

Vinnitsia National Technical University

---

(full name of a higher educational establishment)

Faculty of Computer Systems and Automation

---

(full name of institute, name of faculty (department))

Department of Control Computer Systems

---

(full name of the department (subject, cycle commission))

## **Explanatory note**

to the master's qualification work

on the topic «Development of an intelligent system to support the switching of modes of reading the electronic books with audio data»

Fulfilled: 2nd year student of the group

3ACIT-18m

specialty

151 Automation and computer-integrated

---

technologies

---

(cipher and name of the field of training, specialty)

educational-professional program

Information systems and Internet of things

---

(name of the educational-professional program)

**Andrii M. Henri**

---

(surname and initials)

Supervisor

**TetyanaV. Gryshchuk**

---

(surname and initials)

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем та автоматики

Кафедра комп'ютерних систем управління

Освітньо-кваліфікаційний рівень магістр

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології  
(шифр і назва)

Освітньо-професійна програма Інформаційні системи і Інтернет речей

ЗАТВЕРДЖУЮ

Завідувач кафедри АІВТ,  
д.т.н., проф. Р.Н. Кветний

\_\_\_\_\_  
“”\_\_ 2019 року

**ЗАВДАННЯ  
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Генрі Андрію Михайловичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Розробка інтелектуальної системи підтримки перемикання режимів читання електронних книг з аудіо даними»

Керівник магістерської кваліфікаційної роботи Гришук Тетяна Вікторівна, к.т.н., доц. каф. КСУ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “” року №\_\_

2. Строк подання студентом магістерської кваліфікаційної роботи \_\_\_\_\_ року.

3. Вихідні дані до магістерської кваліфікаційної роботи: підтримка ОС: Windows, Linux та MAC; використання засобів для розпізнавання мови – так; препроцесинг текстового файлу – так; обробка природної мови – так; отримання результатів у вигляді структурованого файлу з часовими штампами для слів – так; розробка мобільного додатку для iOS і Android – так; мова графічного інтерфейсу – англійська.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз архітектурних рішень систем-аналогів, аналіз засобів розпізнавання мови, розробка архітектури системи, розробка програмного забезпечення системи, розробка методу синхронізації аудіо і тексту, тестування програмного забезпечення, висновок.

5. Перелік графічних матеріалів (з точним зазначенням обов'язкових креслень): набір схем: концепція зпівставлення аудіо і тексту, алгоритм зпівставлення (обов'язковий), UML-діаграма варіантів використання (обов'язковий), UML-діаграма діяльності системи (обов'язковий), UML-діаграма класів (обов'язковий), екранна форма бібліотеки мобільного додатку, екранна форма режиму читання мобільного додатку.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-5	Грищук Т.В., доц. каф. КСУ		

7. Дата видачі завдання 2019 року

Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Варіантний аналіз архітектурних рішень систем-аналогів		
2	Аналіз засобів розпізнавання мови		
3	Розробка архітектури системи		
4	Розробка програмного забезпечення системи		
5	Розробка UML-діаграм системи		
6	Розробка методу синхронізації аудіо і тексту		
7	Тестування програмного забезпечення		
8	Оформлення пояснювальної записки, графічного матеріалу презентації		
9	Попередній захист		
10	Захист МКР		

Студент \_\_\_\_\_ Генрі А.М .  
( підпис ) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_ ГрищукТ.В.  
( підпис ) (прізвище та ініціали)

Vinnitsia National Technical University  
(full name of a higher educational establishment)  
Faculty of Computer Systems and Automation  
Department of Control Computer Systems  
Education level master's degree  
Specialty 151 Automation and computer-integrated technologies  
(cipher and name)  
Educational-professional program Information systems and Internet of things

APPROVED

Head of the AIIS department

Dr. of Sci., professor Roman N. Kvetny

\_\_\_\_\_  
“” 2019 year

THE TASK  
ON THE MASTER'S QUALIFICATION WORK STUDENT

Andrii M. Henri .

(fullname)

1. Theme of master's qualification work «Development of an intelligent system to support the switching of modes of reading the electronic books with audio data»

Supervisor of master's qualification work Tetyana V. Gryshchuk,

(full name, degree, academic rank)

approved by the order of the higher educational institution from № \_\_\_\_\_

2. The term of submission by the student of master's qualification work .

3. Output data for master's qualification work: OS support: Windows, Linux and MAC; the use of language recognition tools - yes; text file preprocessing - yes; natural language processing - yes; getting results as a structured time stamp file for words - yes; development of mobile application for iOS and Android - yes; GUI language is English.

4. Contents of the settlement and explanatory note (list of issues to be developed): introduction, analysis of architectural solutions of analog systems, analysis of speech recognition tools, development of system architecture, development of system software, development of method of synchronization of audio and text, testing of software, conclusion.

5. List of graphic material (with exact indication of mandatory drawings): set of schemes: the concept of matching audio and text, aligning algorithm (required), UML-diagram of use cases (required), UML-diagram of activity (required), UML-diagram of classes (required), screen view of the mobile application library, screen view of the mobile application reading mode

6. Consultants of sections of master's qualification work

Section	Full name and position of consultant	Signature, date	
		The task was issued	The task was accepted
1-5	Ph. D, associate professor Tetyana V. Gryshchuk		

7. Date of issue of the assignment

Calendar schedule

№	The name of the stages of work	The duration of the stages of work	Note
1	Variant analysis of architectural solutions of analog systems		
2	Analysis of language recognition tools		
3	Development of system architecture		
4	System software development		
5	Development of UML diagrams of the system		
6	Development of audio and text synchronization method		
7	Software testing		
8	Explanatory note, graphic materials and presentation design		
9	Previous protection		
10	Protection of MQW		

Student \_\_\_\_\_ Andrii M. Henri  
(signature)(Full name)

Supervisor of master's qualification work \_\_\_\_\_ Tetyana V. Gryshchuk  
(signature) (Full name)

## ABSTRACT

This qualification work is devoted to the development of an intelligent system to support the switching of modes of reading the electronic books with audio data. Effective solutions have been proposed for creating an algorithm for aligning text and audio files of a book, using speech recognition. The work of the developed system was successfully tested in the presence of books of different lengths and various speakers that recorded the audio version.

## CONTENTS

INTRODUCTION .....	9
1 TECHNICAL JUSTIFICATION OF THE PURPOSE OF DEVELOPMENT.....	12
1.1 Overview of e-book formats .....	12
1.2 Overview of audiobook formats .....	16
1.3 Analysis of existing analog products .....	18
1.4 Formulation of the main experimental design problem.....	22
2 DEVELOPMENT OF THE METHOD OF SYNCHRONIZATION OF TEXT AND AUDIO MODES OF READING THE ELECTRONIC BOOK.....	24
2.1 Speech-to-text aligning process .....	24
2.2 Analysis of speech recognition systems .....	24
2.3 Method for synchronizing text and audio modes of e-book reading.....	29
3 DEVELOPMENT OF THE SOFTWARE.....	31
3.1 Development of system architecture.....	31
3.1.1 UML diagrams .....	31
3.1.1.1 UsecaseUML-diagram .....	31
3.1.1.2 ActivityUML-diagram .....	32
3.1.1.3 ClassUML-diagram.....	32
3.2 Rationale for the choice of software .....	33
3.3 Software development.....	34
3.3.1 Development of an aligning speech-to-text software .....	34
3.3.2 Development of a mobile application.....	38
3.4 Description of mobile application.....	46
3.5 Supporting documentation .....	49

4 QUALITY ASSESSMENT OF THE DEVELOPED SYSTEM.....	51
4.1 Choosing a quality criterion.....	51
4.2 Description of test data for quality assessment.....	52
4.3 Quality assessment results .....	53
5 PLAN FOR IMPLEMENTATION AND DISSEMINATION OF THE DEVELOPED SYSTEM.....	54
5.1 Market and consumer analysis.....	54
5.2 Entering the Android and iOS markets.....	55
5.2.1 Google Play.....	57
5.2.2 App Store .....	57
5.3 Application promotion methods.....	57
CONCLUSIONS.....	65
REFERENCES.....	66
ATTACHMENTS .....	72
ATTACHMENT A .....	73
ATTACHMENT B.....	76
ATTACHMENTC.....	84
ATTACHMENTD .....	87
ATTACHMENTE.....	89



## INTRODUCTION

**Topicality.** There are situations in life when there is almost no free time to read and audiobooks come to the rescue. When it is not possible to read in quiet conditions, and in public transport will always save the audiobook. Not everyone likes to listen to books, but it's still a way to get information. Also e-books are used by people with hearing impairments. It would be convenient to use an application that allows you to switch between the electronic text version of the book and audio at any time while continuing to read from the same place. There are several similar analogues, but none of them supports Ukrainian or Russian.

There are options for converting an audio book to text immediately. The pitfalls here are incorrect punctuation (or almost none at all) and not always correctly recognized text with incorrect punctuation, endings, even even recognized words. There are also attempts to convert a text book into an audiobook. The main problems - the punctuation and intonation of the speaker, and the voice of the speaker, usually very robotic. An option is required that will allow you to use both correctly formatted original text of the book and a well-recorded audiobook, usually by a professional speaker.

Based on the above information, the challenge is to develop an intelligent system for supporting the switching of ebooks reading audiobooks.

**Relationship with working with scientific programs, plans, topics.** The work was performed at the KSU department within the theme 46K2 “Decision-making methods based on computational intelligence”.

**The purpose and objectives of the study.** The purpose of the work is to increase the reliability of the process of switching modes of ebooks with audio support.

To achieve this goal, the following tasks must be accomplished:

- to analyze the current state of the field of application of speech recognition systems;
- analyze the language recognition systems;

- consider the problems of the language recognition process;
- to analyze the existing analog products;
- to justify the choice of the method of solving the problem;
- develop a method for synchronizing audio and text;
- to test the system;
- analyze the results and draw conclusions.

**The object of study** is the process of synchronizing audio and text data.

**The subject of study** - methods of finding similarity of text documents.

**Research methods.** In the process of research were applied: theory of numbers and numerical methods; theory of algorithms; methods of automatic speech recognition.

**Scientific novelty of the obtained results.**

Improved audio and text-based similarity search method based on the use of automatic language recognition to increase the accuracy of switching between normal and audio e-book modes.

**The practical significance of the obtained results** is that, based on the theoretical provisions obtained, algorithmic and software were developed, in particular:

- a module for mapping audio file and text based on the language recognition system;
- software for reproducing text with time stamps with an alignment to an audio file;
- the system architecture was developed;
- an application was created in which the above developments are used.

**The validity of the theoretical provisions** of the master's qualification work is confirmed by the rigorous formulation of problems, the correct application of mathematical methods in the proof of scientific provisions, the strict derivation of analytical relations, the comparison of the results obtained by the methods developed in the work with the known, and the convergence of the results of mathematical results during the implementation of the developed software and hardware.

**Personal contribution of the applicant.** All the results were obtained by the author himself.

**Aprobation of results of work.** None.

**Publications.** There are no publications on the subject of the study.

# 1 TECHNICAL JUSTIFICATION OF THE PURPOSE OF DEVELOPMENT

## 1.1 Overview of e-book formats

An e-book format is a type of file that is able to correctly recognize and play a device for reading electronic books, or an application for a tablet, computer, smartphone [2].

Portable Document Format (PDF) was officially introduced by Adobe Systems in 1993 [3]. This format was created for documents. The purpose was to invent "electronic paper", to make documents that would look the same on any device. For this purpose, both the pros and cons of the format follow.

Pros: Beautiful paper-like books that are equally displayed on any device. PDF properties are like a collection of photos, so this format will never "float" fonts, and the terms will forever be fixed in their place. File encryption from copying is possible. Newer files in this format support interactive elements such as live content, hyperlinks, embedded video and audio, and more. Still in this format, it is very convenient to save scanned books for viewing from your computer. You can add book metadata to a file.

Cons: Very large files, will slowly open on the reader; expensive encryption is not reliable (easy to crack [4]); files are not optimized for phones and small format readers. That is, if the book is facing A4, then on a small screen you will get a tiny copy of a standard sheet that you will have to zoom in and try to read. Some readers have a special feature to correct this inconvenience - reflow - that changes the layout by adjusting it to your screen automatically. However, in most cases the layout breaks down rather than adapts.

Conclusion: suitable for scanned books as well as books where static layout is important: there are many illustrations with captions or formulas or graphs, and for periodicals. Suitable for reading at least 9 inches from your computer or tablet.

Electronic Publication (EPUB) emerged in 1999 and immediately took the lead. In 2009, EPUB became one of the first book formats to have its own browser extension - it became even easier to read books [5].

EPUB understands bitmaps and vector images, newer versions understand formulas very well, allow embedding video, audio, 3D objects, hyperlinks, etc., as well as embedding fonts inside a file. This allows you to use beautiful copyright fonts and not be afraid of them will not work on another device. By the way, it is important: even if the font was not embedded and it is not available, the reader will still be able to read the book - it will simply be displayed in the standard font.

If a PDF is a photo album, then EPUB is the code. It looks like a site that has been archived. The use of HTML and CSS allow the layout engineer to get almost endless layout and design options.

Of course, EPUB also allows you to stitch metadata in the book in maximum detail. It can also be protected by DRM from Adobe but it is expensive and easy-cracking.

Pros: Files take up little space and are readable on almost any device (except some Kindle). Always the correct layout, because it is dynamic - the book itself adapts to your screen, whatever the size of it. Great possibilities for design, layout, interactive filling. Creation speed - Normal EPUB (without demanding design) is made in 1, maximum 2 weeks. The format continues to evolve and is updated frequently. One more important note: EPUB lets you change the book's characteristics simply from the reader. The reader can change the size of the text, font, field width, etc. (if they are not embedded in the file beforehand by the maker of the book). It can almost completely change the design of a book just on a smartphone, tweaking everything to your liking. This, in fact, is the meaning of a dynamic layout - maximum convenience.

Cons: Difficulties in layout of formulas and tables. Unusual appearance - EPUB is more like a site than a paper book. The impossibility of static layout.

Conclusion: Ideal format for books that do not require tables. Suitable for almost all devices (with rare exceptions). Fast, convenient. Not suitable for static layout.

The first version of this format - FB1 - was created in 2004 by a group of Russian developers. In 2005, FB2 was released, which fixed the bugs of the previous version, and in the same year released the first ebook in this format. Interestingly, unlike EPUB, which was specifically designed for e-book creation, FB2 was created as a means of temporarily storing book files for later conversion to other popular formats [6]. This translates into the main advantages of this format: it is easy to convert and weighs a little.

The FB2 and EPUB formats are somewhat similar in that they both consist of code and offer dynamic layout. FB2 is a combination of XML documents. It has metadata for the book, allows you to make styles, fonts and colors in addition to text and images, to "sew" into the book. But it is impossible to encrypt such a file, it does not accept DRM encoding.

Pros: Dynamic layout, the not so heavy files (they can be further archived, then FB2 books will be even lighter than EPUB), easily converted to almost anything.

Cons: The format is no longer being developed and announced in 2008 has not been released. Does not understand mathematical formulas, design options and text formatting are limited (compared to EPUB). Only Russian and some Chinese readers understand this format, and most Westerners have never heard of this format. Books in this format are only sold in CIS stores.

Conclusion: You can publish books that do not require spreadsheets and multimedia. Convenient for online libraries as well as old readers with limited storage: files take up little disk space, allowing you to download many books.

MOBI, AZW, KF8 were created specifically for reading and storing ebooks. MOBI is a specialized format created by Amazon specifically for the Kindle reader. It was designed using PalmDOC format, with some HTML tags added. Like the EPUB, MOBI is like an archive filled with XHTML documents [7]. A dynamic layout model is used. It is a bit difficult to make verse texts because of the restrictions

in the indentation. Also, it is very often incorrectly rendered tables, so they are recommended to make pictures. Images can be completed only small and simply in the middle of the text, with no flow or borders. This format hardly understands mathematical formulas.

AZW - like MOBI, only encrypted [8]. AZW file protection is based on Adobe DRM protection technology. The encrypted file is linked to the account from which the book was purchased and is only opened on devices registered as the device of that particular buyer (up to six devices per account). AZW4 looks a little different - it's a kind of Amazon PDF version. Amazon positions it as an "exact electronic copy of a paper page" with a static layout and interactive elements and functions, in particular - synchronization of reading on different devices of one reader. But offices are not in a hurry to change one "e-paper" version to another.

The most recent of the Amazon formats - KF8, designed for the Amazon tablet Kindle Fire [9]. This format is developed on the basis of an open source EPUB using databases of the same Palm and MOBI. The files are encrypted with Amazon DRM. The format understands HTML5 and CSS3, allows you to embed fonts in the book. KF8 can already be called a worthy rival of EPUB. The only thing that bothers you is the closeness of the KF8 code. If EPUBs can be refined and developed for their needs by anyone who understands the code, then the original KF8 resources are only available to Amazon employees.

Pros: The new Amazon format offers many features; the choice between the three Amazon formats - you can choose the option for both static, dynamic layout, and multimedia.

Cons: Books in these formats are mostly Amazon-only and read only from Kindle (closed ecosystem); few professionals make up these formats, because they are not popular, and they are quite easy to overtake the same EPUB, so there is no point in focusing on these formats.

The book application is a fairly new format. An eBook developed for iOS or Android is created not by publishers, but by developers (sometimes by publisher

order). It is completely interactive: everything moves in response to user actions, there are sounds, videos, built-in mini-games.

Pros: very interesting and beautiful books, loved by children. Real books of the future. You can enter a lot of information into a book or just interact with the text.

Cons: very expensive development. Distraction from text. Focusing only on owners of smartphones and tablets of a particular operating system (owners of gadgets of other systems, as well as PCs or readers will not be able to see the book). Due to the high cost of development - the high selling price, which is irrelevant for the CIS countries.

