * Excellent documentation and community support * Extensive tutorials and support by Django framework * Django recommends PostgreSQL  Disadvantages  * Steep learning curve | Advantages  * Reliable * Good community support  Disadvantages  * Poor Documentation and community support combined with Django * Some queries run very poorly when paired with Django |

## Conclusion

PostgreSQL is the database system chosen for this project.

PostgreSQL is often recognized as feature-rich and stable; It is the primary database system used by Python developers. Therefore it is a good fit to use with Django because there is far better community support, well-written documentation and code examples compared to the use of Django with MySQL.

# Cloud services

## PythonAnyWhere and Heroku

To access the Django server from the frontend, it will be uploaded to the Cloud, where data can be manipulated.

PythonAnyWhere and Heroku were the two Cloud servers researched for this purpose.

The key advantages and disadvantages which distinguished which one to use for this project is described below.

PythonAnywhere Heroku

|  |  |
| --- | --- |
| Advantages  * Super easy to deploy a server to the it * Allow User media to be stored in it  Disadvantages  * Only supports Python * Have to pay to access PostgreSQL | Advantages  * Support many programming languages * Offer many Databases plugins such as PostgreSQL and MySQL  Disadvantages  * The learning curve is steeper compared to PythonAnywhere * User media Cannot be uploaded to Heroku |

## Conclusion

PythonAnywhere is the chosen Cloud service, mainly because it proved to be a very easy application to deploy the Django server.

## Other Relevant Research Done

## Spaced repetition

Spaced repetition is a learning technique, which aids on the learning and memorization of a learned material over a long term. Using spaced repetition, learners increase intervals of time to revise a material instead of cramming. Cramming is not very effective in a long-term, for the reason that the material learned is poorly retained.

Studies have proved that spaced repetition is more effective in helping retaining more information than cramming.[[54]](#endnote-54)

After a certain period, the ability to recall learned material is decreased. Some psychologists have claimed that learners do not forget, the memory is still in their mind however they are unable to access it.[[55]](#endnote-55)

In 1885, Hermann Ebbinghaus studied the inability our recalling memories over time. Ebbinghaus has discovered that the level at which we retain information depends on learner’s strength of memory and the lag time passed since the learner has practiced the material.

The result was a formula that measures how long information remains in our memory, which is called Ebbinghaus’ Forgetting Curve.[[56]](#endnote-56)

p = 2−∆/h.

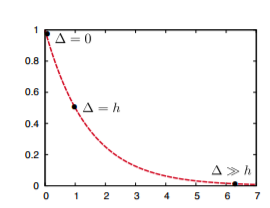


Figure 2.10: Ebbinghaus’ Forgetting Curve [[57]](#endnote-57)

Figure2.10 is an illustration of the Forgetting Curve.

*P* is the probability of remembering a material correctly (e.g. a word).

*∆* demonstrates the ”lag time” which is the time passed since the learner has practiced the material and *h* is the “half-life” or learner’s strength of memory in the long term.[[58]](#endnote-58)

1. ∆ = 0. The material was practiced recently, according to Ebbinghaus the material should be recalled correctly.

2.. ∆ = h. It means that the lag time is equal to half-life and the learner is almost unable to remember the material learned.

3. ∆ >>h.The material has not been revised for a period related to its half-life, so there is a good percentage that the material has been forgotten[[59]](#endnote-59).

The goal of using a spaced repetition algorithm is to reduce the effect of the forgetting curve. Studies have shown that repeating a learned material twenty times over the course of a day is less effective than repeating it ten times over a week.[[60]](#endnote-60)

### Half-life regression

Half-life regression is a spaced repetition algorithm designed by Duolingo.

It is a combination of the formula Ebbinghaus’ Forgetting Curve, illustrated in figure 5, and modern machine learning techniques to predict intervals more accurately at which students should revise material learned, e.g.(Words).

Figure 5 demonstrates Ebbinghaus’ Forgetting Curve with h = 1. Therefore the output from half-life will be shown below:

1. ∆ = 0, is the same as P = 20 = 1.0 the output from half life regression would be 1, the words were revised recently.

2.. ∆ = h, is the same as p = 2−1 = 0.5, the output is 0.5 which means students are close to forgetting the words learned.

3. ∆ >>h. Is the same as P ≈ 0. The output is 0, so students are likely to have forgotten. [[61]](#endnote-61)

### The Pimsleur method

The Paul Pimsleur method is a spaced repetition algorithm designed by Paul Pimsleur in 1967. [[62]](#endnote-62)

It is an audio-based language learning algorithm used in the Pimsleur language learning system.

Newly learned words are presented then tested at increasing intervals, scattered with the revision and introduction of other learned words. For example, learners would listen to words; then they are asked to repeat them 15 seconds later, then 45 seconds later and the interval will keep increasing.[[63]](#endnote-63)

### The Leitner System

Leitner System was a spaced repetition algorithm proposed by Sebastian Leitner in the 1970s.[[64]](#endnote-64)

The algorithm was intended to be used mainly with flashcards. Depending on the student’s performance, intervals can be increased or decreased, which is more adaptative than Pimsleur method.

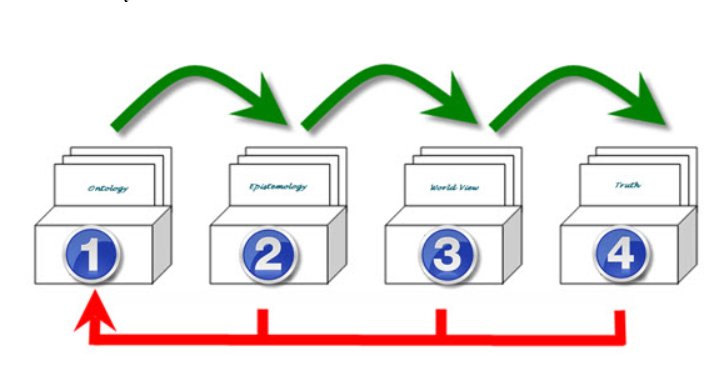


Figure 5.11: The Leitner System for Flashcards[[65]](#endnote-65)

Figure 5.11 illustrates the spaced repetition algorithm. Firstly all the cards start in box one, then if the cards are guessed correctly, they are moved to box two, cards which are guessed correctly in box two are moved to box three and so on.

Cards which are guessed incorrectly in any boxes are demoted to box one. E.g.(Cards in box four are guessed incorrectly, then they are moved to box one).

Students should always start in box one and change to subsequent boxes when box one is empty.

Leitner spaced algorithm is so effective, for the reason that students can revise flashcards that they have forgotten more often and those they remember less frequently.

Many flashcard software such as Anki and Cram.com utilize Leitner system or an algorithm similar to it.

## Deep learning

Deep learning is the science which provides computers with the ability to learn and improve automatically without explicitly programming it. To achieve computer automation, Artificial Neural Network(ANN) is used.[[66]](#endnote-66)

### Artificial Neural network

Artificial Neural Network is a model based on the structure and functions of the brain.[[67]](#endnote-67)

ANN is comprised of three interconnected layers, which interact with each other, those layers are input, hidden and output.

The input layer receive data, execute operations on the data and pass it to the hidden layer which are sent to the output layer where the final predictions or classifications are outputted. Figure 5.12 illustrates an Artificial Neural Network.[[68]](#endnote-68)

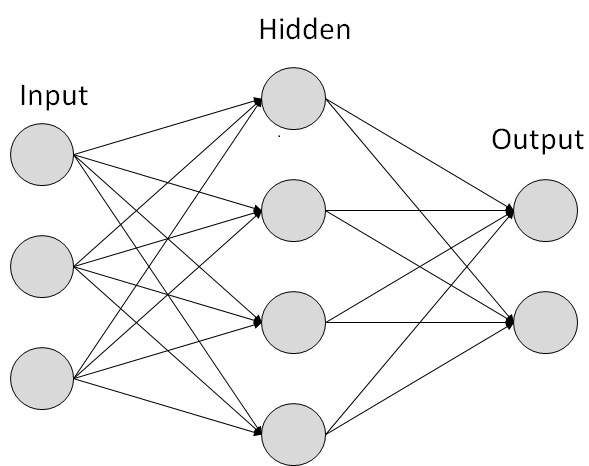


Figure 5.12: Artificial Neural Network[[69]](#endnote-69)

## TensorFlow

TensorFlow is a library which Google developed to facilitate the development of machine learning and deep learning models.

TensorFlow offers tutorials on recurrent neural networks, which will be used to develop the chatbot.

Figure 5.13 illustrates a Recurrent Neural Network. The Recurrent Neural Network is an artificial network that store data and keep it in memory from past nodes even after that data is transferred to a new node, that short-term memory allows the model to output predictions and classifications more accurately.[[70]](#endnote-70)

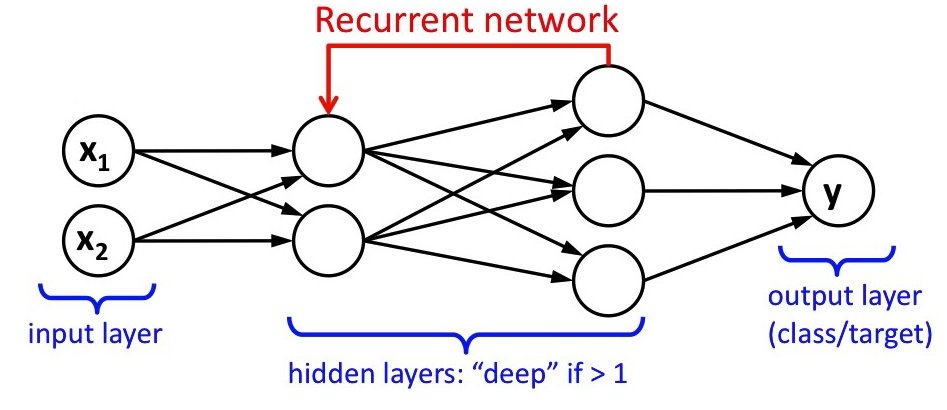


Figure 5.13 Recurrent Neural Network[[71]](#endnote-71)

Figure 5.13 depicts a Recurrent Network. It has a loop, which loops back to previous nodes to learn from past experience. The best analogy to describe a Recurrent Neural Network is to imagine it is a human. Humans when learning maths, they learn the rules, and when they are going to solve a math problem, they remember the rules and use those rules to solve them. The same occurs with a Recurrent Neural Network, it learns from past experiences and maintains that knowledge to improve its output.

## Dialogflow

Dialogflow is an end-to-end development suite based on natural language conversations, which allows users to build conversational interfaces for many platforms such as mobile applications, websites, and so on.

it's mainly used to create chatbots which are capable of rich and natural interactions. Machine learning powers it.

By providing with examples of what a user may say, the machine learning algorithm is able to match to an appropriate response to what a user is saying.

To implement a chatbot, two functionalities from Dialogflow will be used, Agents and Intents.

Agents are natural language understanding modules. These modules are used to convert user requests into actionable data.

When a user inputs a request, the conversion occurs if the input matches with an intent inside the agent.

Intents are developer-defined or predefined components of agents which process user's requests.

