MOBILE APPLICATION DESIGN AND DEVELOPMENT ON TOUCH SCREEN PHONES FOR ADULTS WITH DYSLEXIA

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF MASTER OF SCIENCE IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2014

Norman D Dambul
School of Computer Science

Table of Contents

Word Count : 19588

List of Figures

List of Tables

Abstract

A mobile application was made with the goal of helping dyslexic adults read better on a touch screen phone. Dyslexia is a disability affecting a person's ability to read and recognize words. The first phase of the research involves a detailed look into the causes and treatment for dyslexia including case studies of tools available to manage it. The methods used to conceive the idea for the application, named Touch Screen Overlay(TCO), will be explained as well as methods to evaluate its effectiveness. An in-depth look into the building blocks of an Android application will also be conducted. The design of the application will be based on making a virtual coloured overlay in the form of an Android application as research shows that a different coloured background can help a dyslexic to read better. The application was tested on dyslexics and their feedback was evaluated. The outcome of the research was finding the application helpful for dyslexics in regards to reading.

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

i The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Mancheste certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright,Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements whichthe University has entered into. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=487), in any relevant Dissertation restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Guidance for the Presentation of Dissertations.

1 Introduction

Mobile applications are slowly becoming the platform of choice for most people nowadays. More and more applications are being created to ensure most services which are available on a desktop computer can also be accessed from mobile devices. The market for these applications is growing bigger and bigger. Brand new and innovative ideas and applications keep on coming and adding to an already wide range of services and functions to suit most consumers. But what about the lesser known side of the market? What about the people with disabilities and visual impairments usage of mobile devices? According to a study by Papworth Trust [1], in the year 2012 there were 1.86 million people in the UK suffering from sight loss. From the same report, there are an estimated 40000 people under the age of 25 who are partially blind or sighted. What about the consumers who struggle to use even the most basic functions of a mobile device such as reading a simple text message. Are they considered irrelevant just because of the lack of their commercial impact.

It cannot be denied though that some companies, such as Apple, have given accessibility a lot of focus. For example the inclusion of the VoiceOver function which gives blind and partially blind users more accessibility options to their mobile devices. But Apple is not making sure that a certain standard of accessibility exists in applications made by third party companies where most of the applications are being produced. More focus is being put into making a mobile application interface look nice and clean rather than making it accessible to people with disabilities. It is really saddening to realize that while some applications which could allow a normal person to see the whole world through their mobile devices already exists, some people with disabilities are still struggling to read their text messages and emails. But due to the simplistic nature of mobile interfaces it can be potentially easier to create better accessibility on this platform.

1.1 Project Introduction

This project was built on the basis that more exposure should be given to the fact that there is still a lot to be discovered and improved on mobile applications accessibility compared to what is already available and that more research should be put towards it. The main reason that this topic is not highlighted regularly is because, creating a solution for it will not generate a lot of income as the market is small compared with normal users, and it might also affect just a small percentage of users so it can be easily ignored.

1.2 Project Aims

This project aims to create a mobile application designed for touch screen phones with design specifications to suit adults with dyslexia. Basically it is going to be a floatable application called Touch Colour Overlay (TCO) which is much like a widget, on a mobile screen which will act as a horizontal lens/overlay which will allow change in the colour, creating a different coloured background for the text currently under it. The colours for the lens/overlay would be customizable. Chapter 4 will give a detailed look on the application itself. Corresponding research which examines how people with disabilities interact with mobile applications will also be provided. Dyslexia is mainly a disability affecting a person's ability to read/write and recognize words. According to the Bristish Dyslexic Association [2], 10% of the British population are dyslexic and under the Equality Act 2010, dyslexia is recognized as a disability.

The report will touch on the actual definition of dyslexia, the effects and main causes of this disability, current treatment for this disability, relevant statistics and any journals or articles with useful insights. This is to ensure that a very detailed view is given to the disability itself to recognize all aspects of it and how to best manage it. Case studies on existing mobile applications will also be conducted to see what is available in the market today. The mobile application itself will be built on an android platform and will

utilize the Qslide SDK [3] from LG which will allow for mobile applications to be floatable on a mobile screen device much like a widget. The application built was also tested on volunteers and their results are evaluated on Chapter 5.

1.3 Justification of the project

Most of the applications on offer today provide mostly support on magnifying the text or enlarging the text. Some of them offer the ability to invert the colour and the ability to change the font of the text. As of yet, there is very minimal effort being made to specifically tackle the problem of dyslexia on a mobile device. This project is trying to produce a solution specifically to help dyslexics have better access to mobile devices. It also aims to provide a better assistant for dyslexics while using core functionality of mobile devices such as helping with reading messages from applications such as Whatapps and Facebook, and reading emails with applications such as Gmail instead of just focusing on web pages alone. When the project is complete it could become a stepping stone for an even more effective solution in the future.

1.4 Learning outcomes

The learning outcomes from this project are, an in-depth view into the daily struggle of people with disabilities using mobile applications, mobile application design interfaces which can be useful to those with disabilities and following standard guidelines from W3 accessibility guide, how to build mobile applications with accessibility in mind, how people with dyslexia manage mobile usage and how potential solutions with the use of IT and proper research could be used to help them, and finally using new technology(LG QSlide SDK) to create new and innovative mobile applications.

Those with dyslexia will benefit most from the product that this project will produce. The project is built with the intention of solving a problem related with dyslexia being an issue while using mobile applications and using the help of in-depth research into dyslexia and how to incorporate IT solutions to it. The outcome of the project will also use fairly new technology which most mobile developers have not used yet and this can

be beneficial to the IT community which could help to create even better mobile applications. The project's research outcome will also help shine more light on the importance of providing better accessibility to disabled people using mobile devices and make companies currently developing mobile applications to pay more attention to the accessibility side of mobile applications.

1.5 Evaluation method

Currently the application would be tested through subjective user testing where the user's overall experience and personal opinion on the effectiveness of the application for helping them read and recognize text will be taken. If the reader personally finds that the application helps them to read better or increases their confidence and self belief by making them feel they can read better with the help of it, without the constraints of bias and placebo effect, then the application will be considered a success. This is a realistically achievable evaluation method within the time scope of the project. In the future the ideal way to test the effectiveness of this application is to measure the reading speed of a disabled person using it. For example, a test based on time could be done, where the time it takes for a person to read through a text message or email before and after the usage of the application is measured. If there is an increase in reading speed then the application would be considered a success.

Other further types of evaluation can also be done by comparing the helpfulness level of the application compared with other types of support available to people with disabilities. This includes comparing it with text-to-speech applications which is the most popular way of tackling this problem, and also using specially created fonts such as OpenDyslexic and comparing its effectiveness with it.

1.6 Project Deliverables

The project will produce several items as stated in the table below.

1	Detailed research into dyslexia and case studies or reviews of current applications available to manage the disability
2	A mobile application called Touch Colour Overlay(TCO) to help adults with dyslexia to recognize text and read better on a mobile device, using the principle of coloured overlays.
3	Evaluation of the results from testing the application on dyslexics, and recommendation from dyslexics on other functionalities.

Table 1 : Project deliverables.

1.7 Summary of report

The report will consist of 6 main sections. The first part is the introduction as seen earlier. The next part, Chapter 2, is Background research which is a study into what the problem is actually about, how it is being handled and what applications are currently being used to handle it. Chapter 3, Research methodologies will essentially touch on how the application relates to the background study and research methods being used to actually conduct the study and build the application. Chapter 4, Design and Implementation, relates to how the design of the application was conceived and how the application was built and coded, with details of important functions used. Chapter 5, Testing and Evaluation, is about how the testing of the application with target users was done, how the volunteers were approached and the evaluation of the result of the testing. Finally Chapter 6, Future works and Conclusion, will provide a detailed view into the recommendations made by the volunteers and available technologies to implement them. A final Conclusion will be given based on the report's findings and outcomes.

2 Background research

2.1 Overview of disabilities interfering with mobile usage

Only three disabilities are going to be highlighted here which are colour blindness, low vision and dyslexia. The best way of explaining how these disabilities affect the people involved would be through the use of real life scenarios. The scenarios are based on Stories of Web Users from W3C (Web Accessibility Initiative) [4].

First scenario, Mr Adrian has colour blindness. He finds it very difficult to read text from many web pages due to the poor use of contrast in colours. He finds that many of the colours are unrecognisable due to his red/green colour blindness. Some site instructions such as, popular items are in red text and items which are still available are in green text seems all brown or yellow in colour to him . And when asked to fill only required fields which text was red in colour, he could not tell which one it was. This led to difficulties in using certain popular web pages, causing him to feel left out. Mr Adrian finally had to bookmark only those web pages which he was comfortable using.

Second scenario, Mr Mario has low vision. Mr Mario has trouble reading small text and clicking on small links on web pages. This is especially difficult in mobile screens where everything has to be scaled down so all the relevant content can be fitted through the screen. A lot of the web pages Mr Mario visits does not support text enlarging. Another big problem is CAPTCHA images used to distinguish human and computers apart. It is simply impossible for him to recognize the images and only a few web pages provide alternative methods. Mr Mario finally had to make use of the zoom function to read text which can be very difficult to use when reading a lot of text.

Third scenario, Miss Jane has dyslexia. She finds that some web pages are easier to read compared with others. This is due to the availability of graphics in the web pages which makes it easier to read. But most mobile web pages have a lot of text in them to compensate for the reduced screen size when conveying information. She also has

trouble reading long text messages and emails which have very minimal support for dyslexic readers. She also notices that alternative spelling suggestion applications, which help her a lot while using desktop applications, are very limited for mobile applications. She is reduced to using text-to-speech applications every time a piece of text gets too difficult to read.

All these scenarios can help to give us a brief view into how disabilities can affect people's usage of mobile application and can help us to realize the importance of relevant research in this field therefore further justifying this project. It is also very important to view the problem from the perspective of the people suffering from these disabilities to gain insight for producing the best possible solution.

2.2 Definition of dyslexia

There are many different definitions and descriptions of dyslexia currently available. This may differ due to the context or purpose the word is used. Its definition has also been problematic because different cases provide different levels of effects to the sufferers. Here are some definitions from different sources. The British Dyslexia Association Management Board [5] define dyslexia as a " specific learning difficulty that mainly affects the development of literacy and language related skills. It is likely to be present at birth and to be life-long in its effects. It is characterised by difficulties with phonological processing, rapid naming, working memory, processing speed, and the automatic development of skills that may not match up to an individual's other cognitive abilities. ". The full definition can be obtained from the British Dyslexia Association website.

The UK NHS (National Health Service) [6] defines dyslexia as " a common reading difficulty that mainly affects the way people read and spell words ". The NHS also described dyslexia as a spectrum disorder and people suffering from it could have difficulties with phonological awareness, verbal memory, rapid serial naming and verbal processing speed. And the meaning of the word itself from Oxford Dictionaries [7] is " A general term for disorders that involve difficulty in learning to read or interpret words, letters, and other symbols, but that do not affect general intelligence. ".

From the research done, my understanding of dyslexia is as follows. It is a naturally occurring disability that is gained from birth and affects a person's ability to read and write words and sentences compared to a normal person and this will thus affect their ability to learn because most education systems are built upon the abilities to read and write. The disability is usually only recognized at a later part of life when someone notices their ability to read and write does not match up with another person even when they spend the same amount of time learning how to do it properly. It does not produce any physical symptoms and it does not carry specific characteristic with it. Usually people suffering from dyslexia may also suffer from problems associated with learning disabilities such as motor-perceptual syndromes, language delays, easily distracted, hyperactivity and decreased attention span [8]. All of these different elements may cause confusion on how dyslexia should be viewed and what to look for when actually identifying it.

Dyslexia can generally be categorized into two types which are acquired dyslexia and developmental dyslexia [9]. According to the same resource, acquired dyslexia happens after a person has learn to read and is usually the effect of certain brain traumas which will affect a particular aspect of reading whereas developmental dyslexia on the other hand develops during the development of reading skills. Acquired dyslexia is further categorized into two subtypes which are surface dyslexia and phonological dyslexia. Where surface dyslexia is the ability to decode and read nonsense words accurately but read irregularly spelled words incorrectly and sounding them using learned phonics rules whereas phonological dyslexia is the ability to read real words but inability to decode nonsense words.

2.3 Identifying the relationship between dyslexia and IQ

Information regarding the relationship between dyslexia and IQ is very subjective as the research on dyslexia is still ongoing and much is still not known to the medical world. It is not necessarily correct to say that people suffering from dyslexia have a lower IQ than a normal person. It is only because that the current education system focuses so much on the ability to read and write that a person with dyslexia is viewed as having a

lower IQ. This happens because the problem is measured incorrectly using the same tools whereby it should have been measured using a more diverse set of tools, although this might be difficult to implement in the real world due to the amount of changes that would need to be made to the current system.

An article from Gad Elbeheri and John Everatt from the book The Routledge Companion to Dyslexia (2009) [10] properly debates the theory of the role of IQ in dyslexia identification. The article found that the traditional method of identifying dyslexia was to compare reading ability with IQ-predicted ability where if there was a difference between IQ-predicted reading levels with actual reading ability then the person was diagnosed as dyslexic. A person had to have a huge difference between general mental abilities and reading achievements to be considered dyslexic. This shows that a person with dyslexia is usually normal in every other way except in the ability to read and write.

The article also found that these methods were built on the assumption that IQ and reading achievement are unidirectional (IQ determines reading achievement) and the degree of correlation between IQ and reading achievement are sufficient enough to be able to say that reading achievement can be predicted from IQ when in reality a lot of other research can be used to challenge these assumptions. And therefore IQ might not be the best measurement to detect dyslexia. The arguments between dyslexia and IQ continues to go on and hopefully with more medical advances and research done on the subject, better insight can be obtained.

2.4 Causes leading to dyslexia

The causes which can lead to acquired dyslexia as mentioned earlier are usually related to trauma being received on the brain, therefore the focus with be on developmental dyslexia. The causes for developmental dyslexia are also a very subjective matter as different assumptions of the causes can be argued upon by medical specialists. This generally means that no absolute cause for developmental dyslexia has been found yet. A lot of research have been done on the subject and most of the findings relates to finding out how the brain function of dyslexics differs from a normal person and how

cultures and different languages might affect someone with dyslexia. According to Gavin Reid [11] some factors which might cause dyslexia are, Genetics, Neurobiological (e.g visual and temporal processing, hemispheric symmetry and processing speed) and Phonological.

A book by Gilger [12] found that a male child would have the potential to possess dyslexia up to 40% if his father was dyslexic. The research on genetics are mostly based on heritability of reading sub skills and phonological components. According to Gilger, these research based on genetics are complex as all the research data are from genetic studies. He said that some genetics regions could also affect other aspects of reading and writing skills associated with dyslexia. He concluded that genes does play a part in identifying dyslexia and people could be 'genetically at risk' of dyslexia.

Visual and temporal processing with regards to the Neurobiological aspect, is explained using the report from Singleton [13] which says that visual stress can have an effect on the ability to read and that if it is not handled early in childhood it could lead to poor reading skills in the future. He describes visual stress as the general over-excitation of the visual cortex due to hypersensitivity to contrast or pattern glare. A study by Whiteley and Smith [14] found that around 50% of dyslexics have visual stress. This study can prove that any measures taken to reduce visual stress might be beneficial to dyslexics. Another study by Singleton and Henderson [15] using ViSS, a computer software to screen for visual stress, concluded 41% of dyslexic children are prone to having visual stress while non-dyslexic children was only 23%. Hemispheric symmetry is said to produce difficulty in processing information in dyslexics because of structural differences of the two items in the brain according to Geschwind and Galaburda [16].

Phonological skills is widely regarded as an important aspect of successful reading and many studies have shown that the lack of these skills as a main possible cause for dyslexia. Basically phonological skills are the ability to recognize the sound structure of spoken words. It is often used as a predictor of the reading ability of a child in the future. This is possibly regarded as the most popular cause for dyslexia as studies on phonological skills and dyslexia can be done with better result and credibility compared to other causes. A lot of text-to-speech application are also built focusing on the phonological aspects where the word will be spoken out to the user. These can be

supported by studies from Vellutino et al [17], Ehri [18] and Torgesen [19] which relates how good phonological skills can lead to better reading skills.

2.5 Traditional Treatment for dyslexia

It must be firstly understood that as of now there is no known cure for dyslexia only strategies and techniques to tackle the problem. The way dyslexia affects a person also varies among different cases. For example a person suffering from a serious case of dyslexia might not be able to read even a single sentence in a passage where-else another person suffering from mild dyslexia is not able to recognize just a few words. This is why in order to properly handle the problem of dyslexia, each person must be evaluated personally by an experienced doctor or teacher so that each person can be diagnosed properly as to how serious the disability is affecting them. Here are some traditional treatments method for dyslexia.

Some of the traditional methods for treating dyslexia are Cognitive Behavioural therapy, Self efficacy techniques, Phonic teaching programmes, Reading Recovery system and Multi sensory teaching(e.g. Orton Gillingham methods) . These methods are mostly teaching methods and techniques a dyslexic person can learn to manage dyslexia. It requires teaching in some form and requires time and effort to master. This is most likely the best way to manage dyslexia nowadays although this project is looking for an alternative method to directly help with dyslexics through the usage of visually related solutions and through the use of technology.

