Nelson Mandela
Metropolitan
University

for tomorrow

English isiXhosa Mobile Vocabulary Learner

Author: *Zandré Botha*

Supervisors: *Marinda Taljaard & Natalie Tansley*

Submitted in partial fulfilment of the degree

Baccalaureus Scientiae (Honores) in Computer Science

at the Nelson Mandela Metropolitan University

31/10/2014

Acknowledgements

The author acknowledges and would like to thank the following people for their contribution towards this project:

1. First and foremost the project supervisors, Marinda Taljaard and Natalie Tansley, for their excellent guidance throughout the duration of the project.
2. All my friends and family, most notably my wife, for their encouragement and support.
3. The technical staff of the NMMU Computing Sciences Department who helped with the design and implementation of the app, and the design of the app's database.
4. All students of the NMMU Computing Sciences Department who participated in the two rounds of evaluation and provided objective feedback.
5. Ms Masive Zita for helping to make the audio files used in the app.
6. Mrs Lulama Mani for her guidance during the design of the tutorials.

Summary

There are a lot of mobile applications which aims to teach the user vocabulary of a foreign language. However, none of these applications are developed for South African languages. As a result, it was proposed to develop a mobile application that teaches the user basic vocabulary from either English or isiXhosa.

The proposed solution had to contain innovative and immersive tutorials to teach the user approximately 10 new words/phrases per tutorial. Each tutorial had to contain a series of lessons and exercises which the user must complete successfully before being allowed to move on to the next tutorial. The application also had to be designed in such a way that it can support any two languages, and that more tutorials can be added by the developer after the completion of this project.

A prototype application and database was then designed based upon extant systems and existing literature. A set of functional and non-functional requirements was also specified upon which the application's success was later measured.

After the implementation of the application was complete, an evaluation of the app was conducted. The first evaluation resulted in a list of recommended changes to be made to the app. Upon completion of all the required changes, a second evaluation took place. After analysing the results of the second evaluation, it was found that participants found the application useful, the information quality was satisfactory, and had a pleasant interface. Importantly 83% of the relevant participants said that they have learned new words by using the application.

Declaration of Originality

I declare that the entirety of the work contained in this treatise is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Nelson Mandela Metropolitan University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Zandré Botha

S214326195

Signature: ....

Date: ....

Table of Contents

Contents

List of Figures

List of Tables

Glossary

app

Short for *application*- Application software designed to run on mobile devices. In this treatise it is safe to assume that the Android platform is the default Operating System when 'app' is mentioned.

m-learning

Short for *Mobile Learning*- Any kind of learning that takes place with portable electronic tools (Quinn, 2000).

DSR

Design Science Research, the research methodology used in this project. DSR is a problem solving methodology that combines three different but closely related cycles of activities: The Rigor Cycle; The Relevance Cycle; and The Design Cycle. (Hevner, 2007)

RUP

Rational Unified Process, the software development methodology that will be used in this project. RUP one of the software methodologies that forms part of the Agile development methodologies.

Widget

A standard view to help create user interfaces on the Android platform (Meier, 2012).

- **TextView –** "A standard read-only text label" (Meier, 2012)
- **EditText –** "An editable text entry box" (Meier, 2012)
- **Chronometer** – "A TextView expansion that implements a simple count-up timer" (Meier, 2012)
- **ListView**- "A View Group that creates and manages a vertical list of Views, displaying them as rows within the list" (Meier, 2012)
- **Button** – "A standard push button" (Meier, 2012)
- **ImageButton** – "A push button for which you can specify a customized background image" (Meier, 2012)
- **CheckBox –** "A two-state button represented be a checked or unchecked box" (Meier, 2012)

- **RadioButton** – "A two state grouped button. A group of these presents the user with a number of possible options, of which only one can be enables at a time" (Meier, 2012)
- **ProgressBar** – "Displays a bar to the user representing how far the operation has progressed" (Android Developers, 2014c)
- **RatingBar** – "an extension of SeekBar and ProgressBar that shows a rating in stars" (Android Developers, 2014d)

Action Bar

"The action bar is a window feature that identifies the user location, and provides user actions and navigation modes. Using the action bar offers your users a familiar interface across applications that the system gracefully adapts for different screen configurations." (Android Developers, 2014a)

Toast

"Toasts are transient notifications that remain visible for only a few seconds before fading out. Toasts don't steal focus and are nonmodal, so they don't interrupt the active application." (Meier, 2012)

User Experience

William & Tullis (2013) believes user experience "include three main defining characteristics:

1. A user is involved
2. That user is interacting with a product, system, or really anything with an interface
3. The users' experience is of interest, and observable or measurable"

Satisfaction

"Satisfaction is all about what the user says or thinks about his interaction with the product. The user might report that it was easy to use, that it was confusing, or that it exceeded his expectation." (William & Tullis, 2013)

Learnability

“Learnability is the extent to which something can be learned efficiently. It can be measured by looking at how much time and effort are required to become proficient, and ultimately expert in using something.” (William & Tullis, 2013)

Chapter 1. Introduction

People who want to learn a new South African language can use the Internet or make use of the services provided by a large number of educational institutions, such as schools or universities. There are also numerous online and mobile apps (Siskin, 2009) available to help users learn foreign languages. However, very few of these online and mobile apps are developed for South African languages. This project will investigate the development of a mobile application which teaches the user basic vocabulary from either isiXhosa or English.

1.1 Background

An extant system called Lula Learning[1] was developed in 2013. The word 'lula' means *easy* in isiXhosa, therefore Lula Learning means *easy learning*. Lula Learning is an online system which uses text-based tutorials and exercises to teach the user general words and phrases in English or isiXhosa. However Lula Learning does not consist of any innovative tutorials- the whole system is text-based and doesn't support any type of media such as video or audio.

Mobile learning (henceforth m-Learning) may be defined as learning that takes place with portable electronic tools (Quinn, 2000), and the use of m-Learning is on the rise in the modern society (Cavus & Ibrahim, 2009; Koole, 2009; Kukulska-Hulme, 2009a). There has been a significant amount of research done in the field of m-Learning with regards to learning a new language. Cavus & Ibrahim (2009) explored the potential of learning a new language by making use of Short Message Service (SMS), and reported very positive results. Kukulska-Hulme (2009) investigated the potential benefits and drawbacks of m-Learning, with regards to learning a new language. More recently Sandberg, Maris, & de Geus (2011) developed a mobile application to teach primary school students new words like the characteristics of zoo animals. Sandberg *et al*. (2011) concluded that students are motivated to use m-learning techniques because it allows them to learn in their own time in an informal setting.

1.2 Problem Description

A wide range of mobile apps that can be used to learn a foreign language exists. However the existing apps that teach users South African languages are few, and limited

[1] lulalearning.tk

in functionality. There is a need for an app for South African languages that includes grammar exercises, listening exercises, and teaches the user how to form a sentence.

1.3 Project Goal

The main objective of this project is to develop a mobile app with which the user can learn basic vocabulary in English or isiXhosa. This app must have tutorials which are both innovative and immersive.

- Innovative, meaning the tutorials have to be presented in a way that is not normally found in a formal learning environment. Exciting colours, media such as audio files, and game-type exercises can all be regarded as an innovative way to learn a new language.
- Immersive, meaning the tutorials have to be interesting enough to keep the user engaged in an informal learning environment in his own time. The user must be interested enough in the app that he wants to invest his own time to learn the new language. Creating innovative tutorials and exercises will go a long way to ensure the app's immersiveness.

These tutorials will have to include grammar exercises, listening exercises, and lessons on how to form a basic sentence. The goal of this app is therefore not just to teach the user English or isiXhosa vocabulary (as extant systems such as Lula Learning did), but rather to teach the user the correct use of the vocabulary in different contexts, in an innovative way.

1.4 Scope

This project aims to investigate whether an m-learning app can be developed to successfully teach the vocabulary of a South African language. This app will *not* teach the user to speak a new language fluently; the app will merely teach how to use general words or phrases.

This application should be generic (it should be able to support any two languages), but for the purposes of this project the focus will be on learning English from isiXhosa, or vice versa. The app will include text-based tutorials and exercises, as well as audio for listening/pronunciation exercises. Extant systems such as Duolingo and Babbel (see Extant Systems in Chapter 2) use voice recognition for their pronunciation exercises, however this is considered as future work for this project.

1.5 Relevance to the domain

This project will provide access to free education. The finished app can be used by anyone wanting to learn vocabulary from either English or isiXhosa, or the app can be used to complement the content of formal learning courses where English or isiXhosa are taught.

This project also provides a platform for future work. The finished app will only support some of the English and isiXhosa vocabulary, but future developers can add more vocabulary or other languages because the app will be generic.

1.6 Risk Areas and Limitations

One of the risks involved in this project is the accuracy of the data to be used for the app. Electronic resources in general are scarce for South African languages, and as such the isiXhosa data will have to be verified by a expert speaker of isiXhosa before it can used in the app.

Another risk is the design of the tutorials and exercises. Because isiXhosa grammar differs so much from English grammar, a language expert needs to be consulted when designing the tutorials and exercises.

As mentioned in 1.4, this app will not include voice recognition. Voice recognition is utilized when a system/app includes pronunciation exercises in the content. This project considers voice recognition as future work as it falls outside of the scope of an honours project. However the app will have a self-evaluating pronunciation exercise which does not use voice recognition. In this type of exercise the user can record himself saying a foreign word, and match it to a audio file of a native speaker's pronunciation of the word.

1.7 Overview of Treatise

The remainder of this treatise is structured as follows:

- **Chapter 2** will review the research methodology, as well as the resources/developmental tools that will be utilized by this project.
- **Chapter 3** will describe the architecture of the application, give functional and non-functional requirements, provide a task list based upon the requirements, discuss an initial database design, and create initial screen designs for the app.
- **Chapter 4** explores the development and implementation of the application, along with the problems that was encountered and solutions to these problems.
- **Chapter 5** will evaluate this project based on the success criteria that was defined in Chapter 3, and provide suggestions as to what improvements can be made.
- **Chapter 6** concludes this project. It gives an overview of the entire project by discussing the literature review, prototype design, implementation and evaluation of the prototype. Future work is lined out, and a section for the author’s reflection on the project is included.

Chapter 2. Research Methodology

Design Science Research (henceforth DSR) is a research methodology that combines three different, but closely related cycles of activities (Hevner, 2007). DSR is a problem solving methodology, and in the case of this project the problem is to provide users with an app that teaches part of the English or isiXhosa vocabulary. This project will be using DSR because it provides guidelines on how to solve the problem. According to Hevner (2007), DSR has three main cycles:

1. The Rigor Cycle
2. The Relevance Cycle
3. The Design Cycle

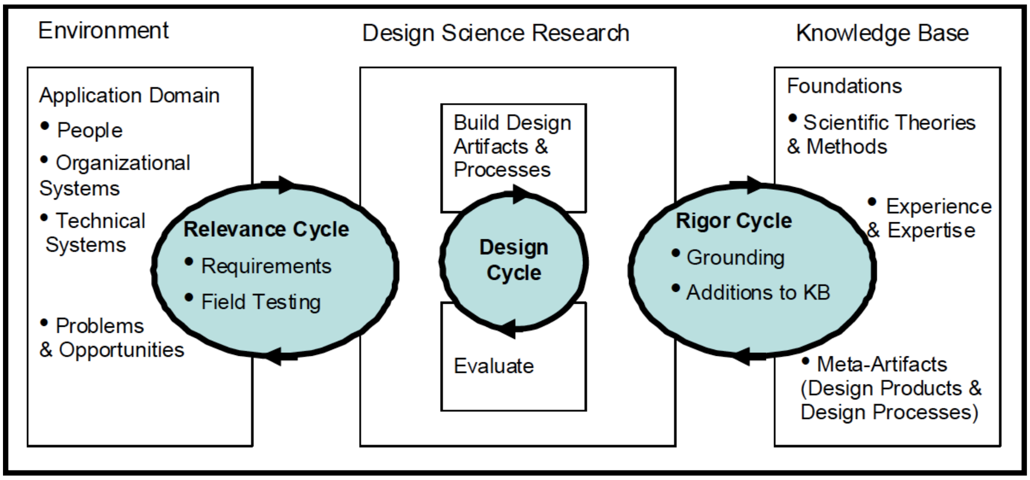


Figure 2-1 – Design Science Research (Hevner, 2007)

The purpose of each cycle is described in detail in the subsequent sections. Figure 2-1 gives a graphical representation of each of the cycles in DSR, and how they relate to each other.

2.1 The Rigor Cycle

The Rigor Cycle provides past knowledge of the application domain, and focuses on extensive research. Extant systems, experience, and expertise are all exploited as resources to gain a thorough understanding of the application domain. This ensures that the research done is well grounded and credible, and that designs contribute to the application domain as research artefacts.

This project explores five extant systems and compares them to each other to gain a thorough understanding of what has been done in the past, and identify what can still be done. Language experts are used to guide the development of the tutorials in a foreign language. Extensive research has been done to determine best-practices in the application domain.

This project will employ the following research methods:

- Survey Method- when gathering input from a language expert
- Exploration Method- gathering and filtering isiXhosa and English data to be used by the app
- Observation Method- when testing the app with users

The design cycle is where the application is created. This project uses RUP (see The Design Cycle) as the software development methodology. This cycle is mentioned last because it is dependent on the other two cycles- the requirements come from the relevance cycle, and the background knowledge of the application domain comes from the rigor cycle.

2.1.1 Related Work

This subsection presents a review of literature related to the application domain of this project. These include a mobile application to teach primary school users new words, a study to guide the development of learning material for mobile devices, and a study which explores the state of language learning apps.

2.1.1.1 The MEL app

In 2011, Sandberg *et al.* conducted an experiment to test the use of m-learning as a method to teach new words for users in primary school. They developed the Mobile English Learning app (MEL) which taught the user new English words, their focus being on the characteristics of zoo animals. MEL used different game types as an interactive way to engage with students, such as multiple-choice quizzes, spelling tests, memory games, yes or no games, and jigsaw puzzles. The students were divided into three different test groups:

1) The first group had normal classroom lectures about zoo animals

2) The second group also took classroom lectures but had use of the m-learning application on location in a zoo,

3) The third group received the same treatment as the second, but was allowed to take the application back to use at home for two weeks.

A pre- and post-test was then conducted, and results showed that the third group improved the most. Sandberg *et al.* (2011) concludes that formal learning, such as classroom lectures, can be complemented by informal learning techniques such as m-learning.

The MEL app is relevant to this project because:

1. This project also aims to teach the user new words. The difference is that this project aims to teach the user new words of a foreign language, whereas the students that tested the MEL were learning new words in their native language.
2. Some of the game techniques such as spelling tests or quizzes will be used in this project as well.

2.1.1.2 A Model for Framing Mobile Learning

Koole (2009) devised a model for the development of future m-learning applications. According to Koole, there are certain aspects which have to be taken into consideration when developing an m-learning application, such as the device aspect, learner aspect, and the social aspect. The device aspect includes considerations to the physical device itself, such as input and output capabilities, processor speed, etc. The learner aspect takes into account the users cognitive abilities, memory, prior knowledge, etc. The social aspect takes into account the social interaction and cooperation. The three circles in Figure 2-2 represent the device, learner, and social aspects which need to be taken into account when designing mobile learning systems, with the spaces where circles overlap representing attributes that belong to both aspects.

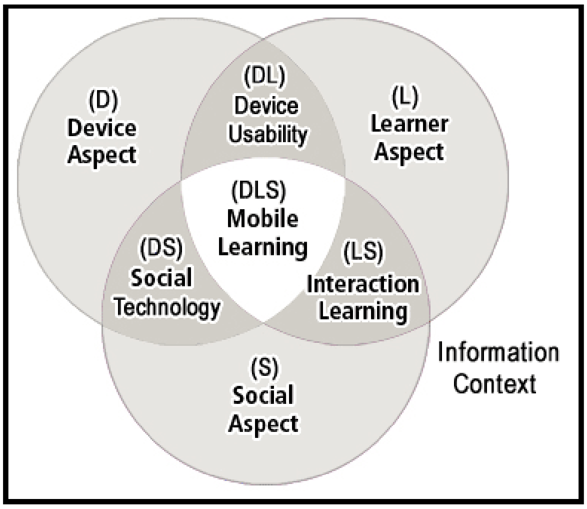


Figure 2-2 – The FRAME Model (Koole, 2009)

Koole argues that the primary intersection, which is the convergence of all three aspects, would hypothetically define the ideal learning situation for mobile learning. Developers can use the FRAME model to measure the degree to which they utilize each of the different aspects of mobile learning, in order to design more effective mobile learning experiences.

2.1.1.3 Language learning apps for mobile devices

A previous study by Godwin-jones (2011) explored rising m-learning applications with regards to learning a new language. This study provides various lists of language learning applications and resources used by m-learning developers, as well as analysing a few popular existing applications.

2.1.2 Extant Systems

This section explores extant systems that are similar in functionality to the mobile application proposed in this project. A brief description of each system's features, advantages, and disadvantages are given.

2.1.2.1 Aweza

Aweza is a mobile app that was released in 2014 for Android smartphones. This app is the first multi-lingual phrase translation app for South African languages. It allows the user to pick his native language and a language that he wants to learn. The user then records himself while speaking a word/phrase that is displayed on screen, and uploads the recording onto the app. Other users can then vote to validate or discard the recording. Figure 2-3 shows screenshots for the Aweza app. On the left is the Aweza home screen, in the middle the user can choose from different categories, and on the right is where the user can record a new phrase.

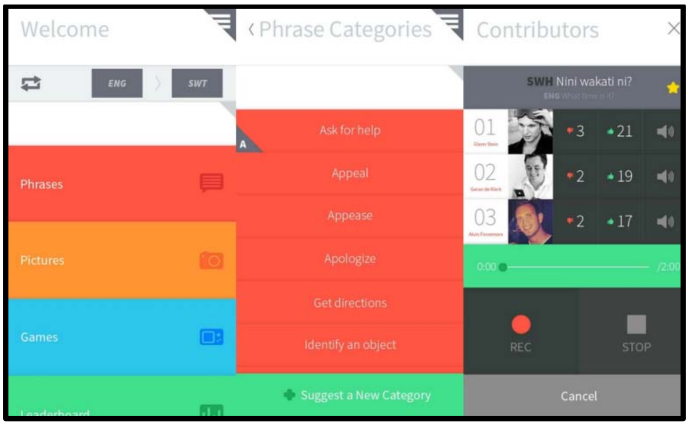


Figure 2-3 – Aweza

Aweza has the following features:

- Supports all 11 official South African languages
- Audio- plays word/phrases from audio file
- Recording- so that the user can record his own words/phrases
- Images are displayed of with the captions in the native language and another language that the user picks
- Crowd sourcing- the user uploads his recording, and other users can then vote on it
- A leader board which displays which users has the most uploads, positive, and negative votes

Aweza is free to download, but some users have found problems with registration. This app does not have any interactive or immersive tutorials where the user can learn how a sentence is formed or use grammar, the user only has the opportunity to learn words/phrases which the app displays. Aweza is not a very user-friendly app.

2.1.2.2 Speak easy Xhosa

Speak easy Xhosa[2] is an online system which aims at equipping the user with the language skills (frequently used phrases such as greeting or directions) to communicate in isiXhosa ("Speak easy Xhosa," 2014). This system provides tutorials, and then monitors how much the user has learned by testing the user after each tutorial. The system has three different levels that the user has to complete, each level more complex than the previous level. Speak easy Xhosa also provides a brief history on the isiXhosa language and its native speakers, the amaXhosa.

The Speak easy Xhosa system has the following features:

- Supports English and isiXhosa
- Online, text-based tutorials. These tutorials teach the user to use general words and phrases, as well as some grammar rules
- Audio
- Revision exercises to test the users knowledge
- Online tests, which is evaluated online and gives the user immediate feedback
- Video lectures

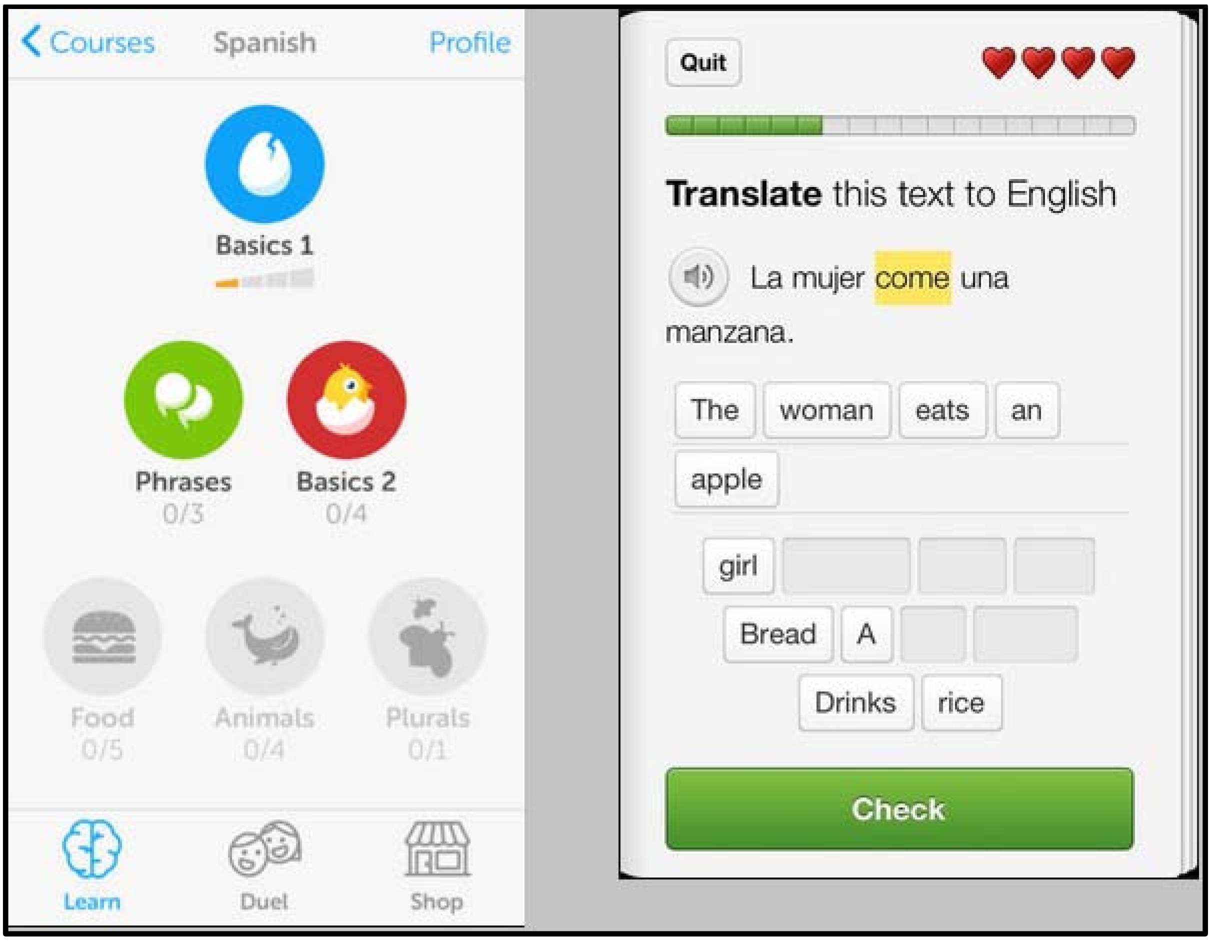
[2] http://www.speakeasyxhosa.co.za/

- Translation between English and isiXhosa

Although Speak easy Xhosa covers a broad range of topics, it does not have any interactive or immersive tutorials. The user learns all the content by text-based exercises and tests. Speak easy Xhosa also has an expensive subscription fee.