Agents can also be designed as a chatbot to manage conversation flow in different ways.

Figure 5.14 depicts the DialogFlow architecture.

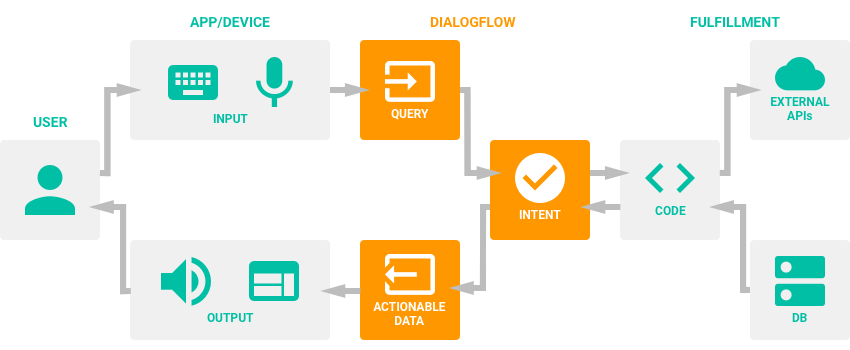


Figure 5.14 DialogFlow architecture[[72]](#endnote-72)

Intents are the representation of a mapping between user's input and actions which the software should take. The intent interface has four sections which are, Training phrases, Action, response, context.

Training phrases are phrases input by developers, which is expected from users, if user’s input match with one of the phrases, the intent will be triggered.

Context is used to remember parameter values which can be passed from one intent to another, for example, a follow up intent can be stored as a context to remember that after one intent is triggered there will be another one.

Actions is a component of an intent which send data to fulfilment to process it and send back a response.

Response is any text which will be delivered to a user.[[73]](#endnote-73)

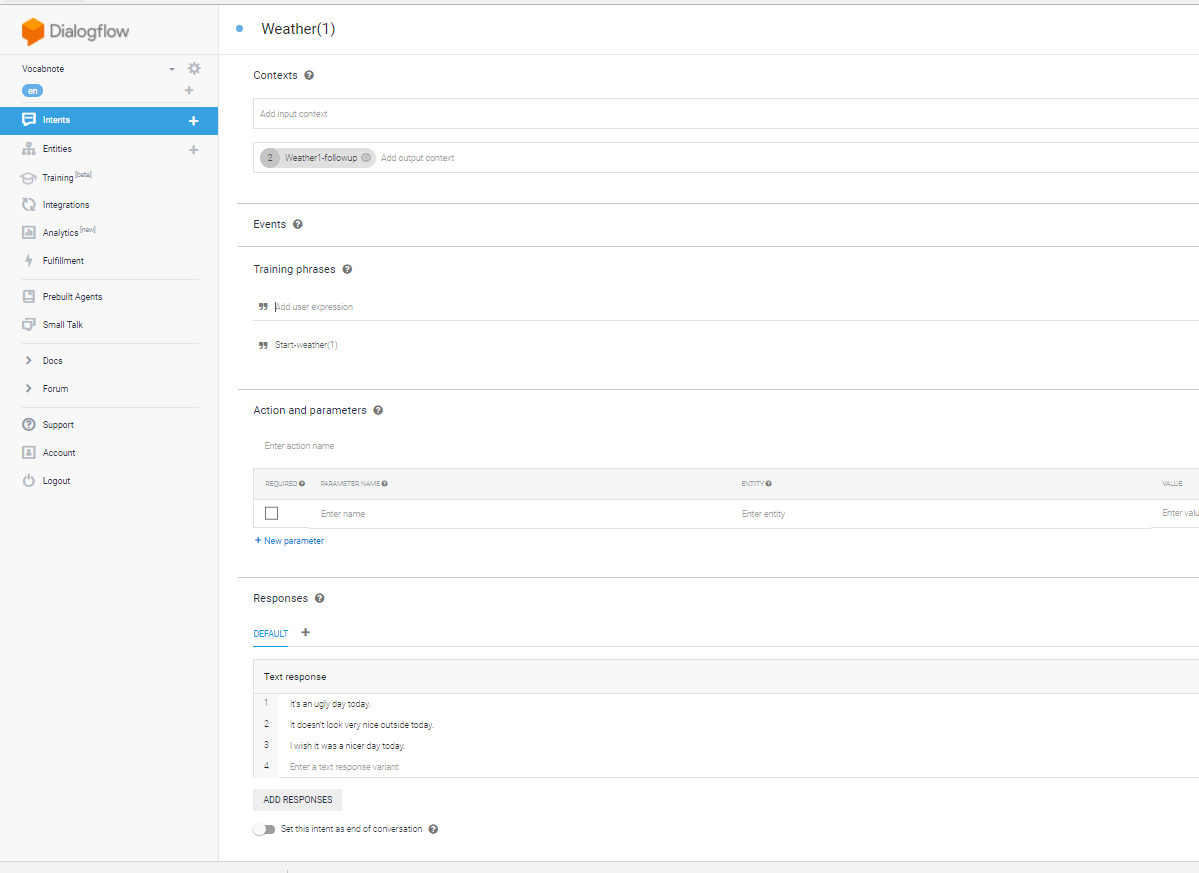


Figure 5.15 Intent interface

# Design

## Methodology

# Agile methodologies are more suitable to be utilized in a team. Since this, a solo project many methodologies such as Scrum and Kanban flow are not suitable for it. However, with some adaptations, Extreme programming (XP) seemed to be adequate.

#### Extreme programming

# Extreme programming (XP) is an agile software development methodology, which is responsive to customer requirement changes and has a goal to improve software quality. As it is an agile methodology, Extreme programming emphasizes the division of a project in iterations and small releases.

# Extreme programming solves many problems encountered in others software development methodologies such as Waterfall and RUP. These methodologies advocate the implementation of a software development life cycle(Requirement, analysis, and design, implementation, test, and deployment) step by step.

# These methods can be prone to fail for the reason that customers can change their minds in the middle of implementation, then the requirement, analysis, and design would have to be revaluated thus wasting time and money.

# The same would happen if a tester found many bugs in the project, time would have to be allocated to debug it, time could run out, and the project would be delivered with many bugs and incomplete.

Changes are welcome in Extreme programming; it can be achieved successfully

by gathering continuous feedback from clients, keeping interactions short, design and redesigning, implementing and testing the project frequently, eliminating bugs early, keeping the clients informed and involved in the project and most important deliver working products.[[74]](#endnote-74)

#### Extreme Programming values

Extreme Programming advocates five values:**[[75]](#endnote-75)**

1. Communication

It is key to a successful project; teams work together at any cycle of a software development, problems are solved together.

1. Simplicity

Simple steps are taken to complete the tasks, with the intention to decrease overhead.

1. Feedback

At the end of an iteration, any feedback from clients is welcome, changes can be discussed, and the iterations continue.

1. Courage

Respect is precious to maintain a healthy work environment.

#### Extreme programming best practices

The best practices involve:[[76]](#endnote-76)

1. The Planning Game

The planning game is a plan developed by the business and technical people.

The Business people decide the scope of the project, prioritize the main features, dates, and structure of each release.

The technical people estimate how long it will take to complete every release, organize the team, implement a detailed schedule informing about the risks and which user stories should be developed first.

User stories are detailed requirements which contain enough information, which developers can produce an estimation of the time it will take to complete each user stories.

The planning game needs to be adapted for this project instead of a team deciding all stated above, only one person will decide and manage accordingly.

The planning games are advantageous because features which are not very important can be expelled from the project thus reducing time and guesswork.

1. Short Releases

Iterations are implemented in a short amount of time, thus enabling small releases. The duration of each release may vary, however keeping it around one to two weeks is best practice.

In this project, the releases will be around two to three weeks due to having to execute background research before starting the implementation. However, it can be changed at any time.

Short releases are advantageous for the reason that feedback from the supervisor can be frequent and reduces the risk of an incomplete and undesirable project.

1. Metaphor

Metaphor in this context means the language used by everyone in the team, to define the architecture and to improve communication.

In a team, it is advantageous to have excellent communication between each other and reduce confusion. However, this does not apply to this project.

1. Simple Design

The design of the project should be simple, extra complexity will only increase overhead.

The design is implemented at the beginning of each iteration, and it should be the designed for that iteration only, unlike other methodologies which design the whole project in advance.

Simple design is advantageous for the reason that it reduces overhead, easier to understand and can be refactored at any time without compromising progress of the project.

1. Testing

Tests should be written first before developing the user stories.

There should be automated tests, and customers should test the features after each release also.

For this project unit tests will be implemented first, then at the end of each iteration, three English students will be testing the app to guarantee it is user-friendly and is bug-free.

Testing is advantageous for the reason that it helps in reducing bug and increases the quality of the software.

1. Refactoring

Extreme programming embraces refactoring for the reason that, it allows the developers to keep improving the software by eliminating duplications and structure of the code, thus improving performance without changing its behavior.

It is advantageous because the software can have a better quality.

1. Pair Programming

Pair programming means two developers develop features together,

one develops write the code, and the other can help solve problems and think of ways to improve code quality.

Pair programming is an excellent extreme programming practice for the reason that it can help developers to solve problems faster. However, this practice will not be used in this project because it is a solo project.

1. Collective Ownership

In extreme programming, the entire team is responsible for the outcome; they work collectively to solve problems.

It is an excellent practice because more experienced developers can help less experienced ones. However, this practice will not be used throughout the project for the same reason cited above.

1. Continuous Integration

Integration of the software should be done daily. It can reduce bugs, and enable small releases.

1. 40-Hour Week

Extreme Programming emphasizes forty hours work a week, to keep team member fresh, creative and confident.

After 40 hours the developers start to be less efficient, so emphasizing this practice be excellent to keep a great work environment, maximizing productivity and maintaining good code quality

1. On-Site Customer

Users of the software can be included in the team to help the team prioritize user stories, test the software and resolve disputes.

The advantage is that the most critical user stories are prioritized correctly.

Three English students will be performing this role.

1. Coding Standards

Developers should keep a good coding standard, to reduce the amount of work, avoid duplications and allow other developers to have a good understanding of the code.

This practice is advantageous because developers have less problem understanding code written by each other, thus improving productivity.

#### Visual management

Visual management is a practice used in agile for visualizing the progress of a project. This practice will be used throughout the project to maintain track of the project progress, analyze priorities and also remind what should be completed in each iteration.

The planning game was executed at the beginning of the project, Conforming Extreme Programming methodology.

This project is broken down into five iterations. The length of each iteration may vary, however, ideally, twenty-five days are enough to allow for the completion of each iteration.

### User stories

Requirements were gathered as user stories; each user stories have an estimation of the time it will take to complete them.

User stories are estimated by using a point based system or ideal days.

Ideal days refer to how many days is necessary to complete each user stories.

Some user stories might be big and take roughly five to fifteen days to be completed, and other might take only two to three days. However, user stories estimations are not accurate, a developer may be having technical issues, and it could be required more days to complete a user story and vice versa, which would require constant estimation changes for each iteration. A point based system is a better approach.

