Software Requirements Specification

for

Chashm

Version 1.0 approved

Prepared by

Ali Amir Khawaja 344200

Nirmal Rai 337582

Hashim Mahmood 334973

National University of Sciences and Technology (NUST)

November 20, 2022

Table of Contents

Table of Contents ii

1. Introduction 1

The Project we are working on is about to teach basic learning about native language (URDU) and Basic Calculations (MATH) to visually impaired children and teenagers. 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 5

2.6 Assumptions and Dependencies 6

3. External Interface Requirements 7

3.1 User Interfaces 7

3.2 Hardware Interfaces 8

3.3 Software Interfaces 8

3.4 Communications Interfaces 8

4. System Features 9

4.1 Mobile App 9

4.2 Input/Output through BKD 9

4.3 Connection to BKD 9

4.4 Navigation 10

4.5 Student Login 10

4.6 Take Lessons 10

4.7 Take Tests 11

4.8 Save Student Progress 11

4.9 Display Results 12

5. Other Nonfunctional Requirements 12

5.1 Performance Requirements 12

5.2 Portability Requirements 12

5.3 Scalability Requirements 12

5.4 Security Requirements 12

5.5 Software Quality Attributes 13

6. Other Requirements 13

# Introduction

# The Project we are working on is about to teach basic learning about native language (URDU) and Basic Calculations (MATH) to visually impaired children and teenagers.

## Purpose

The main purpose behind this project was to facilitate with a flexible and powerful educational tool for visually impaired people who are unfamiliar with native language and do not know basic math which is important in our daily life, how to write or read it. So that they should be left out from this fast growing world. It would be tough task in start but not so difficult as people get used to it. We will do it by help of physically touch and sounds. Normally devices provided to teach these student is too costly and middle class family will not purchase it but this product

## Document Conventions

Normally words and terms used in our document are easy to understand and not too much technical terms are used in our description, but some of them are used like visually challenged, dim sighted or near-blind, All three have same meaning and referencing to same thing.

## Intended Audience and Reading Suggestions

Intended audience for this project are impaired children and teenagers age range between 4-16 who are just keen to learn basics of languages and math but couldn’t make to it due to visually challenges. Similarly this software can be very potent educational aid while teaching basics. The step-by-step explanations of concepts like spelling, grammar and punctuation and gain an understanding of how text is formatted on the page to make teaching these abstract topics easy and intuitive.

## Product Scope

Product Scope for this Assistant is to ease understanding of Basic Knowledge and to create a convenient and easy-to-use application for visually Impaired People trying to implement abstract learning concepts. By use of this tool, Person can intuitively listen the voice of sound plus there will be directions given how to write that words. Including some animations for guider and Progress will be mentioned at the end of each test or learning assignments.

## References

[Google](https://www.google.com/)

[American Foundation for Blind](https://www.afb.org/aw/2/3/15030)

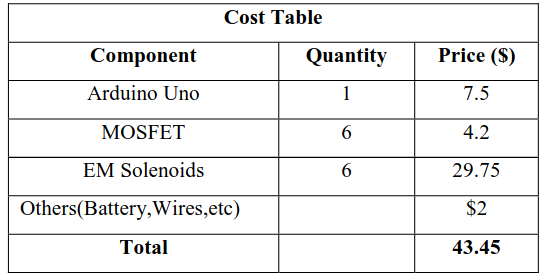
[Braille Bug](https://braillebug.org/)

# Overall Description

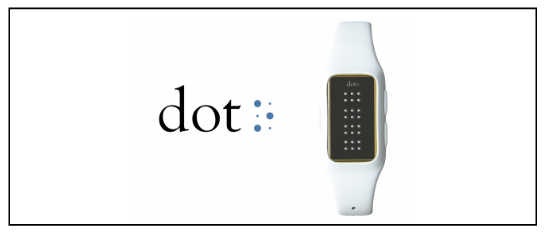
## Product Perspective

For the visually impaired, the process of understanding and learning is very difficult as they are unfortunately deprived of receiving visual information from the brain. Braille language is used for the visually impaired, so they can read and write.

