Design and Analysis of a System to Read Braille with Remote Cloud Architecture

Rao Krishna Virinchi*, Paulson K Antony†, Shreyaa Saravanan*, Dhakshain Balaji V‡, Vaishnavi Suresh Krishnan*,
Edith Susanna Andrews*, Chris Joy Kokkat*, Melvin Noel S†, Sreejeeth Vijay TK†, Harshit Goyal‡ and C. Vijayalakshmi §
*School of Computer Science and Engineering
Vellore Institute of Technology, Chennai
Chennai, India
†School of Electronics Engineering
Vellore Institute of Technology, Chennai
Chennai, India
‡School of Mechanical Engineering
Vellore Institute of Technology, Chennai
Chennai, India
§Central University of Tamil Nadu
Thiruvarur, India
¶ Email: raokrishna.virinchi2017@vitstudent.ac.in, paulsonk.antony2017@vitstudent.ac.in,
shreyaa.saravanan2018@vitstudent.ac.in, dhakshain.balaji2018@vitstudent.ac.in, s.vaishnavi2018@vitsudent.ac.in,
edithsusanna.andrews2017@vitstudent.ac.in, chrisjoy.kokkat2017@vitstudent.ac.in, melvinnoel.s2017@vitstudent.ac.in,
tk.sreejeethvijay2017@vitstudent.ac.in, harshit.goyal2017@vitstudent.ac.in, vijayalakshmi@cutn.ac.in

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342520

***Abstract*—Recognizing that physical Braille books have a high cost of production and purchase, using a refreshable Braille display is a viable solution. This paper discusses the design and architecture of a system which grants the visually impaired access to a number of books on a refreshable Braille display by using the huge number of existing e-books. Understanding that almost all the books that are available to a person who isn't visually impaired, the design described in the paper takes e-books and converts them into Braille that is directly rendered on the refreshable Braille display's interface with solenoid based actuation, thus eliminating the extra costs incurred in producing and purchasing specially printed Braille books. The design describes a 3D printable physical device which uses a NodeMCU for its computation and has internet connectivity to utilize the support of a web app and cloud servers whose architecture and functionalities are also discussed.**

***Keywords*—assistive technology, refreshable braille display, solenoid, consumer electronics, visually impaired, cloud server, software hardware codesign, actuators, NodeMCU, 3D printing**

AMS Classification : 68M07, 68P25, 68P30

I. Introduction

Braille, invented by Louis Braille in 1824 [1], was effectively universalized by United Nations Educational, Scientific and Cultural Organization (UNESCO) in 1950. Since then, there have been various technological and cultural developments in order to better accommodate the needs and requirements of the blind. Braille has been effective in allowing the visually impaired to read and learn [2]. Conventional braille mechanisms use 6-dot mode [3], which allows for the representation of 64 characters, each made up of one to six raised dots arranged in a cell. Before the invention of the refreshable Braille tablets, Braille typewriters such as the Hall-Braillewriter [4] and the Perkins Brailler [5] were used to emboss paper with raised dots. Today, the introduction of refreshable braille tablets has facilitated reading, writing, and learning for the visually impaired.

According to the World Health Organization (WHO), the total visually impaired population is estimated at 285 million, of which 39 million people are blind [6]. Studies have shown unemployment rates among disabled people are higher than people with no disability. We also know that blind people who are Braille illiterate face a greater risk of unemployment [7]. A major reason for Braille illiteracy is that various countries face a shortage of Braille educators [8] and books [9]. Moreover, we learn that only 10% of blind adults know how to read Braille and only 10% of blind children learn Braille [10]. This poses a great challenge to the employability and upliftment of the blind community.

This paper proposes a solution to the existing problem by introducing a novel device that enables blind users to read and learn Braille effectively, by means of a refreshable tablet with a cloud-based remote architecture. A few factors were kept in mind while designing this device; the device facilitates searching for pages and words, solving the issue of time disability. The device is also user-compatible and friendly, while equipped with state-of-the-art technology like solenoid actuators on the hardware side and cloud-based servers on the software end.