Using a point-based system, each user stories is classified into small(1 point), medium(3 points) and large(5 points). The point-based system is a more accurate system because there is no need to change the planning game. The points merely determine that user stories could be small, which only a few days are enough for the completion, medium or large, which more days are required.

Figure 3.1 illustrates the planning game. It displays each user stories and estimations, the estimated starting dates for each iteration, total points, completed user stories and iterations, and average velocity of each iteration.

As shown below the fourth iteration has thirteen points and the last iteration has twenty-six, that means they are the longest and hardest iterations.



Figure 3.1: The planning game, User stories, and estimation

A burndown chart was also implemented. Burndown chart is utilized in agile to track the progress of releases. The whole project has fifty-eight points.

Figure 3.2, shows the deduction of points in each iteration release and estimated start dates.

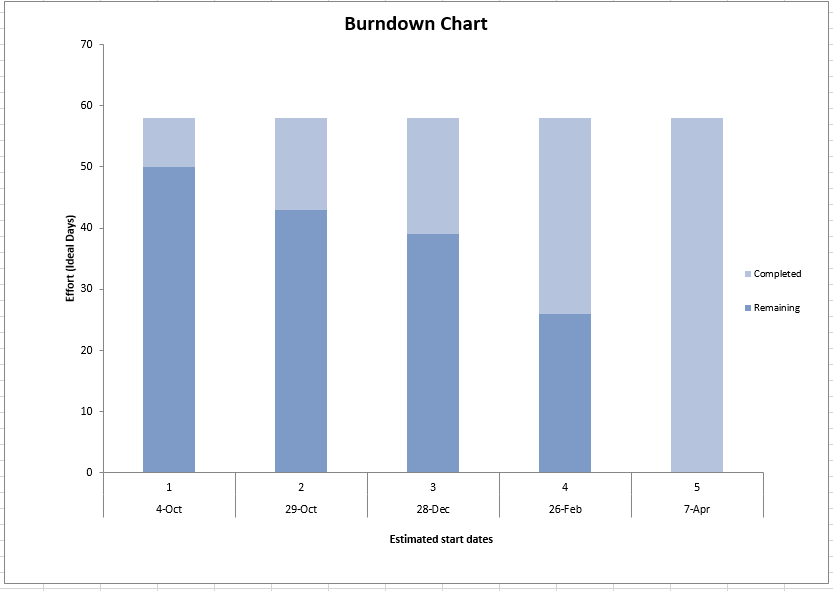


Figure 3.2: Burndown Chart

## Design of Project Components

#### User Interface Design

In this project, it was used Material design to offer users a great user interface.

Material design is a visual language developed by Google, with the combination of classic principle of good design such as Equitable use, flexibility in use, simple and intuitive and so on, with possibility and innovation of science and technology.[[77]](#endnote-77)

Some components from Material design used in this project are:

* Recycler view - which is a new kind of Listview. Recycler view provides a better performance and it is easier to use.
* CardView - information can be displayed in a consistent cards
* Navigation drawer - it a sliding menu which provides navigation for different section of the app.
* Tabs - it enables for a better content organisation and make it easy to explore the app and switch between different views.

#### Login Screen

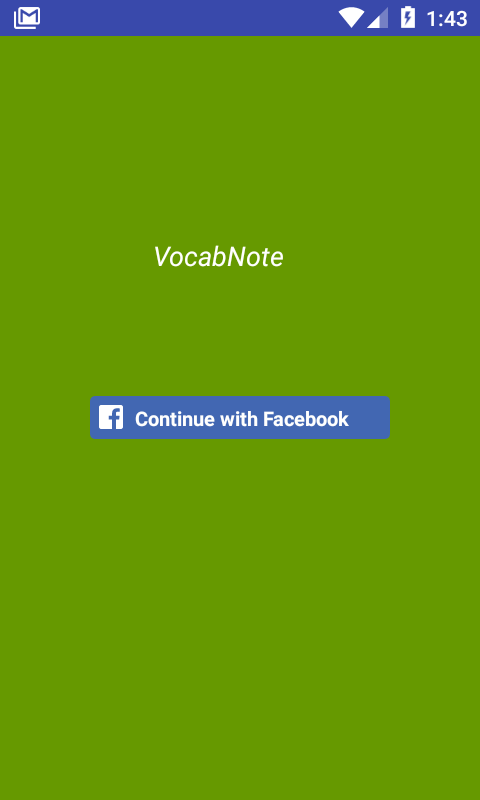


Figure 3.3 Screenshot of Login Screen

On the login screen, users can log into the app by using Facebook login API

After the authentication, the user's data(FacebookID authentication, name and email) is stored in the Django server; the Django server was deployed to PythonAnywhere. Figure 3.3 depicts the Login screen.

The Django server which sits on PythonAnywhere, stores data fetched from the Facebook API authentication.

Facebook is the most popular social media, therefore using facebook api to authenticate a user is safer , simple, intuitive and faster.

#### Main Screen

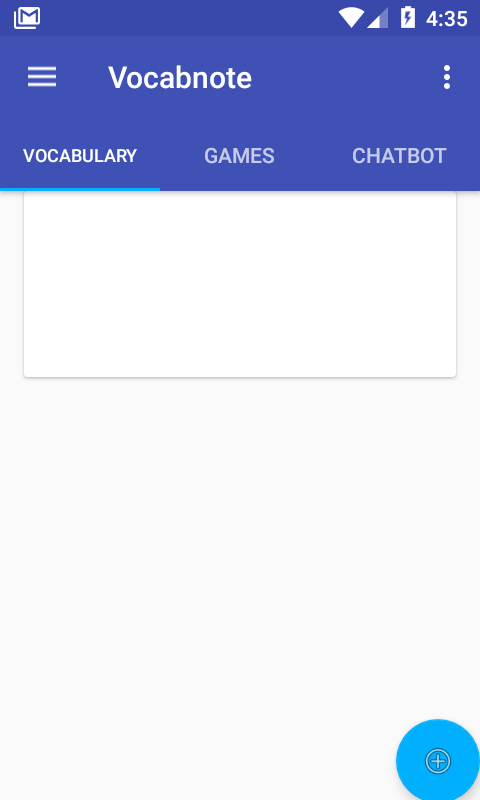


Figure 3.5 Screenshot of Main Screen after user logs in

On the Main Screen, users can explore the app by using the view tab and navigation drawer. Users will be able to access all the functionalities of the app, which are vocabulary, games, chatbot and Games stats.

On the vocabulary tab, they can add categories and list of words by tapping onto the add button on the bottom right corner and enter the categories and words as shown on the figure 3.5 and figure 3.6 below.

Figure 3.5 interface to Figure 3.6 interface to Add Categories Words

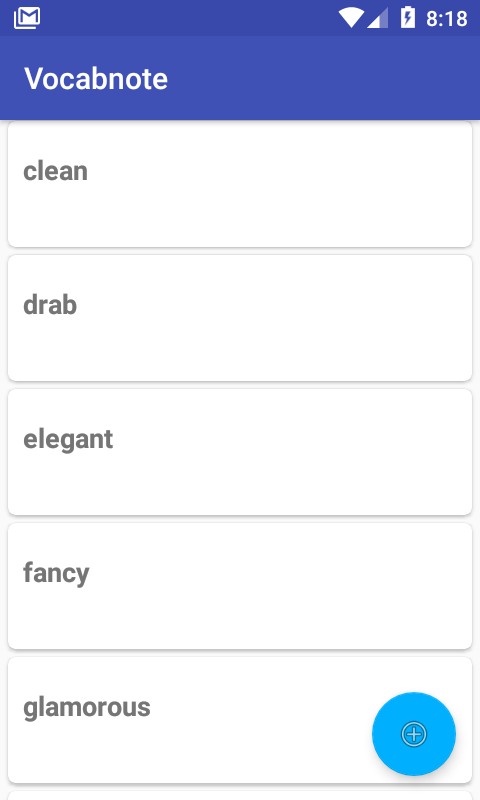
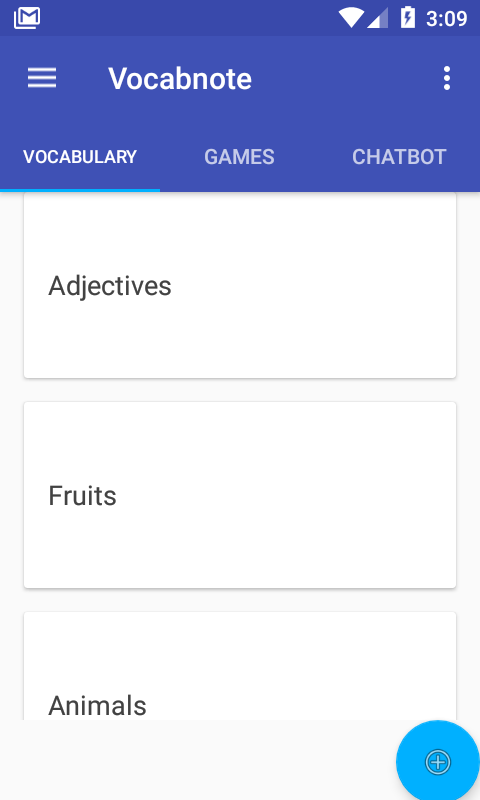
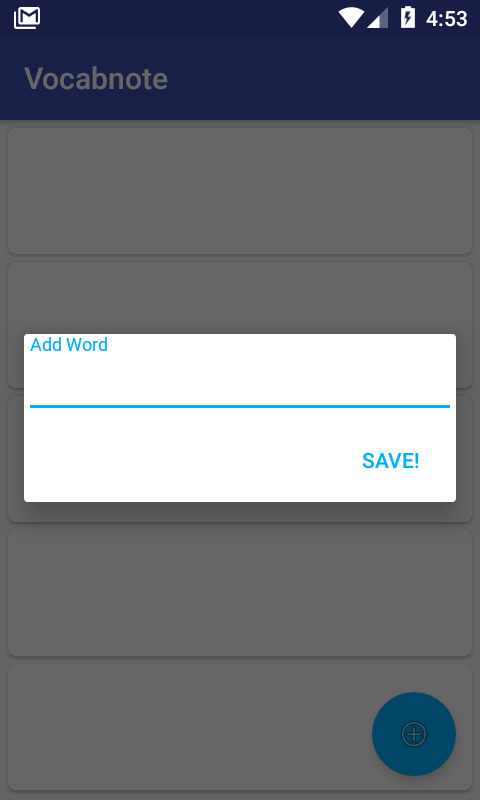
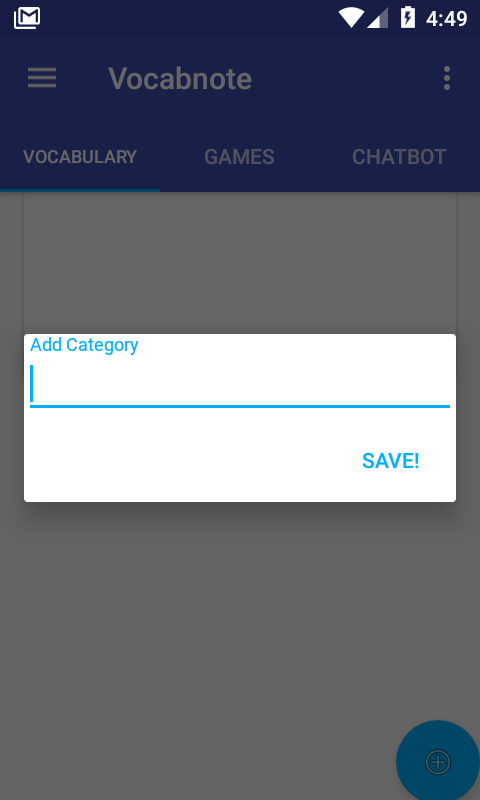


Figure 3.7 List of Categories Figure 3.8 list of words

A dictionary has been implemented, the users can select a word, and a dictionary will be displayed with many different meaning of a word, pronunciation, and examples of use as shown below.