2.1.2.3 Duolingo

Duolingo is a mobile app that enables the user to learn foreign languages in a fun and exciting new way. Duolingo mainly supports European languages such as Spanish, French, German, Portuguese, Italian, and English. Figure 2-4 shows screenshots from the Duolingo app. The left screenshot is the user's progress in learning a new language, in this case Spanish. On the right is an example of an exercise that the user has to complete before he can move on to the next section.



Translate this text to English

La mujer come una manzana.

The woman eats an apple

Figure 2-4 – Duolingo

Duolingo has the following features:

- Writing exercises which teaches the user how to spell words

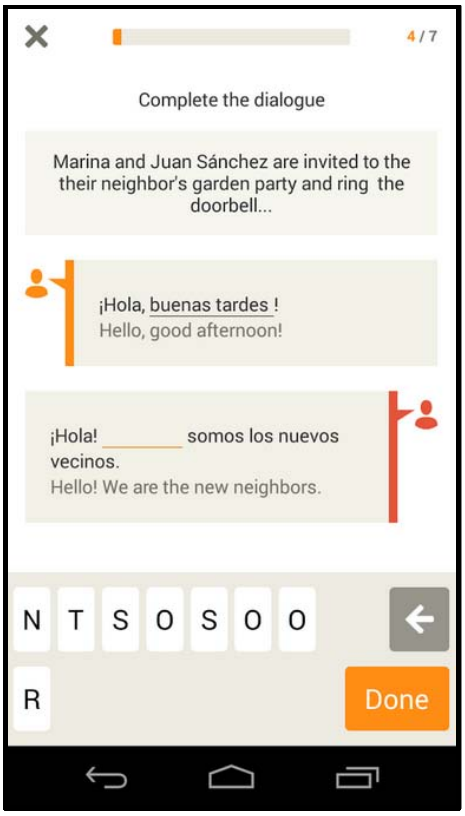
- Grammar exercises which teach the user how a sentence is formed in a foreign language.
- Speaking exercises- voice recognition is used to test the user's ability to say words and phrases in the foreign language.
- Listening exercises- the user has to listen, recognize, and then write down what he just heard from an audio file that was played in the foreign language

All exercises in Duolingo are presented in an immersive and interactive style. The app is easy to understand and very user-friendly. Duolingo is completely free of charge and doesn't contain any advertisements. The only drawbacks of Duolingo is that a user cannot edit their profile after subscription, and the voice recognition feature occasionally fails to pick up the user's voice.

2.1.2.4 Babbel – Learn Languages

Babbel is a mobile app which supports 13 languages (for example- French; German; Spanish) and enables the user to learn a foreign language through various tutorials and tests. Babbel is very similar to Duolingo in features and functionality. The tutorials are innovative and immersive, and the system overall is very easy to understand and use.

There also exists an online system for Babbel, which the user can then synchronize with his mobile app. With this synchronization feature the user can make use of either his mobile device or personal computer to learn the new language, but doesn't have to redo levels that have been completed on a different device. The user can choose his level of expertise in the foreign language and start at the appropriate section of the tutorials. Figure 2-5 shows a simple example of an exercise in the Babbel app.



Complete the dialogue

Marina and Juan Sánchez are invited to the their neighbor's garden party and ring the doorbell...

¡Hola, buenas tardes !
Hello, good afternoon!

¡Hola! ________ somos los nuevos vecinos.
Hello! We are the new neighbors.

Figure 2-5 – Babbel

Babbel is ahead of Duolingo in that it supports more languages, but only the first lesson of each category in the app is free of charge. If the user wants to continue beyond the first lesson of a new level, he will have to subscribe to the Babbel system (if he has not already done so) and purchase the subsequent lessons.

2.1.2.5 50 Languages

50 Languages is a mobile app that enables the user to learn how to speak short, fluent sentences in real-world situations. The app corresponds to the Common European Framework levels A1 and A2. Figure 2-6 shows the home screen on the left, the different languages that the user can choose to learn in the middle, and some of the content covered by the app on the right side.

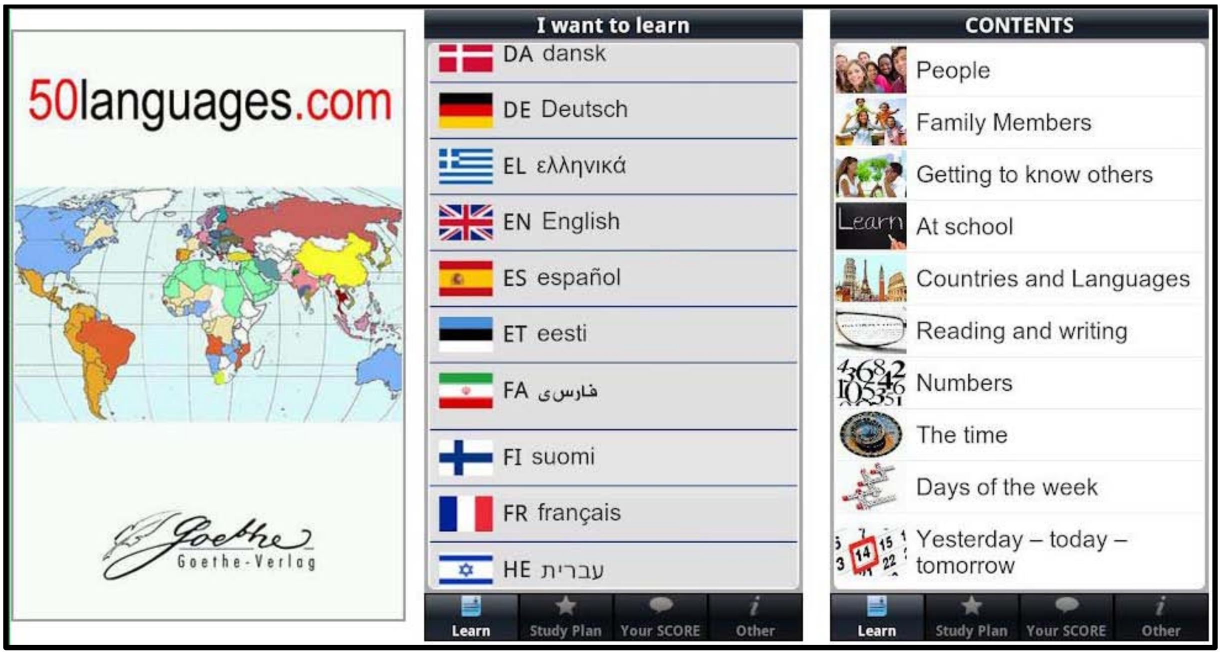


Figure 2-6 – 50 Languages

50 Languages supports the following features:

- Supports 50 languages
- Voice recording
- Audio- the correct pronunciation of a word/phrase is played
- Writing exercises

50 Languages contains a large amount of content. The free version of the mobile app contains 30 lessons, while the full version contains 100 lessons. 50 Languages is not as immersive and innovative as Duolingo or Babbel because the tutorials and exercises are not as interesting or exciting. The app is also not very user friendly. Other users have found audio files not to be working, and incorrect grammar and translations in the content.

2.2 The Relevance Cycle

Ultimately the goal of the relevance cycle is to improve the artefact. The relevance cycle defines the success criteria with which the application will eventually be evaluated on and focuses on how the application can improve certain areas in the application domain.

The relevance cycle also continuously identifies areas in the domain where improvements can still be made, and how the creation of artefacts can effect said improvements. However, by constantly identifying new areas for improvement new requirements will also be identified.

As in the case of this project, the relevance cycle identifies a problem- currently there is no mobile app that teaches users vocabulary of a South African language that is both innovative and immersive. How will such an app improve the application domain? How can the success of this app be measured? These questions will be answered in subsequent sections.

2.3 The Design Cycle

The design cycle is where the application is created based on the research gathered from the two other cycles. The design cycle is dependent on the rigor cycle for the research and expertise in the application domain, and the requirements of the application from the relevance cycle.

This project will make use of an Agile software development methodology. Agile development is an iterative or incremental way to develop software, and normally has a work cycle of two weeks (Agilemethodology.org, 2012). Agile software development continually revisits processes such as requirement and design so that the stakeholders can “inspect and adapt”. This also lowers the overall risk to the project, in addition to reducing the development cost of the software.

More than one Agile development methodologies exist, but this project will use the Rational Unified Process (henceforth RUP). RUP focuses on developing software incrementally (IBM, 1998). This allows for the development team to understand the problem better as the project progresses, and for requirements to change. Because the software development happens in increments the overall risk is reduced.

IBM (1998) defines the best practices that has been observed and proven successful in the industry. The six best practices, along with a brief description, as described below:

2.3.1 Develop Software Iteratively

After each increment, a finished portion of functionality is completed. Because each iteration ends with a completed portion of functionality, the development team stays focussed. Developing software incrementally also allows for changes in the requirements, features, and schedule.

2.3.2 Manage Requirements

RUP focuses on using tools such as use-cases and scenarios to document, organize, and track functional requirements.

2.3.3 Use Component-based Architectures

RUP focuses on developing an initial version early. It describes how to build a prototype that is flexible, can handle changes, is easily understandable, and is reusable.

2.3.4 Visually Model Software

A visual representation of the system allows the development team to more thoroughly understand the system, and how the different component fit together. Unified Modelling Language (UML) can be used to visually model software successfully.

2.3.5 Verify Software Quality

In order for a system to be deemed successful, it must meet the requirements specified at the start of the project. The quality of software should be tested for reliability, functionality, application performance and system performance, using guidelines specified by RUP. Verifying software quality should be done during each increment of the RUP life cycle.

2.3.6 Control Changes to Software

This process describes how to control, track, and monitor changes to the project. This allows the development team, as well as end users to see how the project has grown.

The RUP development methodology was chosen for this project because it has different components that vary in nature and complexity. Each of these components will be completed and tested separately, forming one increment of the RUP life cycle. The final increment will be to connect the different components and test the system as a whole.

2.4 Developmental Tools

The following resources will be utilized during the development of this project:

2.4.1 The Android Platform

The application developed by this project will be developed to run on the Android platform. Android is the chosen platform for the following reasons (Meier, 2012):

1. It's a free, open-source operating system for any compatible device which includes mobile smartphones and tablets
2. The development platform is open-source
3. The Android platform, unlike the iPhone/iOS and Blackberry, supports a wide variety of different devices
4. Anyone can develop for Android, no certification is required
5. Android apps does not require approval for distribution
6. Developers can use the Google Play store to distribute and sell their apps free of charge
7. According to DStv Online (2012), Android is the second most used platform for mobile devices in South Africa

2.4.2 Interface Development

2.4.2.1 The Eclipse IDE

This project will make use of Eclipse, along with the Android Developer Tools (ADT) plug-in. Eclipse will be used for this project as it is open-source, and also the recommended development platform for Android applications. The ADT plug-in for Eclipse simplifies development for Android project, and integrates some of the following features into Eclipse (Meier, 2012):

- An Android Project Wizard, which includes basic templates for Android app's
- Automatic building of Android projects
- The Android Emulator
- Runtime debugging, which enables the developer to set breakpoints and view call stacks
- All Android log and console outputs

2.4.2.2 Justinmind Prototyper IDE

Justinmind's prototyper is freely available to download and will be used to design initial Visual User Interfaces (VUI's) for the app. The IDE is user-friendly and provides a wide range of mobile layouts and widgets for Android and iOS for the user to model his VUI design.

2.4.3 Data

Initially the data to be used by this project was provided by the Counsel of Scientific and Industrial Research (CSIR). This data consists of parallel aligned words in English and isiXhosa. However, after the data has been evaluated by a language expert, it was decided not to make use of the CSIR data. The CSIR data was discarded because:

1. **Irrelevant entries**- Words such as “desert” (“ntlango” in isiXhosa) which are not useful for a novice when learning a new language.
2. **Missing entries**- Some of the English entries does not have a corresponding isiXhosa translation. As an example, “possess” (entry 648 in the CSIR data) does not have a translation to isiXhosa.
3. **Wrong translations**- According to the language expert who evaluated the data, several of the translations from English to isiXhosa are incorrect.

After consulting with the project supervisors, it was decided to rather make use of language experts to decide what data to use for the app and what content to cover when teaching a new language.

2.4.4 SQLite

The Android platform provides a lightweight relational database for each application via SQLite (Meier, 2012). By default each database can only be accessed by the application that created it. SQLite only allow data to be stored as primitive types, such as integers, floats, doubles, and strings. SQLite is chosen for this project because:

1. It is open-source
2. It has a reputation of being reliable
3. It is lightweight
4. The database is running as a separate ongoing process which reduces external dependencies and minimizes latency.

2.4.4.1 Database Browser for SQLite

Database Browser for SQLite is a software tool that is freely available on sourceforge.net. This tool allows non-technical user to create and edit SQLite databases using a spreadsheet-like interface.

2.4.5 MySQL Workbench

MySQL Workbench will be used for the implantation of the remote database (see Figure 3-3), and is also freely available to download at http://www.mysql.com/. MySQL Workbench is a unified visual tool database architects, developers and DBA’s (ORACLE, 2014), and provides a wide range of features such as:

- Data modelling
- SQL development
- Administration tool for server configuration

2.4.6 Audacity

Audacity is a software tool that allows the user to edit and record audio files (Audacity Developers, 2014). Audacity is open source software that is free for download for Windows, MAC, Linux and other operating systems, and can be translated into various different languages. Audacity will be used in this project to edit the recordings for the tutorials (see Recordings in Chapter 4), ensuring good clear quality and volume.

2.5 Conclusion

From the discussion above it can be concluded that a need for an app that teaches the user English and isiXhosa vocabulary exists. This app will be designed to run on the Android platform, as software to develop for Android devices are freely available and the application will then be usable to a large amount of South African users (DStv Online, 2012).

This project will use DSR as research methodology. DSR is comprised out of three different but related cycles. The rigor cycle focuses on research and provides all past-knowledge of the application domain. The relevance cycle defines the success criteria for this project. The design cycle is where the application is created, based on the research done in the rigor cycle and the criteria it has to meet from the relevance cycle. This project uses RUP as a development methodology, which focuses on compartmentalizing a project into small fragments and then developing software incrementally.

The next chapter will discuss the application domain, the functional and non-functional requirements of the app, a recommended task list, and the initial designs for the user interface and database.

Chapter 3. Prototype Design

This chapter falls under The Design Cycle of DSR and carries out the inception phase of the RUP developmental methodology. Firstly the application domain is analysed to determine what requirements should be met by the app. A use case diagram is then illustrated to determine how someone would interact with the app, and in what ways they would interact with it. Functional and non-functional requirements is specified and explained in detail. After all the requirements are gathered from the analysis of the application domain, the use case diagram, and the functional and non-functional requirements, a recommended task list, initial database and screen designs are developed for the prototype.

3.1 Analysis of Application Domain

To identify what the app must provide to the user, an analysis of the application domain is required. This analysis forms part of The Relevance Cycle in Chapter 2. The purpose of this analysis is to gather requirements, which in turn forms part of The Design Cycle in Chapter 2.

The target user for this app is anyone who wants to learn English or isiXhosa. The tutorials in the app will require the user to have at least a primary school level understanding of one of the languages. For example, if the user's native language is English and he wishes to learn isiXhosa, he must at least have a primary school education to be able to complete the tutorials in the app. Users who are already learning English or isiXhosa in a formal setting, such as a school or university, will also benefit from using this app because it is meant to complement formal learning material by providing the user a way to learn in an informal setting in his own time.

3.2 Use Case Diagram

A use case diagram is described by Bentley and Whitten (2007) as a graphical description of who will use the system and in what ways the user expects to interact with system.

Figure 3-1 represents the use case diagram for the Molo[3] app. In this diagram there are two actors present- the user and the developer. An actor is anything that needs to interact with the system to exchange information (Bentley & Whitten, 2007). The user

[3] *Molo* is the name of the software artefact created by this project

in this case refers to anyone who uses the Molo app to learn a new language. The developer is the person who writes the code for the Molo app and maintains it after deployment.

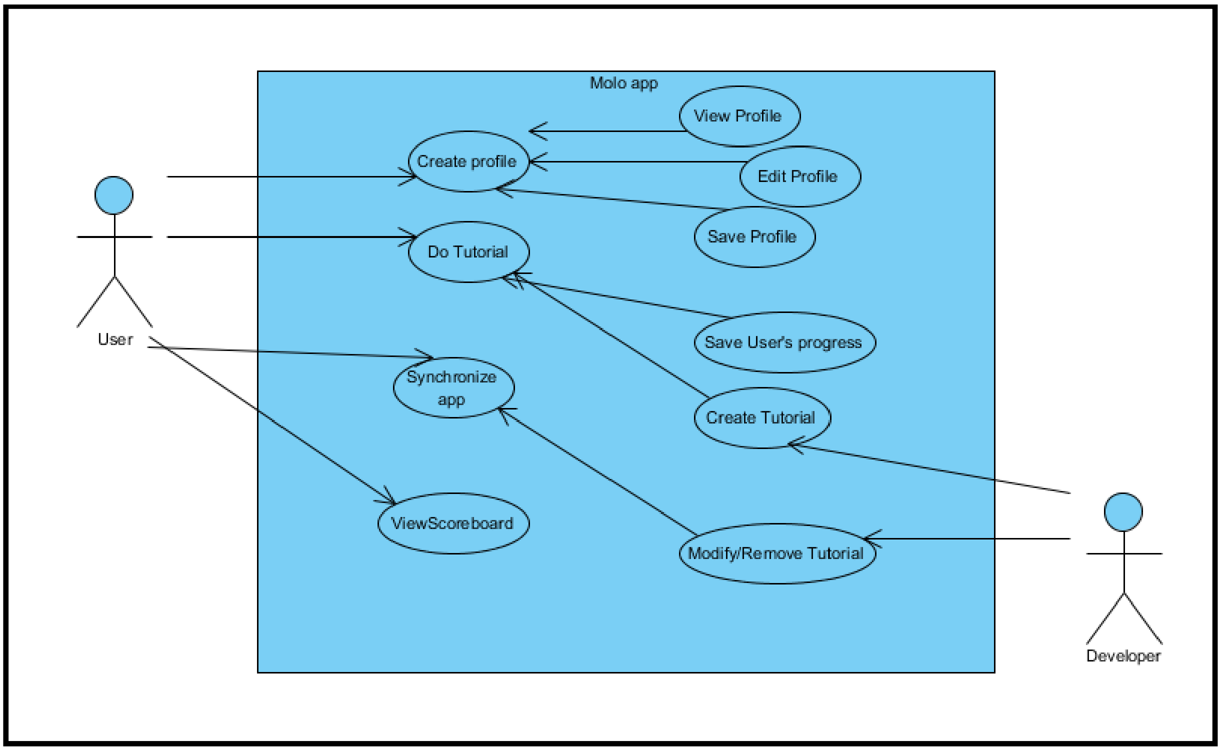


Figure 3-1 – Use Case Diagram

Four main use cases are specified for the user to interact with in Figure 3-1. The first use cases allow the user to create, edit, view and save his profile. The second use case allows the user to take part in tutorials and exercises, after which the app stores the user's progress. The third use case allows the user to synchronize the app if tutorials have been added, modified or removed by the developer. The fourth and final use case allows the user to view the progress that he has made.

Two use cases are specified for the developer. The first use case enables the developer to create a tutorial, and the second use case allows him/her to make changes to existing tutorials.

3.3 Requirements

The use case diagram in 3.2, which depicts a graphical illustration of how users would use the app, provides a rough guideline as to what requirements the app must fulfil. The subsequent subsections outline the functional and non-functional requirements that the app must satisfy.

3.3.1 Functional Requirements

Functional requirements describe the functions the app must provide for the user. The Molo app will include the following functional requirements:

1. The app will allow the user to create a new profile. When the user creates the profile, the system must check if it does not already exist in the database. The user must also be allowed to edit and save their profile.
2. The app must include tutorials to teach the user important vocabulary (as decided by a language expert) from a foreign language. The tutorials will be integrated with exercises which the user must complete before he can move on. It is important to note that the app produced by this project will only teach small sections of English and isiXhosa. Each tutorial should teach the user approximately 10 new words/phrases. Teaching an entire foreign language falls outside of the scope of this project. The tutorials will include the following:
    a. How to greet a person in the foreign language.
    b. How to formulate a question in the foreign language.
    c. A category for the different parts of a human's body.
    d. A small part of the foreign language's grammar. As an example, the tutorial might teach the user what the correct way of addressing a person is in a formal or informal manner.
    e. New Lessons. This part of the tutorial teaches the user a new word or phrase. The user will also have the opportunity to listen to a recording of the correct pronunciation of the new word/phrase.
    f. Voice exercises. An audio file of some word/phrase from the foreign language is played. The user then records himself speaking the same word. Afterwards the user can listen to his recording and evaluate how good his pronunciation of the foreign word was.
    g. Writing exercises, where the focus is on the spelling of foreign words.
    h. Multiple choice exercises, where the user is given a hint and a series of potentially correct answers. The user must then pick the correct answer in the series.
    i. Matching exercise, where the user must make more than one association between word/phrases from his native language and the foreign language.

Points e - i above is mentioned last because they will be integrated into points a – d.

3. The app must store the user's progress. When a user has completed a tutorial, he should not have to complete the same tutorial again the next time he logs into the system.
4. The user must be able to synchronize the app's local database with the remote database.
5. The user must be able to view the progress that he has made. This will include a graphical illustration of a scoreboard. The scoreboard will show the user's completed tutorials, along with the time that it took for the user to complete the tutorial and the score he achieved.

3.3.2 Non-Functional Requirements

Non-functional requirements describe the properties or qualities of the app. The non-functional requirements for this project have been split into the following categories:

3.3.2.1 Learnability of the app

The Molo app will be user-friendly, meaning that new users will not have any difficulty to learn how to use the app. The app will adhere to the Android developer guidelines (Android Developers, 2014b) will be used to ensure the consistency of the app's theme, so new users will not have any difficulty in navigating around the app.

3.3.2.2 User Satisfaction

The user must feel satisfied with the app's performance during use. The app will satisfy the user experience by not showing any adverts or making the user wait for long periods of time. A colourful interface could also contribute to the user's satisfaction of the app.

