Nyx - An Educational Assistant for the Visually Impaired

A PROJECT REPORT

Submitted by

HARIHARAN S [211420104089]

GOPINATH NV [211420104083]

GOWTHAM J [211420104085]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonmous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"Nyx - An Educational Assistance For theVisually Impaired"** is the Bonafide work of "HARIHARAN S (211420104089), GOWTHAM J (211420104085), GOPINATH NV(211420104083)" who carried out the project work under my supervision.

Signature of the HOD with date

Dr L JABASHEELA M.E., Ph.D.,
Professor and Head,

Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai - 123

Signature of the Supervisor with date

Dr N PUGHAZENDI M.E., Ph.D.,
Professor

Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai- 123

Submitted for the Project Viva – Voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We GOPINATH NV (211420104083), GOWTHAM J (211420104085), HARIHARAN S (211420104089) hereby declare that this project report titled "**NYX – AN EDUCATIONAL ASSISTANT FOR VISUALLY IMPARIED**", under the guidance of **Dr.N.PUGHAZENDI, M.E, PhD** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**.,for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording usthe essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA,M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator** and **Project Guide Dr N PUGHAZENDI M.E., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

GOWTHAM J (211420104085)
GOPINATH N V(211420104083)
HARIHARAN S(211420104089)

ABSTRACT

All individuals require quality education to develop and advance in this knowledge driven world. The current educational curriculum is oriented towards the utilization of eyesight. Individuals who are blind or visually impaired hence experience a number of challenges when acquiring education. Visually impaired people often find it troublesome to acquire good reading materials in accessible formats. Lack of Braille knowledge and the inadequate supply of writing and reading equipment such as Braille machines, laptops and computers further make the learning process difficult for them. There are existing applications that allow users to hear words written on digital documents but not from paper documents. The proposed systemis an android application with a voice-user interface for assisting the blind or visually impaired people to study, listen to text, get basic queries answered and also functions as a personal assistant. The application provides various functionalities such as scanning hard copies and converting them to audio books, converting e-books to audio books, answering basic queries through web searching, bookmarking, creating notes and lists. This research study intends to provide a solution for the challenges in acquiring education faced by the blind or visually impaired people. The application provides various functionalities such as scanning hard copies and converting them to audio books, converting ebooks to audio books, answering basic queries through web searching, bookmarking, creating notes and lists. This research study intends to provide a solution for the challenges in acquiring education faced by the blind or visually impaired people. laptops and computers further make the learning process difficult for them. There are existing applications that allow users to hear words written on digital documents but not from paper documents. The proposed systemis an android application with a voice-user interface for assisting the blind or visually impaired people to study, listen to text, get basic queries answered and also functions as a personal assistant.

TABLE OF CONTENTS

CHAPTER No		TITLE	PAGE No
		ABSTRACT	V
		LIST OF FIGURES	VI
		LIST OF ABBREVATIONS	VII
1		**INTRODUCTION**	1
	1.1	Problem Definition	1
2		**LITERATURE REVIEW**	2
3		**THEORETICAL BACKGROUND**	4
	3.1	Implementation Environment	4
	3.2	System Architecture	5
	3.3	Proposed Methodology	7
	3.4	Module Design	8
4		**SYSTEM IMPLEMENTATION**	18
	4.1	Modules Explanation	18

5		**RESULTS AND DISCUSSION**	24
	5.1	Testing	24
	5.2	Results	28
	5.3	Discussion	29
6		**CONCLUSION AND FUTURE WORK**	32
		REFERENCES	34
		APPENDICES	36
	A.1	Source Code	36
	A.2	Screen Shots	48
	A.3	Plagiarism Report	60
	A.4	Paper Publication	61

LIST OF FIGURES

FIG NO	DESCRIPTION	PAGE NO
3.1	System Architecture	5
3.4.1.1	System Design	8
3.4.2.1	Sequence Diagram	9
3.4.3.1	Use Case Diagram	10
3.4.4.1	Activity Diagram	11
3.4.5.1	Collaboration Diagram	12
3.4.6.1	Class Diagram	13
3.4.7.1	Deployment Diagram	16
3.4.8.1	Component Diagram	17
3.4.9.1	Level 0 Data Flow Diagram	14
3.4.9.2	Level 1 Data Flow Diagram	14
3.4.9.3	Level 2 Data Flow Diagram	15
5.3.1	Number of blind people Vs Age	30

LIST OF ABBREVATION

ABBREVATION	DESCRIPTION
JDK	Java Development Toolkit
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

There is a growing awareness among parents, teachers, blind youth, and the adult blind community that the education which blind children are receiving is failing them. They are not receiving a quality education which can prepare them to compete in the demanding high tech economy and society of the 21st Century. They are not learning to use and trust the alternative techniques which blind persons must have if they are to be successful. They are not developing the positive. Every *blind* people irrespective of whether he/she is studying in special school or IE program must have access to a Braille kit. A Braille kit contains basic and All individuals require quality education to develop and advance in this knowledge driven world. The current educational curriculum is oriented towards the utilization of eyesight. Individuals who are blind or visually impaired hence experience a number of challenges when acquiring education. Visually impaired people often find it troublesome to acquire good reading materials in accessible formats. Lack of Braille knowledge and the inadequate supply of writing and reading equipment such as Braille machines, laptops and computers further make the learning process difficult for them. There are existing applications that allow users to hear words written on digital documents but not from paper documents. The proposed system is an android application with a voice-user interface for assisting the blind or visually impaired people to study, listen to text, get basic queries answered and also functions as a personal assistant. The application provides various functionalities such as scanning hard copies and converting them to audio books, converting e-books to audio books, web searching, bookmarking, creating notes and lists. This research study intends to provide a solution for the challenges in acquiring education faced by the blind or visually impaired people. There are existing applicationsthat allow users to hear words written on digital documents but not from paper documents. The proposed system is an android application with a voice-user interface for assisting the blind or visually impaired people to study, listen to text, get basic queries answered and also functions as a personal assistant

CHAPTER 2

LITERATURE REVIEW

Prasanna et al.,[1] "Design and Development of a MobileBased Application for Automated SMS System Using Voice Conversion Techfor Visually Impaired People."helps the blind in reading text messages received on the phone.The user is able to convert the text messages into voice as well as the received messages using a Hidden Markov Model. Accelerometer sensor is used to detect the movement of device such as shake, swing, rotation. It also enables to open/close the app automatically.

Gayatri.S et al.,[2] Department of Computer Science can send or receive text messages, make voice calls, make notes on mobile using voice and check the battery level. Here, messages received will be notified as voice itself. The technology adapted for recognizing voice includes speech synthesis technology.

Nishank M et al.,[3]"Voice Assistant for Visually Impaired People" aids the visually challenged in accessing the internet. In the proposed system, the user does not need to remember the complex braille commands. Instead, the user can simply voice out the proper commands and the software will execute this accordingly.

Abhijeet Mohanta et al.,[4] "Application for the Visually Impaired People With Voice Assistant" accepts text and speech commands from the user and performs the corresponding tasks. It is developed using Android Studio Platform. It helps the blind to easily access the electronic gadgets via TTS technology. They will be able to use electronic systems such as mobile phones, tablets etc.

Khandelwal et al.,[5] International Journal of Innovative Technology which focuses on helping the visually impaired people to live independently. This paper talks about how to use a smartphone to capture real-time input data so that the user can easily read menu cards in restaurants, hotel room number, or even find their lost belongings. This technology uses a control feedback mechanism which enables user to detect objects around his surroundings. DAVID – a

digital assistant which is discussed in paper

Gupta, Lavanya et al.,[6] "Approaches in Assistive Technology helps the visually impaired people in identifying text in real world time and provides a feedback which is in audio format. This system uses voice user interface technology such as speech recognition and speech synthesis to interact with the user. There are future works yet to implement.

Dubey, Nilesh et al.,[7] "IoT Based SelfNavigation Assistance for Visually Impaired." which is built using Raspberry Pi module, which is in turn connected to the camera in order to capture the input image. Later the image enhancement is done using image processing techniques. Tesseract OCR inside Raspberry Pi helps in searching the text in the enhanced image and later converts it to digital document. This is then analyzed with a semantic check.

Tejal Adep1 et al.,[8] "Visual Assistant for Blind People using Raspberry Pi"- Information Technology, Bharati Vidyapeeth's College of Engineering for Women's Pune, Maharashtra, India2AssociateProfessor,InformationTechnology,BharatiVidyapeeth'sCollege of Engineering

Vipul Sharma et al.,[9] "Virtual Assistant for Visually Impaired"- Sharan ThanneeruDepartment of Information technology, Pillai College of Engineering, University of Mumbai.

Gupta, Lavanya et al.,[10] Vipasha Verma, Nidhi Kalra, and Seemu Sharma. "Approaches in Assistive Technology A Survey on Existing Assistive Wearable Technology for the Visually Impaired." In ComputerNetworks, Big Data and IoT, pp. 541-556. Springer, Singapore, 2021.