According to J Scott [20], Cognitive Behavioural therapy is a problem oriented approach which helps patients to manage dysfunctional thoughts and maladaptive assumptions they have about themselves. This is not a dyslexia specific management handling method, but has also been used to handle other emotionally related problems such as depression as it is a technique which tries to give the patients more insight into their disorder and help them to get specific cognitive and behavioural skills and techniques to manage it. It is more of a method to make the person view the disorder rationally so that preventive steps can be taken against it. Self-efficacy is a technique used to increase the beliefs of one's ability to complete a particular activity according to

Aaron M. Schmidt and Richard P. DeShon [21]. Although this is also not a dyslexia specific handling method but it is useful in helping dyslexics perform better during reading as it is proven to have positive impact on processes and outcomes [22].

Phonic teaching programmes is a method of teaching a person to read based on how the words structure sounds like while it is spoken and relating that to the word. It has existed for a long time and is used to teach not only dyslexics but also normal people. According to a study by Peter J. Hatcher [23], the relationship between phonological elements with the ability to read and spell among dyslexics were very high, in which they were positive effects of training dyslexics using phonological techniques to recognise words. Andrew J. Holliman and Jane Hurry [24] describes Reading Recovery as an intensive one-to-one teaching programme aimed at children with low literacy after going through formal education. Their study show that there is a great improvement in the short term effect of the programme although the long term effect is still debatable. They also said that it is important to have early intervention on a child's reading ability to get improve reading ability in the future.

Multi sensory teaching techniques is when learning is taught through the use of multiple senses of the body, for example instead of just focusing on visual and sound sense which is what most education systems depend on, learning is encouraged through the use of other sense too such as touch(tactile) and movement(kinetics). This technique has been proven to be very useful in accelerating early language development, as shown by the study by Piia M. Bjorn et al [25], where the study show that children who underwent multi sensory type teaching had better cognitive and receptive language skills compare to those which did not have the same type of teaching.

2.6 Research using computer software in relation to dyslexia

Some research and experiment using information technology to treat dyslexia have been done before, mostly using computer software to provide some form of assisted teaching to people with this disability. These are some studies which have shown the use of Information Technology to tackle the problem of dyslexia.

A study from Saptarshi Purkayastha et al [26] uses the Orton-Gillingham approach (a variation of multi sensory teaching) and ICT devices to teach dyslexic students things such as spelling and recognising words using mobile tablets. They prepared a mobile application called Dyscover on a touch screen table. One function from the application allowed student to write words using their fingers so that they have a better physical sense as to how certain words should feel like. This function triggers multi sensory learning by utilising auditory, visual and kinesthetic elements together. Their view is that ICT can be seen as a potential tutor to children with dyslexia and that children with dyslexia must be evaluated in a holistic way.

Using a combination of specially trained teachers and computer programs were the focus of the study by Joseph K. Torgesen at el [27]. They used computer assisted instructions to prevent early reading disabilities for students at risk of dyslexia. The software used was built by Dr. Jeanine Herron (1995) and focused on phonetic spelling and writing. It uses colourful animation and digitized speech plus practices on phonological awareness, letter-sound corresponded and phonemic decoding. What they found was that the students which received teaching from the combined teaching method, at the end of first grade, performed better than regular students going through the school's normal reading program.

A spelling training software which could turn words into codes was the focus of a study by M. Kast et al [28]. They manage to create a software which could turn word into a multi-sensory representations comprising of visual and auditory codes (e.g. colours and shapes which represent individual letter's information). But for this study an enhanced version of their software was developed to include phonological codes and a word selection controller based on phoneme-based model. It also highlight's the importance of colour to dyslexics children, by associating different colours to similar letters such as b and d. Their findings were that children with dyslexia benefited greatly from the additional phonological codes, and actually manage to improve their spelling skills to the same extent as children without dyslexia with the right support and training.

2.7 Available tools to help dyslexics

Here are a list of currently existing tools which could help adults with dyslexia use mobile devices and applications better. This will include both physical tools and

software related tools. Each will be reviewed and their main functions highlighted. A conclusion based on all the reviews will be presented at the end.

2.7.1 Physical tools (Colour overlays and rulers)

There exist a number of physical tools related to helping dyslexics. But the one product which is most commonly used and associated with dyslexia is colour overlays or colour rulers. These items are transparent pieces of plastic paper which are coloured differently. They exist usually in 2 shapes, which are a whole A4 size paper to cover the whole page of a book, or a ruler which highlight each sentence on a text. These items are proven to help a percentage of dyslexics to read better as shown by the research done in section 2.8 of the report. These items exist as the most visually based tools to help dyslexics and its existence alone is prove on how different colours background on text could significantly help dyslexics read. Below are images of existing coloured overlay and rulers currently sold in the market.

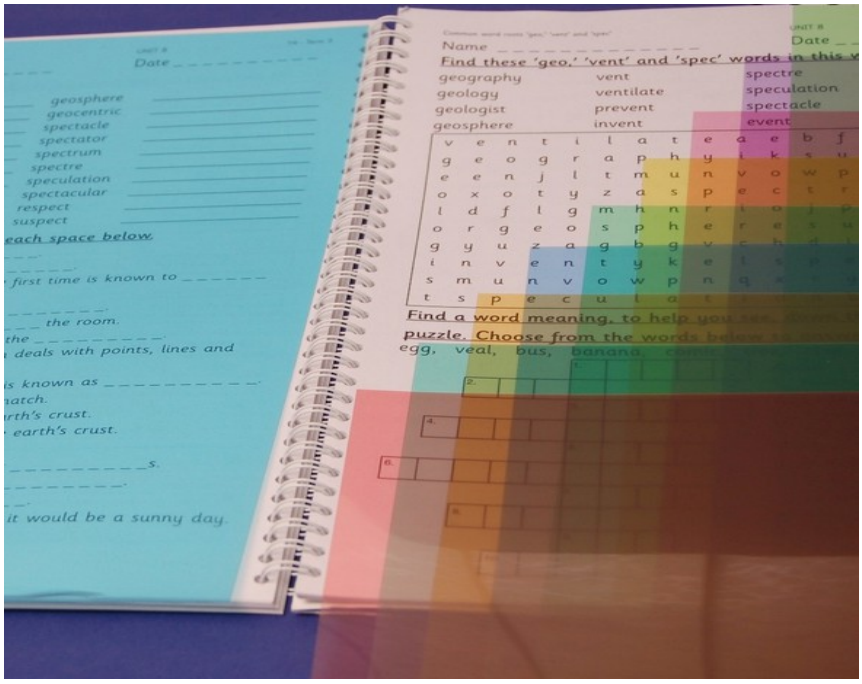

Figure 1: Coloured overlay which covers a whole A4 size page being sold at The Dyslexia Shop UK[29]

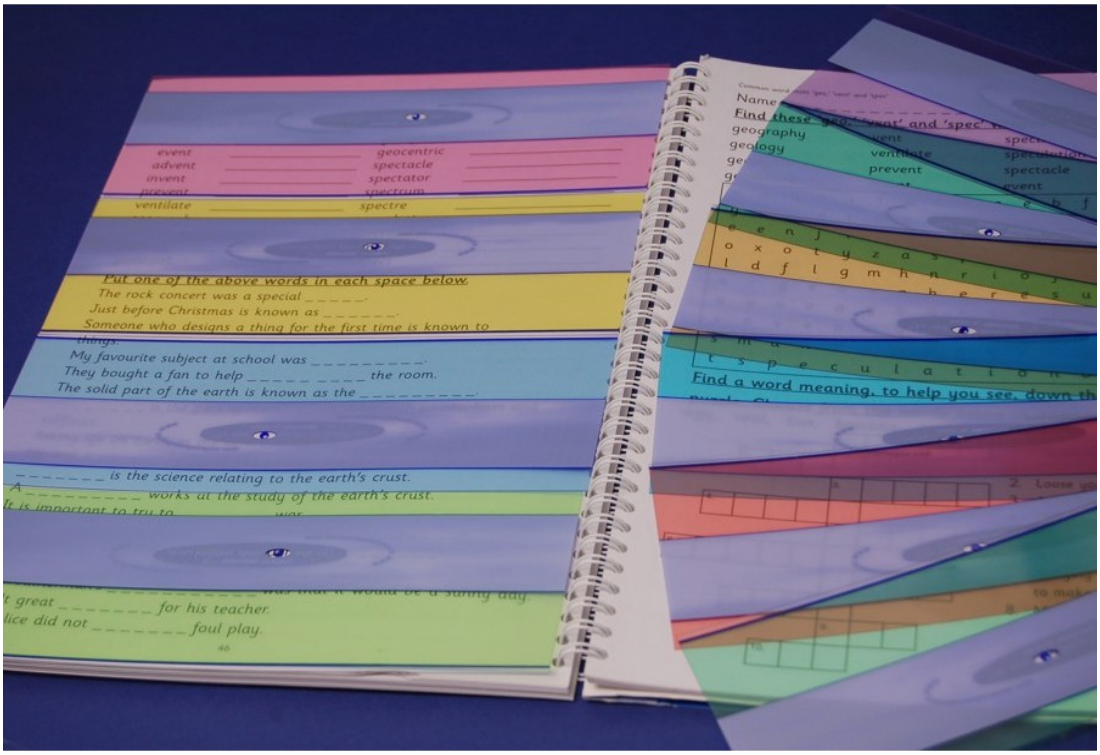

Figure 2: Coloured rulers which highlight each sentence on a page, also sold at The Dyslexia Shop UK[30]

2.7.2 Fonts for dyslexics

A special type of font for dyslexics has been created by OpenDyslexic.org [31]. Based on the makers of the font, it is built to help dyslexics read better. Some elements such as heavy weighted bottoms is applied to the font to give the reader a sense of direction for the letter. Based on the creator, this helps the reader recognize which part of the letter is down and can help to keep the brain from rotating it around. The effectiveness of this font is yet to be proven by any academic research, but it is widely used in other dyslexics assisting software (e.g. Readability). The font might be use to treat Dysgraphia, where one symptom is, it becomes hard to recognise between similarly looking letters such as “b” and “d”. A lot more research and experiments must be done to see its effectiveness because dyslexia affects different people differently. Below are images of the OpenDyxlexic font being compared with other font types.

Figure 3: Comparison of different font. Line 1 Arial, Line 2 Times New Roman, Line 3 OpenDyslexic.

2.7.3 Desktop Screen based Software

2.7.3.1 ClaroRead(Windows Version)

ClaroRead is a software produced by a company called Claro Software [32]. Its main function is to help dyslexics have better accessibility, in term of reading, recognising and typing in words, when using computer based operation system such as Windows and Mac. To use it users need to purchase and install the software from Claro Software. Prices range from £95-£250 (ex. VAT) depending on which version is bought. The way ClaroRead functions is through the usage of a series of tools to help users in the form of a floating toolbar window. The main tool is the ability to highlight any text on screen on any software application or website and then by pressing the "play" button on the toolbar, hear a localized voice read out the text (e.g. Text-to-speech).

It can also highlight all the homophones (e.g. words that sound the same but are spelt differently) on the text. It also has a prediction box while typing in word editing software (e.g. Microsoft word). Text can also be saved as an audio file so that it can be listened to later. Some additional functions of ClaroRead includes, a ScreenRuler function which highlights only a part of the screen for more focused text reading and a function called ClaroView which tints the computer screen using different colours. The application this project aims to build will also incorporate both of these functionalities but on a mobile screen. Some screen shots of the software are included below.

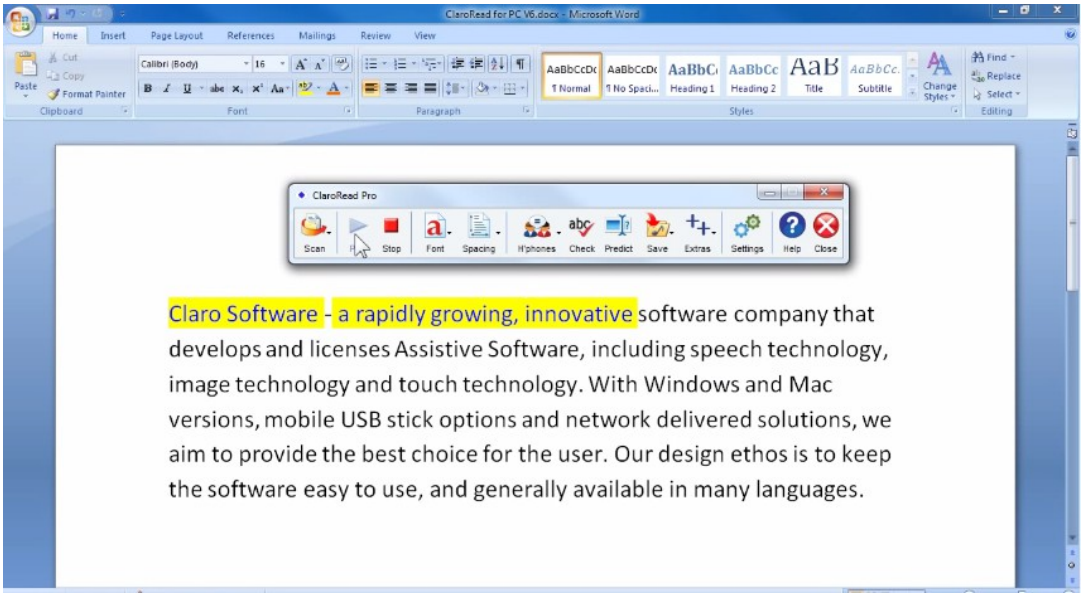

Figure 4: ClaroRead toolbar, where the highlighted text is being read out in a localise voice

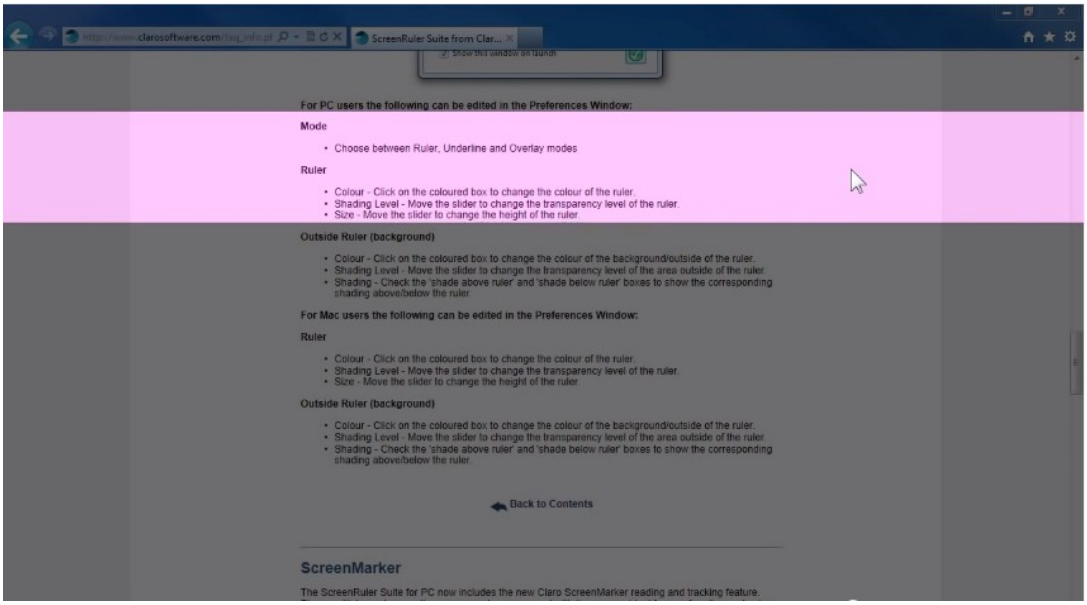

Figure 5: The ScreenRuler function which highlights only a part of the screen so that the text on the highlighted area can be focus on.

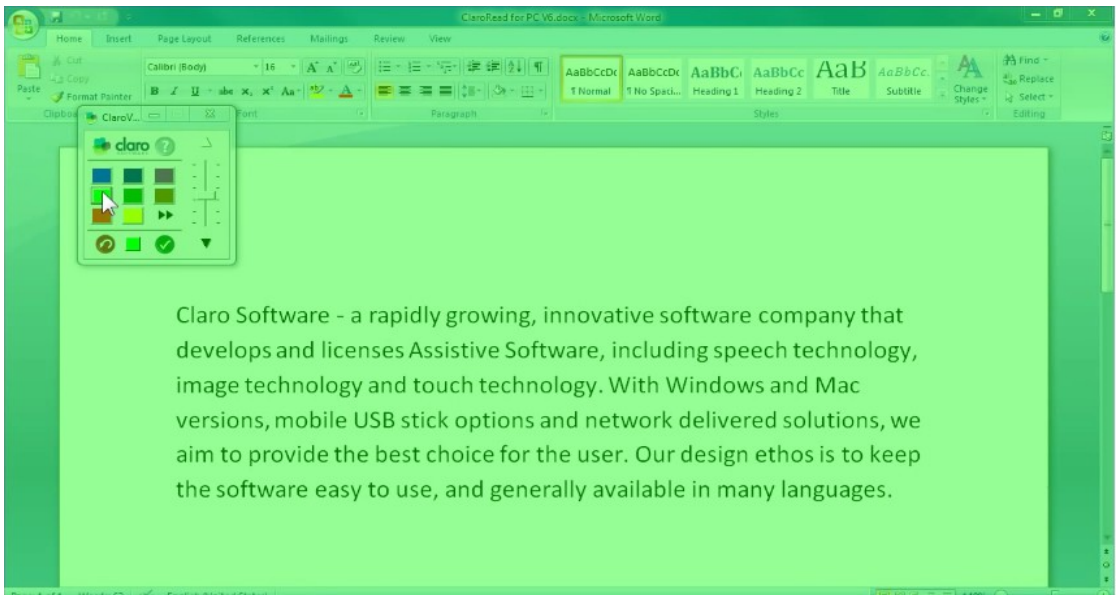

Figure 6: ClaroView function which can tint the computer screen to different colour overlays.