3.3.2.3 Reliability of the Tutorials

The tutorials in the app must be reliable and accurate. This means that there cannot be any grammar or spelling mistakes. The audio files for the listening exercises must be recorded by a native speaker of the language, and have the correct pronunciation. All translations within the system must be verified by language experts.

3.4 Recommended Task List

From the Functional Requirements specified in 3.3.1, a list of tasks is required that the app should provide. The analysis of extant systems in 2.1.2 in Chapter 2 is used as guidance for the recommended task list of the Molo app.

3.4.1 Login

1. Verify that user is registered

3.4.2 Create a user profile

1. Enter an user name
2. Enter an email address
3. Enter a password
4. Select a native language
5. Create the user's profile
6. Edit the information in a user's profile
7. Save the information in a user's profile

3.4.3 Do a tutorial

1. Populate different types of exercises with data from local database
2. Store the score a user achieved in a completed tutorial
3. Store the time it took the user to complete a tutorial
4. Save the user's progress

3.4.4 Synchronize the app

1. Select the option to synchronize the app
2. Update local database
3. Download audio files and images referenced by the local database

3.4.5 View Scoreboard

1. Select the option to view the Scoreboard
2. Show the user's results on the screen
3. Allow user to filter the results

3.4.6 Logout

1. Leave the app

3.5 Database Design

The Molo app will require a local database, as well as a remote database. The local database will store the data out of which the tutorials consist, as well as the user's profile and progress. The purpose of the remote database is to make the app extensible- if the developer wants to add/modify a tutorial, he can make changes to the remote database. The user will have the option to synchronize the local database with the remote database.

3.5.1 Local Database Design

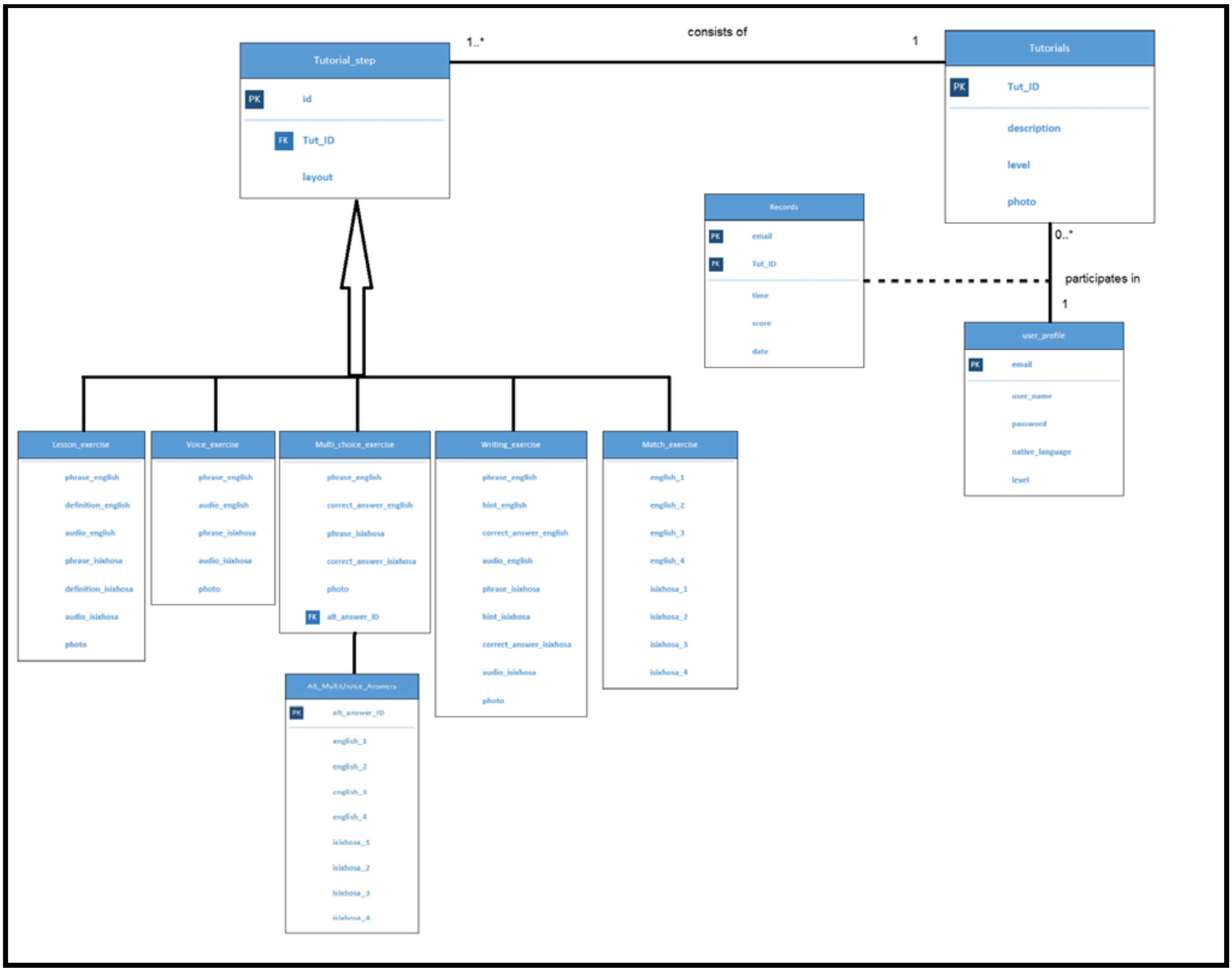


Figure 3-2 - Local database design

The local database will be created according to the design in Figure 3-2. The database will contain the following tables:

1. **user_profile**

   This class will store all the information of a user registered in the database, such as his user name, email address, and his native language. The email attribute is used as primary key in this class.

2. **Tutorials**

   This class contains a list of all the tutorials in the database, and uses the Tut ID attribute (the name of the tutorial, such as *Tutorial 1*) as primary key. Other attributes stored in this table is a short description of the tutorial (*Greetings* for Tutorial 1), and a photo.

3. **Records**

   An association class used to store information whenever a tutorial is successfully completed by a user. The attribute key is a composite key, a combination of the two foreign keys email and Tut_ID.

4. **Tutorial_step**

   The Tutorial_step class is a specialization hierarchy. A specialization hierarchy depicts and arrangement of parent entities and child entities (Coronel, Rob, & Crockett, 2009). In Figure 3-2, the table Tutorial_step is the parent entity, and the classes Voice_Exercise, Match_Exercise, Writing_Exercise, Lesson_Exercise, and Multi_choice_Exercise are its child entities. This means that each step in a tutorial can be a new lesson, voice exercise, multiple choice exercise, writing exercise, or matching exercise. Attributes in the parent entity are inherited by all of the child entities, but each child entity has attributes that is unique to that entity. The parent entity (Tutorial_step) uses the attribute id (a unique integer which increments automatically) as primary key, and the attribute layout as subtype discriminator. A subtype discriminator is the attribute in the parent entity that determines to which child entity it is related to (Coronel et al., 2009). The Alt_Multichoice_Answers class stores a variety of incorrect answers that goes alongside the one correct answer in the Multi_choice_Exercise class.

3.5.2 Remote Database Design

The remote database will be created according to the design shown in Figure 3-3. The remote database is an exact replica of local database, without the Tutorial, Records, and user_profile classes. The structure is so similar because the data used for the tutorials is static, meaning the user is never going to be able to use or change it in any way. The developer will manually update the data used for the tutorials in the remote database. When the user chooses to synchronize the app, the app will check if updates have been made to the remote database. If the remote database has changed since the app's last synchronization, it will simply copy the remote database and replace the relevant tables of the local database.

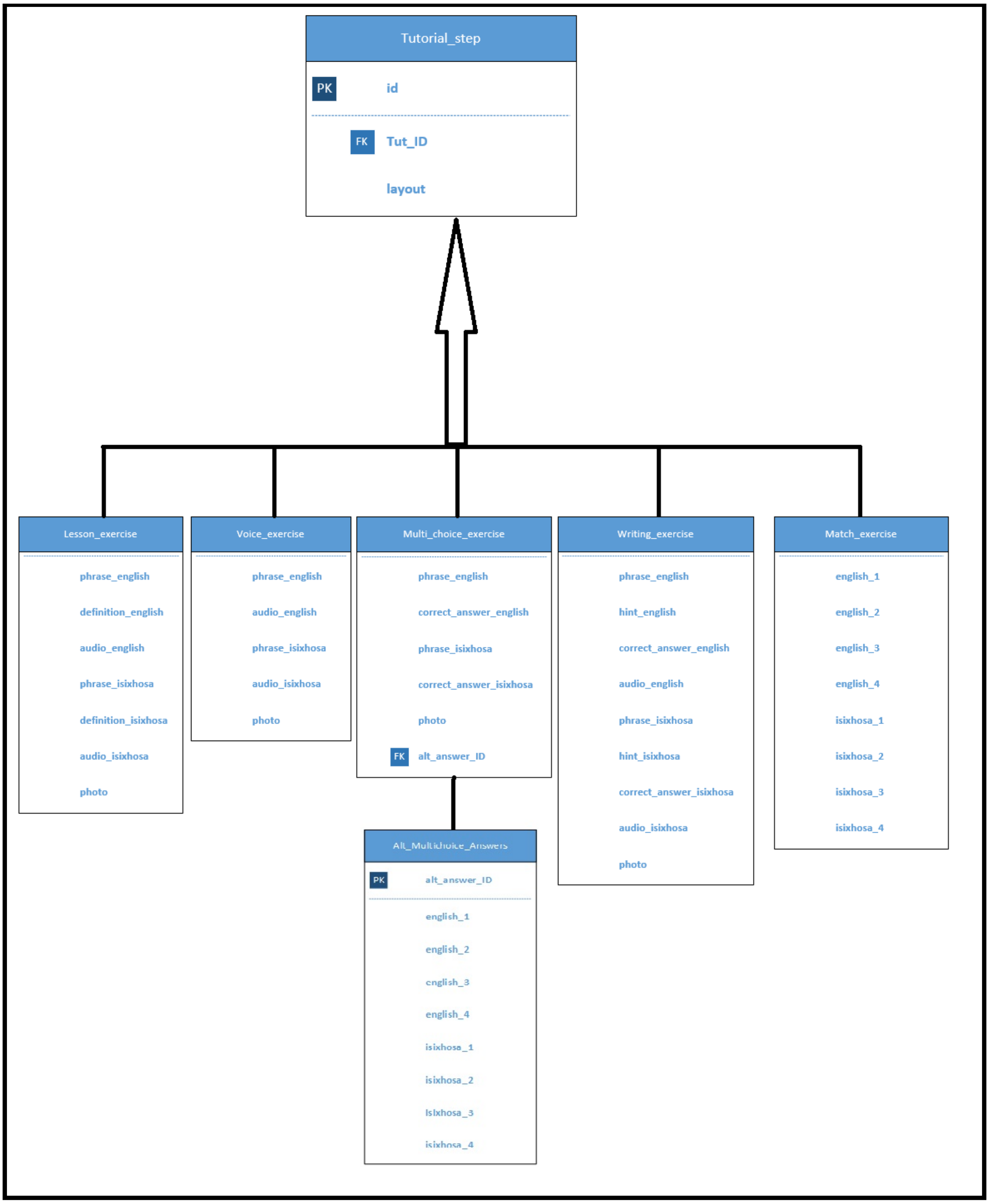


Figure 3-3 - Remote Database design

3.6 Screen Designs

The UI design that depicts the layout in which information is conveyed to the user is presented as a series of screen designs in Figure 3-4. Each screen design is explained in the following sections. All screen designs that is introduced in this chapter was done using Justinmind's prototyper (see Developmental Tools in Chapter 2), following the guidelines in Android Design Patterns (2014). It is important to note that these are just initial designs, and that the finished product may look different from the designs presented in this section.

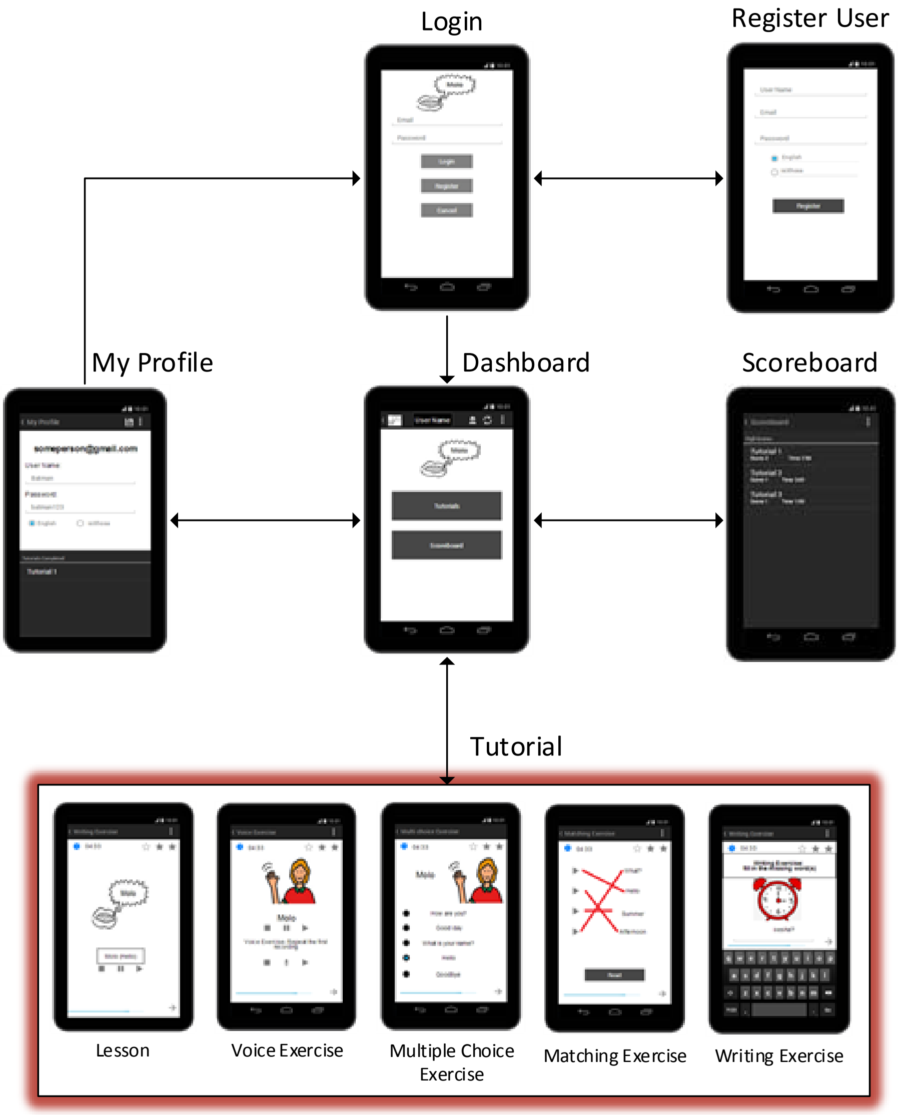


Figure 3-4 – UI design flow

3.6.1 Login & Register

The first screen that the user will see when the app is activated is the login screen illustrated in Figure 3-5. This screen shows the logo for app at the top of the screen, and three buttons.

From here the user can choose one of the following options:

1. *Login*, which will take the user to the Dashboard after verifying if the user's details are correct.

2. *Register,* which will take the user to next screen where he can register as a new user.
3. *Cancel*, to exit the application

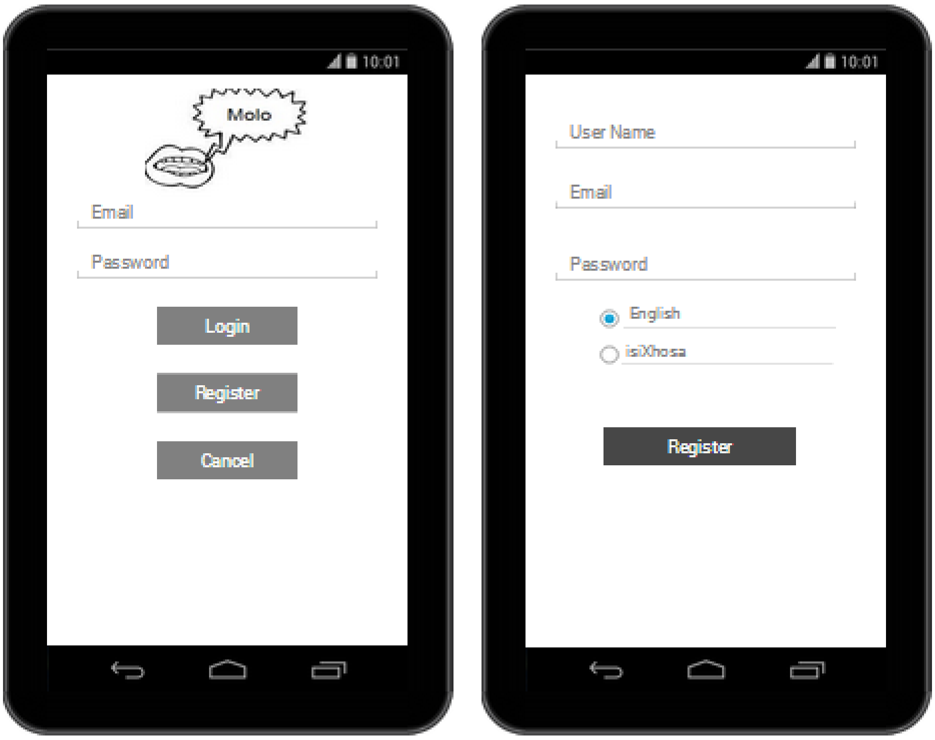


Figure 3-5 – Login (left) and Register (right) UI designs

This Register User screen in Figure 3-5 is accessed when the user presses the *Register* button on the Login screen. This screen allows the user to enter his/her information which is then saved in the app's local database.

The user must enter the following information to register successfully:

1. A *user name*. The user may pick any name. The user name will be used to display on the scoreboard.
2. An *email address*, which will be used to uniquely identify each user.
3. A *password*, which will be used to provide security for each user's profile.
4. The user must also choose his *native language* by selecting one of the provided radio buttons.

When a user has entered all the required information, a new profile will be created and added to the app's local database. From here the user is taken back to the Login screen, where he can now login as a registered user.

3.6.2 Dashboard

The dashboard, illustrated in Figure 3-6, displays the app logo in the middle top part of the screen. The user can navigate to anywhere in the app from the dashboard, but the main options are:

1. *Tutorials*, which will take the user to one of the tutorials that he could participate in. When selecting a tutorial, the user will be taken through a series of sequential exercises.
2. *Scoreboard*, which will take the user to a screen where he can view his previous results, as in Figure 3-13.

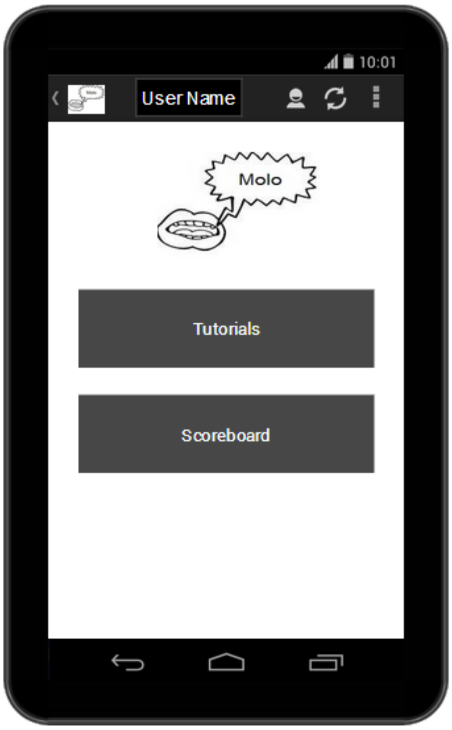


Figure 3-6 – Dashboard UI design

The Action Bar also plays an important role on the Dashboard UI:

1. The middle section will display the user's name currently logged in to the system
2. The user can select on the *profile* button to view his profile.
3. The user can select the *refresh* button to synchronize the Molo app.
4. The user can also select to *log out*, which will appear in the action bar's overflow menu.

3.6.3 Lesson Exercise

The Lesson Exercise, as in Figure 3-7, is where the user has the opportunity to learn new words and their meanings. An image with some connection to the new word is displayed at the top of the screen, while the foreign word and its meaning in the user's native language is displayed at the bottom. The user also has the option to listen to the new word's correct pronunciation by accessing the media player buttons.

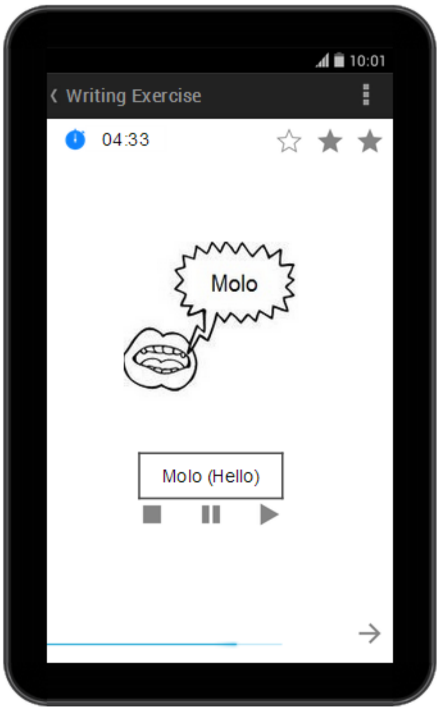


Figure 3-7 – Lesson Exercise UI design

The top of the screen displays a timer on the left, and three stars on the right. The countdown timer stores the time the user took to complete the entire tutorial. The user may repeat the same tutorial in the future to improve his initial time. The stars indicate how many questions the user had incorrect in a certain tutorial and functions as a lifeline- if the user runs out of stars before the tutorial ends he will have to start over. The countdown timer and stars features in all of the other types of exercises as well.

At the very bottom of the screen is a progress bar, which displays how far the user has progressed through the tutorial, and also a right-arrow button which the user can press to continue with the next exercise. The progress bar and the right-arrow also feature in all the other types of exercises.

3.6.4 Voice Exercise

The voice exercise illustrated in Figure 3-8 serves as a self-evaluation exercise. The user is given the spelling and graphical representation of a foreign word. The user will also have access to an audio file which he can repeat any number of times. The user can then record himself speaking the foreign word, and repeat the recording to himself to evaluate how good his pronunciation is.