CHAPTER 3

THEORITICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability

3.2 SYSTEM ARCHITECTURE

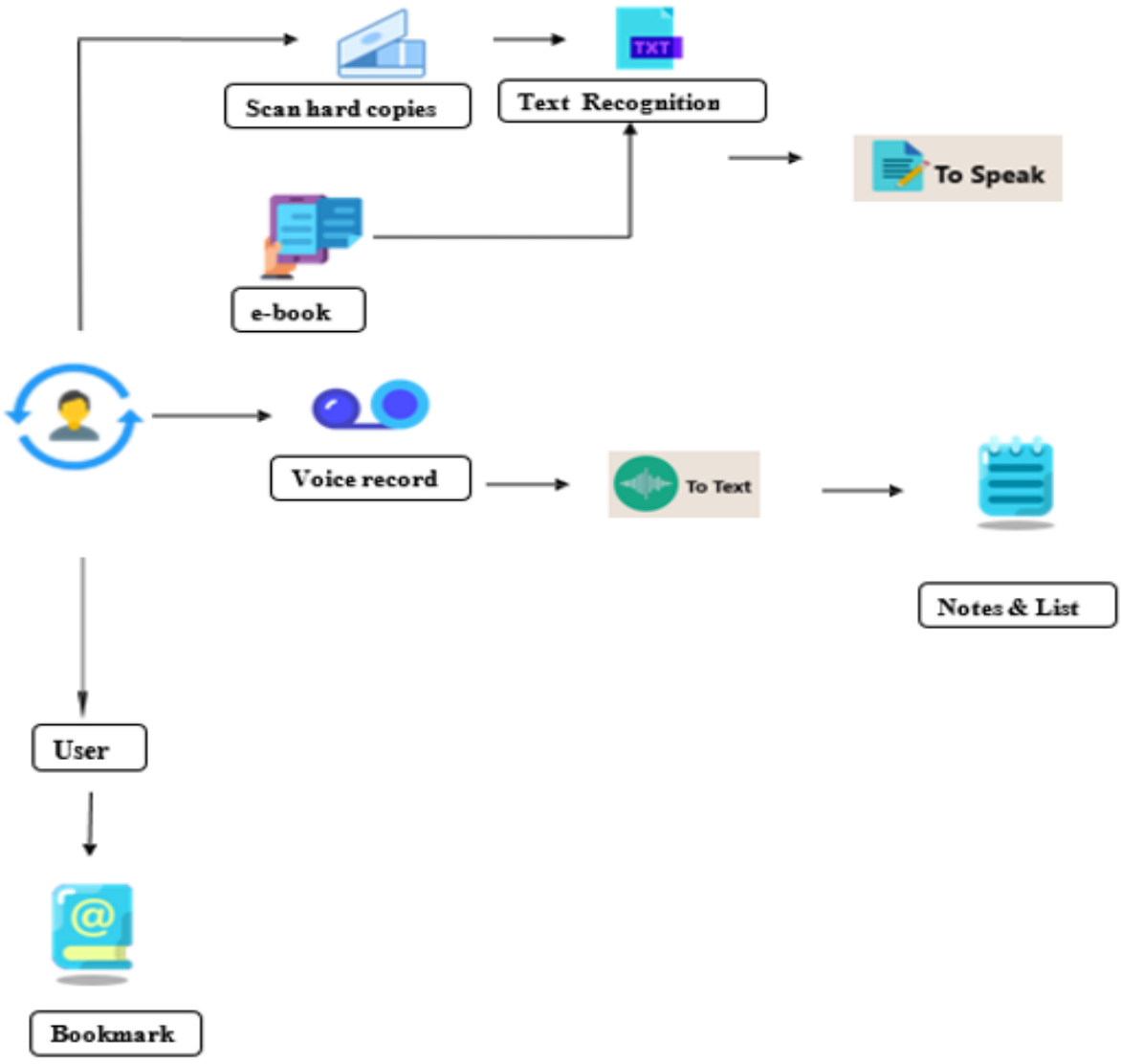

Fig 3.1 System Architecture

System architecture Fig 3.1 The suggested solution is an Android application designed to assist the blind or visually handicapped in learning. The program uses a voice-user interface to allow hands-free access to its numerous functionalities. When the user initially opens the program, they interact with the main menu. The voice assistant displays the available options or modules and instructs the user on how to pick them. To ensure excellent voice recognition, the user will be prompted to begin speaking at all levels. The application offers a variety of features, including scanning hard copies and converting them to audiobooks, turning e-books to audiobooks, Answering simple questions, voice recording, and bookmarking. It allows the user to scan physical copies of books or papers, detect text in the scanned images, and convert the content to an audiobook.The inbuilt voice assistant supports the user in navigating the app and answering basic inquiries via online search. Furthermore, relevant pages and locations in the audiobook may be highlighted, and users can utilize the built-in voice recorder to record themselves speaking and convert it to text for notes or lists Nyx employs a voice-user interface to help its users execute various activities. When you initially launch the app, it introduces itself and reads the menu loudly. It provides instructions on how to use the application. After hearing a "wake word" or instruction ('Nyx' in this case), the program can conduct a variety of tasks before beginning to record.The user will be asked to start speaking at every volume level in order to guarantee superior voice recognition. The program has many functions, such as voice recording, bookmarking, answering basic questions, scanning paper copies and converting them to audiobooks, and converting e-books to audiobooks. It enables the user to scan hard copy books or documents, identify text in scanned photos, and turn the material into an audiobook. The built-in voice assistant helps the user search online for basic answers to common questions and navigate the app. Additionally, users can highlight pertinent pages and locations in the audiobook and record their own voice using the built-in voice recorder, which can then be converted to text for lists or notes. When the user initially opens the program, they interact with the main menu. The voice assistant displays the available options or modules and instructs the user on how to pick them. To ensure excellent voice recognition, the user will be prompted to begin.

3.3 PROPOSED METHODOLOGY

The proposed system is an android application centered on helping the blind or visually impaired in studying. The voice assistant provides the user with the available options or modules and instructions on selecting them. The user will also be provided with an indication to start speaking at all stages, for effective speech recognition. The application provides various functionalities such as scanning hard copies and converting them to audio, converting e books to audio books, answering basic queries, voice recording and bookmarking. It lets the user scan hard copies of books or documents, recognizes the text present in the scanned images and converts the text to audios. The embedded voice assistant helps in answering basic queries through web searching and assists the user in handling the app. Additionally important pages and areas in the audio-book can be marked and users can also use an inbuilt voice recorder to record Themselves speaking and convert it to text to keep notes or lists.

Advantages

- Visually Impaired Students easy to use this application.
- This Application provides voice using book mark.

3.4 MODULE DESIGN

3.4.1 System Design

The suggested solution System design Fig 3.4.1.1 is an Android application designed to assist the blind or visually handicapped in learning. The program uses a voice-user interface to allow hands-free access to its numerous functionalities. When the user initially opens the program, they interact with the main menu. The voice assistant displays the available options or modules and instructs the user on how to pick them. To ensure excellent voice recognition, the user will be prompted to begin speaking at all levels.

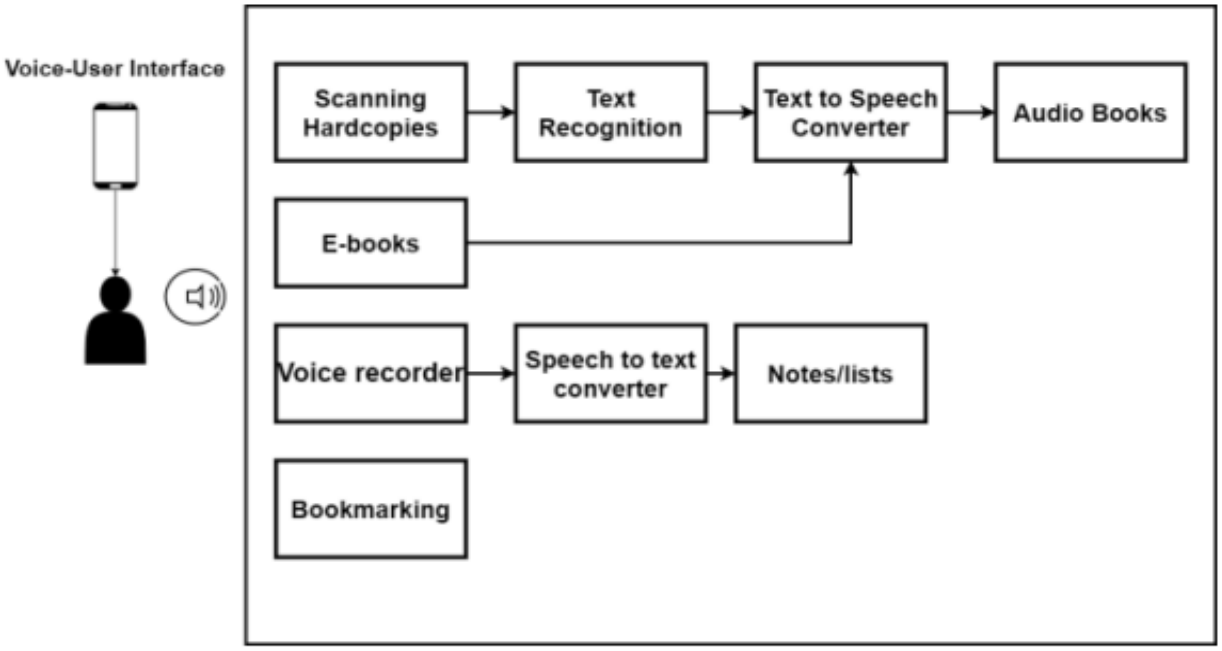

Fig 3.4.1.1 System Design

3.4.2 Sequence Diagram

A Sequence diagram Fig 3.4.2.1 is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

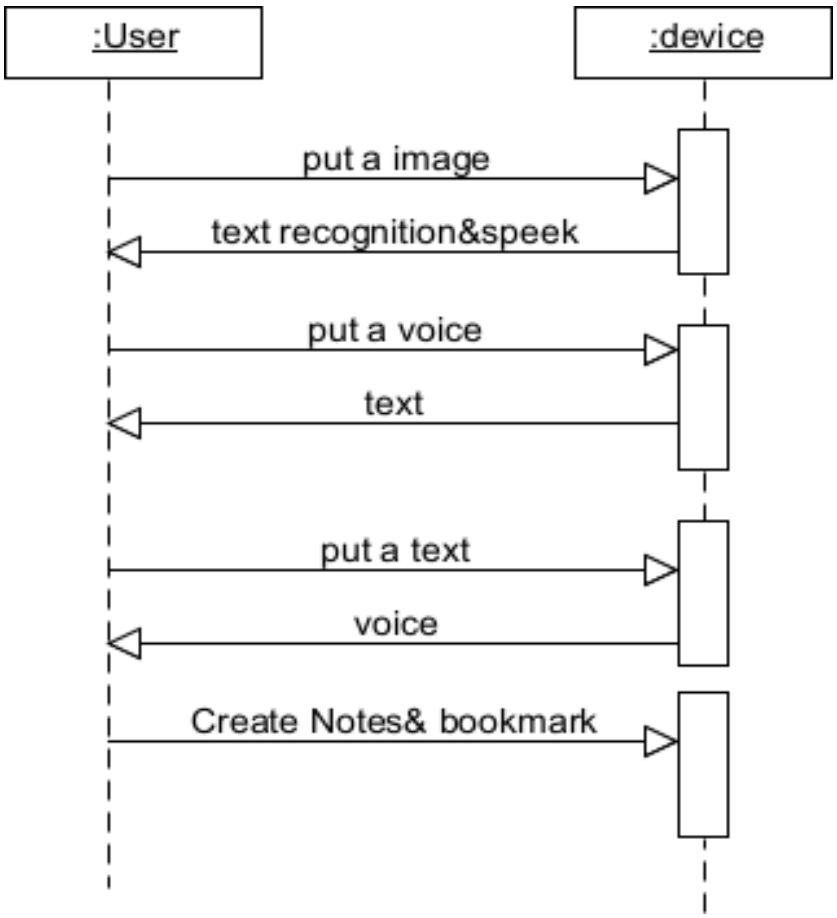

Fig. 3.4.2.1 Sequence Diagram

3.4.3 Use Case Diagram

Unified Modeling Language (UML) Fig.3.4.3.1 is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems.

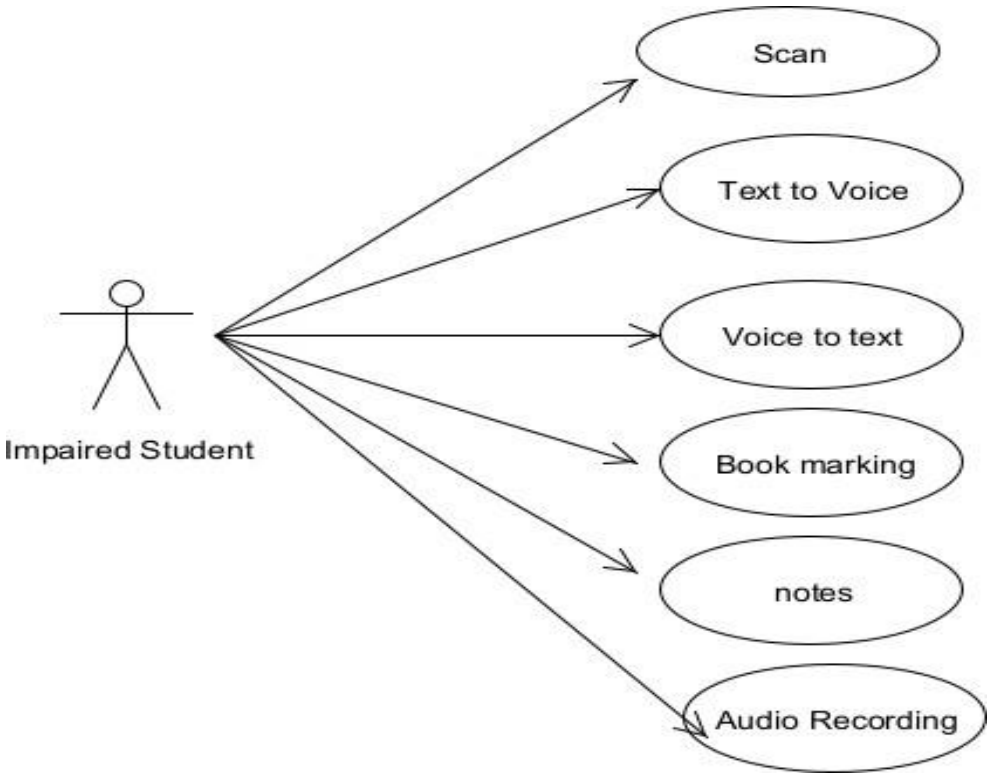

Fig. 3.4.3.1 Use Case Diagram

3.4.4 Activity Diagram

Activity diagram Fig. 3.4.4.1 is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