II. Braille Conversion Forms

The device allows the user to toggle between two modes of reading, the 6-dot mode and the 8-dot mode [11]. These

modes refer to the braille cell, with the 6-dot mode enabling the reader to use 6-dot notation and the 8-dot mode enabling the reader to use 8-dot notation. With the conventional 6-dot Braille, only 64 characters are included, which represent all possible combinations [12]. However, with the extended 8 dot Braille convention, 256 characters can be represented, including words like “and”, “the” and “but”.

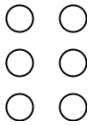

Fig. 1. 6-dot Braille Cell

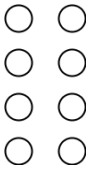

Fig. 2. 8-dot Braille Cell

The device supports both modes through a simple switch that is present on the user interface. The braille to text conversion is enabled as follows:

1) A Python file stores the character and its corresponding notation in Braille.
2) The conversion of text to Braille happens in real-time, according to the line or page of the book that the user is referring to.
3) The translated Braille is then communicated to the actuators on the device. A ‘1’ will signal the corresponding dot on the Braille cell to be raised and a ‘0’ will not raise it.

III. EPUB-to-Text Translation

An electronic publication, or an EPUB, allows the recalibration of text according to screen size and layout, with the capacity of adding bookmarks and highlights. The latest standard version, EPUB3, is based on Hypertext Markup Language (HTML) 5 [13].

An EPUB file is a zipped folder containing metadata, fonts, images, styles, and text grouped into sections. Using the ebooklib library for Python, everything except the text is removed. From here, the BeautifulSoup library is used to parse the HTML files, leaving only the plain text. This is accomplished by a small group of custom-designed modules. The first module removes HTML tags like <p>, <header>, <script>, <meta>and <noscript>. A different module is responsible for targeting the EPUB file and extracting the chapters as HTML files. Another module takes each chapter and converts it into text by using regular expressions. Each cleaned chapter is appended to a list that is finally printed and stored as a .txt file. This file is later converted to either 6-dot or 8-dot Braille.

Apart from EPUB files, books are published in a variety of digital formats. By substituting this EPUB-to-text module call with a different module that converts a different format like a portable document format (PDF) to the text, the device can be used to read electronic books (e-books) available in any format.

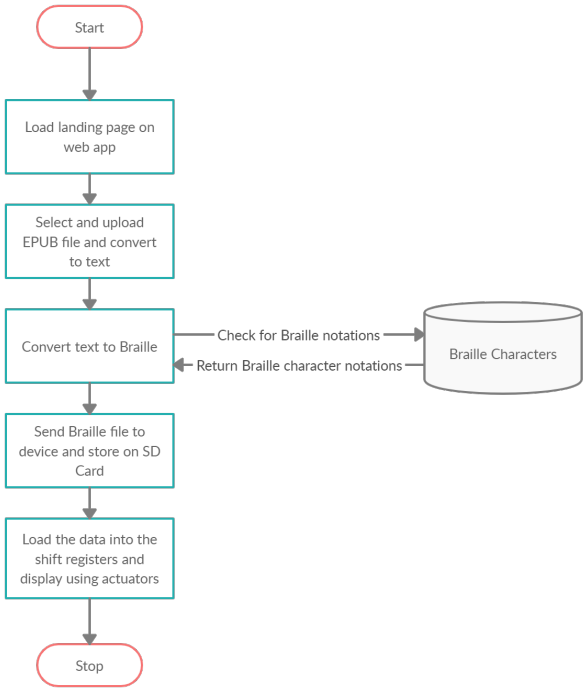

Fig. 3. Conversion Process

IV. Architecture

The device follows a modular architectural pattern. On a macro level, the architecture may be divided into three parts: the local hardware unit, the web app, and the remote database (Fig. 3). The local hardware unit consists of a local computing unit and a local storage, along with transmitters and actuators. The web app is hosted on a suitable cloud server. To reduce possible heating problems in the device unit, the computation is distributed such that all the performance-intensive computation is done on the cloud server, with only essential computations being done on the device’s local computing unit.

