
Software Requirements Specification

for

<Book Reader of Blind People>

Version 2.0 approved

**Prepared by <Shahir Abdullah Bin Shahnoor (201714054) ,
Shariar Riday (201714037), Nazifa Hamid
(201714044), Takik Hasan
(201714038), Zubair Hasan (201714041)>**

<Group 4 Section B>

<14/01/2021>

Table of Contents

Table of Contents	2
1. Introduction.....	3
1.1 Purpose.....	3
1.2 Comparative Discussion... ..	3
1.3 Scope.....	3
1.4 References	3
2. Overall Description.....	4
2.1 System Environment.....	4
2.2.1 Context Diagram.....	4
2.2 Functional Requirements Specifications.....	5
2.2.1 Feature List	5
2.2.2 Use Case Diagram and Tabular Description.....	6
2.2.3 Data Flow Diagram.....	10
2.3 User Characteristics	11
2.3.1 Visually Impaired People.....	11
2.4 Non – Functional Requirements	11
2.4.1 Performance Requirements	11
2.4.2 Safety Requirements	11
2.4.3 Security Requirements	12
2.4.4 Software Quality Attributes	12
3. Specific Requirements	13
3.1 External Interfaces	13
3.1.1 User Interface.....	13
3.1.2 Hardware Interface.....	15
3.1.3 Software Interface	16
3.1.4 Communication Interface.....	16
3.2 Design Constraints	16
3.2.1 Hardware Constraints.....	16
3.2.2 Software Constraints.....	16

1. Introduction

1.1 Purpose

The product is an audio book reader for any blind individual who wants to know the content of a Bangla text document. They can get this information by taking the photo of the document page and then the product or system will read the information of the document out loud with the help of text to speech technology. This SRS document contains the whole system architecture and model of the system along with the scope of the system and necessary functionalities of the system.

1.2 Comparative Discussion

Various android applications are available that provide audio e-books. There are some applications and sites that read printed books but are not free, and come with a hefty price. There are a few stand-alone devices such as Digital Talking Book CD player that contains digital files of books in CDs and can be carried around. But again, these cannot be used to read printed books and are outdated. Another software used by Bangladeshi blind students is named Jaws which helps them use computer by converting all text into speech. But this software doesn't have an android version so the user cannot convert any kind of document unless they take a photo of that content and then scan it to make a pdf file and put it in their computer which is a very lengthy process and they have to rely on friends or family to help them out. Most importantly, all of these systems only provide services for the English language as there is no Bangla text to speech software for android version.

1.3 Product Scope

This product is for blind students in Bangladesh who face several obstacles everyday life during their study period. Although there are several systems for their ease of education there aren't much which provides a good education system based on Bangla language. And since this product has also an android version so the students or users can also read and know about any contents of a text document remotely with just taking the photo of the page of that document. The scope of the product is mainly based on the students but it can be used for story reading purposes too.

1.4 References

- [1] A. Fardous and S. Afroge, "Handwritten Isolated Bangla Compound Character Recognition," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ECACE.2019.8679258.
- [2] M. J. Hasan, M. F. Wahid and M. S. Alom, "Bangla Compound Character Recognition by Combining Deep Convolutional Neural Network with Bidirectional Long Short-Term Memory," *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, Khulna, Bangladesh, 2019, pp. 1-4, doi: 10.1109/EICT48899.2019.9068817.
- [3] Tan, M. and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *ArXiv abs/1905.11946* (2019): n. pag.
- [4] S. A. Sabab and M. H. Ashmafee, "Blind Reader: An intelligent assistant for blind," *2016 19th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, 2016, pp. 229-234, doi: 10.1109/ICCITECHN.2016.7860200