2.7.3.2 Readability Desktop Browser Version

Readability is a web based application created and designed by the Readability team [33]. It started out as a Javascript-based reading tool, created by arc90 [34], that could turn any web page into a customisable reading view. The way Readability works is, firstly a user needs to register for a free account at their website www.readability.com. The user will then be given their account page where they can add items to their reading list. After that users can provide URLs of web pages they want to add to their reading list. All the items on the user' reading list will be customised by Readability. Some of the customisation available once a web page enters a user's reading list includes, the ability to throw away all the styling on the web page so only the text and images remain (done automatically by Readability), resize and change all the fonts on the web page, invert the background colour of the web page (e.g. usually from white to black and vice versa), and increase the spacing between the words.

No installation of software is required but users have the option to install extensions on browsers so that web pages can be put directly into their account reading list. Some downside to Readability is it does not support popular social media applications such as

Facebook although some form of connection is available such as posting your favourites list on Readability on Facebook using the usage of IFTTT recipes [35]. Readability does not market itself as a tool to help Dyslexics but some functions such as inverting background colour will definitely help with visual stress. Almost all functions, such as menu bars using Javascript or Flash, will also be discarded by Readability, reducing the functionality of the customised web page. Here are some screen shots of Readability.

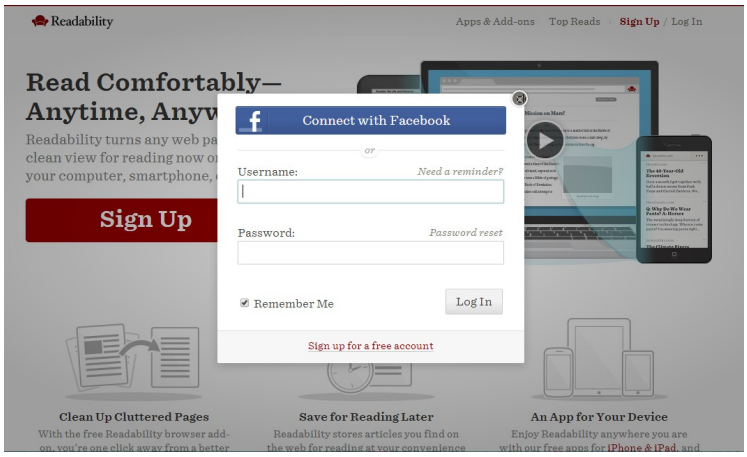

Figure 7: Readability main page and signing in.

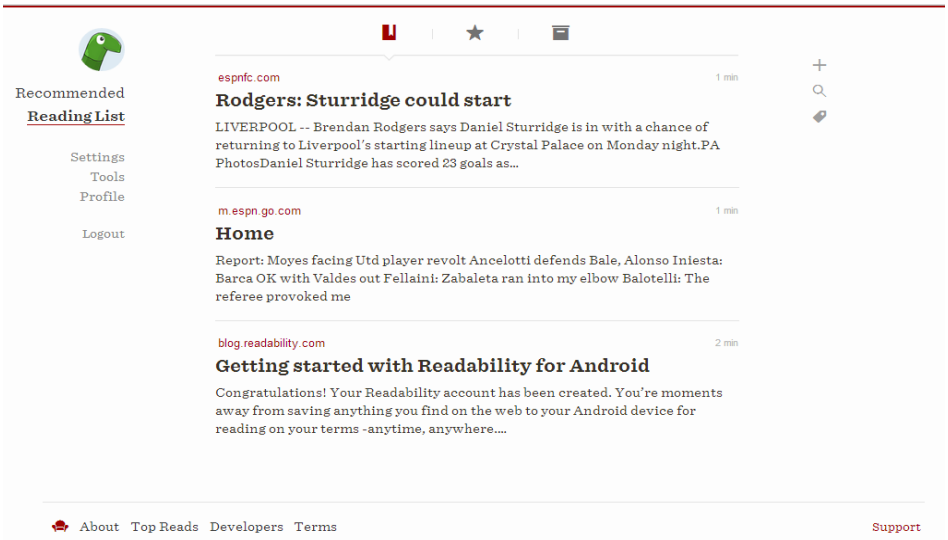

Figure 8: Reading list on Readability, all web page URLs are kept here.

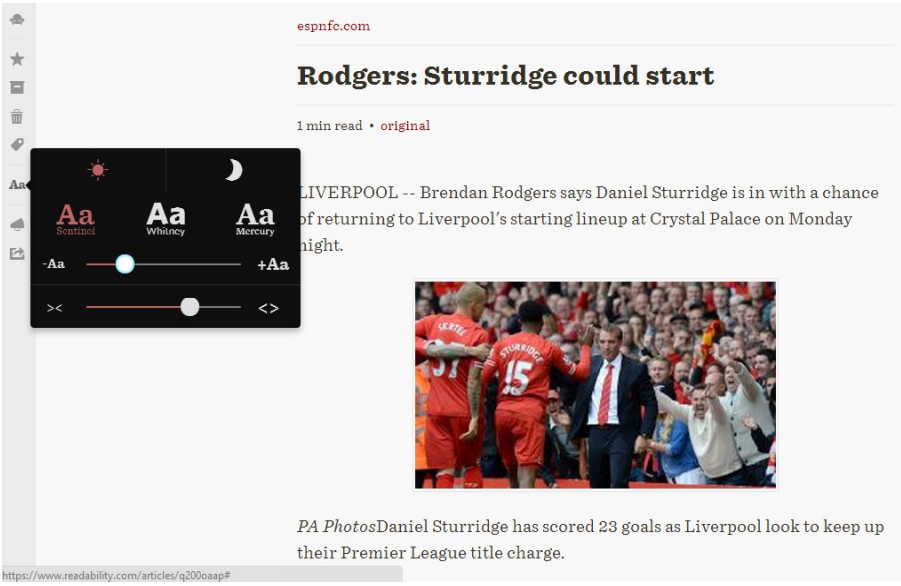

Figure 9: Example of web page being customise by Readability, and tool bar on the left side. The ability to invert background colour, resize/change font and spacing is on the tool bar.

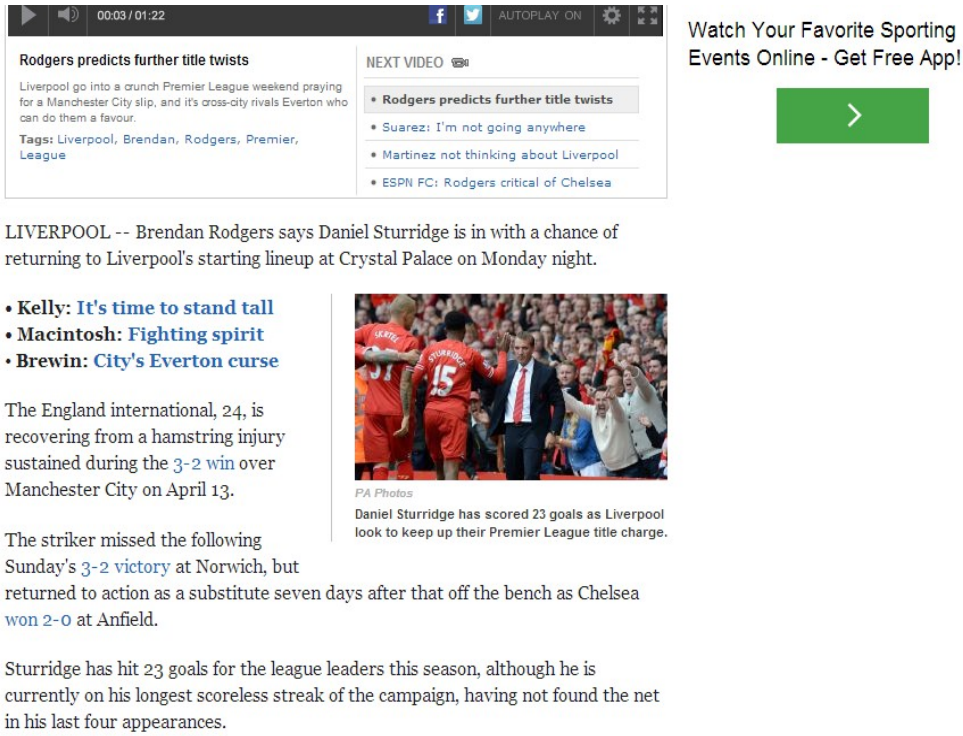

Figure 10: Original web page from the Diagram 9. A lot of styling have been discarded.

2.7.3.3 Read&Write Gold

This software is build by TextHelp Ltd [36] and is currently priced at £320 for the PC/MAC version. There are a lot of functions available on this software and most of them are similar to what is offered by ClaroRead but in comparison Read&Write seems to offer more in functionality. The application is highly customizable but one of its weaknesses is that it seems as though a lot of training is required to use the software properly as it was intended. Here are some of the functions available. The ability to read out text using text-to-speech technology is the first one available as is a standard among almost all software built towards helping dyslexics. This can be done on text documents or on websites. It also has a function to read text from an image making inaccessible text (e.g. cannot be process by a computer software directly in its raw form, although readable by human eye) readable. This is helpful in reading PDFs or advisements in the form of images. It can also change the text in sound files so that it can be carried and listened to any time using appropriate devices.

It also has word related functions such as a spell checker, a text predictor, and a dictionary. One of its more innovative functions is a picture dictionary which will produce an image of the word being checked. This provides more visual results for the users and hopefully will make it easier for the user to understand the meaning of new words. It also has a function to filter out homophones called Sound Like & Confusable words facility. This could help prevent words with similar sounds getting mixed up. A verb checker is also available to ensure the language and grammar being used is correct. Finally it also has a function called Screen Masking which essentially provides an overlay for the text being read where the users can adjust the Screen Masking type and colour. The application being built will try to incorporate this function but on a mobile device.

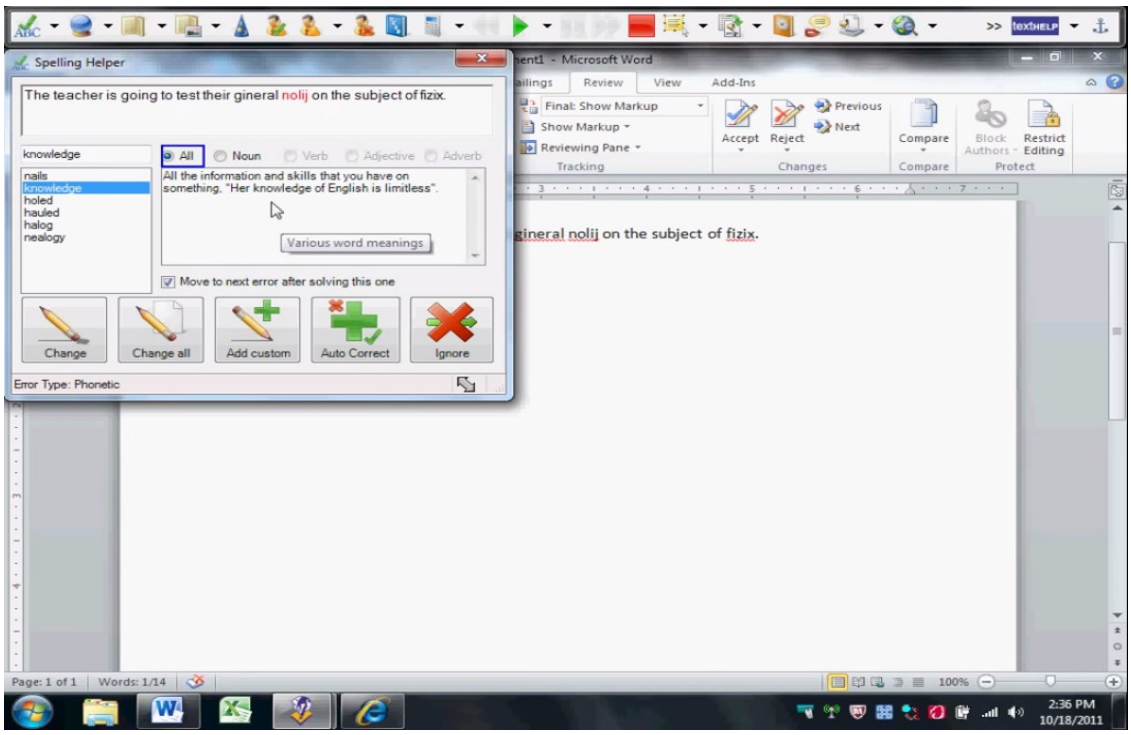

Figure 11: Screen shot showing the spell check function on the software

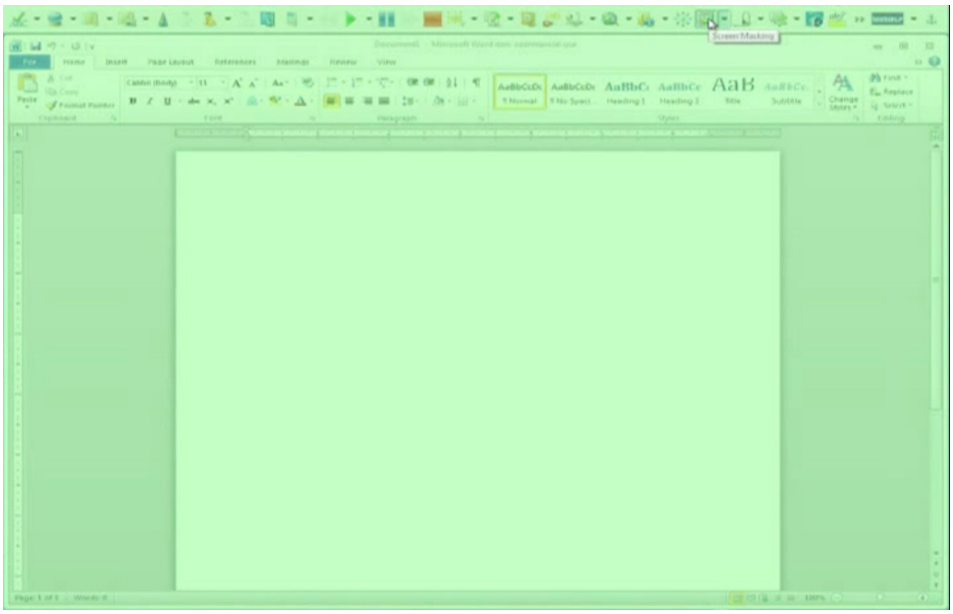

Figure 12: Screen shot showing the Screen Masking function of the application. The colour overlaying the screen can be changed by the user.

2.7.4 Mobile Screen based applications

2.7.4.1 Readability App (Android Version)

This mobile application is based on the Readability desktop web application reviewed earlier. The creators are the same and most functions available on the desktop version are also available here. Reading lists on the desktop version will be automatically synced with this version of Readability and vice versa. For the mobile version, users need to install the Readability app from Google Play. The option to use Readability on a mobile web browser is also possible but it is recommended to use the Readability app due to better design and functionality. Both the desktop and mobile version of Readability lack the function to provide different colour overlays, and also the ability to focus only on certain part of the screen. Here are screen shots of the app.

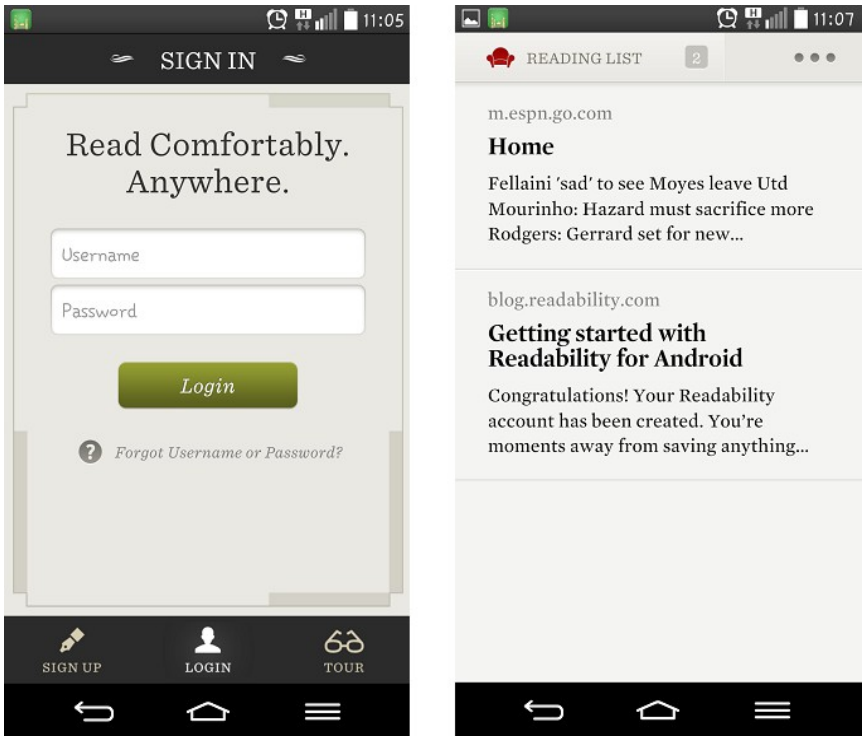

Figure 13: Log in page and Reading List for Readability app

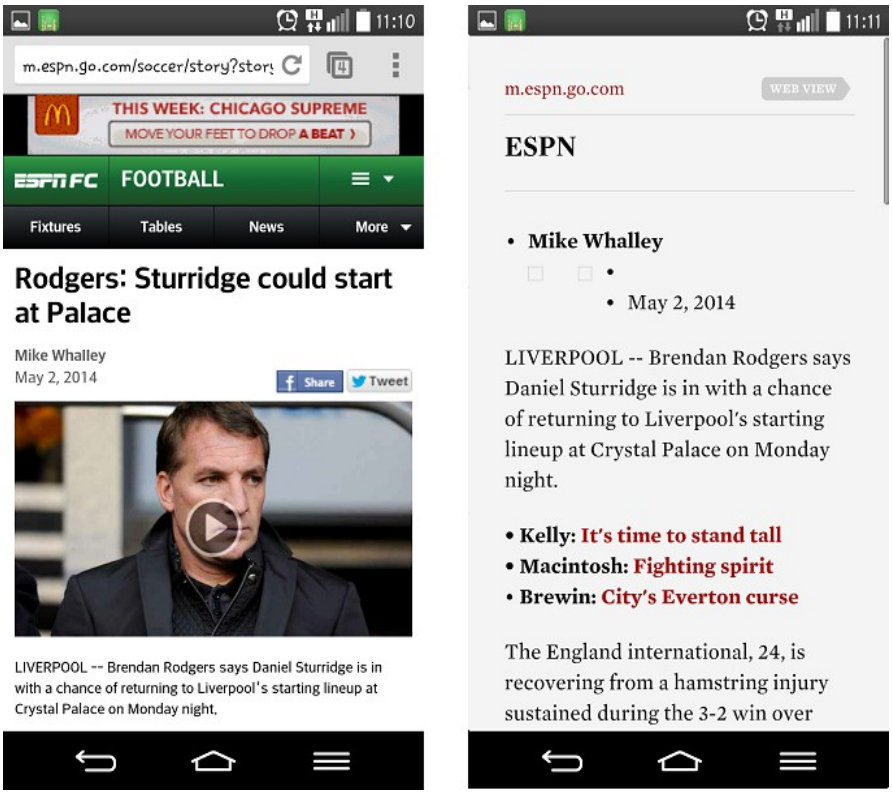

Figure 14: Comparison between normal mobile web page and customised Readability web page

2.7.4.2 Mobile112 App(Android Version)

Mobile112 App [37] is an app developed by Flemming Ast and Bill Bostock [38]. The app costs around £33 but a trial version is available for 30 days. The app functionality includes, hearing out SMS received on a mobile device, speaking out words which are typed into its editor, scan images of text document and reading them out, and also a web browser which can read aloud any web page. The list of functions that the app has is very good, but due to the design of the app which is very technical it will take time to get used to. Here are screen shots of the app.

