Adoption of IT addictions to methodically enhance reading skills

A dissertation submitted for the Degree of Master of Information Technology

D.L.L Abeysuriya
University of Colombo School of Computing

2019

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.
To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D.L.L Abeysuriya

Registration Number: 2016/MIT/002

Index Number: 16550027

____

Signature: Date:

This is to certify that this thesis is based on the work of
Ms. D.L.L Abeysuriya
under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name: Mr. Gihan Seneviratne

____

Signature: Date:

Abstract

When it comes to reading it has been a crucial fact from the earliest ages of mankind to the 21st century. Even though the technology has been improved a lot, still reading is the main source of knowledge and fast learning. In fact, the fast reading has become an important skill to improve which helps in many ways to learn fast. Mobile Applications have almost replaced the main media of delivering the reading materials in the past few decades which was mainly books and newspapers. SpeedRead application is also targeted on that majority, mobile devices as a light weight application for users those who wish to improve their reading skills. It has been developed based on a research done by the Bureau of Study Counsel (BSC) at Harvard University regarding eye movement patterns on effective reading. This app allows users to speed up their reading, practice the patterns of correct eye movements by using the texts from various reading materials based on their interest. SpeedRead application has brought you a verified methodical approach to make your readings fast. Further it has been through a considerable set of evaluation phases to ensure the usability, performance and efficiency of the application. According to the tests it guarantees more than 95% functional accuracy and approximately 62.6% of the android device community will be able to run it without any interference.

Keywords: Mobile Applications, light weight, reading skills, eye movement patterns

Acknowledgement

I would not have been possible to complete this project successfully, without the kind of support, assistance and courage of many people. I would like to thanks to all of them.

I would like to thank my supervisor Mr.Gihan Seneviratne and the other lectures for guidance, supervision and encouragement.

I would like to thank everyone who participated online surveys, interviews and contributing their ideas for critical evaluation.

Finally, I would like to thank my family for their support and encouragement throughout the duration of the project.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

GEM	-	Guided Eye Movement
GTM	-	Google Tag Manager
OCR	-	Optical Character Recognition
OS	-	Operating System
DOD	-	Definition of Done
QA	-	Quality Assurance
UAT	-	User Acceptance Testing
CI & CD	-	Continuous Integration and Continuous Deployment
UI & UX	-	User Interface and User Experience
HTML	-	Hyper Text Markup Language
VR	-	Virtual Reality
API	-	Application Programming Interface

1. Introduction

1.1. Motivation

Reading skills is one of the most challenging and demanding aspects of learners. In today's society people are losing their interest in reading due to poor reading skills. As a result of that, most of the people always suffers from lack of knowledge, poor thinking ability and imagination. Become a fast reader is a key to one of the doors of success. The knowledge you can gain from that is beyond the limits. Therefore, even now, most of the knowledge are buried as text (may be digitized or undigitized) materials all over the globe.

However, the problem is, "how to become a fast reader?". Nowadays people are seeking ways to improve their reading speed or other reading skills. One common way is following courses or training sessions. A well-known example is, Study Strategies course, offered by the Bureau of Study Counsel (BSC) at Harvard University [1]. Drawbacks of those types of courses are time consuming, not cost effective and success rate is not proven. The proposed solution is a mobile application which offers the mobility experience to the user to keep in touch with their daily practices. This will be more convenient than using it in desktops while providing all the features to a single hand. This will consist of a set of sample texts and a set of pre-configured presets. Once the application is installed from Google Play store, users be able to start practicing at once without creating custom configurations. Other than that, readers can make opportunity to practice text in any language.

According to the surveys [8] there are a lot of people who need to improve their reading speed, however, don't know how to do it. In Addition, most of them are, Students who don't have enough money to pay for, Employees who don't have enough time to spend on and People who don't like to spend money on unproven techniques. The proposed solution which is a mobile application will address the above problem, "how to become a fast reader?" by covering all the above aspects as the proposed mobile application will be running on android platforms which is more affordable compared to the other mobile devices, Since this comes as a mobile application, they don't need to dedicate their time on this. As an example, they can use this while travelling and This comes as totally free application, so even though this is unproven, there's no risk.

1.2. Project Goals

1.2.1. Aims and Objectives

This application will serve more with ease where the users will be able to save their configuration adjustments, they made by creating customizable presets. These custom presets can be saved with any name where users can modify any configuration it holds at any time. The proposed application will bring users the flexibility to adjust any of the configurations such as total running time, words per minute, characters per window, words per window, windows per line, windows per minute, words per minute, moving speed, font size, font format and number of overlapping characters.

The place is not a limitation for this application (mobility feature) since the users might be able to save the settings and use them anytime to practicing anywhere, they want and continue to practice once they come back. Unlike a traditional GEM video, this application will provide clearer user-friendly surface to read where the users will be able to grab it easily. In addition to that, having the ability to change font sizes and formats will give them an extra visibility. Inputs will be allowed from Google Drive, Printed Books, Websites, External or internal memory, eBooks, ePubs and from any other internal or external apps. Users will be able to input text from any language for practices.

Since the solution is a mobile application user may be limited to digitized reading content. Going beyond that this application will offer the chance to practice with printed materials by extracting its text using the device's camera. A powerful OCR library provided by Google Vision will be integrated to facilitate this feature with an acceptable level of accuracy as stated by Google.

1.2.2. Scope

The proposed mobile application consists of internal app components similarly practice board, control board, preloaded texts, scan book, clipped texts and practice text file with G-Drive. In order to facilitate the internal app components stated above, this application will use the external components similarly OCR, Google Drive sdk and Clipboard Manager.

1.2.3. Deliverables

As the application a release .apk file will be delivered at the end of the project and Source code of the final release will be shared using a Git repository. For Compatibility Evidence serving as a list of pre tested devices and OS versions will be presented as evidence of compatibility. A configured Google Analytics and Google Tag Manager accounts will be shared with the analysis of major functionalities. Final Project Report will discuss approaches, techniques and technologies to carry out this project.

2. Background Study

2.1. A Literature Review

Harvard University did a research on understanding the eye movements when reading texts. That was a very successful research and based on their findings, a group of lecturers and professors were able to add a new module to the Harvard Course in Reading and Study Strategies, offered by the Bureau of Study Counsel (BSC) at Harvard University [1]. The course was originally developed by William G. Perry, Jr., the founding director of the BSC.

It has been continuously revised to reflect advances in the field of reading cognition and changes in reading demands and media. The main purpose is to aid the reading difficulties in improving text readability which is guided by a book, "The Anthology for Strategic Reader" [2] by Dr. Abigail Lipson(Director of the Bureau of Study Counsel at Harvard University) and Sheila M. Reindl(Associate Director of the Bureau of Study Counsel at Harvard University). With all the technologies existed at that time, they decided to create a flash video with preloaded text and configurations which is entirely predefined as a solution and named it as GEM video [6]. Since then the course has been conducted successfully with a set of similar videos.

Moving forward the Harvard university became partners with Pearson (a leading books publishing and educational software company) and came up with a web application for this course [3]. Any how they managed to embed the same set of videos to that application as predefined course materials. Nowadays, the feedback of students and the authors the usage of GEM videos has several drawbacks just as configurations are predefined, duration of a video cannot be changed, creating a new video is a heavy task, length of text is limited, users are not able to practice with a text they like, users won't be able to practice on the run and using it in mobiles is not practical.

3. Methodology

3.1. Introduction

As mentioned in the problem justification, this project delivers a software solution to a research outcome conducted by the Harvard Course in Reading and Study Strategies, offered by the Bureau of Study Counsel (BSC) at Harvard University [1].

This project follows a methodical approach to derive a systematic and user-friendly outcome in terms of a mobile application.

3.2. Rich Picture

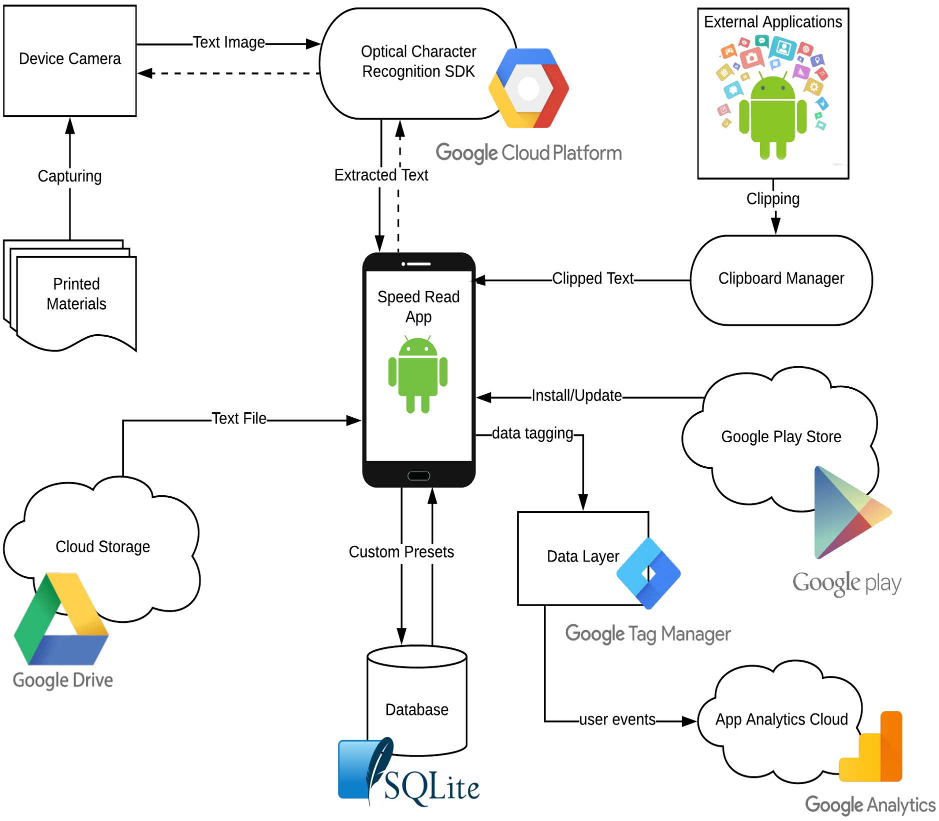


Figure 3.1. High level diagram of the mobile application.

Figure 3.1. High level design provides an overview of the system. It is decomposing system / application into modules and representing the interface and relationship among the modules. The design consisted input layer sub component such as detect preloaded templates, clipping text share into the reader view, text image from phone camera using OCR scanning, text file extraction from google drive.

3.3. Project Strategy

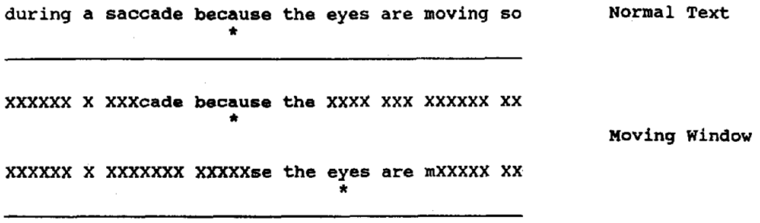
This mobile application consists of text loading and reading mechanisms. First objective was to implement a simulator on Android screen which automatically simulates the reading according to the pre-defined set of configurations. Here, first obvious approach is to use native Android Text View customized to support represent text contents. According to the initial analysis, I was able to discover a few limitations of that like inability to load large text contents, poor support to rich texts, limited variety of fonts and formats, lagging when scrolling a huge content.

To solve this problem, and Android WebView has been used with the support of a JavaScript interfaces which has been bound to the application. This has the ability to represent any type of text formats with a higher simulation capability. Then a smoothly moving bubble has been implemented using a JavaScript based animation method to ensure that the user gets an uninterrupted experience.

Practice any text from input text can make multiple sources using preloaded text, external text share, text scanning and open files from Google Drive. Preloaded text should be stored with preloaded template. External share only link with text and choose reader app to share text for practice. Scan texts from any printed sources, convert images has considered using an aid of an open source OCR library.

3.4. Algorithmic Analysis

The various tasks examined respect to reading are displaying changes in the read text support for mixed control model of eye movements in reading. The results were made as a window of text moved in synchrony with the eyes. One condition, size of the window was constant from fixation to fixation. Other condition, size of the window varied from fixation to fixation.



```
during a saccade because the eyes are moving so
                          *

XXXXXX X XXXcade because the XXXX XXX XXXXXX XX
                          *

XXXXXX X XXXXXXX XXXXXse the eyes are mXXXXX XX
                                *
```

Normal Text
Moving Window

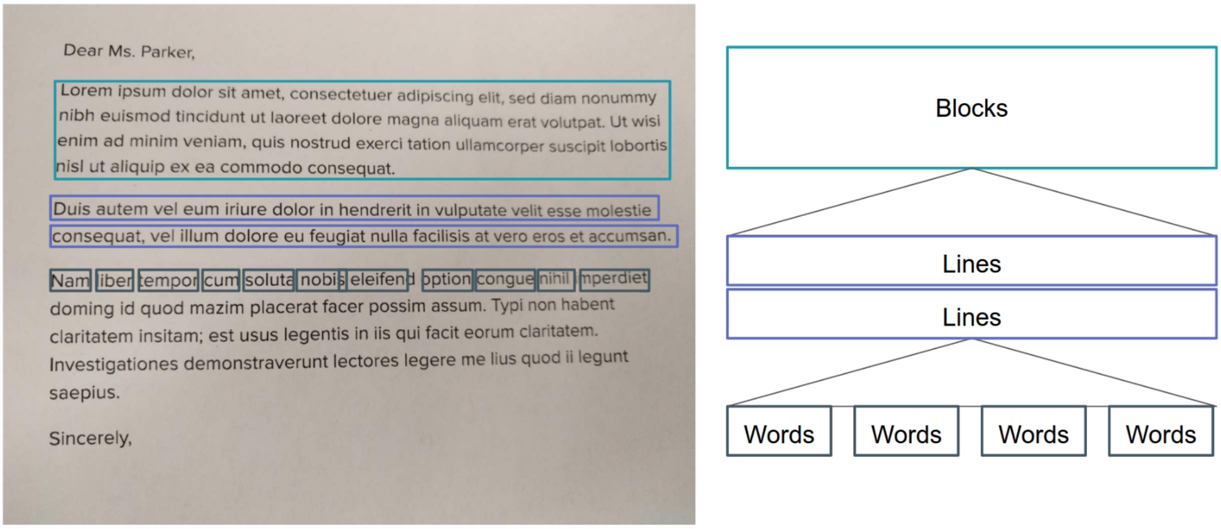
Figure 3.2. Sample window pattern.