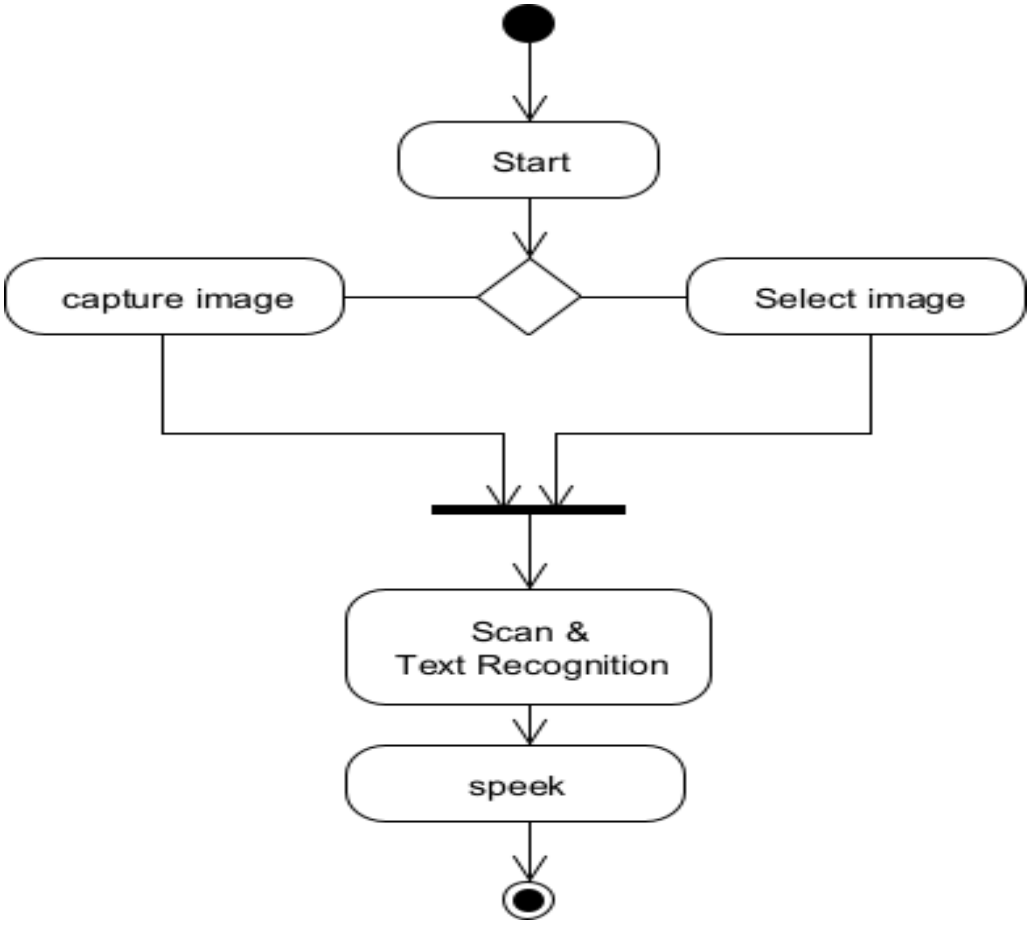

Fig. 3.4.4.1 Activity Diagram

3.4.5 Collaboration Diagram

UML Collaboration Diagrams Fig.3.4.5.1 illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

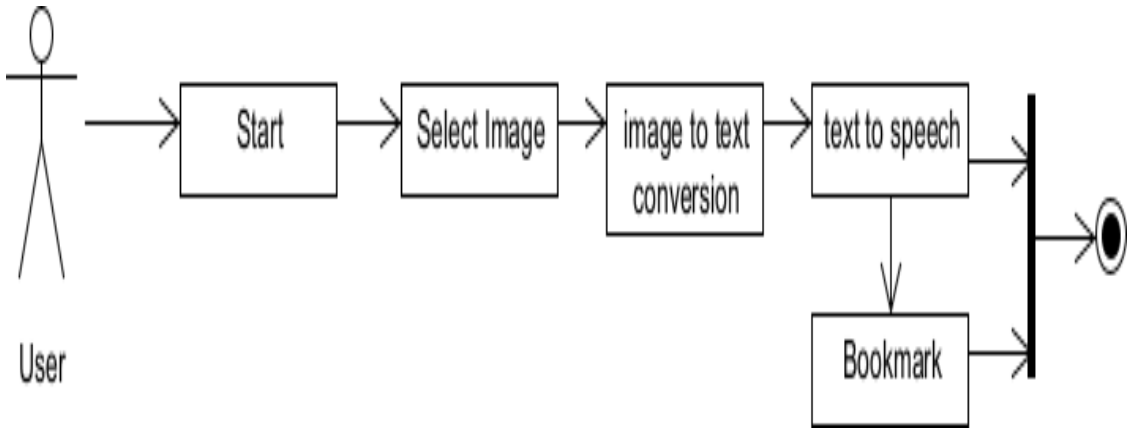

Fig. 3.4.5.1 Collaboration Diagram

3.4.6 Class Diagram

Class Diagram Fig.3.4.6.1 is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. Includes a set of graphic notation techniques to create visual models of software intensive systems.

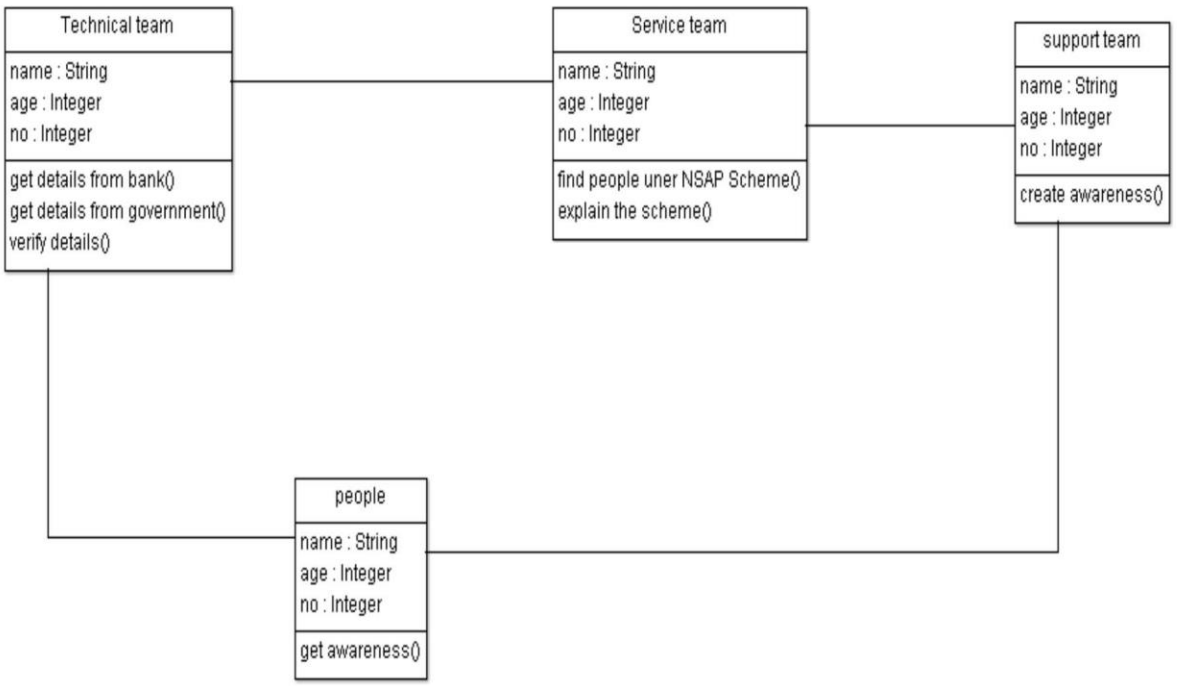

Fig 3.4.6.1 Class Diagram

3.4.7 Deployment Diagram

A deployment diagram Fig 3.4.7.1 is a type of diagram in the Unified Modeling Language (UML) that shows the physical aspects of an object-oriented software system. It models the run-time configuration in a static view, and visualizes the distribution of components in an application. Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems.

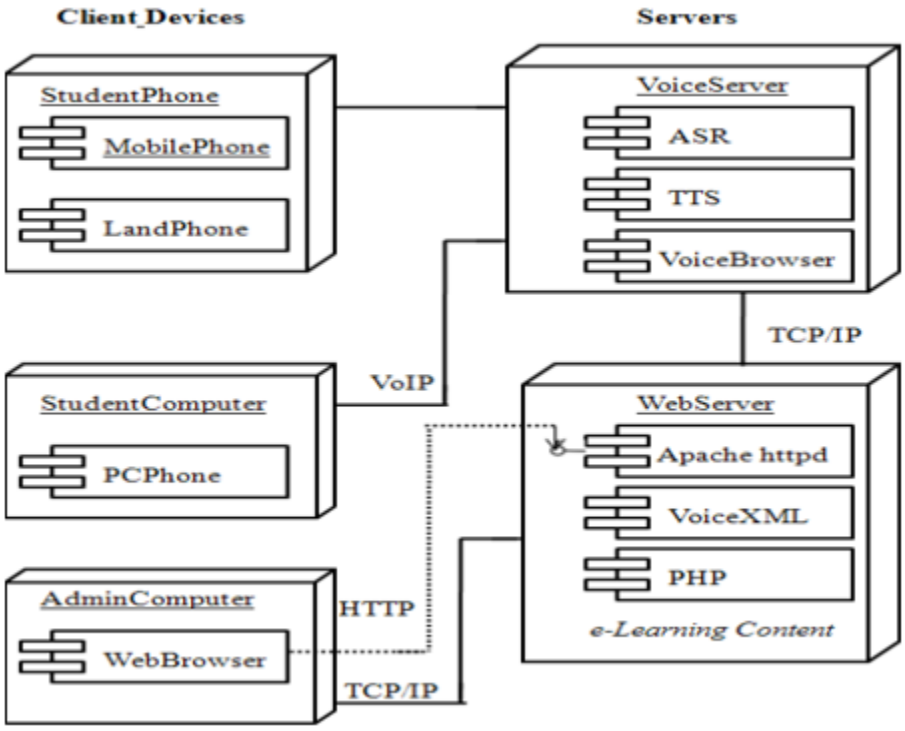

Fig 3.4.7.1 Deployment Diagram

3.4.8 Component Diagram

The purpose of a component diagram Fig 3.4.8.1 is to show the relationship between different components in a system. For the purpose of UML 2.0, the term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.There exists a whole development approach that revolves around components: component-based development (CBD). In this approach, component diagrams allow the planner to identify the different components so the whole system does what it's supposed to do.

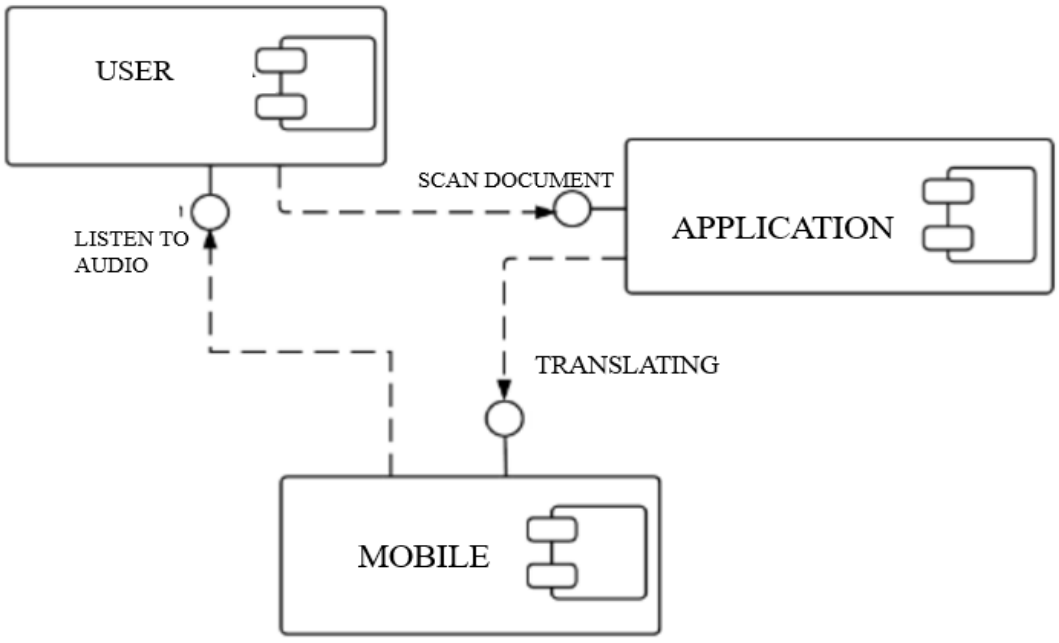

Fig 3.4.8.1 Component Diagram

3.4.9 Data Flow Diagram

A Data Flow Diagram (DFD) Fig. 3.4.9.1 is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