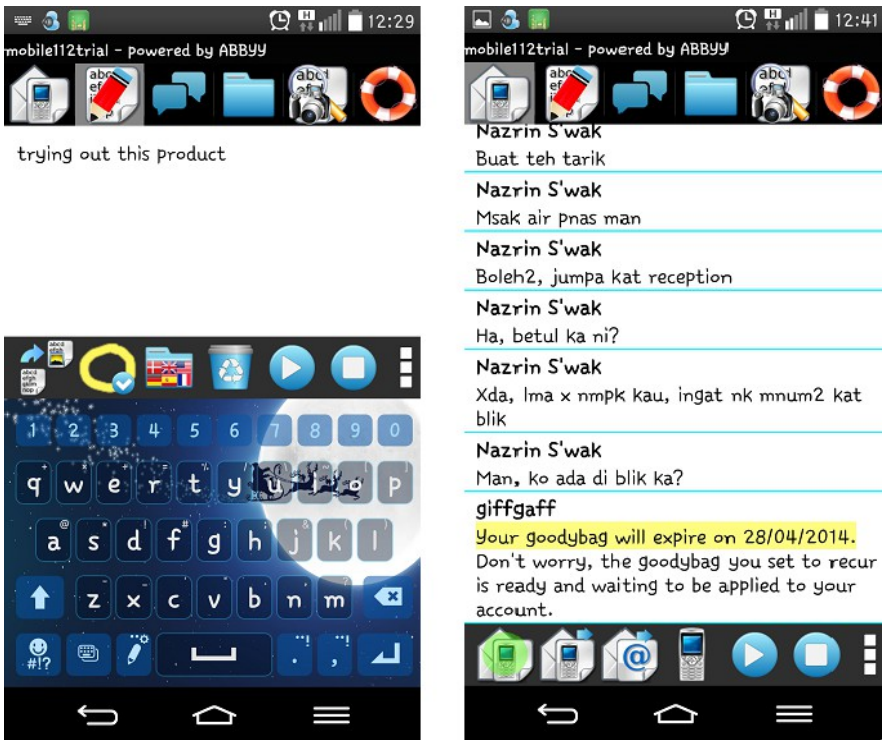

Figure 15: Left: The text editing tool which will read out any text written. Right: The app will scan the mobile device for received SMS and read them out.

Figure 16: Using the app web browser to read out text in webpages.

2.7.4.3 iWordQ US(iOS Version)

The iWordQ app[39] is a mobile app created for the iOS by Quillsoft Ltd[40]. The app costs around $25. Some of its features include, word prediction as a selection of words are spelt out during the typing of words, speech feedback where the user can touch and hold on a word and it will be spoken out (can also be used to proofread a sentence), Text Chunking function which basically helps to break sentences to smaller segments before reading each segment out to the user (customisable pause time, reading speed and line spacing) and finally Abbreviation-Expansion which allows user to simplify their writing. The app does not have any back ground colour manipulation function. The interface of the app is very clean and is very easy to use. But it does not have any function which allow it to be used on web pages although it has the ability to email what the user has written down. Some screen shots are provided below.

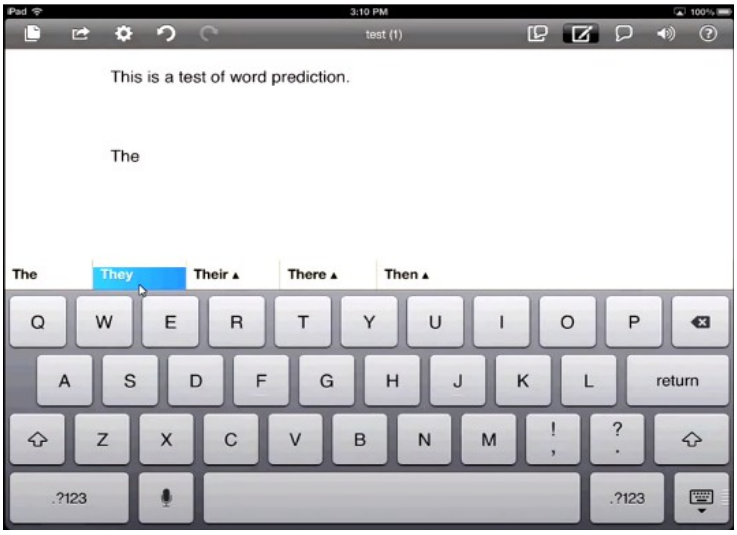

Figure 17: Editor function of the app where words are predicted on top of the keyboard. Touch any of the words and they will be spoken out.

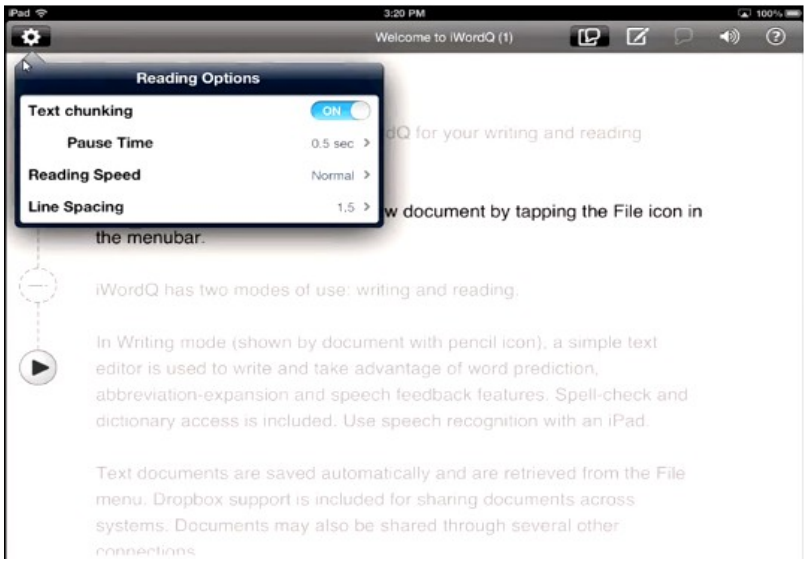

Figure 18: Text Chunking function which will make whole texts into chunks and some customisable elements as show on the drop down settings menu.

2.7.4.4 Dyseggxia

Dyseggxia [41] was created by Clara Bayarri, Luz Rello and Azuki in 2012. It is essentially a mobile game which helps children with dyslexia get over their reading/writing problems in Spanish by using fun word games. According to their website, all the games are designed to target specific reading/writing errors by analysing real errors in Spanish children's essays. The games are built based on five tasks that the children playing them need to do. They are Insertion(Insert missing letter from available answers), Omission(Extra letter in the word must be deleted), Substitution(A wrong letter in the word must be substituted with the correct one), Derivation(User must choose the correct ending to a word from a list of choices) and Sentence separation(Breaking up several words put together). Overall the application uses the findings from real research done on dyslexia and how it can create problems in spelling to create a fun and helpful game to help children spell better. This type of application make it easier for children to learn in a very fun and entertaining manner.

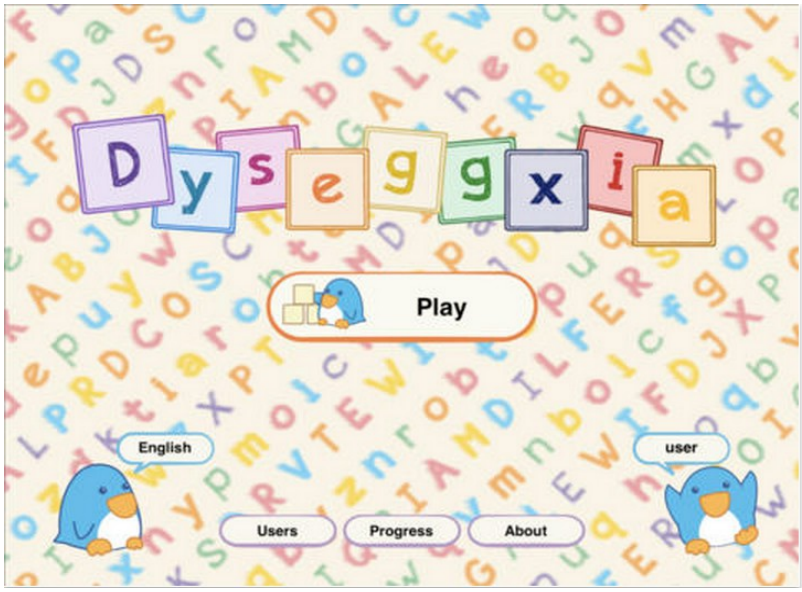

Figure 19: Main screen of the application. It has a very child friendly user interface which make it more accessible to children.

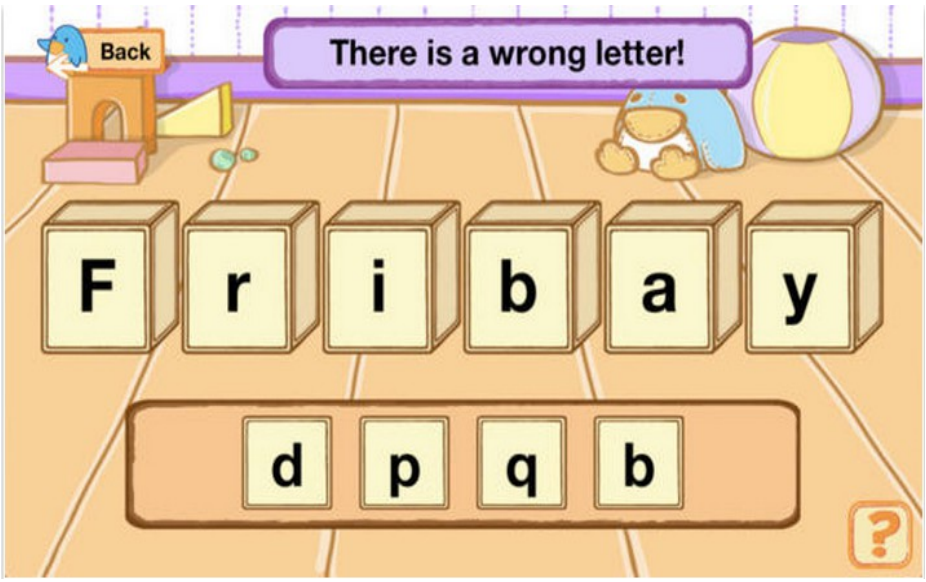

Figure 20: An example of the one of the games being played. The objective is to change a letter in the word to make it correct.

2.7.5 Conclusion of reviews

Here are the findings from the physical tools, desktop software and mobile application reviews. Firstly most of the products provide support based on text-to-speech recognition technology where text will be spoken out. Text-to-speech recognition might prove useful in most cases but in the case of using it to read SMS or simple messages where abbreviations are regularly used, it does not function as well. Secondly font manipulation where font type and font sizes are changed is common among all these software/applications. But not enough effort is being put into the usage of colours, and on mobile applications there is almost no support on different colour overlays much like the ones available physically, which can help reduce visual stress. The ClaroView function from ClaroRead which is on a desktop computer is the one of the few functions which try to incorporate colours into its product. This function mimics the use of coloured rulers (as mention in section 2.7.1). The only function available next to that is to invert the colours of text and background colour. There are only a small number of applications which serve to add more accessibility to popular social media sites such as Facebook, Instagram and Twitter. The users of social media websites are usually restricted to the level of accessibility provided by the developers of the sites only.

Other findings were, firstly a lot more applications which deal with accessibility exist on the iOS compared to the Android platform. This is probably because there is more support from Apple to create more accessible apps compared with Google(Android's principal owner). IOS users are also more willing to purchase apps for a fee compared to Android apps on Google Play which are mostly free. Secondly some of these applications are quite expensive maybe because the market is not big, so the developers need to have sufficient profit margin on each purchase so that they can keep developing high quality product. This makes it inaccessible to a lot of people causing its true effectiveness hard to measure because it is not being used in a large scale. Finally the ease of use factor involving the software provided. For example, although Read&Write Gold is full of functions which may prove useful to dyslexics, it has a guide of over 40 pages explaining each and every function. This could prove to be very difficult for a

dyslexic to read and go through. ClaroRead by contrast has fewer functions but emphasises ease of learning and use. The application should be designed to work easily and instinctively with what the user wants. Ease of use and simplicity in design will be focused on during the building of the application, to ensure the target users will have no problem in understanding how it works and ultimately using it.

2.8 Studies relating overlays with dyslexia

After reviewing the background research, it was noticed that most visual solutions to manage dyslexia had to do with colour or more specifically providing different backgrounds for texts. So further research was done to find out if the use of coloured overlays which could provide a different background for text could be useful to dyslexics. Here was what the research found out. Firstly according to a statistic presented earlier in the report, around 50% of dyslexics suffer from visual stress. Therefore, dyslexia is heavily related to visual stress and there is research which shows that the use of tinted/coloured lenses can decrease the effect of visual stress on a person.

This connection could be used as a basis to support this relationship. An article by Gary J. William et al [42] states that the use of colour overlays can have a positive effect on reading ability and visual perception. It also has the ability to minimise reading discomforts such as light sensitivity, eye strain and blurring of text/print. This ability can become very useful during the usage of mobile devices with its LCD screen which may cause discomforts on prolong usage. Another study by Anita J. Simmers et al [43] which tests the effect of tinted lenses for relief of visual discomforts and visual stress found that for some people the use of tinted lenses made their eyes more accommodative towards visual stress.

Wilkins A. J. et al [44] also states that tinted lenses are beneficial to reduce perceptual distortion for dyslexics. Generally people who are dyslexics will also have visual discomfort when viewing stimuli such as text. This is sometimes accredited to their low level of tolerance towards visual stress. The application which will be built would have tinted/coloured overlays which might work to help dyslexics read better. A study by Peter M. Allen et al [45] also found that individuals who were more prone to have

pattern-related visual stress had improve reading speed with the usage of coloured overlays. The same study also suggest that the use of coloured filters resulted in an increase of voluntary accommodative response of the eyes which translate to better comfort for the eyes thus allowing for better reading and recognising of text.

An experiment by Roger Hall et al [46] which test the use of a double colour filter on 73 delayed children readers found that reading improved significantly after the 3 month trial period with 40% of the children improving their reading age by 6 months or more. Wilkins A. J. et al [47] has ran a Rate of Reading Test to assessed the effects of coloured overlays on reading. The findings were those who chose to use coloured overlays saw an increase in reading speed where they saw an increase from 97 words a minute to 104 words on the first test. All these research supports the theory of using coloured overlays/lenses to reduce visual stress for people with disabilities in reading and thus improving the reading rate of the person. This allows them to read better and faster.

The next section, research methods, will go through how the application was conceived based on the background research done and how they are related. The methods used to evaluate the effectiveness of the application will also be explained. It will also touch on the process of building a mobile application on an android platform.

3 Research Methods

3.1 Introduction

This part of the report presents how the built application will help dyslexics, and what methods are used to measure its effectiveness. It will basically touch on how the application relates to the background research. It will also touch on the research methods as to how to actually start building a mobile application from scratch and provide best practices for building Android applications. The final version of the application (in regards to design and functionality) was conceived based on the background study of dyslexia and available applications and tools connected to it.