A. Remote Database

A not only structured query language (NoSQL) database such as MongoDB is used to store all the data required for the system to function. For the purposes of this system, a replication factor of three was chosen to achieve a balance between data robustness and storage usage. The database can be divided into two sections, each represented by a separate collection on the database, one for e-book data, and one for user data. The e-book collection has all the e-books uploaded by device users. By indexing all the e-book files in a separate collection and linking them to users through a foreign key mechanism, the system optimizes the amount of space taken, as only one copy of the e-book needs to be stored on the database, despite multiple users reading it. The User collection stores all the data pertaining to the users, with login credentials being stored in an encrypted format. Apart from

the login details, the User collection also holds the following data: indexes of the e-books owned by the user (linked to the e-book collection using a foreign key), the page numbers on each book, the bookmarks made by the user on a particular book, and the email of the user.

B. Web Application

The web application, or web app for short, allows the user to perform less frequent actions, such as pairing a device unit to an account, resetting login credentials, uploading books, etc. The web app is divided into 2 parts: the frontend, and the backend. The frontend has all the user navigation menus and controls necessary for the user to interact with the device. The backend has a cloud server, where most of the intensive computing is done. This is where the e-book file is converted into a Braille compatible format. The logic for when to make remote calls to the database, framing of the database manipulation calls, and storage management logic are all hosted here.

C. User Interface and Experience

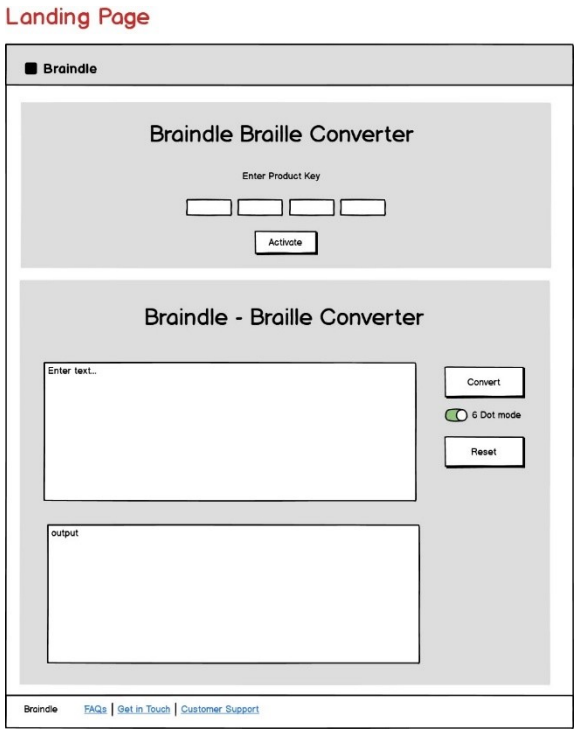

Fig. 4. Landing page of User Interface (UI)

Fig. 4 shows the landing page of the web application. It allows the user to activate their device using the product key. Furthermore, it allows a simple text to braille conversion in 6 and 8-dot modes with a cap on the word limit. Upon account activation, the user is redirected to the homepage. The homepage is where the main file conversion happens. It allows the user to upload a text file, convert it to braille and open it on their refreshable braille device. The option ”Open in Device” activates the device and the current page (what is being viewed) is opened. The users can access other books in their library from here or go to the homepage through the home link.