2 Overall Description

2.1 System Environment

2.1.1 Context Diagram

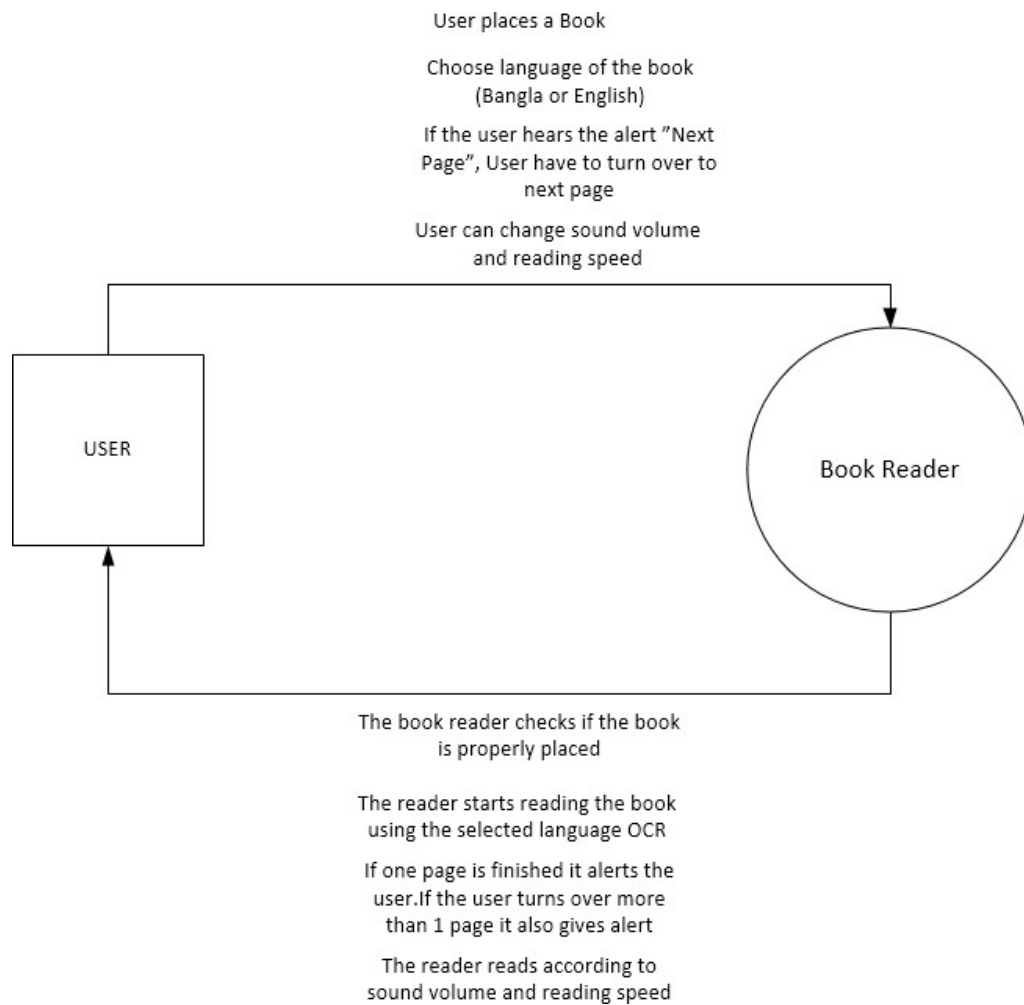


Fig1: Context Diagram

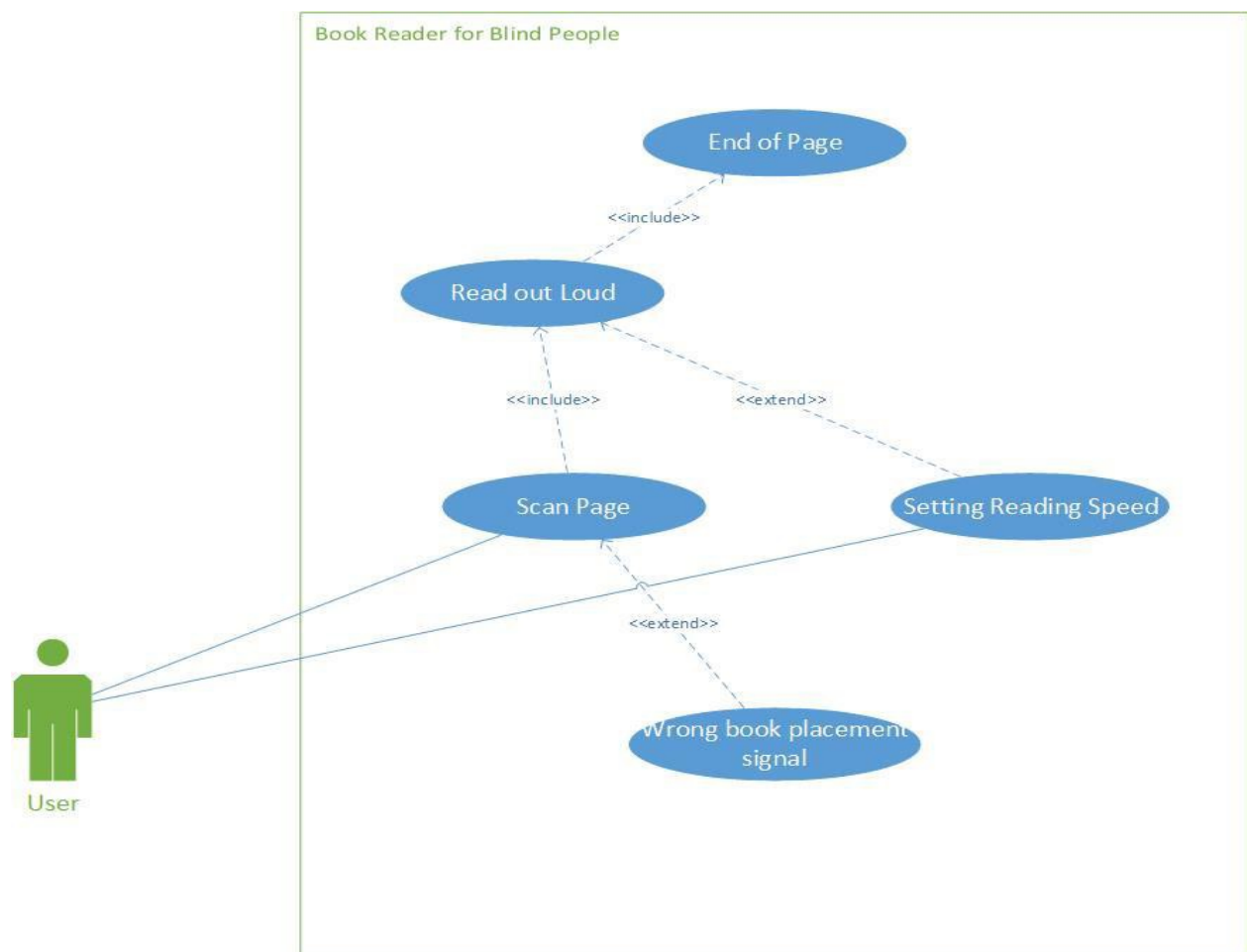
2.2 Functional Requirements Classifications

2.2.1 Feature List

Our designed system contains the following features:

- Taking picture of the text document with a camera placed on the stand.
- Extracting the text from the document using OCR
- Converting extracted text to speech.
- Controlling speed, volume of the speech and navigation with user input through external buttons.

2.2.2 Use Case Diagram and Tabular Description



Primary actor: User

Secondary actors: -

Pre-condition: The system has a book for scanning.

Post-condition: -

Main Flow:

1. The use case is activated when the user requests it.
2. The user puts a book or a page to listen to.
3. The book is placed correctly.
4. The system initiates scanning.
5. If the scanning is finished, and [include “Read out loud”].

Alternative Flow:

- 3a. The book is not placed correctly.
 - 3a1. [Extends use case “Wrong Book Placement Signal”].

Use Case: Wrong book placement signal.

Primary actor: Scan page use case.

Secondary actor: - User.

Pre-condition: The book is placed wrongly.

Post condition: The book is placed correctly.

Main flow:

1. The wrong book placement signal is activated by the scan page use case extension.
2. The user is notified that the book has been placed wrongly.
3. If the user places the book correctly, the use case is concluded.

Alternative flow:

- 3a. The book has not been placed correctly.
 - 3a1. Return to the “Wrong book placement signal” use case.

Use Case: Read out loud.

Primary actor: User.

Secondary actor:

Pre-condition: The book page is scanned.

Post condition: The scanning has reached the end of the page.

Main flow:

1. The use case is activated by the request of the user.
2. The use case has converted the scanned text file into audio file.
3. The audio file is being read by the system.
4. If the user is happy with the reading speed, reading is being continued at the default speed.
5. If the end of page is reached, the use case is concluded and [include use case “End of Page”].

Alternative flow:

- 4a. The user wants to change the reading speed of the system.
 - 4a1. The use case extends the use case “Setting Reading Speed”.
 - 4a2. Return to the main flow.

Use Case: End of Page.

Primary actor: User.

Secondary actor: - System.

Pre-condition: The book has been read till the end.

Post-condition:

Main flow:

1. The use case is activated by the end of the use case “Read out loud”.
2. The use case concludes the reading of the current page in the system.