3.2 Building a mobile Android Application for Dyslexics

After doing extensive research on dyslexia and available mobile applications for it, it was determined that applications for this disability are very little in amount, and most of them (on mobile platforms) are not actual tools for dyslexics but rather just things to help dyslexics learn better and mostly to explain to people what dyslexia is. A simple and easy to use idea for an application to help dyslexics has come forth from the research which is one which tackles the visual aspect of the disability. The application will target this aspect due to various reasons. The main reason would be because most applications available on mobile focuses on audio elements from the user, such as reading out aloud text and changing speech to text software.

Not much emphasis has been given towards visual solutions. This application could prove a step towards the creation of better visually centred solutions for dyslexics rather than just audio based solutions on a mobile device which would prove hard and awkward to use in places with a lot of people and noise such as train and bus stations. The application will specifically target dyslexics tendencies and reaction towards the use of colour overlays and how it could have a significant effect on their ability to read and focus to text on screen. Android was chosen as the platform of choice mainly due to the reason that is it the mobile device available for use and experiment, and the library

provided by LG to allow for floatable windows would as of now only work with an LG Android phone.

3.2.1 Dyslexics and colour overlays tools

As shown in the previous section, a lot of studies have been done to show the relationship between reading and how the use of colour overlays could improve them. Tests were done using transparent coloured rulers which act as colour overlays during reading and most of the results were positive and showed how it could help in reading and focus. And as of now there has not been a simple solution for this problem on a mobile application. Dyslexics need to have something simple to use because most applications provided today, although very plentiful in function, can sometimes be very difficult to learn and use not to mention expensive.

Dyslexics need something simple and easy with straight forward functionality and with minimal training in its usage. It is based on these reasons that the idea for the application was conceived. The application is built to simulate a resizeable colour overlay on the screen of a mobile application. It can be adjusted to fit the whole screen or resized to resemble a ruler. It may be simple in nature but the reality is no one has provided a mobile application for this specific function before. The application must be simple to use so that dyslexics can learn to use it fast and effectively. The application will ultimately provide an alternative background colour for the text on the screen which is usually black coloured text on white background.

3.2.2 Evaluation of the application

It is very important that the application be evaluated by people suffering from dyslexia. This is because normal people would not be able to see the world from their eyes and experience their difficulties. To properly assess the application, it must be used by volunteers suffering from dyslexia on a mobile device and their feedback and suggestions must be properly obtained. It is best to do the testing in small groups or one-to-one as feedback and suggestions could be more freely expressed. The evaluation being done currently is a qualitative one. A written questionnaire would also be provided with very simple questions as to not make it difficult for the dyslexics to read.

A sample of the questionnaire will be provided in the appendix printed using a special font mentioned earlier to make it easier for dyslexics to read. The testing and evaluation chapter will explain how these volunteers were recognized and approached. A small number of normal people with no history of dyslexia will also be approached to try out the application and act as a control group.

In the future in which more time was available, the current evaluation style could be improved and a more detailed experiment could take place. That evaluation method would be something like this, each test user will be asked to read through a body of text and their time will be recorded. Then it will then be repeated on a different body of text but this time with the help of the developed mobile application. Their times will be measured and compared to see if the application has improved their reading times or not. Meanwhile a control group consisting of normal people will be asked to do the same experiments and their time will also be recorded. This is to provide a comparison to see if the application might actually help normal people read faster too. Though realistically this type of evaluation is difficult to carry out effectively and would require more time.

3.3 Android Mobile application development

The application for this project called Touch Colour Overlay (TCO) will be build on the Android platform from scratch using libraries provided by Google and third party developers (e.g. LG). The following sub-chapters will give a detailed overview regarding the methods and tools required to create an android application and the workings of its inner structures. It is important to get an overview of the Android system as a whole before beginning the development process.

3.3.1 Android Environment and Tools set-up

According to the Android Developer website [48], the processes and work flow involved in producing an android application are as follows. First is the set-up of the development environment where all the tools are installed. These tools includes the Android SDK, Android Development Tools and Android Platforms. Eclipse is also the

preferred IDE to develop Android application due to its ability to invoke directly the tools needed for development although other IDE can also be used. The next step would be to set-up an Android Virtual Device (AVD) using an Android Emulator [49], which would allowed the simulation of an Android application on a desktop pc. This can be done using Eclipse AVD Manager or from the command line. It is also important to set up an actual hardware device (e.g. smart phones running Android OS) to run the application once it has been developed [50]. This can be done using the tools already in the SDK kit provided by Android or using Eclipse to directly installed it onto a mobile device. But it must be kept in mind that applications must also be tested on the emulator on different settings and configuration (e.g. different Android versions, varying screen sizes).

Next would be to set-up Android projects (e.g. .apk files) which would act as containers to place codes and resource files. A specific structure must be followed due to constraints from the SDK tools. Some other different types of project which would support the Android development process includes Test projects(test application codes) and Library projects(shareable Android source code to be referenced in Android projects). Some things to consider during this stage includes resource conflicts and using prefixes to avoid them, library projects can include JAR libraries, the platform version must be equal to Android project and finally library projects cannot include raw assets[51]. After the application has been coded it can then be debugged and tested. The Android project is firstly built into a debuggable .apk package which can be installed and run on an Android emulator or device. Then the application must be debugged using a JDWP-compliant debugger that is used together with debugging and logging tools which are provided by Android SDK. The application can then be tested on an integrated testing framework already provided by the Android framework [52]. The diagram below from the Android Developer website can summarise the process.

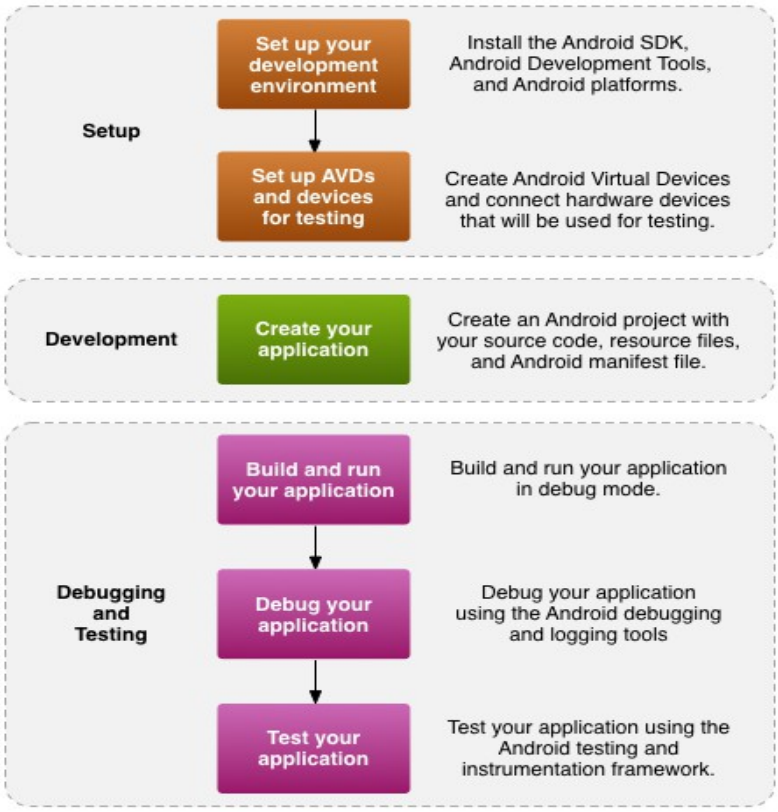

Figure 21: A summarised diagram of the Android Development Process

3.3.2 Android Architecture

The Android system architecture can be explained using the diagram below from the Android Official Developer website [53].

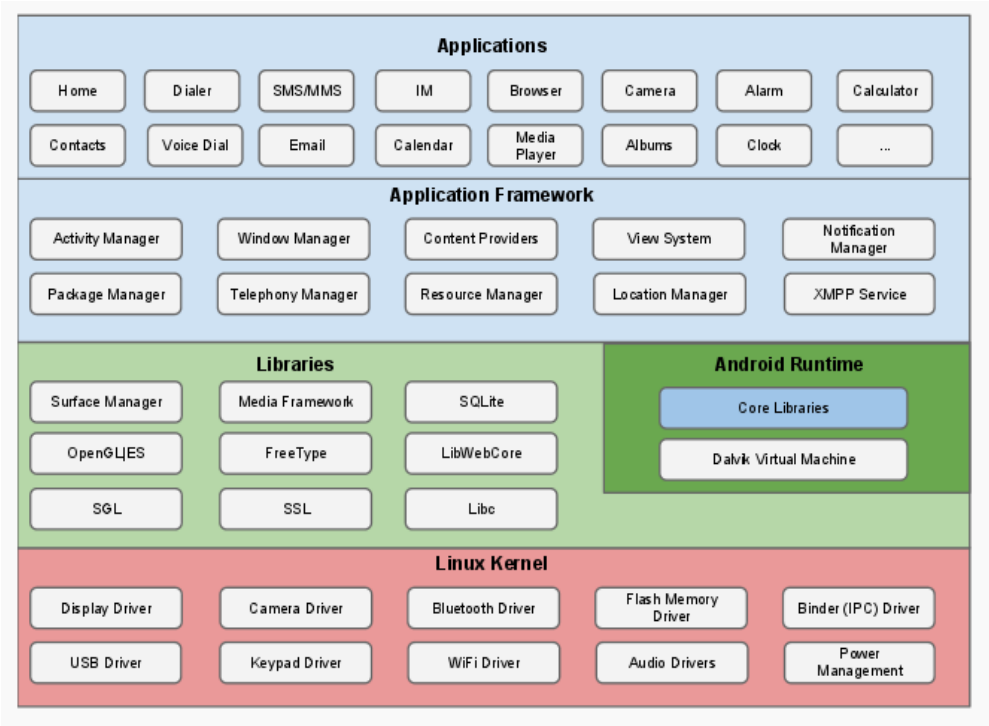

Figure 22: Android System Architecture

A tutorial video from Android Developer [54] described the architecture as follows starting from the kernel layer. The Linux kernel acts as a hardware abstraction layer. Linux is used because it has a lot core operating system infrastructure such as proven driver model, memory management, process management and etc. The next layer is the libraries layer where everything here is coded in C and C++. The core power of the Android platform comes from here. For example the surface manager is the item which controls every pixel which appears on the screen and manages the actual drawing of different windows of applications. OpenGLES (3D graphics) and SGL (Scene Graph Library, 2D graphics) make up the core of Android's graphic libraries. The media framework holds the codec for the usage of media on Android. Next is Android's runtime layer where the main component is the Dalvik virtual machine.

It is designed to run in an embedded environment with limited usage of batteries, memory and CPU. Its speciality is the ability to have multiple instances of the virtual machine running at the same time in different processes. On top of that is the Core Libraries which are written in Java. All the collection classes such as utilities, and IO come from here. The next layer is the Application framework. All the classes here are

also coded in Java and contains the tool kit which is used by all applications. All applications use the same framework and same APIs. Some of their component includes, the Activity Manager which manages the life cycle of the applications.

The Package Manager keeps track of all installed applications on the device. The Window Manager handles all windows and Telephony Manager contains the phone application APIs. The Content Providers allow different applications to share data together, for example contacts which can be accessed by other applications so that they may use the information. The View System contains the building blocks for the UI(e.g. buttons, lists, layouts) of the system. The final layer which is the Applications layer is where all the applications get written including the application this project aims to build. It also uses the same application framework provided by the layers below.

3.3.3 Android Application building blocks

The Application building block of Android is build out of four major components which are Activities, Services, Broadcast receiver and Content provider [55]. Activities [56] are basically user interfaces which allow for users' interactions. One activity usually just represents one single screen of an interface. One application usually consists of multiple activities working together. For example a calendar application might have one activity to display all written events while another activity displays the form to actually input information when adding the events. A different application can also open up an activity from another application. For example opening up email activity for sending email using the camera after taking a picture to send the picture to someone. Services [57] are basically tasks which run in the background of the system. They handles long-running applications and remote processes. They do not have a user interface and are usually invoked by an activity.

An example for the usage of a service would be to say when an activity opens up a music player and then starts playing music, but then the activity decides to open up another activity or application, the service is what keeps the music playing in the background even though a different application has been opened. It can be asked to stop by binding an activity to it to send Intents (messages) for it to stop playing music. The Broadcast receiver [58] responds to any changes to the system being broadcast to it with the use of Intents. It can wake up any processes it is assigned to. Basically it

registers codes which would not be running until an external element awakes it. For example, a Broadcast receiver would open up a list of network connections once it senses an available network to connect to. It also does not have any UI. And finally the Content Providers [59] allows data sharing between applications by accessing any file system or database (e.g. SQLite) on the Android system . It is also used to write data private to your application and not shared. The diagram below represents how the different component interact together. The application being built also works on the same principles.

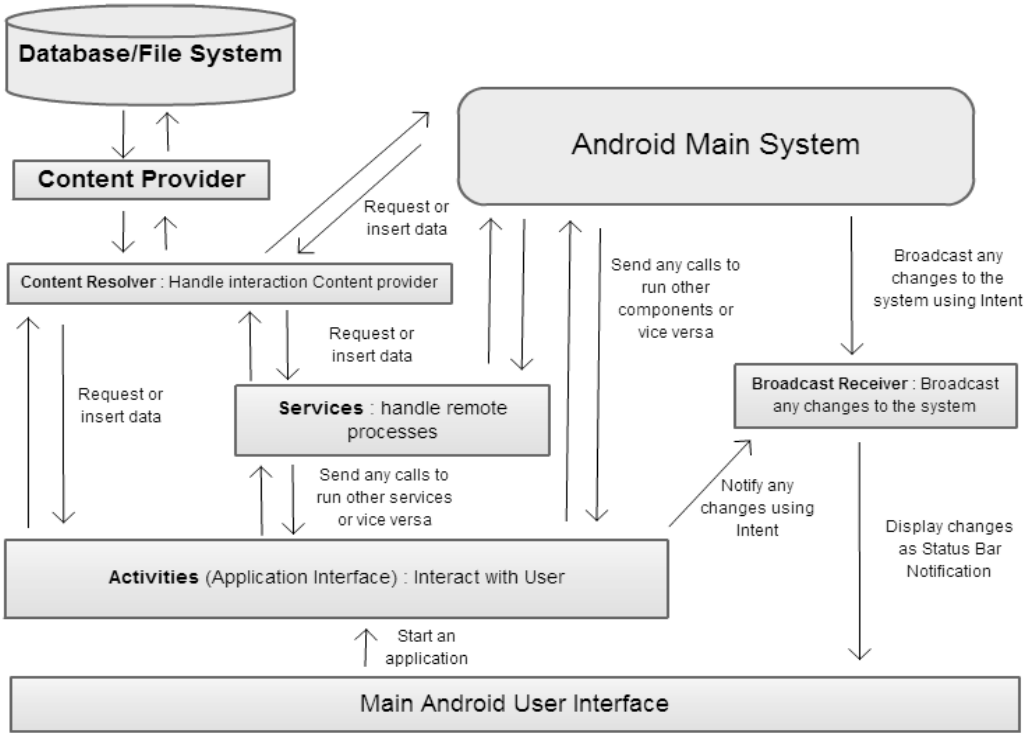

Figure 23: Interaction between main components for Android

3.3.4 Additional SDK/Tools for Android development (QSlide SDK)

This new software development kit from LG allows an application to be put into a minimised window on a mobile screen, so that it can be simultaneously open with another application. This allows the minimised application to overlay with an already opened application to create the illusion of a lens on top of the application. The size of

the window in which the application is running can also be freely changed. Web applications such as YouTube will keep on running even when the app is minimised allowing people to view video while for example using Facebook. As of now it is only possible to open 2 minimised windows simultaneously. The SDK can be installed into an existing Android Development Platform as an additional library where its functions can then be called and used.

3.3.5 Android Developer Best Practices

To ensure that the application has good performance and a user friendly design, the application being built will try to incorporate as much as possible within the time available and when relevant guidelines provided by Google's Android Developer website [60] on Best Practices during development. The four aspects it will touch on include performance, user's interface/input, user interaction/engagement and application testing.

3.3.5.1 Application Performance

According to the Android Developer website on performance [61], developers need to be aware of the amount of RAM the application uses during operation. This is especially important since a mobile operating system physical memory is often constrained. Here are some techniques [62] suggested to make application more memory efficient.

Firstly, use services sparingly because once a service starts, the RAM it uses cannot be used by anything else. According to them one of the worst memory management mistakes is leaving a service running when it is not needed. Instead use an IntentService function which will cancel itself once the operation is done. Secondly, when the application is not used or hidden from the user, release memory being hold by the application using onTrimMemory() callback in activity classes. This should be used with TRIM_MEMORY_UI_HIDDEN level which will notify when the application UI process becomes hidden from the user. Thirdly, checking how much memory should be used using getMemoryClass() which will reveal how much memory is allocated to the

application by the system. In the case that the allocated memory is not enough, the application could request for more memory being allocated to it although this should only be done with justifiable reasons and should be seen as a last option.

Some other performance tips [63] include, avoid the creation of unnecessary objects as each object created will being using memory. Using static methods instead of virtual ones, as invocations will be 15%-20% faster. Using Static Final for constant values which does not change. This is also good practice for any programming done for most applications. Know and use available libraries instead of using your own hand coded version as usually they are faster at doing certain processes. And finally, the existing performance of the application being built must be measurable. The application will try and incorporate these practices wherever possible.

3.3.5.2 Application User interface/input

According to the Android Developer website on User interface [64] and input [65] here are some guidelines to follow to build a good application. A good UI for an application should take into account that the application will be used in a variety of different screen sizes. So it is essential that the application is designed with this in mind. By using wrap_content and match_parent attributes for the view component of the application(usually on the xml files), the application can resize itself to either the minimum size to fit the content in that view or follow the size of the parent view which should make the view size much more flexible. This will be used to make sure that the size of the overlay being built is flexible and can be changed according to the users needs.