Nowadays, advanced braille products are very costly. Unfortunately, these products can’t be easily purchased by individuals in developed or developing countries. Therefore, this document defines the description and requirements needed to create an affordable braille display for blind individuals. The device and application that is being developed, uses electromagnetic solenoids to move the small rods and dots in vertical positions out of the grid of six dots. These movements are enabled using solenoids, batteries, and Arduino. This device will be used to read and input alphabets within the system. This project helps to manufacture a product that is under $100, with additional features like portability as well as affordability.

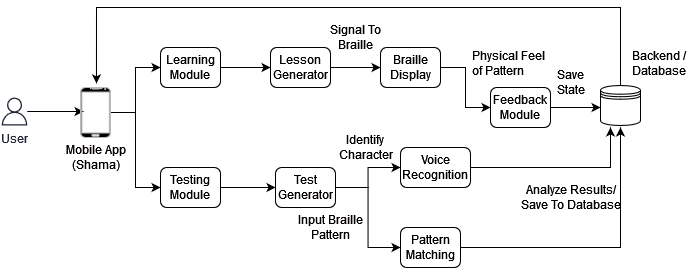
Fig.1 Table showing the cost of all components.

At present in the world, the only solution for the blind is to use external assistance. The only company that makes a product for visually impaired individuals is DOT,

Fig.2   A DOT Braille watch.

The watch provided by DOT has a current cost of above $300 which is not affordable for the majority of people. The statistics show the majority of 90% of visually impaired people are currently residents of developing or under-developed countries. Therefore, the project will greatly reduce the expensive burden because of being cost-effective.

## Product Functions



## User Classes and Characteristics

There will be a learning module and a testing module. These modules will contain two main classes student and parent classes.

Student class:

* Create an account by entering profile details such as name, username, password etc.
* Input and output functions take user input and communicate using the BKD.

Teacher class:

* Create assignments using the interface.
* Function to add or delete lessons for that particular week.
* Function to take feedback for students related to lessons, assessments and teaching methods.

Lesson class:

* Generate a lesson according to the knowledge level of the student, deciding from its academic details.
* Function to send appropriate signals to BKD to display the pattern.
* Function to take feedback to ensure that students understand the pattern corresponding to an alphabet in Urdu.

Testing class:

* Generate tests of random questions picked from the server/database.
* Take inputs from the students for each MCQ and add selections to a file.
* Once the student submits and finishes the test, analyse the file and print out the scores.
* Update the records for each student in the database.
* The results will be only viewed by the organizer of the assessment.

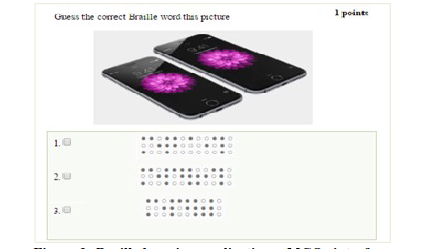


Fig 3. Braille learning application – MCQs interface

## Operating Environment

**The BKD will support the following software and hardware platforms:**

**Software:**

* The mobile application will operate on both android and ios platforms.
* It will support devices with Android version 6 or above.
* Devices with IOS 11 or above versions will also be catered.
* The activities performed by the students and records of assessments can be downloaded into a .csv file, which would be supported with online applications like Google sheets and offline applications like MS Excel.

**Hardware:**

* The BKD will be connected to the mobile application and external earphones for reading out the questions and the MCQ options for blind individuals and teaching activities.
* The external devices will be connected to the BKD using either wired or wireless transmission mediums.
* For wired, USB type A data transfer cable can connect BKD to mobile.
* For wireless, both Wifi and Bluetooth mediums can connect the BKD with Mobile App to receive and send responses.
* The earphones will be connected to the BKD using an Aux cable or through Bluetooth.