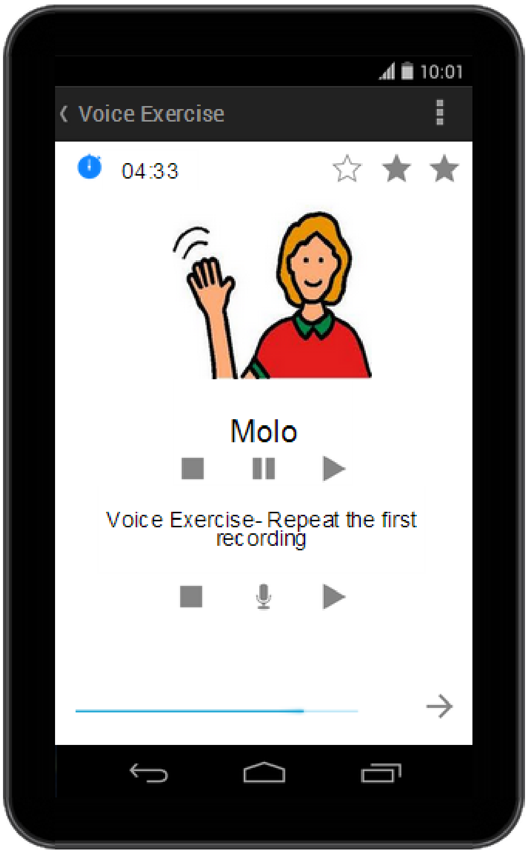
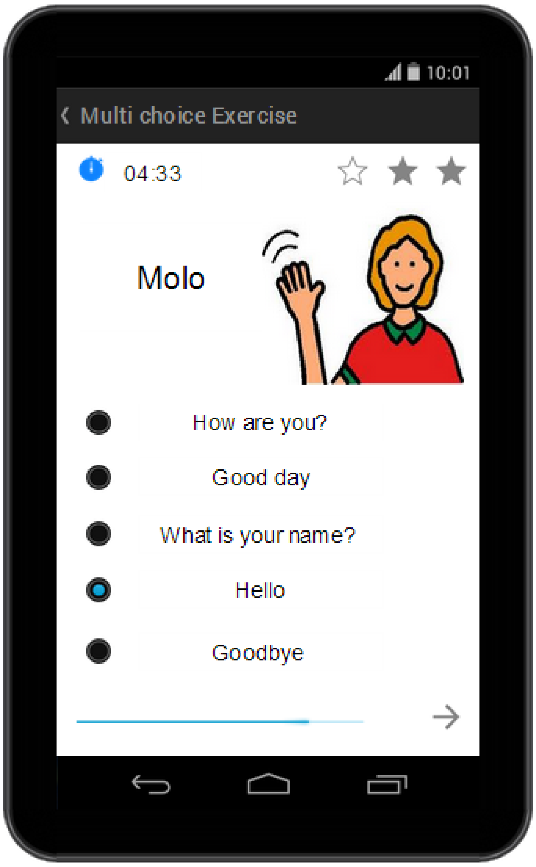


Figure 3-8 - Voice Exercise UI design

With this exercise the user will not lose any stars, and the countdown timer will be set to pause. This is done because it is a self-evaluating exercise, and repetition is a key component when learning the pronunciation of new words.

3.6.5 Multiple Choice Exercise

In the multiple choice exercise illustrated in Figure 3-9, the user is given hints, such as the word/phrase in the foreign language, and a graphical representation of the word/phrase.



Molo

● How are you?

● Good day

● What is your name?

◉ Hello

● Goodbye

Figure 3-9 - Multiple Choice Exercise UI design

The user must then select the correct answer using the radio button options in the bottom half of the screen.

3.6.6 Matching Exercise

In the matching exercise illustrated in Figure 3-10, the user is given a set of hints on the left side of the screen. He must then make associations between the hints by dragging (swipe) them to the correct answer on the right hand side.

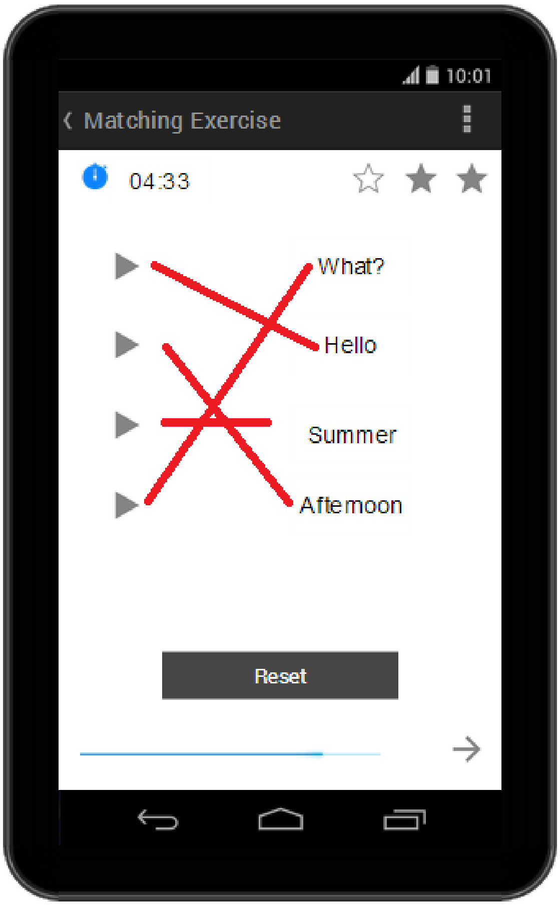


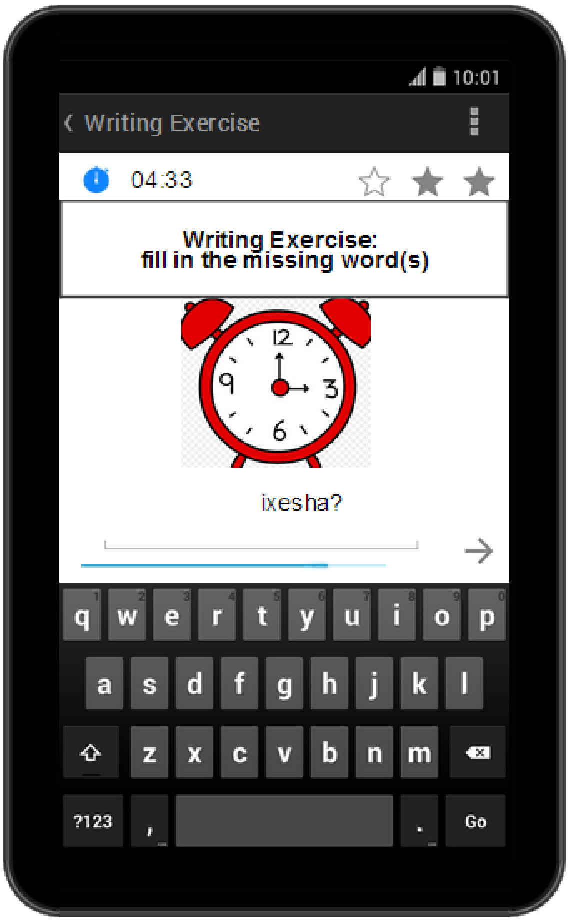
Figure 3-10 - Matching Exercise UI design

In the example the user can play one of four audio files (left side of the screen) in the foreign language, and then associate them with a word of his native language (right side of the screen). The hints can be either normal words/phrases, or more interesting things like audio files or images.

In the matching exercises, the user will also have the option to reset the screen to its initial state. The user has this option because he/she might realize they made a mistake, and want to alter their previous answers.

3.6.7 Writing Exercise

The last type of exercise is the writing exercise, as in Figure 3-11. In this exercise the user is given part of a sentence or asked a question in his native language.



10:01

Writing Exercise

04:33

Writing Exercise:
fill in the missing word(s)

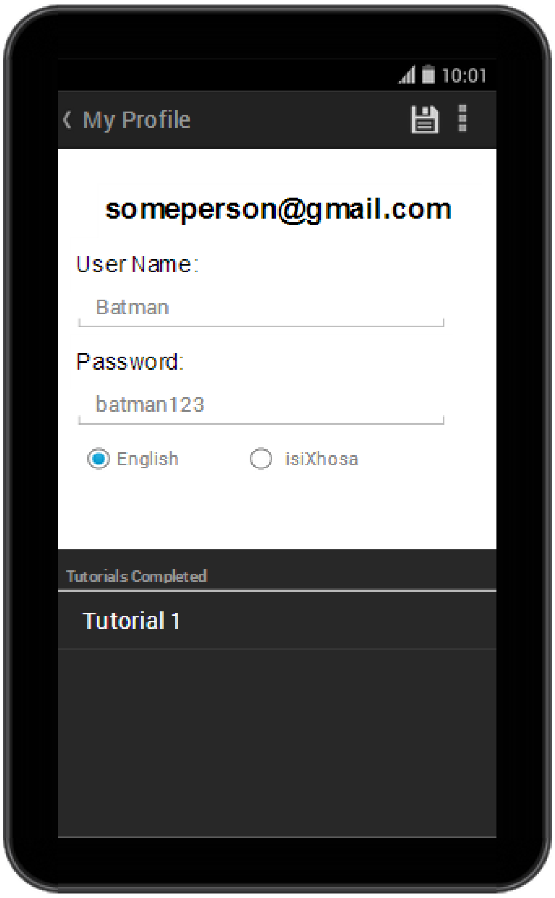
ixesha?

Figure 3-11 - Writing Exercise UI design

The user must complete the sentence/answer the question in the foreign language. An image on the screen is also given and serves as a clue to the correct answer. The focus of this exercise is to test the user's knowledge of the spelling of foreign words.

3.6.8 My Profile

The My Profile screen illustrated Figure 3-12 is accessed when the user wants to view his profile. The user’s email address is displayed at the top of the page. The user’s user name, password, and native language are displayed beneath the email address with a combination of edit texts and radio buttons.



someperson@gmail.com

User Name:
Batman

Password:
batman123

● English ○ isiXhosa

Tutorials Completed

Tutorial 1

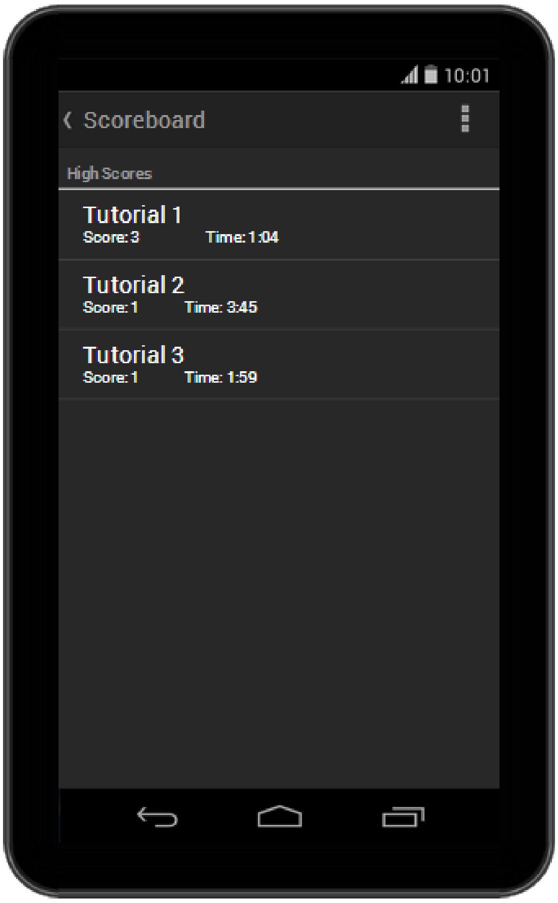
Figure 3-12 – My Profile UI design

The user can edit any of his details, with the exception of his email address (which is used as a primary key in user_profile, see Figure 3-2), by selecting the option to *save* in the action bar at the top of the screen. If the user selects *save*, his information will be updated in the local database and the user will be logged out of the Molo app. The user will then have to log in again using his new details.

The bottom of the screen displays a list of all the tutorials which the user was able to successfully complete. The user cannot edit information in this list. It serves as a way for the user to quickly track his progress without having to access the scoreboard.

3.6.9 Scoreboard

The scoreboard screen shown in Figure 3-13 displays the high scores for each tutorial.



High Scores

Tutorial 1
Score: 3 Time: 1:04

Tutorial 2
Score: 1 Time: 3:45

Tutorial 3
Score: 1 Time: 1:59

Figure 3-13 – Scoreboard

The information displayed by each item in the list is the tutorial name, the time it took the user to complete the tutorial, and the score he achieved. The user will also have the option of filtering the results, so that he can see all the results, only his results, and the results per tutorial.

3.7 Conclusion

The aim of this chapter was to design a prototype that meets the requirements specified in 3.3. This chapter included an analysis of the application domain, a use case diagram, an analysis of the functional and non-functional requirements, a recommended task list, initial design for the local and remote databases, and screen designs for the app.

The analysis of the application domain in 3.1 defines the target market for this app. The primary users will be anybody who wants to learn English or isiXhosa and has at least primary school understanding of one of these languages. User can also use this app to complement the formal learning material they receive from schools or universities.

The use case diagram in 3.2 identifies the main activities which the user and developer of the app can interact with. The functional in 3.3.1 and non-functional requirements in 3.3.2 are derived from the use case diagram. These requirements are specified in order to guide the implementation of the app. After the requirements were specified, a recommended task list was composed in 3.4. The task list was derived from a combination of the Requirements in 3.3, and analysis of Extant Systems in Chapter 2

The initial designs for the local and remote databases were discussed in 3.5. The database designs show how data will be stored on the mobile app and on the server side. The app allows for the user to synchronize the local database with data from the database.

Finally the screen designs serves as guidelines for the developer during the implementation phase. All screen designs were created keeping the guidelines presented in Android Design Patterns (2014) in mind. Each screen design shows how the different layouts satisfy the requirements specified in the sections preceding it.

The next chapter will discuss the implementation of various components of the app. The implementations will be discussed in terms of the requirements they meet and the problems faced during the implementation phase.

Chapter 4. Implementation

4.1 Overview

This chapter will focus on the implementation of the various aspects of the Molo mobile app. The implementation of the Molo still forms part of The Design Cycle of DSR. The various components were implemented using the RUP development methodology (IBM, 1998). This entails developing components of the Molo app incrementally. After each iteration a finished portion of functionality was completed.

The components in this chapter include registering as a user, the different exercises in a tutorial, the scoreboard feature, and the user-profile feature, how the app is synchronized, and the local and remote database of the app. This chapter will also explain the app's design so that it could be extended by future developers, who can either add more tutorials to the app, or add more languages alongside English and isiXhosa. The end of this chapter will give a brief overview of the challenges and problems faced during a particular part of the implementation phase, and the solutions the developer used to overcome these problems.

4.2 Register User

When a user registers himself on the Molo app, he has to enter the relevant information as in Figure 4-1.

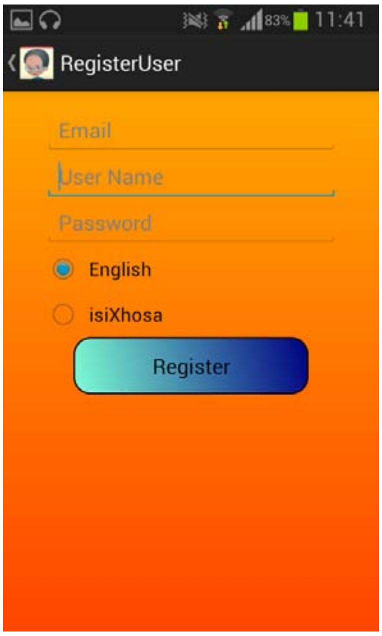


Figure 4-1 - Register User Screenshot

While normal *EditTexts* and *RadioButtons* (part of Android's widget toolbox) are used, several checks are performed to ensure the user's security and prevent users entering null values in the relevant fields.

- The user's email address must be at least 10 characters long and contain the '@' sign. The email address must also be unique because it is used as the primary key in the *user_profile* class in the local database, illustrated in Figure 3-2 in Chapter 3.
- A user's name must be at least five characters long.
- In addition to the user's password being at least eight characters long, it must contain digits as well as alphabetical letters.

A *Toast* will appear at the bottom of the screen if a user has entered details into one of the *EditTexts* that does not adhere to the rules explained above. The *Toast* will explain what how the details must change for the user to register successfully.

4.3 The Tutorials

The tutorials in the Molo app can be seen as the core of this project. The user learns new words and phrases by seeing their correct spelling, listening to their proper pronunciation, and being tested by various types of exercises in the tutorial. The following subsections will describe the implementation of each of the types of exercises as well as the content of each exercise. The tutorials make use of reusable screens, which is populated with data from the local database. The data which populates each exercise depends on the type of exercise the user is currently busy with, and also what the user's native language is.

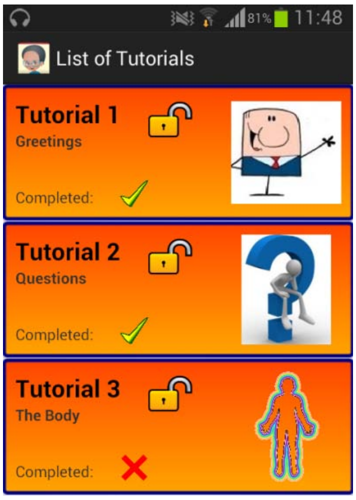


Figure 4-2 - List of tutorials

When the user navigates from the Dashboard to the Tutorials, he must first choose from a list of available tutorials as in Figure 4-2. This list uses a customized layout and Android's *ArrayAdapter* class (Vogel, 2014). Each item in the list displays the following information:

- The tutorial name, for example *Tutorial 1*
- A short description of the content of the tutorial, for example *Greetings*
- An image with some connection to the content of the tutorial on the right hand side of the item.
- An image that indicates if the user has access to a specific tutorial or not. A user must successfully complete the preceding tutorial(s) to gain access to the next one. A locked image indicates that the user does not yet have access to a tutorial, while an unlocked image indicates that he does have access.
- An image that indicates if a specific tutorial has been successfully completed. A red cross indicates that the user has not yet completed a tutorial, while a green checkmark indicates that the tutorial has already been completed at least once. The user is allowed to redo any tutorial to improve his previous score.

4.3.1 General Layout

All of the exercises in the tutorials have some generic features. These features allow the user to feel a sense of familiarity by using popular widgets. The *ActionBar* at the top will display whether the user is busy with a lesson or an exercise. Below the *ActionBar* are a *Chronometer* widget on the left, and a *RatingBar* widget on the right. The *Chronometer* displays how much time the user has spent on the tutorial, while the *RatingBar* indicates how many mistakes the user has made so far. The *RatingBar* acts as a lifeline to the user- once the user has made three mistakes the tutorial ends and the user will have to start over. The *RatingBar* is often used when the user has to rate a service or product of some kind (a movie rating for example); however the user cannot change the value of the *RatingBar* in this app.

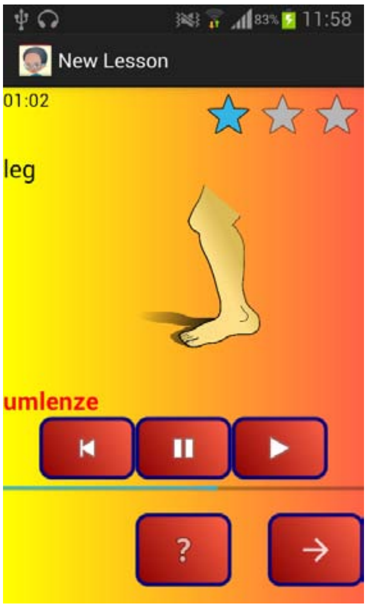


Figure 4-3 – New Lesson Screenshot

The bottom of the screen shows a *ProgressBar* above a *Help* and *Next* buttons. The *ProgressBar* indicates how through the tutorial the user has progressed. The Help button makes a *Dialogue* appear with a detailed explanation of how to complete a certain exercise. The *Next* button closes the current exercise and takes the user to the exercise in the series of exercises.

4.3.2 New Lesson

The first and simplest part of a tutorial is always a new lesson. From this 'exercise' the user must learn new words/phrases. In the latter stages of the tutorial the user's recall will be tested by other types of exercises.

The foreign word is displayed in bright red (*umlenze* in this example) at the bottom of the screen, as in Figure 4-3. The middle of the screen is dominated by an image with some connection to the new word. The meaning of the foreign word is displayed in black letters at the top of the screen in the user's native language (*leg* in this example).

The user can also listen to the correct pronunciation of the foreign word via media player controls at the bottom of the screen. Android's *MediaPlayer* class is called to manage the playback of the audio files. The audio files used by this exercise are pre-recorded in .wav format and stored in the Molo app's *assets* or *external downloads* directory.

4.3.3 Voice Exercises

The voice exercise serves as a self-evaluation exercise for the user. The term *self-evaluation* is used because the user is not formally evaluated or timed while doing this exercise- the user can decide for himself how many times and how long he wants to spend doing this part of the tutorial without it having any effect on his overall score and time for the tutorial.

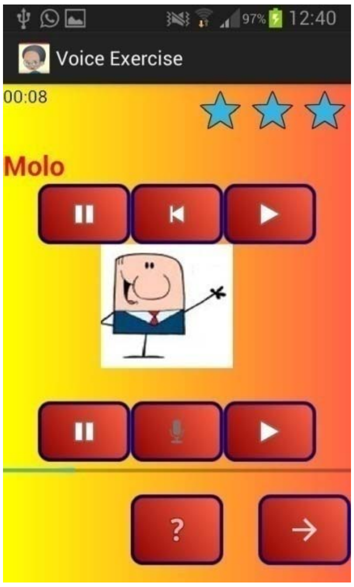


Figure 4-4 - Voice Exercise Screenshot

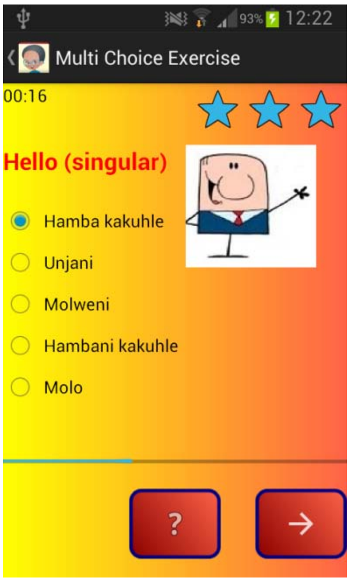
By accessing Android's *MediaPlayer* and *MediaRecorder* classes, the user can listen to the pronunciation of the foreign word and then try to replicate the correct pronunciation by recording and listening to his own pronunciation. The *MediaPlayer* for this exercise works precisely the same as for the New Lesson's in 4.3.1. The *MediaRecorder* functions by accessing the mobile device's microphone, setting the recording format to .mp3, and storing it in the devices *external storage* directory.

4.3.4 Multiple Choice Exercises

The multiple choice exercise is one of the exercises where the user must make an association and his recall is tested. The user is asked a question at the top of the screen in his native language and must then select one of five *RadioButtons* in the foreign language, as in Figure 4-5.

The five choices presented to the user consist of only one correct answer, along with four alternative incorrect answers. The alternative answers are similar to the correct answer to confuse the user and test his ability to recall the correct answer. The order of the texts alongside the *RadioButtons* is randomized to prevent users from always selecting an answer at a certain position. An image is displayed at the top right-hand side of the screen to aid the user in selecting the correct answer.

When the user presses the *forward* button, the *RadioButton* he selected is compared to the correct answer from the local database, and the user moves on to the next exercise.