Level 0

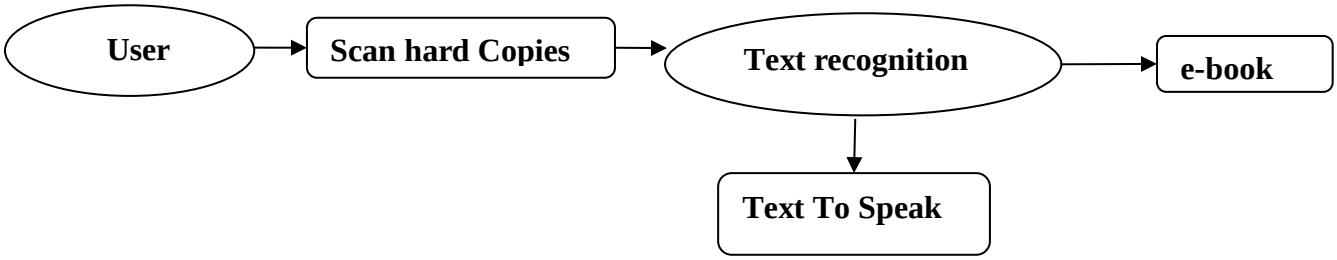

Fig 3.4.9.1 Level 0 Data Flow Diagram

Level 1

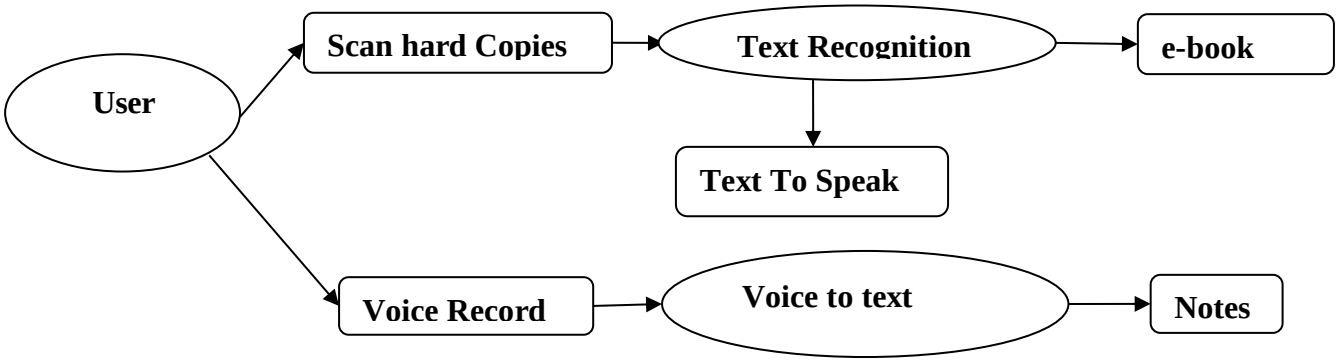

Fig 3.4.9.1 Level 1 Data Flow Diagram

Level 2

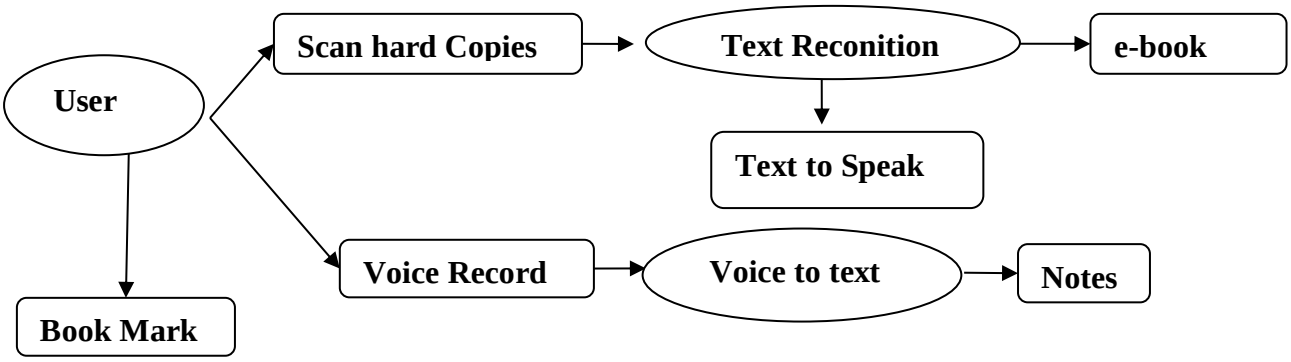

Fig 3.4.9.1 Level 2 Data Flow Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

4.1 MODULE EXPLANATION

4.1.1.E-Book

The user can add e-books into the application and further convert the books or documents into audio. An eBook, short for electronic book, is a digital version of a printed book that can be read on a computer, tablet, smartphone, or dedicated eBook reader device. eBooks have gained significant popularity over the past few decades due to advancements in technology and the widespread adoption of digital devices. Here's a detailed explanation of various aspects related to eBooks

Formats eBooks are available in various formats, including EPUB, PDF, MOBI, AZW, and others. Each format has its own specifications and compatibility with different devices and eReaders. EPUB is one of the most common formats and is widely supported across multiple platforms.

Access eBooks can be purchased from online retailers such as Amazon Kindle Store, Apple Books, Google Play Books, and Barnes & Noble, or borrowed from libraries that offer digital lending services. Some eBooks are available for free, especially classics that are in the public domain.

Reading Devices eBooks can be read on a wide range of devices, including smartphones, tablets, eReaders (such as Amazon Kindle, Barnes & Noble Nook, or Kobo), and computers. Dedicated eReaders offer features such as adjustable font sizes, backlighting, and long battery life, making them popular choices for reading eBooks.

Features eBooks often include features that enhance the reading experience, such as the ability to change font sizes, adjust text formatting, search for specific words or phrases, highlight passages, take notes, and bookmark pages. Some eBooks also support multimedia elements such as audio and video.

Advantages

- **Portability** eBooks allow readers to carry an entire library of books in a single device, making it convenient for travel and commuting.

- **Instant Access** eBooks can be purchased and downloaded instantly, eliminating the need to visit a physical bookstore or wait for shipping.

- **Environmentally Friendly** Since eBooks are digital, they do not require paper or ink, reducing the environmental impact associated with traditional print books.

- **Cost-effective** eBooks are often cheaper than their printed counterparts, and many classic titles are available for free.

Challenges

- **Eye Strain** Extended screen time when reading eBooks on devices with backlit displays can cause eye strain and fatigue.

- **Compatibility Issues** Different eBook formats may not be compatible with all devices, requiring users to convert files or use specific apps or software.

- **Digital Rights Management (DRM)** Some eBooks are protected by DRM, which restricts copying, sharing, or transferring them to other devices.

- **Publishing** Authors and publishers can easily self-publish eBooks through online platforms, eliminating the need for traditional publishing houses. This has led to a

proliferation of independent authors and a more diverse range of content available to readers.

4.1.2 My Notes

This option lets the user create and keep their own notes or lists . The voice assistant provides an indication to the speaker to start speaking and records whatever the user speaks after the indication signal. The user can then listen to the notes when needed. Also the recorded notes can be converted to text if the user wishes to keep them as documents.

Creating a detailed report on "MyNote," which is assumed to be a favorite note feature for a user, involves outlining the purpose, functionality, user interface, and potential improvements or enhancements. Below is a detailed report

Introduction

"MyNote" is a feature designed to enable users to mark and organize their favorite notes within an application or platform. This feature aims to enhance user experience by providing a convenient way to access important or frequently referenced information.

Purpose

The primary purpose of MyNote is to allow users to

- Easily bookmark and access important notes.
- Organize favorite notes into categories or tags for better management.
- Quickly retrieve saved notes for reference or further action.

Functionality

- Bookmarking Users can mark any note as a favorite by selecting a dedicated "Favorite" button/icon within the note interface.
- Organizing Users can categorize their favorite notes by assigning tags or labels. This allows for better organization and retrieval of notes based on specific topics or themes.
- Search A search feature enables users to quickly locate specific favorite notes by entering keywords or phrases.
- Sorting Options for sorting favorite notes by date, title, or category can enhance usability.
- Editing Users may have the option to edit or annotate their favorite notes directly within the MyNote interface.
- Sharing Integration with sharing functionality allows users to share favorite notes with others via email, messaging apps, or social media platforms.

- Offline Access Capability to access favorite notes offline ensures users can view their saved content even without an internet connection.

User Interface

The user interface of MyNote should be intuitive and user-friendly. Key elements include

- Favorites List A centralized list/grid view displaying all favorite notes.
- Tagging/Categorization Options to create, edit, and apply tags to organize favorite notes.
- Search Bar A prominent search bar for quick retrieval of specific notes.
- Options Menu Access to additional functionalities such as sorting, editing, sharing, and settings.
- Note Preview A brief preview of each favorite note, including title, content snippet, and any attached media.
- Actions Clear buttons/icons for bookmarking, editing, sharing, and deleting favorite notes.

Potential Enhancements

- Sync Across Devices Implementing synchronization capabilities to ensure MyNote data is accessible across multiple devices.
- Customization Allowing users to customize the appearance and layout of their MyNote interface.
- Integration Integration with other applications or platforms, such as task managers or calendar apps, to provide a comprehensive productivity solution.
- Collaboration Collaboration features enabling users to share and collaborate on favorite notes with others in real-time.
- Analytics Providing insights and analytics on usage patterns, such as most frequently accessed notes or popular tags/categories.

Conclusion

MyNote is a valuable feature designed to enhance user productivity and organization by enabling the bookmarking and organization of favorite notes within an application or platform. With intuitive functionality, a user-friendly interface, and potential enhancements, MyNote aims to provide a seamless and efficient note management solution for users.

Trends The eBook market continues to evolve, with trends such as subscription-based services (e.g., Kindle Unlimited), audiobook integration, and enhanced eBooks with interactive elements gaining traction.

Overall, eBooks offer a convenient and flexible way to access and enjoy reading material in a

digital format, catering to the evolving preferences of modern readers.

4.1.3 Scan

This option lets you scan pages of hard copies. The user can also add already existing images if required. The application uses OCR to extract the text present in the scanned images and prints it out. The text can then be converted to audio

Scanning a document involves converting a physical document, such as a piece of paper, into a digital format. This digital format can be stored, edited, shared, or printed using electronic devices such as scanners, smartphones, or digital cameras. Here's a detailed explanation of the process and its components

Preparation Before scanning a document, it's essential to prepare the document by ensuring it's clean, flat, and free from any wrinkles or folds. This helps to produce a clear and legible scan.

Selection of Scanning Device There are various devices available for scanning documents, including flatbed scanners, sheet-fed scanners, multifunction printers with scanning capabilities, dedicated portable scanners, and smartphone apps. The choice of device depends on factors such as document size, quality requirements, and portability needs.

Configuration Settings Most scanning devices allow users to adjust settings such as resolution (measured in dots per inch or DPI), color mode (black and white, grayscale, or color), file format (PDF, JPEG, TIFF, etc.), and scanning area (full page, custom crop, etc.). These settings can affect the quality and file size of the scanned document.

Placement of Document Place the document face-down on the scanner glass or into the feeder tray, ensuring proper alignment and orientation. For multiple-page documents, arrange the pages in the correct order to maintain sequence.

Initiating the Scan Depending on the scanning device, initiate the scan using physical buttons on

the scanner, software interface on the computer, or mobile app on the smartphone. Follow the on-screen prompts to adjust settings and preview the scanned image if available.

Reviewing and Editing After scanning, review the scanned document to ensure clarity, legibility, and completeness. Many scanning software applications offer editing tools such as cropping, rotating, adjusting brightness and contrast, and removing background noise or artifacts.

Saving and Naming Choose a location on your computer or device to save the scanned document. Provide a descriptive filename that reflects the content of the document for easy identification later. Consider organizing scanned documents into folders or categories to facilitate management and retrieval.

Post-Processing (Optional) Depending on the intended use of the scanned document, additional post-processing may be necessary. This could include OCR (Optical Character Recognition) to convert scanned text into editable and searchable text, compression to reduce file size, encryption for security purposes, or conversion to a specific file format.

Sharing and Distribution Once scanned and saved, the document can be shared electronically via email, cloud storage services, messaging apps, or shared network folders. Ensure compliance with any relevant privacy or security policies when sharing sensitive documents.

Archiving and Storage Consider long-term storage and archiving options for scanned documents, especially for important or legal documents. Back up scanned documents regularly to prevent loss due to hardware failure or data corruption.