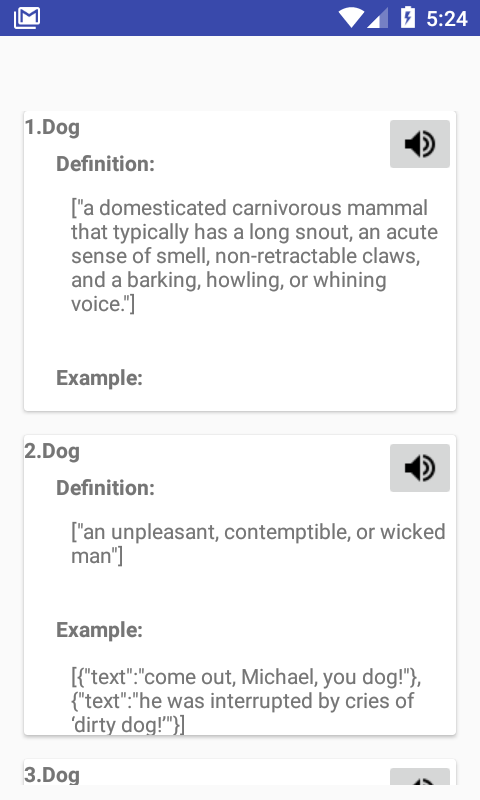


Figure 3.9 Screenshot showing dictionary

Vocabnote has two levels, to level up, users have to have 3000 points to unlock more functionalities in the app.

To achieve this score, users have to play the game which is available in level one, which is called Synonyms or interact with the chatbot available also in the same level.

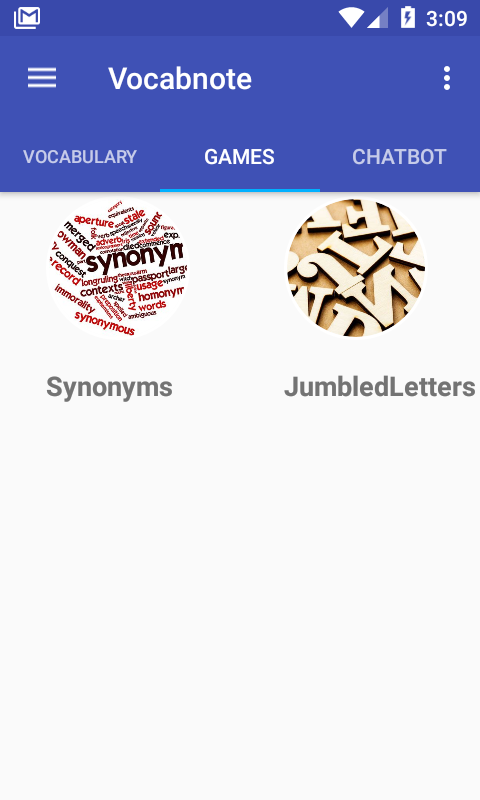
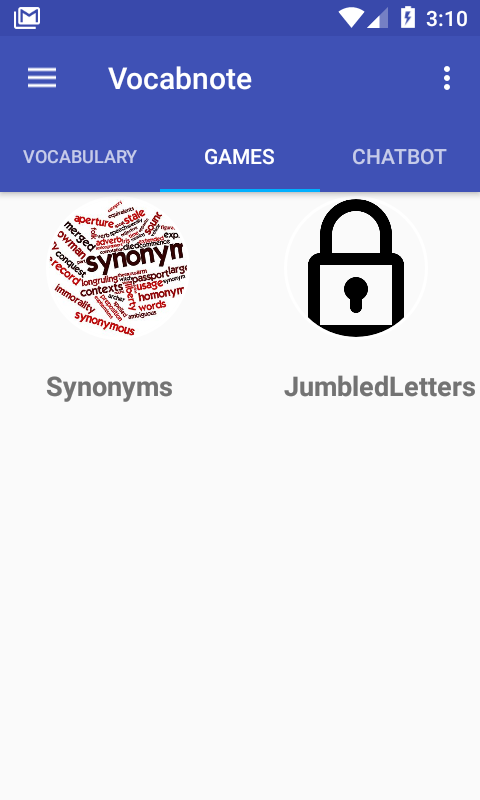


Figure 3.10 Game Screen Figure 3.11 Game Screen with level

with level 2 locked 2 unlocked

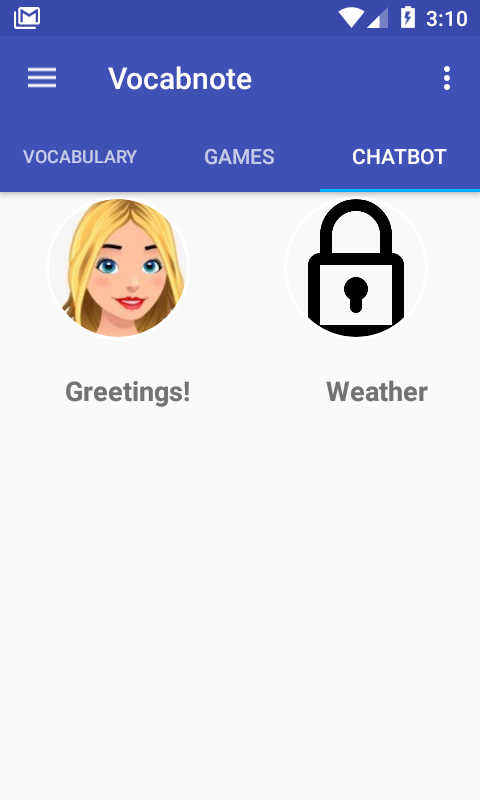
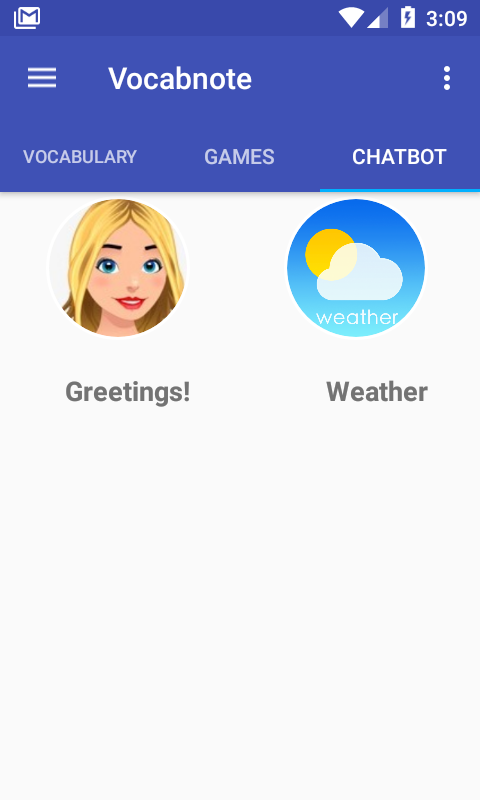


Figure 3.12 Chatbot screen Figure 3.13 Chatbot screen

level 2 locked level 2 unlocked

Vocabnote has two games, which are called synonyms and jumbled letters.

When users access the Synonyms game, they will be able to tap on the about button to check the description of the game, tapping on the instructions button, they will have access to the instructions on how to play and finally the play button, which will take them to the game.

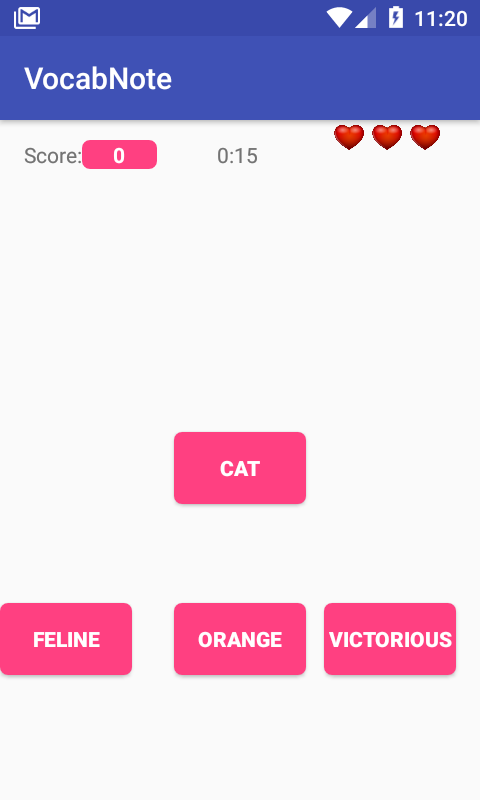
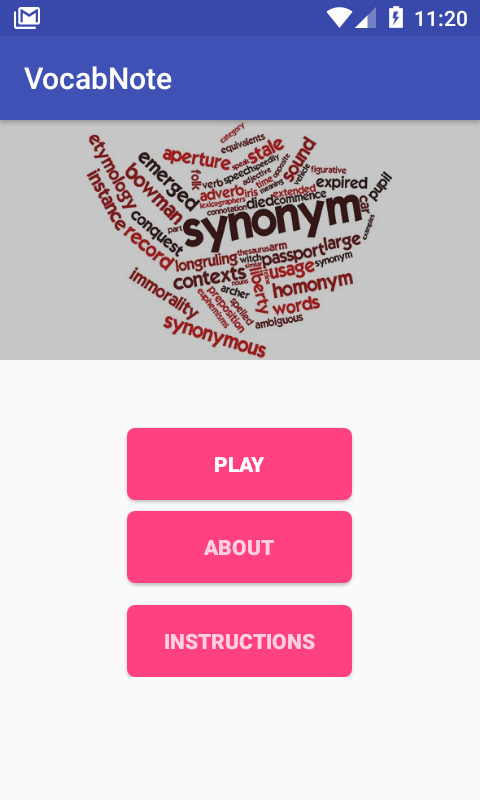


Figure 3.14 Synonyms game Figure 3.15 Synonyms game screen

menu screen

In synonyms game, users will have to guess the synonym of a word.

As shown in figure 3.15, there is the main word on top, which is Cat and three more words below. The words are generated randomly from the list of words entered by the users and the synonym of the main word is retrieved from Oxford dictionary API.

The synonym of cat is feline, if a user guesses it correctly, he/she will score points. If guessed incorrectly, points and a heart will be removed. The hearts means the attempts which users can guess incorrectly, if guessed incorrectly three times then it is game over!.