Book sites is an online book that is stored on the server and can only be accessed if you have an Internet connection. It is a bit like an add-on book, but is usually designed not as much as a game, but as a textbook or project. It is possible to add photos, videos, audios, a bunch of links, embed questionnaires and tests, interactive graphics, 3D models, pop-up texts and more.

Pros: A great learning tool. A lot of information, all interactive, interesting, is an opportunity just to check the assimilation of the material by the reader. The spectrum of possibilities - all possibilities of the Network. The text is not lost, but supplemented as much as possible.

Cons: High cost of developing a book. Access to the book only online (inability to read offline).

## 1.2 Overview of audiobook formats

An audiobook is a voiced literary work recorded on any sound medium. This format of books is dictated by the author, actor or speaker to familiarize themselves with the work in the absence of the ability to read a text version of a publication. An audiobook is a modern solution that allows almost completely eliminating the problem of accessibility for people with visual impairments, as well as reducing the



time spent on reading for people with a high degree of employment and simplifying the absorption of information for children who just have to learn to read [10].

The term “audiobook” came into use in the 1970s, when recordings began to be made on audio tapes. As an industry standard, the term was adopted by the Audio Publishers Association in 1994. To date, audio books occupy a significant niche in the publishing segment and are increasingly popular with readers [11].

In the process of developing the audiobook industry, a new layer of the population is being formed, using this format in its daily routine. It is represented by reading people who prefer an audiobook to a printed publication in order to save time and relieve eye strain during normal reading. Audio books capture the audience of students and working citizens with their convenience in combining reading with household chores, travel, and so on. At the moment, the demand for audio books is increasing annually.

Among other things, audio books are actively used in kindergartens and pre-educational institutions for the formation in children who are still not able to read interest in literature and for educational purposes.

Also, an audiobook has become one of the ways to learn foreign languages. Most of those who want to learn another language are not able to communicate with its native speakers. An audiobook is one of the simplest and most affordable solutions to constantly maintain foreign language by ear and to develop the skill of recognizing learned words in dialogs. Audio books are a synthesis of written and spoken text. The listener simultaneously watches the text on the screen and listens as the announcer reads this text. It has been firmly established that listening to audio books read by professional broadcasters, together with the parallel reading of their text, facilitates language learning. Many language schools use the audiobook format as one of the most effective ways to educate their students [13].

WAV is one of the first audio formats. Typically used for storing uncompressed audio recordings (PCM), identical in sound quality to recordings on compact discs (audio-CDs). On average, one minute of wav sound takes about 10

megabytes. Often an audio CD is copied into it, and then converted to mp3 using an audio converter [15].

MP3 (MPEG Layer-3) is the most widely used sound format in the world. MP3, like many other formats with loss of quality, cuts down the sound that is not heard by the human ear, thereby reducing the file size [16]. At the moment, mp3 is not the best format for the ratio of file size to sound quality, but due to its popularity and support by most devices, many people store their recordings in it.

WMA (Windows Media Audio) is a format owned by Microsoft. Initially, this format was introduced as a replacement for MP3, which, according to Microsoft, has higher compression characteristics. But this fact has been called into question by some independent tests. The WMA format also supports data protection via DRM [17].

OGG is an open format that supports audio encoding with various codecs. The most commonly used ogg codec is Vorbis [18]. The quality of compression format is comparable to MP3, but less common in terms of support in audio players and players.

AAC is a patented audio format that has great capabilities (number of channels, sampling rates) compared to mp3 and gives a slightly better sound, with the same file size. At the moment, aac is one of the highest quality lossy sound coding algorithms. The format is supported by most devices. A file of this format can have the extensions aac, mp4, m4a, m4b, m4p, m4r.

FLAC is a popular lossless compression format [19]. It does not make changes to the audio stream and the sound encoded with it is identical to the original. Often used to listen to sound on high-level sound systems. It has limited support for devices and players, so usually in order to listen to flac in the player, it is pre-converted.

### 1.3 Analysis of existing analog products

Most people are often limited in time, so reading a book can take months. It would be convenient to continue reading in cases that do not require full concentration. And the solution is audiobooks. Yes, it's usually a lot slower to listen than read, but still better than nothing (and in general there is an option to accelerate audio playback). And such a solution is worth using. It would be convenient at any time to switch from an ebook to an audio version without manually searching for position, where you have stopped. This work also offers the following features - support for switching modes of reading ebooks with audio data. Due to the fact that the idea is quite original, it was difficult to find analogues.

Voice Dream Reader allows you to convert text files into languages. The program uses high-quality language synthesizers to convert text to language. User-friendly interface, supporting a large number of e-book formats, Option ist customize the appearance of text on the screen and read text synthesizer language - all this will be useful and convenient for any user. Connect to Dropbox, G-Drive, Evernote, Pocket, Instapaper, Gutenberg and Bookshare [21]. There are 78 high quality language synthesizers available in 20 languages from Acapela [22] and NeoSpeech [23]. The interface and functionality are presented in figure 1.1.

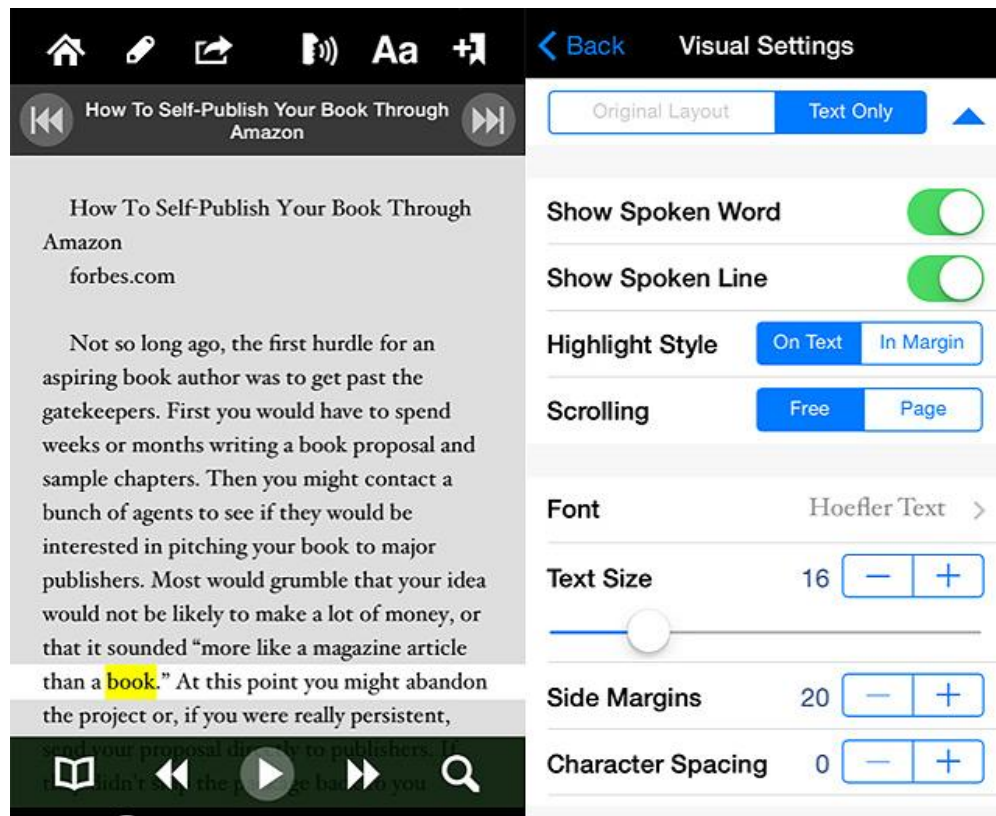


Figure 1.1 – Interface of VoiceDreamReader

The main features are the ability to continue your reading while driving, walking or running (switching from text to audio version of the book); the read word is highlighted to improve understanding and quality of memorization; auto-scroll and full-screen, abstract mode to help the reader focus. Advantages of this application is that even if there is no audio versions of this book, it will be created, and the ability to work with many languages. The main drawback is the "robotic" (synthesized) emotionless voice of the speaker, which is significantly inferior to that of a professional radio host, or of a voice-over actor.

The following is similar to my idea - a small HTML5 Audio Read-Along project [24]. This is a work created when the author was in college. He used the ESV API [25], which on request provided text and audio for one of the Bible chapters. The retained text was indexed by time, with some time to be found in the passage. The ESV team only indexed at chapter and verse level. The author of the application had to manually scroll through the audio file and find the timing of each word. The result of his project is presented in figure 1.2. The user can switch to the right moment of

the audio file by clicking on the word, and vice versa - on the word by clicking on the playback band at the right time.

The main drawback of the project is that the author had to manually record time indices. This is very time consuming with large amounts of data.

## HTML5 Audio Read-Along Demo

Read the [background and instructions](#) for this demo.

### Luke 2:1-20 (ESV)

0:23 / 2:08

Reading rate: 1x

Auto-focus/auto-scroll

<sup>1</sup>In those days a decree went out from Caesar Augustus that all the world should be registered. <sup>2</sup>This was the first registration when Quirinius was governor of Syria. <sup>3</sup>And all went to be registered, each to his own town. <sup>4</sup>And Joseph also went up from Galilee, from the town of Nazareth, to Judea, to the city of David, which is called Bethlehem, because he was of the house and lineage of David, <sup>5</sup>to be registered with Mary, his betrothed, who was with child. <sup>6</sup>And while they were there, the time came for her to give birth. <sup>7</sup>And she gave birth to her firstborn son and wrapped him in swaddling cloths and laid him in a manger, because there was no place for them in the inn.

<sup>8</sup>And in the same region there were shepherds out in the field, keeping watch over their flock by night. <sup>9</sup>And an angel of the Lord appeared to them, and the glory of the Lord shone around them, and they were filled with fear. <sup>10</sup>And the angel said to them, "Fear not, for behold, I bring you good news of great joy that will be for all the people. <sup>11</sup>For unto you is born this day in the city of David a Savior, who is Christ the Lord. <sup>12</sup>And this will be a sign for you: you will find a baby wrapped in swaddling cloths and lying in a manger." <sup>13</sup>And suddenly there was with the angel a multitude of the heavenly host praising God and saying,

<sup>14</sup>Glory to God in the highest,  
and on earth peace among those with whom he is pleased!

Figure 1.2 – Interface of HTML5 Audio Read-Along

Whispersync for Voice - is a technology of Amazon, which allows users of e-books or smartphones with the addition software switch between book and audio books, continuing to read/listen to the same position [26]. The interface is shown in figure 1.3 [27]. Synchronization is automatic and requires no action from or from the user, just buy text and audio versions of the book. An interesting feature is the complete synchronization of the position with other devices added to the account. The developers have not provided information on how this is implemented, but it is likely that Amazon Transcribe technology is being used.

It is an automatic language recognition service (ASR) that allows developers to easily add language-to-text capabilities to their applications. Using the Amazon Transcribe API, you can analyze audio files stored in Amazon S3 and retrieve text files with language decryption. You can also stream audio to Amazon Transcribe and receive real-time language transcripts [28]. The disadvantages are that sync happens only on pages, not on every word. There is also no highlighting of the current position in the text.

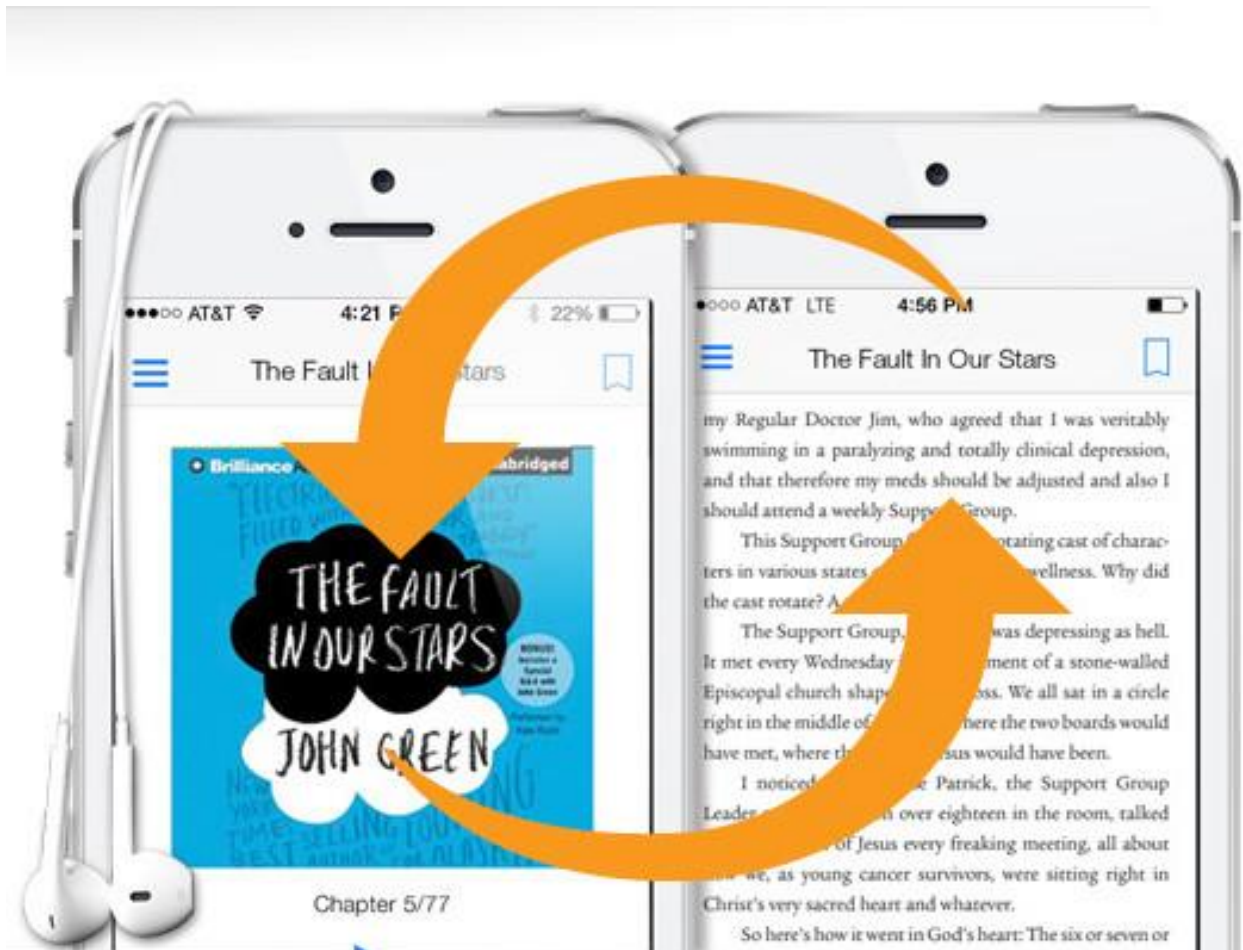


Figure 1.3 –Whispersync for Voice interface

#### 1.4 Formulation of the main experimental design problem

The object of the study, namely the intelligent system for supporting the switching of reading modes of ebooks with audio data, is based on a language recognition system.

The aim is to implement a ready language recognition system for processing incoming audio files and an application for reading end-user books.

Functions to be implemented by the system:

- automatic transcription of audio files of the book read by the speaker;
- automatic comparison of the original text of the book with the received text;
- the presentation of pre-processed data, in the form understood by the end user of the application.

Also, in order to improve the functionality of the system, you should consider using different languages in the future.

## 2 DEVELOPMENT OF THE METHOD OF SYNCHRONIZATION OF TEXT AND AUDIO MODES OF READING THE ELECTRONIC BOOK

### 2.1 Speech-to-text aligning process

Aligning audio input with its corresponding text is a well studied research problem in speech processing. This paragraph contains an overview of the current state of the art algorithms used for audio alignment.

We model the speech input as a Hidden Markov Model i.e. the states of speech follow a Markov model which is unknown (hidden), and the sequence of observed features form tokens that can be used to decide which state system is in. For force aligning the audio data with a transcription, we assume that the transcription resembles the actual content in the utterance. However, this assumption is not always true. In real life data, utterances may contain disfluency or the transcription itself might not be accurate.

### 2.2 Analysis of speech recognition systems

Nowadays, there is a large selection of language recognition projects, services and libraries. The most famous of these are: Google Speech-To-Text, CMU Sphinx Speech Recognition Toolkit, Yandex SpeechKit, Hidden Markov Model Toolkit (HTK), Microsoft Speech API. Some of them are open source.

Google Speech-To-Text is a Google product originally developed for voice search with speech recognition technology. The technology is integrated into gadgets and computers where you can enter information through voice. Since June 2011, Google has launched a search engine [29], which has been operating steadily to this day. Voice Search on PC is only supported by Chrome (a Google-developed browser). Voice control is also supported on Android-based gadgets. The Google Cloud Speech API allows you to convert language to text in more than 80 languages.



Recognition occurs remotely on Google servers, so the application will need to be connected to the Internet. In the presence of a good connection, recognition occurs almost immediately. The API can also separate the language from the noise in the input audio stream, which improves recognition quality. The Google Cloud Speech API can be used in a wide variety of applications and devices that can send REST or gRPC requests [30].

HTK (Hidden Markov Model Toolkit) is a language recognition tool that uses a hidden Markov model [31]. It is mainly intended for language recognition, but it is also used for other other recognition applications that use the hidden Markov model (including language synthesis, character recognition, and DNA ordering).

HTK consists of a suite of library modules and tools available in C. The tools provide sophisticated tools for language analysis, HMM training, testing and results analysis. The software supports HMM, using both Gaussian mixtures of continuous density and discrete distributions, and can be used to create complex HMM systems.

HTK was originally developed by the Machine Intelligence Laboratory (formerly known as the Speech Vision and Robotics Group) of the Engineering Department of Cambridge University (CUED), where it was used to create large speech recognition systems. In 1993, Entropic Research Laboratory Inc. acquired the rights to sell HTK and the development of HTK was fully transferred to Entropic in 1995. In 1999, Microsoft acquired Entropic. Microsoft has now licensed HTK back to CUED and provides support to allow CUED to redistribute HTK and provide development support through the HTK3 website [31].