Overall, scanning documents provides a convenient and efficient way to digitize physical content, enabling easy access, sharing, and management of information in a digital format.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 TESTING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

TEST CASE ID	**TESTCASE/ ACTIONTOBE PERFORMED**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/ FAIL**
1.	Selecting “TO SPEAK” button	Display to speak page	Display to speak page	Pass
2.	Selecting “SCAN” button	Display scan page	Display scan page	Pass
3.	Select “TO TEXT” button	Enter into To text page	Enters into to text page	Pass
4.	Select “SAVE FILE” Button	Enter into save file page	Enters into save file page	Pass

5.	Selecting "CREATE A NEW CONTENT "button	Display create a new content page	Displays create a new content page	Pass
6.	Selecting" SAVE" button	Display save page	Displays save page	Pass
7.	Selecting "VIEW PROFILE"	Display the details and, etc.,	Display the details and, etc.,	Pass
8	Selecting "CANCEL"	Display cancel new request, in progress request.	Displays cancel new request, in progress request.	Pass
9.	Clicking the " TAKE A PHOTO "	Display full details about the request	Display full details about the request	Pass
10.	Clicking the "ADD TO BOOK" button	Add product to the book.	Add products to the book	Pass
11.	Selecting "SCANNER"	Display SCAN	Displays SCAN	Pass

12.	Selecting "AUDIO RESPONSE" button	Displays audio details	Displays audio details	Pass
13.	Select "SIGNOUT" button	Logs out the user	Logs out the user	Pass
14.	Select "E BOOK" button	Show the ebook page and book to select	Show the ebook page and book to select	pass
15.	Select "CREATE NOTE" button	Show the create note button and works properly	Show the create note button and works properly	pass
16.	Select"AUDIO" button	Show the audio to play and book to add	Show the audio to play and book to add	pass
17.	Select "BROWSE " button	Show the browser to add book to scan	Show the browser to add book to scan	pass

Table No 5.1.1 Test Case and Report

5.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows. An integrated voice assistant on the Android platform for blind people described in , can send or receive text messages, make voice calls, make notes on mobile using voice and check the battery level. Here, messages received will be notified as voice itself. The technology adapted for recognizing voice includes speech synthesis technology.There is a growing awareness among parents, teachers, blind youth, and the adult blind community that the education which blind children are receiving is failing them. They are not receiving a quality education which can prepare them to compete in the demanding high tech economy and society of the 21st Century. An integrated voice assistant on the Android platform for blind people described in , can send or receive text messages, make voice calls, make notes on mobile using voice and check the battery level. Here, messages received will be notified as voice itself. The technology adapted for recognizing voice includes speech synthesis technology. The blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows. When you initially launch the app, it introduces itself and reads the menu loudly. It provides instructions on how to use the application. After hearing a "wake word" or instruction ('Nyx' in this case), the program can conduct a variety of tasks before beginning to record.The user will be asked to start speaking at every volume level in order to guarantee superior voice recognition.

5.3 RESULTS

Technology Integration -

- Explore and implement cutting-edge assistive technologies such as screen readers, tactile learning materials, and adaptive software to enhance the learning experience.

- Develop and test user-friendly educational apps specifically designed for the visually impaired.

Inclusive Curriculum Development -

- Collaborate with educators, content creators, and accessibility experts to adapt existing educational materials and develop new, inclusive curriculum content.

- Ensure that the curriculum covers a wide range of subjects and is tailored to various educational levels.

Teacher Training and Support

- Provide training programs for educators on best practices for teaching visually impaired students. - Establish a support system for teachers, including resources and professional development opportunities.

Community Engagement

- Foster a sense of community among visually impaired students through mentorship programs

Pilot Programs

- Launch pilot programs in collaboration with select educational institutions to test and refine the effectiveness of technology solutions and curriculum adaptations.

Partnerships

- Establish partnerships with tech companies, educational institutions, and non-profit organizations to
resources and expertise.

Feedback Mechanisms

- Implement feedback mechanisms involving students, educators, and parents to continuously improve and adapt the project's strategies based on real-world experiences.

Educational Achievement

- Track the academic performance of visually impaired students participating in the program compared to previous years.

User Satisfaction

- Collect feedback from students, teachers, and parents to assess the satisfaction levels with the project's initiatives.

Inclusion Metrics

- Monitor the increased enrollment and retention rates of visually impaired students in mainstream educational settings.

5.3 DISCUSSION

Understanding User Needs: It's crucial to conduct thorough research and engage with visually impaired communities to understand their specific needs, challenges, and preferences. This can involve focus groups, surveys, and interviews to gather insights directly from users.

Accessibility as a Priority: Accessibility should be at the forefront of the app design process. This means adhering to accessibility guidelines such as WCAG (Web Content Accessibility Guidelines) and ensuring compatibility with screen readers, voice commands, and Braille displays.

Customization and Personalization: Recognize that visually impaired users have diverse needs and preferences. Provide customization options such as adjustable font sizes, color schemes, voice settings, and gesture controls to accommodate individual preferences and abilities.

Seamless Navigation and User Experience: Design an intuitive and easy-to-navigate user interface that minimizes complexity and maximizes usability. Utilize clear and concise language,

logical navigation paths, and consistent layout and design elements to enhance user experience.

Integration of Advanced Technologies: Explore the integration of advanced technologies such as machine learning, computer vision, and natural language processing to enhance the app's functionality. Features like object recognition, text-to-speech, and intelligent voice assistants can greatly benefit visually impaired users.

Continuous Improvement and User Feedback: Emphasize the importance of ongoing iteration and improvement based on user feedback. Establish channels for users to provide feedback, report issues, and suggest enhancements, and prioritize incorporating these insights into future updates.

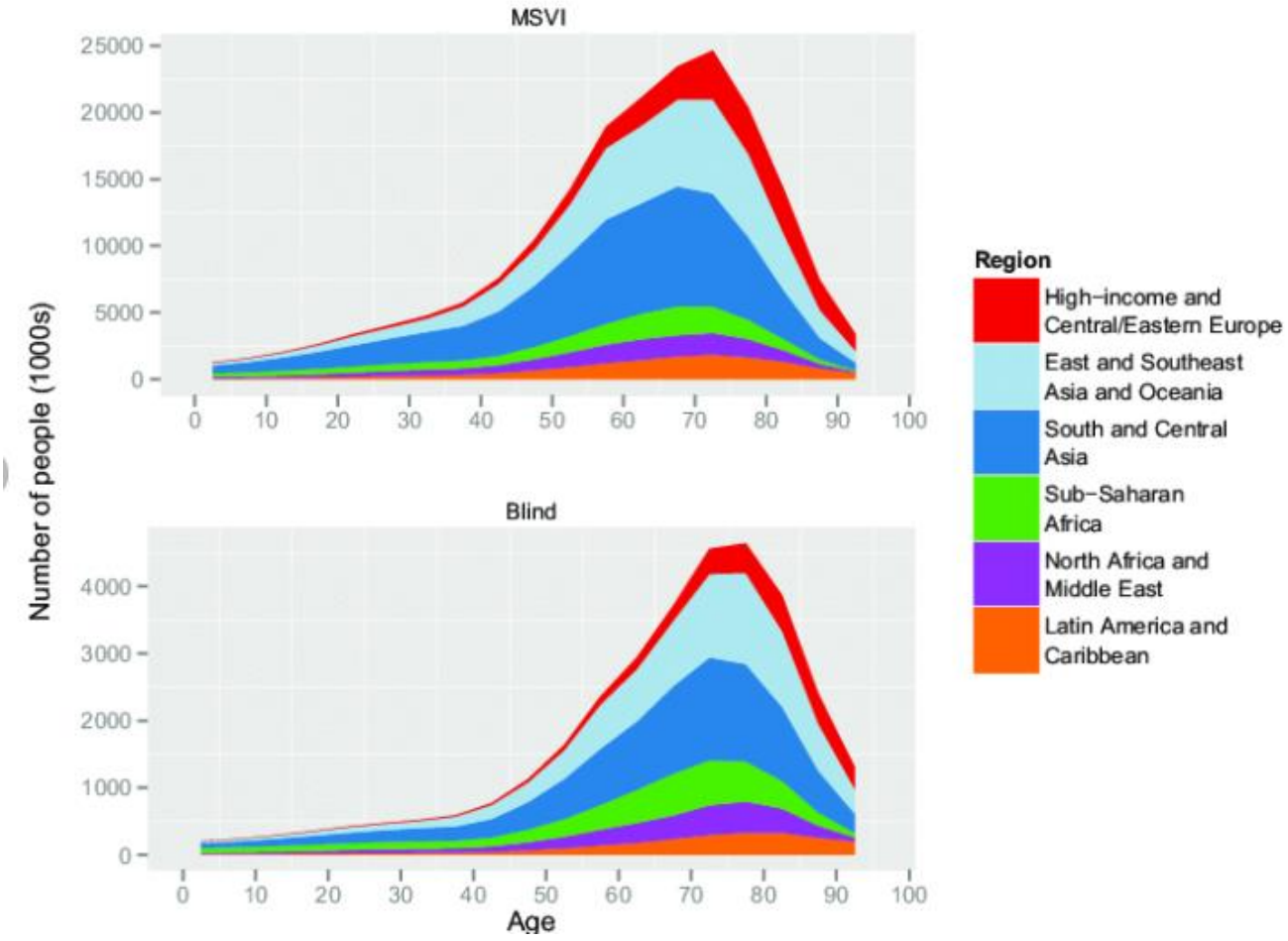

Fig 5.3.1 Number of blind people vs Age

Graph Fig 5.3.1 showing the global population of blind persons and those with moderate and severe vision impairment by region and age. The 21 subregions used in this study are combined into 6 groups. The leading causes of blindness and vision impairment worldwide are: Cataracts, Refractive errors, Diabetic retinopathy, Glaucoma, and Age-related macular degeneration.

Community and Support Networks: Consider incorporating features that foster community engagement and support networks within the app. This could include forums, social networking capabilities, and access to resources and services tailored to the needs of visually impaired individuals.

Collaboration and Partnerships: Collaborate with organizations, experts, and advocates in the field of visual impairment to ensure the app addresses relevant issues and incorporates best practices. Partnering with accessibility-focused organizations can also help raise awareness and reach a wider audience.

Education and Training Resources: Provide access to educational resources, tutorials, and training materials within the app to help users maximize its utility and develop digital literacy skills. Offer guidance on utilizing accessibility features, navigating digital environments, and leveraging assistive technologies effectively.

Ethical Considerations and Privacy: Uphold strict ethical standards and prioritize user privacy and data security throughout the development and deployment process. Ensure transparency regarding data collection practices, obtain informed consent from users, and adhere to relevant privacy regulations.

CHAPTER 6

CONCLUSION & FUTURE WORK

6.1 CONCLUSION

The proposed virtual assistant named “Nyx” is an assistive technology which aids the visually impaired. Nyx provides various functionalities such as scanning-books as well as hard copies, reading texts, keeping notes and adding bookmarks while is to the audio books. The suggested virtual assistant "Nyx" is an assistive device for the visually handicapped. Nyx offers a variety of features, including the ability to scan both e-books and paper copies, read texts, take notes, and make book marks while listening to audiobooks.
The application's main goal is to make it easier for the blind to access educational materials.Nyx is implemented with a variety of technologies, including natural language processing, machine learning, and TTS APIs. We believe that this document will help those who are blind or visually impaired learn more easily. Despite the fact that there are several voice assistants on the market, aside from a few one-off prototypes, there are no commercially accessible voice assistants that are purely designed for visually impaired persons. Nyx cannot directly compete with integrated voice assistants like Google Assistant, Siri, and Bixby. It operates alongside them while assisting the specialized user group to whom it caters.
The application mainly focuses on providing easy access to study materials for the blind.

6.2 FUTURE ENHANCEMENTS

Voice Commands and Control: Integrate advanced voice commands and control features so users can navigate the app hands-free. Allow users to perform actions such as opening different sections, reading content, and initiating tasks using voice commands.

Enhanced Voice Assistants: Incorporate a sophisticated voice assistant that can understand natural language queries and provide intelligent responses. This can help users perform tasks more efficiently and access information more easily.

Text-to-Speech (TTS) Customization: Offer customization options for TTS, allowing users to adjust the speech rate, pitch, volume, and voice style according to their preferences. Some users may find certain voices easier to understand than others.

Braille Display Support: Provide support for Braille displays, allowing visually impaired users to connect their Braille devices to the app and read content through tactile feedback. Ensure compatibility with popular Braille displays available on the market.