Smart use of RelativeLayout could also be used to align components apart such as side by side, instead of just LinearLayout which provides a straight line type of interface. Using configuration qualifiers would allow the runtime to automatically select the best layout design for different screen sizes. Such as using two pane pattern for large screens, so that content is displayed in two screens for bigger screen but only show one pane on smaller screens. Some User input guidelines includes using onTouchEvent() function on the view which received touch events. It is called on every gesture made on

the view. MotionEvent tracks the user's finger from the moment it touches the screen until it leaves the screen, and sends the data to onTouchEvent(). Different gestures can be detected using the GestureDetector class which includes functions such as onDown(), onLongPress() and onFling(). Velocity of input(e.g. gestures) can also be tracked using VelocityTracker. The OnDragListener could also be used to detect the dragging of an object on a view. This could prove useful while manipulating the overlay around the screen and position it correctly.

3.3.5.3 Application User interaction/engagement

Designing Effective Navigation [66] according to the Android Developer website is one of the main areas to focus on. Since apps usually work as a number of screens with different functionalities, it is important to plan screens and their relationships. Using entity-relationship diagrams (ERDs) is a good way to manage this. The ERD used is usually similar to any previously done system using Java or PHP. The important thing is that in this case each screen acts similarly or is defined as one function or having just one functionality. Additionally through the use of Diagram Screen Relationships, the application's direct screen changes can be tracked, which means setting up which screen comes out after the an option/function is chosen.

3.3.5.4 Application Testing and Optimization

Each main activity in the application should be tested to make sure that it works accordingly. The guidelines provided by Android Developer website [67] states that test cases should be build based on Android's custom testing framework based on JUNIT due to the fact that most android applications internal codings are build using JAVA whereas its UI can be built using XML natively or HTML/CSS/JAVASCRIPT(e.g. using a wrapper). UI testing can be done using Instrumentation on the UI thread. Usually Test methods are added to ensure that the correct behaviour is achieve with the UI. Unit Testing can also be done to check for any bugs in the application. A tool call ProGuard [68] provided by Android Developers can also be used to optimize the application before release. It does this by shrinking, optimizing and obfuscating the applications code and making it harder to reverse engineer. The Systrace [69] tool can

be used to analyse the performance of the application using collected data on execution times and displaying them to the developer. It is especially useful in detecting slow to draw or stutters in the display of motion/animation. Finally, the uiautomator can be used to test your UI using automatically created functional UI test cases.

4 Design and Implementation

4.1 Application Design

To ensure that the application(Touch Colour Overlay) can be used easily by the target users it has to be easy to access and learn without the need for extensive explanation and instructions. This part of the report will justify why the application looks the way it does, why its functions are very visually oriented, why the controls and settings are set up and created that way and finally how the colours are chosen. The application design is build on the idea to provide a virtual coloured overlay on mobile devices replicating physical ones already in existence . The design of the application will try to focus on simplicity as to make it as simple as possible for the dyslexic users to use. Basically the application is split into two screen functions. The Main screen which is also the Settings screen, is used to pick a colour for the overlay. The Floatable screen will function as the overlay for the mobile device based on the colour chosen on the Main/Setting screen.

4.1.1 Main Screen Design

The main screen will act as the settings page of the application where the colours for the overlay can be chosen. It is very straight forward where once the application is opened the user can immediately see all available colours which could act as an overlay on the screen. The colours available will be arranged in small boxes along side each other with each box filled in with different colours. The colours were chosen based on research done on available physical coloured rulers and overlays on sale today by different companies which are used to help with reading. The user will then just pick the colour they want as an overlay and it will confirm the choice by appearing below the colour choices in a much bigger box. This will indicate that the colour has been chosen and then the user can proceed to press the minimize button on the top right of the tool bar to minimize the application to its floatable screen. The design of the settings page will contain minimal word instruction and will be very visual, as the user need to only pick a colour and then minimize the window to use it as an overlay. This level of simplicity is necessary to ensure that it is easy to use with no complicated

instructions which might confuse the users.

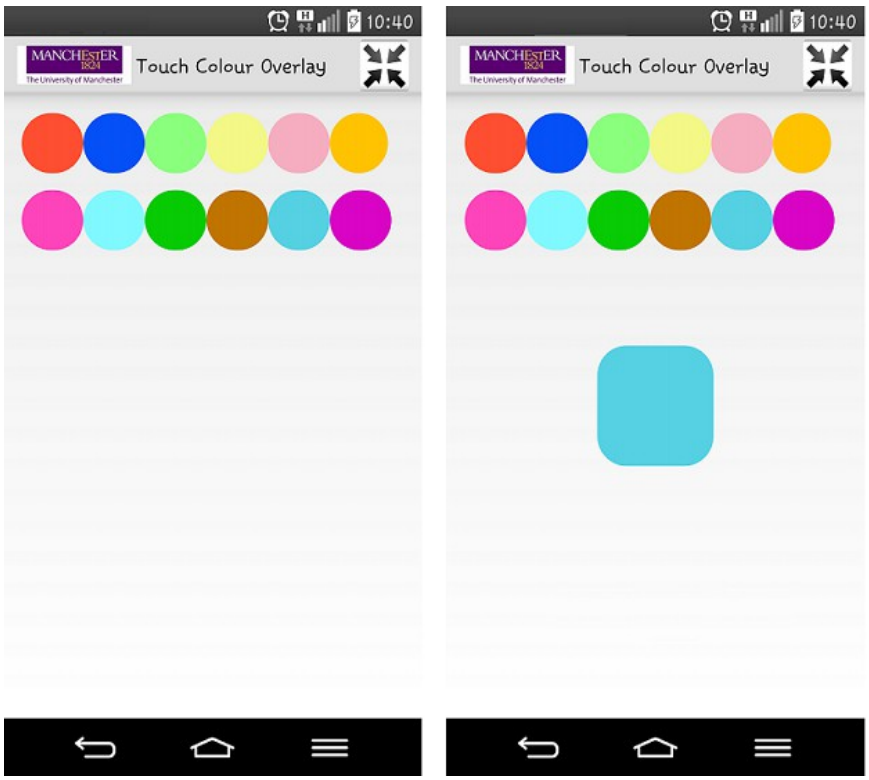

Figure 24: The main/settings screen of the application. Once a user touches a colour, it will appear as shown on the right image.

4.1.2 Floatable Overlay Screen Design

The design of the floatable screen comprises just the primary screen which will act as the overlay, the contrasting tool bar, the button to maximize the application back to its full size and also a side button to resize the application. Floatable here means being able to overlay on top of other applications. The colour overlay section of the application will follow the colour chosen in the main screen previously. The contrasting tool bar will be able to change the contrast of the primary screen colour and provide varying intensity for the colours being used. A button on the extreme bottom right of the application will allow the users to change the size of the application according to their preference. And finally a button on the top right of the screen will allows users to return to the main screen or settings screen. The reason why the settings of colours for the primary screen is provided on a separate screen(main/setting screen) instead of just

combining it with this one is because most dyslexics prefer only one colour as an overlay, they do not need to change it very often. This would also prevent over crowding of settings on this screen's tool bar keeping things simple.

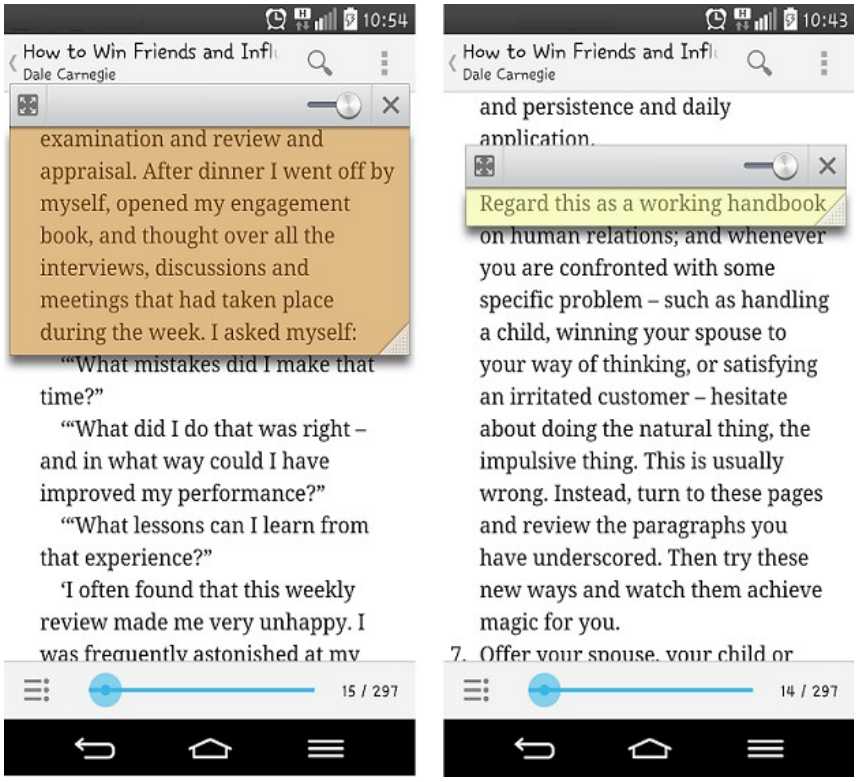

Figure 25: The application will become floatable and overlay on top of other images as shown above. Its width and height can also be changed. This is what most volunteers viewed during the usage of TCO.

4.2 Application Implementation

The implementation of the application will be explained here such as how the application is built, important code snippets and list of important functions used. An explanation of third party library functions which were used will also be provided here. Before beginning to implement the application, the environment needed for development must have already been set up based on information mentioned in the previous chapter on Android Environment and Tools set-up.

The application UI screen will be coded in XML while the main functions such as the

changing of colours will be coded with Java. This ensure that UI code does not get mixed up with core codes. It is important to understand that while the UI codes are separated into different XML files for each screen, the core coding in Java (which handles most of the function calls) is used by both screens on just a single Java file, much like a web application structure. The main body of code is built upon a sample provided by LG developer on how its library works. This sample acts as a base for the codes, and more functionality was added based on what the application needed. This library was created especially for LG devices which means as of now it can only work properly on LG mobile devices. The UI was also changed to accommodate what the application required.

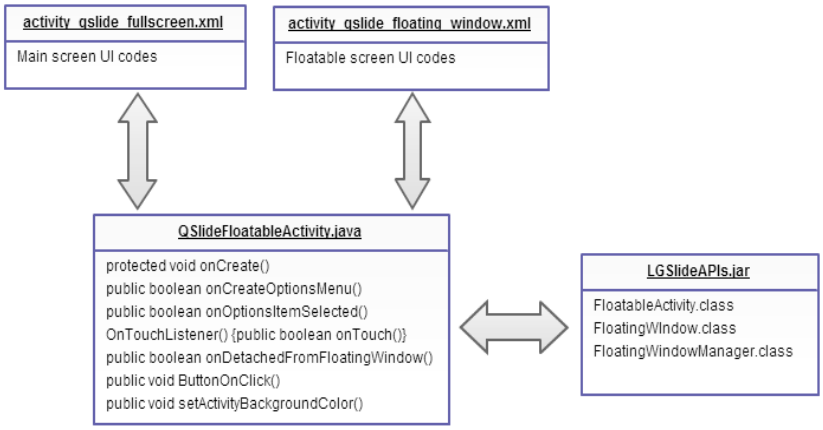

Figure 26: Simple UML diagram of the application where the xml files will take the user's input and the java file will process them.

4.2.1 Main Screen Implementation

The main screen consists of two parts which are the UI part coded in XML format, and the main functions section coded in Java. The two parts work together similar to a PHP/JSP and HTML combination on a web application where one side handles all the UI elements while the other handles all the calls to functions. Usually XML is used to handle the sending of data on web applications, so to use them as a language to build UI presents a very interesting challenge. The two main functions available on the main screen are to choose a colour for the overlay and the button to minimize the view

window or screen and make the application floatable.

4.2.1.1 Choosing a colour

On the UI section of the screen most of the codes are used to display the colour available for the overlays. The colours are built as ImageButton items so that when they are touched on the screen by the users they can send out an intent which will in turn activate a function on the Java side of the application. The function being called is ButtonOnClick() which would then assign the colour to be displayed in a larger image below the colour selection. The function also immediately sets the colour for the overlay or the floatable window view screen using setActivityBackgroundColor() which gets the current view state of the floatable window view, and using the setBackgroundColor() function, sets the colour of the view.

4.2.1.2 Minimizing the screen/view size, Making it Floatable

As explained on the design section, to minimize the screen into its floating form the user needs to press the button on the top right of the application. The function on the button to change the current state of the application into floating form is handled by onOptionsItemSelected(). To make the application floatable which means being able to overlay on top of other applications, a third party library is used provided by LG developers website. To use this function firstly the class in which these functions are stored needs to be called using " import com.lge.app.floating ". Then it is simply using the function provided which is switchToFloatingMode() to change the application into its floating form. Here you can set different perimeters for the application such as allowing it to be resizeable or not. Further perimeter settings can be set such as minHeightWeight/minWidthWeight variables which sets the minimum height/weight that the application can be changed into while in floating form, using updateLayoutParams().

4.2.2 Floatable screen Implementation

The floatable screen main function is to provide a see through screen which would act

as an overlay for text and provide an alternative background colour for text on the screen. It also includes a function to adjust the contrast for the screen and a button to maximize the view window into the main/settings screen.

4.2.2.1 Functions on Floatable screen

There are only three functions on the Floatable screen of the application. These functions are the contrast bar/slider, the resizing tool on the bottom right of the screen, and the button to maximize the application to its full size. Most of these functions are already provided by the library by default although the developer has the option to remove them. When the switchToFloatingMode() function is called it takes certain parameters values such as useOverlay(make the contrast bar/slider available) and isResizable(allow the floating window to be resizeable) to make the functions available. The contrast slider actually uses the setOpacity() function to change the contrast although in reality the floating window's opacity is the attribute changing. All the functions in the floating screen are controlled by the FloatingWindow.LayoutParams CLASS. For example, useOverlay field is the item which handles the contrast bar/slider item to change opacity. And resizeOption field handles information on how the user can resize the floating window.

4.2.3 Conclusion from Implementation

Overall the actual coding of the application feels a lot like a web application. There are separate files which handle the UI and functions being used which is quite similar to the HTML/JSP and PHP/JAVA combination usually used by web applications. Due to the availability of the library by LG it made the process of changing a window into floating form quite easy. This library could be used to create applications using multiple windows which could prove very useful in handling multiple tasks. Mobile screens are also getting larger by each release, allowing the screen to fit more than one window at any given moment. The time may just be right to introduce multiple windows function on a mobile device, much like what is available on desktop computers.

4.3 Application Testing and Installation

The application quality and standard will be tested based on the guidelines provided by the Android developer website on the Core App Quality section[70]. The table below represent the Test Procedures tested on the application.

Type	Test	Description	Result
Core Suite	CR-0	Navigate to all parts of the application. And test all available colours for overlay.	Pass
	CR-1	From each screen, press the home button and navigate back to the app using All Apps screen.	Pass
	CR-2	From each screen, switch to another running app and return to the app using Recents apps switcher.	Pass
	CR-3	From each screen, press Back button.	Pass
	CR-5	From each screen, rotate device between landscape and portrait orientation 3 times.	Pass
	CR-6	Check if the application has any running services after being send into the background by checking under Running tab, on the Apps screen. It should be no running processes.	Pass
	CR-7	Press sleep button while on the application and check if it still function when the device is awaken.	Pass
	CR-8	Repeat CR-7 but set it to lock-screen and then unlock the screen.	Pass
	CR-12	Examine permission requested by application. Make sure only the minimum permission to function granted.	Pass

Table 2 : Testing for application.

More in-depth testing of the application should be done once more advanced functionalities are added especially if the usage of databases is going to be added. As of now, since the functions of the application is still limited, testing should be focused on making sure that the application does not crash easily, can be installed properly, all the colours on display can be chosen and shown and transition to floating mode goes well even on horizontal mode or portrait view.

The application is installed onto the mobile device using eclipse. The process is usually very straight forward where once a mobile device is attached to a laptop running eclipse, the program will automatically recognize the device and decide whether the current application can be installed to it. But the device must be allowed to undergo "USB Debugging" so that eclipse is allowed to make changes to the device through the laptop it is currently connected to. For LG G2 (device being used), this can be achieved by going to Settings> General > About Phone> Software information> Build Number and then tap on the Build number field repeatedly until a message such as "You're a Developer" appears. The option for Developer options, should now appear on Settings> General menu. After installing the application, you can check the application size by checking Settings> General >Apps and choosing the application. The application is currently sized at 168 KB which is very small.

Then just run the application in eclipse and the developer can choose which detected device to install it to. The main thing to keep in mind here is making sure that the application is compatible with the current version of Android currently installed on the mobile device. The application can also be installed into a virtual Android installation on a virtual device usually on a desktop computer. But after testing it out, this has proven to be very time consuming and should only be recommended for developers with powerful machines. The ability to actually test the touch aspect of the application is also lost as touch is simulated using a mouse pointer, which is not able to produce an accurate representation of how the application's UI would properly response. The table below contains the specifications of the device used.