Although Microsoft retains the copyright of the original HTK code, anyone can modify the source code and add it to HTK3.

The CMU Sphinx Speech Recognition Toolkit is a speaker-independent continuous recognition system that uses hidden Markov acoustic models (NMMs) and an n-gram statistical language model. It was developed by Kai Fu Li [32]. Sphinx demonstrated the ability to continuously recognize speech with a large lexical volume independent of the speaker.

Much of the CMU Sphinx development is underway at Carnegie Mellon University. At various times, both the Massachusetts Institute of Technology and the now deceased Sun Microsystems Corporation worked on the project. The project source codes are distributed under the BSD license and are available for both commercial and non-commercial use [33]. Sphinx is not a custom application, but rather a toolkit that can be used in end-user application development. It is dicto-independent and capable of recognizing the language in a continuous flow. Sphinx is now one of the big language recognition projects. It consists of several parts:

- Pocketsphinx - a small, fast program that processes audio, acoustic models, grammars and dictionaries;
- Sphinxbase library required for Pocketsphinx operation;
- Sphinx4 - its own recognition library;
- Sphinxtrain - a program for training acoustic models (human voice recordings).

Unlike HTK, it is used more for applied purposes than research.

Yandex SpeechKit is a technology of speech recognition and speech synthesis from the Russian company Yandex. The service infrastructure is designed to take into account high loads to ensure the availability and trouble-free operation of the system, even with a large number of simultaneous applications [34].

The service works with audio and texts in three languages: Russian, English and Turkish. Support for each of these languages is easy to add at any time - you will not need to connect third-party services when you reach a new audience [34]. Yandex SpeechKit is the language of more than a million individual phonemes, and the intonation is given by a neural network trained in many real-life examples. Therefore, when the system plays the text, the result does not cut the hearing.

The complex of Yandex language technologies has been successfully applied in various directions: multimedia car voice control system, which is a competitor of Car Play and Android Auto (signed contracts with KAMAZ, Toyota and Honda) [35]; Alice's voice assistant [36]; money transfer by voice in Binbank [37]; automatic

call tagging first introduced with Calltouch [38]; car audio (on the website of Izvestiya newspaper).

Microsoft Speech API, a product of Microsoft, performs parsing and synthesis of language in Windows applications. It is designed to be integrated as easily as possible through interfaces. It is also possible to create or modify existing speech recognition and synthesis mechanisms [39].

The recognition program (text recognition engine) iteratively compares the recognized text with grammar rules and, when matching text with a series of rules, generates an XML output stream using the SML semantics markup language. The output stream contains recognized text, values of probabilities of correct recognition and may contain semantic values assigned by markup interpretation of semantics [40]. Recognized text is typically used to dictate data and to control applications using language commands.

For a clear comparison of different speech recognition systems, we compile a summary table on their most important characteristics. Table 2.1 for Google Speech-To-Text and Yandex SpeechKit does not specify the size of the dictionary, as they are constantly trained using neural networks [41, 42].

Table 2.1 - Comparison of speech recognition systems

<b>Criteria</b>	<b>Google Speech-To- Text</b>	<b>Yandex SpeechKit</b>	<b>Microsoft Speech API</b>	<b>CMU Sphinx</b>	<b>HTK</b>
<b>API</b>	Yes	Yes	Yes	No	No
<b>Continuous language recognition</b>	Yes	Yes	Yes	Yes	Yes
<b>The size of the dictionary</b>	-	-	60 000	Up to 100 000	Up to 100 000
<b>Documentation</b>	Detailed online documentation	Available but not detailed	Detailed online documentation	Detailed online documentation	HTK Book - comprehensive information
<b>Support for</b>	Yes	Yes	Yes	Partly	No

<b>Russian</b>					
<b>Requests amount per day (free)</b>	Up to 60 minutes	Only canvas	Only canvas	Unlimited	Unlimited

The data from the studies described in the articles by fellow scientists were used for performance analysis [43, 44, 45]. The result is shown in table 2.2

Table 2.2 - Results of technology comparisons

	<b>Google Speech API</b>	<b>Yandex SpeechKit</b>	<b>Microsoft Speech API</b>	<b>CMU Sphinx</b>	<b>HTK</b>
<b>Getting features</b>	MFCC, PLP	MFCC, PLP	MFCC, PLP	MFCC, PLP, LPC	MFCC
<b>Acousting modeling</b>	Neural networks	HMM, Neural networks	HMM, Neural networks	HMM	HMM
<b>Language modeling</b>	WFST	WFST, a recurrent neural network	WFST	N-gramm, FST, CFG	N-gramm
<b>WER, %</b>	8,9	-	-	22,7	19,8
<b>WRR, %</b>	91,1	-	-	77,3	80,2
<b>SF</b>	0.4	-	-	1	1,4

Paid products were immediately discarded due to the inability to pre-test them. As a result of the technology review, CMU Sphinx was selected, namely Sphinx4 [46] on the Java platform. Even yielding precision to some competitors, the choice was refined for advantages such as cost - a free license (even in commercial projects); speed - the ability to work independently of the Internet without waiting for a

response from the server, not dependent on the bandwidth of the channel and other factors. Accuracy is not the main characteristic that a project needs because we compare the obtained result again with the original text.

### 2.3 Method for synchronizing text and audio modes of e-book reading

The text of the book (preferably, but not necessarily in the exact same edition / translation as audio) and audio goes to the input of the program. The program, using the Sphinx4 speech recognition library, conducts recognition from an audio file based on a pre-created dictionary with transcriptions. The library uses Hidden Markov Models for acoustic modeling. In recognition, the model is additionally trained, so the passage is made several times. This improves the accuracy of recognition of words from the audio file. Recognition occurs with a breakdown into "sentences", which consist of a small number of words. Immediately after recognizing the "sentence", the timings (the beginning and end of each word) are recorded. Further, the recognized text is matched aligned to the source text. This is to ensure that recognition does not take into account grammar symbols, such as periods, dashes, commas, etc. At the output of the program, we get a generated json file in which each object is a word from the original text (with original punctuation) and time intervals for the audio file. Flowchart of algorithm is shown on Figure 2.1.

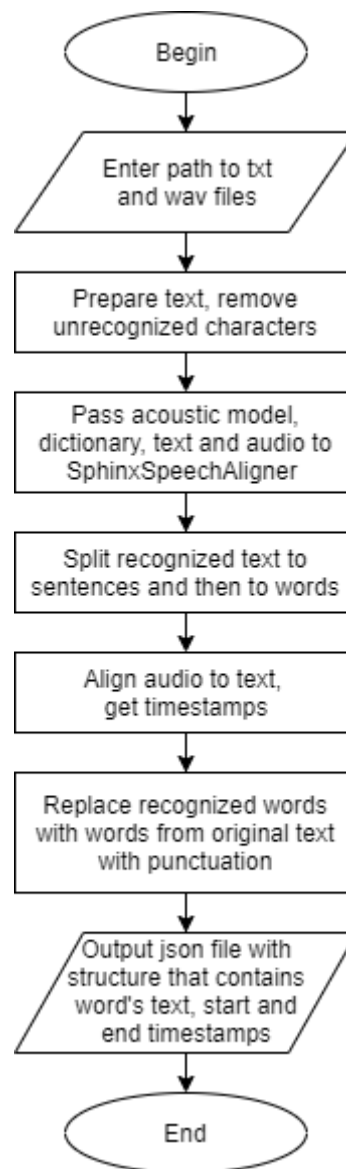


Figure 2.1 – Audio to text aligning algorithm

## 3 DEVELOPMENT OF THE SOFTWARE

### 3.1 Development of system architecture

#### 3.1.1 UML diagrams

##### 3.1.1.1 Use case UML-diagram

Use case diagrams describe the relationships and dependencies between the groups of use cases and the actors involved in the process. It is important to understand that the use case diagrams are not intended to display the project and can not describe the internal system of the system. Use case diagrams are designed to simplify interaction with future users of the system, with customers, and especially useful for determining the required characteristics of the system. In other words, the usage diagrams indicate what the system should do without specifying the methods themselves. The UML diagram of the use cases is presented in figure 3.1.

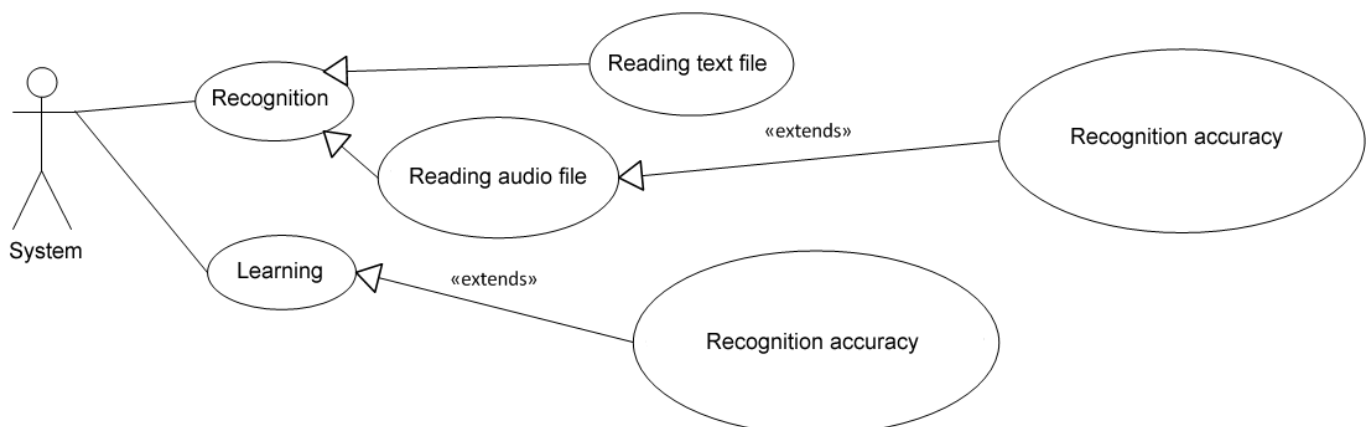


Figure 3.1 – Use case UML-diagram

Initially, the system learns from the audio samples of the speaker. When the system is trained, audio and text versions of the book are submitted for input. It recognizes the language, compares the result with the text of the original and matches it with timestamps and then saves it to a file.

3.1.1.2 Activity UML-diagram

Activity UML diagram is an UML diagram that shows the actions whose states are described in the state diagram. Activity refers to the specification of executable behavior in the form of coordinated sequential and parallel execution of subordinate elements - nested activities and individual actions of the English. action, interconnected flows that flow from the outputs of one node to the inputs of another.

The activity UML-diagram is shown in figure 3.2.

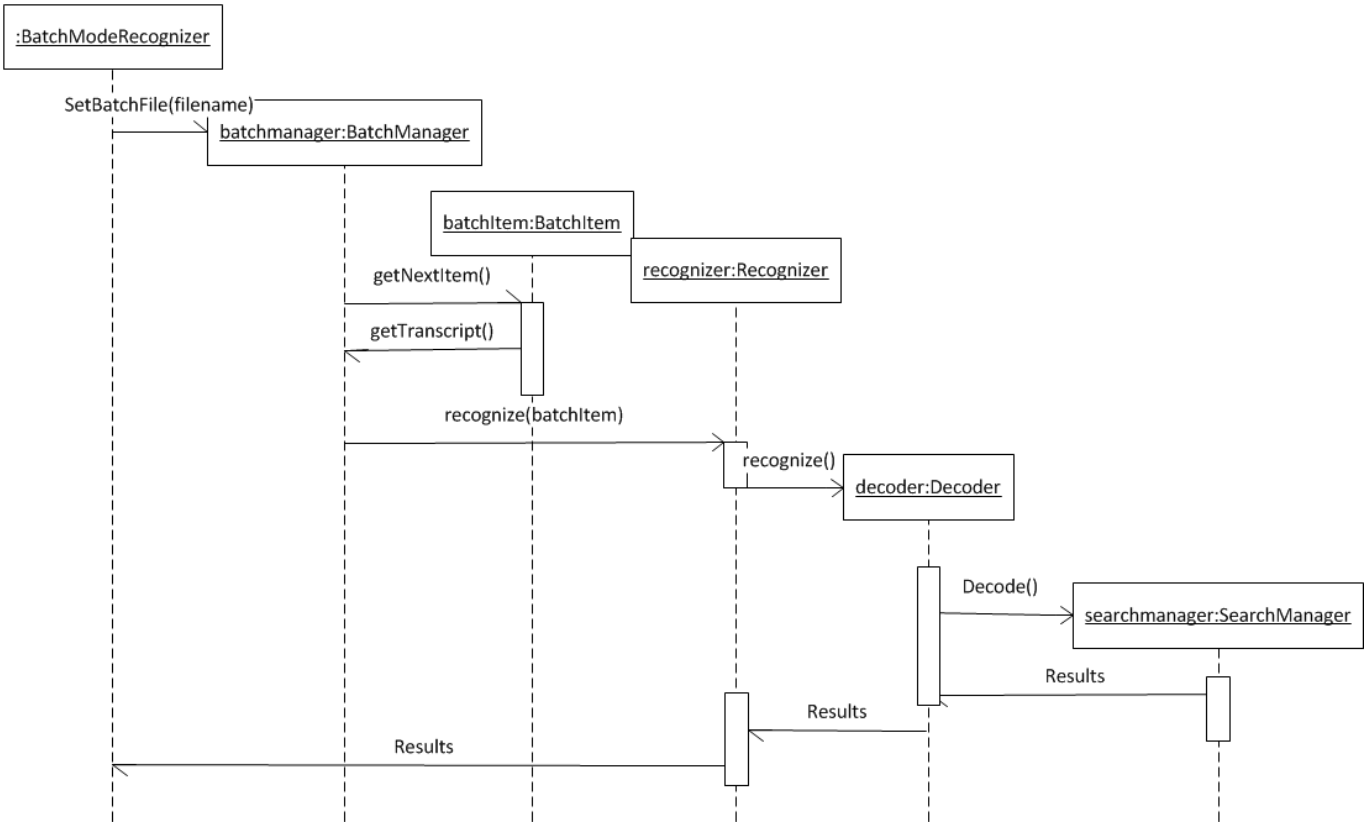


Figure 3.2 – Activity UML-diagram

3.1.1.3 Class UML-diagram

Class UML diagram is an UML diagram that describes the structure of a system by showing the system's classes, their attributes, operations or methods, and the relationships among objects.

The class UML-diagram is shown in figure 3.3.



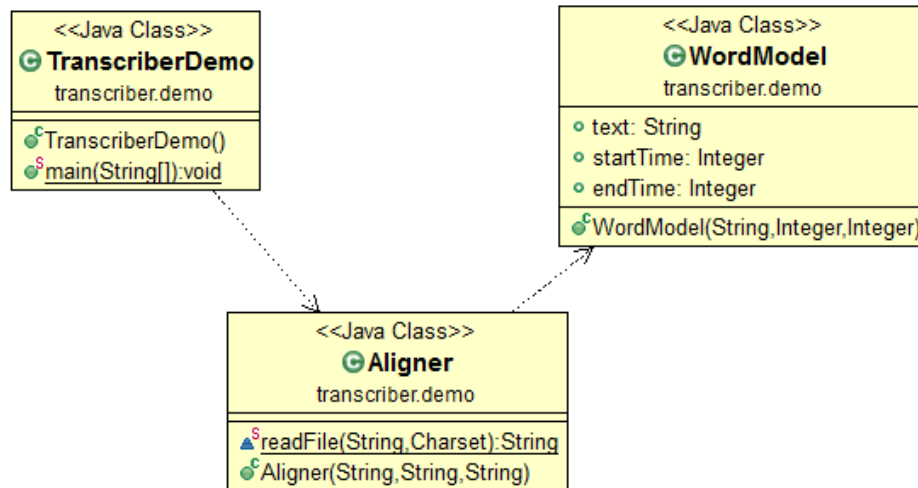


Figure 3.3 – Class UML-diagram

### 3.2 Rationale for the choice of software

The developed software consists of two projects. The first project is a system that processes input data, does language recognition and audio and text matching. Eclipse was used to develop the subsystem software in the Java programming language. Eclipse is a free modular integrated software development environment [47]. Convenient and free application, with a clear interface and the ability to add the necessary plugins during the development process. To build the project, the Apache Maven Project [48] was used. This is a framework for automating the assembly of projects based on the description of their structure in POM files. The structure of the pom.xml file with the dependencies used is shown dependencies E.

The second project is an application where the user will be able to use the result of the subsystem work, namely at any convenient time for him to switch from reading the text version of the book to audio, and vice versa. It was developed in the Visual Studio Code environment, a tool for creating, editing and debugging modern web applications and applications for cloud systems. Visual Studio Code is distributed free of charge and is available in versions for Windows, Linux, and OS X platforms [49].

To develop the mobile application, Angular 8 [50] and Ionic Framework 4 [51] were selected together with Cordova [52]. The use of Angular is due to the very fact that the application is divided into modules. Such a convenient structure makes it easy and simple to add new functions, make changes and expand the functionality of the developed software. Ionic and Cordova were chosen due to their cross-platform nature and the ability to easily create and copy an application for both Android, iOS or Windows.

### 3.3 Software development

#### 3.3.1 Development of analoging speech-to-text software

A Java-based application was developed to implement the audio file recognition system and match it to the original text of the book. As stated in section 1, the Sphinx4 library was selected.