## Design and Implementation Constraints

**Three types of constraints were encountered:**

**Hardware:**

* A proper and appropriate voltage supply for the solenoids on the device has to be powered to work soothingly. The solenoids only work when two 5V voltage batteries are connected in parallel. If higher-powered batteries greater than 5V are connected, that results in severely damaging the solenoids in the system.
* A heating problem also occurred in the system. The solenoids overheated if the system remained enabled and the current supply was consistent. Continuous conduction of current resulted in the solenoids getting permanently damaged. Therefore, to avoid this problem, a timer of about 20 seconds was added if there was no response or activity from the user. After the timer goes up current supply automatically stops to the solenoids.
* In the system, users face difficulty inputting and identifying different characters through the braille keypad. The input problem indicated that some necessary upgrades to the keypad have to be implemented. For an English braille keypad experiment, users found it difficult to understand some symbol patterns. For example, an experiment was conducted with 12 people, and it was found that some faced difficulty to identify the characters “q”, “y”, and “0”.

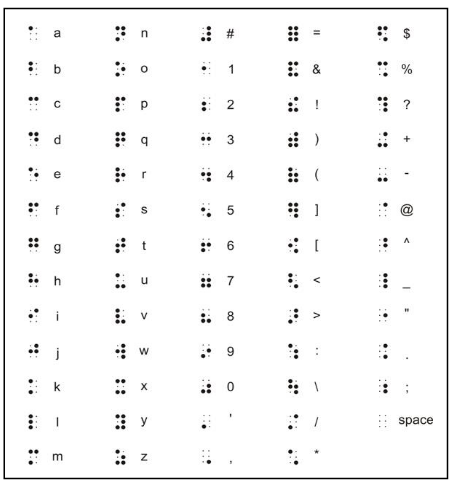
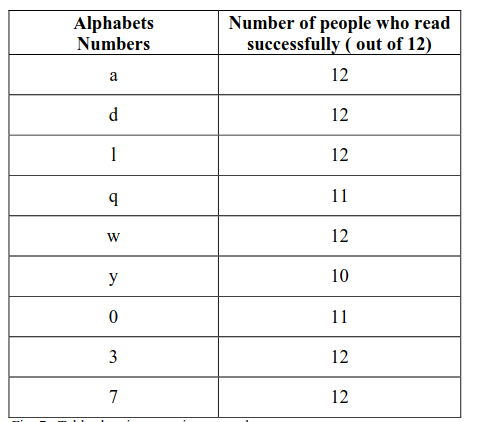


           Fig 4. Experiment Results. Fig 5. Input alphabet mapping to braille pattern

**Software:**

* The system has a fixed storage size allocated for each user. Each user has an equal portion of storage assigned to the total space available. Therefore, to add more recent files and data, some old unused files and content must be deleted or moved.
* Adding security features such as fingerprints and face id recognition for authorized access to user accounts would demand additional requirements.
* One would be adding hardware components like cameras and fingerprint sensors. Relevant programs and software would also have to be implemented and integrated into the system. The system would result in more processing delays and computation, which would increase the manufacturing cost for the project.

**Human Resource:**

* An assistant would need to operate the system and conduct the assessment activities using the device.
* The users must be trained to conduct activities using the braille device and mobile application.
* The students should also be taught relevant braille patterns, which are mapped corresponding to the Urdu language alphabet.