Figure 3.2. The first line shows normal line of text with fixation location marked by an asterisk. The next two lines show an example of two successive fixations with other letters replaced with Xs and the spaces between words preserved. When the eye movement change occurs during the saccade so that the reader is not aware of the change.

3.5. Text Recognition

Optical Character Recognition (OCR) is a field of pattern recognition, artificial intelligence and machine learning. OCR is converted digital image to editable text. There were several OCR engines such as Google Vision, Tesseract, ABBYY (paid) and Aspire. OCR using Google Vision API for recognizing text from camera preview. Google Vision was the most accurate open source OCR engine. It works on Linux, Windows and MAC OS X. Also, it allowed to be used Android and iPhone platforms.

Text Recognition API can read a variety of images and translate text in over 18 Latin based languages (English, French, German, Swedish, Spanish, Italian, Latin, Turkish etc.). It's currently developed by Google [10].



Dear Ms. Parker,

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Sincerely,

Figure 3.3. Test Recognizer segments text structure.

Figure 3.3. The first highlighted "Block" is a contiguous set of text lines, such as a paragraph or column. The second set of highlighted a "Line" is a contiguous set of words on the same vertical axis. Finally, the third set of highlighted a "Word" is a contiguous set of alphanumeric characters on the same vertical axis.

3.6. Case Study

3.6.1. Use Case Diagram

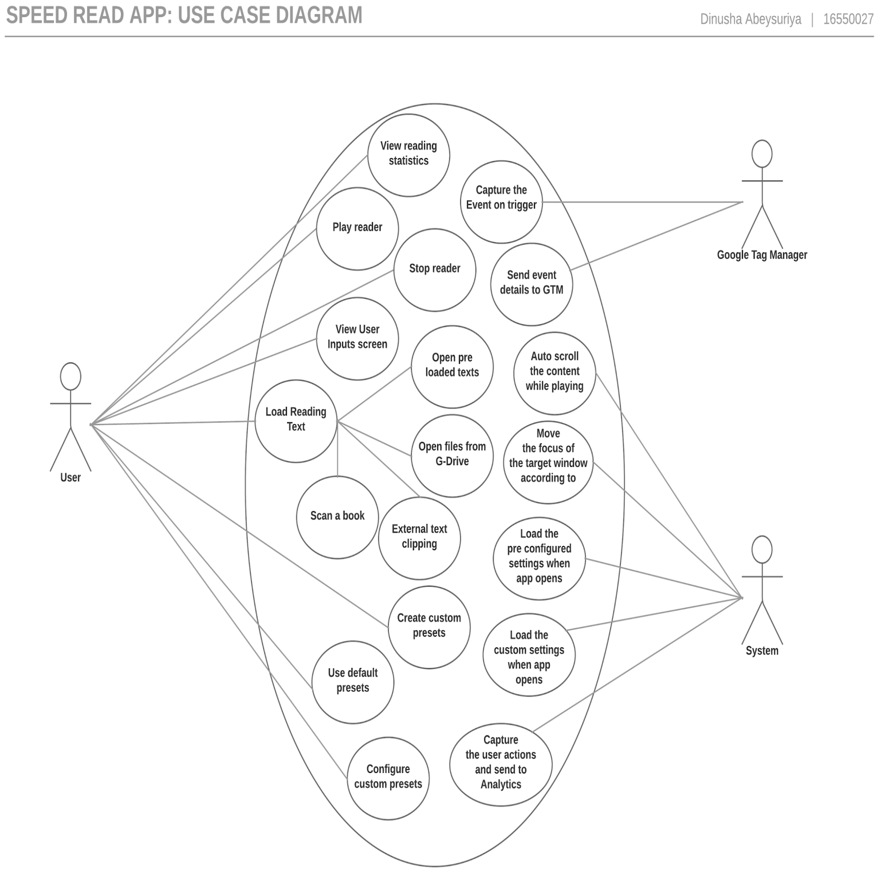

SPEED READ APP: USE CASE DIAGRAM
Dinusha Abeysuriya | 16550027
View reading statistics
Capture the Event on trigger
Play reader
Stop reader
Send event details to GTM
Google Tag Manager
View User Inputs screen
Open pre loaded texts
Auto scroll the content while playing
User
Load Reading Text
Open files from G-Drive
Move the focus of the target window according to
Scan a book
External text clipping
Load the pre configured settings when app opens
System
Create custom presets
Load the custom settings when app opens
Use default presets
Capture the user actions and send to Analytics
Configure custom presets

Figure 3.4. Use case diagram of the mobile application.

Figure 3.4. The use case diagram of mobile application. There main roles including one user role, two internal and external system roles can be found. Google Tag Manager is the system roles which listening to the user's action and do the logging to Google Analytics.

Use cases initiate by user : View reading statistics (Avg. words per minute/ Windows per line/ Total running time), Play reader, Stop reader, View User Inputs screen, Open pre-loaded texts, Open files from G-Drive, Scan a book, External text clipping, Use default presets, Create custom presets (window method/ word method), Configure custom presets (speed, font sizes, font family, windows per line, overlapping characters).

Use cases initiated by system : Auto scroll the content while playing, Move the focus of the target window according to the given stats, Load the pre-configured settings when app opens, Load the custom settings when app opens, Capture the user actions and send to Analytics, Capture the event trigger, Send event details to GTM

3.6.2. Class Diagram

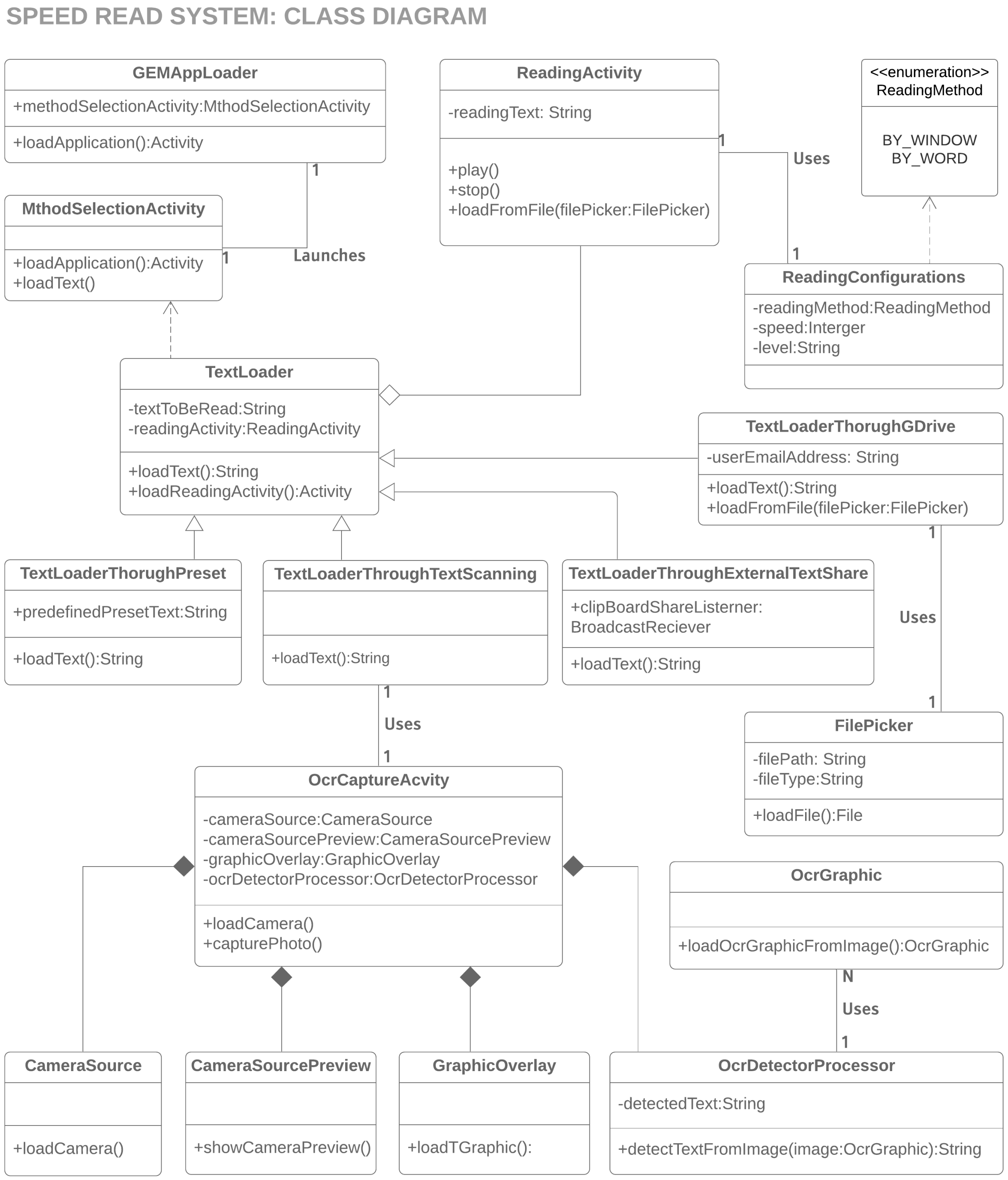

SPEED READ SYSTEM: CLASS DIAGRAM
GEMAppLoader
+methodSelectionActivity:MthodSelectionActivity
+loadApplication():Activity
ReadingActivity
-readingText: String
+play()
+stop()
+loadFromFile(filePicker:FilePicker)
<<enumeration>>
ReadingMethod
BY_WINDOW
BY_WORD
1
Uses
1
1
MthodSelectionActivity
+loadApplication():Activity
+loadText()
1
Launches
ReadingConfigurations
-readingMethod:ReadingMethod
-speed:Interger
-level:String
TextLoader
-textToBeRead:String
-readingActivity:ReadingActivity
+loadText():String
+loadReadingActivity():Activity
TextLoaderThorughGDrive
-userEmailAddress: String
+loadText():String
+loadFromFile(filePicker:FilePicker)
1
Uses
1
TextLoaderThorughPreset
+predefinedPresetText:String
+loadText():String
TextLoaderThroughTextScanning
+loadText():String
TextLoaderThroughExternalTextShare
+clipBoardShareListerner:
BroadcastReciever
+loadText():String
1
Uses
1
FilePicker
-filePath: String
-fileType:String
+loadFile():File
OcrCaptureAcvity
-cameraSource:CameraSource
-cameraSourcePreview:CameraSourcePreview
-graphicOverlay:GraphicOverlay
-ocrDetectorProcessor:OcrDetectorProcessor
+loadCamera()
+capturePhoto()
OcrGraphic
+loadOcrGraphicFromImage():OcrGraphic
N
Uses
1
CameraSource
+loadCamera()
CameraSourcePreview
+showCameraPreview()
GraphicOverlay
+loadTGraphic():
OcrDetectorProcessor
-detectedText:String
+detectTextFromImage(image:OcrGraphic):String

Figure 3.5. The class diagram of the mobile application.

Figure 3.5. 'ReadingConfiguration' class is detected two types of Reading Methods. These methods do not depend on each other. Window method algorithm is calculated the research from pattern of eye movements in reading. Word method capture highlighted word part will move through the reader board. 'TextLoader' all the text input functionalities are managed by this class. ‘ReadingActivity’ reading process done by this class. Manage play, stop changing mechanism. ‘ReadingConfiguration’ similarity measurement is done by this class. Highlighted text is detected by using this class.

3.6.3. Basic Feature/Functionality Breakdown

Input Methods are preloaded text, External Text Share, Text Scanning, Open file from G- Drive. App Preferences are window Method (Researched and recommended), Word Method (Game feature), Default Presets, Custom Presets, Settings, Window Adjustments are number of windows per line, Overlapping characters, Reading speed (Windows per min/ Words per min), Font Adjustments are font size, font family, save adjustments. Reader View displays reader stats board and reading layout. Reader stats board contains Avg. words per minute, Windows per line and Total running time. Reading layout contains Auto scrolling, Smooth animations, Blur and bubble effects to gain more focus and Readable background and font colors.

3.7. Project Approach

Agile methodology was followed during the entire project. At the end of each sprint a demo will be presented to the product manager(supervisor) as User Acceptance testing DOD. Each requirement will be considered as completed after a successful implementation, Dev testing, QA testing and UAT.

3.8. Methodological Design

3.8.1. Input Variations

Android native applications uses few row text input methods other than direct inputs. When it comes to reading apps, some of them were frequently used according to the reviews and stats in Google Play store. When designing this application, a lot of attention has been put to ensure that it covers those important input methods. As the initial stage, four input methods have been added as follows.

1. Share text from external

Number of apps installed in particular device has been increased according to the current stats of global mobile usage. In such situations, user might need to use texts from other applications for his practice sessions. As an example, New York Times application is the widely used newspaper application which contains a lot of text contents which might get users attention. SpeedRead application has been designed with the capability to extract any text content from such external applications so easily.

It was a challenge to decide how users can get those texts to the SpeedRead app so easily and accurately. Few options came across during the designing phase were, giving the user the option to copy and paste the content, print the screen and extract the text, integrate a google search to the application and share the text using a note clipper. Sharing the text using a global note clipper was chosen based on few important facts like a smaller number of taps need to be performed, flexibility to use in any application, easy global access anywhere within the device, use of inbuilt Android share intents. As expected, this was the most frequently used input method in SpeedRead app among users.

2. Pre-loaded texts

This input method facilitates the beginners among the users. A new user to the application might not have a clear understanding about how this app works. Some pre-loaded texts have been included in the application which gives the user a quick start. With these text users can quickly start practicing and getting to know about the core features. Even though this is the direct benefit of this feature, there are some other indirect benefits when considering the usage. By giving the fast access to the user, he can understand the core features in no time. This will help to keep the user's interest with the application and to avoid the possibility of quick uninstallations which occurs in most applications due to the miss guidance and time consumption.

3. Open from Google Drive

Google has become a famous tech giant nowadays especially in data storing and manipulation. According to the statistics, Google Drive is the place where Android users used to store data more frequently. So, there's a high possibility to have stored text contents in Google Drive for a particular user. Based on that assumption, a mechanism to load data from Google Drive application has been designed as a main input method in this application.