In the absence of a sufficiently complete dictionary for the Russian language, it was decided not to use a ready-made version but to create one of its own. Python was perfect for the chosen goal, which is why it was used. A part of the generated dictionary is shown in figure 3.4.

```

абонемент а0 б о0 нj е0 мj е1 н t
абонемента а0 б о0 нj е0 мj е1 н t а0
абонементу а0 б о0 нj е0 мj е1 н t у0
абонементы а0 б о0 нj е0 мj е1 н t y0
абонент а0 б о0 нj е1 н t
абонента а0 б о0 нj е1 н t а0
абонентам а0 б о0 нj е1 н t а0 м
абонентами а0 б о0 нj е1 н t а0 мj i0
абонентов а0 б о0 нj е1 н t о0 v
абонентом а0 б о0 нj е1 н t о0 м
абонентский а0 б о0 нj е1 н t s kj i0 j
абоненту а0 б о0 нj е1 н t у0
абоненты а0 б о0 нj е1 н t y0
абонированную а0 б о0 нj i1 r о0 v а0 н n у0 j у0
абонированы а0 б о0 нj i1 r о0 v а0 н y0

```

Figure 3.4 – Dictionary for Russian language

An audio and text version of the book are submitted to the program entranceto the instance of Aligner class (Figure 3.5).

```
Aligner aligner = new Aligner("C://song.wav", "C://smallprince.txt", "C://3.json");
```

Figure 3.5 – Input data

Firstly, we have to prepare original text (remove unreadable characters) and pass info about dictionary and acoustic model to program. This part of code is shown on Figure 3.6

```
URL audioUrl;
String transcript, originalText;
String acousticModelPath = "file:src/main/resources/model";
String dictionaryPath = "file:src/main/resources/model/ru.dic";
audioUrl = Paths.get(audioPath).toUri().toURL();
originalText = readfile(textPath, StandardCharsets.UTF_8);
StringBuilder original = new StringBuilder (originalText);
transcript = originalText;
transcript = transcript.replaceAll("[^а-яА-ЯёЁаа-зА]", " ");
transcript = transcript.replaceAll("\\p{Cc}", "");
transcript = transcript.trim().replaceAll(" +", " ");
```

Figure 3.6 – Text preparations

Immediately after recognizing a word from an audiofile, we create structure with recognized word and it's timestamps. Code is shown on Figure 3.7.

```

List<WordResult> results = aligner.align(audioUrl, transcript);
List<String> stringResults = new ArrayList<String>();
for (WordResult wr : results) {
    stringResults.add(wr.getWord().getSpelling());
}
LongTextAligner textAligner = new LongTextAligner(stringResults, 2);
List<String> sentences = aligner.getTokenizer().expand(transcript);
List<String> words = aligner.sentenceToWords(sentences);
int[] aid = textAligner.align(words);
int lastId = -1;
for (int i = 0; i < aid.length; ++i) {
    if (aid[i] == -1) {
        System.out.format("- %s\n", words.get(i));
    } else {
        if (aid[i] - lastId > 1) {
            for (WordResult result : results.subList(lastId + 1,
                aid[i])) {
                System.out.format("+ %-25s [%s]\n", result.getWord()
                    .getSpelling(), result.getTimeFrame());
            }
        }
        wordsArray.add(new WordModel(results.get(aid[i]).getWord().getSpelling(),
            (int)results.get(aid[i]).getTimeFrame().getStart(),
            (int)results.get(aid[i]).getTimeFrame().getEnd()));
        System.out.format("  %-25s [%s]\n", results.get(aid[i])
            .getWord().getSpelling(), results.get(aid[i])
            .getTimeFrame());
        lastId = aid[i];
    }
}
if (lastId >= 0 && results.size() - lastId > 1) {
    for (WordResult result : results.subList(lastId + 1,
        results.size())) {
        System.out.format("+ %-25s [%s]\n", result.getWord()
            .getSpelling(), result.getTimeFrame());
    }
}
}

```

Figure 3.7 – Words recognition, timestamps adding

Next we have to find the word in the original text (not anywhere, but gradually moving through the text, discarding the already processed one). Figure 3.8 shows this part of the code.

```

int startIndex = 0;
int endIndex = 0;
String searchedWord;
ArrayList<WordModel> finalArray = new ArrayList<WordModel>();
int oldIndex = 0;
for (int i=1; i<wordsArray.size(); i++) {
    searchedWord = wordsArray.get(i).text;
    startIndex = original.toString().toLowerCase().indexOf(searchedWord.toLowerCase(), endIndex);
    if (startIndex !=-1) {
        endIndex = startIndex + searchedWord.length();
        //adding text for previous iteration
        if (wordsArray.get(i-1).startTime != null) {
            finalArray.add(new WordModel(original.substring(oldIndex, startIndex),
                wordsArray.get(i-1).startTime,
                wordsArray.get(i-1).endTime));
            oldIndex = startIndex;
        }
    }
}
// adding last record into array
finalArray.add(new WordModel(original.substring(oldIndex),
    wordsArray.get(wordsArray.size()-1).startTime,
    wordsArray.get(wordsArray.size()-1).endTime));

```

Figure 3.8 – Original words searching

And finally, the result, namely the timings (beginning of the word/phrase sound and its end) are recorded as a JSON file (Figures 3.9, 3.10) keeping the punctuation, paragraphs of the original.

```

//writing to json
Gson gson = new Gson();
String json = gson.toJson(finalArray);
try (PrintWriter out = new PrintWriter(jsonPath)) {
    out.println(json);
}

```

Figure 3.9 – Output results to file

```

{
  "text": "никому ",
  "startTime": 7890,
  "endTime": 8290
},
{
  "text": "не ",
  "startTime": 8300,
  "endTime": 8430
},
{
  "text": "интересная ",
  "startTime": 8440,
  "endTime": 9080
},
{
  "text": "желтая звезда.\r\n\r\nВокруг нее, на ",
  "startTime": 12420,
  "endTime": 12550
},
{
  "text": "расстоянии ",
  "startTime": 12560,
  "endTime": 13200
},
{
  "text": "примерно ",
  "startTime": 13210,
  "endTime": 13730
},
{
  "text": "девятиста ",
  "startTime": 13740,
  "endTime": 14350
}
}

```

Figure 3.10 – Output JSON file

The following files will be used in the end-user application.

### 3.3.2 Development of a mobile application

Using a modular structure according to Angular concepts made the development very convenient. The structure of the mobile application is shown in Figure 3.11.

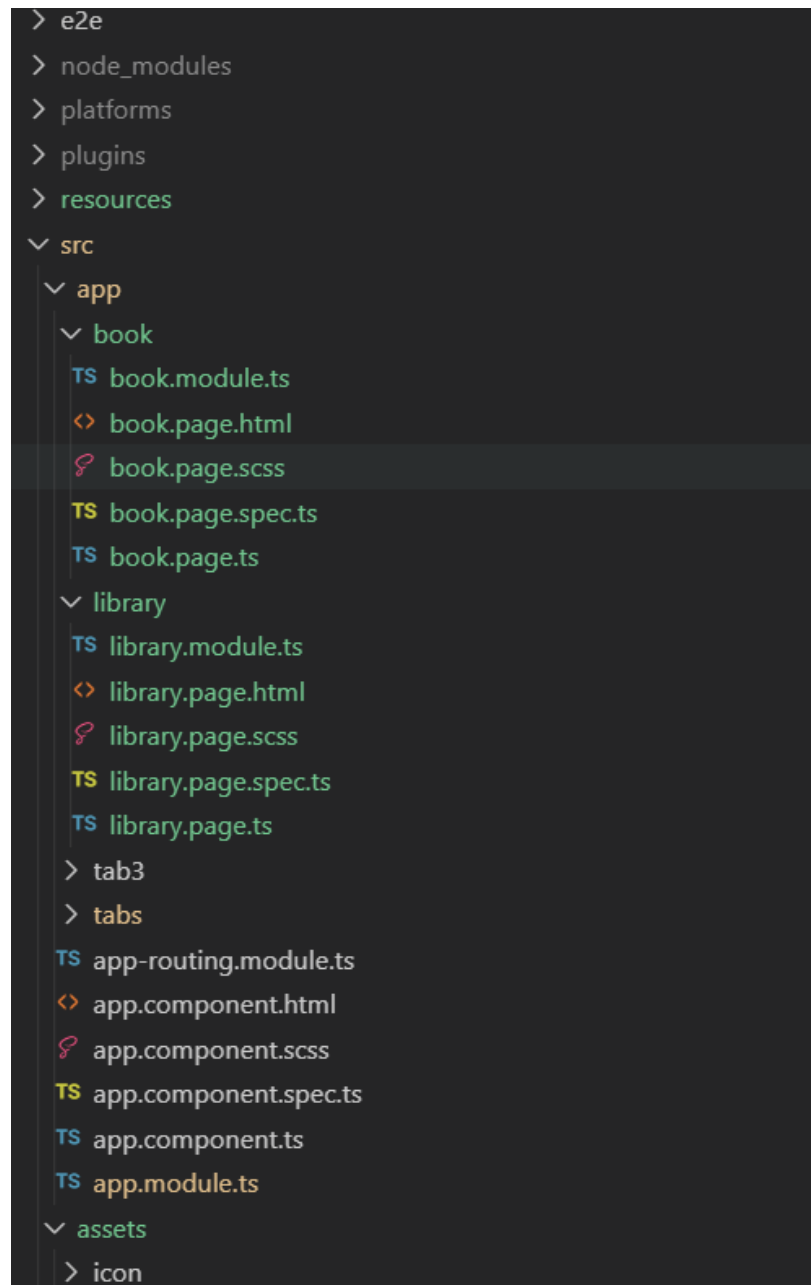


Figure 3.11– Mobile application modules structure

Below is a description of each component of the mobile application. The component "library". This component displays books that have been downloaded from the server to the library. Using convenient Angular technologies, we render each book in a loop (looping through all the available books). The component template code is depicted in Figure 3.12. For each book, the title of the book, the authors of the book, and the cover are rendered. When you click on an element with the class "book", the openBook function is called.

```

<ion-content>
  <div class='books-container'>
    <div *ngFor="let book of books" class='book' (click)="openBook(book)">
      <img class='artwork' src={{book.imageUrl}}>
      <div class='title'>{{book.title}}</div>
      <div class='authors'>{{book.authors}}</div>
    </div>
  </div>
</ion-content>

```

Figure 3.12– Template of "library" component

There are only a few features in the current component. The first, "getBooksList", is responsible for retrieving books from the server (Figure 3.13). The second, "openBook," is responsible for navigating when selecting a book, and transferring the book structure to the next component using router and NavigationExtras (Figure 3.14). After calling this function, the user will redirect to the "book" component.

```

getBooksList() {
  this.http.get('assets/books.json').subscribe(res => {
    this.books = res;
  },
  (err) => {
    alert('Failed loading a list of books');
  });
}

```

Figure 3.13– getBooksList function

```

openBook(book) {
  let navigationExtras: NavigationExtras = {
    state: {
      book: book
    }
  };
  this.router.navigate(['tabs/book'], navigationExtras);
}

```



Figure 3.14– openBook function

The template component of the book (Figure 3.15) consists of two main elements - the player and the block with the text of the book. The player receives the url of the audio file of the book. In the block with the "fulltext" class, the entire text of the book is rendered. The main feature is that each word (several words in some cases) is a separate span element with the class "word-wrapper".

These elements have quite a few attributes: data-starttime - the timestamp value of the word beginning is recorded, data-endtime - the timestamp value of the end of the word is recorded, id is exactly the same as date-starttime. It is worth paying attention to the fact that if the current word does not have the value of the startTime property, then the word is rendered without all the properties presented above. When you click on such a word, nothing will happen, unlike the previous ones. When you click on a word, the wordClick function is called (Figure 3.22).

```

<ion-content>
  <audio preload="auto" id="player" controls>
    <source [src]=book.audioUrl type="audio/mpeg">
    Your browser does not support the audio element.
  </audio>
  <div class='fulltext'>
    <ng-container *ngFor="let word of data" >
      <span *ngIf="word.startTime !== undefined" class='word-wrapper'
        (click)="wordClick($event.target)"
        [attr.data-starttime]=word.startTime
        [attr.data-endtime]=word.endTime
        [attr.id]=word.startTime
        [style.background-color]="closestData == word.startTime ? 'red' : 'white'">
        {{word.text}}
      </span>
      <span *ngIf="word.startTime == undefined" class='word-wrapper'>
        {{word.text}}
      </span>
    </ng-container>
  </div>
</ion-content>

```

Figure 3.15– Template of "book" component

Another very interesting point is the style attribute "background-color", namely this line: [style.background-color] = "closestData == word.startTime? 'red': 'white'". In

this place, depending on the value of the `closestData` variable determines the color of the background word of the word, the color changes in real time and the variable is constantly checked when its value changes.

The code for the book component consists of several functions. At the very beginning, even before the component is initialized, it is checked whether the object of the current book has arrived. If not, the user is redirected back to the "library" component. The process is depicted in Figure 3.16.

```
ionViewWillEnter() {
  console.log(this.book);
  if (this.book == null) {
    this.router.navigate(['tabs/library']);
  } else {
    this.getBookData();
  }
}
```

Figure 3.16– `ionViewWillEnter` function

When the component is initialized (Figure 3.17), we get the player object by its id and add eventListeners to it.

```
ngOnInit() {
  this.player = <HTMLAudioElement>document.getElementById("player");
  this.player.addEventListener("timeupdate", () => {
    this.selectWord();
  });
  this.player.addEventListener("play", () => {
    this.audioPlaying();
  });
  this.player.addEventListener("pause", () => {
    this.audioPaused();
  });
}
```

Figure 3.17– «Book» component `ngOnInit`

The `selectWord` function (Figure 3.18) is responsible for the value of the global variable `closestData`, which, as mentioned above, determines the background-color of each word.

```
selectWord() {
  this.closestData = this.closestWord(this.player.currentTime * 1000, this.arrTimeStampsStart);
}
```

Figure 3.18– `selectWord` function

The `closestWord` function (Figure 3.19) is quite interesting, it responds to finding a word whose `startTime` timestamp is closest to the current time value of the music player. The value of time and an array with all `startTime` timestamps are passed to the function.

```
closestWord(num, arr) {
  var curr = arr[0];
  var diff = Math.abs(num - curr);
  for (var val = 0; val < arr.length; val++) {
    var newdiff = Math.abs(num - arr[val]);
    if (newdiff < diff) {
      diff = newdiff;
      curr = arr[val];
    }
  }
  return curr;
}
```

Figure 3.19– `ClosestWord` function

Functions `audioPlaying` and `audioPaused`, shown in Figure 3.20, are responsible for transferring the current state of the music player to `MusicControls` (an element with buttons and the ability to control the player from the curtain).

```

audioPlaying() {
  this.settingMusicControl();
}

audioPaused() {
  this.musicControls.listen();
  this.musicControls.updateIsPlaying(false);
}

```

Figure 3.20– audioPlaying and audioPaused functions

The `getBookData` function (Figure 3.21) is called at the very beginning, in order to prepare the component and get all the data from the book. If the user selected the same book, then the player will not be stopped. The data variable is also prepared (the presence of `startTime` timestamps is checked).

```

getBookData() {
  if (this.book.id !== this.oldBookId) {
    this.player.pause();
  }
  this.http.get(this.book.textUrl).subscribe(res => {
    this.data = res;
    this.data.forEach(each => {
      if (each.startTime !== undefined) this.arrTimeStampsStart.push(each.startTime);
    });
  },
  (err) => {
    alert('Failed loading json data');
  });
  this.oldBookId = this.book.id;
}

```

Figure 3.21– `getBookData` function

Function `wordClick` is called when you switch the time slider on the audio player. A small function is depicted in Figure 3.22.

```

wordClick(elem) {
  this.player.currentTime = elem.dataset.starttime / 1000;
}

```

Figure 3.22– `wordClick` function

settingMusicControl is a fairly large function that is responsible for setting the music controls in the curtain. Here, clicks and various states are controlled, which are transmitted to the player and from the player to the buttons. The function code is shown in Figure 3.23.

```

settingMusicControl(){
  this.musicControls.destroy(); // it's the same with or without the destroy
  this.musicControls.create({
    track: this.book.title,           // optional, default : ''
    artist: this.book.authors,         // optional, default : ''
    cover: this.book.imageUrl,        // optional, default : nothing
    isPlaying : true,                  // optional, default : true
    dismissable : true,                // optional, default : false
    hasPrev : false,                   // show previous button, optional, default: true
    hasNext : false,                   // show next button, optional, default: true
    hasClose : true,                   // show close button, optional, default: false
    hasSkipForward : false,            // show skip forward button, optional, default: false
    hasSkipBackward : false,           // show skip backward button, optional, default: false
    skipForwardInterval: 15,           // display number for skip forward, optional, default: 0
    skipBackwardInterval: 15,         // display number for skip backward, optional, default: 0
  });
  this.musicControls.subscribe().subscribe((action) => {
    console.log('action', action);
    const message = JSON.parse(action).message;
    console.log('message', message);
    switch(message) {
      case 'music-controls-pause':
        this.player.pause();
        this.musicControls.listen();
        this.musicControls.updateIsPlaying(false);
        break;
      case 'music-controls-play':
        this.player.play();
        this.musicControls.listen();
        this.musicControls.updateIsPlaying(true);
        break;
      case 'music-controls-destroy':
        this.player.pause();
        //moving to current word to resume reading
        document.getElementById(this.closestData).scrollIntoView();
        break;
      default:
        break;
    }
  });
  this.musicControls.listen(); // activates the observable above
  this.musicControls.updateIsPlaying(true);
}

```

Figure 3.23– settingMusicControlfunction

When you are ready to deploy your Ionic-Cordova application, you have various options for deployment.

For the simplest deployment, create a production build. Start with the production build using the next command: `ionic cordova build --release android` or `ionic cordova build --release ios` (according to platform you needed). This approach will create a compressed and ready to use apk file.

Builded application file will be located in the output folder (`%appname%\platforms\%platform_type%\app\build\outputs\apk\release`), depending on chosen platform. This is the simplest production-ready deployment of your application which will be ready to use for any end users.

### 3.4 Description of mobile application

The presented description of the mobile application is based on the Android platform. For iOS, all functionality will be similar with minor changes in the visual part of the interface, according to the current trends in design.