Multi Choice Exercise

00:16

Hello (singular)

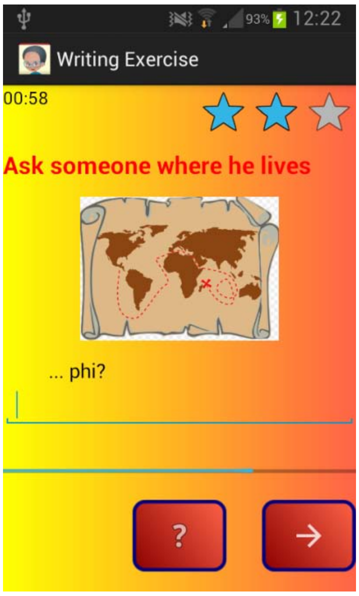
● Hamba kakuhle
○ Unjani
○ Molweni
○ Hambani kakuhle
○ Molo

Figure 4-5 - Multiple Choice Exercise Screenshot

4.3.5 Writing Exercises

The Writing Exercises tests two aspects, the user's ability to remember the correct answer, as well as the correct spelling of the answer. In the example shown in Figure 4-6, a question is asked in the user's native language at the top of the screen, and an image is placed below the question to aid the user to remember the correct answer. A hint is also given below the image to aid the user. The hint is part of the word/phrase that the user must spell correctly with some letters or words missing.

When the user has completed his answer and presses the *forward* button, the user's answer is checked. Both the user's answer and the correct answer are transformed to lowercase, and all punctuation is removed from both answers.


Writing Exercise

00:58

Ask someone where he lives

... phi?

Figure 4-6 - Writing Exercise Screenshot

In the example shown in Figure 4-6, the user is asked to *Ask someone where he lives* in isiXhosa. The correct answer to the example is *Uhlala phi?*, however the user can answer any one of the following and it will still be considered correct.

- uhlala phi?
- UHLALA PHI
- Uhlala Phi.

4.3.6 Matching Exercises

The Matching Exercise can be seen as an extension of the multiple choice exercise, but in this case the user has to make associations between four different pairs of answers, as shown in Figure 4-7. The implementation of the matching exercise was greatly help by Francis (2013)

The Matching Exercise attaches Android's *OnTouchListener* to objects, such as the red-colour texts in Figure 4-7, and uses the *setOnTouchListener* method. The *OnDragListener* is attached to the blue containers, and the *setOnDragListener* method is implemented to allow users to drag red-coloured texts to the appropriate container.

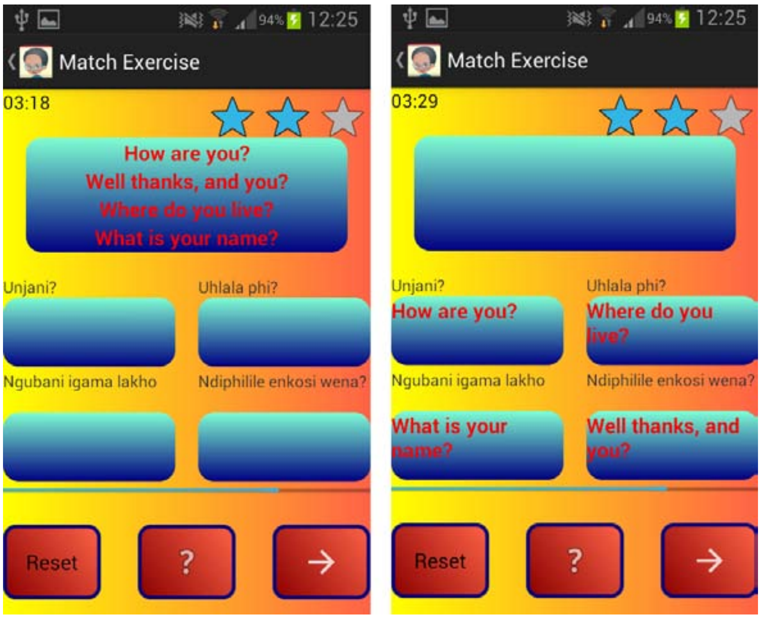


Figure 4-7 - Matching Exercise Screenshot

By using the *getChildCount* method, the user is not allowed to drop more than one text in a container, and no text is allowed to be dropped outside the containers (the orange parts of the screen). If a text is dropped somewhere illegally, it will simply jump back to its previous position.

The answers to this exercise is evaluated by implementing the *getChildAt(position)* method to determine which text's the user dropped in each container. All four containers must contain the correct answer for the user to be marked correct.

4.3.7 Content of the Tutorials

At the beginning of this project, the plan was to use data provided by the CSIR as the learning material in the tutorials. However, after an isiXhosa language expert reviewed the data it was decided to follow a different route. The content is split into 3 different categories which the language expert deemed the most important when learning a new language. The three categories are:

1. Greetings
2. Asking a simple question
3. The different parts of a human's body

In each category the user learns approximately 10 new words or phrases. The creation of these tutorials were also guided by *Learn Xhosa with Anne Munik* (Munik, 1994), which the language expert uses as material to teach isiXhosa to primary school learners.

4.3.8 Recordings

The recordings that are used in the Molo app were recorded with a mobile device. The initial plan was to make use of a recording studio, but it was found that the use of the mobile device was sufficed (provided that the recordings be made in a silent place), and that the recording's quality is clear enough. An honours student (Ms Masive Zita) from the Computing Sciences Department who speaks English as well as isiXhosa fluently was used to make the recordings.

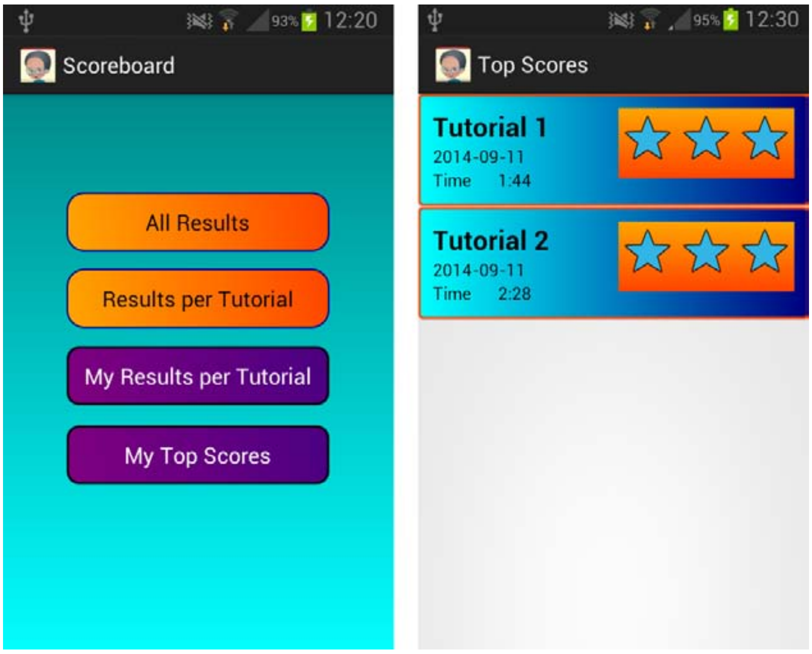
The recordings were edited using Audacity to ensure that they were loud and clear enough, and in .wav format. Audacity is a software program that enables the user to modify/enhance certain aspects of audio files. A more detailed explanation of Audacity is given in 2.4.6 in Chapter 2.

4.4 Scoreboard

The scoreboard of the Molo app allows the user to view results registered in the local database. As illustrated in Figure 4-8 the user has the option to filter the results in four different ways

1. All results - This option displays all of the results in the local database, sorted in ascending order by the user's email addresses
2. Results per tutorial – Shows all the results in the local database for a certain tutorial
3. My results per tutorial – Shows the results of the current user for a certain tutorial
4. My Top Scores – Shows the top scores of the current user for all the tutorials he/she has successfully completed.

The results are displayed by using SQL queries in the local database.



Tutorial 1
2014-09-11
Time 1:44

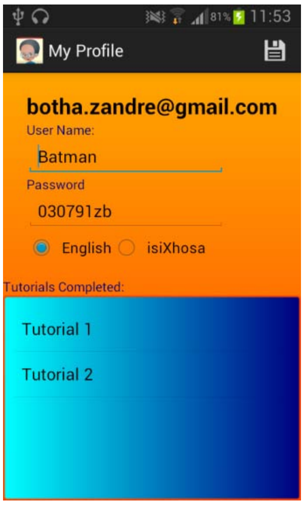
Tutorial 2
2014-09-11
Time 2:28

Figure 4-8 - Scoreboard Screenshots

The scores displayed are done using Android's *ArrayAdapter* and a customized layout the same way as in 4.3.

4.5 My Profile

The user profile feature allows the user to view and edit his details in the local database, as well as track his progress through the tutorials in the app. The top half of the screen is populated with the user's email, user name, password, and native language, by using SQL queries in the user_profile class in Figure 3-2 in Chapter 3. The user has the option of modifying any of his information, with the exception of the email address.



botha.zandre@gmail.com

User Name:
Batman

Password
030791zb

● English ○ isiXhosa

Tutorials Completed:

Tutorial 1

Tutorial 2

Figure 4-9 - Screenshot of the user's profile

The bottom half of the screen displays a list of all the tutorials that the user has successfully completed. The list is created using Android's *ListView* widget, and populated by SQL queries in the Records class illustrated in Figure 3-2 in Chapter 3.

4.6 Local Database

The local database was created according to the design shown in Figure 3-2 in Chapter 3. Using Data Browser for SQLite, fully explained in 2.4.4.1 in Chapter 2, different classes and its attributes can be created easily enough along with its constraints. The populated database was then placed in the Molo app's *assets* directory. Upon installation, the app checks whether a local database has been created. If a local database is already in existence nothing happens, otherwise a local database is created from the file in the *assets* directory by using the *SQLiteOpenHelper* class in Android.

4.7 Remote Database

The remote database was created according to the design shown in Figure 3-3 in Chapter 3. MySQL Workbench, fully explained in 2.4.5 in Chapter 2, was used to populate the remote database. The database was populated manually using SQL *insert* commands. Special care was taken when the data was being entered to use the correct spelling as a lot of the fields in the database contained textual references to audio or image files.

4.8 Synchronization

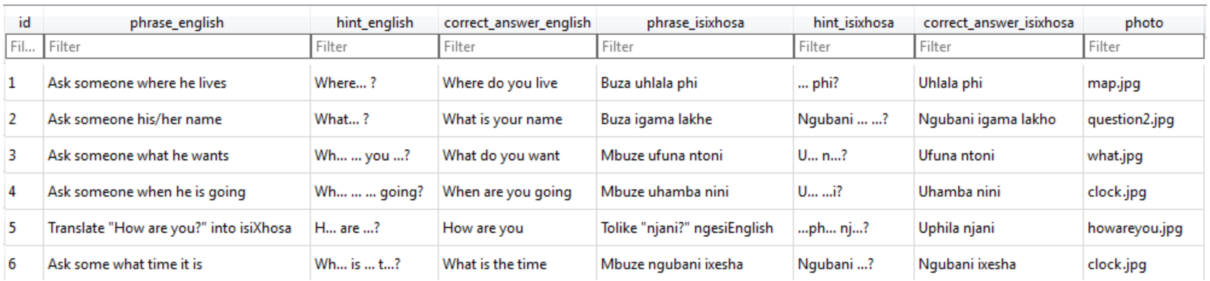
The synchronization of the app is done by using PHP scripts to query the remote database. When the user selects the option to synchronize the Molo app, Android's *AsyncTask* class is called, which creates a new thread that runs in the background.

Firstly, the PHP scripts are called to update all of the tables of the local database from the remote database. Because the database contains only text and integer values, this step of the synchronizations is completed very quickly.

Once the local database is up to date, SQL queries are combined with Android's *getAssets()* and *getExternalFilesDir()* methods to determine which filenames exist in the local database, but not somewhere on the device. Using the *DownloadManager* class, the missing audio and image files are then downloaded from the server into the Molo app's external downloads directory.

4.9 The Generic Design

As explained in 4.3, all of the information used by the tutorials is generated from the local database. The database was designed and implemented in such a way that it could be populated by any two languages, and the app will still work perfectly. Figure 4-10 illustrates some of the data used for the writing exercises in the app. The developer could populate this table and others easily enough with any two languages as he sees fit.



id	phrase_english	hint_english	correct_answer_english	phrase_isixhosa	hint_isixhosa	correct_answer_isixhosa	photo
Fil...	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Ask someone where he lives	Where... ?	Where do you live	Buza uhlala phi	... phi?	Uhlala phi	map.jpg
2	Ask someone his/her name	What... ?	What is your name	Buza igama lakhe	Ngubani ... ...?	Ngubani igama lakho	question2.jpg
3	Ask someone what he wants	Wh... ... you ...?	What do you want	Mbuze ufuna ntoni	U... n...?	Ufuna ntoni	what.jpg
4	Ask someone when he is going	Wh... ... ... going?	When are you going	Mbuze uhamba nini	U... ...i?	Uhamba nini	clock.jpg
5	Translate "How are you?" into isiXhosa	H... are ...?	How are you	Tolike "njani?" ngesiEnglish	...ph... nj...?	Uphila njani	howareyou.jpg
6	Ask some what time it is	Wh... is ... t...?	What is the time	Mbuze ngubani ixesha	Ngubani ...?	Ngubani ixesha	clock.jpg

Figure 4-10 - Data in the app's local database

This treatise does not include guidelines on how to extend the Molo app so that more languages could be added to it. However by using the design of the local database as a guide, more columns can be added which contains data for the new language. After the database design is updated, it is simply a case of changing some of the parameters in app. For example, when a user registers in 4.2 he can only choose between English and isiXhosa as his home language. Extending the app will allow the user to choose from more than just two languages as his home language. When the user goes to the tutorials he will learn the alternative language of what he chose as his home language when registering, i.e. if the user's selected English as his native language he will now learn isiXhosa. Extending the app must allow the user to choose from more than one language to complete the tutorials in, the user's native language excluded.

4.10 Problems experienced during implementation

This section explores some of the problems encountered during implementation phase, as well the solutions to each particular problem.

4.10.1 Screen Space

One of the major problems experienced during implementation was the lack of screen space on devices with smaller screens. Especially during the implementation of the Multiple Choice in Figure 4-5 and Matching Exercises in Figure 4-7, where various different information and widgets need to be displayed, it was hard to find ways to make the most out of the available screen space.

4.10.2 Changing the Matching Exercise

Initially in 3.6.6 in Chapter 3, the plan was to implement the Matching Exercise so that the user must make associations by dragging different items on the left side of the screen to matching items on the right side of the screen. The idea was that items could be texts, images, or even audio files. However, during implementation it became clear that the scope for such an exercise is too big and would consume too much time. The developer resolved to rather only match text items in the exercise, and instead of dragging items to each other, the user must now drag and drop items in labelled containers.

4.10.3 Unavailable Language Expert

During implementation the developer consulted two isiXhosa language experts to help decide what content should be in the tutorials of the app. Of the two isiXhosa language experts consulted, one was often unavailable and hard to reach. Several interviews had to be rescheduled which resulting in a lot of time being wasted. Fortunately Ms Lulami Mani, an isiXhosa teacher at Collegiate Primary School, was very helpful so that the developer could depend on her for guidance and consultation.

4.10.4 Database Design Changing Frequently

The initial database design was somewhat flawed due to the lack of experience in database design and implementation by the developer. These flaws were only noticed and corrected during the implementation, resulting in the final database design being quite different from the initial version.

4.10.5 Manually Populating the Database

Another main issue is that there does not exist a desktop application to create the tutorials for the Molo app- the tutorials consist of data that had to be entered into the database manually using SQL commands. This means that any spelling mistake will appear in the Molo app’s tutorials as well. Special care needs to be taken when entering fields which references file names in the app (such as audio or image names), because spelling mistakes or even unused white spaces will result in the relevant file not being found.

4.11 Conclusion

This chapter aimed to discuss how the various components of the Molo app were implemented to satisfy the Functional Requirements in Chapter 3. The problems encountered during the implementation phase along with their solutions were also discussed.

Functional Requirement 1 was met by functionalities provided by Register User in 4.2 and My Profile in 4.5. This requirement is met by using the app's local database and SQL commands.

Functional Requirement 2 was met by The Tutorials in 4.3. The Tutorials included new lessons, voice; multiple choice; writing; matching exercises, which satisfy Functional Requirements 2.a - 2.i. These exercises were implemented by using a wide range of Android widgets, and populating the screens with exercises from the local database.

Functional Requirements 3 and 5 were met by the Scoreboard in 4.4 and My Profile in 4.5. These features were implemented by adding data to the app's local database when a user manages to successfully complete a tutorial. The user can view his progress from either the Scoreboard or the My Profile feature. The app also makes use of different levels- the user must first complete a preceding tutorial before being granted access to the next tutorial in the series.

Functional Requirement 4 was met by the Synchronization feature in 4.8. The synchronization of the Molo app was implemented using PHP scripts to update the relevant tables of the local database with data from the remote database. Audio files and images were downloaded using Android's *DownloadManager* class.

The next chapter will discuss the evaluation of the Molo prototype. The evaluation will aim to determine if the Non-Functional Requirements specified in Chapter 3 are met successfully.

Chapter 5. Evaluation

5.1 Overview

In this chapter the Molo app is evaluated to determine if it satisfies the requirements in Chapter 3. The evaluation of the app falls under The Relevance Cycle of DSR, which aims to determine if the success criteria for the system was met and tries to identify areas which can still be improved.

A usability study is conducted to determine if the requirements are met satisfactorily. Ultimately the goal of the usability study is to optimize the user experience of the Molo app. The results gathered during the usability study is analysed and discussed, resulting in recommended changes to the Molo app.

5.2 Usability Study

The usability study aimed to measure the Molo app's satisfaction and learnability. These two metrics are used to determine if the software artefact can be viewed as successful or unsuccessful.

The participants were asked to fill in a pre-test questionnaire (Appendix B – Pre-test questionnaire) to determine how much isiXhosa words they knew before using the app. Participants could only be selected from the NMMU's Computing Science Department. The evaluator did not determine if the isiXhosa speaking participants know any English words before testing because the Computing Science Department is English. The participants were then asked to complete several tasks in the Molo app (Appendix C – Task List). Once the participants completed all the tasks they were asked to fill in a post-test questionnaire (Appendix D – Post-test Questionnaire). The post-test questionnaire is composed of the following sections:

1. **Biographical Questions** - Personal information about the participant such as his/her age and gender are gathered without the participant having to identify him/her.
2. **Post-Test Questionnaire** - This section of the usability study measures the overall satisfaction of the Molo app, and is based upon the Computer System Usability Questionnaire (CUSQ) presented in William & Tullis (2013). The questionnaire is modified slightly to suit the Molo app. All questions are positively worded and the participant must rate his agreement on a 5-point Likert scale. The CUSQ

questionnaire is used because it was designed to be used when a usability study is not conducted in a usability lab (William & Thomas, 2013).

3. **Task Load Index Questionnaire** - This section of the questionnaire tests how difficult it was for the participant to complete the tasks, mentally and physically. Hart & Staveland's (1988) NASA-TLX questionnaire is modified to suit a 5-point Likert scale, where the participant must rate how demanding a certain aspect of the Molo app was.
4. **Comments** - This section gives the participant an opportunity to identify which aspects of the app he liked/disliked the most, and to suggest some changes that could improve the user experience. The participant also states if he/she learned any new vocabulary from the foreign language.

5.2.1 Biographical Information of Participants

The participants used for the usability study have at least a primary school understanding of English or isiXhosa. The Molo app runs on the Android platform, however as anyone with a basic understanding of English/isiXhosa could use the app it was not a prerequisite for a participant to own an Android device.

A total of 15 students/staff members from the MMU Computing Sciences Department took part in the usability study. Figure 5-1 illustrates the biographical information of the participants. The population of participants has a fairly even distribution between male and female users, while most of the participants has the same level of education and are roughly the same age. The participants were selected so have an equal distribution between their home languages- 33% were English speakers, 27% were isiXhosa speakers, while 40% of the population had another home language (divided between Afrikaans, Shona, and Setswana). Figure 5-1 also shows whether the users owned Android devices, and the level of experience they have with Android touch screen devices.

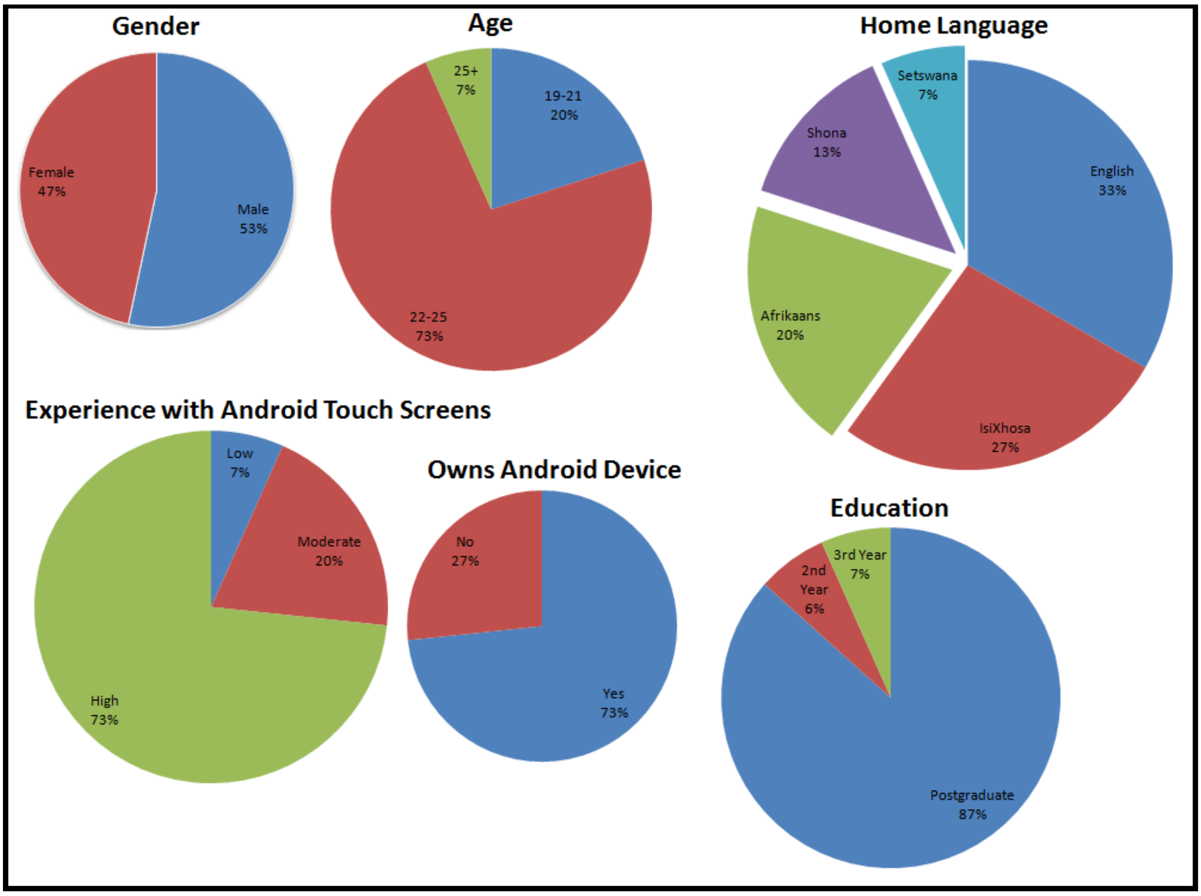


Figure 5-1- Biographical Information of participants in 1st evaluation (n = 15)

5.2.2 Environment and Equipment

The usability study was conducted in the Korbitec foyer of the NMMU Computing Sciences Department. This enabled the participant to complete the questionnaires and task list in a relaxed environment. The evaluator began the study by giving a brief description of the questionnaires and tasks. The evaluator then moved away to give the participant the freedom to answer the questions honestly and make objective suggestions. The evaluator was still close enough to provide the participant quick assistance, but not close enough so that the participant would feel the evaluator was hovering over him/her.