4. Scan a book

Even though all the above text input mechanisms cover the basic usage of a mobile user with accepted level of technical literacy, there are considerable amount of readers around the world who still use books as their main reading media. So that this can't be ignored so easily when designing this kind of application specially focused on improving reading skills. A mechanism to scan a book and extract the texts from it, has been designed to facilitate this specific usage. There, an OCR has been integrated to the application which can scan a book or newspaper on top of the live camera feed.

There were plenty of OCR libraries available in Android market so that it was a bit difficult to find the most suitable one for this application design. Given the fact that most frequently used and trusted library for Android is the Google OCR comes under Google Vision libraries. This has a high capability of scanning texts even under low light conditions. In addition, when considering the application design, this integrated component should have a low coupling with the other core modules of the application. As an input method, this is not a core functionality of the application, so this has been designed to be easily unplugged with a minimum effort. Google had been developed a sdk for the OCR capability so that it was the most suitable way of achieving this goal when designing.

3.8.2. Application Components

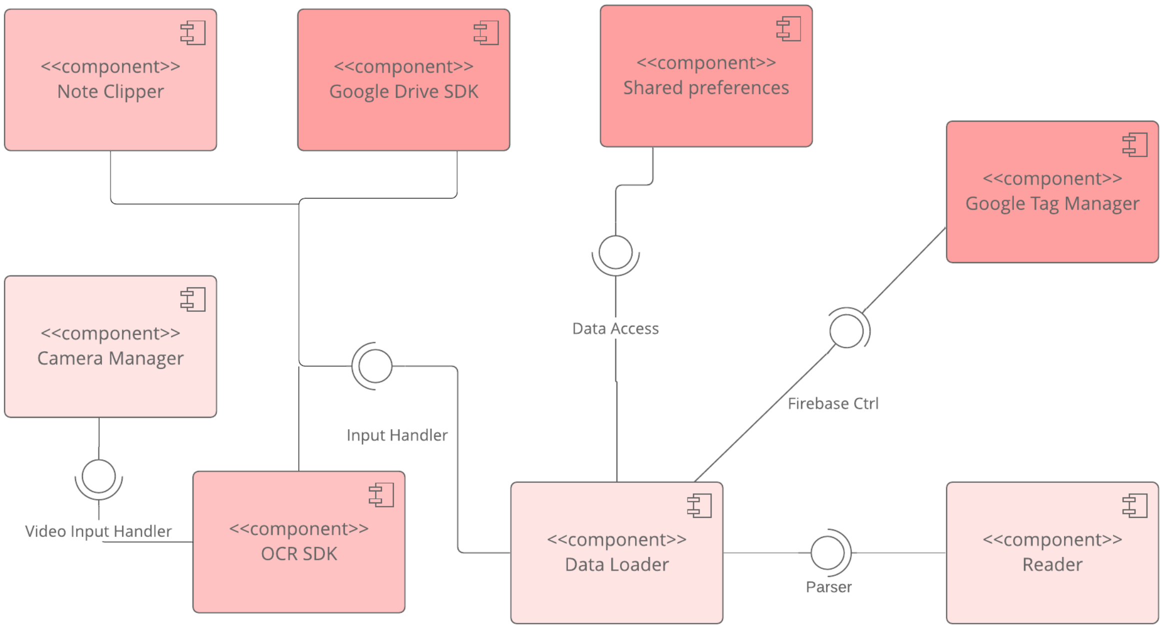


Figure 3.6. The component diagram of the mobile application.

Figure 3.6. The entire application has been designed as a collection of key components which has a low coupling in order to facilitate a few key development goals like incremental development, easy maintenance, light weight, code readability, etc. Each component has given a set of responsibilities carefully designed with appropriate method interfaces which can be easily pluggable with other components.

3.8.3. Reading View Design

When designing the reading view the biggest challenge was to find a suitable UI component to hold the text content. In Android design practices, there were no such direct recommendations can be found for this kind of scenarios, since this has been considered as an edge case in typical Android development. In a general mobile application, a sample text or label contains hundred to five hundred characters in maximum. Although from the nature of this application the size of the text was unknown, and it can change from hundreds to thousands.

Few well known UI components has been tested against this requirement such as Edit-Text, Text-View, etc. However, most of them worked well and efficiently under low level of character capacity but failed perform under large scale text contents. Those components seemed to be lagging and freezing the UI which caused sudden application terminations. With those findings Android WebView has been chosen by considering its ability to behave normally under high level of character capacity and more clarity given by large variety of font styles and sizes. When using WebView, few difficulties also faced such as lack of smooth scrolling capability, common security vulnerabilities and poor communication between other components.

Since the WebView behaves as browser client, it was not straight forward to communicate with the rest of the native components. In order to overcome those mentioned problems, a JavaScript interface has been designed as the communication media between the WebView and the other native components.

3.8.4. Data Storing Mechanism

Since Android OS is intended to be lightweight, there are no local database management systems available in the market. Instead lightweight databases based on single file data storing mechanisms are available such as SQLite and Realm. This application is designed to store only presets data and sample text data in a very lightweight manner. So that as recommended the fastest way was to store it in shared preferences as serialized objects or primitives.

In terms of security, Android guarantees that shared preferences are stored in a private location where any external application except the operating application won't be able to read or modify the data. There was another fact which led the application to this particular design which was the data insensitivity. The data this application processes is not sensitive since they are just public text contents non application level settings.

3.8.5. Readability and Reliability

The main programming language of the project is Android. Also, all functionality & coding standards were written in java. The importance of coding standards was can easy to debug & maintain the project. Also, it increases the quality of the project and quality of the code. Android is an open source software tool kit for mobile devices as to create by Google and Open Handset alliance.

3.8.6. Bundling and Performance

This application has been bundled using the newest apk bundling technology recently introduced by Google. The Android App Bundle is Android's new, official publishing format that offers a more efficient way to build and release applications. The Android App Bundle lets you more easily deliver a great experience in a smaller app size, which can improve install success and reduce uninstalls. It's easy to switch. Refactoring the code is not needed to start benefiting from a smaller app. And once switched, it'll benefit from modular app development and customizable feature delivery.

3.8.7. Execution Design

This application contains a sequential execution process starting from data input initiation to start reader view.

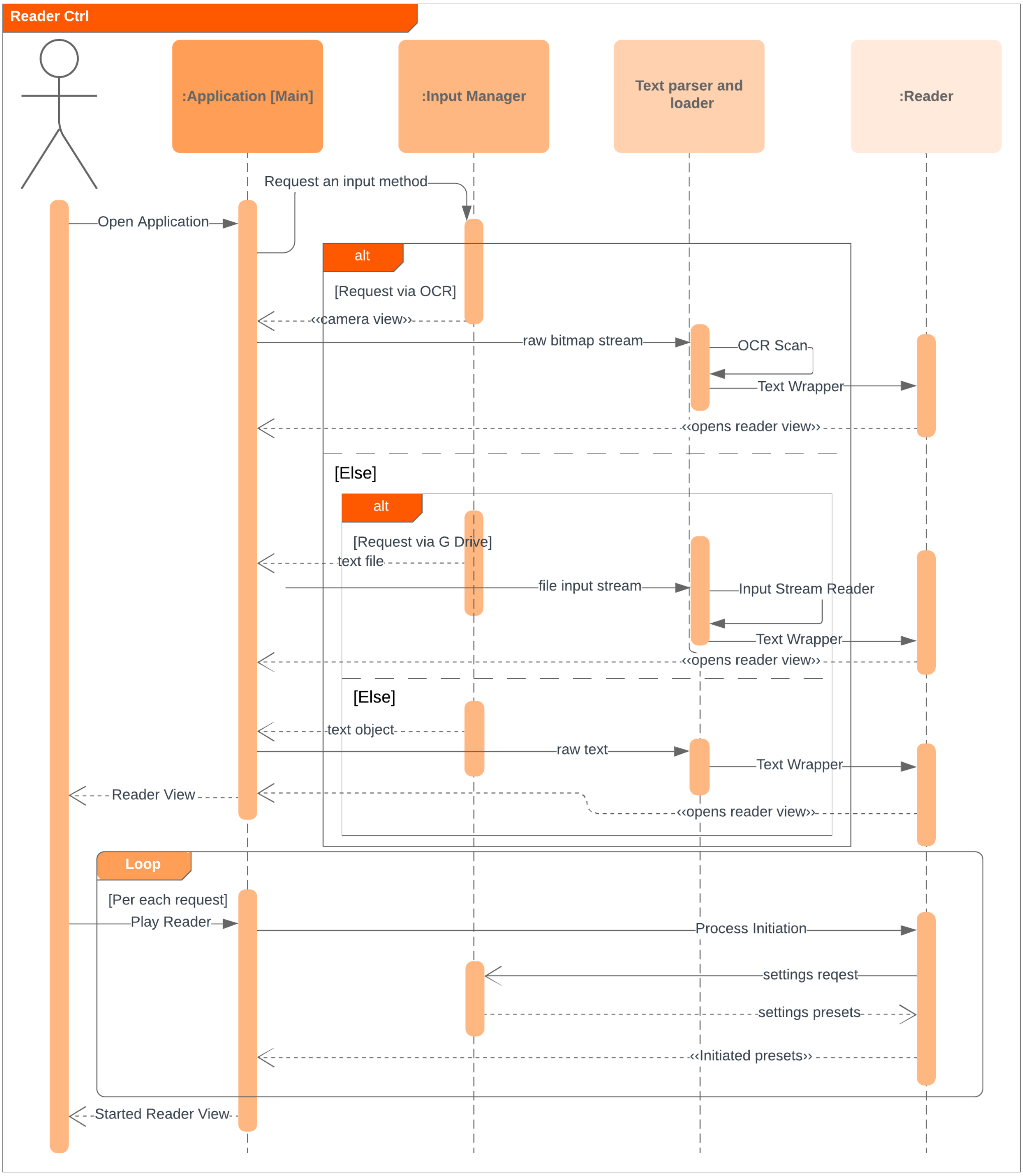


Figure 3.7. The sequence diagram of the mobile application.

3.9. Software Development and Deployment Methodology

3.9.1. CI & CD (Continuous Integration and Continuous Deployment)

A repository will be managed to track versions. A main branch will be managed to keep the ready to release code. And a separate branch will be created for each feature or user story. At the end of each sprint, the created feature branches will be integrated to the main branch after a successful testing. At the end of each sprint a release apk will be shared from the main branch as the deployment with the product manager (supervisor).

3.9.2. Deployment

At the end of the project an apk file will be released which can be deployable in any android device from OS version 6.01 (Marshmallow) above.

3.9.3. Analytics

Google Analytics and Google Tag Manager will be integrated to the application and instrumented with the major functionalities to capture the user interactions during the pilot tests.

3.10. Evaluation Methodology

Prototype Evaluation is release of product built to test a concept or release apps and update more frequently. Wireframe utilizes to visualize the app at a very basic level, fully functional prototype ensure UX is consistent and After the user evaluation, another prototype will be built based on feedback from users.

“Beta Testing” pre-release for user testing, each phase of user validation is unique evaluation and feedback of product. During this period can find how many issues are discovered and how many new features are released. Screens and Events Analysis reports are set of functionality analysis reports will be presented via Google Analytics to understand the audience and user behaviors. Dev Testing will be done at the successful completion of each implemented story. QA Verification Testing apply once dev testing is completed a set of test cases will be written and tested them against the implemented and dev tested functionality.

Integration Testing at the end of each sprint, the implemented, dev tested, and QA verified functionality of user stories will be merged into the main development branch. After resolving conflicts, the entire code base with newly integrated code will be tested again against the test cases. User Acceptance Testing at the end of each sprint an apk file will be released to the product manager(supervisor) for UAT. After the testing a set of feedback will be gathered and addressed as a set of bugs or new user stories within the next immediate iteration.

Regression Testing two main releases will be scheduled during the project development period. At each release stage, a regression test will be conducted against the entire functionality. Pilot Testing using the closed beta concept, the released application will be deployed via a Google Plus community for a target set of users. Beta Testing using the open beta concept the final application will be released to Google Play community to test and gather feedback.

3.11. Ethical considerations

This project is developed based on a research conducted by Harvard University [2]. However, this has carefully excluded, copying the content, copying the approach, copying the domain and selling the outcome. This application will be free to use by anyone through Google Play stores. The original research outcome was a flash video which runs a reading simulation. Although this project delivers a mobile app which allows users to read anything as they wish. Furthermore, methodical approaches are completely different in terms of outcome and technologies used as the original was developed using Adobe Flash while this mobile application is developed using Native Android.

3.12. Project Limitations

3.12.1. Accuracy Level of OCR

This app will be using Google Vision as the OCR library which is the best product in the market at this time for mobile applications. This library has been introduced by Google and the level of accuracy can be expected as stated by them.

3.12.2. Clipboard Limitations

Clipboard Managers has the capability to keep not only the text data but also rich text formats as well as images. Only the plain texts coming through the clipboard will be accepted to use within the application and rich text formats or text with images will be accepted after extracting the text from it.

3.12.3. Types of Documents

Pdfs and other types of documents which may contain images will not be allowed to use within the application. Only the text files will be accepted through the Google Drive.

3.13. Conclusion

The research methodology helps to learn how the enhancements has been done in terms of development and technology to give a better user experience based on a valuable research outcome. Helps to understand the attitude of others and creates awareness of special needs of the research process.

4. Evaluation

The overall approach to evaluating mobile apps on contents of materials and user evaluations. The reason to choose android platform current market nearly used half of the android smartphone's users. This app is suitable for both children and adults as stated.

4.1. Evidences

The entire project android application of the "Speed Read" UI and UX development will be analyzed to get the solution of proposed application.