Use Case: Setting reading speed.

Primary actor: User.

Secondary actor: - Read out loud use case.

Pre-condition: The user requests setting reading speed use case.

Post condition: The reading speed of the system has been changed according to the user's need.

Main flow:

1. The use case is activated by the request of the user.
2. The use case has changed the reading speed of the system.
3. If the reading speed is set to the user's need the use case concludes the shifting of speed and returns to the "Read out loud" use case.

Alternative flow:

- 3a. The user wants to change the reading speed of the system.
 - 3a1. The use case returns to the main flow.
-
-

Use Case Scenario:

1. Listening a page :

Goal in context: The user will listen to the system and know the contents of the page.
Scope: System.
Pre-condition: User will place the book correctly on the system.
Success end condition: User will know the content of the page.
Failed end condition: User will not know the content of the page.
Primary actor: User.

Main Success Scenario:

1. System scans the page correctly.
2. System reads the page out loud.
3. Reading speed is right for the user.
4. When the end of page is being reached, reading ends.

Extensions:

- 3a. Reading speed is not right for the user.
 - 3a1. Returns to "setting reading speed" case.
 - 3a2. Return to main scenario.

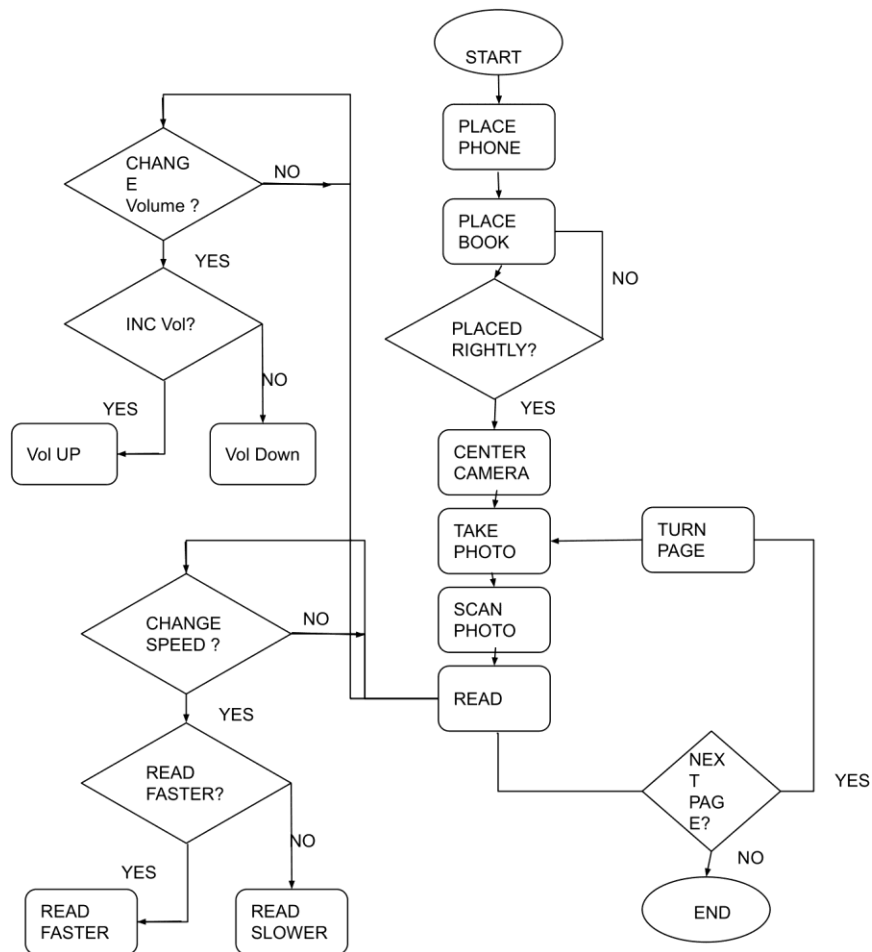
2. Wrong book placement:

Goal in context: The user will place the book in wrong direction.
Scope: System.
Pre-condition: User will not place the book correctly on the system.
Success end condition: User will know the content of the page.
Failed end condition: User will not know the content of the page.
Primary actor: User.