The Molo app was installed on a Samsung SIII mini mobile device, which met the hardware requirements of the app. To emphasise the quality of the voice recordings a Philips headphone set was used. This also provided the participant with a means to listen to the voice recordings without possible noise disturbances in the vicinity. The headphone set also included a microphone for recording purposes.

5.2.3 The Tasks

Each participant was instructed to complete a list of tasks, as in Appendix C – Task List. The tasks were designed to test the main functionality of the app, using the use cases in Chapter 3, Use Case Diagram, as guidelines. A brief description of each task follows below.

Task 1- Register as a new user

Task 2- Login to Molo

Task 3- Find the Tutorials

Task 4- Complete Tutorial 1 and Tutorial 2

Task 5-

a. Go to the Scoreboard. Check if the results can be filtered effectively.
b. Verify that your scores are displayed correctly.

Task 6- Synchronize Molo

Task 7- View your profile

Task 8- Modify some of your profile details and save them

5.2.4 Results

This section gives the results of the usability study. Because learnability and satisfaction is measured the results are in a qualitative form. The results were obtained from the task list and usability questionnaire, and are visually represented here in charts. All raw data collected in the usability study can be seen in Appendix E – Raw Data from Evaluation 1.

5.2.4.1 Task Completion Rate

When the participants were finished with the task list, a frequency list could be made from all the successfully complete tasks. Figure 5-2 illustrates the completion rate for each task. Considering the nature of tasks 1, 2, 3, and 7, and that all of the participants could complete them successfully, they are not mentioned in the discussion below.

Completing tutorial 1 & 2 (Task 4) is shown to have been the most difficult task for participants to complete. However, taking into account the nature of the task, and that some user's don't know any isiXhosa vocabulary at all, this should not be seen as a system or usability problem.

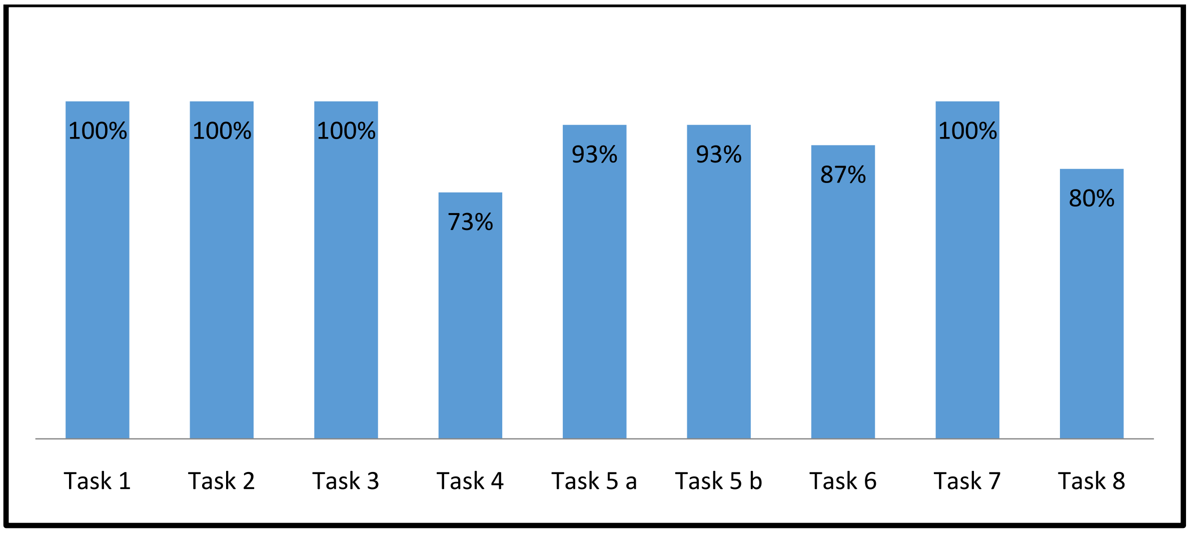


Figure 5-2 - Completion Rate per task during 1st evaluation (n = 15)

Filter the scoreboard (Task 5 a) and verifying that the correct results are displayed (Task 5 b) both scored a completion rate of 93%. The feedback section of Task 5 showed that participants would like to see on the scoreboard where they got answers wrong in the relevant tutorials.

Synchronizing Molo (Task 6) shows a completion rate of 87%, the other 13% not having filled in the answer on the task list. 47% of the participants did not fill in the correct answer on the task list. To successfully complete Task 6, a user has to wait a sufficient amount of time for the local database to be updated and all images and audio files to be downloaded completely. The high error rate could possibly be attributed to participants not following the instructions correctly and going through the task too quick.

5.2.4.2 Post-Test Questionnaire

Section B of the use study questionnaire determines the participant's opinion of the app's usefulness, information quality, and interface quality. It then combines three aforementioned aspects to calculate the participants overall satisfaction of the app. Figure 5-3 illustrates the mean results of each aspect.

From the graph illustrated in Figure 5-3, the reader can conclude that participants thought that the app was useful, and had a well-designed interface. The overall satisfaction of the participants also receives a reasonably high score. However, participants felt that the information or instructions provided by the system could improve and be on par with the other aspects of the system.

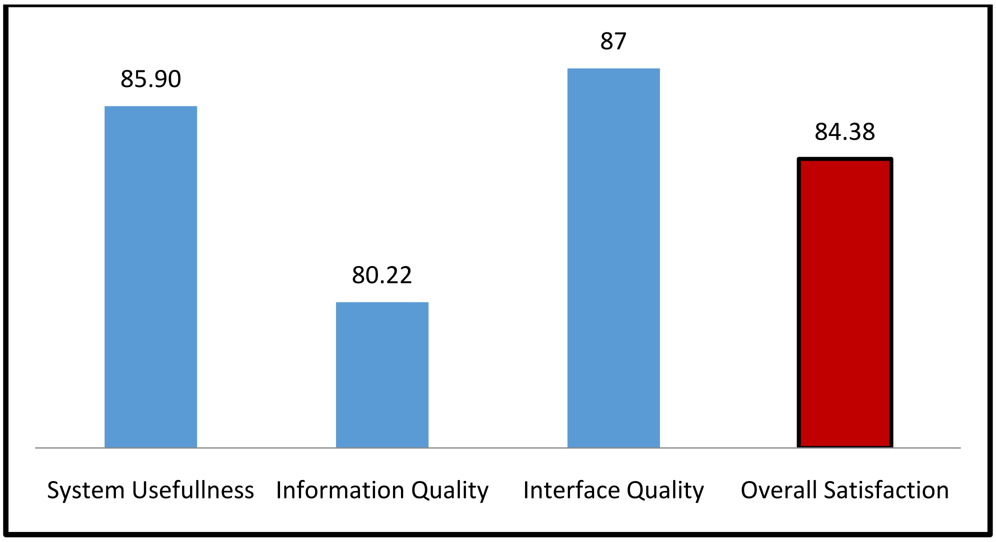


Figure 5-3 - Quantitative Results of Post-Test Questionnaire during 1st evaluation (n = 15)

5.2.4.3 Task Load Index Questionnaire

Section C of the usability study questionnaire determines how difficult the Molo app was to use, mentally and physically. Figure 5-4 illustrates the quantitative results of this section. The graph presents the mean of each question, where 1 represents the lowest possible demand and 5 the highest possible demand.

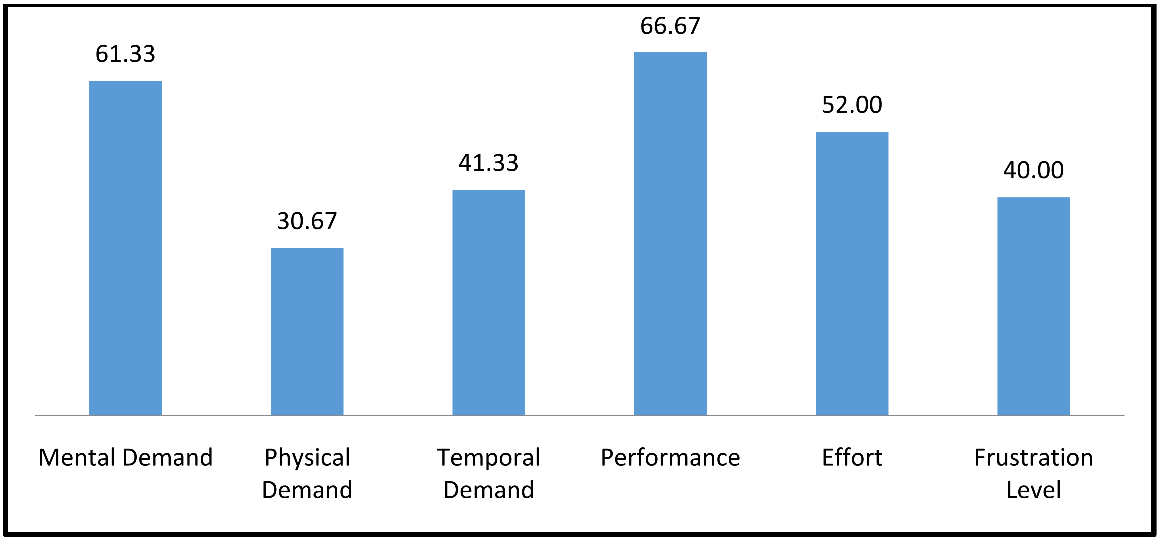


Figure 5-4 - Quantitative Results of Task Load Index Questionnaire during 1st evaluation (n = 15)

At first glance, the *Mental Demand*, *Effort*, and *Frustration Level* may seem high enough to cause concern. However the reader should keep in mind that this is an e-learning app where participants were taught new words entirely foreign to them, and being tested only minutes later on what they have learned. The high scores in these three categories should therefore not be viewed as problems with the app's interface or design, but it rather gives indication of how challenging it was for participants to learn new words/phrases.

The *Performance* bar indicates that participants feel they were reasonably successful in completing most of the tasks. Again the reader must keep in mind that one of the tasks was to complete Tutorial 1 and Tutorial 2 in the app, which means learning and being tested on up to 20 new words/phrases. This may have had a severe impact on the *Performance* bar because only 73% of the participants could complete the task. The *Physical* and *Temporal Demand* is low compared to the other results, as to be expected.

5.2.4.4 Participant Feedback

This section provides the participant's feedback of the app. The data gathered came from Section D of the usability study questionnaire, as well as suggestions that were written as answers on the task list.

Figure 5-5 reveals that most participants liked the fact that the app is easy to use. The second highest most-liked aspect is the audio feature of the app.

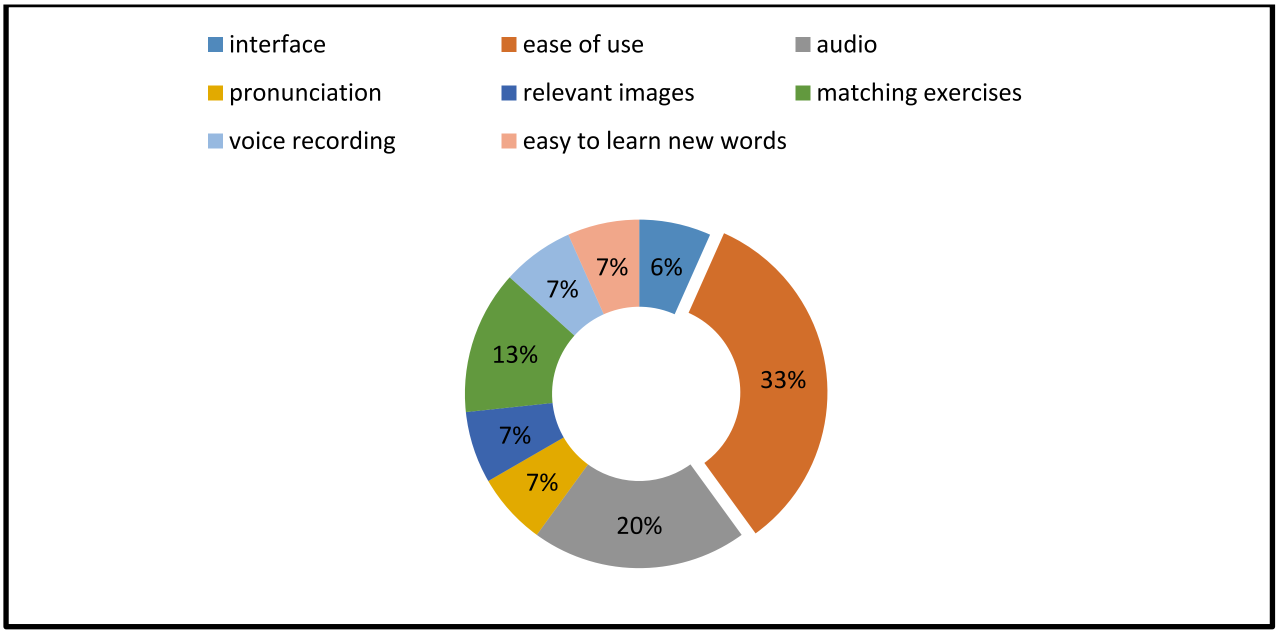


Figure 5-5 - Most Positive Aspects of the Molo app during 1st evaluation (n = 15)

Figure 5-6 reveals that most participants did not find any negative aspects with the app. However, 13% of the participants felt that the instructions/information provided by the app is too vague.

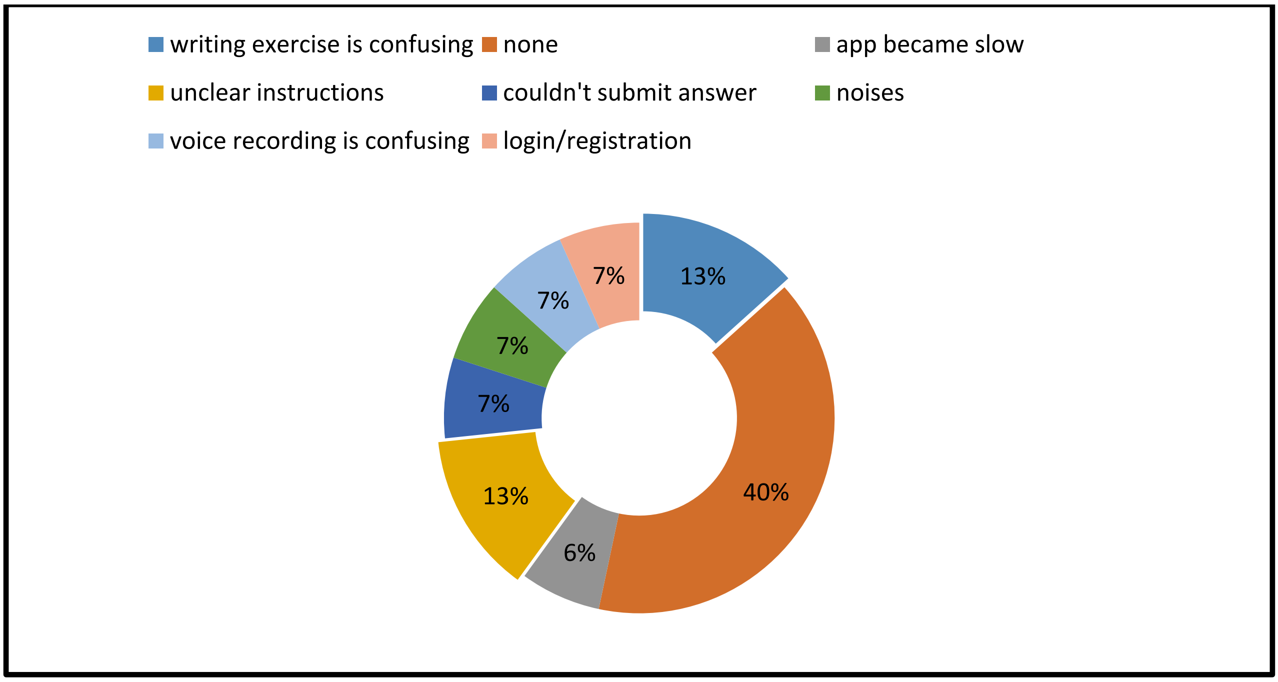


Figure 5-6 - Most Negative Aspects of the Molo app during 1st evaluation (n = 15)

Figure 5-7 illustrates all of the participants who felt they learned new words. The four isiXhosa speaking participants are excluded from this graph because it would be irrelevant to ask them if they learned any new English words. 82% of the participants said they did learn new foreign words.

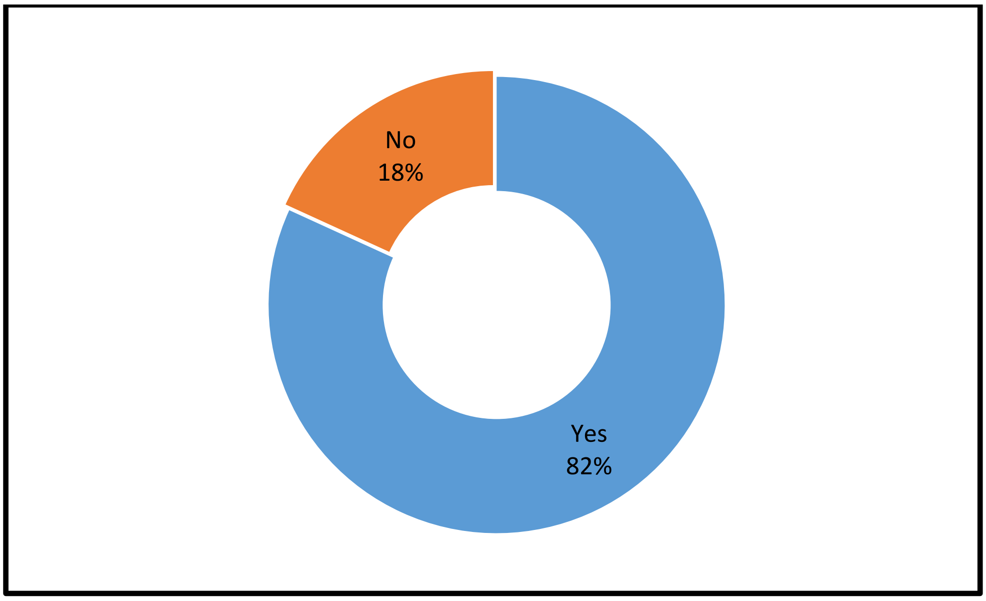


Figure 5-7 - Relevant Participants who learned new words during 1st evaluation (n = 15)

The participants also had the opportunity to suggest improvements/changes to the app. The most frequently suggested changes are listed below:

- Make the voice recording exercise simpler

- Make the instructions to complete each exercise more apparent
- Display which answers users got wrong

5.2.5 Recommended Changes

The recommended changes to the Molo app originates from suggestions listed by participants, what participants listed as the most negative aspect of the app, and conclusions drawn from the task completion rate. Table 5-1 shows what changes are to be made to improve the user experience for the Molo app. The table lists the change to be made, a justification/reason for the change, and how vital the change is to improve the user experience.

	Change	**Justification**	**Severity**
1.	Instructions as to how to complete different exercises should be made clearer.	The low score of *Information Quality* in Figure 5-3, *negative aspects* listed in Figure 5-6, and *suggested improvements* listed in 5.2.4.4	High
2.	The voice pronunciation exercise layout should be simpler.	The *negative aspects* listed in Figure 5-6 as well as *suggested improvements* listed in 5.2.4.4	High
3.	The scoreboard should be easier to filter.	The *feedback* participants wrote on Task 5 in Appendix C – Task List, as well as observations made by participants to the evaluator after the usability study was complete.	Medium
4.	The scoreboard should display the user's incorrect answers.	The *feedback* participants wrote on task 5 in Appendix C – Task List, as well as *suggested improvements* listed in 5.2.4.4	Medium
5.	Move the *next* and *help* button up to the action bar.	This problem was identified in *Chapter 4- Screen Space*. Moving these two buttons to the action bar will free up valuable screen space and allow other important items to be displayed larger.	Medium

6.	The app should indicate when it is starting to synchronize, and when the synchronization process has completed.	The *low completion rate* of Task 6 in Figure 5-2.	Medium
7.	Allow user to turn sound on/off when the answer is marked.	The *negative aspects* listed in Figure 5-6 as well as *suggested improvements* listed in Appendix D – Post-test Questionnaire	Low
8.	User should be able to see what he typed in to the password category when registering/logging to Molo.	The *negative aspects* listed in Figure 5-6 as well as *suggested improvements* listed in Appendix D – Post-test Questionnaire	Low

5.3 Second Evaluation

After the first evaluation, changes were made to the app to determine if the results obtained from the first evaluation could be improved upon. All eight recommended changed listed in Table 5-1 could be made successfully. After the changes were made to the app a second usability study was conducted.

For the second evaluation, the same pre-test questionnaire (Appendix B – Pre-test questionnaire), task list (Appendix C – Task List), and post-test questionnaire (Appendix D – Post-test Questionnaire) was used as in the first evaluation.

5.3.1 Biographical Information of Participants

Eight participants provided feedback for the second evaluation. Participants of the second evaluation were new to the Molo app, meaning they have not seen it before and were not part of the first evaluation. Biographical information of the participants could be seen in Appendix F – Raw Data from Evaluation 2.

The results of the second evaluation can be compared to the first evaluation because the participants' biographical information is relatively similar. The gender, age, experience with Android touch screens, and participants owning Android devices, of participants in the first and second evaluation look very much alike. The home languages of participants in the second evaluation are more evenly distributed, while less postgraduate students participated in the second evaluation.