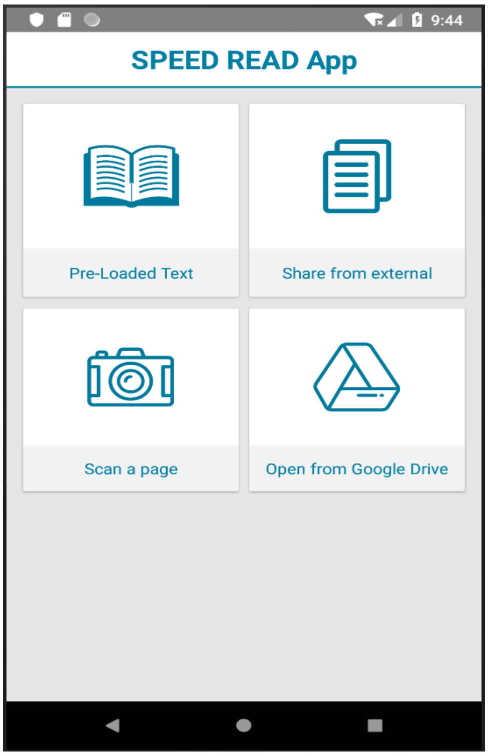
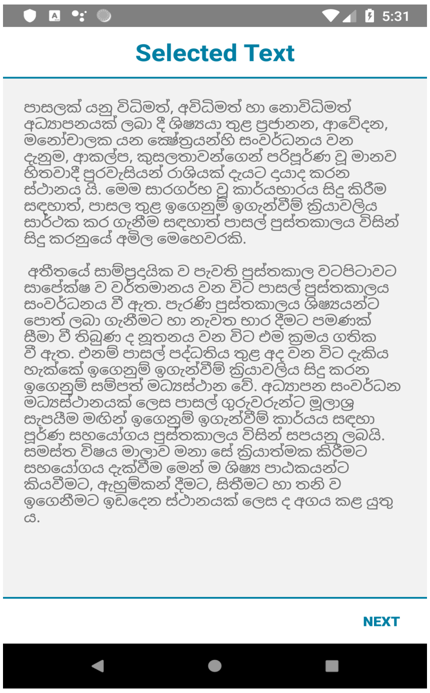


Figure 4.1.

- Main input dashboard is presented in a very attractive manner to the user.
- Color codes of the app gives a clear attention to the user.
- Number of options/clickable buttons are available in dashboard.
- Display four user inputs such as preloaded text, text scanning, open file from Google Drive and external text share.
- Selection able options are available, so when it's click user to easily redirect to the other pages.

Figure 4.1. Main user interface of the mobile application.



Selected Text

පාසලක් යනු විධිමත්, අවිධිමත් හා නොවිධිමත් අධ්‍යාපනයක් ලබා දී ශිෂ්‍යයා තුළ ප්‍රජානන, ආවේදන, මනෝචාලක යන ක්ෂේත්‍රයන්හි සංවර්ධනය වන දැනුම, ආකල්ප, කුසලතාවන්ගෙන් පරිපූර්ණ වූ මානව හිතවාදී පුරවැසියන් රාශියක් දැයට දායාද කරන ස්ථානය යි. මෙම සාරගර්භ වූ කාර්යභාරය සිදු කිරීම සඳහාත්, පාසල තුළ ඉගෙනුම් ඉගැන්වීම් ක්‍රියාවලිය සාර්ථක කර ගැනීම සඳහාත් පාසල් පුස්තකාලය විසින් සිදු කරනුයේ අමිල මෙහෙවරකි.

අතීතයේ සාම්ප්‍රදායික ව පැවති පුස්තකාල වටපිටාවට සාපේක්ෂ ව වර්තමානය වන විට පාසල් පුස්තකාලය සංවර්ධනය වී ඇත. පැරණි පුස්තකාලය ශිෂ්‍යයන්ට පොත් ලබා ගැනීමට හා නැවත භාර දීමට පමණක් සීමා වී තිබුණ ද නූතනය වන විට එම ක්‍රමය ගතික වී ඇත. එනම් පාසල් පද්ධතිය තුළ අද වන විට දැකිය හැක්කේ ඉගෙනුම් ඉගැන්වීම් ක්‍රියාවලිය සිදු කරන ඉගෙනුම් සම්පත් මධ්‍යස්ථාන වේ. අධ්‍යාපන සංවර්ධන මධ්‍යස්ථානයක් ලෙස පාසල් ගුරුවරුන්ට මූලාශ්‍ර සැපයීම මඟින් ඉගෙනුම් ඉගැන්වීම් කාර්යය සඳහා පූර්ණ සහයෝගය පුස්තකාලය විසින් සපයනු ලබයි. සමස්ත විෂය මාලාව මනා සේ ක්‍රියාත්මක කිරීමට සහයෝගය දැක්වීම මෙන් ම ශිෂ්‍ය පාඨකයන්ට කියවීමට, ඇහුම්කන් දීමට, සිතීමට හා තනි ව ඉගෙනීමට ඉඩදෙන ස්ථානයක් ලෙස ද අගය කළ යුතු ය.

Figure 4.2.

- If user taps “Pre-Loaded Test”, Pre-loaded template should be shown in the Reader view.
- Input text can be any language.
- WebView for reader view instead of custom HTML page.
- Reader View which accept any number of characters in the language which introduced by native WebView.
- Apply custom CSS styles to fine tune the reader view.
- JavaScript function to perform actions on the reader view.

Figure 4.2. Selected text view user interface of the mobile application

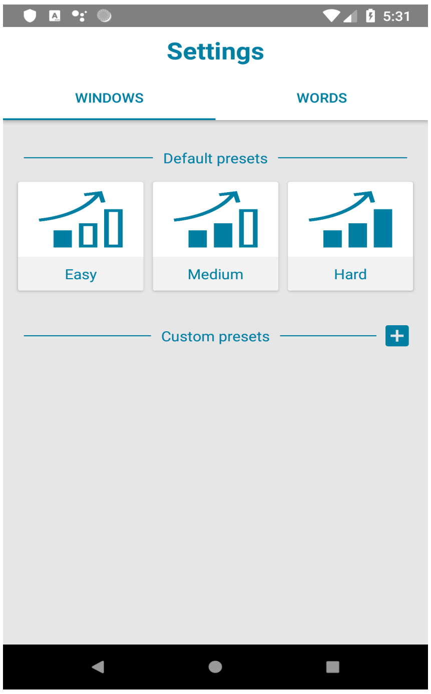
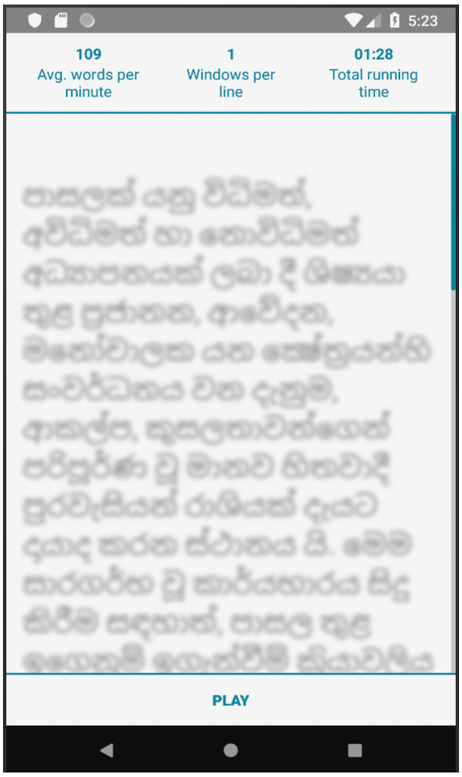


Figure 4.3.

- Create custom presets (window method, word method).
- Configure custom presets (speed, font sizes, font family, windows per line, overlapping characters).

Figure 4.3. Setting user interface of the mobile application.



109 Avg. words per minute

1 Windows per line

01:28 Total running time

Figure 4.4.

- Implementing reader view, auto scroll the content while moving of the reading bubble as a window.
- View reading statistics (Avg. words per minute / Windows per line / Total running time).

Figure 4.4. Reader View user interface of the mobile application.

4.2. Feature Evaluation

Following are the major features provided in the app.

1. Reader View

Reader view has implemented to clearly view the text content from any supported source. Furthermore, this is the most important of this application as this allows the reader to practice reading. As mentioned in the project objectives, this Reader View has enough performance to load the content very quickly. In addition, smooth scrolling is performing well without giving the user an experience of lagging the content. In addition, it has a carefully designed typography which makes clearer the reading and soften the eyes.

Functionality of this feature has been evaluated in few phases like dev testing, integration testing and regression testing. Only few minor bugs were found like a slowness when loading a massive content and not supporting the rich texts. All of them have been addressed and retested along with the requirement. Usability of this feature will be evaluated under all the usability testing phases. Such as the reported user experience will be addressed and developed as feature enhancements.

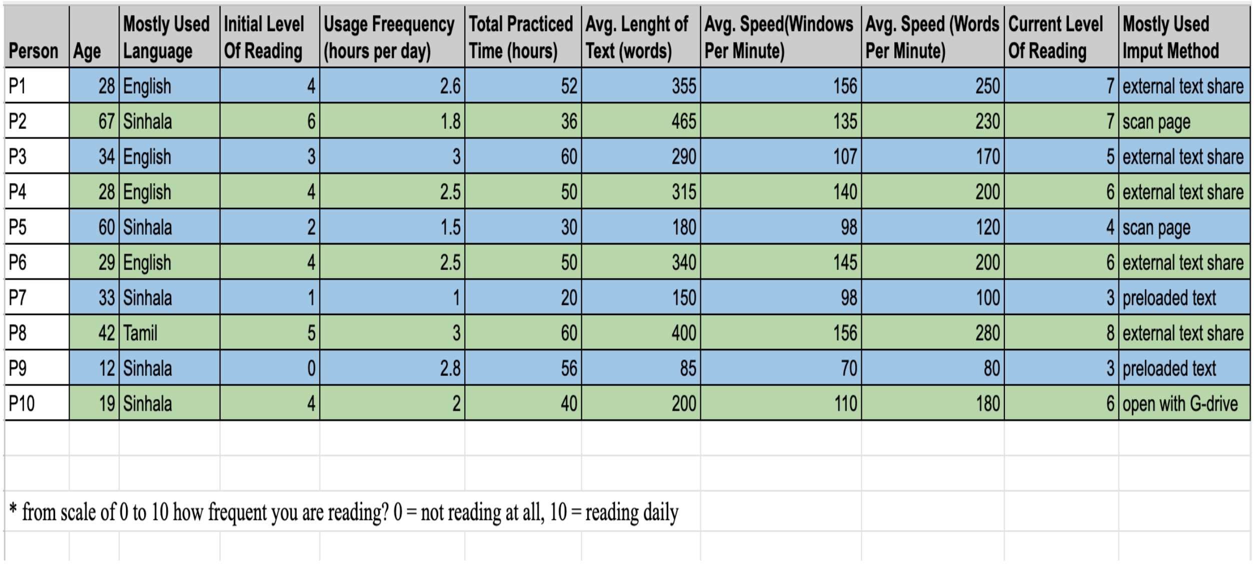
2. Source Selector

This allows user to grab text content in many ways. This is the next major feature which has been introduced to select a text and practice. Instead of giving a one-way text injection, four major text injection methods have been introduced under this feature, clipping text from external application, select text files from Google Drive, scan a book and extract the text and pre-loaded content which is ready to be used. All of these input mechanisms have been tested and the bugs were reported during the testing phases like dev testing, integration testing and regression testing. The reported bugs like unable to create and save custom presets has been addressed successfully in the final version. Usability of this feature will be evaluated under all the usability testing phases. Furthermore, the reported user experience will be addressed and developed as feature enhancements.

3. Adjustment Controller

When running the selected text content, users will be able to adjust the properties of practice session and save them as custom presets. Change window size, moving speed, font size, font family, windows per line, words per minute and many more. This gives the chance to the user to improve his reading by enhancing his reading skills. This feature was the hardest one to test since it has considerable amount of combinations. Along with selected text content all the adjustments were tested during the testing phases like dev testing, integration testing and regression testing. Furthermore, this part will be thoroughly evaluated under usability testing and keep continuous improvement. One biggest challenge faced here is the way of present and arranging the controllers. UX design was more useful here than the UI design since this has many options but frequently used. During the testing period, few bugs related to the adjustments were identified and all of them were fixed successfully.

4.3. APP Usage Statistics

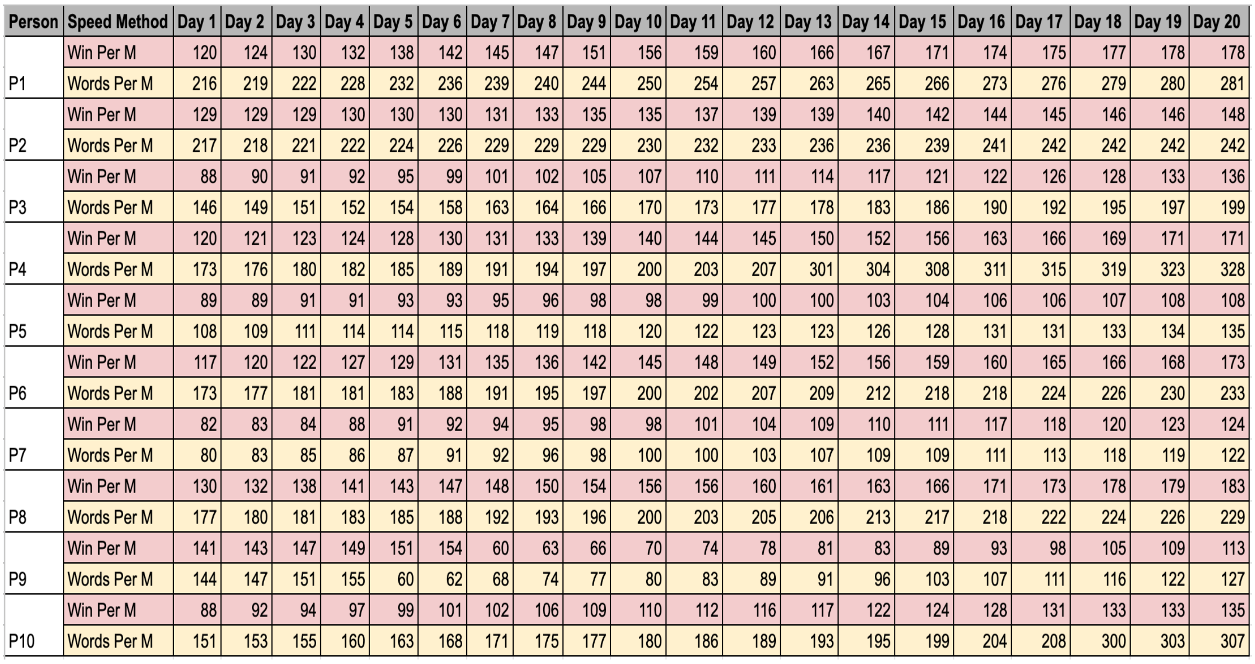


Person	Age	Mostly Used Language	Initial Level Of Reading	Usage Freequency (hours per day)	Total Practiced Time (hours)	Avg. Lenght of Text (words)	Avg. Speed(Windows Per Minute)	Avg. Speed (Words Per Minute)	Current Level Of Reading	Mostly Used Imput Method
P1	28	English	4	2.6	52	355	156	250	7	external text share
P2	67	Sinhala	6	1.8	36	465	135	230	7	scan page
P3	34	English	3	3	60	290	107	170	5	external text share
P4	28	English	4	2.5	50	315	140	200	6	external text share
P5	60	Sinhala	2	1.5	30	180	98	120	4	scan page
P6	29	English	4	2.5	50	340	145	200	6	external text share
P7	33	Sinhala	1	1	20	150	98	100	3	preloaded text
P8	42	Tamil	5	3	60	400	156	280	8	external text share
P9	12	Sinhala	0	2.8	56	85	70	80	3	preloaded text
P10	19	Sinhala	4	2	40	200	110	180	6	open with G-drive

* from scale of 0 to 10 how frequent you are reading? 0 = not reading at all, 10 = reading daily

Figure 4.5. Test sample data usage - reading statistics.