Gesture-based Navigation: Implement gesture-based navigation controls for users who may have difficulty interacting with touchscreens. Allow users to perform actions such as swiping, tapping, and dragging using customizable gestures.

Color and Contrast Customization: Enable users to adjust the color scheme and contrast settings to improve readability and reduce eye strain. Provide options for high-contrast modes and color inversion for users with low vision.

Location-based Services: Integrate location-based services to help visually impaired users navigate their surroundings more effectively. Offer features such as real-time navigation, nearby points of interest, and audible directions using GPS technology.

Object Recognition: Develop functionality for object recognition using the device's camera. Allow users to point the camera at objects, documents, or text and receive audio descriptions or spoken text readings to identify them.

Community Engagement Features: Include features that facilitate community engagement and support networks for visually impaired individuals. This could include forums, social networking capabilities, and peer support groups within the app.

Continuous Feedback and Improvement: Regularly solicit feedback from visually impaired users to identify areas for improvement and incorporate their suggestions into future updates. Actively engage with the user community to ensure that the app meets their evolving needs and preferences.

REFERENCES

[1] Prasanna, S., K. Vani, and V. Venkateswari. “Design and Development of a MobileBased Application for Automated SMS System Using Voice Conversion Techfor Visually Impaired People.” In Innovative Data Communication Technologies and Application, pp. 307-317. Springer, Singapore, 2021.

[2] Gayatri.S, Porkodi Venkatesh, Pushpapriya Premkumar, Department of Computer Science and Engineering Jeppiaar SRR Engineering College, Chennai, Tamil Nadu, India, “Voice Assistant for Visually Impaied”, International Journalof Science and Computing March 2019.

[3] “Virtual Assistant for the Visually Impaired”- Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020) IEEE Conference Record 48766; IEEE Xplore ISBN 978-1-7281-5371-1

[4] “Voice Assistant for Visually Impaired People”-International Research Journal of Engineering and Technology (IRJET), Nishank M. Tembhurne1, Sumedh V Vaidya, Afrin Shiekh, Prof. Swapnil Dravyakar, Computer Science and Engineering, Priyadarshini Institute of Engineering and Technology, Nagpur.

[5] “Application for the Visually Impaired People With Voice Assistant”- Abhijeet Mohanta, Shah Yash Jitendra, Khandelwal Nikita Dinesh, Wable Saurabh Suhas, Aruna K. Gupta, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN 2278-3075, Volume-9 Issue-6, April 2020.

[6] “Digital Assistant for the Visually Impaired ”- Ezekiel Marvin, Department of ComputerSystemEngineering,UniversitasPrasetiyaMulya,BSD,Indonesia; IEEEXploreISBN978-1-7281-4985-1/20

[7] “ An Intelligent Text Reader based on Python”- Proceedings of the Third International Conference on Intelligent Sustainable Systems [ICISS 2020] IEEE Xplore Part Number

CFP20M19-ART; ISBN 978-1-7281- 7089-3.

[8] “Visual Assistant for Blind People using Raspberry Pi”- Tejal Adep1 , Rutuja Nikam1 , Sayali Wanewe1 , Dr. Ketaki B. Naik2 Engineering for Women’s Pune, Maharashtra, India2AssociateProfessor,InformationTechnology,BharatiVidyapeeth’sCollege of Engineering

[9] “Virtual Assistant for Visually Impaired”- Vipul Sharma, Vishal Mahendra Singh, Sharan Thanneeru, Prof. Amol Kharat, Department of Information technology, Pillai College of Engineering, University of Mumbai.

[10] Gupta, Lavanya, Neha Varma, Srishti Agrawal, Vipasha Verma, Nidhi Kalra, and Seemu Sharma. “Approaches in Assistive Technology A Survey on Existing Assistive Wearable Technology for the Visually Impaired.” In ComputerNetworks, Big Data and IoT, pp. 541-556. Springer, Singapore, 2021.

APPENDICES - I

A.1 SOURCE CODE

```
//Home

package com.prjct.nyx;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.app.SearchManager;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.prjct.nyx.audio.MainPage;

import java.util.ArrayList;
import java.util.Locale;

public class Home extends AppCompatActivity {
    private CardView speechtotext, texttospeak, imagescantospeak, pdftospeak, Audio, CreateNotes;
    private EditText editTextInput;
    private ImageView mic;
    private static final int REQUEST_CODE_SPEECH_INPUT = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);


        final Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
        texttospeak = findViewById(R.id.cardview1);
        speechtotext = findViewById(R.id.cardView2);
```

```
        imagescantospeak = findViewById(R.id.cardView3);
        pdftospeak = findViewById(R.id.cardView4);
        Audio = findViewById(R.id.cardview5);
        CreateNotes = findViewById(R.id.cardView6);
        mic = findViewById(R.id.micimageView);

        editTextInput = (EditText) findViewById(R.id.editsearch);

        speechtotext.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final VibrationEffect vibrationEffect1;
                if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                    // this effect creates the vibration of default amplitude for 1000ms(1 sec)

vibrationEffect1=VibrationEffect.createOneShot(1000,VibrationEffect.DEFAULT_AMPLITUDE);

                    // it is safe to cancel other vibrations currently taking place
                    vibrator.cancel();
                    vibrator.vibrate(vibrationEffect1);
                    Intent intent = new Intent(getApplicationContext(), SpeechToText.class);
                    startActivity(intent);
                }

            }
        });
        texttospeak.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                final VibrationEffect vibrationEffect1;

                // this is the only type of the vibration which requires system version Oreo (API 26)
                if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                    // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                    vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                    // it is safe to cancel other vibrations currently taking place
                    vibrator.cancel();
                    vibrator.vibrate(vibrationEffect1);

                    Intent intent = new Intent(getApplicationContext(), TextToSpeak.class);
                    startActivity(intent);
```

```
            }

        }
    });
    imagescantospeak.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            final VibrationEffect vibrationEffect1;

            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                // it is safe to cancel other vibrations currently taking place
                vibrator.cancel();
                vibrator.vibrate(vibrationEffect1);

                Intent intent = new Intent(getApplicationContext(), ScanToSpeak.class);
                startActivity(intent);
            }
        }
    });
    pdftospeak.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            final VibrationEffect vibrationEffect1;

            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                // it is safe to cancel other vibrations currently taking place
                vibrator.cancel();
                vibrator.vibrate(vibrationEffect1);
                Intent intent = new Intent(getApplicationContext(), PdfToSpeak.class);
                startActivity(intent);

            }

        }
    });
    Audio.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {
            final VibrationEffect vibrationEffect1;

            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                // it is safe to cancel other vibrations currently taking place
                vibrator.cancel();
                vibrator.vibrate(vibrationEffect1);

                Intent intent = new Intent(getApplicationContext(), MainPage.class);
                startActivity(intent);
            }
        }
    });
    CreateNotes.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            final VibrationEffect vibrationEffect1;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                // it is safe to cancel other vibrations currently taking place
                vibrator.cancel();
                vibrator.vibrate(vibrationEffect1);

                Intent intent = new Intent(getApplicationContext(), AddingNotes.class);
                startActivity(intent);
            }
        }
    });

}

public void onSearchClick(View v)
{
    try {
        Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
        String term = editTextInput.getText().toString();
        intent.putExtra(SearchManager.QUERY, term);
```

```
            startActivity(intent);

        } catch (Exception e) {
// TODO: handle exception
        }
        editTextInput.getText().clear();
    }
    public void getSpeechInput(View view) {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());

        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(intent, 10);

        } else {
            Toast.makeText(this, "Your Device Don't Support Speech Input", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode){
            case 10:
                if(resultCode==RESULT_OK && data!=null){
                    ArrayList<String> result=data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    editTextInput.setText(result.get(0));
                }
                break;
        }



    }
}

//Voice Search

package com.prjct.nyx;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.pm.PackageManager;
```

```
import android.content.pm.ResolveInfo;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

public class VoiceSeach extends AppCompatActivity {
   EditText ed;
   TextView tv;
   private static final int REQUEST_CODE = 1234;
   Button speak;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_voice_seach);

      ed = (EditText) this.findViewById(R.id.editText1);
      tv = (TextView) this.findViewById(R.id.textView1);
      speak  = (Button) findViewById(R.id.speakButton);
      this.speak = speak;

      // Disable button if no recognition service is present
      PackageManager pm = getPackageManager();
      List<ResolveInfo> activities = pm.queryIntentActivities(new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
      if (activities.size() == 0)
      {
         speak.setEnabled(false);
         speak.setText("Recognizer not present");
      }
      ed.addTextChangedListener(new TextWatcher()
      {@
            Override
      public void beforeTextChanged(CharSequence s, int start, int count, int after)
      {
         // TODO Auto-generated method stub
      }@
            Override
      public void onTextChanged(CharSequence s, int start, int before, int count)
```

```
        {
            // TODO Auto-generated method stub
        }@
                Override
        public void afterTextChanged(Editable s)
        {
            // TODO Auto-generated method stub
            speak.setEnabled(false);
        }
        });
    }
    /**
     * Handle the action of the button being clicked
     */
    public void speakButtonClicked(View v)
    {
        startVoiceRecognitionActivity();
    }
    /**
     * Fire an intent to start the voice recognition activity.
     */
    private void startVoiceRecognitionActivity()
    {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Voice searching...");
        startActivityForResult(intent, REQUEST_CODE);
    }
    /**
     * Handle the results from the voice recognition activity.
     */
    @
            Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data)
    {
        if (requestCode == REQUEST_CODE && resultCode == RESULT_OK)
        {
            // Populate the wordsList with the String values the recognition engine thought it heard
            final ArrayList< String > matches =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            if (!matches.isEmpty())
            {
                String Query = matches.get(0);
                ed.setText(Query);
                speak.setEnabled(true);
```

```
            }
        }
        super.onActivityResult(requestCode, resultCode, data);

}
}
```

//User Search

```
package com.prjct.nyx;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Build;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;

public class UserSearch extends AppCompatActivity implements TextToSpeech.OnInitListener{

    private ImageView mic,speech;
    private TextView text;
    String textdata;
    HashMap<String, String> hash = new HashMap<>();
    private String data1;
      String   inndata="null";
    TextToSpeech textToSpeech;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_search);
```

```
        else {
            textToSpeech.speak(inndata, TextToSpeech.QUEUE_FLUSH, null);
        }
    }
    public void speakText() {

        if(textToSpeech != null) {
            textToSpeech.speak(inndata, TextToSpeech.QUEUE_FLUSH, null);
        } else {
            Log.e("YOUR_TAG", "TextToSpeech Null");
        }
    }

}

//Scan To Speak


package com.prjct.nyx;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.text.method.ScrollingMovementMethod;
import android.util.DisplayMetrics;
import android.util.Log;
import android.util.SparseArray;
import android.view.View;
import android.widget.ImageView;
```

```
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.vision.Frame;
import com.google.android.gms.vision.text.TextBlock;
import com.google.android.gms.vision.text.TextRecognizer;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Locale;

public class ScanToSpeak extends AppCompatActivity  implements TextToSpeech.OnInitListener {
    private static final int REQUEST_GALLERY = 0;
    private static final int REQUEST_CAMERA = 1;
    private static final int MY_PERMISSIONS_REQUESTS = 0;
    private static final int REQUEST_CODE_SPEECH_INPUT = 1;

    private static final String TAG = MainActivity.class.getSimpleName();
    private Uri imageUri;
    private TextView detectedTextView;
    private ImageView speak,voice;
    TextToSpeech textToSpeech;
     String detectedText;
    SimpleAdapter adapter;
    int position;
    int noteID;

    @Override
    public void onDestroy() {
        if (textToSpeech != null) {
            textToSpeech.stop();
            textToSpeech.shutdown();
        }
        super.onDestroy();
    }
    private void texttoSpeak() {
        hash.get(textdata);

        text.setText(hash.get(textdata));
        inndata=hash.get(textdata);
```

```
      /* String textuser1 = text.toString();
       if ("".equals(textuser1)) {
          textuser1 = "Please enter some text to speak.";
       }*/

       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
          textToSpeech.speak(inndata, TextToSpeech.QUEUE_FLUSH, null, null);
       }
       else {
          textToSpeech.speak(inndata, TextToSpeech.QUEUE_FLUSH, null);
       }
    }
    public void speakText() {

       if(textToSpeech != null) {
          textToSpeech.speak(inndata, TextToSpeech.QUEUE_FLUSH, null);
       } else {
          Log.e("YOUR_TAG", "TextToSpeech Null");
       }
    }

}

//Scan To Speak


package com.prjct.nyx;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
```

```
import android.provider.MediaStore;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.text.method.ScrollingMovementMethod;
import android.util.DisplayMetrics;
import android.util.Log;
import android.util.SparseArray;
import android.view.View;
import android.widget.ImageView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.vision.Frame;
import com.google.android.gms.vision.text.TextBlock;
import com.google.android.gms.vision.text.TextRecognizer;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

@Override
        public void onClick(View view) {
            final VibrationEffect vibrationEffect1;

            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                // this effect creates the vibration of default amplitude for 1000ms(1 sec)
                vibrationEffect1 = VibrationEffect.createOneShot(1000,
VibrationEffect.DEFAULT_AMPLITUDE);

                // it is safe to cancel other vibrations currently taking place
                vibrator.cancel();
                vibrator.vibrate(vibrationEffect1);

                Intent intent = new Intent(getApplicationContext(), MainPage.class);
                startActivity(intent);
            }
        }
    });
    CreateNotes.setOnClickListener(new View.OnClickListener() {
        @Override
```