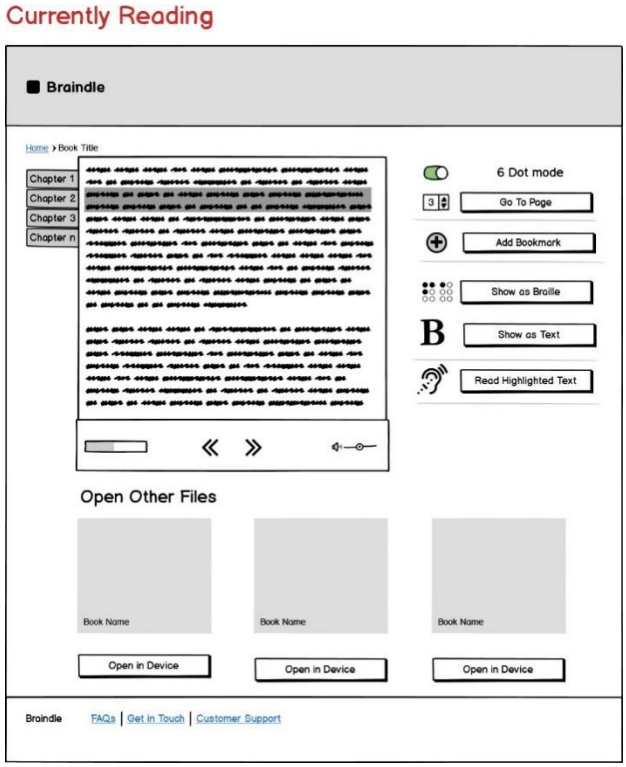

Fig. 5. Currently Reading page of UI

As soon as a file is opened on the hardware device, the “Currently Reading” page is loaded on the web app, as shown in Fig. 5. The original text is viewed here. The highlighted part of the text indicates the currently displayed text on the device. The progress of the file read can be viewed on the grey progress bar below. The arrows are used to turn pages, which is also possible via the hardware device. In addition to this, the user can skip pages using the “Go To Page” button, add bookmarks, and shift between braille and text display on the web app. The file is opened on the device in 8-dot mode by default. The mode switch can be used to toggle the display mode. Apart from these functionalities, in case the converter fails to recognize a foreign character, the “Read Highlighted Text” option can be used to simply read out the corresponding text.

D. Role of the Physical Unit

The NodeMCU communicates directly with the backend of the web app, which in turn makes calls to the remote database and transfers data to the NodeMCU as needed. The NodeMCU functions both as a local computing unit and as a local storage unit. The local storage of the NodeMCU is where the e-book that is currently being read is stored in a Braille compatible format, along with a line pointer that holds the location of the exact line that the user is reading.

The line pointer is stored in the permanent storage of the NodeMCU. This lets the user pick up from where they left off even after the device is restarted. The actuation logic lets the NodeMCU decide which Braille cells to actuate depending on the software representation of the letters. The local computing unit on the NodeMCU also handles the connection and pairing logic between the hardware and the web app. Most of the processing will be done on the web app. When uploaded to the web app, the e-book is converted into a format which is understandable to the device. This converted file is then transmitted to the device where it will be stored persistently. The structure consists of blocks of characters equal to the number of cells in the device. Each block is mapped onto the actual text. This provides for a navigation system that exists in the device as well as in the web app.

V. Internet Connectivity Protocol

Letting the device connect to a wireless fidelity network (Wi-Fi) which is preferable to the consumer is a major design hurdle. In order to tackle this, the device uses a specifically designed protocol described in Fig. 6. Every device will have a randomly generated and pre-configured Wi-Fi Server Set Identifier (SSID) and password, which will be attached to the device's body through a sticker. When the device's pairing mode is activated, it searches through the available Wi-Fi networks in the area for one with the same SSID. Once a network is found, it attempts to connect to the network using its preconfigured password. The user can set up the Wi-Fi connection easily using a mobile hotspot.

Once a successful connection is established, the device connects to the web server, and sends a message to the server, relaying the unique ID of the device. When the server receives this message, it adds the device ID to a list of actively pairing devices. This entry is automatically deleted after 20 minutes. Once the entry is added to the list of actively pairing devices, the consumer logs in using the web app, from where they will be able to 'claim' their device by entering their device ID.

When a claim is issued for a particular device ID, the server checks the actively pairing device list for an entry with the same device ID as the one entered. If there is an entry that matches, the consumer is prompted to send an accept request on the web app to the device. An accept request is a request sent to the device from the server, which sets the device into 'accept' state for the next 30 seconds (while still maintaining pairing mode). If the consumer presses and holds the confirm button for 5 seconds, it will accept the claim issued, and pair the device. The server checks if an accept request has been issued to the device in the past 30 seconds, if so, it allows the user to send an accept request, or makes the user wait. A user can send only 5 accept requests from a particular IP address within 20 minutes. This prevents fraudulent claims to the device from being accepted by accident. Once the device has been successfully paired to a consumer's account, the consumer can use the web app to add a list of Wi-Fi SSIDs and passwords which the device will use to connect to the Wi-Fi the next time the device checks for Wi-Fi.