Figure 4.5. When considering the reading level, which was a feedback given by the users before they start practicing with application and after performing a 20 days of practicing period, their reading level has been improved or the less. The reading level is defined from a scale of 0 to 10 which shows how frequent you are reading where 0 represents not reading at all and 10 represents reading daily.



Person	Speed Method	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20
P1	Win Per M	120	124	130	132	138	142	145	147	151	156	159	160	166	167	171	174	175	177	178	178
	Words Per M	216	219	222	228	232	236	239	240	244	250	254	257	263	265	266	273	276	279	280	281
P2	Win Per M	129	129	129	130	130	130	131	133	135	135	137	139	139	140	142	144	145	146	146	148
	Words Per M	217	218	221	222	224	226	229	229	229	230	232	233	236	236	239	241	242	242	242	242
P3	Win Per M	88	90	91	92	95	99	101	102	105	107	110	111	114	117	121	122	126	128	133	136
	Words Per M	146	149	151	152	154	158	163	164	166	170	173	177	178	183	186	190	192	195	197	199
P4	Win Per M	120	121	123	124	128	130	131	133	139	140	144	145	150	152	156	163	166	169	171	171
	Words Per M	173	176	180	182	185	189	191	194	197	200	203	207	301	304	308	311	315	319	323	328
P5	Win Per M	89	89	91	91	93	93	95	96	98	98	99	100	100	103	104	106	106	107	108	108
	Words Per M	108	109	111	114	114	115	118	119	118	120	122	123	123	126	128	131	131	133	134	135
P6	Win Per M	117	120	122	127	129	131	135	136	142	145	148	149	152	156	159	160	165	166	168	173
	Words Per M	173	177	181	181	183	188	191	195	197	200	202	207	209	212	218	218	224	226	230	233
P7	Win Per M	82	83	84	88	91	92	94	95	98	98	101	104	109	110	111	117	118	120	123	124
	Words Per M	80	83	85	86	87	91	92	96	98	100	100	103	107	109	109	111	113	118	119	122
P8	Win Per M	130	132	138	141	143	147	148	150	154	156	156	160	161	163	166	171	173	178	179	183
	Words Per M	177	180	181	183	185	188	192	193	196	200	203	205	206	213	217	218	222	224	226	229
P9	Win Per M	141	143	147	149	151	154	60	63	66	70	74	78	81	83	89	93	98	105	109	113
	Words Per M	144	147	151	155	60	62	68	74	77	80	83	89	91	96	103	107	111	116	122	127
P10	Win Per M	88	92	94	97	99	101	102	106	109	110	112	116	117	122	124	128	131	133	133	135
	Words Per M	151	153	155	160	163	168	171	175	177	180	186	189	193	195	199	204	208	300	303	307

Figure 4.6. Practiced record information about speed.

Figure 4.6. This data sample has been taken from ten different people in different age groups and different reading levels. The practice session has been conducted within twenty days from each reading mechanism. In each mechanism, most of them show a considerable improvement of reading speed, which proves that this application has been able to improve the reading speed in different age groups.

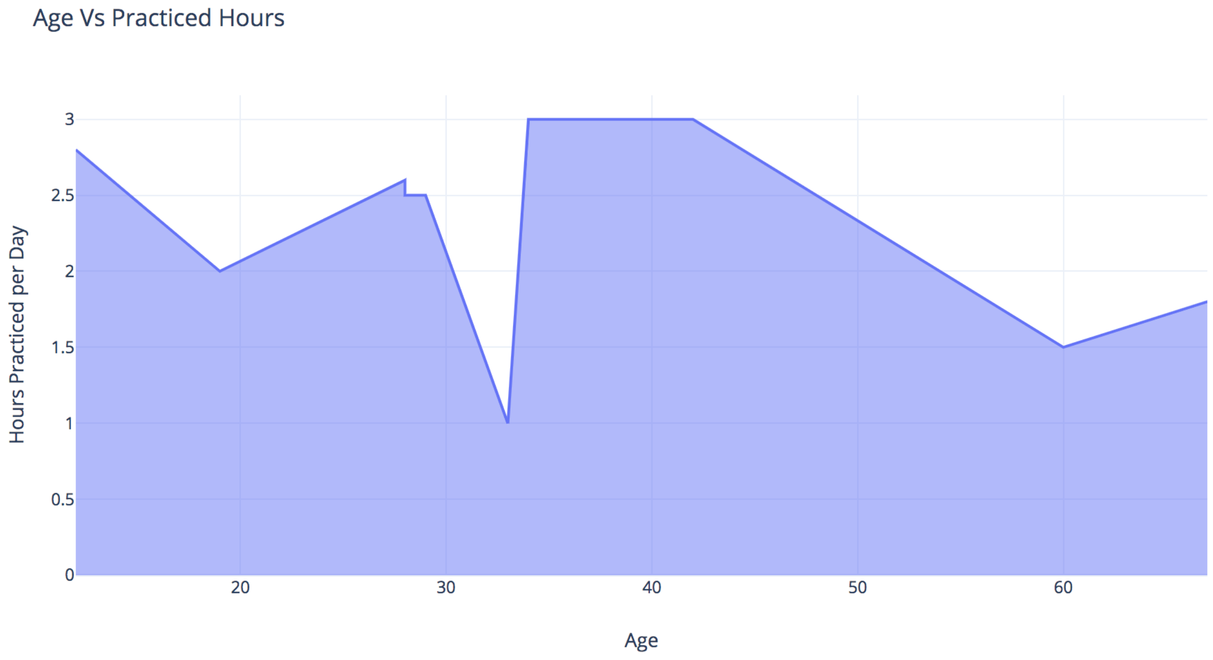


Figure 4.7. Visualization of Age and Practiced Hours.

Figure 4.7. This graph shows how the age groups spend their time on reading. This gives us a better idea of which age groups should be focused more with personalized features in future releases. When applying this in larger sample this will provide a clearer idea about how those age groups respond against their reading habits.

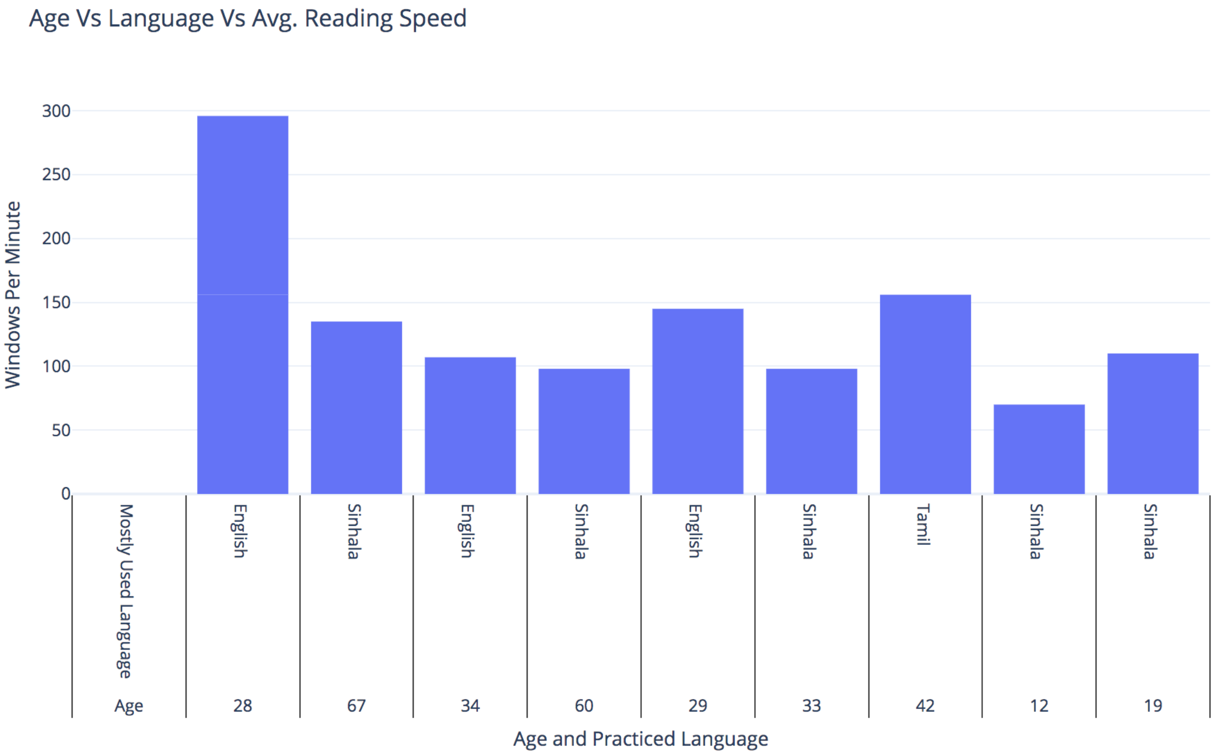


Figure 4.8. Visualization of Age and Practiced Language.

Figure 4.8. This has proven one of the major research findings which is the window segmentation is different between different age groups. Some of the age groups has shown an increased reading speed, in fact, their window width is a bit larger than the other age groups. So, they can read a larger number of characters within a single window segment. In other words, their reading is much faster compared to others.

Growth Of Reading Speed - Against Main Practicing Mechanisms Vs Users

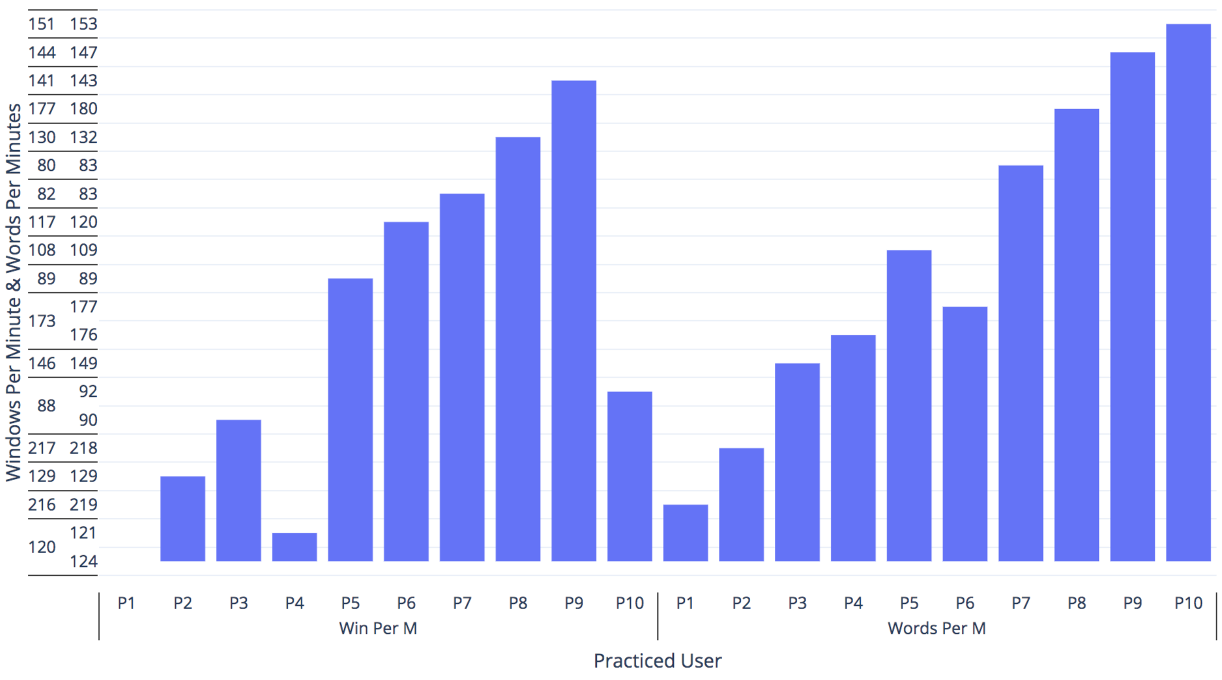


Figure 4.9. Growth of reading speed - against main practicing mechanism and users.

Figure 4.9. According to the above statistics, overall practicing of readers has been well performed and the reading speed has been improved in both reading mechanisms, window and word. Furthermore, the reading speed has been varied among different age groups.

4.4. Usability Evaluation

Usability evaluation has been planned to conduct in few ways once the app is launched. Firstly, the app will be launched to a selected set of users as a closed internal test track in Google Play. Once the testing is completed the same build will be promoted to alpha testing to the same user group. These are the first two approaches to be evaluated before the app is going public.

The next usability evaluation will be the Google Play open beta testing. Under this approach the app will reach the users publicly as a beta. Google Play will be notified the user as this app is under construction and asking to join the beta program as a public tester. There will be a separate mechanism to get users feedback on the fly via Google Play during the beta period where they'll be asked to share their experience with the app. All the recorded feedback will be analyzed carefully after a one-month period and they'll be addressed and released as a patch version. After a few cycles the beta testing will be completed, and the final build will be moved to the production.

Even though the app will be live and open for the users by this stage the usability evaluation will be continued to one more step forward. Google Analytics reporting platform is integrated to the application via Google Tag Manager live tagging support. This will be able to capture all the user’s actions on the app and send immediately to the Google Analytics reporting platform. This is a big data analysis platform introduced by Google so the data received will be processed and analyzed slowly and passed to the reporting platform. Using the reports, it’ll be easy to identify the hot areas of the apps where the users access more frequently and the areas, they get blocked. Based on these reports the existing features will be enhanced and new features will be introduced as a new version of the app.