When the user starts up the application he is seeing the library page that is showing books user is able to choose. Library page is shown in figure 3.24.



Figure 3.24– Library page

On library page user can choose a book he wants to read or continue reading from the last position.

The next page is Book page. This is the module where the user can read the text of the book itself. Bookpage is shown in figure 3.25.

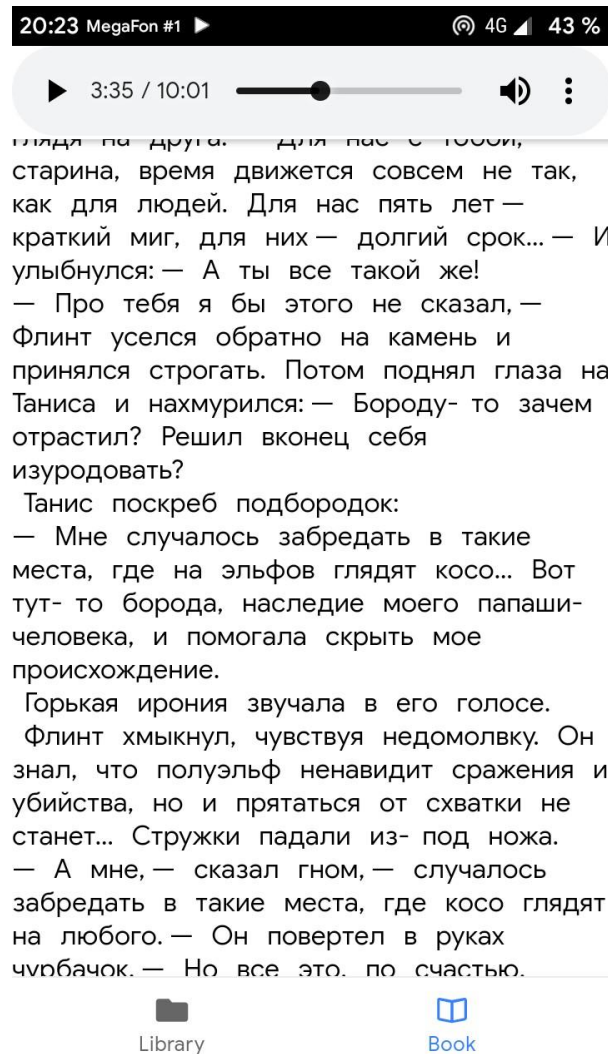


Figure 3.25– Bookpage

When you turn on the audiobook playback mode, media controls appear on the notification panel, on which you can see the name of the book, author, cover and buttons to enable/disable playback. When closing media controls, the user will automatically switch to the current place in the text of the book. Media controls are shown in figure 3.26.



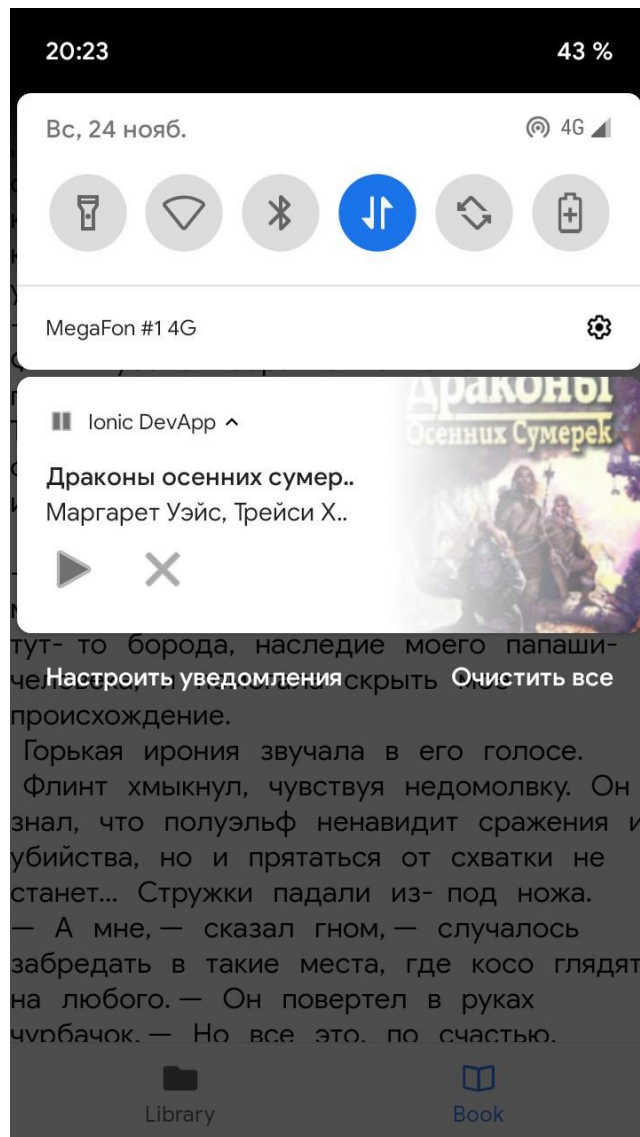


Figure 3.26– Media controls

### 3.5 Supporting documentation

This computer system is designed to automatically align audio to text of a book. Using a language recognition library, an input file system creates a json file that contains text splitted to words with timestamps. This system has been developed in Java programming language in Eclipse environment.

Software Requirements:

- Your computer must be running Windows 7 or higher (iOS or Linux may also available because of cross-platform).

- Eclipse programming environment.

Technical Requirements:

- disk space should be at least 400 MB;
- RAM must be at least 2000 MB;
- The graphics card should be 64 MB.

To run the program, you need to extract the *Aligner.rar* archive into the directory. To run the program, you need to run the *aligner.exe* file through the console with submitting a txt text file and a wav audio file to the input. The final json file that will be created after the data is processed will be created in the same directory.

All the necessary files are contained in the *Aligner.rar* archive on disk.

## 4 QUALITY ASSESSMENT OF THE DEVELOPED SYSTEM

### 4.1 Choosing a quality criterion

To select a quality criterion for evaluating the results of the work, a Word Error Rate (WER) was chosen. WER is a generally accepted criterion for the quality of the speech recognition system. In essence, it reflects the proportion of incorrectly recognized words and is determined by the formula 4.1 [54].

$$WER = \frac{S+I+D}{N} * 100\% = \frac{S+I+D}{C+S+D} * 100\% \quad (4.1)$$

where  $S$  is the number of substitutions (anytime a word gets replaced),  $D$  is the number of deletions (anytime a word is omitted from the transcript),  $I$  is the number of insertions (anytime a word gets added that wasn't said),  $C$  is the number of correct words,  $N$  is the number of words in the reference ( $N=S+D+C$ ).

Example of "recognition issues" is in Figure 4.1 [55]

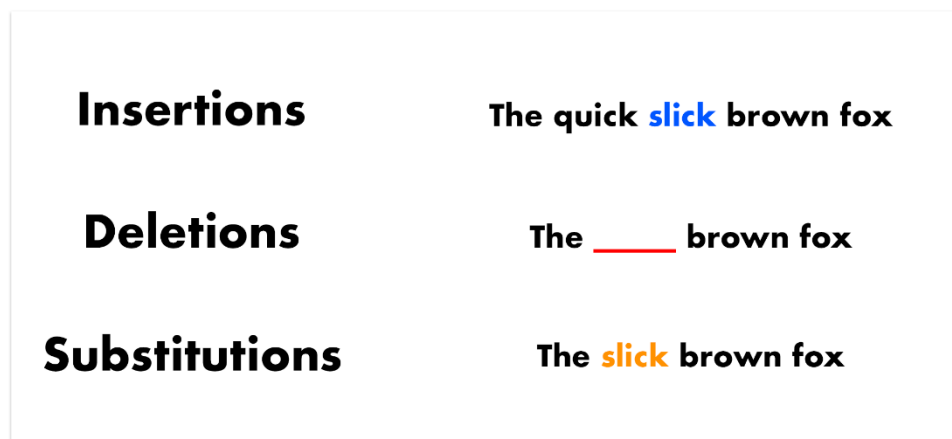


Figure 4.1 – Example

In our case, the appearance of insertions is impossible due to the comparison of audio with the original text of the book. Substitutions are also screened out before the json file is generated (words that were incorrectly recognized are marked as unrecognized). Therefore, we will proceed from the number of unrecognized words.

## 4.2 Description of test data for quality assessment

Due to the peculiarities of the json file with recognized text (Figure 3.10), the data will first need to be prepared. The file structure is such that the "startTime" and "endTime" fields are only present if the data from the "text" field were recognized. Another feature is that in the "text" field there can be either one word or several (if they were not recognized, they are combined with the nearest recognized word). Also, punctuation marks should be excluded from this field.

For simplicity, a small feature was created in the Javascript programming language. It loops through the file, analyzes the fields based on the logic above. The count of all words and the count of unrecognized words are recorded in the variables deletions and wordsCount, respectively. Next, to obtain WER, we use formula 4.1, which is given in the section above. Javascript code is given in figure 4.2

```
function getWER(json) {
  deletions = 0;
  wordsCount = 0;
  json.forEach(each => {
    let words = each.text.split(/[\\s,]+/).filter(el => { return el.length != 0 });
    if (words.length > 1) {
      wordsCount += words.length;
      if (each.startTime == null || each.endTime == null) {
        deletions += words.length;
      } else {
        deletions += words.length - 1;
      }
    } else {
      wordsCount++;
      if (each.startTime == null || each.endTime == null) deletions++;
    }
  });
  console.log('Deletions: ' + deletions);
  console.log('Words Count: ' + wordsCount);
  console.log('WER: ' + deletions*100/wordsCount);
}
```

Figure 4.2 – Get WER function

### 4.3 Quality assessment results

The data was analyzed based on three texts in Russian for which speech recognition and aligning were performed: The Hitchhiker's Guide to the Galaxy by Douglas Adams (narrator - Sergey Kuznetsov); Dragons of Autumn Twilight by Margaret Weis and Tracy Hickman (narrator - SHRDLU); The Little Prince by Antoine de Saint-Exupery (narrator - Alexander Glushkov).

These three texts and three audio files, respectively, were submitted to the system. After analyzing the received files using the written function (Figure 4.2), we got the values of WER for each of them. Table 4.1 summarizes the results.

Table 4.1 - Quality assessment results

<b>Text</b>	<b>Words count</b>	<b>Deletions count</b>	<b>WER, %</b>
The Hitchhiker's Guide to the Galaxy	1651	142	8.6
Dragons of Autumn Twilight	1587	177	11.15
The Little Prince	10705	2334	21.8

WER is pretty good for the first two texts. According to table 1.2, the average WER for speech recognition for CMU Sphinx is 22.7%. We were able to reduce the number of errors by aligning audio to text. It is worth paying attention to the fact that for the third book the percentage of errors is higher. This is due to the fact that the text and audiobook were not completely identical (due to different translations from a foreign language). But still, this is a pretty good result.

## 5 PLAN FOR IMPLEMENTATION AND DISSEMINATION OF THE DEVELOPED SYSTEM

### 5.1 Market and consumer analysis

The main prospective consumers of the developed product are smartphone users who are interested in reading as well as listening to books. It is known that many people used Ebooks before, however, since 2013, according to statistics, the number of such users is decreasing every year [66].

Smartphones, which are an integral part of the life of almost every person, quite successfully assumed the function of devices for reading books.

Another focus group is educated people. Students and schoolchildren will benefit from a resource that makes it possible to study the material even in those conditions in which reading can be difficult. For example, transport, the corridor of the educational institution, street.

Also, the application will be useful to people with disabilities. In particular, blind or visually impaired. In the first case, the help of another person is needed, who can start listening to the book.

According to statistics, only about 43% of Ukrainians and 53% of Russians read one book a year [67, 68]. This is a rather small volume, which may be due to the inconvenience of reading in conditions of a fast pace of life. In today's world, people are more and more inclined toward the auditory study of information, with which this application, created based on the preferences of modern people, can help.

Based on all of the above, you should occupy the market based on the quality of the books offered. Definitely, it makes sense to pay attention to the recently published books, bestsellers, classical works, while updating the application library as often as possible. It is also important not to forget about the educational literature, which students of different profiles and schoolchildren will need. In this case, not only a diverse choice is important, but also a selection of books relevant to the scientific world.

### 5. 2 EnteringtheAndroidandiOSmarkets

The application was originally developed as cross-platform, so it is suitable for both Android and iOS devices. According to statistics, the ratio of Ukrainian user users is approximately 79% Android and 18% iOS. The graph is shown in Figure 5.1 [69].

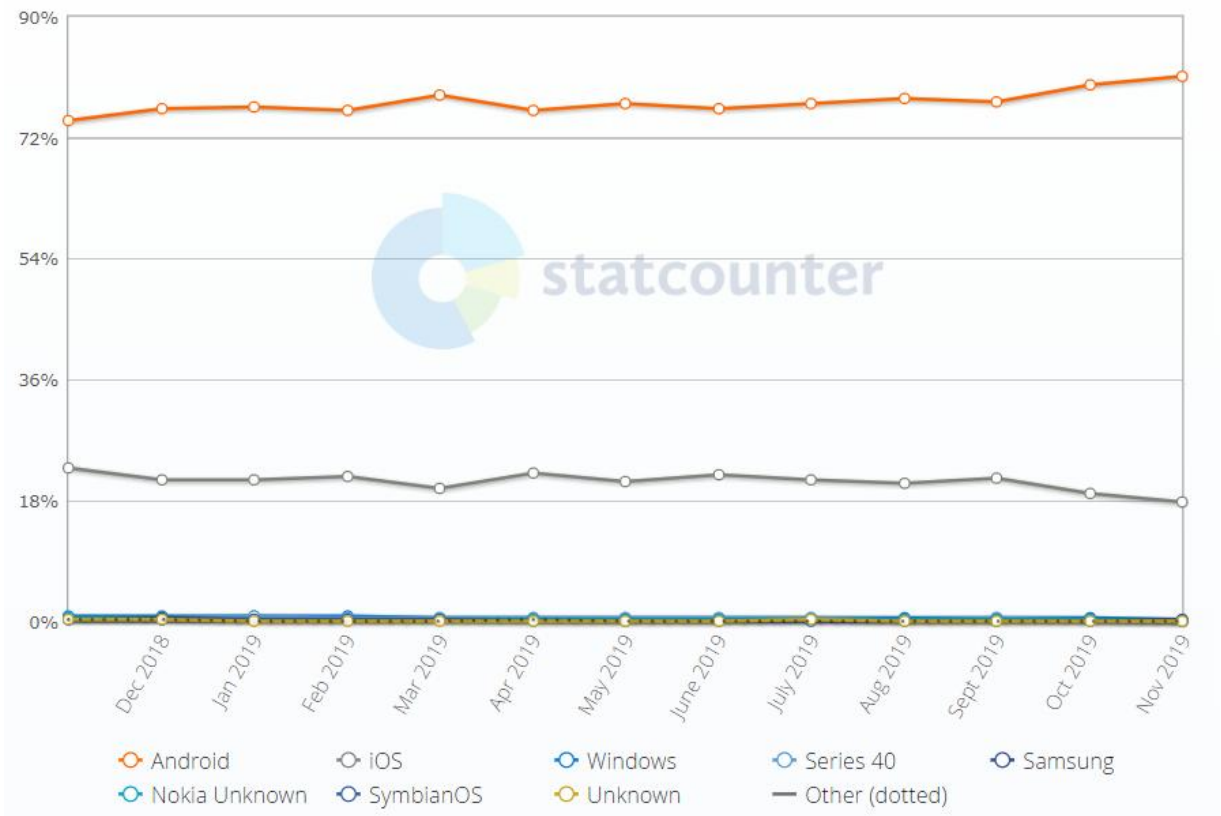


Figure 5.1 – Mobile OS

Today, Android maintains a leading position in the number of users in the world by a wide margin from other market participants.

The financial capabilities of users of the two main operating systems are regularly compared by almost all IT research firms. From year to year, the results do not change dramatically, but you can trace some dynamics in the existing changes. The research company App Annie has clearly demonstrated that despite the increase in downloads of applications from Google Play, Apple users are much more willing to pay for content. (Figures 5.2 and 5.3) [70].