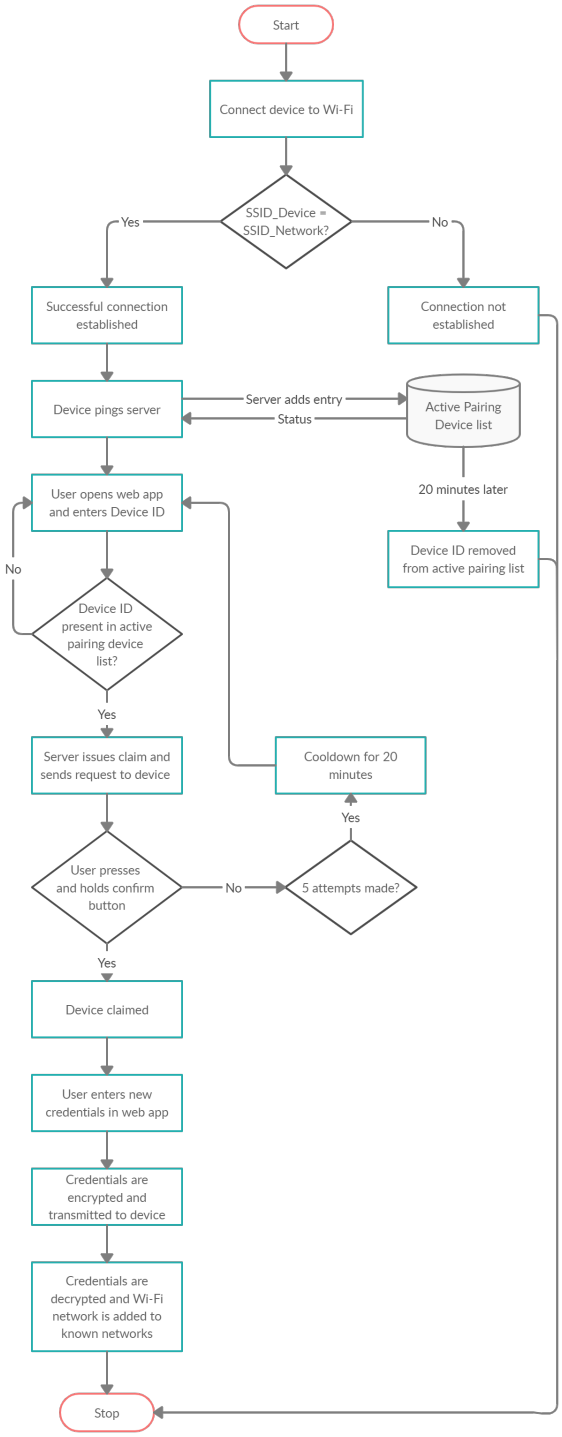

Fig. 6. Connection Protocol Design

When a new combination of SSIDs and passwords is entered on the web app, this information is encrypted using the AES algorithm [14] and transmitted to the device, which decrypts the data and adds it to its memory, so that the device can connect to the consumer's choice of Wi-Fi when it is performing the rest of its functionality.

VI. Hardware

A. Character Pin Assembly

The goal of this device is to be affordable and this serves as the first prerequisite for implementing the pin mechanism. Therefore, the design consists of commercially available or easily manufacturable parts. 3D Printing expands the possibilities here [15]. The basic pin assembly consists of a three dimensional (3D) printed body around which the solenoid coil is wound.

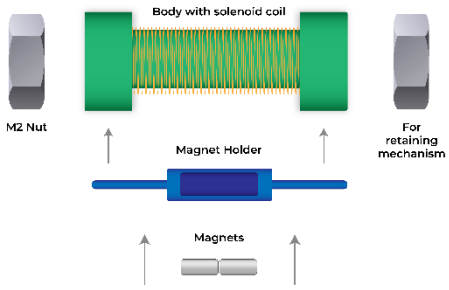

Fig. 7. Pin Mechanism Using Solenoid

The solenoid core houses the actual pin, which is hollow inside to make room for magnets. The body is capped shut by two M2 nuts. When the current is applied to the coil, the magnet moves in a linear direction due to Faraday's law of electromagnetism. The M2 nut restricts the range of linear motion and so the pin stays in place when the signal is applied. This configuration is illustrated in Fig. 7. Each character consists of 8 pins. This translates to 8 different signals for each character to operate. The device is an 18-character display. Therefore, 144 signal connections are required to run the whole display. This array of input signals is fed by shift registers. The device uses an array of 18 74HC595N 8-bit serial-in, parallel-out shift registers. The serial input to the shift registers is given by the NodeMCU. This is a practical and simple approach to the problem as compared to other solutions involving magnetic actuation mechanisms [16].