4.5. Operating System Evaluation

Most popular mobile operating systems among the mobile users are Android and iOS. Majority of 95% has been taken place by those two operating systems. From that Android usage is given as 75% which is why it has been selected to run Speed Read application. When considering the remaining percentage which is iOS, should also be evaluated in terms of development, platform compatibility and current application market.

New top-end mobile phones such as Apple iPhone and iPad are allowing new classification of mobile to emerge in international market. Apple companies recommend selecting the iOS as a platform. Business development procedure must be fast therefore the device classification less compared with android. Furthermore, also, overall period of faster deployment for iOS app gets developed in order to limit iOS app to certain devices. So can take advantage of the development of the iOS app by displaying quality standard. Have rich experience in developing and delivery high quality iOS apps which make us better place to sell in the global market.

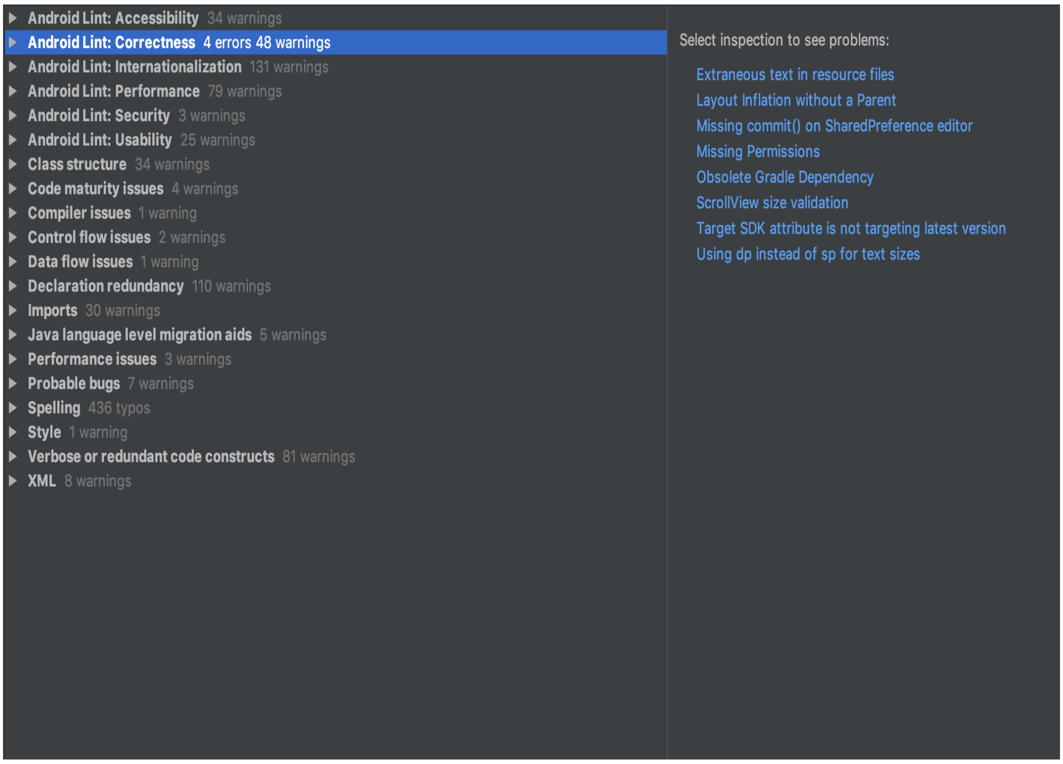
4.6. Problem Faced

Reader View is a sort of overlay that use to practice reading speed, which does not accept the functionality interfaces provided by Android native "TextView” or “EditText". Therefore TextView/EditText is showing poor performance on scrolling when loading large number of characters, Unable to move the window across multiple TextView/EditText and Customizing android native components will not be helpful. So, find another mechanism to bring up the Native WebView for exact solution of the requirement. Another problem is the textual issues having may not stem from the scanning, but from the text itself. These issues can be more difficult to solve therefore you cannot change the content of the original document. Although low contrast in original documents can be adjusted in a photo manipulation tool.

4.7. Testing Tools

1. Android Lint

Android Studio provides a code scanning tool called lint [10]. The lint tool checks Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization. The lint tool can manually run inspection in android studio.



▶ Android Lint: Accessibility 34 warnings
▶ Android Lint: Correctness 4 errors 48 warnings
▶ Android Lint: Internationalization 131 warnings
▶ Android Lint: Performance 79 warnings
▶ Android Lint: Security 3 warnings
▶ Android Lint: Usability 25 warnings
▶ Class structure 34 warnings
▶ Code maturity issues 4 warnings
▶ Compiler issues 1 warning
▶ Control flow issues 2 warnings
▶ Data flow issues 1 warning
▶ Declaration redundancy 110 warnings
▶ Imports 30 warnings
▶ Java language level migration aids 5 warnings
▶ Performance issues 3 warnings
▶ Probable bugs 7 warnings
▶ Spelling 436 typos
▶ Style 1 warning
▶ Verbose or redundant code constructs 81 warnings
▶ XML 8 warnings

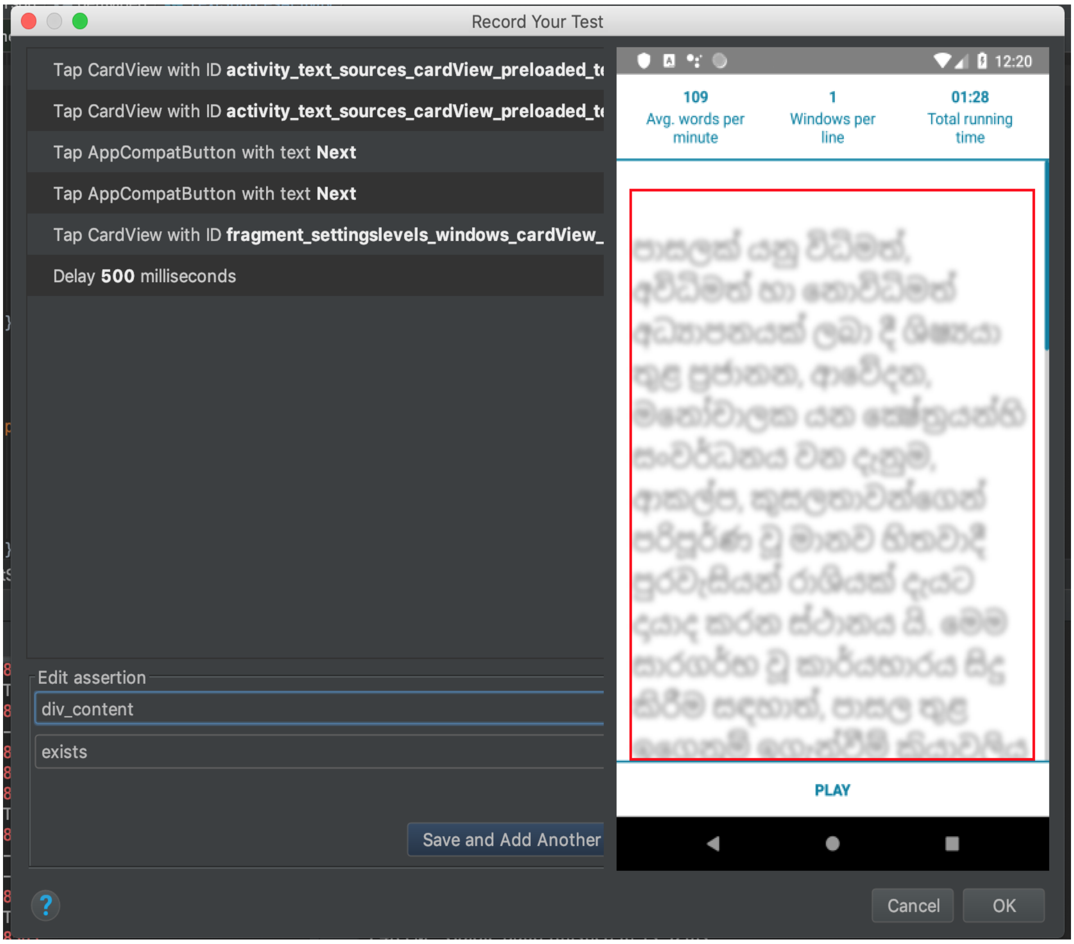
Select inspection to see problems:

Extraneous text in resource files
Layout Inflation without a Parent
Missing commit() on SharedPreference editor
Missing Permissions
Obsolete Gradle Dependency
ScrollView size validation
Target SDK attribute is not targeting latest version
Using dp instead of sp for text sizes

Figure 4.10. Inspection result window.

2. Espresso

Use Espresso test recorder reliable to Android UI tests. As an open source tool and very easy to extend with their working environment [11].



Tap CardView with ID **activity_text_sources_cardView_preloaded_t**

Tap CardView with ID **activity_text_sources_cardView_preloaded_t**

Tap AppCompatButton with text **Next**

Tap AppCompatButton with text **Next**

Tap CardView with ID **fragment_settingslevels_windows_cardView_**

Delay **500** milliseconds

Edit assertion

div_content

exists

Figure 4.11. Example of test record result

5. Conclusion

5.1. Summary of the Results of the Project

This project reached following conditions. Successful implementation of Practice Board, Successful implementation of Adjustable Predefined Configurations, Successful integration of OCR, Successful Integration of Google Drive sdk, Successful integration of Clipboard Manager. Reader view is an overlay that use to practice reading speed, there are a variety of input types to deal with Reader View. Able to input text from any language for practices. User input screen has multiple options. Preloaded text option should be opened with preloaded templates, External text share option only link with text and choose reader (SpeedRead) app to share, text scanning from any printed sources and Open text files from Google-drive.

Create custom presets base two selection methods like window method and word method. Both methods have predefined configurations and adjustable custom presets. Configure custom presets from using speed, font size, font family, overlapping characters, segmentation of window etc. Implement the seamless moving read bubble for window method is going to be text overlapping characters at the end of each fixation, according to the research. Furthermore, word of text move read bubble being preferring for word method.

5.2. Identify Deficiencies (shortcoming / limitation) in Final Product

There are few areas to be improved with some added customizations to the given features. Uploading files via Google Drive is currently supporting only for text (.txt) files. It can be improved to accept more file types like .docx, .ppt, .pdf, etc. Furthermore, the OCR is working well with printed text materials. Its shows poor performance when detecting the written texts. App has been integrated with the Google Vision OCR sdk which is improving rapidly. According to Google we can expect more improvements in the coming versions.

5.3. Highlight of Improvement of Future Work

Moving on this project can be improved in many ways. When considering the user's perspective, there should be a way to attract new users to the application and keep the existing user's attraction. Furthermore, the most important thing among all would be keep their interest in reading. Integrating gamification to the application will be a good solution. By doing that we can introduce a few new exciting features such as sharing the reading materials among users, graphical reports of the reading improvements, daily reading challenges as push notifications, challenge a friend, best readers weekly leaderboard, posting achievements to the social media, etc.

Integrating VR technology is another good strategy of improving the way of presenting the application to users. This will allow the users to involve more with the reading by minimizing distractions. Supporting multiple languages is another improvement planned to be added in a future version of the application. This will allow the application to be launched in different regions.

6. References

[1] Bsc.harvard.edu. (2019). “Reading Courses”. [online]
Available at: https://bsc.harvard.edu/reading-course [Accessed 27 May 2019].

[2] Google Books. (2018). “Print Anthology”. [online] Available at: https://books.google.lk/books/about/Print_Anthology.html?id=ZcyuswEACAAJ&redir_esc=y&hl=en [Accessed 2 Jun. 2018].

[3] Pearson.com. (2018). “Reading Comprehension Strategies | Strategic Reader”. [online] Available at: https://www.pearson.com/us/higher-education/products-services-teaching/learning-engagement-tools/strategic-reader.html [Accessed 2 Jun. 2018].

[4] Www-4.unipv.it. (2018). “Eye-movement analysis of reading from computer displays eReaders and printed books” [online] Available at: http://www-4.unipv.it/ereading/wp-content/uploads/2012-Eye-movement-analysis-of-reading-from-computer-displays-eReaders-and-printed-books.pdf [Accessed 1 Jun. 2018].

[5] Link.springer.com. (2018). [online] Available at: https://link.springer.com/content/pdf/10.1007/s10936-016-9418-2.pdf [Accessed 1 Jun. 2018].

[6] Google Drive. (2018). “Reading Comprehension Strategies | Strategic Reader Module .mp4”. [online] Available at: https://drive.google.com/open?id=19s_6xw5G4Zi2yxqlPnjE2JP_NWbVLEGN [Accessed 2 Jun. 2018].

[7] Lenhart, A., Madden, M., Smith, A. and Macgill, A. (2007). “Teens and Social Media”. [online] Pew Research Center: Internet, Science & Tech. Available at: http://www.pewinternet.org/2007/12/19/teens-and-social-media/ [Accessed 23 Jul. 2018].

[8] Saddiki, H., Cavalli-Sforza, V. and Bouzoubaa, K. (2017). “Enhancing Visualization in Readability Reports for Arabic Texts. Procedia Computer Science”, 117, pp.241-247.

[9] Google Developers. (2019). “Text Recognition API Overview | Mobile Vision | Google Developers”. [online] Available at: https://developers.google.com/vision/android/text-overview [Accessed 29 May 2019].

[10] Android Developers. (2019). “Improve your code with lint checks | Android Developers”. [online] Available at: https://developer.android.com/studio/write/lint [Accessed 29 May 2019].