APPENDICES - II

A.2 SCREENSHOTS

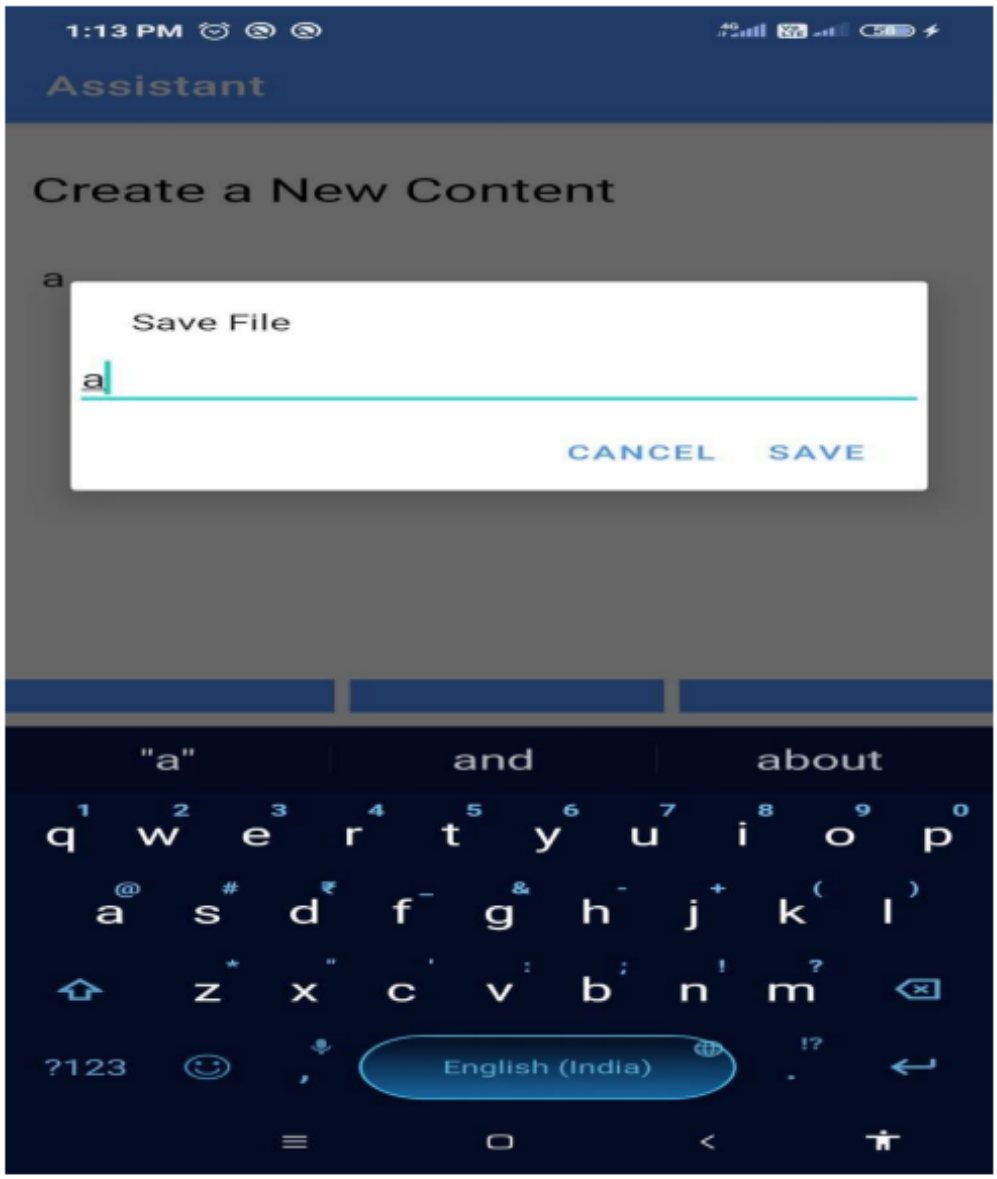

Fig A.2.1 Save file page

Purpose of Saving Files Fig A.2.1

- Preservation Saving files ensures that data is retained even after the current session or application is closed.
- Accessibility Saved files can be accessed at a later time or by other users, providing convenience and flexibility.
- Collaboration Saved files can be shared with others for collaboration or review purposes.
- Backup Saving files to multiple locations or backup systems helps protect against data loss due to hardware failure, software errors, or accidental deletion.

File Formats

- Files can be saved in various formats depending on the type of content and intended use. Common file formats include
  - Documents DOCX (Microsoft Word), PDF (Portable Document Format), TXT (Plain Text), ODT (OpenDocument Text).
  - Images JPEG, PNG, GIF, TIFF, BMP.
  - Videos MP4, AVI, MOV, WMV.
  - Audio MP3, WAV, FLAC, AAC.
  - Spreadsheets XLSX (Microsoft Excel), CSV (Comma-Separated Values).
- Choosing the appropriate file format ensures compatibility with software applications and devices.

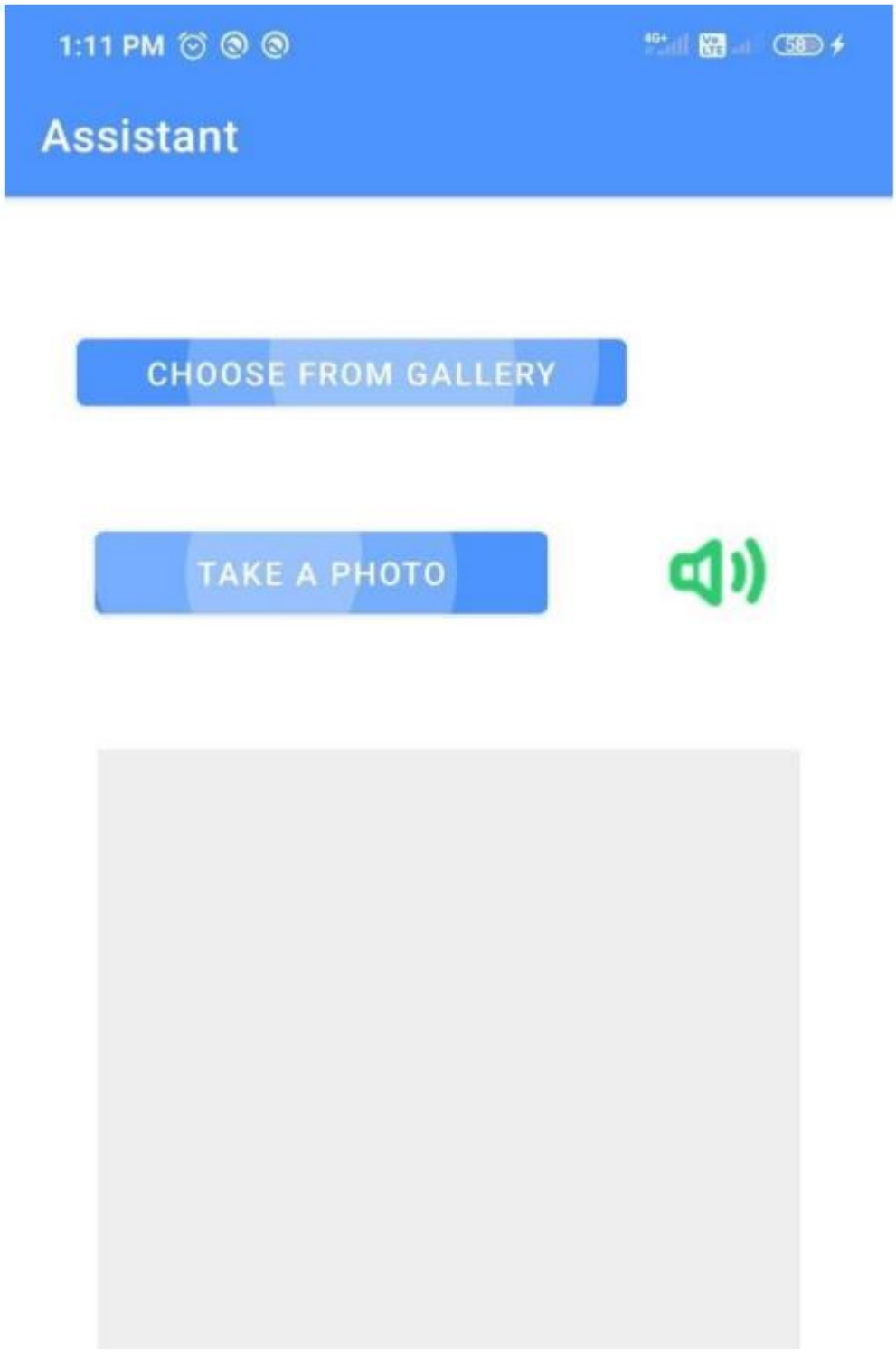

Fig A.2.2 Choose from gallery

To take a photo from a phone for an app, you typically follow these steps Fig A.2.2

Open the Camera App Locate and open the camera app on your smartphone. Most smartphones come pre-installed with a default camera app, but there are also many third-party camera apps available for download.

Position the Camera Position your phone's camera so that it is pointing towards the subject or scene you want to capture. Make sure the subject is well-lit and in focus.

Adjust Camera Settings (Optional) Depending on your camera app, you may have the option to adjust settings such as

- Flash Turn the flash on or off depending on the lighting conditions.
- HDR (High Dynamic Range) Enable HDR mode to capture more detail in high-contrast scenes.
- Gridlines Enable gridlines to help with composition and framing.
- Timer Use the timer function to delay the capture of the photo.

Focus and Exposure Tap on the screen of your phone to set the focus and exposure. Most camera apps will automatically focus on the area you tap and adjust the exposure accordingly. You can usually tap and hold to lock the focus and exposure if needed.

Capture the Photo Once you have framed the shot and adjusted the settings, press the shutter button to capture the photo. Depending on your phone's settings, you may also be able to use physical buttons, voice commands, or gestures to take the photo.

Review and Retake (Optional) After capturing the photo, you can review it to make sure it turned out as expected. If necessary, you can retake the photo by tapping the appropriate button in the camera app.

Save the Photo Once you are satisfied with the photo, you can save it to your phone's gallery or camera roll. Some camera apps may also offer options to share the photo directly to social media, messaging apps, or other apps installed on your phone.

Use the Photo in the App If you took the photo specifically for use in an app, you can now navigate to the app and upload or attach the photo as needed. Depending on the app's functionality, you may be able to select the photo directly from your phone's gallery or camera roll, or you may need to access the photo through the app's interface.

Fig A.2.3 Text to Speak

Here's a guide on how to use the "Text to Audio" feature in an app: Fig A.2.3

Launch the App: Begin by opening the app on your device. You can do this by tapping on the app's icon from your home screen or app drawer.

Locate the "Text to Audio" Feature: Once the app is open, navigate to the section or feature that allows you to convert text to audio. This might be labeled as "Text to Speech," "Voice Synthesis," or something similar, depending on the app's interface.

Input Text: In the "Text to Audio" section, you'll typically find a text box or field where you can input the text you want to convert to audio. Tap on this text box to bring up the keyboard and type or paste the desired text into it.

Adjust Settings (Optional): Depending on the app, you may have options to customize the audio output. This could include selecting a preferred language, voice type, speech rate (speed), volume, or pitch.