The primary objective is to achieve 3000 points to level up, which will unlock more functionalities in the app and to practice the vocabulary entered by the user.

After achieving 3000 points, users level up to level two and unlock another game called jumbled letters. The game jumbled letters has more aspects of gamification than the first game.

When accessing jumbled letters game as shown in figure 3.16, users will be able to see its menu. The menu also display the description of the game and instructions.

Users can choose between three difficulty levels, easy, medium and hard.

Level easy, only words with less than four-letters will be displayed, medium, the word length is less than six and hard, the word length is greater than six.

The average English word length is 5.1 letters.[[78]](#endnote-78)

Due to the average English word length and by asking eight testers to test the level of difficulty of the game, the appropriate word length for each level was chosen.

The countdown timer in jumbled game is set to five minutes.

Each word displayed in the game is chosen randomly from the user’s list of words. The letters are jumbled and users will have to guess the word by choosing the letters in order wich the word is written. Figure 3.17 displays the game. The word in figure 3.17 is “Whale”, so users will have to guess each letter in the right order.

If the letters are guessed in the wrong order, points will be removed each time it is guessed incorrectly.

In each level of difficulty there are a different point system, level easy, users can get 20 points for each letter guessed correctly, medium 50 points and hard 100 points.

The words can be very difficult to guess for the reason that the letters are jumbled, therefore below the letters, there are three ways in which users can get help. If a user taps on the lamp button, a window will pop up displaying the definition of the word. the hearts on the top right mean the attempts which the users have, by tapping on the heart on bottom below the letters, another attempt is added, by tapping on the letter dice, a letter will be displayed. Everytime users use any help, fifty points will be deducted from the score.

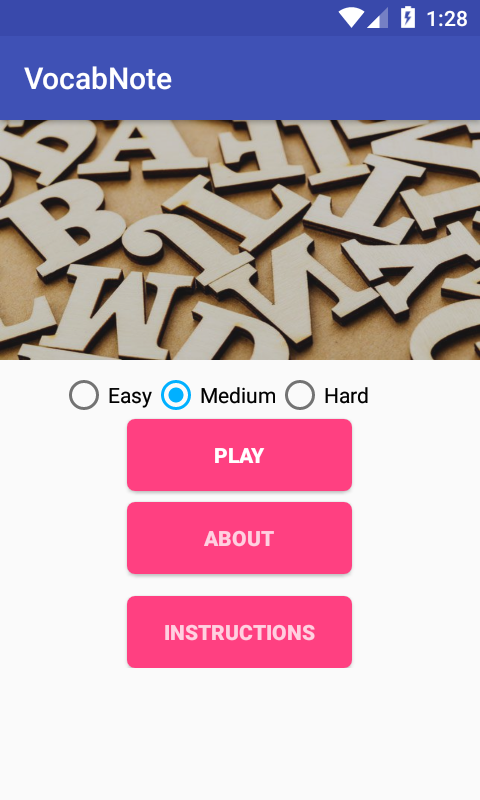
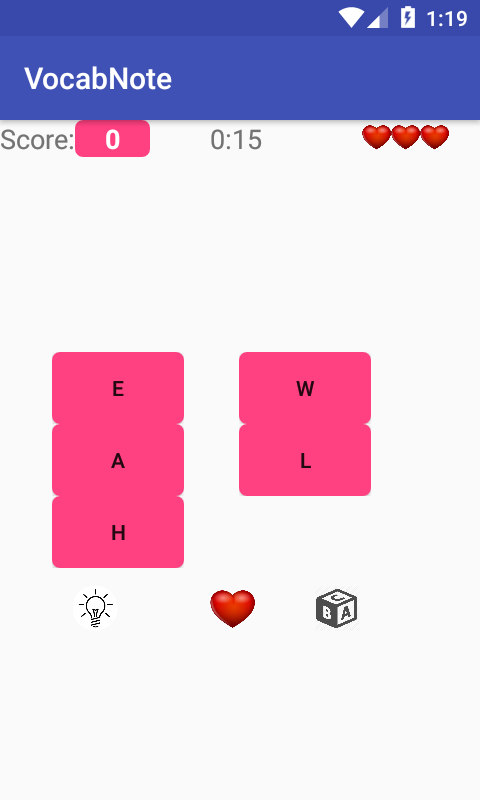


Figure 3.16 Jumbled letters Figure 3.17 Jumbled letters

game menu screen game screen

As shown in figure 3.18, a pop-up window will be displayed, showing the

Word guessed.

If users guess incorrectly many times and run out of attempts or time is up, a pop-up window displaying the game stats will be shown.

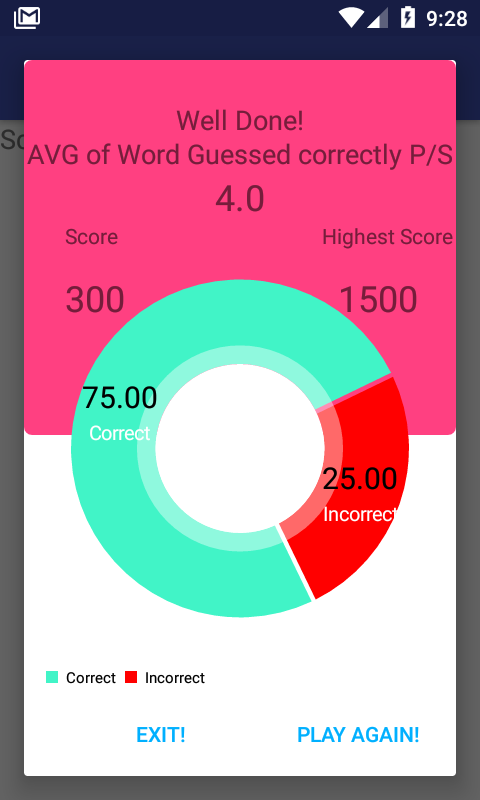
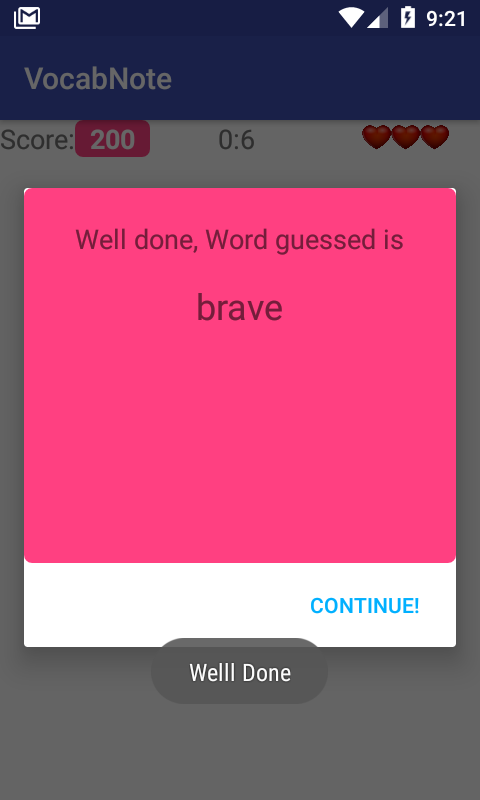


Figure 3.18 pop up window showing Figure 3.19 Games Stats.

The word guessed correctly.

At the end of the games, the game stats for each specific round will be displayed.

The game stats, give information about the performance of the users each time they play the games.

As shown above, in figure 3.19, users can see the average of words guessed correctly per second, highest score, current score, and a pie chart showing the average of words guessed correctly and incorrectly. The data is displayed mainly to motivate users to improve their performance and consequently learn English vocabulary faster.

Users can also access a chatbot. The chatbot was designed to prepare users for real-life conversations without anxiety and awkwardness, which can happen when having an English conversation in person.

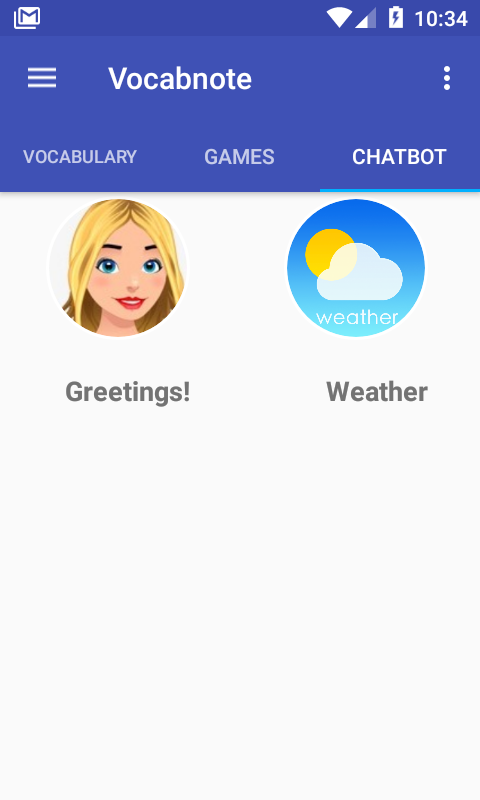


Figure 3.20 Chatbot screen

# Prototyping and Development

As described above, Extreme programming is the methodology chosen for this project. The project was broken down into five iterations; three were completed so far.

Using point-based as cited above, the iterations were estimated, the sum of the first three iterations is twenty-one points, the fourth has twenty-four and the fifth twenty-eight points.

It is easy to notice that the last two iterations have the most points, they are estimated to start at the beginning of December and the last iteration in January. The reason for that is time management, after completing the estimations for each iteration, it was decided that the following two months(December and January )will be ideal, more time can be allocated for the completion of the project.

Two user stories were allocated for iteration one:

* Develop a Django backend to store data
* Store Django backend on PythonAnywhere

## Iteration one: Django server development

Django is known for allowing developers to focus more on the frontend and it takes care of the backend, however as Django offer many functionalities, learning Django can be time-consuming, but it is worth for what Django has to offer.

The first step to develop a backend was to create models.

Models are descriptions of data; it contains attributes such as Name, Age, Date, among others, the most used term for models is tables.

After designing an ERD containing the three models: User, Classification, and Word, the development started.



Figure 6: Django models

Figure 6 illustrates the models chosen for this project.

The models in Django are written as python classes. Django automatically converts them into models and store in PostgreSQL.

After writing the models, the code below has to be executed To command Django to create them.

manage.py makemigrations

Subsequently, the code below has to be executed to send the models to the database.

Manage.py migrate

The User model has six attributes. The only attributes that are going to be displayed to the users are User\_Auth\_ID, User\_Name, and User\_Picture.

The other information about the users such as Age, Gender, and Email is stored in the database for future data analysis.

For example, the Age attribute can be analyzed to discover which age groups are using VocabNote the most, in order to develop functionalities such as gamification techniques and games, which are more appealing to specific age groups.

The classification model store the word classifications, e.g.(Animals, Food, Travels). The Classification and User models have a many to many relationships, that means a user can fetch and insert many word classifications, and a classification can be related to many users.

With only one line of code, Django allows the creation of this relationship.