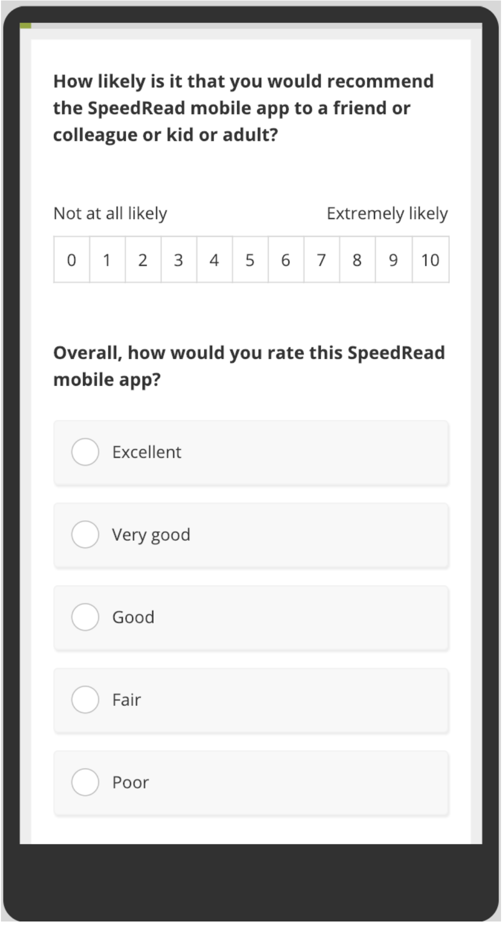
[11] Android Developers. (2019). “Espresso | Android Developers”. [online] Available at: https://developer.android.com/training/testing/espresso [Accessed 29 May 2019].

7. Appendices

7.1. Survey Evaluation

Add new users through the Crashlytics app and Users should now receive an email inviting them to test the “Speed Read” mobile app. So, users made it easily download the app.

According to the requirements gathering techniques an online questionnaire survey has been used. The questionnaire link was distributed among undergraduates and school students. Furthermore, paper prototype was distributed among students and expert and observing their ideas. Having survey evaluation pay attention to user feedback after the participant was involved to the speed-read mobile app. Fixed Question is present in here.



How likely is it that you would recommend the SpeedRead mobile app to a friend or colleague or kid or adult?

Not at all likely — Extremely likely

0	1	2	3	4	5	6	7	8	9	10

Overall, how would you rate this SpeedRead mobile app?

○ Excellent

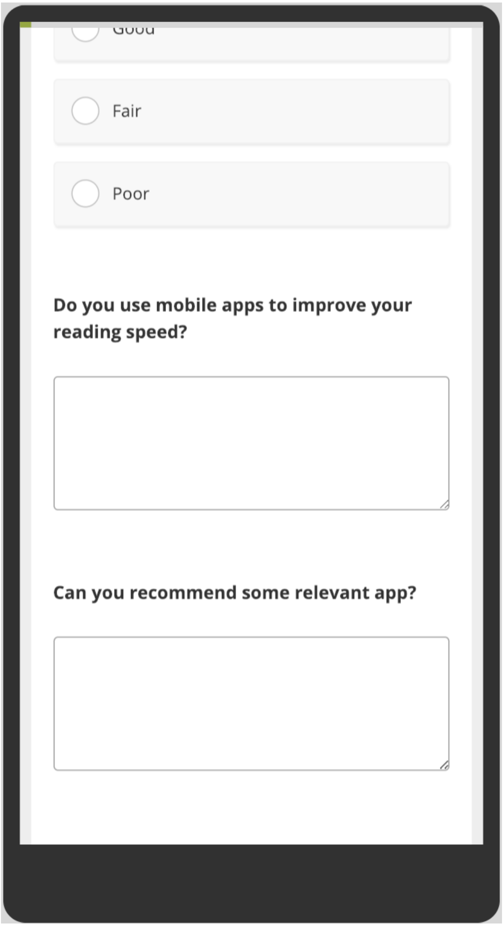
○ Very good

○ Good

○ Fair

○ Poor

Questionnaire 1.



○ Good

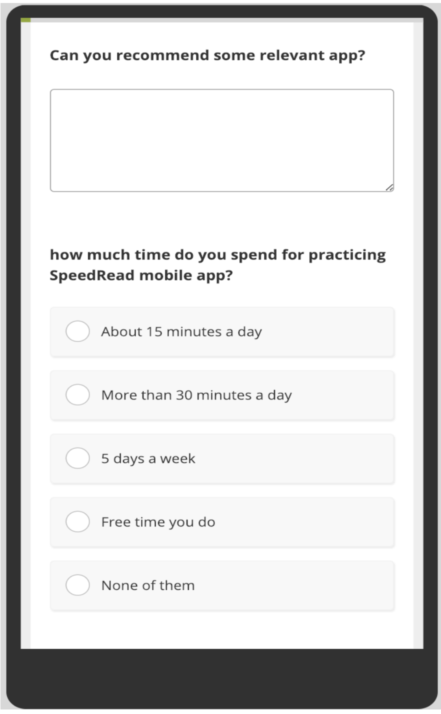
○ Fair

○ Poor

Do you use mobile apps to improve your reading speed?

Can you recommend some relevant app?

Questionnaire 2.



Can you recommend some relevant app?

how much time do you spend for practicing SpeedRead mobile app?

○ About 15 minutes a day

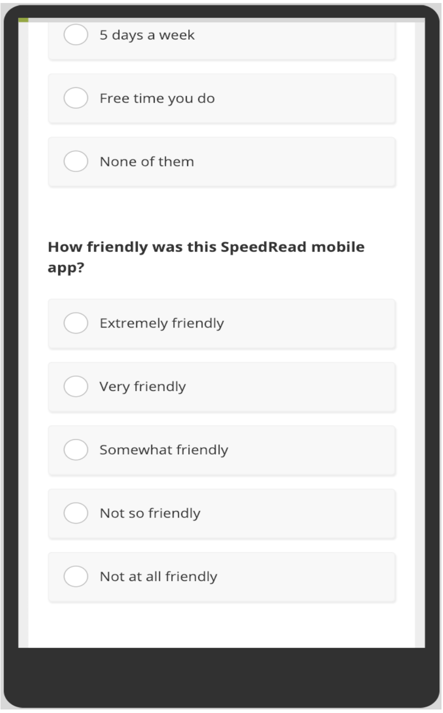
○ More than 30 minutes a day

○ 5 days a week

○ Free time you do

○ None of them

Questionnaire 3.



○ 5 days a week

○ Free time you do

○ None of them

How friendly was this SpeedRead mobile app?

○ Extremely friendly

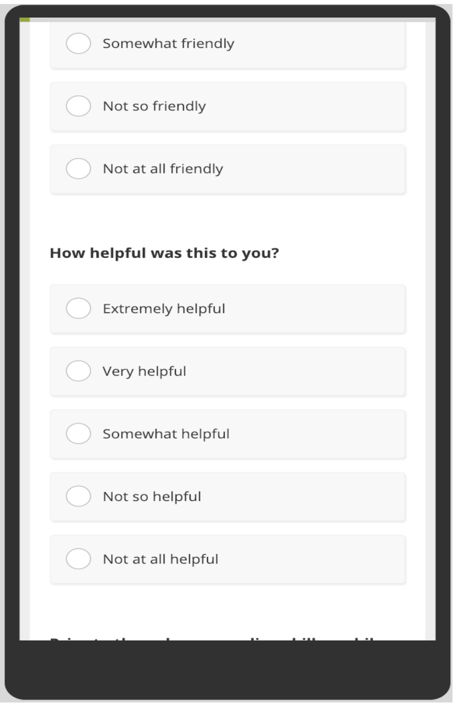
○ Very friendly

○ Somewhat friendly

○ Not so friendly

○ Not at all friendly

Questionnaire 4



○ Somewhat friendly

○ Not so friendly

○ Not at all friendly

How helpful was this to you?

○ Extremely helpful

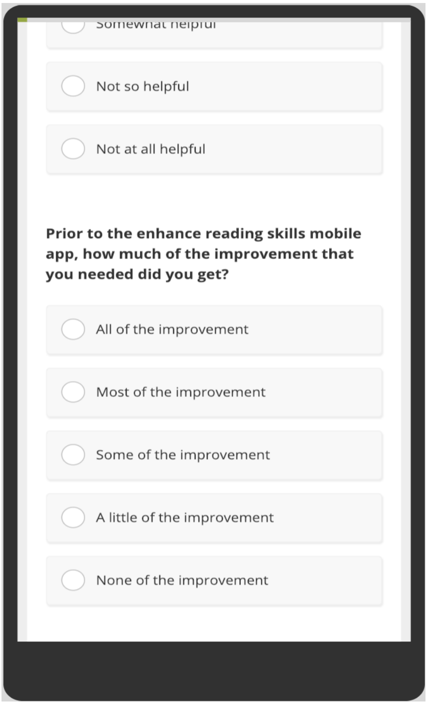
○ Very helpful

○ Somewhat helpful

○ Not so helpful

○ Not at all helpful

Questionnaire 5.



○ Somewhat helpful

○ Not so helpful

○ Not at all helpful

Prior to the enhance reading skills mobile app, how much of the improvement that you needed did you get?

○ All of the improvement

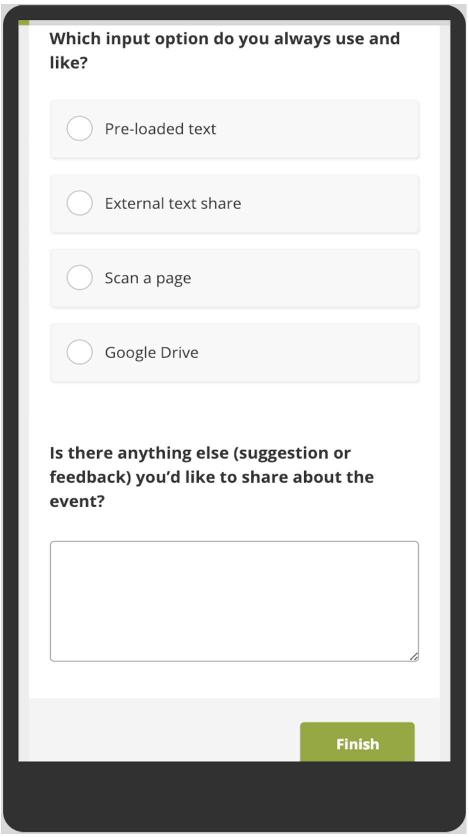
○ Most of the improvement

○ Some of the improvement

○ A little of the improvement

○ None of the improvement

Questionnaire 6.



Which input option do you always use and like?

○ Pre-loaded text

○ External text share

○ Scan a page

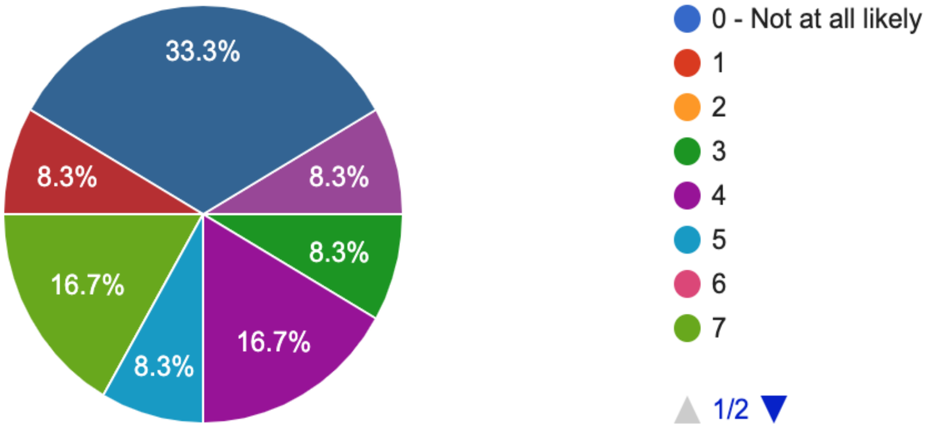
○ Google Drive

Is there anything else (suggestion or feedback) you'd like to share about the event?

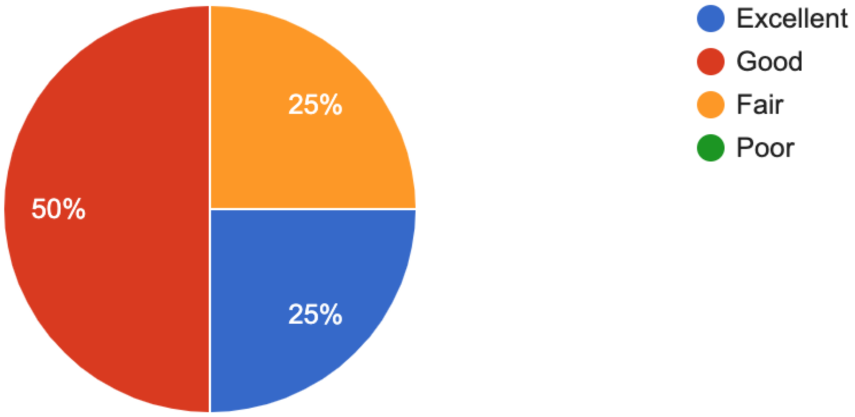
Questionnaire 7.

7.2. Results of Survey

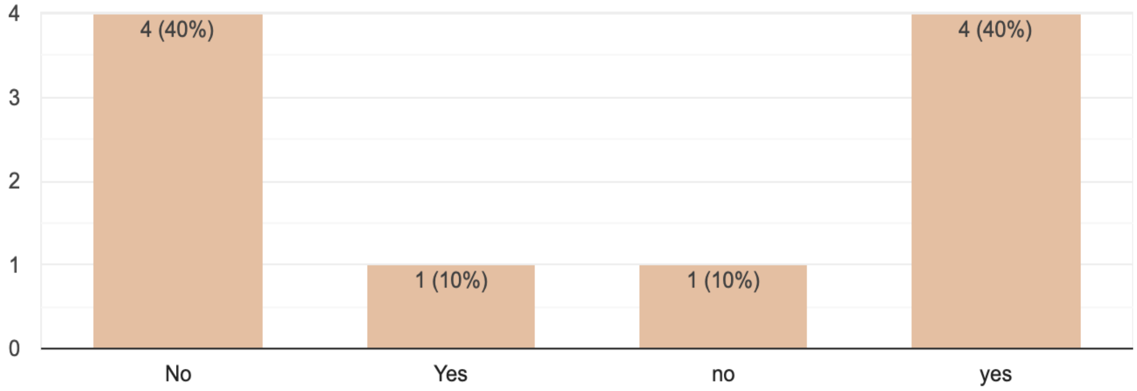
1. How likely is it that you would recommend the SpeedRead mobile app to a friend or colleague or kid or adult?



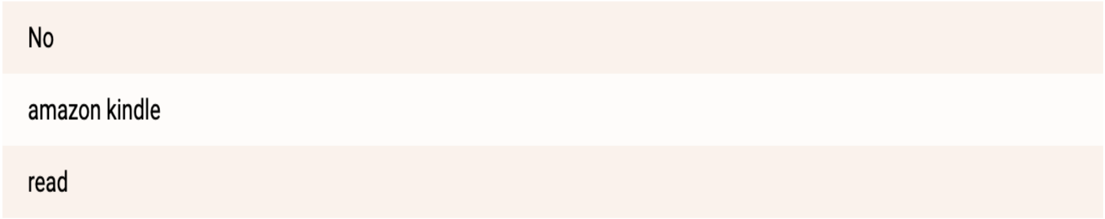
2. Overall, how would you rate this SpeedRead mobile app?