B. Processing

The actuation signals are provided by the NodeMCU, which in turn receives it from the web app, which sends Braille information in the form of bits. Each bit corresponds to a pin on the display. It is connected to the web app via Wi-Fi. It is powered by a regulated 5-volt supply from the power supply circuit. The NodeMCU also handles the navigation buttons on the device. The 4mb of onboard flash memory stores the bits for several characters at once to reduce latency and the processing power required to fetch characters every time the user moves to a new section in the document.

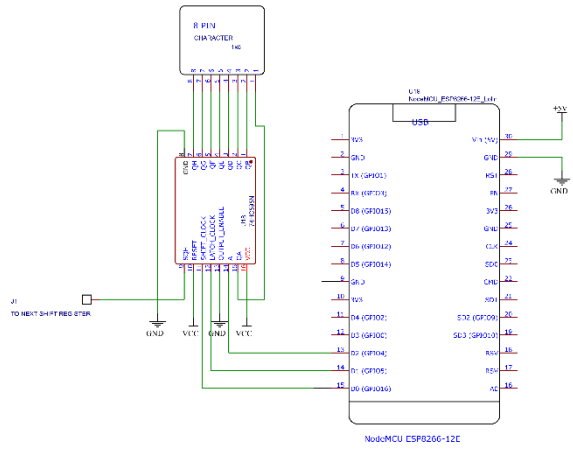

Fig. 8. Connecting NodeMCU to Shift Registers

C. Power Supply

The device receives power from a 5V/3A wall adapter which is connected to the device via a 5-volt regulator circuit. This circuit is based on the LM7805 regulator IC and is configured as shown in Fig. 9. This 5-volt regulated line powers the NodeMCU as well as the array of shift registers required to toggle the pins on the Braille display.

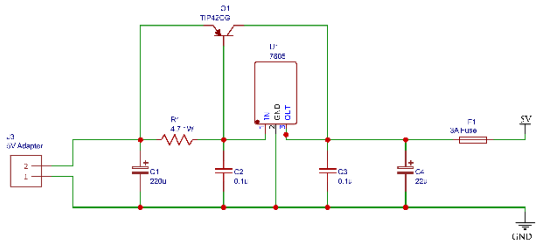

Fig. 9. Power Supply Circuit Diagram

D. 3D Printed Enclosure

There are two navigation buttons on the device, one on either side of the character array. Three miscellaneous buttons are also provided, which serve different functions that aid the user to navigate efficiently through the document. The buttons and the enclosure are manufactured using 3D Printing.

VII. Results and Discussion

Assistive technology has seen a rise in production in the last decade [17]. As of February 1, 2018, the World Health

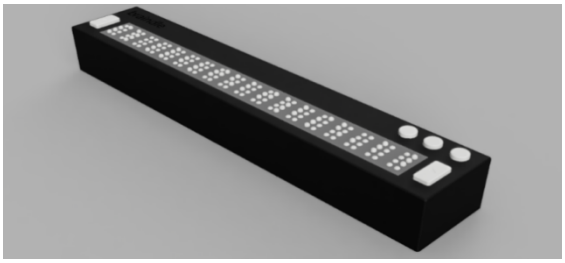

Fig. 10. 3D Model of Device

Organization (WHO) called for awareness to be raised and encouraged the integration of cost-effective assistive products to reduce disability burden. Since then, several companies and start-ups have been working towards developing technologies to assist blind people, with respect to reading. At Massachusetts Institute of Technology (MIT), [18] a team called Tactile developed a prototype for a Braille translator, which translates printed text into raised-dot language. However, the current version is limited to the number of characters it can translate and display, as only 6 characters can be displayed at once, which makes reading very sedate. The system and device proposed in this paper aims to solve the underlying problem of current technology as the device is supported by cloud architecture, with security measures, such as encryption and decryption. The modularity and the unique assembly accounts for the innovation of the system and device. The device also solves time disability, user inefficiency and eases implementation.