Figure 5.2 – Downloads

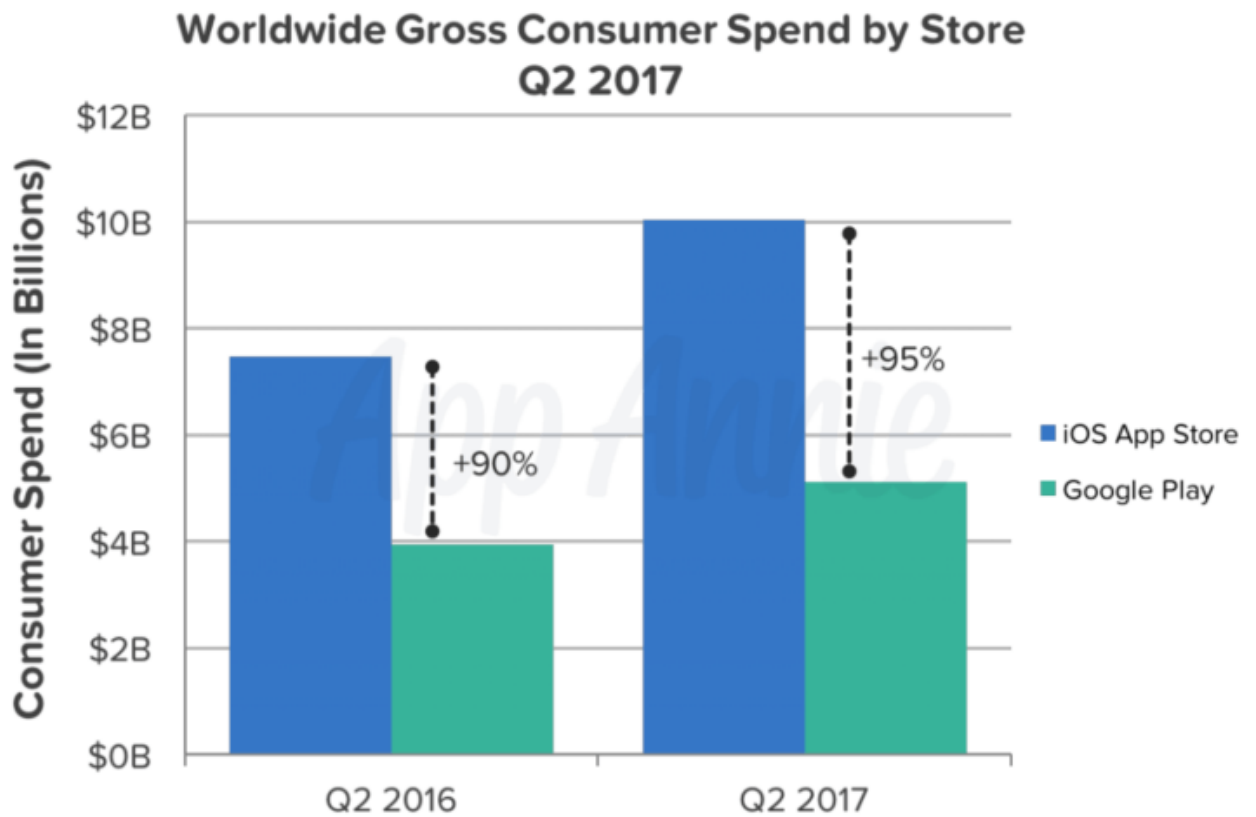


Figure 5.3 – Consumer Spend



Speaking about the dynamics, I want to pay attention to the gap in revenue between the App Store and Google Play - it is almost not narrowing. This is where the issue of solvency of device owners on these platforms is revealed.

### 5.2.1 Google Play

Steps for publishing the application on Google Play [71]:

- Create an account in the Google Play Developer Console
- Checkout privacy policy.
- Prepare marketing materials (icon, screenshots, APK, banner, text, promo video).
- Ensure assembly with a digital signature certificate.
- Set up payment for using the application.
- Send the assembly to Google Play.

### 5.2.2 App Store

Steps for publishing on the App Store [72]:

- Create an account on the App Store
- Publish the application from the developer account
- Prepare marketing materials (icon, screenshots, text, video for previewing the application);
- Attach a digital signature certificate to the assembly;
- Set up payment for using the application;
- Submit the assembly to the App Store.

## 5.3 Application promotion methods

There few important steps to promote a developed application.

Step 1. Get the audience involved in creating the application

In order to bring the product to the top, you must provide thousands or tens of thousands of installations in the first days after adding the application to the store. Unmotivated downloads are easier to get if users know and are waiting for your

product. Hitster service developers recommend testing the application with audience before run [73]. With the help of testing, you solve the following tasks:

- Introduce the audience to the product;
- Form loyalty;
- Get information about bugs and missing features of the application.

The test before launch gives you interested users and the opportunity to modify the product in accordance with the opinion of the audience.

- Promise active test participants free access to paid features;
- Win valuable prizes between volunteers;
- Announce the publication of the application and invite a potential audience to participate in the tests. The announcement can be published on the corporate website, in social networks, on specialized resources.

Step 2: Ensure the online presence of the project

Choose the online presence method depending on the product theme and audience preferences. Create a landing page for ad campaigns and product promotion in natural search. Public in social networks will appeal to the audience of gaming applications. You can also create a separate site dedicated to the program and a description of its capabilities.

Step 3. Ensure app visibility in stores

To add the product to Google and Apple stores, you will need developer accounts. When adding applications, it is important for you to ensure their visibility in the search and attractiveness to the user.

To let users find your application on Google Play, follow these guidelines:

- Do not limit yourself to the name of the application. Google Play allows you to enter 30 characters in the title. In addition to the name, try to fit into this limit a brief description of the product.

- Publish a detailed description of the program. Its length can be up to 4 thousand characters. Create text for people, but naturally use search phrases in it with which users can search for your application. Google recommends using the basic "keys" up to five times. Take care of the readability of the description: structure the

text, use bulleted or numbered lists. Describe in detail the functionality and benefits of the product.

- Publish a promotional video application.
- Post screenshots of the application. With their help, you can show the capabilities of the program, as well as make the description page on Google Play visually attractive.
- Make an attractive icon. This element should attract the attention of users in application stores and on the screens of mobile gadgets.

You can use these guidelines when publishing an app on the App Store. Please note that the Apple App Store has a separate field for keywords. Indicate in it the "keys" by which users search for your product.

#### Step 4. Tell people about the application

Publish information about the application on the corporate website, the application website and in the public on the social network. Be sure to notify the release of specialized editions. If you have released an application oriented to the CIS market, write to the editors of such resources: AppleInsider, Ferra.ru, w3bsit3-dns.com, Life hacker, AppStudio, Zuckerberg will call, Planet iPhone.

Send press releases to editors on related sites and blogs. For example, if you are promoting a game application, write about it in the editorial board of game portals. And if you have released a mobile trading platform or financial news aggregator, feel free to send the release to the editors of Bloomberg and RBC.

#### Step 5: Post reviews on themed sites

If you have created a really good and relevant application, some specialized publications will write about you for free. But you cannot accurately predict who and when will release material about your product. Therefore, publish paid reviews in several publications.

The choice of sites for paid publication is determined by the application theme, promotion budget, and market geography. If you created a program for Russian-speaking users, pay for the publication of the review on w3bsit3-dns.com,

Macdigger.ru, iPhones.ru and other sites and blogs that the potential audience of your product reads.

Create an information kit or press kit for publication authors. Include the following in it:

- Press release;
- Links to the application;
- Information about the application. Indicate the properties and features of the program that should be mentioned in the review;
- Information about the developers: facts about the company, achievements, awards, success stories;
- Icons and screenshots;
- Contact details, including links to the site and social networks.

#### Step 6: Socialize the application

This step is implemented at the development stage. Create conditions under which users can easily recommend your product to friends, share achievements. Here are some ideas:

- Provide the ability to log in using profiles on social networks;
- Add the ability to share information, screenshots, successfully solved tasks on social networks;
- Integrate notifications of achievements of friends on social networks.

#### Step 7: Ask users to rate the app

You remember that ratings affect the position of apps in stores. Therefore, integrate offers into the program interface to evaluate and comment on the product. The following recommendations will help you get more good grades:

- Invite the user to evaluate the application before shutting down or after several successful actions.
- Do not show a reminder every time you start the program.
- Owners of smartphones and tablets are always in a hurry. Therefore, use clear and short calls to action in reminders.

- Use the trick from the creators of the Mint application: offer to rate in the program interface. If the user puts from one to three points, invite him to write to the developers and get help. If he puts four or five points, redirect him to the store asking him to confirm the rating.

Requests to evaluate the product distract the user from solving current problems. In this sense, they can be compared with pop-ups.

#### Step 8: Advertise the application

Use Yandex and Google contextual advertising systems, advertise the application on social networks, and display banners to visitors to theme sites.

To improve your ad performance, follow these guidelines:

- Choose the morning and evening time for publishing ads on public social networks. Users view news feeds on the way to and from work.

- Link to the application using a QR code. Use it on printing products, product packaging, presentation slides, t-shirts.

Measure the effectiveness of advertising by the ratio of impressions or conversions and installations. This will help you determine the cost of attracting a user through advertising and evaluate the financial effectiveness of this method of promotion.

#### Step 9: Drive installs with email distribution

If you have a subscriber base, tell her about the application after adding it to stores. Devote a separate letter to the product: tell us about its capabilities and refer to the program on Google Play and the App Store.

Remind users periodically about the existence of the application. Be sure to include the link to the application in stores in the letter template.

#### Step 10: Emphasize the possibility of a free installation

Do not take money for installing the application. Do not tell your audience about a paid account. Allow users to install and run the application for free. Do not scare customers by advertising. Give your audience the opportunity to appreciate the benefits of your product. Invite users to give up money only after they realize that they cannot live without an application.

### Step 11: Use video advertising

Show it on YouTube and other video services. Check the performance of TV shows. For example, Amazon is actively showing commercials to viewers. According to research company Nielsen, 40% of smartphone and tablet owners use mobile gadgets daily while watching TV [74].

### Step 12: Reduce the size of the application as much as possible

Pay attention to the illustration in Figure 5.4 [75] of the dependence of the installation frequency on the application file size.

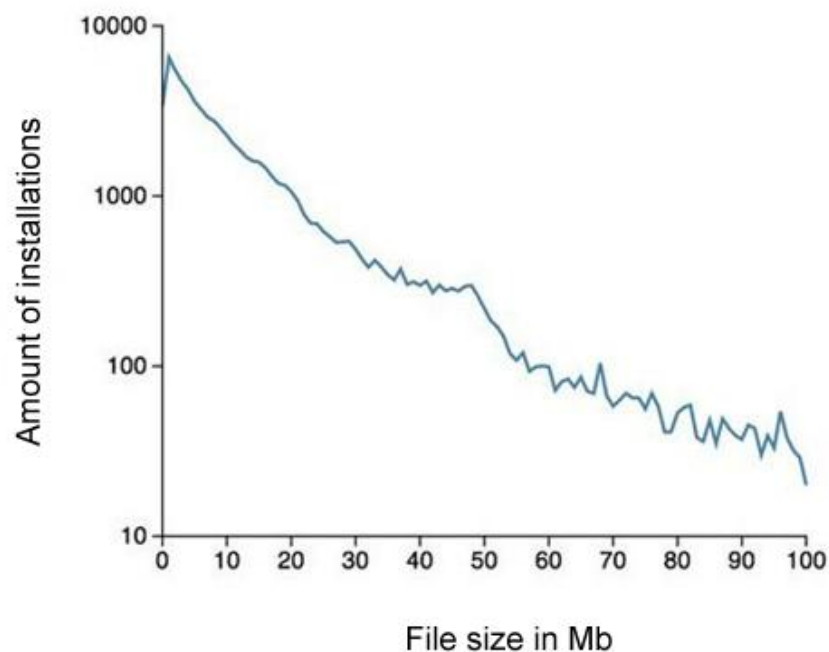


Figure 5.4 – Dependence on file size of application

According to Scott Logic, the less the application file “weighs”, the more often users install it. If you want users to download your program, the file size should not exceed 50 MB. According to statistics, owners of smartphones and tablets try to download more “heavy” applications only if they have access to Wi-Fi.

### Step 13: Buy promotional messages from famous people

You can personally contact the owners of the promoted accounts, offer them to test the application and share their impressions with subscribers. Participants in

domestic and foreign Internet marketing forums argue that the effectiveness of this tactic increases dramatically if the advertiser offers account holders financial rewards.

If you do not want to bargain with celebrities, buy posts and tweets on exchanges at a fixed price. This method has two serious drawbacks. First, popular people rarely sell messages in their accounts openly. Secondly, many promoted accounts that are highly rated in advertising services are bots.

You can buy messages from famous people and upgraded bots on the following resources: Twite, Adly (Here you can buy tweets from Kim Kardashian and Snoop Dogg), MyLikes, Sponsored Tweets, PaidPerTweet.

#### Step 14: use push notifications

This tool will help you stimulate user activity: program launches, frequency and duration of use. You can also influence the target actions of customers. For example, using a push notification, you can report a promotion or sale. Finally, notifications can be used to advertise your other apps.

#### Step 15: Analyze Application Performance

In the context of promoting the application, you should monitor the effectiveness of marketing activities. Use the following metrics:

- Total number and dynamics of installations;
- Sources of installations;
- The cost of attracting the user;
- The ratio of ad impressions to settings;
- The ratio of advertising costs to revenue generated by the application.

You can evaluate the effectiveness of the application using the following indicators:

- Number of starts;
- Duration of sessions;
- The average frequency of use of the application;
- Data on the targeted actions of users in the application.

Using analytics systems, you can determine the most effective promotion channels, as well as assess the needs of users and choose directions for the development of the program.

Do not forget about the need to constantly improve the product. You should be interested in any factors affecting the attractiveness of the application to users. Improve design and usability, expand functionality, stimulate user activity, try different business models.



## CONCLUSIONS

The work consists of five parts, which describe in detail the process from idea to development and testing.

In the first chapter an analysis of the technical justification of the purpose of development and an overview of e-book and audiobook formats. Existing analogues were considered, and as a result, due to the fact that they did not completely satisfy the conditions of the task, conclusions were drawn about the development's worthiness.

In the second chapter is described the development of the method of synchronization of text and audio modes of reading the electronic book. An analysis of speech recognition systems was also carried out, as a result of which the choice was stopped on CMU Sphinx.

In the third part were developed system architecture and software according to it, namely: audio to text aligning system, a module for displaying audio files and text based on a language recognition system; module for reproducing text with time stamps aligned with the audio file; a mobile application that uses the above developments. In the future, the application will have a larger library, including many sections: scientific and fiction, children's, science fiction, medical, general education, etc. The ability to leave notes, bookmarks, save quotes, highlight text and fix notes will be introduced. The application will be developed and edited in accordance with criticism and the wishes of users to improve the quality of an existing project.

In the fourth chapter, quality assessment of the developed system was carried out. WER for identical audio and texts did not exceed 11.2%. This is an excellent result, considering that for simple recognition using the same system, WER is 22.7%.

The fifth chapter examined the possible market for the finished application and its customers. Particular attention is paid to the release of the application on Google Play and App Store. The situation in the advertising sphere was also analyzed and steps for promoting the application were suggested.

## REFERENCES

1. Формати електронних книжок | Віртуальне видавництво БукРі – URL: [http://bookri.com.ua/bookri-school/vydy\\_e-knyg.html](http://bookri.com.ua/bookri-school/vydy_e-knyg.html)
2. Форматы электронных книг: советы авторам и читателям – URL: <https://www.triumph.ru/formaty-elektronnyh-knig.html>
3. PDF REFERENCE AND ADOBE EXTENSIONS TO THE PDF SPECIFICATION – URL: [https://www.adobe.com/devnet/pdf/pdf\\_reference.html](https://www.adobe.com/devnet/pdf/pdf_reference.html)
4. How to Crack Secured PDF Files and eBooks in Four Easy Steps <https://gilsmethod.com/crack-secured-pdf-files-ebooks-easy-steps>
5. EPUB 3 Community Group – URL: <https://www.w3.org/publishing/groups/epub3-cg/>
6. Что такое FictionBook - FictionBook – URL: [http://www.fictionbook.org/index.php/%D0%A7%D1%82%D0%BE\\_%D1%82%D0%B0%D0%BA%D0%BE%D0%B5\\_FictionBook](http://www.fictionbook.org/index.php/%D0%A7%D1%82%D0%BE_%D1%82%D0%B0%D0%BA%D0%BE%D0%B5_FictionBook)
7. MobileRead Wiki - MOBI – URL: <https://wiki.mobileread.com/wiki/MOBI>
8. MobileRead Wiki - AZW – URL: <https://wiki.mobileread.com/wiki/AZW>
9. Новый формат Kindle 8 с поддержкой HTML5 / Хабр – URL: <https://habr.com/ru/post/130975/>
10. Аудиокнига, что это такое – URL: <https://eksmo.ru/slovar/audiokniga/>
11. Audiobook - Wikipedia – URL: <https://en.wikipedia.org/wiki/Audiobook>
12. Аудиокнига, история, технология создания – URL: <http://www.cultin.ru/books-audiokniga>
13. Аудиокниги, как средство изучения английского языка – URL: <https://www.native-english.ru/articles/audiobooks>
14. Подробнее об аудио форматах – URL: [https://online-audio-converter.com/ru/help/audio\\_formats](https://online-audio-converter.com/ru/help/audio_formats)
15. WAV & BWF Metadata Guide – URL: <http://wavmetadata.blogspot.com/>
16. RFC 3003 - The audio/mpeg Media Type – URL: <https://tools.ietf.org/html/rfc3003>

17. File Name Extensions - Win32 – URL: <https://docs.microsoft.com/ru-ru/windows/win32/wmp/file-name-extensions?redirectedfrom=MSDN>
18. Xiph.org: Ogg – URL: <https://www.xiph.org/ogg/>
19. ISO/IEC 13818-7:1997 Information technology — Generic coding of moving pictures and associated audio information — Part 7: Advanced Audio Coding (AAC) – URL: <https://www.iso.org/standard/25040.html>
20. FLAC - changelog – URL: [https://xiph.org/flac/changelog.html#flac\\_1\\_0](https://xiph.org/flac/changelog.html#flac_1_0)
21. AppStore: VoiceDreamReader– URL: <https://apps.apple.com/ru/app/voice-dream-reader/id496177674>
22. Acapela Group – URL: <https://www.acapela-group.com/>
23. NeoSpeech: Text-to-Speech (TTS) Software – URL: <https://neospeech.com>
24. GitHub - westonruter/html5-audio-read-along: HTML5 Audio Read-Along Sector – URL: <https://github.com/westonruter/html5-audio-read-along>
25. ESV API | ESV.org – URL: <https://api.esv.org/>
26. What is Whispersync for Voice? – URL: [https://audible.custhelp.com/app/answers/detail/a\\_id/7371/~/~what-is-whispersync-for-voice%3F](https://audible.custhelp.com/app/answers/detail/a_id/7371/~/~what-is-whispersync-for-voice%3F)
27. How to Use Amazon’s Whispersync for Voice and Immersion Reading Features – URL: <https://blog.the-ebook-reader.com/2016/04/02/how-to-use-amazons-whispersync-for-voice-and-immersion-reading-features/>
28. Amazon Transcribe – URL: <https://aws.amazon.com/ru/transcribe/>
29. Audio Pronunciation in Google Search – URL: <http://googlesystem.blogspot.com/2011/08/audio-pronunciation-in-google-search.html>
30. Поиск оптимальной системы аудио распознавания речи с закрытым исходным кодом, но имеющими открытые API– URL: <https://habrahabr.ru/post/231629/>
31. HTK Speech Recognition Toolkit – URL: <http://htk.eng.cam.ac.uk/>
32. CMUSphinx Documentation – CMUSphinx Open Source Speech Recognition – URL: <https://cmusphinx.github.io/wiki/>