Main Success Scenario:

1. System doesn't scan the page correctly.
2. System calls "Wrong book placement" use case.

2.2.3 Data Flow Diagram (System Flowchart)



2.3 User Characteristics

2.3.1 Visually Impaired People:

Mainly visually impaired people will use this. They will use this to listen to the readings of books that they want to read but cannot due to their circumstances. The book reader will read the book for them and the generated audio will be what actually helps the blind people with their reading.

2.4 Non – Functional Requirements

2.4.1 Performance Requirements

1. Internet/Server Connection:

Our app sends API request to the server to get the generated audio from the captured image. For this reason, not only do we need internet connection, the faster and more stable the connection is, the better.

Input: API request from the phone to the server.

Output: Getting the generated audio from the server to the phone.

Testing Type: Unit testing because we have tested individual components of our tool.

Criteria Assessment: Usability and response time.

2. Taking Usable Pictures of the Page:

The captured image must be very sharp and usable because it will be only input when the server tries to process the image and generate audio from the image. If this image isn't usable, nothing will be accurate.

Input: Pressing the capture button.

Output: To capture a clear and usable image of the relevant page of the book.

Testing Type: Integration testing because we have tested to ensure connection between hardware and software.

Criteria Assessment: Usability and accuracy.

3. Correctness of the Generated Audio:

The server has to first crop all the individual words from the given image using a machine learning model. It then uses another model to create the audio files for each word. If there is any mistake in any of the steps here, the generated audio will not be useful to the user.

Input: Captured image of the page.

Output: To generate audio reading from the image.

Testing Type: Unit testing because we have tested individual components of our tool.

Criteria Assessment: Accuracy and response time.

2.4.2 Safety Requirements

For using the tool we need to follow some safety requirements. They are:

- 1) The stand that we use to place the book and phone is quite heavy. If it falls on a person's legs, it can cause injury.
- 2) Since the phone and book will be attached to the stand, if it falls, then the phone and the book can get damaged severely.
- 3) In our current implementation, the circuit is not protected very much using outer shells and covers, so careful handling is required to ensure device health.
- 4) The buttons should be pressed gently so that no harm is done.

2.4.3 Security Requirements

For software security we are using **https** connection over **http** connection for the API calls. As a result, the security and safety of the data and connection is maintained. Also, after each API call, we don't store any data in the server so that private data of the users are never stored and so there is no chance for privacy issues.

2.4.4 Software Quality Attributes

We have ensured some Software Quality Attributes. They are:

- 1) Usability and response time in internet/server connection.
- 2) Usability and accuracy in taking usable pictures of the page.
- 3) Accuracy and response time in correctness of the generated audio.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interface

- a. **Taking Photo:** The user takes the photo by pressing the top portion in the app.
- b. **Uploading the Image:** To upload the captured image to the server for processing.
- c. **Starting the Reading:** This is one big button in the middle of the screen that allows the user to start listening to the generated audio file.