Item	Description
Phone	LG G2 (2014)
Manufacturer	LG Electronics Corporation
Chipset	QualComm Snapdragon 800
Memory	16 GB
RAM	2 GB
Operating System	Android 4.4.2 KitKat
Kernel Version	3.4.0 Version
Display	Full HD IPS Display (5.2 inches, 1920 X 1080 pixels)
USB	USB 2.0 HS
Wi-Fi	802.11 a/b/g/n
Battery	3,000mAh Li-Polymer (Embedded)

Table 3 : Hardware Specifications.

4.4 Application Screen Shots in usage

This section provides screen shots to show the application in usage with some different application such as Facebook messenger and Gmail. Also the application being used horizontally.

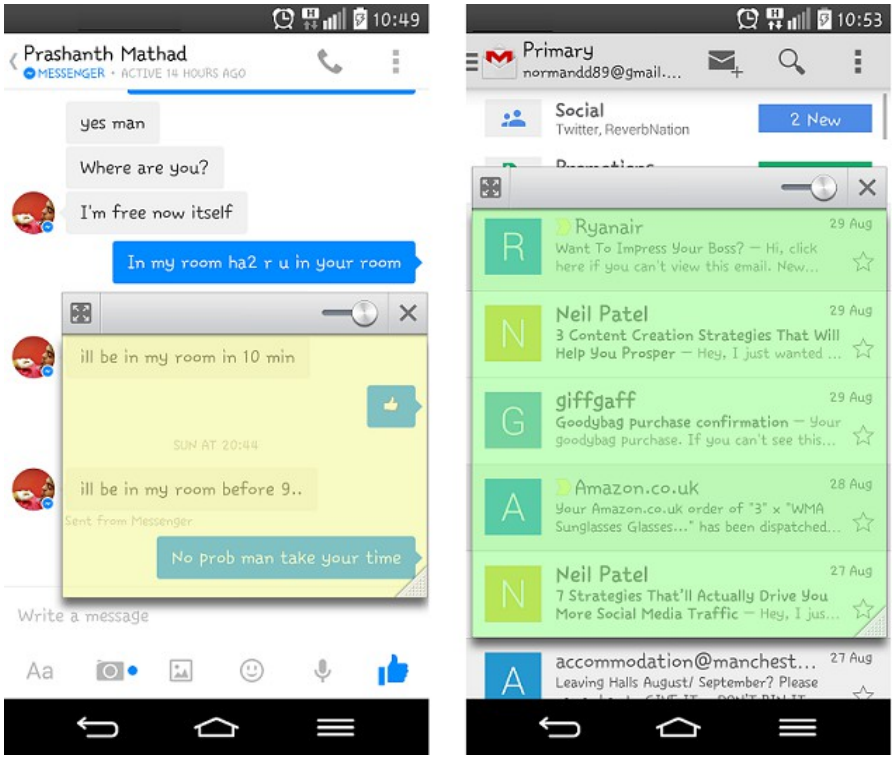

Figure 27: TCO being used on top of Facebook Messenger and Gmail.

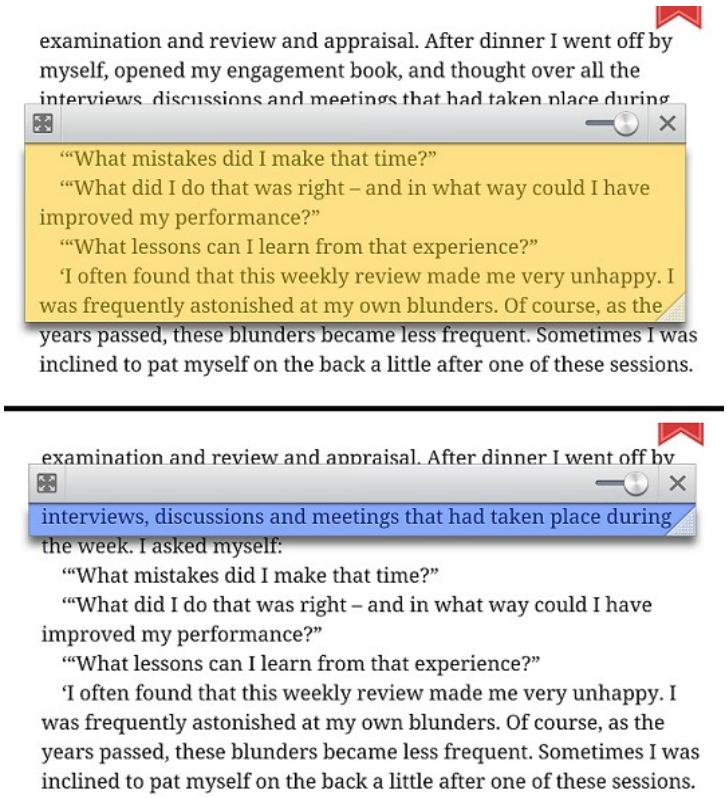

Figure 28: TCO being used horizontally with varying sizes.

5 Testing and Evaluation

This section of the report will explain how the volunteers testers were obtained, challenges faced during the testing phase, the actual testing with volunteers and a future works section which compiles all the recommendations from the users as to how the application could be better and how as a developer even more advanced functionality could be added towards building an even better application.

5.1 Gathering target volunteers

The volunteers for the testing stage were gathered through the help of the Manchester Dyslexia Self Help Group [71] which conducts a meeting between members who are mostly dyslexics every two weeks. The group chair person was introduced to me by my supervisor, and the same chair person Mr Roger B. invited me to attend one of their meetings because he was interested in the application. The application was shown to the group, and through this way the volunteers for the applications were acquired. Mr Roger Broadbent is also a study skills tutor working with dyslexic university students through Access Summit, which according to their website [72] is one of the largest and leading integrated H.E. based assessment and support providers in the UK.

Another group of volunteers were also acquired for this test as a control group consisting of adults with no history of reading problems or dyslexia just to monitor if the application had any effect on regular adults.

5.2 Ethical issues

Due to the fact that dyslexics are considered to be people with disabilities under UK Law as mentioned earlier and therefore vulnerable ethical approval was required from the University Central Ethics Committee before testing the application on the volunteers. Each volunteer was also required to sign a consent form provided by the University to ensure that they were fully aware that the testing was being done on them because they were dyslexics. This proved to be no problem as most of the people from

the group knew they had dyslexia and were very helpful in providing any sort of assistance to any causes that might help in creating solutions to manage the disability. The group's help and support were vital to this research. The ethical approval was submitted and has been approved with the approval number of CS138.

5.3 Application Testing with volunteers

When the first meeting with the group took place the application was presented to the group with the idea to collect some informal evaluation. This was to try and make the application as accommodating as possible and make certain adjustment to the application so that during the actual one-to-one testing everything will work out fine. The initial view of the application was good and the group members only had good things to say about it. They commented on how having a different coloured background on their text could definitely improve their ability to read and recognize text especially when dealing with text with black coloured font on white background which is the standard on most mobile devices and emails. This standard colour combination actually makes it hard to read for dyslexics and they preferred having a different colour for the background so that the words do not seem jumbled up, moving around or making it hard to perceive similar letters such as “b” and “d”.

The testing was finally done with five dyslexic volunteers trying out the application properly and having filled in the questionnaire forms. Some additional feedback from the volunteers were taken to add more substance and insight into the effectiveness of the application. The way the test was done was to provide the volunteers with the mobile device already installed with the application and then using it to read through a body of text on the device, which was a sample black text on a white background. They could choose and pick any available colours to act as an overlay over the text they were reading. The size of the application overlaying the text could also be changed according to their needs as some preferred the overlay to cover the whole screen while some just a line of text to help them focus better. After finishing with this, they could use the application with a combination of different types of application such as emails according to their liking.

5.4 Evaluation of test results

The study found that all five of the volunteers had felt improvement in reading ability while using the application according to their questionnaires and during the test sessions which is very encouraging. The volunteers were make up of both genders and varying ages. The study also found that most volunteers had a specific colour which was suitable for them such as yellow or purple and would just stick to one instead of constantly changing colours. Although colours with a lighter shade such as light yellow and light purple seems to be preferred. Most of the volunteers had viewed colour to be a lot of help in helping them recognize text better and it in no way affected their vision.

A few of the volunteers said that “it's just simple” and “it's straight forward” referring to how the application's design compares with existing desktop application or software which does the same function. Most of the volunteers were already using touch screen mobile devices, so they could understand the usage aspect of the application quite easily. One volunteer also likes the fact that the application could overlay on top of any other application including for example a calendar which makes it easy to use with existing applications. One volunteer even said “I have problem reading my emails” and commented how the application might help with that. They also wondered if it would be applicable for the Kindle from Amazon because currently according to them the Kindle only provided 3 types of viewing colours and this might be very useful because Kindle is mainly used for reading.

One volunteer during the experiment also commented on how black text on a white background is very hard to read and said “if you can select the colour background most helpful to you, that would be great” while using the application. The volunteer also commented that “the colour selection there is really good”, but further commented that “a paler orange/brown” could be better and generally the darker colours are not that effective. The volunteers also liked the fact that the contrast of the overlay could be changed because some like less brighter or on lower contrast. Volunteers also like the fact that the application size is adjustable making it possible to have varying size from a ruler size which highlights each line so that the sentences can be focused on more easily

to having it made bigger to view bigger chunk of text. Although a few volunteers would have liked to have the application cover the whole mobile device view area while reading for example a book on their mobile devices because currently the application is not able to cover the whole mobile device screen.

Based on the questionnaires, most volunteers gave the application an average of 4 and 5 points out of 5 on the applications ability to help with reading. An average result of 4 points out of 5 on the ability to recognize text/sentences better. An average of 4 points out of 5 on increased reading speed and an average of 5 point out of 5 on whether the application affected vision in a positive way. All dyslexic participants also did not feel any discomfort while using the application. 3 out of 5 volunteers have had experience in using software which could assist them, and some of them have been highlighted in previous section of the report such as Read&Write software and Text-to-Speech applications.

From the testing done on dyslexics it can be concluded that the positive effect of different colour background to a dyslexic person's vision is quite significant as stated in the research done previously. This further supports the application's effectiveness although in the future more volunteers could be approached to build up an even more concrete evidence base. Some recommendations by the volunteers will be included in the section future works to discuss how the application could be better improved in the future as well as available technology that could be added to the application with more development time.

A control group consisting of five normal users with no history of reading difficulties were also approached for testing just to get better insight into how normal people without dyslexia would view and comment on the application. From the experiment most of the users found that the application did not help with reading and actually found it harder to read on a different colour background compared to white. It also affected some of the normal users vision in a negative way making it harder for them to focus on the words and sentences. This result was to be expected, but the important issue to realize here is that, if developers, who are mainly people without dyslexia, cannot even see the obstacles being faced by dyslexics and even find application build for dyslexics not useful and bothersome, how are they expected to create application

specifically for dyslexics. This is another challenge faced by the developer community to begin actively creating application for dyslexics. Much more research should be done towards finding a way to promote developers to produce applications which are user friendly to dyslexics.

Overall the testing phase was able to collect a lot of interesting information on how mobile applications for dyslexics should be built. It could help developers to gain a better understanding into how dyslexics view applications and what their pattern of usage is. This study could be used as a platform for future students projects and could give better insight into what it takes to build and test out assistive applications. Hopefully in the future this study would be able to help more students to create applications suited for dyslexics or in fact towards any other disability.

6 Future works and Conclusion

6.1 Future works

This section contains recommendations and advanced functionalities for the application in the future based on the feedback collected from the study and research on additional technology available which would provide the application with more functionality but making sure its simplicity in usage is maintained as sometimes added functionality would slowly create a more complex application.

6.1.1 Recommendations from users

Here are some recommendations from dyslexic users on how the application could be better. Firstly would be making it available on different platforms such as Apple and Kindle, and also on different mobile devices manufacturers such as Samsung. As mentioned earlier, the application uses a library from the LG developers website, which means that it will have trouble functioning on other devices other than an LG device. To have it worked on a Samsung, Apple or Kindle device, a similar library with the ability to make the application floatable would have to be acquired and the coding would have to be changed to suit the target device environment.

The next recommendation would be the ability for the application to provide an overlay for the whole width and size of the screen as previously mentioned. This could provide an overlay to fit the whole screen which according to the volunteers would help them with things like reading an entire screen text filling application such as a book or magazine application. Another addition they would like to have is a dictionary function on the application itself where by when a person pressed any word currently below the application or overlay, it would be able to give out its definition, meaning and maybe read it out, similar to a dictionary function already available on Apple's iPhones. The only difference is that this will be processed by the application itself where anything under the overlay including PDFs, emails, websites, messages and images could be scanned for text.

Adding a function which could change the actual font of the text being viewed was also suggested since different fonts could prove to be more accommodating to dyslexics compared to others as shown by the creation of the OpenDyslexic font mentioned earlier. This could be done by extracting out the text currently being viewed by the application using screen shots of text currently under the overlay and then displaying them on the application screen again but with different font. This would require a high degree of image manipulation techniques to make it function at realtime speed.

The next recommendation would be to have some sort of pointer on top of each individual word to act as a guide for each word. People with severe dyslexia would find this especially useful as each word is guided by a tool to point exactly what they should be reading. The pointer would also move along each word based on a time frame set by the user. The pointer would exist within the scope of the overlay size so that it does not point to anything the user would not want. Another function that could be added since the screen size could be changed to resemble a ruler, would be a function which could be set to make the application move automatically down a page of text so that user would not have to manually scroll the application down after finishing reading each line. The timeframe for this movement would also be set by the user to ensure its speed is suitable and it could also be used as a training method to train the user to read faster and improve reading speed by gradually increasing the scrolling speed.

One functionality that could help make the application more user friendly, would be to enable the floatable screen size to be changed using two fingers, much like enlarging a picture on a touch screen mobile device. During the testing phase, most users instinctively did this action with two fingers to try and enlarge the application screen instead of using the tool provided on the bottom right of the application as shown on the screen shots. A custom colour chart to allow users to mix their own colour could also be added to allow even more freedom for the users regarding colour choices. Little things like this could help the user gain a better experience while using the application. Sometimes the design of mobile application needs to take into account what the users are already accustomed to and try to build based on that.

6.1.2 Advanced functionalities

This section will highlight technologies currently available to make a number of the recommendation from the users possible, but due to the time constraint of the project, the current application was not able to implement them. Most of the recommended functions such as the dictionary and font change could be achieved using Optical Character Recognition(OCR) [73] technology which will be able to recognize characters on an image and then translate them into computer-readable text which can then be manipulated by the application. OCR on mobile devices is usually used to extract text from an image taken by a mobile phone camera. An example of its usage would be, a tourist from the UK could be visiting Italy for a holiday trip, but once arriving at Italy the tourist finds that all the signboards are in Italian. To overcome this problem the tourist could then take an image of the signboard whose language is in Italian and use an OCR software to extract the text from the image and translate it into English once the image is saved inside the phone. An example of an existing application is Google Goggles [74] available on Google Play.

From this example, the same technique could be applied to the application with a little difference. The way it could work is, once the application is overlaying a certain text, the item under the overlay would be print screened and converted into an image in real time. And through this printed screen image, OCR libraries would be able to process and extract text out of them. Once the text has been extracted into a computer readable text type its font can then be changed. The reason it is better to do it this way is because the user would not have to manually highlight the words themselves as is the usual action by users, for example when they need to copy and paste something on a mobile device. This library would allow processing to be done real time and constantly changing any text below the overlay without the user having to do any action at all. They just need to make sure the text in being overlay by the application. The font would be set by the user themselves.

Some OCR libraries already available for Android include ABBYY Mobile OCR Engine for Android [75] (Commercial) and Tesseract-ocr [76] (Open sourced) available

on Google code. These libraries can be used to create Android applications with OCR capabilities. Some of the functions provided by these libraries include, to save, edit, send via email text already extracted from images, support for Android and IOS operating systems, support for multiple languages, and produces low resource requirements and optimized memory management. These libraries, based on an article by Ray Smith [77] from Google Inc., use a variety of algorithms to extract text. Some of them include, line finding algorithms which leads to baseline fitting and fixed pitch detection. To fully understand the technology behind OCR and fully utilise its capabilities much more time is required, as it steps more deeply into the boundary of complex image manipulation techniques. Although with the availability of these libraries hopefully it will be packaged into a more accessible form for mobile applications developers to use.

6.1.3 Marketing it on Google Play

The final step to make the application accessible is publishing it on Google Play. Based on research done earlier, it was discovered that in the UK alone around 10% of the population suffers from some form of Dyslexia, and since the UK population is around 64.1 million, according to the Office for National Statistics [78], translating this into numbers means that there is a potential user range of about 6.4 million for the application. And given that current desktop software to help dyslexics is expensive, users might be interested to try a mobile solution which would cost less. Getting an application successfully published on Google Play requires it to fulfil certain criteria set by Google. According to the guidelines provided by Google on publishing or launching an application [79], developers are advised to follow the following check-list items.

1	Understanding the publishing process set by Google by reading the Publishing overview document [80] which includes information such as preparing the application for release(e.g. Removing log calls, Building and signing a release version) and determining the way the application is release(e.g. Google Play, email or website)
2	Understanding Google Play policies and agreements to prevent the application from being suspended or termination of developer account.
3	Test application for quality as explained in the design and implementation chapter.
4	Determining Application content rating to check its maturity levels.
5	Determining which country to distribute to by doing some market research.
6	Application overall size is determine. Max .Apk size is 50 mb.
7	Confirming the applications target platform and screen compatibility.
8	Decide if it will be priced or setting the application as a free download. Free applications must remain free.
9	Consider whether to use billing through the application or not.
10	Setting the price for the application, if it is chosen to be priced.
11	Start localization the application based on the release area.
12	Preparing promotional graphics, screen shots, and videos.
13	Build and Upload the Release-ready APK.
14	Planning a Beta release by following Google's beta program for your application.
15	Complete the App's Store listing.
16	Using Google Play Badges and links in your promotional campaign.
17	Final checks and publishing, by signing in to the Developer Console.
18	Provide users with support after launch.