(User = models.ManyToManyField(User, related\_name=**'Classification'**))

Using many other frameworks, it would be required to write more code to achieve the same behavior, and it would demand a creation of a weak entity, which is another model used only to link the two many to many models.

The last model is Word, all the words inserted by users will be stored in this model. It has a one to many relationships with the Classification model, that means one Classification can have many words.

These models can be linked by using this line of code below.

Class\_Name = models.ForeignKey(Classification, related\_name=**'Word'**, on\_delete=models.CASCADE)

These relationships suited best for this project to improve performance, avoid duplications and permit scalability.

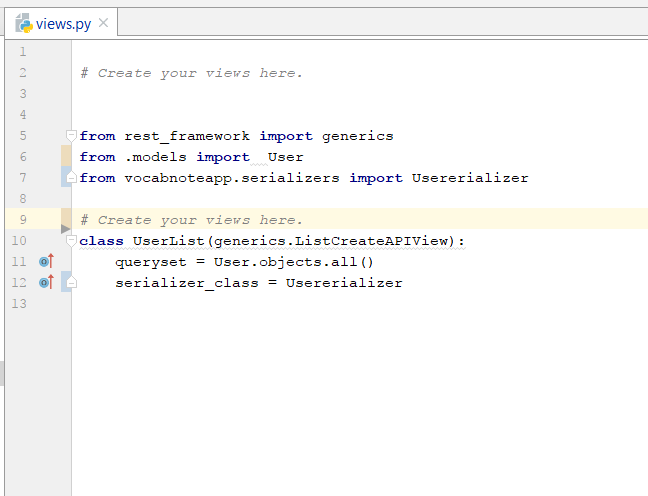


Figure 6.1: views.py

The view file in Django is used to take a request and return a response[[79]](#endnote-79); this is where one can write the queries to insert, update delete and fetch data.

Figure 6.1 illustrates the view file implemented for iteration 1. There are two ways to write views in Django:

* View functions[[80]](#endnote-80)

**def** current\_datetime(request):

now = datetime.datetime.now()

html = "<html><body>It is now **%s**.</body></html>" % now

**return** HttpResponse(html)

* Class-based-view

**class** UserList(generics.ListCreateAPIView):  
 queryset = User.objects.all()  
 serializer\_class = Usererializer

### 

Class-based-view is the chosen format to implement a view file, for the reason that it is the most convenient and efficient approach.

Class-based-views offer built in functions such as generics.ListCreateAPIView and generics.RetrieveUpdateDestroyAPIView.

These functions allow the creation, update, retrieval, and deletion of data with less code which improves the performance of the project.



Figure 6.2: Serializer.py

The Serializer.py file is comprised of three class. They are used to serialize the data stored in the database.

Serialize means, the conversion of objects or data structure from a format to another.

Figure 6.2 illustrates a Serializer file implemented for this project. The data is being serialized from a python format to JSON. JSON (Javascript Object Notation), is a data structure used to interchange data. Many APIs such as Github API use JSON format.

After the serialization is performed, an API is created to allow the communication between the frontend and backend.



Figure: 6.3: VocabNote API

Figure 6.3 illustrates an API with user information in a JSON format, which was implemented to connect VocabNote app to Django server.

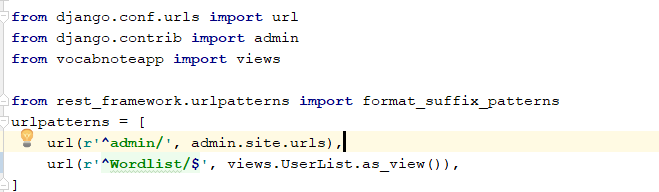


Figure 6.4: urls.py

Figure 6.4 illustrates the file where the URLs are written,

The URL used to find VocabNote API is the following:

http://jonatans.pythonanywhere.com/Wordlist/

## Iteration two

Three user stories were chosen to be implemented in interaction two:

* Add Facebook Authentication
* Connect to VocabNote API
* Retrieve data from VocabNote API

### Iteration two Add Facebook Authentication

With Facebook API, users will be able to login into VocabNote, the data retrieved from Facebook API will be stored in the database.

Implementing Facebook API login is very straightforward, Facebook provides a tutorial, which is very well documented.

The first step is to go to facebook API and create an app, which will give an ID number to connect to the API.

The second step is the implementation of login button.

<com.facebook.login.widget.LoginButton

android:id="@+id/login\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="30dp"

android:layout\_marginBottom="30dp" />[[81]](#endnote-81)

The Third step is to is to link the login button to a callback, which will check if the user is valid and retrieve the data and set permission which is the type of data required to be retrieved. Also If method onSucess is executed, a method called getInfoFB is then called, which assign values retrieved from Facebook API to strings.

*// findIds***loginButtonFB** = (LoginButton) findViewById(R.id.***login\_button***);  
  
**callbackManager** = CallbackManager.Factory.*create*();  
*//Permissions***loginButtonFB**.setReadPermissions(**"email,"**,**"public\_profile"**);  
*// Callback registration***loginButtonFB**.registerCallback(**callbackManager**, **new** FacebookCallback<LoginResult>() {  
 @Override  
 **public void** onSuccess(LoginResult loginResult) {  
 *// App code* getInfoFB(loginResult.getAccessToken());  
  
 }  
  
 @Override  
 **public void** onCancel() {  
 Toast.*makeText*(getApplicationContext(), ***TAG***, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** onError(FacebookException exception) {  
 Toast.*makeText*(getApplicationContext(),***TAG***, Toast.***LENGTH\_SHORT***).show();  
 }  
});

The fifth step is to retrieve the data from Facebook API

*// get Profile Info***public void** getProfileInformationFacebook(AccessToken accToken) {  
 GraphRequest request = GraphRequest.*newMeRequest*(  
 accToken,  
 **new** GraphRequest.GraphJSONObjectCallback() {  
 @Override  
 **public void** onCompleted(  
 JSONObject obj,  
 GraphResponse response) {  
  
 *//declaring variables* String fbId = **null**;  
 String fbEmail = **null**;  
 String FBUserName= **null**;  
 String fbGender = **null**;  
 String fbPropic = **null**;  
 **try** {  
  
 **if**(obj.has(**"email"**)){  
 fbEmail = obj.getString(**"email"**);  
 }  
 **else** {  
 fbEmail = **""**;  
 }**if**(obj.has(**"name"**)){  
 FBUserName = obj.getString(**"name"**);  
 }  
 **else** {  
 FBUserName =**""**;  
 }**if**(obj.has(**"gender"**)){  
 fbGender = obj.getString(**"gender"**);  
 }**else** {fbGender = **""**;}  
 fbPropic = **"https://graph.facebook.com/\"+ fbId +\"/picture?type=small"**;  
  
  
  
  
  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 Bundle parameters = **new** Bundle();  
 parameters.putString(**"fields"**, **"id,name,email,gender"**);  
 request.setParameters(parameters);  
 request.executeAsync();

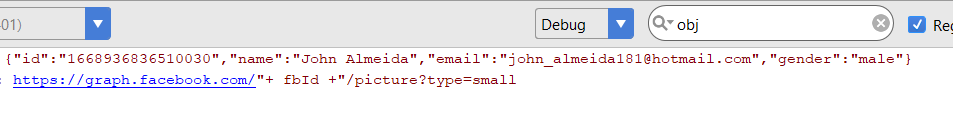


Figure 6.5 Facebook API data Retrieval

Figure 6.5 Illustrates the data retrieved from Facebook API.

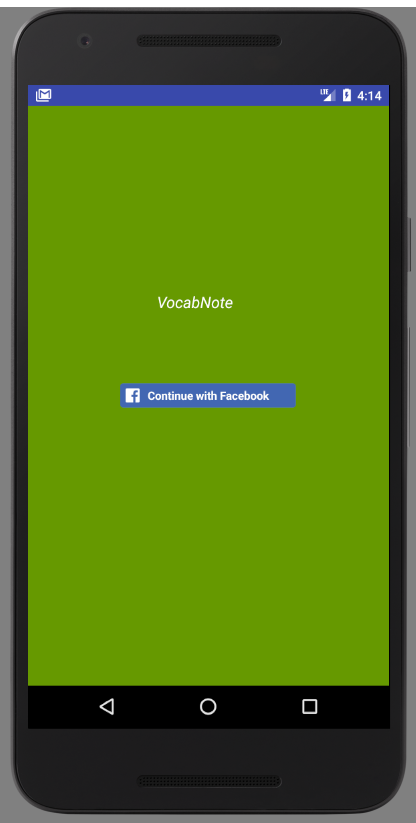


Figure 6.6: Login screen

Figure 6.6 illustrates the Login screen. In iteration four, a Google login button will be added. Users will be able to choose to login with each of them, having the two most popular login APIs allow for fast and safe login.

### Iteration two Connection and retrieval of data from VocabNote API

To achieve the connection with VocabNote API, a library called Retrofit is used.

Retrofit is a library which allows for convenient asynchronous connection with APIs, with a combination of another library called GsonConverterFactor,

it retrieves the JSON data from the APIs and transforms it into a java format.

Here is the implementation of Retrofit explained with comments on the code below.

**private void** getUserLists() {  
 Retrofit retrofit = **new** Retrofit.Builder()  
 .baseUrl(Api.***BASE\_URL***)  
 *//Here we are using the GsonConverterFactory to directly convert JSON data to object* .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
  
 Api api = retrofit.create(Api.**class**);  
  
 *// Call the API interface* Call<List<UserInfo>> call = api.getUserList();  
  
 *//Initiate the callBack* call.enqueue(**new** Callback<List<UserInfo>>() {  
 @Override  
 **public void** onResponse(Call<List<UserInfo>> call, Response<List<UserInfo>> response) {  
  
 *// assign List of objects to Users ArrayList* List<UserInfo> users = response.body();  
  
 *//loop trough UserClass Variable and assign words to UserWords* **for**(**int** i = 0; i < users.size(); i ++)  
 {  
 **UserClass** = users.get(i).getClassification();  
 }  
 Log.*e*(**"getClass"**, **UserWords**.toString());  
 Log.*e*(**"UserClass"**, **UserClass**.toString());  
 }  
 @Override  
 **public void** onFailure(Call<List<UserInfo>> call, Throwable t) {  
  
 *//display error on screen if fails* Toast.*makeText*(getApplicationContext(), t.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 System.***out***.println(**"On failure"**+getApplicationContext()+ t.getMessage());  
 }  
 });



Figure 6.7: Data fetched from VocabNote API

Figure 6.7 illustrates data fetched from VocabNoteAPI by using Retrofit and GsonConverterFactor libraries.