REFERENCES

[1] J. Jiménez, J. Olea, J. Torres, I. Alonso, D. Harder, and K. Fischer, "Biography of Louis Braille and Invention of the Braille Alphabet," *Survey of Ophthalmology*, vol. 54, no. 1, pp. 142–149, 2009.

[2] R. Massof, "The role of Braille in the literacy of blind and visually impaired children," *Archives of ophthalmology*, vol. 127, no. 11, pp. 1530–1531, 2009.

[3] C. Moore and I. Murray, "An electronic design of a low cost Braille typewriter," *The Seventh Australian and New Zealand Intelligent Information Systems Conference*, 2001, Perth, Western Australia, 2001, pp. 153-157, doi: 10.1109/ANZIIS.2001.974067.

[4] J. B Curtis, "The inventor of the Hall-Braillewriter," *Journal of Visual Impairment & Blindness*, vol. 21, no.1, pp. 5–5, 1927.

[5] J. Seymour-Ford, "Story of the Perkins Brailler," *Perkins School for the Blind*, Apr-2002. [Online]. Available: https://www.perkins.org/assets/downloads/research/story-of-brailler-11-17-09.pdf. [Accessed: 20-Nov-2020].

[6] S. Resnikoff et al., "Global data on visual impairment in the year 2002," *Bulletin of the World Health Organization*, vol. 82, no. 11, pp. 844–851, 2004.

[7] R. Ryles, "The impact of Braille reading skills on employment, income, education, and reading habits," *Journal of Visual Impairment & Blindness*, vol. 90, no. 3, pp. 219–226, 1996.

[8] "There's a national shortage of Braille teachers and the situation is dire," *Good Morning America*. [Online]. Available: https://www.goodmorningamerica.com/living/story/national-shortage-braille-teachers-situation-dire-66013086. [Accessed: 20-Aug-2020].

[9] "Visually challenged students face shortage of braille books, reading assistants," *The Indian Express*, 06-Jan-2016. [Online]. Available: https://indianexpress.com/article/education/visually-challenged-students-face-shortage-of-braille-books-reading-assistants/. [Accessed: 20-Aug-2020].

[10] "The Braille Literacy Crisis in America", *National Federation of the Blind*, 2009

[11] A. K. Garg, "Braille-8 — The unified braille Unicode system: Presenting an ideal unified system around 8-dot Braille Unicode for the braille users world-over," *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bangalore, 2016, pp. 1-6. doi: 10.1109/ANTS.2016.7947839

[12] V. Argyropoulos et al., "Refreshable Braille displays and reading fluency: A pilot study in individuals with blindness," *Education and Information Technologies*, pp. 1-18, 2020.

[13] "EPUB Publications 3.0.1," 26-Jun-2014. [Online]. Available: http://idpf.org/epub/301/spec/epub-publications-20140626.html. [Accessed: 20-Aug-2020].

[14] A. Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data", *Cryptography and Network Security*, 2017.

[15] G. R. Peñaloza-Mendoza, A. A. Sánchez-Rodríguez, P. Y. Melgoza-Rivera and S. Espinosa-Hurtado, "Design of an electronic prototype to teach Braille," *Health and Technology*, vol. 10 no. 2, pp. 411-416, 2020.

[16] D. Leonardis, C. Loconsole, & A. Frisoli, "A passive and scalable magnetic mechanism for Braille cursor, an innovative refreshable Braille display," *Meccanica*, pp. 1-15, 2020.

[17] World Health Organization. Regional Office for the Eastern Mediterranean, "Improving access to assistive technology," 2016.

[18] "Tactile," *MIT Innovation Initiative*, 14-Nov-2017. [Accessed: 20-Aug-2020. [Online]. Available: https://innovation.mit.edu/pathway-post/tactile/. [Accessed: 20-Aug-2020].