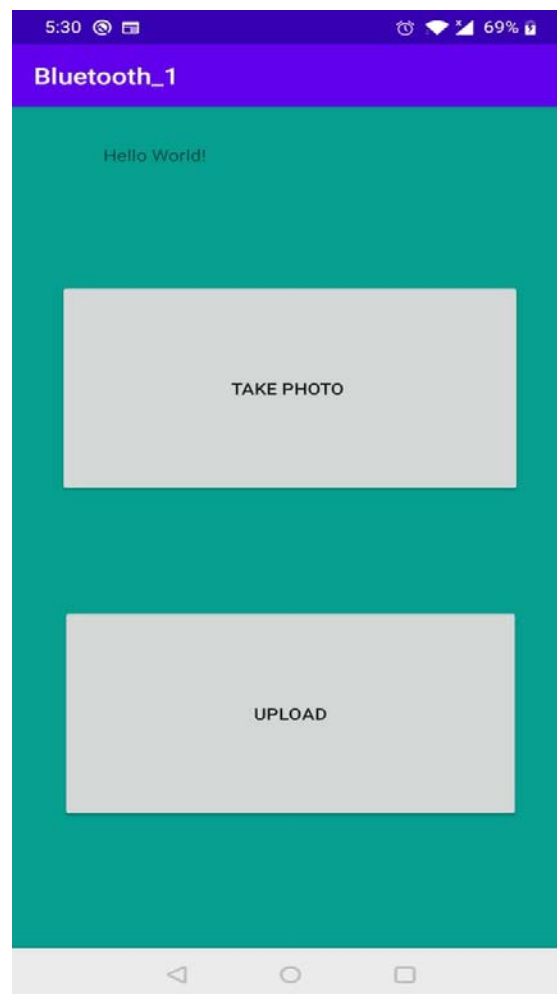


Fig: Taking photo and uploading interface

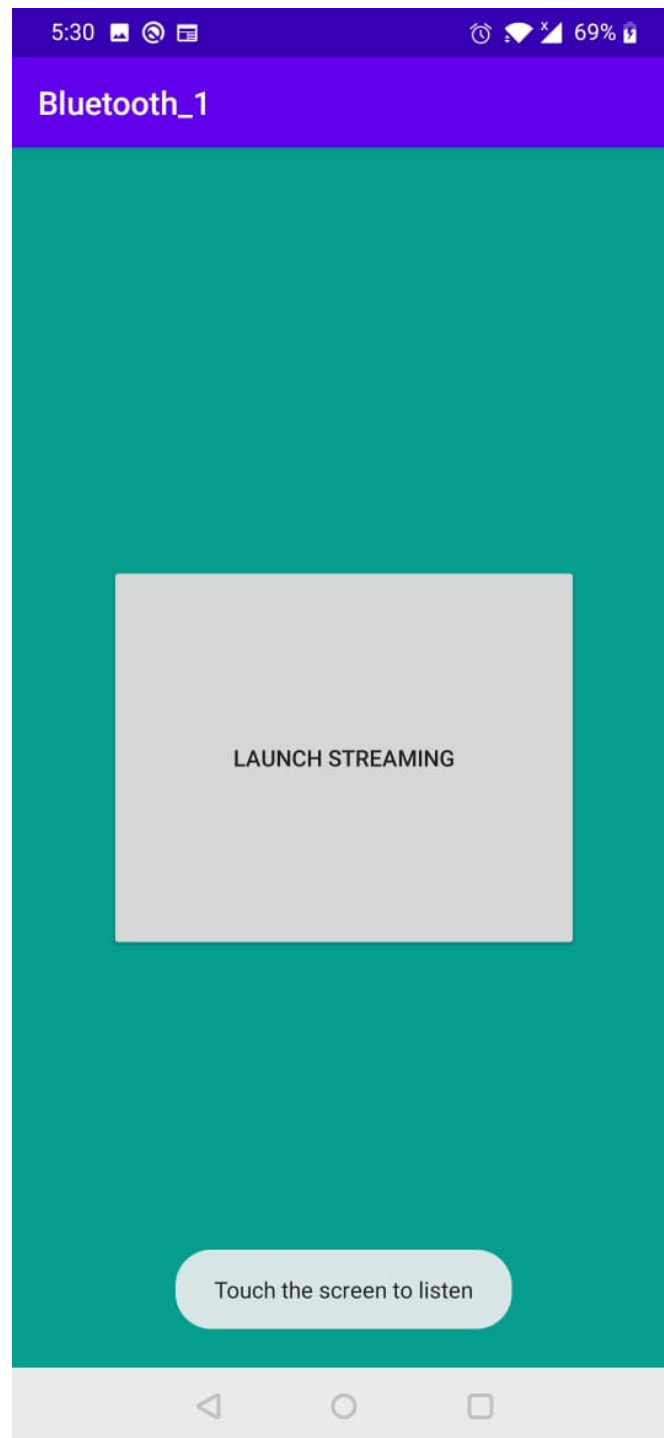


Fig: Starting the reading interface

3.1.2 Hardware Interface

Our prototype contains a board on which the book will be placed. Buttons are attached on it and inside it the Arduino board and other electrical equipment is attached. On top of this board is a stand that holds a head. A head is a holder for the phone. User will place their phone on the holder and will be able to adjust the stand vertically so that the camera can have a wider or narrower view. User can use this system in two ways:

- i. **Using the app in the phone:** User can navigate through the app using the app interface. For example, user opens the app, presses “take photo” to take a picture, then presses on “scan photo” to scan, presses “read” to listen to the audio of the text, presses speed increase and decrease button for speed change, uses volume buttons of the button to change volume.
- ii. **Using the buttons attached to the board:** User can also navigate through the app using the external buttons attached to the board. User presses the buttons to take picture, go to the next step, to increase and decrease volume and speed etc.

In summary, user according to their preference can operate completely using only their phone or only the external control buttons on the board.