Initiate Text-to-Audio Conversion: After inputting the text and adjusting any desired settings, look for a button or option to start the text-to-audio conversion process. This may be labeled as "Convert," "Speak," "Generate Audio," or similar.

Listen to the Audio Output: Once the conversion process is initiated, the app will generate an audio file or stream the synthesized speech based on the input text and selected settings. Listen to the audio output to ensure it meets your expectations.

Save or Share the Audio: Depending on the app's capabilities, you may have options to save the audio file locally on your device or share it with others via messaging apps, email, or social media platforms.

Additional Features and Tips:

- Some apps may offer additional features such as saving custom voice presets, highlighting text as it's spoken, or adjusting pronunciation for specific words.
- Experiment with different settings to find the voice and settings that best suit your preferences and needs.
- If you encounter any difficulties or have questions about using the "Text to Audio" feature, consult the app's help documentation or user guide for further assistance.

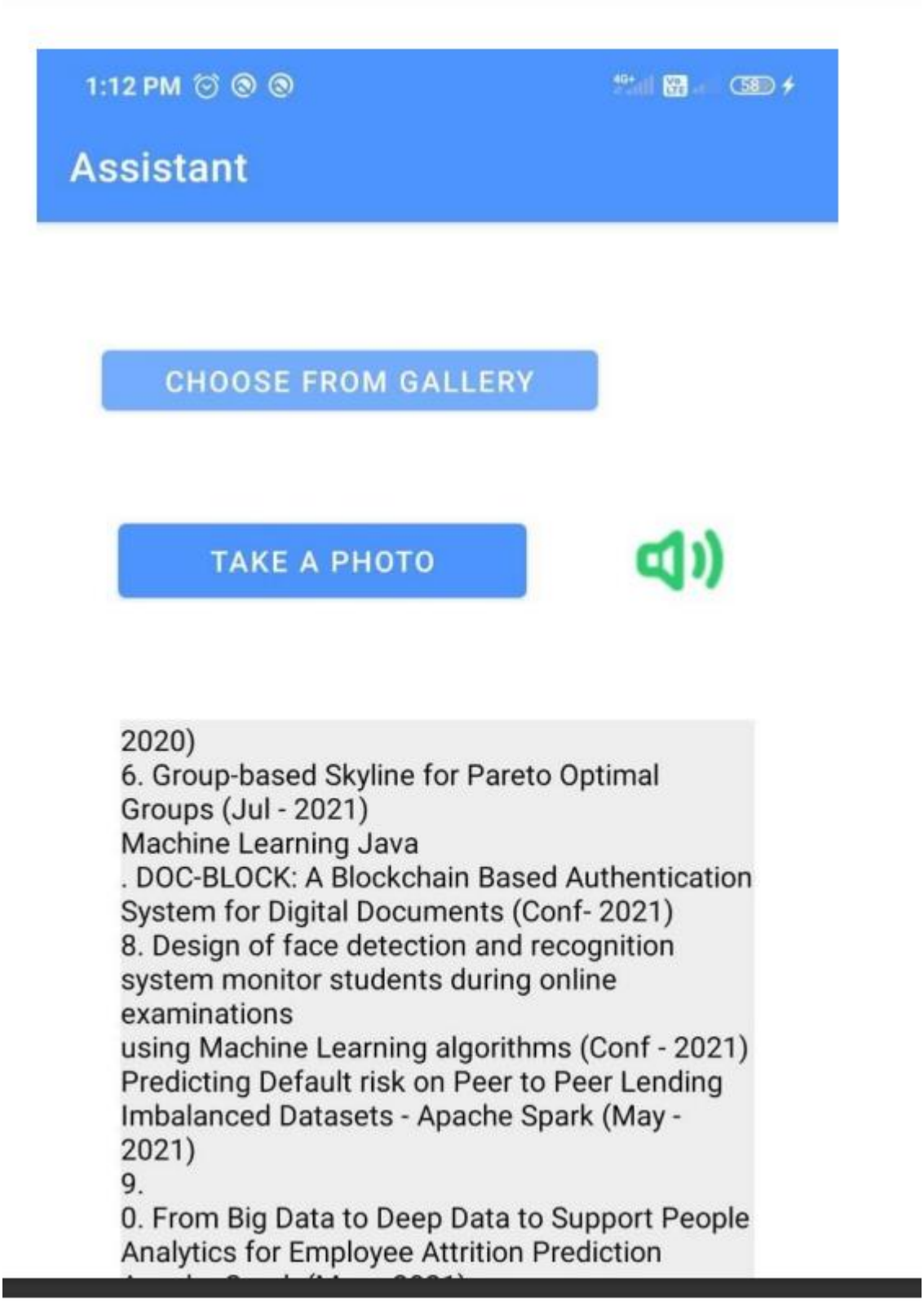

Fig A.2.4 Take a photo

To choose a photo from your device's gallery or photo library, follow these general steps: Fig A.2.4

Open the App: Launch the app on your device that you want to use to select a photo. This could be a messaging app, social media app, photo editing app, etc.

Access the Photo Upload/Attachment Feature: Navigate to the section of the app where you can upload or attach a photo. This may be a compose or messaging window, a profile editing section, or any other area where you can add photos.

Select the "Choose Photo" Option: Look for a button or icon that allows you to choose a photo from your device. This button may be labeled as "Choose Photo," "Upload Photo," "Add Photo," or something similar.

Access Your Device's Gallery: After selecting the "Choose Photo" option, your device will typically prompt you to select the source from which you want to choose the photo. Select "Gallery," "Photos," or a similar option to access your device's photo library.

Navigate to the Desired Photo: Once you're in your device's photo library, browse through the available albums or collections to find the photo you want to select. You can usually scroll vertically to view more photos and horizontally to switch between different albums.

Select the Photo: Tap on the photo you want to choose. Some apps may allow you to select multiple photos if needed. After selecting the photo(s), look for a confirmation button or option to proceed.

Complete the Action: Depending on the app and context, you may need to perform additional steps to complete the action. For example, if you're attaching the photo to a message, you may need to tap a "Send" or "Attach" button to include the photo in the message.

Additional Tips:

- If you're unable to find the desired photo in your device's gallery, make sure it's not located in a specific album or folder. Some devices allow you to create custom albums or organize photos into folders.
- If the app you're using doesn't provide an option to choose a photo, you may need to grant it permission to access your device's photo library in your device's settings.

By following these steps, you can easily choose a photo from your device's gallery and use it within the app for various purposes.

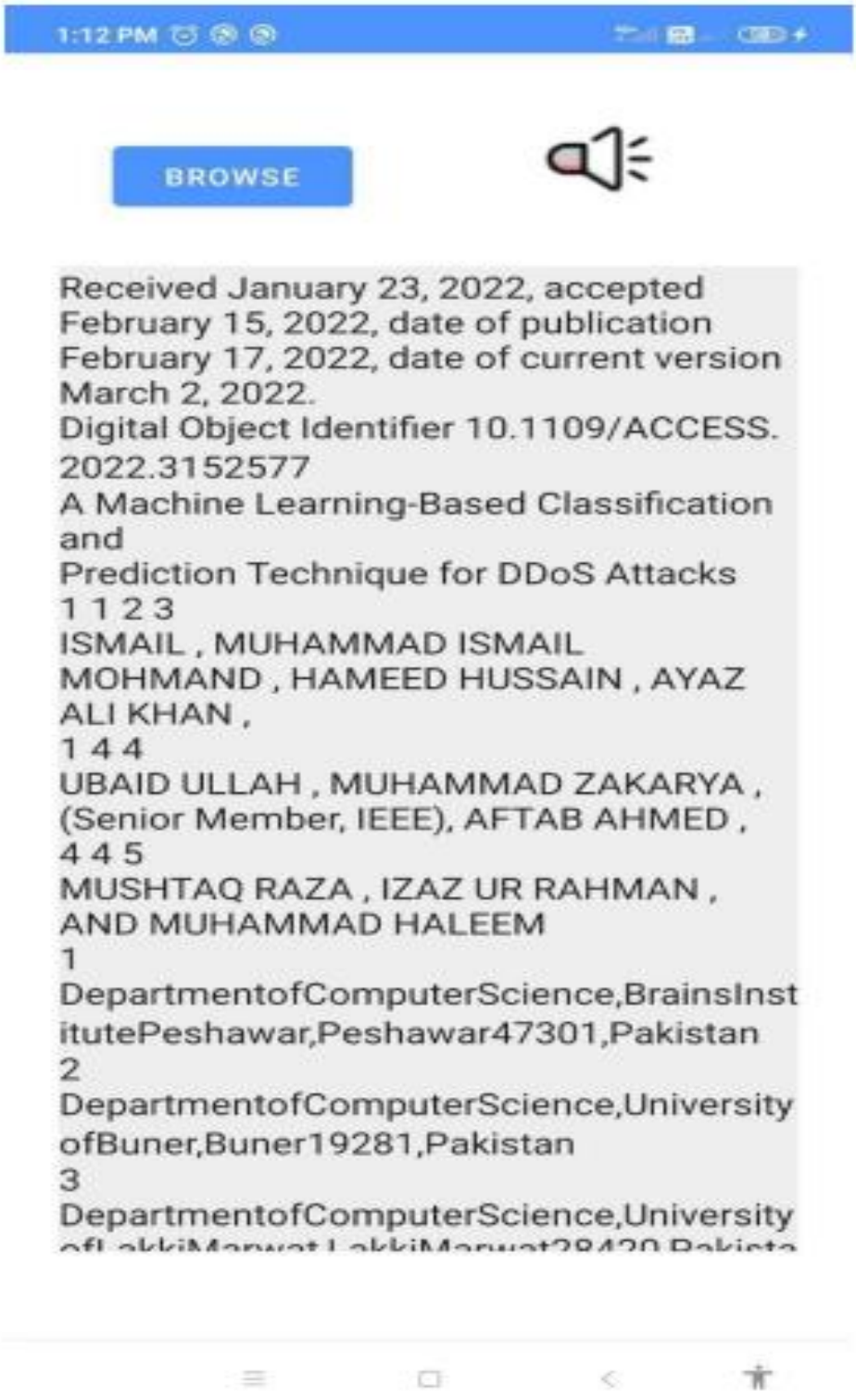

Fig A.2.5 Browse

To browse the document from your mobile Fig A.2.5 :

Turn on Your Device: Ensure your smartphone or computer is turned on and connected to the internet. You can connect via Wi-Fi, mobile data (on smartphones), or Ethernet (on computers).

Open a Web Browser: On your device, locate and open a web browser application. Common web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (on Apple devices), and Opera.

Enter a Website Address (URL): In the address bar at the top of the browser window, type in the URL (Uniform Resource Locator) of the website you want to visit. For example, "www.example.com" or "https://www.example.com". Press Enter or Go on your keyboard to navigate to the website.

Navigate Within Websites: Once you're on a website, you can navigate through its pages by clicking on links, buttons, or tabs within the webpage. Links are usually highlighted or underlined text, images, or buttons that you can click on to go to another page or section of the website.

Use Bookmarks: If you frequently visit certain websites, you can save them as bookmarks for easy access later. Most web browsers allow you to bookmark websites by clicking on the star icon in the address bar or by accessing the bookmarks menu.

Search the Web: If you're looking for specific information, you can use a search engine to find it. Simply type your query into the search bar of your web browser and press Enter. You'll be presented with a list of search results relevant to your query.

Browse Privately (Optional): Most web browsers offer a private browsing mode, also known as Incognito mode (Chrome), Private Browsing (Firefox), or InPrivate Browsing (Edge). This mode allows you to browse the internet without saving your browsing history, cookies, or other data.

Manage Settings and Preferences: You can customize your browsing experience by adjusting settings and preferences in your web browser. This may include changing the default search engine, blocking pop-up windows, clearing browsing data, and managing cookies.

Stay Safe and Secure: When browsing the internet, it's important to stay safe and secure. Be cautious when entering personal information on websites, use strong and unique passwords, and keep your device's software up to date to protect against security threats.

By following these steps, you can effectively browse the internet on your smartphone or computer using a web browser.

A.3 PLAGIARISM REPORT

Result

The system found similarities with other sources, however, the text successfully passed the plagiarism check.

Analysis

Result	10%
Document title	NYX-BasePaper.pdf
Content hash	bd6ac7ead2865018a7b53e26739d54f4
Date	2024-03-14 07:08:51
Check time	17 seconds
Character count	10,000
Special character count	19
Word count	1,543
Number of plagiarized words	104