33. CMU Sphinx download | SourceForge – URL: <https://sourceforge.net/projects/cmusphinx/>
34. Yandex SpeechKit | Яндекс.Облако – URL: <https://cloud.yandex.ru/services/speechkit>
35. «Яндекс» готовит на экспорт речевые технологии: для автопрома, телекомов и медицины – URL: <https://roem.ru/06-09-2016/232358/yandeks-golos-export/>
36. Алиса – голосовой помощник от компании Яндекс – URL: <https://yandex.ru/alice/index>
37. Россияне смогут переводить деньги с помощью голосовых команд – URL: <https://iz.ru/632953/anastasiiia-alekseevskikh/rossiiane-smogut-perevodit-dengi-s-pomoshchiu-golosovykh-komand>
38. Коллтрекинг-сервис Calltouch запустил услугу распознавания качества звонков при помощи технологий «Яндекса» – URL: <https://vc.ru/flood/16530-calltouch-predict>
39. Выбор оптимальной системы распознавания речи в задаче анализа новостной информации – URL: <http://web.snauka.ru/issues/2016/05/67547>
40. Microsoft Speech – Разработка приложений для Windows – URL: <https://developer.microsoft.com/ru-ru/windows/speech>
41. Зачем голосовому Поиску Google нейронные сети? / Блог компании Google / Хабр – URL: <https://habr.com/ru/company/google/blog/269747/>
42. Распознавание речи от Яндекса. Под капотом у Yandex.SpeechKit? / Блог компании Яндекс / Хабр – URL: <https://habr.com/ru/company/yandex/blog/198556/>
43. Bohouta Gamal, Kępuska Veton Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx) // Int. Journal of Engineering Research and Application, 2017/03/01
44. Беленко М. В. Сравнительный анализ систем распознавания речи с открытым кодом / М. В. Беленко, П. В. Балакшин // Международный научно-исследовательский журнал. — 2017. — № 04 (58) Часть 4

45. Даниил Бойко Системы распознавания речи с открытым исходным кодом // LVEE Winter 2018 – Минск, Belarus
46. GitHub - cmusphinx/sphinx4: Pure Java speech recognition library – URL: <https://github.com/cmusphinx/sphinx4>
47. Eclipse - Wikipedia – URL: <https://uk.wikipedia.org/wiki/Eclipse>
48. Maven - Welcome to Apache Maven – URL: <https://maven.apache.org/>
49. Visual Studio Code - Code editing. Redefined – URL: <https://code.visualstudio.com/>
50. Angular – URL: <https://angular.io/>
51. What is Ionic Framework? – URL: <https://ionicframework.com/docs/intro>
52. Введение - Apache Cordova – URL: <https://cordova.apache.org/docs/ru/latest/guide/overview/>
53. Speech Recognition Python – Converting Speech to Text – URL: <https://www.simplifiedpython.net/speech-recognition-python/>
54. How to Calculate Word Error Rate (WER) – URL: <https://www.rev.ai/blog/how-to-calculate-word-error-rate/>
55. Challenges in Measuring Automatic Transcription Accuracy – URL: <https://medium.com/descript/challenges-in-measuring-automatic-transcription-accuracy-f322bf5994f>
56. Distributed computing network for Machine Learning – URL: <https://anryze.com/>
57. Мазуренко И. Л. Компьютерные системы распознавания речи // Интеллектуальные системы. Т. 3. Вып. 1–2. – М., 1998. – С. 117–134
58. Juang B. H. Speech Recognition in Adverse Environments // Computer Speech and Language. Vol.5. N. Y. 2002. – P. 275–294
59. Speech Recognition Grammar Specification Version 1.0 – URL: <http://www.w3.org/TR/speechsynthesis>
60. Speech Synthesis Markup Language (SSML) Version 1.0 – URL: <http://www.w3.org/TR/speechgrammar/>

61. Фролов А. В., Фролов Г. В. Синтез и распознавание речи // Современные решения, 2003

62. Холоденко А. Б. О построении статистических языковых моделей для систем распознавания русской речи // Интеллектуальные системы. Т. 6. Вып. 1–4. – М., 2002. – С. 381–394.

63. Поиск оптимальной аудио-системы распознавания речи на базе открытого исходного кода – URL:<https://habr.com/post/230133/>

64. Russian-language speech recognition based on DeepSearch – URL: <https://indico.jinr.ru/event/447/session/12/material/slides/0?contribId=302>

65. Learn how to Build your First Speech-to-Text Model in Python – URL:<https://www.analyticsvidhya.com/blog/2019/07/learn-build-first-speech-to-text-model-python/>

66. Рынок мобильных устройств и приложений: книжный ландшафт – URL: <http://www.unkniga.ru/bookrinok/knigniy-rinok/1816-rynok-mob-ustroystv-knizhniy-landshaft.html>

67. Менше половини українців за останній рік прочитали хоча б одну книгу – URL: [https://dt.ua/UKRAINE/menshe-polovini-ukrayinciv-za-ostanniy-rik-prochitali-hocha-b-odnu-knigu-267324\\_.html](https://dt.ua/UKRAINE/menshe-polovini-ukrayinciv-za-ostanniy-rik-prochitali-hocha-b-odnu-knigu-267324_.html)

68. Что читают россияне в 2019 году – URL: <https://animedia-company.cz/what-do-the-russians-read-in-2019/>

69. Mobile Operating System Market Share Ukraine | StatCounter Global – URL: <https://gs.statcounter.com/os-market-share/mobile/ukraine>

70. Как iOS и Android разделили мобильный рынок - AppTracktor – URL: <https://apptractor.ru/measure/app-store-analytics/kak-ios-i-android-razdelili-mobilnyiy-ryinok.html>

71. Инструкция по публикации Android-приложения в GooglePlay – URL: <https://habr.com/en/company/livotyping/blog/326874/>

72. Инструкция по публикации iOS-приложения в AppStore – URL: <https://habr.com/en/company/livotyping/blog/324958/>

73. Кейс из России: Как музыкальный сервис Hitster протестировал на пользователях своё Android-приложение до запуска – URL: <https://vc.ru/flood/7888-hitster-case>

74. 40% OF TABLET AND SMARTPHONE OWNERS USE THEM WHILE WATCHING TV – URL: <https://www.nielsen.com/us/en/insights/news/2011/40-of-tablet-and-smartphone-owners-use-them-while-watching-tv.html>

75. A Statistical Analysis of the Apple App Store – URL: <https://blog.scottlogic.com/2014/03/20/app-store-analysis.html>

## ATTACHMENTS



## ATTACHMENT A

(Required)

Ministry of Education and Science of Ukraine  
Vinnytsia National Technical University  
Faculty of Computer Systems and Automation

Approved

Head of the AIIS Department, Dr. of Sci., Prof.

\_\_\_\_\_ Roman N. Kvetny

“ \_\_\_\_ ” \_\_\_\_\_ 2019 year

Approved

Head of the course work, Ph. D, associate professor

\_\_\_\_\_ Tetyana V. Gryshchuk

“ \_\_\_\_ ” \_\_\_\_\_ 2019 year

DEVELOPED BY

Student of the 3ACIT-18m group

Andrii M. Henri

“ \_\_\_\_ ” \_\_\_\_\_ 2019 year

## TECHNICAL TASK

for master's qualification work

on the topic

«DEVELOPMENT OF AN INTELLIGENT SYSTEM TO SUPPORT THE  
SWITCHING OF MODES OF READING THE ELECTRONIC BOOKS WITH  
AUDIO DATA»

Designated product name: Intelligent System to Support the Switching of  
Modes of Reading the Electronic Books with Audio Data.

Product use area: Work results can be used by regular users in everyday life. The achievements in speech recognition can be used in other industries.

Reason for product development:

1. Curriculum of specialty 151 – Automation and computer-integrated technologies.
2. Individual assignment for master's qualification work

Main functions of the program:

1. Recognise text from audiobook
2. Align book's audiofile to original text
3. Output data with timestamps.
4. Use output file in interface to allow user switch between modes

Requirements for a software product:

1. Operating system – WINDOWS/Linux/MacOS.
2. Programming environment – arbitrary.
3. Required memory – 10 GB of disk space
4. Software should be tested and made available on its suitability for use.

All software and accompanying technical documentation must comply with the following GOSTs:

GOST 19.701-90

ISO 5807-85 - GOST for development of software documents, schemes of program algorithms, data and systems.

GOST 19.781-74 - requirements for software development.

GOST 19.401-78 – The text of the program. Requirements for content and design.

GOST 19.106-78 – Requirements for software documentation.

GOST 7.1.-84 and DSTU 3008-95 – development of technical documentation.

**Development stages:**

1. Classification of the type of task

2. Setting the task
3. The choice of methods for solving the problem
4. Software implementation
5. Testing
6. Conclusions

**The order of control and acceptance of master's qualification work:**

Obtaining a task for qualification work – «\_\_» \_\_\_\_\_ 2019 year.

The term of the term paper is checked – till «\_\_» \_\_\_\_\_ 2019 year.

Term of protection of qualification work – till «\_\_» \_\_\_\_\_ 2019 year.

## ATTACHMENT B

(Required)

APPROVED  
 Head of the AIIT department,  
 Dr. of Sci., Prof. Roman N. Kvetny  
 \_\_\_\_\_  
 “ \_\_\_\_\_ ” \_\_\_\_\_ 2019 year

LIST  
 OF GRAPHIC MATERIALS

for the protection of master’s qualification work  
 on the topic

«DEVELOPMENT OF AN INTELLIGENT SYSTEM TO SUPPORT THE  
 SWITCHING OF MODES OF READING THE ELECTRONIC BOOKS WITH  
 AUDIO DATA»

1. The concept of matching audio and text
2. Aligning algorithm
3. UML-diagram of use cases
4. UML-diagram of activity
5. UML-diagram of classes
6. Screen view of the mobile application library
7. Screen view of the mobile application reading mode

Developed by Andrii M. Henri

\_\_\_\_\_  
 (signature)

\_\_\_\_\_  
 (date)

Checked by Tetyana V. Gryshchuk

\_\_\_\_\_  
 (signature)

\_\_\_\_\_  
 (date)

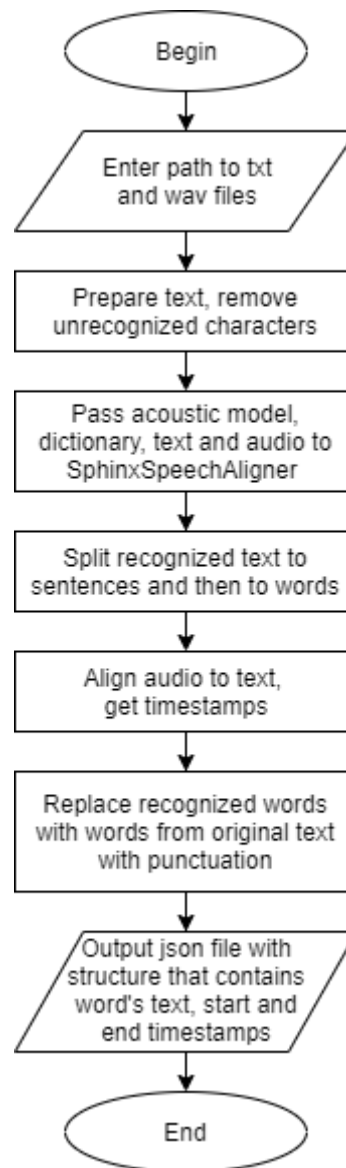
Reviewed by Yevhen A. Palamarchuk

\_\_\_\_\_  
 (signature)

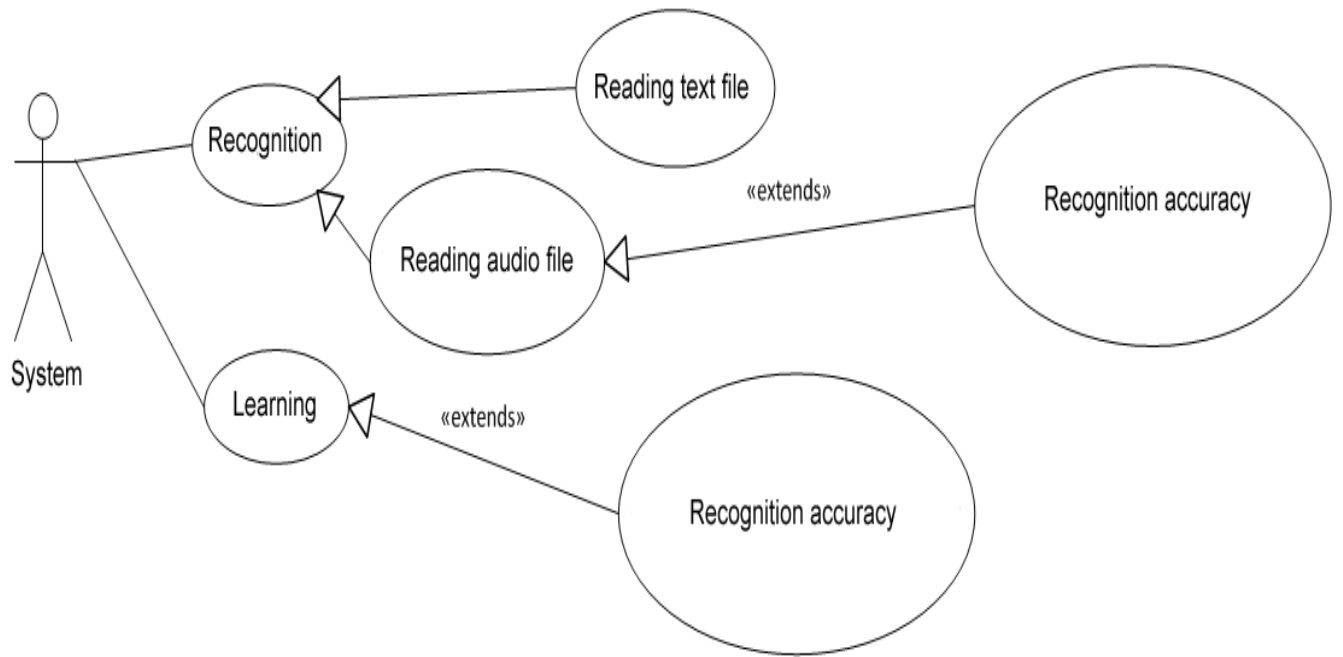
\_\_\_\_\_  
 (date)