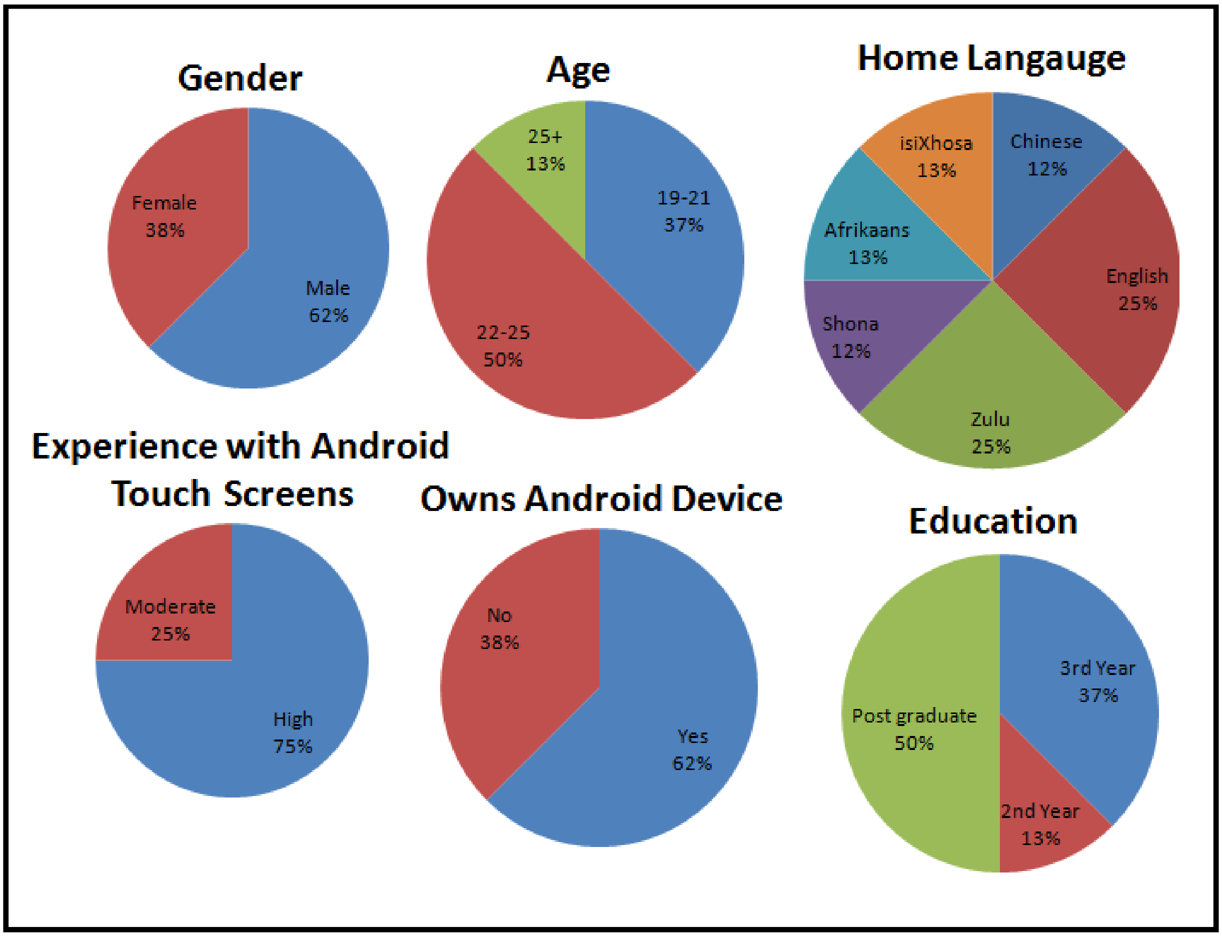


Figure 5-8 - Biographical Information of participants in 2nd evaluation (n = 8)

5.3.2 Comparison of Results

The results obtained from the second evaluation could be compared to the results in 5.2.4. The following subsection gives feedback on the results obtained during the second evaluation, while also making the relevant comparisons to results obtained from the first evaluation.

5.3.2.1 Task Completion Rate

Figure 5-9 compares the task completion rate of the first and second evaluation. Task 1, 2, 3, and 7, is again left out of the discussion due to the simplicity of and high completion rate of the tasks respectively.

The completion of Tutorial 1 & Tutorial 2 (Task 4) rose from 73% during the first evaluation to 88% during the second evaluation. This improvement can be attributed to making the instruction per exercise clearer and also improving the layout of the voice exercises, recommended change 1 and 2 respectively in Table 5-1. Only one participant failed to complete Task 4, but bearing in mind the difficult nature of the task this is to be expected.

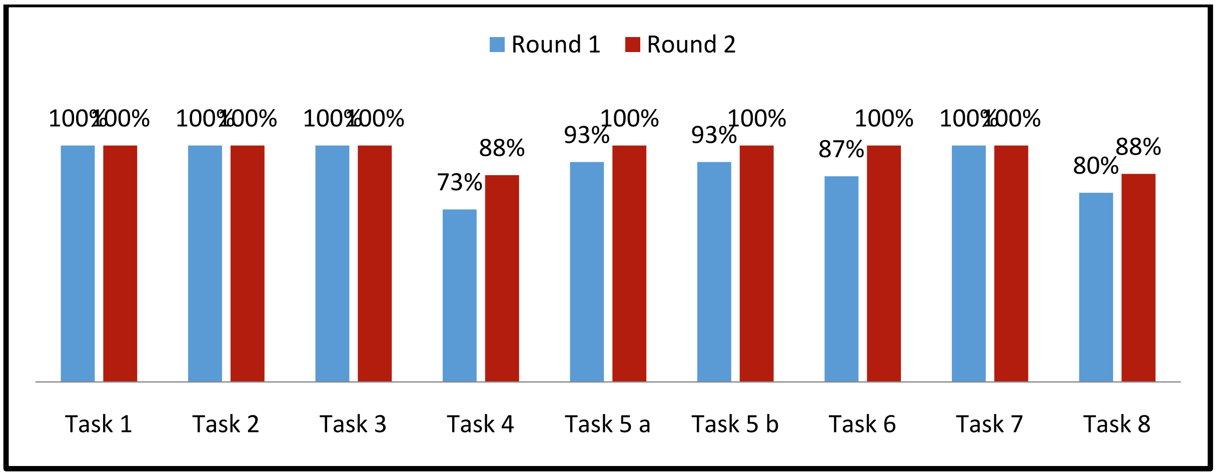


Figure 5-9 - Task Completion Rate comparison between 1st (n = 15) and 2nd (n = 8) evaluation

The completion rate of verifying the scoreboard results (Task 5 a) and filter the scoreboard results (Task 5 b) both improve from 93% during the first evaluation to 100% during the second evaluation. The improvement is a direct result of making the scoreboard easier to filter, recommended change 3 in Table 5-1.

The completion rate of synchronizing the Molo app (Task 6) rises from 87% during the first evaluation to 100% during the second evaluation. The improvement can be attributed to making the app indicate when the synchronization has started and when the synchronization process is complete, recommended change 6 in Table 5-1.

The completion rate of modifying the participants profile and logging back in to Molo with the new details (Task 8) improved from 80% during the first evaluation to 88% during the second evaluation. The improvement can be attributed to allowing the participants see what they typed into the password category, recommended change 8 in Table 5-1.

5.3.2.2 Post-Test Questionnaire

Figure 5-10 illustrated the comparison of results obtained during the first and second evaluation obtained from the post-test questionnaire (Appendix D – Post-test Questionnaire).

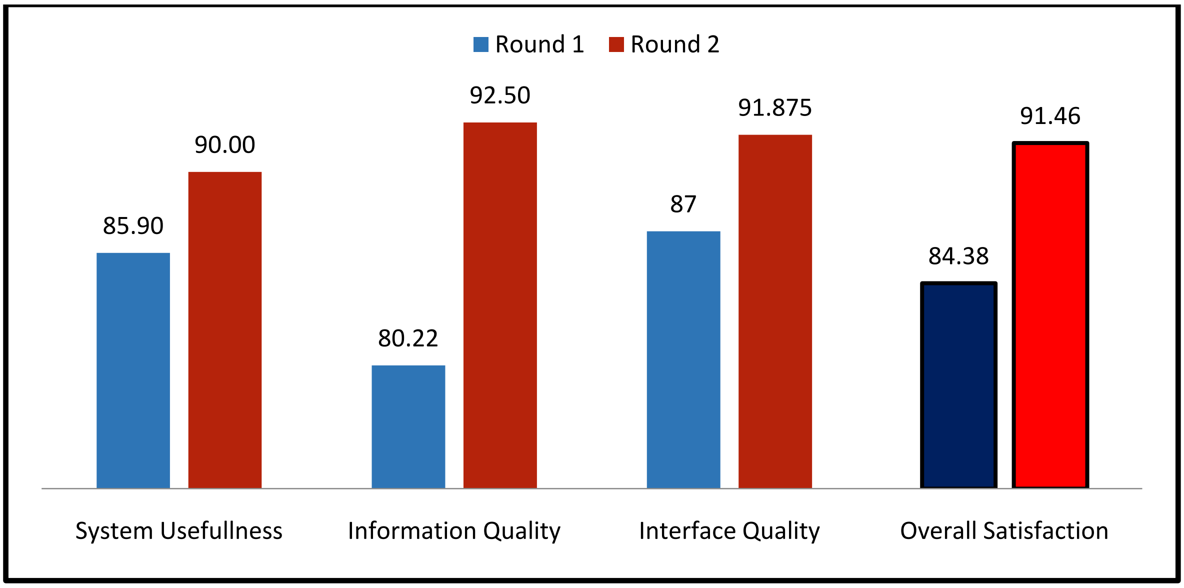


Figure 5-10 - Post Test Questionnaire Results comparison of 1st (n = 15) and 2nd (n = 8) evaluation

The system usefulness of the app improves from 85.9% to 90%, the information quality improves from 80.22% to 92.5%, and the interface quality improves 87% to 91.875%. The improvement across all three aspects results in the overall satisfaction of the app rising from 84.38% during the first evaluation to 91.46% during the second evaluation.

5.3.2.3 Task Load Index Questionnaire

The results of the Task Load Index Questionnaire was also evaluated and compared to the results of the first evaluation, as illustrated in Figure 5-11. All categories show improved results, with Temporal Demand being the only exception.

Results obtained from the second evaluation showed that participants felt the app was less mentally and physically demanding; they needed to exude less effort to complete the tasks; the frustration level is lower; the performance is higher.

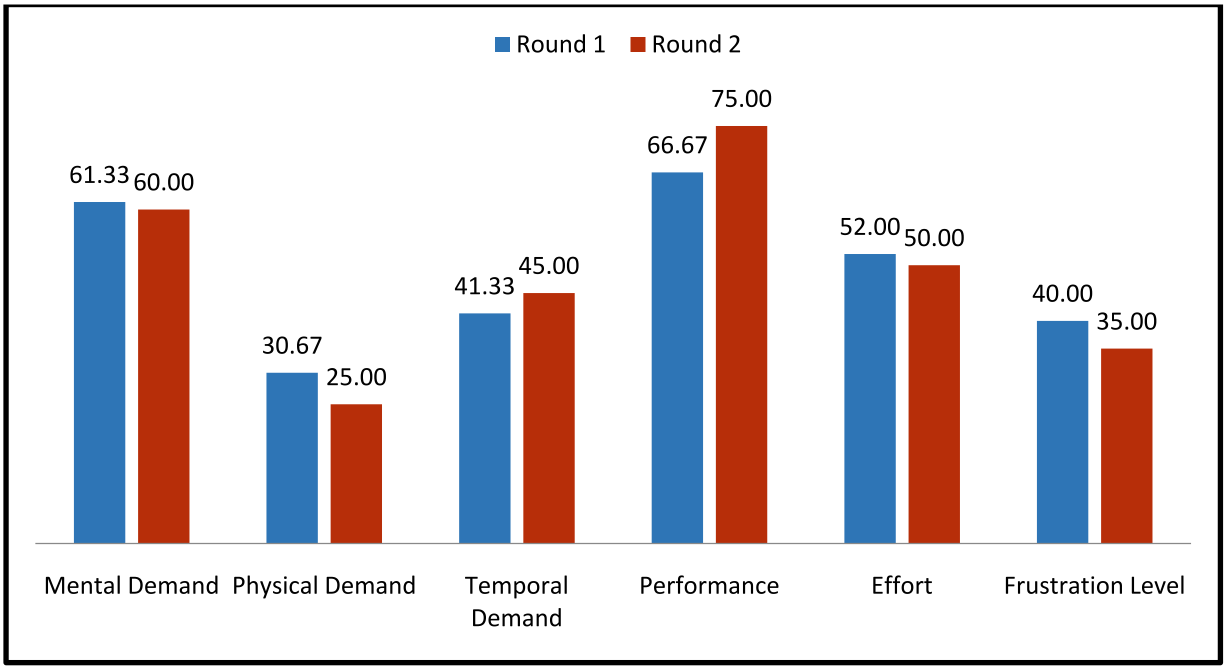


Figure 5-11 - Task Load Index Questionnaire Results comparison of 1st (n = 15) and 2nd (n = 8) evaluation

5.3.2.4 Participant Feedback

Figure 5-12 illustrates what aspects of the Molo app participants like the most. Three participants said the app was easy to use, while two participants said they liked the interface of the app. During the second evaluation 83% of participants said they have learned new words by using the Molo app.

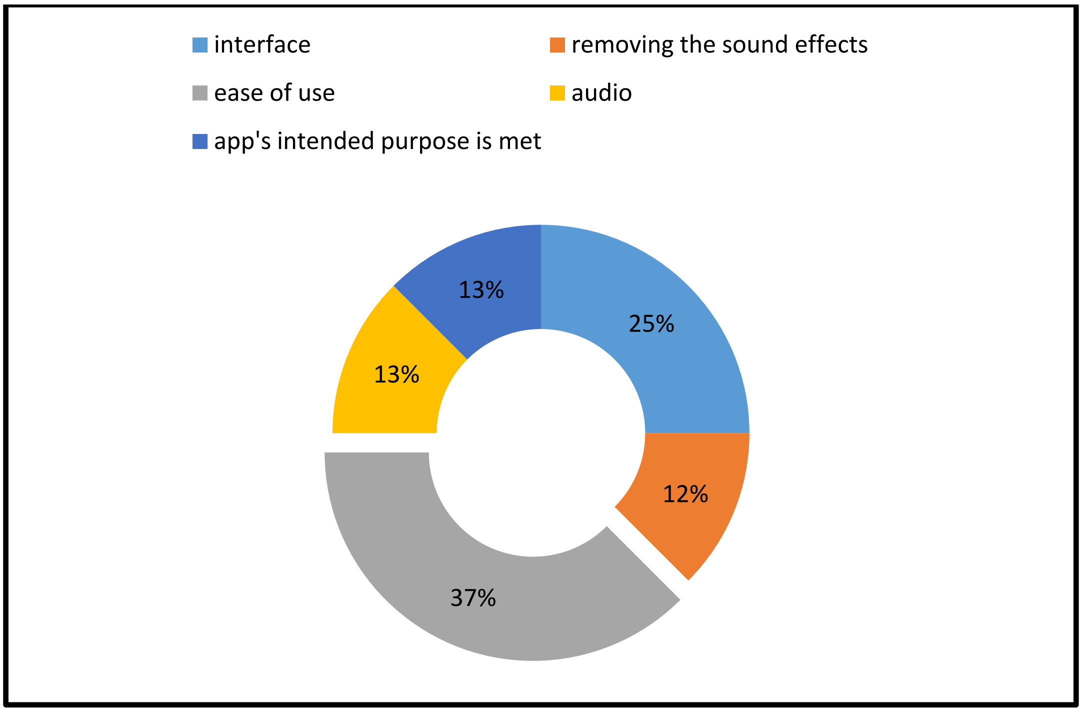


Figure 5-12 - Most Positive Aspects of the Molo app during 2nd evaluation (n = 8)

Figure 5-13 illustrates what aspects of the Molo app participants disliked the most during the second evaluation. Half of the participants said that there was not anything that they disliked about the app, rising from 40% during the first evaluation.

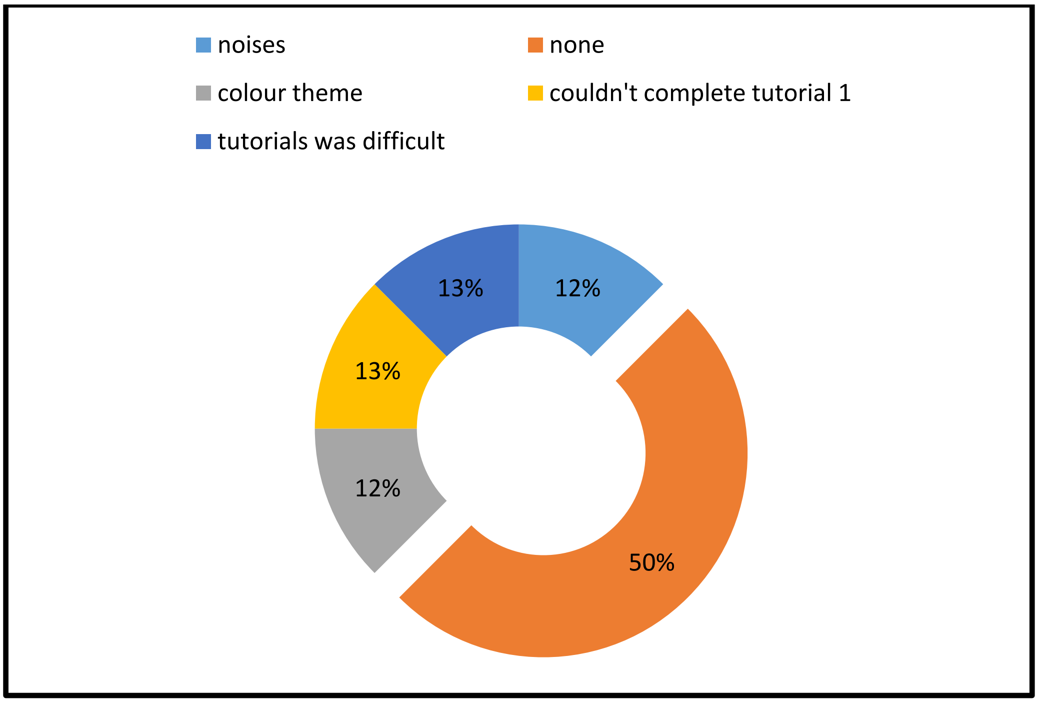


Figure 5-13 - Most Negative Aspects of the Molo app during 2nd evaluation (n = 8)

5.4 Conclusion

The evaluation of the Molo app was done to determine whether the app successfully met the criteria specified in Chapter 3. The evaluation consisted of a usability study that involved participants filling in a pre-test questionnaire, complete a task list, and fill in a post-test questionnaire. The goal of the pre-test questionnaire was to determine how many isiXhosa words the participants knew before using the Molo app. isiXhosa speakers were not asked how many English words they knew. The task list was created by using the Recommended Task List in Chapter 3 as guideline. The post-test questionnaire was used to determine the participants overall opinion and satisfaction of the app.

Fifteen participants partook in the evaluation. After analysing the results, the developer could infer that several changes, listed in 5.2.5, could be made to improve the user experience. Most significant was that the information/instructions provided by the app were not clear enough.

After all the recommended changes were made to the app, a second evaluation, consisting of eight participants, was conducted. Comparing the results of the first evaluation in 5.2.4 with the results of the second evaluation in 5.3.1, the reader can draw the conclusion that the changes made to the app made a significant improvement to the user's overall satisfaction of the app. Figure 5-10 shows that the Molo app is useful, provides good clear-cut instructions, and has a pleasant interface. When combining these three aspects, the Molo app scores an overall satisfaction of 91.46%.

During the first evaluation 82% of the participants said they have learnt new words by using the Molo app. After making the recommended changes, 83% of the participants during the second evaluation said they learnt new words.

The next chapter will conclude this project. The aim of the next chapter is to determine if the project, and by extension the Molo app, has satisfied the Project Goal in Chapter 1.

Chapter 6. Conclusion

6.1 Project Overview

The aim of this project was to determine whether a mobile app that teaches the user basic vocabulary of a foreign language could be used as a supplemental tool to formal learning. An app called Molo was then created to teach the user vocabulary of English or isiXhosa. Molo runs on the Android platform and was created using DSR as methodology, and RUP as development methodology. The following sections describe how each part of the project was executed and which methodology was used.

6.1.1 Literature Review

The literature review of this project formed part of The Rigor Cycle in DSR. The goal of the literature review was to provide rationalization to decision made during the design and implementation of the software artefact.

When extant systems were evaluated, the developer found that there is currently no mobile app that teaches the user basic vocabulary of South African languages (English excluded). Such apps do however exist for other languages, such as French and German. The Aweza app comes the closest to teach a user new words and phrases. However Aweza never evaluates the user's answer- the user uploads his voice recording onto the app's database and other users can then vote for it.

Thus the need for an app that teaches a user basic vocabulary was identified. This app would be different from its predecessors in that it would teach the user new words/phrases using innovative and immersive tutorials. Tutorials would consist of various lessons and exercises.

6.1.2 Prototype Design

This section falls under The Design Cycle of DSR. Using the knowledge obtained from The Rigor Cycle, the application domain of the app was analysed. A use case diagram was set up to graphically depict in what ways the developer and the user expects to interact with the app. Using the evaluation of extant systems, along with the analysis of the application domain and use cases as guidelines a set of functional and non-functional requirement was determined. The requirements led to the creation of a recommended task list that the app should be able to do. The initial database design and screen designs were then created to adhere to requirements of the app.

6.1.3 Implementation and Evaluation of Prototype

The designs and requirements specified in The Design Cycle of DSR was then used to create a working software artefact, the Molo app. Molo was created using RUP and provided all of the functionality that it was expected to have.

A first evaluation was conducted by having 15 participants test Molo. After analysing the results a list of recommended changes was created based on the participants' feedback. The most notable problem participants experienced was the information quality of the app- participants could not immediately figure out how to complete each exercise.

After making the recommended changes to the app, a second evaluation consisting of 8 participants was conducted. The comparison of first and second evaluation results showed improvement in the app's usefulness, information quality, and interface quality. Consequently the overall satisfaction the app provided also increased.

Important also was that 82% and 83% of the relevant participants said that they learnt new words using Molo, during the first and second evaluation respectively. This proves that the app could be used as a supplemental tool to formal learning.

6.2 Future Work

There are several different ways in which the Molo app can be improved upon and taken forward. The most obvious way is to add more tutorials. The app currently includes three tutorials: Greetings; Questions; Body Parts. Future developers could add tutorials where the user learns directions, or how to tell time, in the foreign language. Developers who would want to add more tutorials to the app need only follow the steps that was used to create the existing tutorials, explained in Chapter 4.

Other South African languages besides English and isiXhosa could also be added to the app. The database and system is designed in such a manner that minimal changes would have to be made when adding other languages. Because electronic resources are so scarce for South African languages (English being the obvious exception), the Molo app is the ideal tool to learn another South African language.

Lastly, voice recognition would be an interesting option to consider with regards to the voice exercises in the app. The voice exercise currently serves as a self-evaluation exercise for the user. By implementing voice recognition, the user would have to correctly pronounce the word before being allowed to move on to the next exercise. Voice recognition was not part of the scope of this project because no electronic resources are available in isiXhosa.