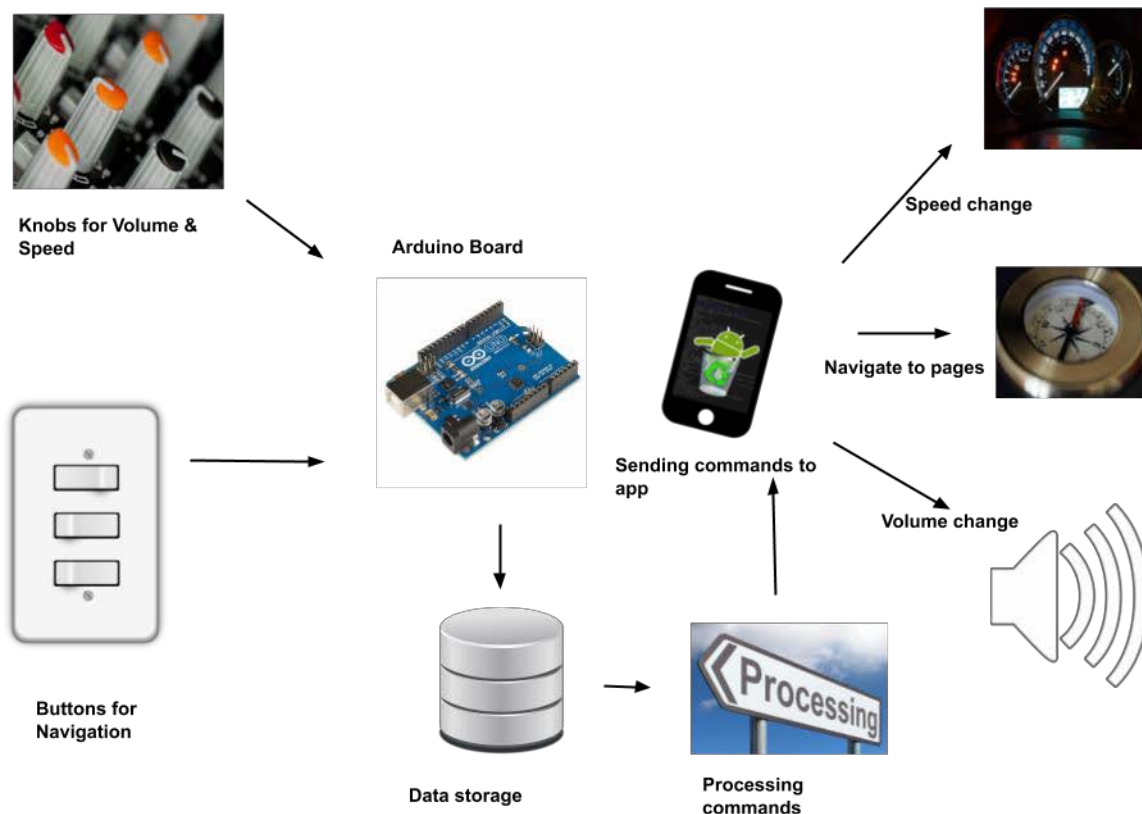


Fig: System Architecture

Inputs from volume, speed change and navigation are stored and processed in the Arduino board. The processed outputs are commands for volume, speed change, reading, scanning etc. These commands are sent to the android application through the Bluetooth module. Commands are executed and result in the desired change of volume, speed, navigation etc.

3.1.3 Software Interface

For software we have chosen an Android OS based application with heavy backend processes that run in the server. The local server is created using ngrock. The application is for android version 4+.

3.1.4 Communication Interface

For connecting hardware part of the tool with the mobile application we have used Bluetooth as it is available with all the mobile devices. Internet connectivity is also required for the much required API calls.

3.2 Design Constraints

3.2.1 Hardware Constraints

Our tool can only read a single page at a time. There is no automatic page-turning system. For this reason, the user must turn the pages by hand. After each page has been read, this action has to be repeated. If the book or phone alignment isn't correct there is no error detection or correction. The user has to use his feeling of touch, intuition, outside help etc.

3.2.2 Software Constraints

If there is any problem with the captured image, the software cannot detect it automatically. The user has to listen to the result and deduct whether or not the captured image was good enough. Because we need to send the image to the server and do some heavy processing before sending it back to the phone, total time required is a bit high. Even the internet speed is a contributing factor to this problem. Also because of our dataset being handmade and incomplete, our OCR performance is not very good which leaves room for improvement. Also, our TTS module can't properly detect punctuation which is why our system can't provide optimum listening experience.