Table 4 : Google's Launch Check-list.

6.2 Conclusion

In conclusion, the study has shown the positive effects that the application has for dyslexics where most dyslexic users find that it helps with reading and recognizing text. Although more thorough tests in much bigger numbers should be conducted to gain statistical proof of its effectiveness. The unique relationship between colour and dyslexia should also be taken into account for future development of mobile applications with the intention of providing better accessibility to people with dyslexia.

The challenge of finding dyslexic adults to collect feedback from could be overcome with the establishment of a new scheme, possibly in collaboration with the Disability Support Office of the University or through UMIP(University of Manchester Intellectual Property Commercialisation Agent). It would be easier to implement this scheme into an already existing structure. For example, the Disability Support Office could advise developers on how to properly work with disabled users and find suitable volunteers for the developer. UMIP could help by making sure that any software or application being produce or endorse by them would require a certain standard of accessibility for disabled users before being cleared to commercialise. This would also help introduce this usually unseen aspect of application design and development to future developers.

As seen in the future works sections, a lot more can be done in terms of making the application better and with increased functionality, although the main issue would be finding future talents willing to bring the idea forward. Overall the findings have shown that with adequate research, suitable and simple solutions can be created for dyslexics and hopefully this will encourage more developers to look into this untapped section of mobile development.

REFERENCES

[1] Papworth Trust (2012) Disability in the United Kingdom 2012 [Online]
Available:http://www.papworth.org.uk/downloads/disabilityintheunitedkingdom2012_120910112857.pdf [17 April 2014]

[2] The British Dyslexia Association Website (n.a) About Us page [Online]
Available : http://www.bdadyslexia.org.uk/about-us.html [17 April 2014]

[3] LG Qslide SDK (2014) LG Developer, Introduction to the LG QSlide SDK [Online]
Available : http://developer.lge.com/resource/mobile/RetrieveOverview.dev?categoryTypeCode=ANRS [17 April 2014]

[4] W3C Web Accessibility Initiative (2012) Stories of Web Users, Status : Draft Updated 1 August 2012 [Online]
Available : http://www.w3.org/WAI/intro/people-use-web/stories [17 April 2014]

[5] The British Dyslexia Association Website (n.a) Dyslexia Research Information page [Online]
Available : http://www.bdadyslexia.org.uk/about-dyslexia/further-information/dyslexia-research-information-.html [26 April 2014]

[6] The NHS Official Website (2014) Dyslexia [Online]
Available : http://www.nhs.uk/conditions/dyslexia/Pages/Introduction.aspx [26 April 2014]

[7] Oxford Dictionaries Official Website (2014) Definition of Dyslexia in English [Online]
Available : http://www.oxforddictionaries.com/definition/english/dyslexia [26 April 2014]

[8] R.N. Malatesha and P.G. Aaron (1982) Reading Disorders: Variety and Treatment

[9] Catherine Christo, John Davis and Stephen E. Brock (2009) Identifying, Assessing, and Treating Dyslexia at School [Online]
Available : http://download.springer.com/static/pdf/757/bok%253A978-0-387-88600-8.pdf?auth66=1398683959_2c7d6710580bbe3906610f5a08608c2c&ext=.pdf [26 April 2014]

[10] Gad Elbeheri and John Everatt (2009) 'Dyslexia and IQ : Research to practice' in Gavin Reid (ed.) The Routledge Companion to Dyslexia, 2, 22-30

[11] Gavin Reid (2009) Dyslexia : A Practitioner's Handbook , Forth Edition, Wiley-Blackwell, 2, 13-20

[12]Gilger J. W., Pennington B. F. and De Fries J. C. (1991) Risk for reading disability as a function of parental history in three family studies, 3, 205-218

[13]Singleton C. (2009) 'Visual Stress and Dyslexia' in Gavin Reid (ed.) The Routledge

Companion to Dyslexia, 4, 43-54

[14]Whiteley H. E. and Smith C. D. (2001) The use of tinted lenses to alleviate reading difficulties. Journal of Research in Reading, 24, 20-40

[15]Singleton C. H. and Henderson L. M. (2007) Computerised screening for visual stress in children with dyslexia. Dyslexia, 13, 130-151

[16]Geschwind N. and Galaburda A. (1985) Cerebral lateralisation biological mechanisms associations and pathology: A hypothesis and a programme for research. Archives of Neurology, 42, 428-459

[17]Vellutino F. R., Fletcher J. M., Snowling M. J. and Scanlon D. M. (2004) Specific reading disability(dyslexia): What have we learnt in the past four decades? Journal of child psychology and Psychiatry, 45(1), 2-40

[18]Ehri L., Nunes S., Willows D., Schuster B., Yaghoub-Zadeh Z. and Shanahan, T. (2001). Phonemic awareness instruction helps children learn to read: Evidence from the National Reading Panel's meta-analysis. Reading Research Quarterly, 36, 250–287.

[19]Torgesen J.,Wagner R. and Rashotte, C. (1994). Longitudinal studies of phonological processing and reading. Journal of Learning Disabilities, 27, 276–286.

[20]J Scott (1994) The British Journal of Psychiatry : Cognitive Therapy [Online] Available : http://bjp.rcpsych.org/content/165/1/126.long [2 May 2014]

[21]Aaron M. Schmidt and Richard P. DeShon (2010) Journal of Applied Psychology: The Moderating Effects of Performance Ambiguity on the Relationship Between Self-Efficacy and Performance, Vol. 95, No. 5, 572–58

[22]Bandura, A., & Locke, E. A. (2003). Negative self-efficacy and goal effects revisited. Journal of Applied Psychology, 88, 87–99.

[23]Peter J. Hatcher (2000) Sound links in reading and spelling with discrepancy-defined dyslexics and children with moderate learning difficulties, Reading and Writing : An Interdisciplinary Journal, 13, 257-272

[24]Andrew J. Holliman and Jane Hurry (2013) The effects of Reading Recovery on children's literacy progress and special educational needs status: a three-year follow-up study, Educational psychology : An International Journal of Experimental Educational Psychology [Online]
Available : http://www.tandfonline.com/doi/pdf/10.1080/01443410.2013.785048 [2 May 2014]

[25]Piia Bjorn, Irma Kakkuri, Pirkko Karvonen and Paavo H. T. Lapannen (2012) Accelerating early language development with multi-sensory training [Online]
Available : http://www.tandfonline.com/doi/full/10.1080/03004430.2011.646716#.U2Nq1fldWxQ [2 May 2014]

[26]Saptarshi Purkayastha, Namrata Nehete, and Jashodhara Purkayastha (2012) Dyscover : An Orton-Gillingham approach inspired multi-sensory learning application for dyslexic children, World Congress on Information and Communication Technologies, 685-690

[27]Joseph K. Torgesen, Richard K. Wagner, Carol A. Rashotte, Jeannine Herron and Patricia Lindamood (2010) Computer-assisted instruction to prevent early reading difficulties in students at risk for dyslexia: Outcomes from two instructional approaches, Annals of Dyslexia, 60, 40-56

[28]Monika Kast, Gian-Marco Baschera, Markus Gross, Lutz Jancke and Martin Meyer (2011) Computer-based learning of spelling skills in children with and without dyslexia, Annals of Dyslexia, 61, 177-200

[29]OpenDyslexic Font Official website (2014) About page [Online]
Available : www.opendyslexic.org/about [3 May 2014]

[30] The Dyslexia Shop (2014) Crossbow A4 Coloured Overlays [Online]
Available : http://www.thedyslexiashop.co.uk/crossbow-coloured-overlays-pack-of-5-of-single-colour.html [31 August 2014]

[31]The Dyslexia Shop (2014) Crossbow Coloured Reading Ruler [Online]
Available : http://www.thedyslexiashop.co.uk/crossbow-coloured-reading-ruler-duo-pack-of-10-of-single-colour.html [31 August 2014]

[32]Claro Software Official website (2014) Home Page [Online]
Available : www.clarosoftware.com [3 May 2014]

[33]Readability Official website (2014) About Page [Online]
Available : www.readability.com [3 May 2014]

[34]Arc90 Official website (2014) Home Page [Online]
Available : www.arc90.com [3 May 2014]

[35]IFTTT Official website (2014) Connect Readability to Facebook [Online]
Available : https://ifttt.com/connect/readability/facebook [3 May 2014]

[36]TextHelp Ltd website (2014) Read&Write Gold [Online]
Available : http://www.texthelp.com/UK/Our-products/Readwrite [31 August 2014]

[37]Mobile112 Download page on Google Play (2013) Google Play Website [Online]
Available : https://play.google.com/store/apps/details?id=net.mobile112.mobile112 [3 May 2014]

[38]Mobile112 Official website (n.a) Home Page [Online]
Available : http://www.mobile112.net [3 May 2014]

[39]iWordQ Information page on iTunes (2013) Apple iTunes website [Online]
Available : https://itunes.apple.com/gb/app/iwordq-us/id557929840?mt=8 [4 May 2014]

[40]Quillsoft Ltd. Official website (2014) Home page goQ : iWordQ News [Online]
Available : https://www.goqsoftware.com [4 May 2014]

[41]Dyseggxia offical website (2012) Home page [Online]
Available : http://dyseggxia.com [31 August 2014]

[42]Gary J. Williams, Gregory Kitchener, Leonard J. Press, Mitchell M. Scheiman and Glen T. Steele (2004) The Use of Tinted lenses and Coloured Overlays for the Treatment of Dyslexia and Other Related Reading and learning Disorders, OPTOMETRY, 75, 720-722

[43]Anita J. Simmers, Lyle S. Gray and Arnold J. Wilkins (2002) The influence of tinted lenses upon ocular accommodation, Vision Research 41, 1229-1238

[44]Wilkins A. J., Milroy R., Nimmo-Smith I., Wright A., Tyrell R., Holland K., Martin J., Bald J., Yale S., Miles T. and Noakes (1992) Preliminary observations concerning treatment of visual discomfort and associated perceptual distortion. Ophthalmic and Physiological Optics, 12, 257–263

[45]Peter M Allen1, Sonia Dedi, Dimple Kumar, Tanuj Patel, Mohammed Aloo, Arnold J Wilkins (2012) Accommodation, pattern glare, and coloured overlays, Perception, 41, 1458-1467

[46]Roger Hall, Nicola Ray, Priscilla Harries, and John Stein (2013) A comparison of two-coloured filter systems for treating visual reading difficulties, Disability and Rehabilitation, 35(26), 2221-2226

[47]Wilkins A. J., Jeanes R. J., P. D. Pumfrey and M. Laskier (1996) Rate of Reading Test : its reliability, and its validity in the assessment of the effects of coloured overlays, Ophthal. Physiol. Opt. ,16, 491-497

[48]Android Developer Official website (2014) Workflow : Introduction [Online]
Available : http://developer.android.com/tools/workflow/index.html [7 May 2014]

[49]Android Developer Official website (2014) Setting Up Virtual Devices : Managing Virtual Devices [Online]
Available : http://developer.android.com/tools/devices/index.html [7 May 2014]

[50]Android Developer Official website (2014) Using Hardware Devices [Online]
Available : http://developer.android.com/tools/device.html [7 May 2014]

[51]Android Developer Official website (2014) Setting Up Projects : Managing Projects [Online]
Available : http://developer.android.com/tools/projects/index.html [7 May 2014]

[52]Android Developer Official website (2014) Testing [Online]
Available : http://developer.android.com/tools/testing/index.html [7 May 2014]

[53]Android Source Official website (2014) Devices : Android Security Overview

[Online]
Available : https://source.android.com/devices/tech/security/ [7 May 2014]

[54] Android Developer channel on Youtube (2007) Androidology - Part 1 of 3 - Architecture Overview[Online]
Available : https://www.youtube.com/watch?v=QBGfUs9mQYY [7 May 2014]

[55]Android Developer Official website (2014) Introduction : Applications Fundamentals [Online]
Available : http://developer.android.com/guide/components/fundamentals.html [7 May 2014]

[56]Android Developer Official website (2014) App Components : Activities [Online]
Available : http://developer.android.com/guide/components/activities.html [7 May 2014]

[57]Android Developer Official website (2014) App Components : Services [Online]
Available : http://developer.android.com/guide/components/services.html [7 May 2014]

[58]Android Developer Official website (2014) Develop : Reference : BroadcastReceiver [Online]
Available : https://developer.android.com/reference/android/content/BroadcastReceiver.html [7 May 2014]

[59]Android Developer Official website (2014) App Components : Content Providers [Online]
Available : http://developer.android.com/guide/topics/providers/content-providers.html [7 May 2014]

[60]Android Developer Official website (2014) Best Practices [Online]
Available : http://developer.android.com/guide/practices/index.html [31 August 2014]

[61]Android Developer Official website (2014) Best Practices for Performance [Online]
Available : http://developer.android.com/training/best-performance.html [31 August 2014]

[62]Android Developer Official website (2014) Managing Your App's Memory [Online]
Available : http://developer.android.com/training/articles/memory.html [31 August 2014]

[63]Android Developer Official website (2014) Performance Tips [Online]
Available : http://developer.android.com/training/articles/perf-tips.html [31 August 2014]

[64]Android Developer Official website (2014) Best Practices for User Interface [Online]
Available : https://developer.android.com/training/best-ui.html [31 August 2014]

[65]Android Developer Official website (2014) Best Practices for User Input [Online]

Available : https://developer.android.com/training/best-user-input.html [31 August 2014]

[66]Android Developer Official website (2014) Best Practices for Interaction and Engagement [Online]
Available : https://developer.android.com/training/best-ux.html [31 August 2014]

[67]Android Developer Official website (2014) Best Practices for Testing [Online]
Available : https://developer.android.com/training/testing.html [31 August 2014]

[68]Android Developer Official website (2014) Proguard [Online]
Available : http://developer.android.com/tools/help/proguard.html [31 August 2014]

[69]Android Developer Official website (2014) Systrace [Online]
Available : http://developer.android.com/tools/help/systrace.html [31 August 2014]

[70]Android Developer Official website (2014) Core App Quality [Online]
Available : http://developer.android.com/distribute/essentials/quality/core.html [31 August 2014]

[71]Manchester Dyslexia Self Help Group Official website (2014) Welcome Page [Online]
Available : http://manchesterdyslexicselfhelpgroup.weebly.com/ [31 August 2014]

[72]Access Summit Official website (2014) Home Page [Online]
Available : https://access-summit.org.uk/ [31 August 2014]

[73]Wikipedia website (2014) Optical character recognition [Online]
Available : http://en.wikipedia.org/wiki/Optical_character_recognition [31 August 2014]

[74]Google Play website (2014) Google Googles [Online]
Available : https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=en_GB [31 August 2014]

[75]ABBYY official website (2014) ABBYY Mobile OCR Engine [Online]
Available : http://www.abbyy.com/mobileocr/ [31 August 2014]

[76]Google Code website (2014) tesseract-ocr [Online]
Available : https://code.google.com/p/tesseract-ocr/ [31 August 2014]

[77]Ray Smith (2007) An Overview of the Tesseract OCR Engine : Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society (2007), pp. 629-633

[78]Office for National Statistics (2013) Population [Online]
Available : http://ons.gov.uk/ons/taxonomy/index.html?nscl=Population [31 August 2014]

[79]Android Developer Official website (2014) Launch Checklist [Online] Available : http://developer.android.com/distribute/tools/launch-checklist.html [31 August 2014]

[80]Android Developer Official website (2014) Publishing Overview [Online] Available : http://developer.android.com/tools/publishing/publishing_overview.html [31 August 2014]

APPENDIXES

MANCHESTER 1824

The University of Manchester

Participant's questionnaire form

1) Did the application help with reading	No			Yes	
2) On what scale did the application help with reading	1	2	3	4	5
3) Were you able to recognize text/sentences better	1	2	3	4	5
4) Did you feel an increase in reading speed	1	2	3	4	5
5) Did the application affect your vision on the screen	1	2	3	4	5
6) Did you feel any discomfort while using the application	No			Yes	
7) Previous experience with using assistive software	No			Yes	

8) If yes, please explain

_____

_____

9) Any other comments or improvements/functions you would like to see on the application

_____

_____

Figure 1 : Questionnaire in Times New Roman Font.

MANCHESTER 1824

The University of Manchester

Participant's questionnaire form

1) Did the application help with reading	No		Yes		
2) On what scale did the application help with reading	1	2	3	4	5
3) Were you able to recognize text/sentences better	1	2	3	4	5
4) Did you feel an increase in reading speed	1	2	3	4	5
5) Did the application affect your vision on the screen	1	2	3	4	5
6) Did you feel any discomfort while using the application .	No		Yes		
7) Previous experience with using assistive software	No		Yes		
8) If yes, please explain _____ _____					
9) Any other comments or improvements/functions you would like to see on the application _____ _____					

Figure 2 : Questionnaire in OpenDyslexic Font.

MANCHESTER 1824

The University of Manchester

Participant's questionnaire form

1) Did the application help with reading	No			Yes	
2) On what scale did the application help with reading	1	2	3	4	5
3) Were you able to recognize text/sentences better	1	2	3	4	5
4) Did you feel an increase in reading speed	1	2	3	4	5
5) Did the application affect your vision on the screen	1	2	3	4	5
6) Did you feel any discomfort while using the application .	No			Yes	
7) Previous experience with using assistive software	No			Yes	
8) If yes, please explain _____ _____					
9) Any other comments or improvements/functions you would like to see on the application _____ _____					

Figure 3 : Questionnaire in Arial Font.