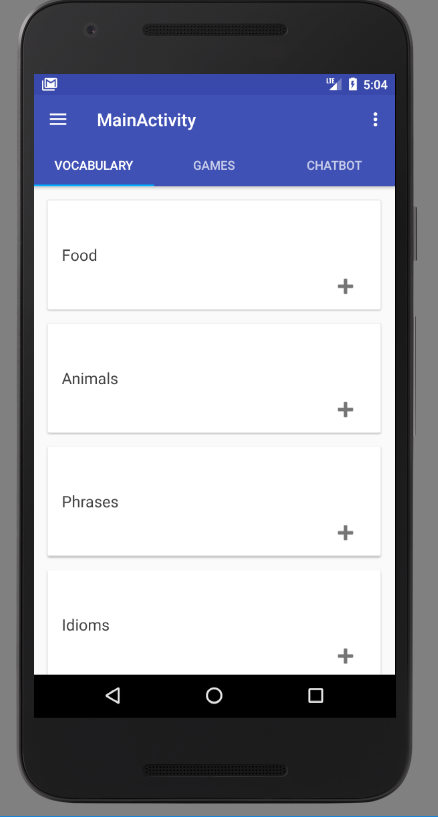


Figure 6.8: Main screen Vocabulary tab

The image illustrated in figure 6.8 is the main screen of the app. It is comprised of three tabs, Vocabulary, Games, and Chatbot. Each tab has a fragment.

Fragment is an interface which can be placed inside a screen\activity.

The vocabulary screen has a list of word classifications. In iteration four, a list of words will be implemented. User will be able to select a classification, which will direct them to a wordlist, e.g.(Apple, Orange),

# Testing

To test the app three English students were chosen, they are testing and giving feedback about the user interface and functionalities in each iteration.

The primary concern at the beginning of this project was to know whether the functionalities chosen were useful for the users.

In the first iteration, the testers were asked to focus on the functionalities and decide whether they thought those functionalities were useful or not. It turned out that one functionality had to be dropped.

At first, a spell checker was going to be implemented to correct grammar mistakes from users while they are interacting with the ChatBot.

However it was dropped after Two of the tester concluded that a spell checker was useless, for the reason that many built-in keyboards on smartphones have spellcheckers, therefore, implementing a spellchecker would only increase overhead.

## Unit testing

Unit testing is a testing technique used to test each unit of a system.

It tests each function and values from each unit.

Sometimes it is required to use the same variables in multiple methods of code, by accident a value can be changed without the notion of the developer, that can cause many unforeseen bugs, which can be hard to debug.

Unit testing solves this problem; a test case can be written with specific values which will be checked, in later iterations if those specific values are changed, then the testing case fails, which warns the developer where the change occurred.

Unit testing is an excellent technique to avoid bugs.

Extreme programming encourages the practice of writing test cases first, before the development, this approach will be used throughout the project development.

## Logcat

Android studio allows developers to test for correct values by using the Logcat window. The Logcat window output values, which developers wish to check.

In this project, Logcat was used several times to check the data being retrieved from VocabNote API. Below there is an example of using Logcat. The code below was used to check if the user profile data from Facebook API was bein retrieved.

Log.*e*(**"fbPropic"**,fbPropic);  
 Log.*e*(**"name"**,FBUserName);

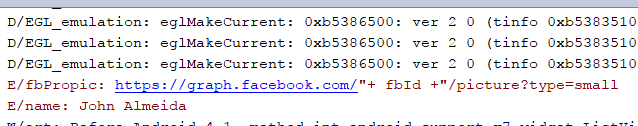


Figure 6.9: LogCat window, Facebook profile data

## Android Debugger

Android studios offer a debugger where one can examine the variable of values.

Android debugger was also used many times in this project.

To use the debugger, a breakpoint has to be set, then the app can run in debug mode, and a developer can examine the values from variables. The image below illustrates the use of Android debugger.

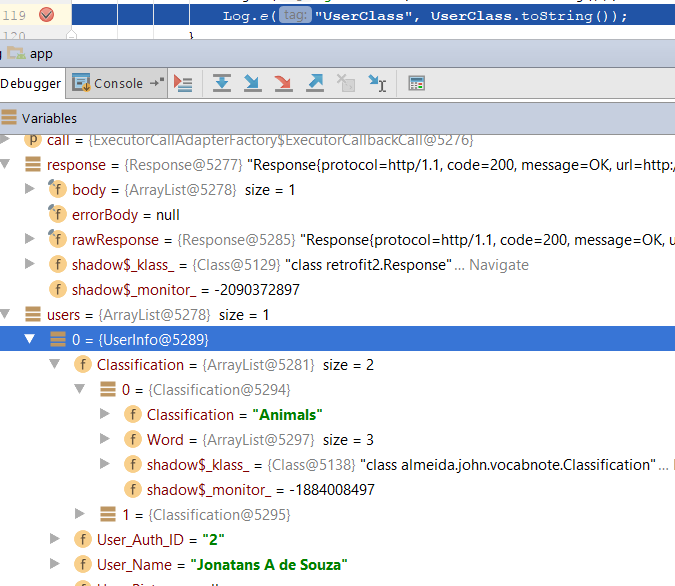


Figure 6.10 Android Debugger

# Issues and Risks

The primary challenge faced on this project so far is time management. Throughout the first two months, October and November, there were too many unrelated assignments which needed more focus, for the reason that they had an earlier deadline.

This issue was foreseen at the beginning of this project, to solve that, a minimum number of user stories were chosen to be implemented, and the rest of user stories will be implemented in the last two iterations.

As cited above, more time can be allocated for the completion of this project in the next two months(December and January).

Another challenge which is imminent is the training and development of a Deep learning model to implement a chatbot.

To train a model, it is required a lot of computer power, on a standard computer, it can take days to train a model, which is terrible for debugging. For example, a model can be trained, then the output from the model is not accurate, or it has many bugs. It would be time-consuming having to debug, improve and train it over and over again.

To solve this issue, a Cloud server will be used, specifically for that. There are many Cloud servers, which provide this service, the main ones are Google Cloud Platform and FloydHub.

Also, plenty of time will be allocated to learn Deep learning. TensorFlow provides an excellent tutorial, teaching how to uses its sequence to sequence model to develop a Neural Machine Translation. The implementation of a Neural Machine Translation has a similar algorithm to the one which will be used to implement a ChatBot.

There is also a course on Udemy called “Complete Guide to TensorFlow for Deep learning with Python” which will be completed prior to starting the ChatBot development.

The ChatBot will also have to be accurate with its answers; no swear words will be allowed, and it will have to use simple English so that users will understand it.

The goal is to find an English one to one conversation dataset. The reason for narrowing it for only one to one conversations is because the ChatBot will have a one to one conversation with each user, having a group conversation dataset can cost accuracy.

## Evaluation Criteria

This project can be considered successful if a ChatBot is implemented successfully and users find the games, learning strategies and user-interface useful, entertaining and user-friendly.

A significant risk, which can prevent users from continuing using the app, is lack of novelty.

After a while games can get tedious, the brain reward system can stop responding well to the games, which can drop enjoyment and motivation to keep playing and learning.

To mitigate that, feedback from testers and users will be welcome, more games or functionalities can be implemented depending on the input from users

# Plan and Future Work

There are two iterations left to complete the project.

The fourth iteration will start in December as estimated on the image below.

Figure 9 illustrates the user stories which will be implemented in December.

|  |
| --- |
|  |
| Figure 9: Iteration 4  The key deliverables for iteration four are An English dictionary, ChatBot and the implementation of progress analysis of each game.  Iteration five is estimated to start in January as depicted below.  Figure 9.2 illustrates all the user stories to complete the project.    Figure 9.2 Iteration 5  The key deliverables for iteration five are A FlashCard and the games. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Conclusions

This project has been excellent for my personal development. I have learned about many new technologies such as Django, PythonAnywhere, and Deep learning.

I have learned a lot about project management and project management tools to keep me productive.

The technologies learned from this project are very useful for a future career.

A seventy percent of the project still has to be implemented. However, I believe that this project will be finished by January, and will be published on Google play store.