6.3 Reflection

During the course of this project, I found that isiXhosa is much more widely used than I initially thought. As someone who did not know a single isiXhosa word at the start of the project, and who has learned quite a few working on this project, I now often recognize isiXhosa being spoken in public places. I also find that isiXhosa people respond very friendly when I greet them in their native language.

A particularly difficult challenge during the design and implementation of the exercises was how to teach someone new words/phrases in such a way that they would remember. The challenge of creating immersive and innovative learning material was hard, but nonetheless I found it very interesting and pleasant to work on. I believe the combination of the colourful interface of Molo, the audio feature, and relevant images displayed during the exercise will help users learn new words faster and remember them for longer.

One of the aspects that I did not always appreciate during the course of the project was the vagueness of information obtained during interviews with the language experts. As an example, I would ask “What do you consider to be the most important thing to teach your students when they are learning a new language?” There never seemed to be a clear-cut answer to those types of questions, and previous literature on the matter also did not satisfactorily answer my questions.

6.4 Conclusion

The goal of this project was to create a mobile app that could teach the user vocabulary of a foreign language. The Molo app that was created provided all the required functionality and fulfilled all the specified requirements. Of the relevant participants, 83% said that they learned new words during the second evaluation of the app, proving that Molo could be used as a supplemental tool to formal learning. As such this project is considered to be a success.

Bibliography

Agilemethodology.org. (2012). Agile Methodology. Retrieved from http://agilemethodology.org/

Android Developers. (2014a). Action Bar. Retrieved October 21, 2014, from http://developer.android.com/guide/topics/ui/actionbar.html

Android Developers. (2014b). Android Design Patterns. Retrieved October 21, 2014, from https://developer.android.com/design/patterns/index.html

Android Developers. (2014c). ProgressBar. Retrieved October 21, 2014, from http://developer.android.com/reference/android/widget/ProgressBar.html

Android Developers. (2014d). RatingBar. Retrieved October 21, 2014, from http://developer.android.com/reference/android/widget/RatingBar.html

Audacity Developers. (2014). Audacity: About Audacity. Retrieved October 21, 2014, from http://audacity.sourceforge.net/

Bentley, L. D., & Whitten, J. L. (2007). *Systems Analysis & Design for the Global Enterprise* (7th editio.). New York, New York, USA: McGraw-Hill/Irwin.

Cavus, N., & Ibrahim, D. (2009). m-Learning: An experiment in using SMS to support learning new English language words. *British Journal of Educational Technology*, *40*(1), 78–91. doi:10.1111/j.1467-8535.2007.00801.x

Coronel, C., Rob, P., & Crockett, K. (2009). *Database Systems: Design, Implementation & Management* (internatio.). CENGAGE learning.

DStv Online. (2012). Mobile Phone Usage in South Africa.

Francis, W. J. (2013). Try Android's useful drag and drop API. Retrieved October 21, 2014, from http://www.techrepublic.com/blog/software-engineer/try-androids-useful-drag-and-drop-api/

Godwin-jones, R. (2011). Emerging technologies: Mobile apps for langauge learning. *Language Learning & Technology*, *15*(2), 2–11.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, (52), 139–183.

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2), 87–92.

IBM. (1998). Rational Unified Process Best Practices for Software for Software Development Teams. *Rational Software White Paper*, *11*(1).

Koole, M. L. (2009). Mobile Learning - A Model for Framing Mobile Learning.

Kukulska-Hulme, A. (2009a). Innovation in Mobile Learning : A European Perspec- tive.

Kukulska-Hulme, A. (2009b). Will mobile learning change language learning. *ReCALL*, *21*(2), 157–165.

Meier, R. (2012). *Proffesional Android 4 Application Development*. (D. Ulery, Ed.). John Wiley & Sons.

Munik, A. (1994). *Learn Xhosa with Anne Munik* (5th editio.). Pietermaritzburg, KwaZulu-Natal: Shuter & Shooter.

ORACLE. (2014). MySQL Workbench. Retrieved October 21, 2014, from http://www.mysql.com/products/workbench/

Quinn, C. (2000). mLearning: Mobile, Wireless, In-Your-Pocket Learning. *Linezine*.

Sandberg, J., Maris, M., & de Geus, K. (2011). Mobile English learning: An evidence-based study with fifth graders. *Computers & Education*, *57*(1), 1334–1347. doi:10.1016/j.compedu.2011.01.015

Siskin, C. B. (2009). Language Learning Applications for Smartphones, or Small Can Be Beautiful.

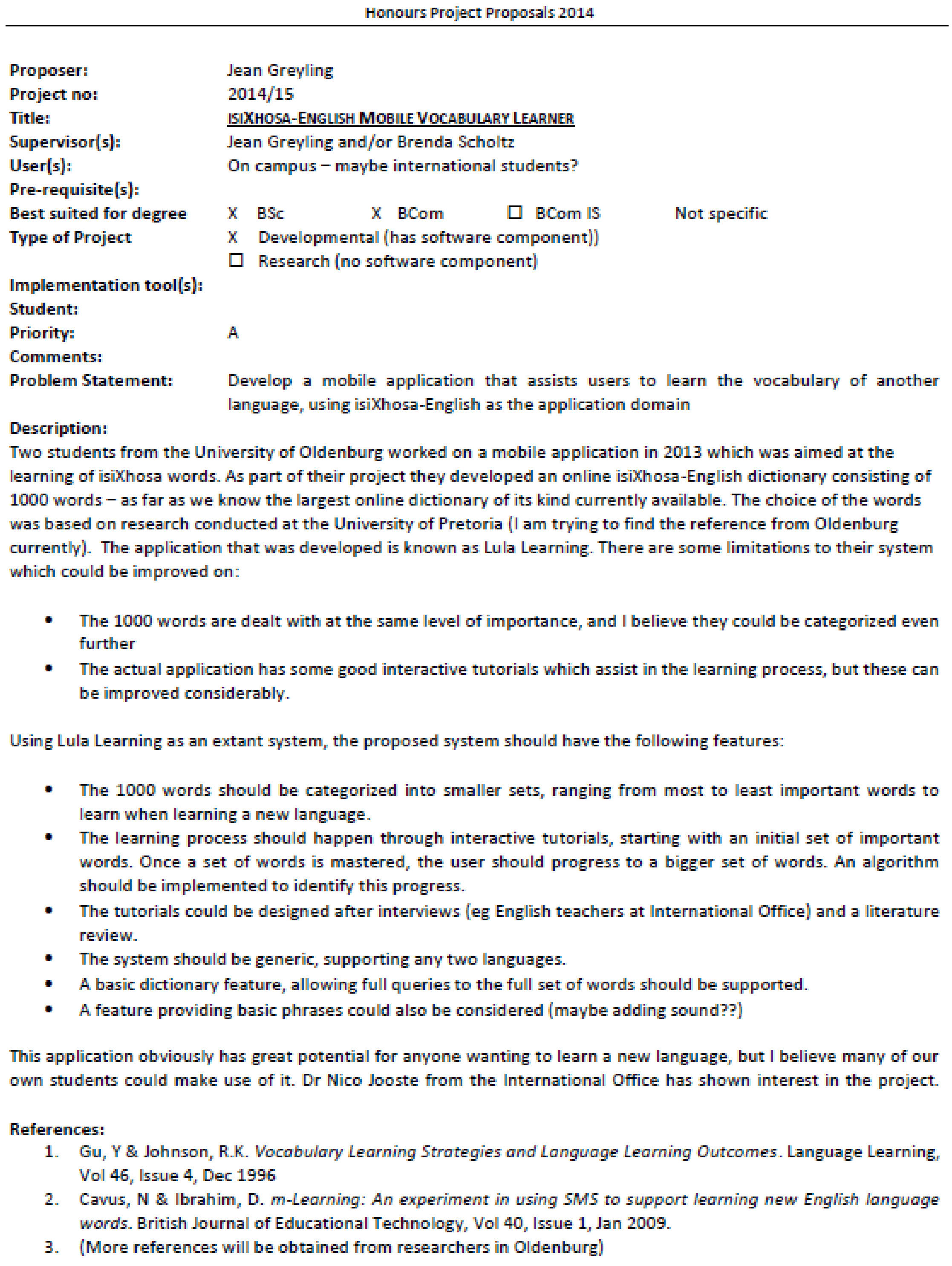
Speak easy Xhosa. (2014). Retrieved March 15, 2014, from http://www.speakeasyxhosa.co.za/index.php?lang=en

Vogel, L. (2014). Using lists in Android (ListView) - Tutorial. Retrieved October 21, 2014, from http://www.vogella.com/tutorials/AndroidListView/article.html

William, A., & Thomas, T. (2013). *Measuring the user experience: collecting, analyzing, and presenting usability metrics* (2nd ed.). Elsevier Inc.

William, A., & Tullis, T. (2013). *Measuring the user experience* (2nd editio.). Newnes: Morgan Kaufmann.

Appendix A – Project Proposal



Honours Project Proposals 2014

Proposer: Jean Greyling
Project no: 2014/15
Title: ISIXHOSA-ENGLISH MOBILE VOCABULARY LEARNER
Supervisor(s): Jean Greyling and/or Brenda Scholtz
User(s): On campus – maybe international students?
Pre-requisite(s):
Best suited for degree X BSc X BCom □ BCom IS Not specific
Type of Project X Developmental (has software component))
□ Research (no software component)
Implementation tool(s):
Student:
Priority: A
Comments:
Problem Statement: Develop a mobile application that assists users to learn the vocabulary of another language, using isiXhosa-English as the application domain

Description:

Two students from the University of Oldenburg worked on a mobile application in 2013 which was aimed at the learning of isiXhosa words. As part of their project they developed an online isiXhosa-English dictionary consisting of 1000 words – as far as we know the largest online dictionary of its kind currently available. The choice of the words was based on research conducted at the University of Pretoria (I am trying to find the reference from Oldenburg currently). The application that was developed is known as Lula Learning. There are some limitations to their system which could be improved on:

- The 1000 words are dealt with at the same level of importance, and I believe they could be categorized even further
- The actual application has some good interactive tutorials which assist in the learning process, but these can be improved considerably.

Using Lula Learning as an extant system, the proposed system should have the following features:

- The 1000 words should be categorized into smaller sets, ranging from most to least important words to learn when learning a new language.
- The learning process should happen through interactive tutorials, starting with an initial set of important words. Once a set of words is mastered, the user should progress to a bigger set of words. An algorithm should be implemented to identify this progress.
- The tutorials could be designed after interviews (eg English teachers at International Office) and a literature review.
- The system should be generic, supporting any two languages.
- A basic dictionary feature, allowing full queries to the full set of words should be supported.
- A feature providing basic phrases could also be considered (maybe adding sound??)

This application obviously has great potential for anyone wanting to learn a new language, but I believe many of our own students could make use of it. Dr Nico Jooste from the International Office has shown interest in the project.

References:

1. Gu, Y & Johnson, R.K. *Vocabulary Learning Strategies and Language Learning Outcomes*. Language Learning, Vol 46, Issue 4, Dec 1996
2. Cavus, N & Ibrahim, D. *m-Learning: An experiment in using SMS to support learning new English language words*. British Journal of Educational Technology, Vol 40, Issue 1, Jan 2009.
3. (More references will be obtained from researchers in Oldenburg)

Appendix B – Pre-test questionnaire

Participant number:

Do you know any isiXhosa words?

Yes □

No □

If you answered 'Yes' above, how many isiXhosa words do you know?

0 □

1-10 □

11-20 □

21-50 □

50 or more □

List the isiXhosa words you use most frequently

1)
2)
3)
4)
5)

Appendix C – Task List

Participant number:

Task 1

Register as a new user in Molo

Task 2

Login to Molo using the details you used in Task 1

Task 3

Find the Tutorials

How many Tutorials are there to choose from?

____

Task 4

Complete Tutorial 1 & Tutorial 2

Task 5

Go to the Scoreboard.

Are your results listed on the Scoreboard correctly?

Yes □ □

Can you filter the Scoreboard results effectively?

Yes □ □

If ‘No’, say why not

____

Task 6

Click the refresh/synchronize button on the Dashboard. After waiting approximately 15 seconds, go to Tutorials.

How many tutorials are there to choose from now?

____

Task 7

Click the profile button on the Dashboard.

Are your profile details displayed correctly?

Yes □ □

If 'No', say why not

____

Task 8

Modify your password and click save.

Can you log back into Molo using your new details?

Yes □ □

Appendix D – Post-test Questionnaire

Participant number:

Section A: Biographical Questions

Gender: □ Male □ Female

Age:

State your home language □ Afrikaans □ English □ isiXhosa

□ Other: ___________________________

Do you own an Android smart phone? □ Yes □ No

State your experience with Android touch screen smart phones □ Low □ Moderate □ High

Education: □ 1st Year □ 2nd Year □ 3rd Year □ Postgraduate

Section B: Post-Test Questionnaire

Please indicate your response to the following statements by placing an 'X' in the appropriate block on the scale:

Overall, I am satisfied with how easy it is to use this system.

STRONGLY DISAGREE STRONGLY
AGREE

1	2	3	4	5

It was simple to use this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

I can effectively complete the tasks using this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

I am able to complete the tasks quickly using this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

I am able to efficiently complete the tasks using this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

I feel comfortable using this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

It was easy to learn to use this system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

I believe I became productive quickly using the system.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

Whenever I make a mistake using the system, I recover easily and quickly.

STRONGLY DISAGREE STRONGLY

AGREE

1	2	3	4	5

The information (such as instructions, options, and relevant buttons) provided with this system is clear.

STRONGLY DISAGREE STRONGLY AGREE

1 2 3 4 5

It is easy to find the information I needed.

STRONGLY DISAGREE STRONGLY AGREE

1 2 3 4 5

The information provided for the system is easy to understand.

STRONGLY DISAGREE STRONGLY AGREE

1 2 3 4 5

The information is effective in helping me complete the tasks and scenarios.

STRONGLY DISAGREE STRONGLY AGREE

1 2 3 4 5

The organization of information on the system screens is clear.

STRONGLY DISAGREE STRONGLY AGREE

1 2 3 4 5

The interface of this system is pleasant.

STRONGLY DISAGREE STRONGLY
AGREE

1	2	3	4	5

I like using the interface of this system.

STRONGLY DISAGREE STRONGLY
AGREE

1	2	3	4	5

This system has all the functions and capabilities I expect it to have.

STRONGLY DISAGREE STRONGLY
AGREE

1	2	3	4	5

Overall, I am satisfied with this system.

STRONGLY DISAGREE STRONGLY
AGREE

1	2	3	4	5

Section C: Task Load Index Questionnaire

Mental demand: What amount of mental and perceptual activity was required from you?

VERY LOW VERY HIGH

1	2	3	4	5

Physical demand: What amount of physical activity was required from you?

VERY LOW VERY HIGH

1	2	3	4	5

Temporal demand: What amount of time pressure did feel due to pace of the tasks?

VERY LOW VERY HIGH

1	2	3	4	5

Performance: How successful were you in completing the tasks?

VERY LOW VERY HIGH

1	2	3	4	5

Effort: What amount of effort was required from you to complete the tasks?

VERY LOW VERY HIGH

1	2	3	4	5

Frustration level: How insure, discouraged, irritated, stressed and annoyed did you feel?

VERY LOW VERY HIGH

1	2	3	4	5

Section D: Comments

What aspect of the Molo app did you like the most?

What aspect of the Molo app did you dislike the most?

What changes/improvements would you suggest for the Molo app?

Did you learn any new words using the Molo app?

□ Yes □ No

If ‘Yes’, list up to 5 words that you have learnt

1.

2.

3.

4.

5.

Appendix E – Raw Data from Evaluation 1

Participant	Gender	Age	Home Language	Owns Android Device	Experience with Android Touch Screen Devices	Education
1	Male	21	English	Yes	High	Postgraduate
2	Male	24	isiXhosa	Yes	High	Postgraduate
3	Male	24	English	Yes	High	Postgraduate
4	Male	25	Shona	Yes	High	Postgraduate
5	Male	24	Afrikaans	Yes	High	Postgraduate
6	Female	24	Afrikaans	Yes	High	Postgraduate
7	Male	23	English	Yes	High	Postgraduate
8	Female	25	Shona	No	High	Postgraduate
9	Female	22	English	No	Low	Postgraduate
10	Male	20	isiXhosa	Yes	High	3rd Year
11	Male	22	English	Yes	High	2nd Year
12	Female	38	Setswana	Yes	Moderate	Postgraduate
13	Female	21	isiXhosa	No	Moderate	Postgraduate
14	Female	22	isiXhosa	Yes	High	Postgraduate
15	Female	23	Afrikaans	No	Moderate	Postgraduate

System Usefulness

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q1	0	0	1	5	9	68	4.53	90.67
Q2	0	0	0	9	6	66	4.40	88.00
Q3	0	0	4	8	3	59	3.93	78.67
Q4	0	0	4	7	4	60	4.00	80.00
Q5	0	1	1	8	5	62	4.13	82.67
Q6	0	0	1	7	7	66	4.40	88.00
Q7	0	0	0	5	10	70	4.67	93.33
						64.43	**4.30**	**85.90**

Information Quality

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q9	0	3	5	4	3	52	3.47	69.33
Q10	0	2	3	6	4	57	3.80	76.00
Q11	0	0	3	7	5	62	4.13	82.67
Q12	0	0	2	4	9	67	4.47	89.33
Q13	0	1	1	9	4	61	4.07	81.33
Q14	0	1	3	4	7	62	4.13	82.67
						60.17	**4.01**	**80.22**

Interface Quality

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q15	0	0	1	7	7	66	4.4	88
Q16	0	0	2	5	8	66	4.4	88
Q17	0	0	3	6	6	63	4.2	84
Q18	0	0	1	7	7	66	4.4	88
						65.25	**4.35**	**87**

Task Load Index	Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q1	Mental Demand	2	5	2	2	4	46	3.07	61.33
Q2	Physical Demand	9	4	2	0	0	23	1.53	30.67
Q3	Temporal Demand	6	5	2	1	1	31	2.07	41.33
Q4	Performance	2	3	1	6	3	50	3.33	66.67
Q5	Effort	2	4	7	2	0	39	2.60	52.00
Q6	Frustration Level	8	2	2	3	0	30	2.00	40.00
							36.50	**2.43**	**48.67**

Completion Rate

	Yes	No	Errors
Task 1	100%	0%	
Task 2	100%	0%	
Task 3	100%	0%	0
Task 4	73%	27%	
Task 5 a	93%	7%	
Task 5 b	93%	7%	
Task 6	87%	7%	7
Task 7	100%	0%	
Task 8	80%	20%	

Most Positive Aspect	Frequency
Easy to learn new words	1
Voice Pronunciation	4
Fun	1
Easy to use	5
You can practice saying the words	1
Audio playback/Recording	1
Matching Exercises	2
Pictures are relevant to the words on screen	2
Sounds make app interesting	2

Most Negative Aspect	Frequency
Some information/instructions are unclear	2
None	6
Writing exercise is case sensitive	1
Login/Registration	1
Writing exercise marks correct answer as correct	1
Voice Exercise instructions	1
Noises	1
Got stuck in Tutorial 1	1
App became slow/crashed	1

Changes/Improvements	Frequency
Make instructions clearer	2
5 stars instead of 3	1
Scoreboard filter should be more clear	1
Enter password twice to validate	1
Make voice recording/playback simpler	3
Option to switch noises on/off	1
Hints in Writing exercise confuse users	1
More languages	1
None	2
give user 2nd chance to answer some questions	1
Show characeter for a short while when entering password	1
Go straight to Dashboard after registration	1
Scoreboard should display what user answered wrong	2

Participant	Learnt new words	Wrote down
1	Yes	2
2	No	0
3	Yes	0
4	Yes	3
5	Yes	3
6	No	0
7	Yes	1
8	Yes	5
9	Yes	3
10	No	0
11	Yes	3
12	No	0
13	No	0
14	No	0
15	Yes	5

Appendix F – Raw Data from Evaluation 2

Participant	Gender	Age	Home Language	Owns Android Device	Experience with Android Touch Screen Devices	Edu
1	Male	20	Chinese?	Yes	High	3rd
2	Male	23	English	Yes	High	3rd
3	Female	26	Zulu	No	Moderate	3rd
4	Male	23	Shona	Yes	High	Post
5	Male	24	English	Yes	High	Post
6	Female	19	Zulu	No	High	2nd
7	Female	21	isiXhosa	No	Moderate	Post
8	Male	22	Afrikaans	Yes	High	Post

System Usefulness

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q1	0	0	0	3	5	37	4.63	92.50
Q2	0	0	0	5	3	35	4.38	87.50
Q3	0	0	1	2	5	36	4.50	90.00
Q4	0	0	0	4	4	36	4.50	90.00
Q5	0	0	0	4	4	36	4.50	90.00
Q6	0	0	2	1	5	35	4.38	87.50
Q7	0	0	0	3	5	37	4.63	92.50
						36.00	4.50	90.00

System Usefulness

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q1	0	0	0	3	5	37	4.63	92.50
Q2	0	0	0	5	3	35	4.38	87.50
Q3	0	0	1	2	5	36	4.50	90.00
Q4	0	0	0	4	4	36	4.50	90.00
Q5	0	0	0	4	4	36	4.50	90.00
Q6	0	0	2	1	5	35	4.38	87.50
Q7	0	0	0	3	5	37	4.63	92.50
						36.00	4.50	90.00

Interface Quality

Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q15	0	0	1	2	5	36	4.50	90
Q16	0	0	0	3	5	37	4.63	92.5
Q17	0	0	0	2	6	38	4.75	95
Q18	0	0	0	4	4	36	4.50	90
						36.75	4.59375	91.875

Task Load Index	Index	1	2	3	4	5	Total	Mean per Question	As Percentage
Q1	Mental Demand	1	2	3	0	2	24	3.00	60.00
Q2	Physical Demand	6	2	0	0	0	10	1.25	25.00
Q3	Temporal Demand	3	2	1	2	0	18	2.25	45.00
Q4	Performance	0	0	3	4	1	30	3.75	75.00
Q5	Effort	2	2	3	0	1	20	2.50	50.00
Q6	Frustration Level	5	0	3	0	0	14	1.75	35.00
							19.33	**2.42**	**48.33**

Most Positive Aspect	Frequency
interface	3
sound effects are optional	1
voice exercise	2
ease of use	3
clear instructions	2
overall tutorials	1
app's purpose is met	1

Most Negative Aspect	Frequency
sound effects	1
password cursor moves around	1
none	4
colours	1
could not complete tutorial 1	1
tutorials are difficult	1

Changes/Improvements	Frequency
separate instructions for each exercise	1
none	2
display words/phrases bigger in matching exercise	1
back button to review past lessons	1
changing user profile is not clear	1
make experience less juvenile	1
make tutorials easier	1
add more tutorials	1

Participant	Learnt new words	Wrote down
1	yes	3
2	yes	2
3	no	0
4	yes	5
5	yes	3
6	no	0
7	no	0
8	yes	5