3. Do you use mobile apps to improve your reading speed?



4. Can you recommend some relevant app?

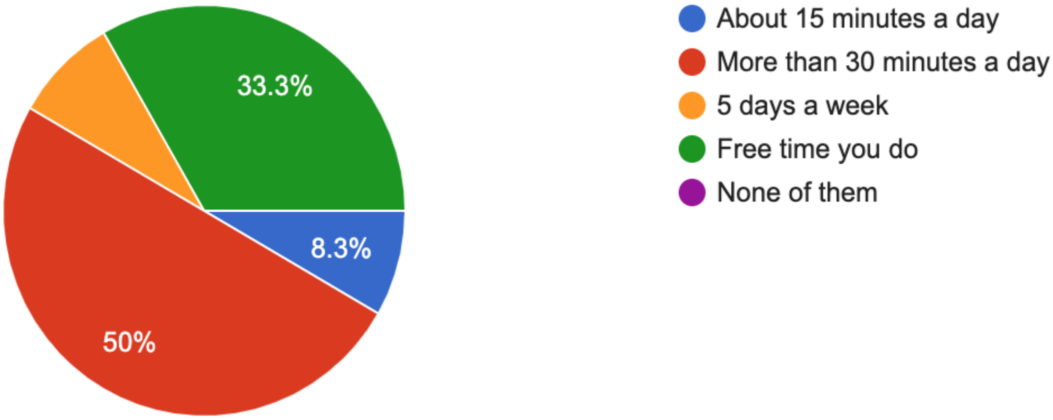


No

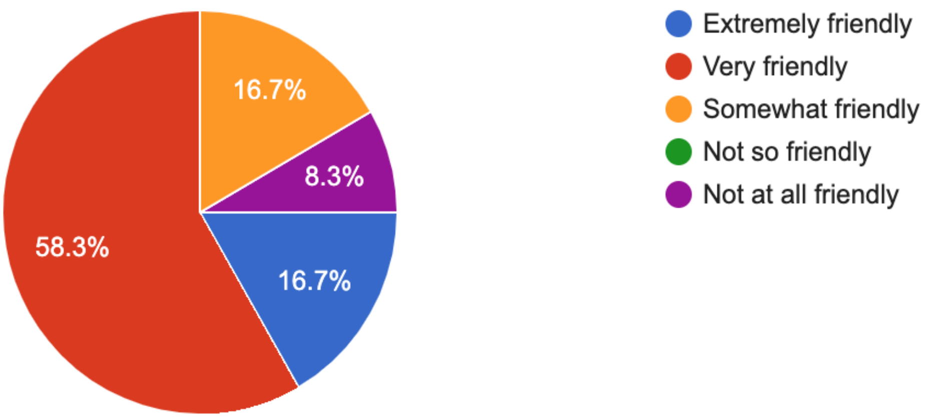
amazon kindle

read

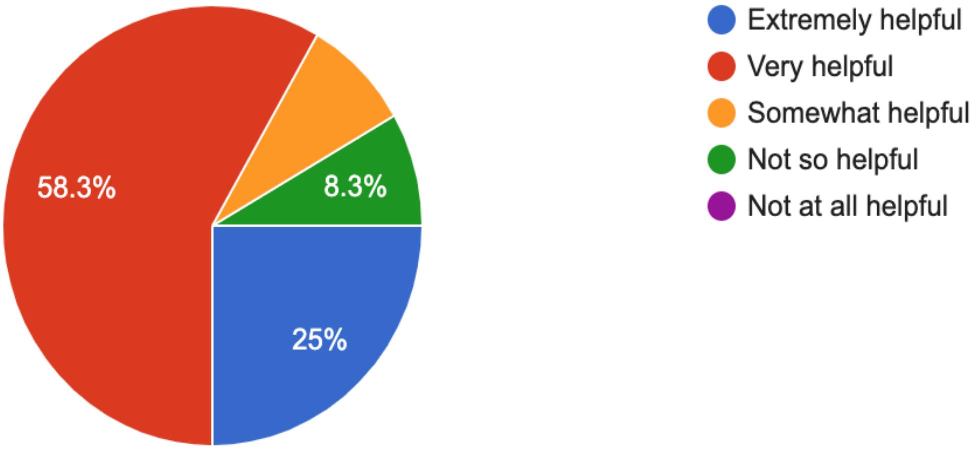
5. How much time do you spend practicing SpeedRead mobile app?



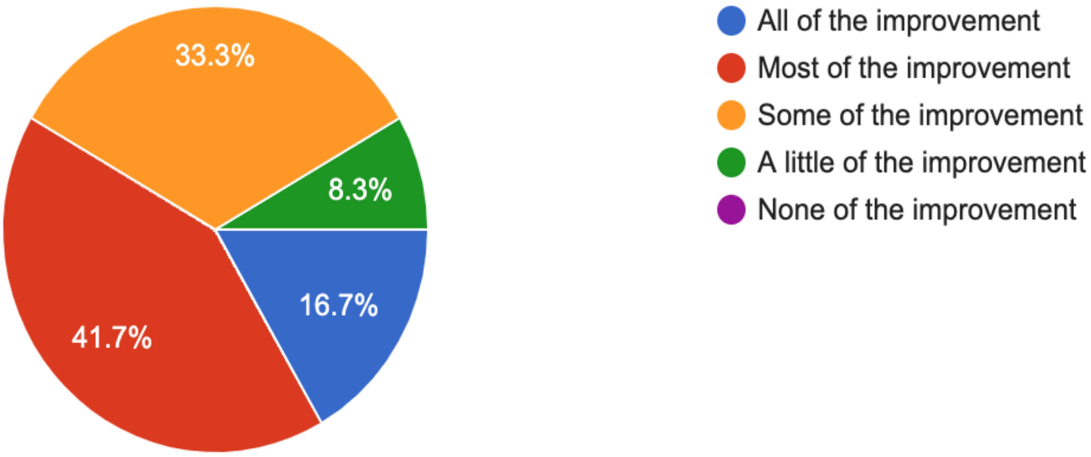
6. How friendly was this SpeedRead mobile app?



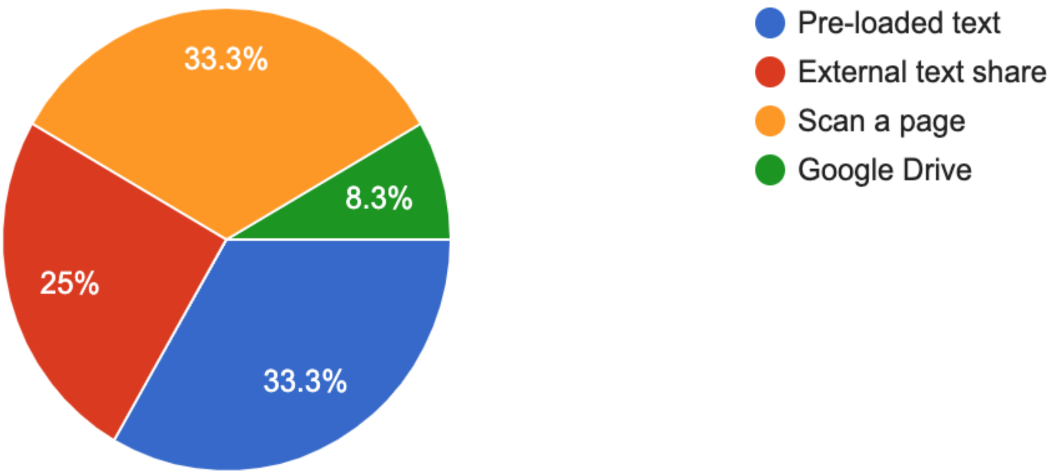
7. How helpful was this to you?



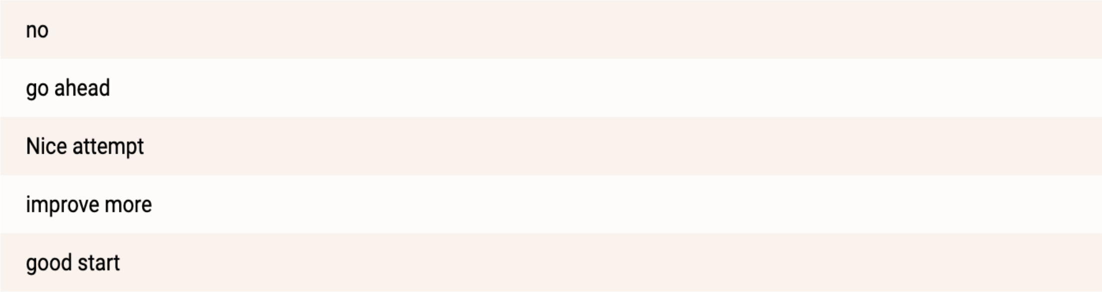
8. Prior to enhance reading skills mobile app, how much of the improvement that you needed did you get?



9. Which input option do you always use and like?



10. Is there anything else (suggestion or feedback) you'd like to share about the event?



no

go ahead

Nice attempt

improve more

good start

7.3. Test Scenarios

Test Case No	1.0
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	View input option screen
Test Steps	Open the application
Test Data	none
Expected Output	Input option screen should be visible, including the following options. 1). Preloaded Text 2). Share from External 3). Scan a Page 4). Google Drive
Actual Output	Visible as expected
Test Result	Pass

Table 7.1. View input option screen test case

Test Case No	2.0
Test Device	LG - G5
Operating System	6.0 Marshmallow
Test Case Description	Selected Text View
Test Steps	Display the selected text
Test Data	Input text from preloaded text or share from external text or scan page image text or google drive text.
Expected Output	● Allow to display text note. ● Move to another page, can be click “next” button.
Actual Output	Visible as expected
Test Result	Pass

Table 7.2. Selected text view test case

Test Case No	3.0
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Setting Screen
Test Steps	Select setting tabs
Test Data	none
Expected Output	Allow to select window method tab or word method tab
Actual Output	Visible as expected
Test Result	Pass

Table 7.3. Setting screen test case

Test Case No	3.1
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Setting Screen
Test Steps	Update default presets
Test Data	Default presets are “easy”, “medium” or “hard”
Expected Output	User can select preset option one at a time.
Actual Output	Visible as expected
Test Result	Pass

Table 7.4. Setting screen test case

Test Case No	3.2
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Setting Screen
Test Steps	Add custom preset Name
Test Data	none
Expected Output	● Display popup message for enter custom preset name ● Select “OK” button to save new preset name. ● Select “Cancel” button for ignore new preset name.
Actual Output	Visible as expected
Test Result	Pass

Table 7.5. Setting screen test case

Test Case No	3.3
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Window type Setting Screen
Test Steps	Update window custom preset setting
Test Data	Import defaults preset or edit window setting preferences
Expected Output	● Select default preset like import “easy”, “medium” or “hard” ● Configure custom presets such as windows per line, overlapping characters, speed. ● Change the font style and size.
Actual Output	Visible as expected
Test Result	Pass

Table 7.6. Window type setting screen test case

Test Case No	3.4
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Word type Setting Screen
Test Steps	Update word custom preset setting
Test Data	Import defaults preset or edit words setting preferences
Expected Output	● Select default preset like import “easy”, “medium” or “hard” ● Configure custom presets such as words per minute, words per window. ● Change the font style and size.
Actual Output	Visible as expected
Test Result	Pass

Table 7.7. Word type setting screen test case

Test Case No	3.5
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Setting Screen
Test Steps	Delete custom preset
Test Data	none
Expected Output	● Display popup message for confirmation of delete item ● Select “OK” button for delete custom preset name. ● Select “Cancel” button for ignore deletion of preset name.
Actual Output	Visible as expected
Test Result	Pass

Table 7.8. Setting screen test case

Test Case No	4.0
Test Device	LG - G5, Samsung Galaxy S9
Operating System	6.0 Marshmallow
Test Case Description	Reader Board
Test Steps	Practice with Reader Board
Test Data	Select default preset or custom preset.
Expected Output	● Display reading statistics (Avg. words per minute / Windows per line / Total running time). ● Adjusting the reading bubble to properly wrap the target text area to get clear focus on the text. ● Auto scroll down the WebView page while moving text across the window.
Actual Output	Visible as expected
Test Result	Pass

Table 7.9. Reader board test case

7.4. Mind Map

Reading Comprehension Levels	Age Group	Skill	Selection Method	Predefined Configurations
Primary Level Readers	Children of 3 - 10 years belong	Enhance reading skills beginning with ONE language	Word Method	“Easy” - Words per minute = 100 - Avg. windows per line = 0.98 - Total running time = 01:36 “Medium” - Words per minute = 200 - Avg. windows per line = 0.72 - Total running time = 00:48
Secondary Level Readers	Students of 11 - 18 years belong	Enhance reading skills beginning with minimum TWO languages	Word Method & Window Method	“Hard” (Word Method) - Words per minute = 250 - Avg. windows per line = 0.62 - Total running time = 00:38 “Easy” (Window Method) - Avg. words per minute = 88 - Windows per line = 1 - Total running time = 01:48
Tertiary Level Readers	Include college students and adults	Enhance reading skills with TWO or MORE languages	Window Method	“Medium” (Window Method) - Avg. words per minute = 240 - Windows per line = 2 - Total running

<table>
<tr><td></td><td></td><td></td><td></td><td>time = 01:29

“Hard” (Window Method)
- Avg. words per minute = 156
- Windows per line = 3
- Total running time = 01:01</td></tr>
<tr><td colspan="5"><b>Summary</b>
1. Person
2. Age
3. Current level of reading (from a scale of 0 to 10 how frequent you are reading? 0 = not reading at all, 10 = reading daily)
4. Language
5. Min Length (minimum length of the text)
6. Max Length (maximum length of the text)
7. Frequency (How frequent you accessed the app - in hours per day)
8. Total hours practiced
9. Speed chart (Avg. windows per second used in each day during the month)
10. Frequently used input method (preloaded text, external text share, scan page, open text file from G-Drive)</td></tr>
</table>

Table 7.10. Mind map for Speed Read App