# Bibliography

1. Catherine de Lange, ‘Educators Once Opposed Raising Bilingual Children. Experts Now Say It’s Beneficial.’, *Washington Post*, 11 June 2012, section Health & Science <https://www.washingtonpost.com/national/health-science/educators-once-opposed-raising-bilingual-children-experts-now-say-its-beneficial/2012/06/08/gJQAdz9gUV\_story.html> [accessed 16 November 2017]. [↑](#endnote-ref-1)
2. Thomas H. Bak and others, ‘Does Bilingualism Influence Cognitive Aging?’, *Annals of Neurology*, 75.6 (2014), 959–63 <https://doi.org/10.1002/ana.24158>. [↑](#endnote-ref-2)
3. ‘The Brain Benefits of Learning a Second Language’, *Be Brain Fit*, 2015 <https://bebrainfit.com/brain-benefits-learning-second-language/> [accessed 16 November 2017]. [↑](#endnote-ref-3)
4. Hélder Fanha Martins, ‘Perspectives on Business English as a Lingua Franca in Business Communication’, *Teacher Education and Curriculum Studies*, 2.5 (2017), 61 <https://doi.org/10.11648/j.tecs.20170205.11>. [↑](#endnote-ref-4)
5. Martins. [↑](#endnote-ref-5)
6. Lesson Nine GmbH, Lesson Nine GmbH, and Babbel.com, ‘The 10 Most Spoken Languages in the World’, *The Babbel Magazine* <https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world> [accessed 21 November 2017]. [↑](#endnote-ref-6)
7. Luca, ‘What Are the Most Studied Languages in the World?’, *Apps to Quickly Learn Spanish, French, Italian, German, Portuguese on IPhone, IPad and Android - MosaLingua*, 2015 <http://www.mosalingua.com/en/most-studied-languages-in-the-world/> [accessed 21 November 2017]. [↑](#endnote-ref-7)
8. Martins. [↑](#endnote-ref-8)
9. ‘Babbel.Com’ <https://home.babbel.com/prices> [accessed 15 November 2017]. [↑](#endnote-ref-9)
10. ‘The-Neuroscience-Joyful-Education-Judy-Willis-Md.Pdf’ <https://www.psychologytoday.com/files/attachments/4141/the-neuroscience-joyful-education-judy-willis-md.pdf> [accessed 21 November 2017]. [↑](#endnote-ref-10)
11. ‘The-Neuroscience-Joyful-Education-Judy-Willis-Md.Pdf’. [↑](#endnote-ref-11)
12. ‘Gamification’, *Coursera* <https://www.coursera.org/learn/gamification> [accessed 21 November 2017]. [↑](#endnote-ref-12)
13. Juho Hamari, Jonna Koivisto, and Harri Sarsa, *Does Gamification Work? — A Literature Review of Empirical Studies on Gamification*, 2014 <https://doi.org/10.1109/HICSS.2014.377>. [↑](#endnote-ref-13)
14. ‘13). Sze Lui.Pdf’ <http://www.nus.edu.sg/celc/research/books/4th%20Symposium%20proceedings/13).%20Sze%20Lui.pdf> [accessed 24 October 2017]. [↑](#endnote-ref-14)
15. Janna, erson, and Lee Rainie, ‘The Future of Gamification’, *Pew Research Center: Internet, Science & Tech*, 2012, p. the <http://www.pewinternet.org/2012/05/18/the-future-of-gamification/> [accessed 21 November 2017]. [↑](#endnote-ref-15)
16. ‘Gamification Market Worth 11.10 Billion USD by 2020’, p. market <https://www.marketsandmarkets.com/PressReleases/gamification.asp> [accessed 22 November 2017]. [↑](#endnote-ref-16)
17. ‘Healthcare Gamification Market Size - Industry Share Report 2017-2024’ <https://www.gminsights.com/industry-analysis/healthcare-gamification-market?utm\_source=globenewswire.com&utm\_medium=referral&utm\_campaign=Paid\_Globnewswire> [accessed 22 November 2017]. [↑](#endnote-ref-17)
18. ‘Healthcare Gamification Market Size - Industry Share Report 2017-2024’. [↑](#endnote-ref-18)
19. ‘Healthcare Gamification Market Size - Industry Share Report 2017-2024’. [↑](#endnote-ref-19)
20. ‘The Neuroscience of Gamification: 10 Things You Should Know’, *Growth Engineering*, 2016 <http://www.growthengineering.co.uk/the-neuroscience-of-gamification-in-online-learning/> [accessed 22 November 2017]. [↑](#endnote-ref-20)
21. ‘Duolingo’ <https://www.duolingo.com/> [accessed 26 November 2017]. [↑](#endnote-ref-21)
22. ‘Duolingo’. [↑](#endnote-ref-22)
23. ‘Khan Academy’, *Khan Academy* <http://www.khanacademy.org> [accessed 26 November 2017]. [↑](#endnote-ref-23)
24. ‘Duolingo’. [↑](#endnote-ref-24)
25. ‘Coursera | Online Courses From Top Universities. Join for Free’, *Coursera* <https://www.coursera.org/learn/gamification/home/welcome> [accessed 22 November 2017]. [↑](#endnote-ref-25)
26. ‘Coursera | Online Courses From Top Universities. Join for Free’, *Coursera* <https://www.coursera.org/> [accessed 26 November 2017]. [↑](#endnote-ref-26)
27. ‘Runkeeper Challenges’, *Runkeeper* <https://runkeeper.com/challenge> [accessed 26 November 2017]. [↑](#endnote-ref-27)
28. Denny Britz, ‘Deep Learning for Chatbots, Part 1 – Introduction’, *WildML*, 2016 <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/> [accessed 7 April 2018]. [↑](#endnote-ref-28)
29. {Citation} [↑](#endnote-ref-29)
30. {Citation} [↑](#endnote-ref-30)
31. {Citation} [↑](#endnote-ref-31)
32. {Citation} [↑](#endnote-ref-32)
33. Denny Britz, ‘Deep Learning for Chatbots, Part 1 – Introduction’, *WildML*, 2016 <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/> [accessed 18 December 2017]. [↑](#endnote-ref-33)
34. ‘What Is Natural Language Processing?’ <https://www.sas.com/en\_us/insights/analytics/what-is-natural-language-processing-nlp.html> [accessed 7 April 2018]. [↑](#endnote-ref-34)
35. ‘Learn a Language for Free’, *Duolingo* <https://www.duolingo.com/> [accessed 15 November 2017]. [↑](#endnote-ref-35)
36. ‘Learn a Language for Free’. [↑](#endnote-ref-36)
37. ‘DuolingoReport\_Final.Pdf’ <http://static.duolingo.com/s3/DuolingoReport\_Final.pdf> [accessed 22 November 2017]. [↑](#endnote-ref-37)
38. ‘Babbel.Com’ <https://home.babbel.com/prices> [accessed 15 November 2017]. [↑](#endnote-ref-38)
39. ‘Babbel.Com’. [↑](#endnote-ref-39)
40. ‘IDC: Smartphone OS Market Share’, *Www.Idc.Com* <https://www.idc.com/promo/smartphone-market-share/os> [accessed 25 October 2017]. [↑](#endnote-ref-40)
41. \* All products require an annual contract Prices do not include sales tax, ‘Mobile OS Market Share 2017’, *Statista* <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/> [accessed 1 November 2017]. [↑](#endnote-ref-41)
42. ‘Native-Apps-vs-Hybrid-Appsionic-Titanium.Pdf’ <http://www.ijcter.com/papers/volume-2/issue-6/native-apps-vs-hybrid-appsionic-titanium.pdf> [accessed 24 October 2017]. [↑](#endnote-ref-42)
43. ‘TIOBE Index | TIOBE - The Software Quality Company’ <https://www.tiobe.com/tiobe-index/> [accessed 27 November 2017]. [↑](#endnote-ref-43)
44. {Citation} [↑](#endnote-ref-44)
45. ‘Kotlin and Android | Android Developers’ <https://developer.android.com/kotlin/index.html> [accessed 1 November 2017]. [↑](#endnote-ref-45)
46. ‘Python vs R: Which Programming Language Is Better for Data Science?’, *BetaNews* <https://betanews.com/2017/10/09/python-vs-r-data-science-programming-language/> [accessed 1 November 2017]. [↑](#endnote-ref-46)
47. ‘R: What Is R?’ <https://www.r-project.org/about.html> [accessed 5 November 2017]. [↑](#endnote-ref-47)
48. ‘Python vs R’. [↑](#endnote-ref-48)
49. ‘Introduction - Laravel - The PHP Framework For Web Artisans’ <https://laravel.com/docs/4.2/introduction> [accessed 7 November 2017]. [↑](#endnote-ref-49)
50. ‘Slant - Django vs Laravel Detailed Comparison as of 2017’, *Slant* <https://www.slant.co/versus/1746/9622/~django\_vs\_laravel> [accessed 27 November 2017]. [↑](#endnote-ref-50)
51. ‘Django Overview | Django’ <https://www.djangoproject.com/start/overview/> [accessed 1 November 2017]. [↑](#endnote-ref-51)
52. ‘PostgreSQL: About’ <https://www.postgresql.org/about/> [accessed 7 November 2017]. [↑](#endnote-ref-52)
53. ‘MySQL | The Most Popular Open-Source Database | Oracle’ <https://www.oracle.com/mysql/index.html> [accessed 7 November 2017]. [↑](#endnote-ref-53)
54. Chloe Britton, ‘The Science behind Spaced Repetition’ <http://blog.wranx.com/the-science-behind-spaced-repetition> [accessed 22 November 2017]. [↑](#endnote-ref-54)
55. ‘How Quickly We Forget: The Transience of Memory’, *PsyBlog*, 2008 <http://www.spring.org.uk/2008/01/how-quickly-we-forget-transience-of.php> [accessed 22 November 2017]. [↑](#endnote-ref-55)
56. ‘Settles.Acl16.Pdf’ <https://s3.amazonaws.com/duolingo-papers/publications/settles.acl16.pdf> [accessed 22 November 2017]. [↑](#endnote-ref-56)
57. “ A Trainable Spaced Repetition Model for Language Learning by Burr Settles∗ August 7-12, 2016,” accessed November 22, 2017, https://s3.amazonaws.com/duolingo-papers/publications/settles.acl16.pdf. [↑](#endnote-ref-57)
58. ‘Settles.Acl16.Pdf’ <https://s3.amazonaws.com/duolingo-papers/publications/settles.acl16.pdf> [accessed 22 November 2017]. [↑](#endnote-ref-58)
59. ‘Settles.Acl16.Pdf’. accessed November 22, 2017, https://s3.amazonaws.com/duolingo-papers/publications/settles.acl16.pdf. [↑](#endnote-ref-59)
60. Britton. [↑](#endnote-ref-60)
61. ‘Settles.Acl16.Pdf’. accessed November 22, 2017, https://s3.amazonaws.com/duolingo-papers/publications/settles.acl16.pdf. [↑](#endnote-ref-61)
62. ‘Artykuł-Paula-Pimsleura-w-Czasopiśmie-Modern-Language-Journal-1967.Pdf’ <http://lifehacking.pl/wp-content/uploads/2015/09/Artyku%C5%82-Paula-Pimsleura-w-czasopis%CC%81mie-Modern-Language-Journal-1967.pdf> [accessed 22 November 2017]. [↑](#endnote-ref-62)
63. ‘Pimsleur Approach Review - Obsolete, But It Works!’, *Language101.Com* <http://language101.com> [accessed 22 November 2017]. [↑](#endnote-ref-63)
64. ‘Learning Strategy 10: The Leitner Flash Card System’, p. stra <https://www.virtualsalt.com/learn10.html> [accessed 23 November 2017]. [↑](#endnote-ref-64)
65. ‘Learning Strategy 10: The Leitner Flash Card System’. [↑](#endnote-ref-65)
66. {Citation} [↑](#endnote-ref-66)
67. ‘What Is an Artificial Neural Network (ANN)? - Definition from Techopedia’, *Techopedia.Com* <https://www.techopedia.com/definition/5967/artificial-neural-network-ann> [accessed 26 November 2017]. [↑](#endnote-ref-67)
68. tutorialspoint.com, ‘Artificial Intelligence Neural Networks’, *Www.Tutorialspoint.Com* <https://www.tutorialspoint.com/artificial\_intelligence/artificial\_intelligence\_neural\_networks.htm> [accessed 26 November 2017]. [↑](#endnote-ref-68)
69. tutorialspoint.com, ‘Artificial Intelligence Neural Networks’. [↑](#endnote-ref-69)
70. {Citation} [↑](#endnote-ref-70)
71. ‘Recurrent Neural Networks · Artificial Inteligence’ <https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/recurrent\_neural\_networks.html> [accessed 29 November 2017]. [↑](#endnote-ref-71)
72. {Citation} [↑](#endnote-ref-72)
73. ‘Intents’, *Dialogflow* <https://dialogflow.com/docs/intents> [accessed 8 April 2018]. [↑](#endnote-ref-73)
74. tutorialspoint.com, ‘Extreme Programming Introduction’, *Www.Tutorialspoint.Com* <https://www.tutorialspoint.com/extreme\_programming/extreme\_programming\_introduction.htm> [accessed 23 November 2017]. [↑](#endnote-ref-74)
75. ‘Extreme Programming Values’ <http://www.extremeprogramming.org/values.html> [accessed 23 November 2017]. [↑](#endnote-ref-75)
76. tutorialspoint.com, ‘Extreme Programming Practices’, *Www.Tutorialspoint.Com* <https://www.tutorialspoint.com/extreme\_programming/extreme\_programming\_practices.htm> [accessed 24 November 2017]. [↑](#endnote-ref-76)
77. {Citation} [↑](#endnote-ref-77)
78. ‘1208.6109.Pdf’ <https://arxiv.org/ftp/arxiv/papers/1208/1208.6109.pdf> [accessed 11 April 2018]. [↑](#endnote-ref-78)
79. ‘Class-Based Views | Django Documentation | Django’ <https://docs.djangoproject.com/en/1.11/topics/class-based-views/> [accessed 29 November 2017]. [↑](#endnote-ref-79)
80. ‘Writing Views | Django Documentation | Django’ <https://docs.djangoproject.com/en/1.11/topics/http/views/> [accessed 29 November 2017]. [↑](#endnote-ref-80)
81. ‘Android - Facebook Login - Documentation’, *Facebook for Developers* <https://developers.facebook.com/docs/facebook-login/android/> [accessed 29 November 2017]. [↑](#endnote-ref-81)