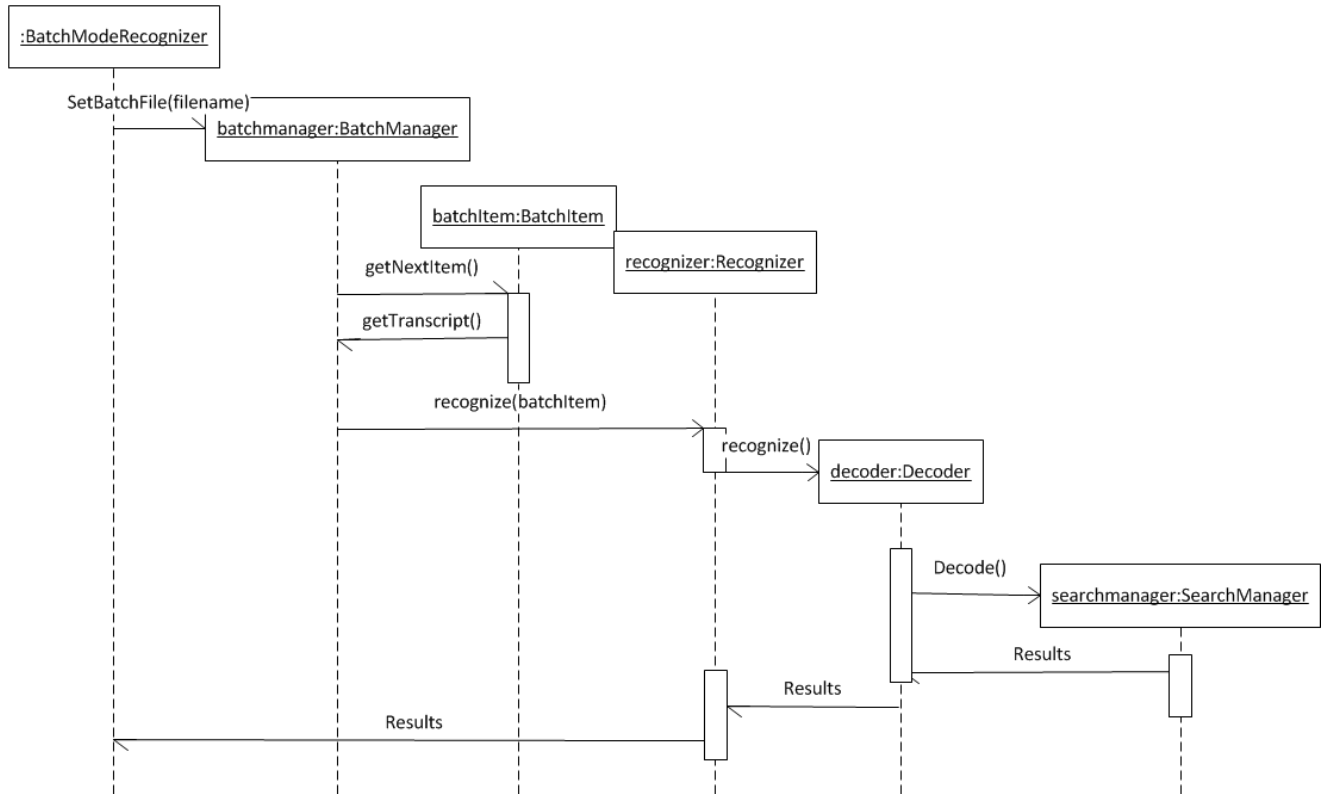
The concept of matching audio and text



Aligning algorithm

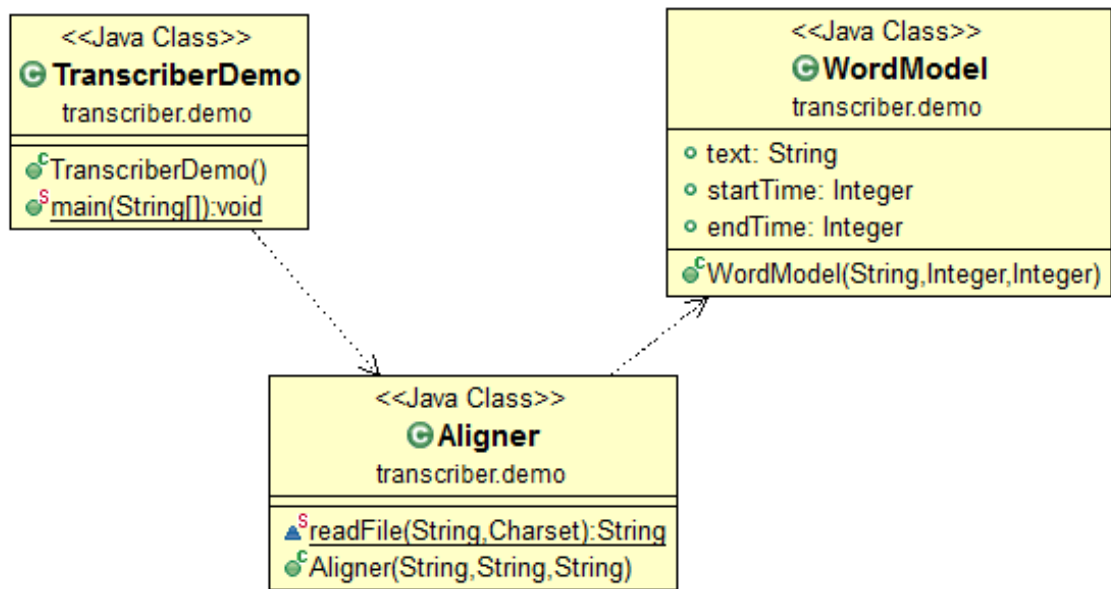


UML-diagram of use cases



UML-diagram of activity

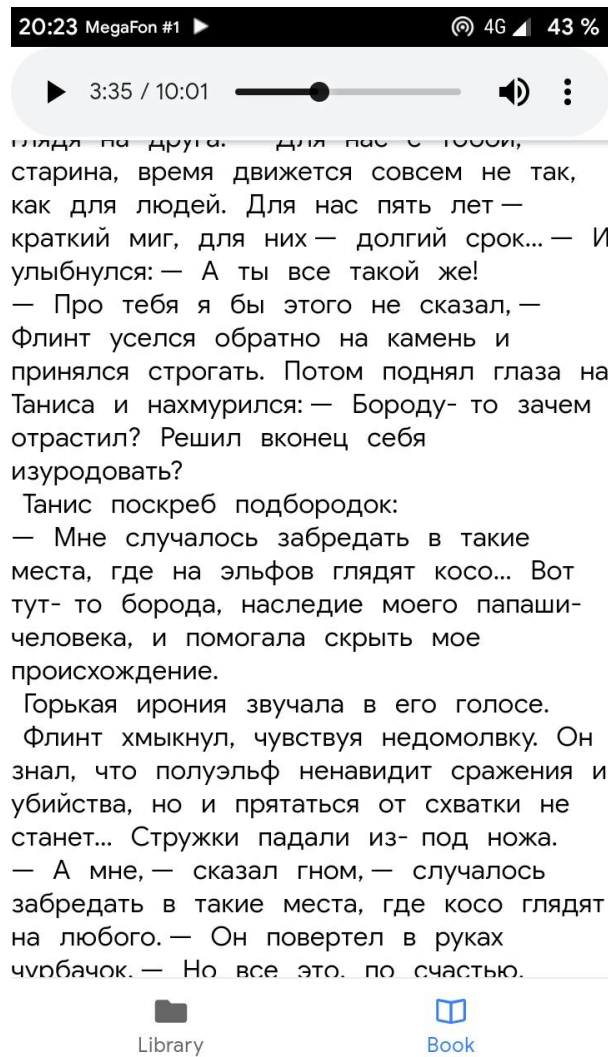




UML-diagram of classes



Screen view of the mobile application library



Screen view of the mobile application library

## ATTACHMENTC

## Code listing of the mobile application

## Listing of book.page.ts:

```

import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { ActivatedRoute, Router } from '@angular/router';
import { MusicControls } from '@ionic-native/music-controls/ngx';

@Component({
  selector: 'app-book',
  templateUrl: 'book.page.html',
  styleUrls: ['book.page.scss']
})

export class BookPage {
  book: any;
  oldBookId: any;
  data: any;
  constructor(public http: HttpClient, private route: ActivatedRoute, private router: Router, private musicControls: MusicControls) {
    this.route.queryParams.subscribe(params => {
      if (this.router.getCurrentNavigation().extras.state) {
        this.book = this.router.getCurrentNavigation().extras.state.book;
      }
    });
  }
  player: HTMLAudioElement;
  arrTimeStampsStart = [];
  closestData: any;

  ngOnInit() {
    this.player = <HTMLAudioElement>document.getElementById("player");
    this.player.addEventListener("timeupdate", () => {
      this.selectWord();
    });
  }

  ionViewWillEnter() {
    console.log(this.book);
    if (this.book.id == undefined) {
      this.router.navigate(['tabs/tab1']);
    } else {
      this.getBookData();
    }
  }

  getBookData() {
    if (this.book.bookId != this.oldBookId) {
      this.player.pause();
      this.player.src = this.book.audioUrl;
      this.initMusicControls();
    }
  }

  this.http.get(this.book.textUrl).subscribe(res => {
    this.data = res;
    this.data.forEach(each => {
      if (each.startTime != undefined)
        this.arrTimeStampsStart.push(each.startTime);
    });
  }, (err) => {
    alert('Failed loading json data');
  });
  this.oldBookId = this.book.id;
}

wordClick(elem) {
  this.player.currentTime = elem.dataset.starttime / 1000;
}

selectWord() {
  this.closestData = this.closestWord(this.player.currentTime * 1000, this.arrTimeStampsStart);
  // $('[data-starttime='${closestData}']')[0].scrollIntoView(); //мешает
}

closestWord(num, arr) {
  var curr = arr[0];
  var diff = Math.abs(num - curr);
  for (var val = 0; val < arr.length; val++) {
    var newdiff = Math.abs(num - arr[val]);
    if (newdiff < diff) {
      diff = newdiff;
      curr = arr[val];
    }
  }
  return curr;
}

initMusicControls() {
  this.musicControls.create({
    track: this.book.title, // optional, default : ''
    artist: this.book.authors, // optional, default : ''
    cover: this.book.imageUrl, // optional, default : nothing
    // cover can be a local path (use fullpath 'file:///storage/emulated/...', or only 'my_image.jpg' if my_image.jpg is in the www folder of your app)
    // or a remote url ('http://...', 'https://...', 'ftp://...')
    isPlaying: true, // optional, default : true
    dismissable: true, // optional, default : false
    // hide previous/next/close buttons:
    hasPrev: false, // show previous button, optional, default: true
    hasNext: false, // show next button, optional, default: true
    hasClose: true, // show close button, optional, default: false
    // iOS only, optional
  });
}

```

```

        album: 'Absolution',          // optional,
default: ''
        duration: 60, // optional, default: 0
        elapsed: 10, // optional, default: 0
        hasSkipForward: true, // show skip
forward button, optional, default: false
        hasSkipBackward: true, // show skip
backward button, optional, default: false
        skipForwardInterval: 15, // display
number for skip forward, optional, default: 0
        skipBackwardInterval: 15, // display
number for skip backward, optional, default: 0
        hasScrubbing: false, // enable scrubbing
from control center and lockscreen progress
bar, optional

        // Android only, optional
        // text displayed in the status bar when
the notification (and the ticker) are updated,
optional
        ticker: 'Now playing "Time is Running
Out"',
        // All icons default to their built-in
android equivalents
        playIcon: 'media_play',
        pauseIcon: 'media_pause',
        prevIcon: 'media_prev',
        nextIcon: 'media_next',
        closeIcon: 'media_close',
        notificationIcon: 'notification'
    });

this.musicControls.subscribe().subscribe(action => {

    function events(action) {
        const message =
JSON.parse(action).message;
        switch (message) {
            case 'music-controls-next':
                // Do something
                break;
            case 'music-controls-previous':
                // Do something
                break;
            case 'music-controls-pause':
                this.player.pause();
                break;
            case 'music-controls-play':
                this.player.play();
                break;
            case 'music-controls-destroy':
                // Do something
                break;

                // External controls (iOS only)
            case 'music-controls-toggle-play-
pause':
                // Do something
                break;
            case 'music-controls-seek-to':
                const seekToInSeconds =
JSON.parse(action).position;
                this.musicControls.updateElapsed({
                    elapsed: seekToInSeconds,
                    isPlaying: true
                });
                break;
            case 'music-controls-skip-forward':
                break;
            case 'music-controls-skip-backward':
                break;
            // Headset events (Android only)
            // All media button events are
listed below
            case 'music-controls-media-button':
                break;
            case 'music-controls-headset-
unplugged':
                break;
            case 'music-controls-headset-
plugged':
                break;
            default:
                break;
        }
    }

    this.musicControls.listen(); // activates
the observable above

    this.musicControls.updateIsPlaying(true);
}
}

```

### Listing of book.page.html:

```

<ion-header>
<ion-toolbar>
<ion-title>
    Bookname
</ion-title>
</ion-toolbar>
</ion-header>

<ion-content>
<audio id="player" controls>
<source [src]=""" type="audio/mpeg"
(ontimeupdate)="selectWord()" >
    Your browser does not support the audio
element.
</audio>
<div class='fulltext'>
<ng-container *ngFor="let word of data" >

<span *ngIf="word.startTime !== undefined"
class='word-wrapper'
    (click)="wordClick($event.target)"
    [attr.data-starttime]=word.startTime
    [attr.data-endtime]=word.endTime
    [style.background-
color]="closestData == word.startTime ? 'red'
: 'white'">
        {{word.text}}
</span>
<span *ngIf="word.startTime == undefined"
class='word-wrapper'
    {{word.text}}
</span>
</ng-container>
</div>
</ion-content>

```

### Listing of library.page.ts:

```

import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { NavigationExtras, Router } from '@angular/router';

@Component({
  selector: 'app-library',
  templateUrl: 'library.page.html',
  styleUrls: ['library.page.scss']
})
export class LibraryPage {
  books: any;

  constructor(public http: HttpClient, private
router: Router) { }

  ngOnInit() {
    this.getBooksList();
  }

  getBooksList() {
    this.http.get('assets/books.json').subscribe(r
es => {
      this.books = res;
    },
    (err) => {
      alert('Failed loading a list of
books');
    });
  }

  openBook(book) {
    let navigationExtras: NavigationExtras = {
      state: {
        book: book
      }
    };
    this.router.navigate(['tabs/book'],
navigationExtras);
  }
}

```

### Listing of library.page.html:

```

<ion-header>
<ion-toolbar>
<ion-title>
  Library
</ion-title>
</ion-toolbar>
</ion-header>

<ion-content>
<div class='books-container'>
<div *ngFor="let book of books" class='book'
(click)="openBook(book)">
<img class='artwork' src={{book.imageUrl}}>
<div class='title'>{{book.title}}</div>
<div class='authors'>{{book.authors}}</div>
</div>
</div>
</ion-content>

```

# ATTACHMENTD

## Code listing of the aligning program

### Listing of Aligner.java:

```
package transcriber.demo;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URL;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

import javax.sound.sampled.AudioFileFormat;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.UnsupportedAudioFileException;

import com.google.gson.Gson;

import edu.cmu.sphinx.alignment.LongTextAligner;
import edu.cmu.sphinx.api.SpeechAligner;
import edu.cmu.sphinx.result.WordResult;
import edu.cmu.sphinx.util.TimeFrame;

public class Aligner {
    static String
    readFile(String path, Charset encoding)
        throws IOException
    {
        byte[] encoded =
Files.readAllBytes(Paths.get(path));
        return new
String(encoded, encoding);
    }

    public Aligner(String
audioPath, String textPath, String jsonPath)
throws Exception {
        ArrayList<WordModel>
wordsArray = new ArrayList<WordModel>();

        System.out.println("Timestamping");

        URL audioUrl;
        String transcript,
originalText;

        String
acousticModelPath =
"file:src/main/resources/model";
        String dictionaryPath =
"file:src/main/resources/model/ru.dic";
        audioUrl =
Paths.get(audioPath).toUri().toURL();
        originalText =
readFile(textPath, StandardCharsets.UTF_8);
        StringBuilder original
= new StringBuilder(originalText);

        transcript =
originalText;
        transcript =
transcript.replaceAll("[^a-zA-ЯёЁаа-zA]",
"");
        transcript =
transcript.replaceAll("\\p{Cc}", "");
        transcript =
transcript.trim().replaceAll(" +", " ");
        SpeechAligner aligner =
new SpeechAligner(acousticModelPath,
dictionaryPath, null);

        List<WordResult>
results = aligner.align(audioUrl, transcript);
        List<String>
stringResults = new ArrayList<String>();

        for (WordResult wr :
results) {
            stringResults.add(wr.getWord().getSpelling());
        }
        LongTextAligner
textAligner =
new LongTextAligner(stringResults, 2);
        List<String> sentences
= aligner.getTokenizer().expand(transcript);
        List<String> words =
aligner.sentenceToWords(sentences);

        int[] aid =
textAligner.align(words);

        int lastId = -1;
        for (int i = 0; i <
aid.length; ++i) {
            if (aid[i] == -1) {
                //wordsArray.add(new
WordModel(words.get(i), null, null));
                System.out.format("- %s\n", words.get(i));
            } else {
                if (aid[i] -
lastId > 1) {
                    for
(WordResult result : results.subList(lastId +
1,
aid[i])) {
                        System.out.format("+
%-25s [%s]\n",
result.getWord()
.getSpelling(), result.getTimeFrame());
                    }
                }
                wordsArray.add(new
WordModel(results.get(aid[i]).getWord().getSpe
lling(),
(int)results.get(aid[i]).getTimeFrame().getSta
rt(),
(int)results.get(aid[i]).getTimeFrame().getEnd
()));
            }
        }
    }
}
```

```

System.out.format("          %-25s      [%s]\n",
results.get(aid[i])

.getWord().getSpelling(), results.get(aid[i])

.getTimeFrame());

                                lastId      =
aid[i];

                                }
                                }

                                if (lastId >= 0 &&
results.size() - lastId > 1) {
                                for (WordResult
result : results.subList(lastId + 1,
results.size())) {

System.out.format("+          %-25s      [%s]\n",
result.getWord()

.getSpelling(), result.getTimeFrame());
                                }
                                }
                                int startIndex = 0;
                                int endIndex = 0;
                                String searchedWord;
                                ArrayList<WordModel>
finalArray = new ArrayList<WordModel>();
                                int oldIndex = 0;
                                for (int i=1;
i<wordsArray.size(); i++) {
                                searchedWord      =
wordsArray.get(i).text;
                                startIndex      =
original.toString().toLowerCase().indexOf(sear
chedWord.toLowerCase(), endIndex);
                                if (startIndex !=-1) {
                                endIndex      =
startIndex + searchedWord.length();
                                //adding      text
for previous iteration
                                //можно
проверить, и если нет startTime, то не
добавлять,чтоб при любом клике все работало
                                if
(wordsArray.get(i-1).startTime != null) {

                                finalArray.add(new
WordModel(original.substring(oldIndex,
startIndex), wordsArray.get(i-1).startTime,
wordsArray.get(i-1).endTime));
                                oldIndex
= startIndex;ra
                                }
                                }
                                // adding last record
into array
                                finalArray.add(new
WordModel(original.substring(oldIndex),
wordsArray.get(wordsArray.size()-1).startTime,
wordsArray.get(wordsArray.size()-1).endTime));

                                /
                                for (int i=0;
i<finalArray.size(); i++) {
                                if
(finalArray.get(i).text.length() > 50) {

                                finalArray.get(i).text      =
wordsArray.get(i).text;
                                }
                                }
                                //writing to json
Gson gson = new Gson();
                                //String      json      =
gson.toJson(wordsArray);
                                String      json      =
gson.toJson(finalArray);
                                try (PrintWriter out =
new PrintWriter(jsonPath)) {
                                out.println(json);
                                }
}

```



## ATTACHMENTE

## Code listing dependencies in pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>transcriber</groupId>
  <artifactId>aligner</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>aligner</name>
  <url>http://maven.apache.org</url>
  <properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <java.version>1.8</java.version>
  </properties>
  <repositories>
  <repository>
  <id>snapshots-repo</id>
  <url>https://oss.sonatype.org/content/repositories/snapshots</url>
  <releases>
  <enabled>>false</enabled>
  </releases>
  <snapshots>
  <enabled>>true</enabled>
  </snapshots>
  </repository>
  </repositories>
  <dependencies>
  <dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>edu.cmu.sphinx</groupId>
    <artifactId>sphinx4-core</artifactId>
    <version>5prealpha-SNAPSHOT</version>
  </dependency>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
  </dependency>
  <dependency>
    <groupId>ws.schild</groupId>
    <artifactId>jave-all-deps</artifactId>
    <version>2.4.5</version>
  </dependency>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-io</artifactId>
    <version>1.3.2</version>
  </dependency>
  </dependencies>
</project>

```