## Assumptions and Dependencies

**The following include the human and software dependencies:**

**Human Resources:**

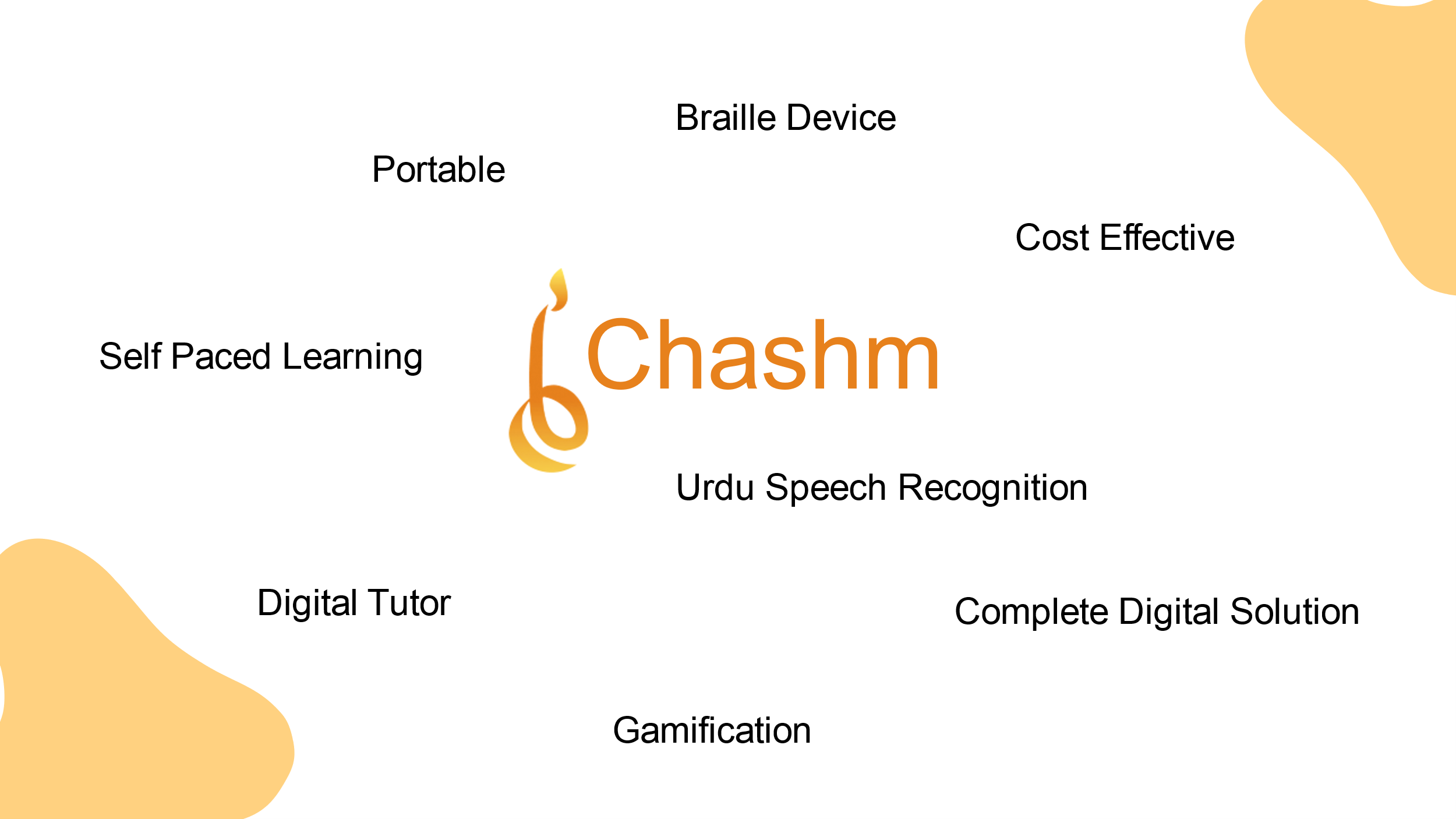
* The users who are operating and using the braille system are assumed to know how to read, write and understand the Urdu language.
* It is assumed that always whenever a person is operating the BKD for an activity or learning, a parent or assistant is assumed to be with the device.
* The mobile application and BKD are operated with stable and quality internet connection.

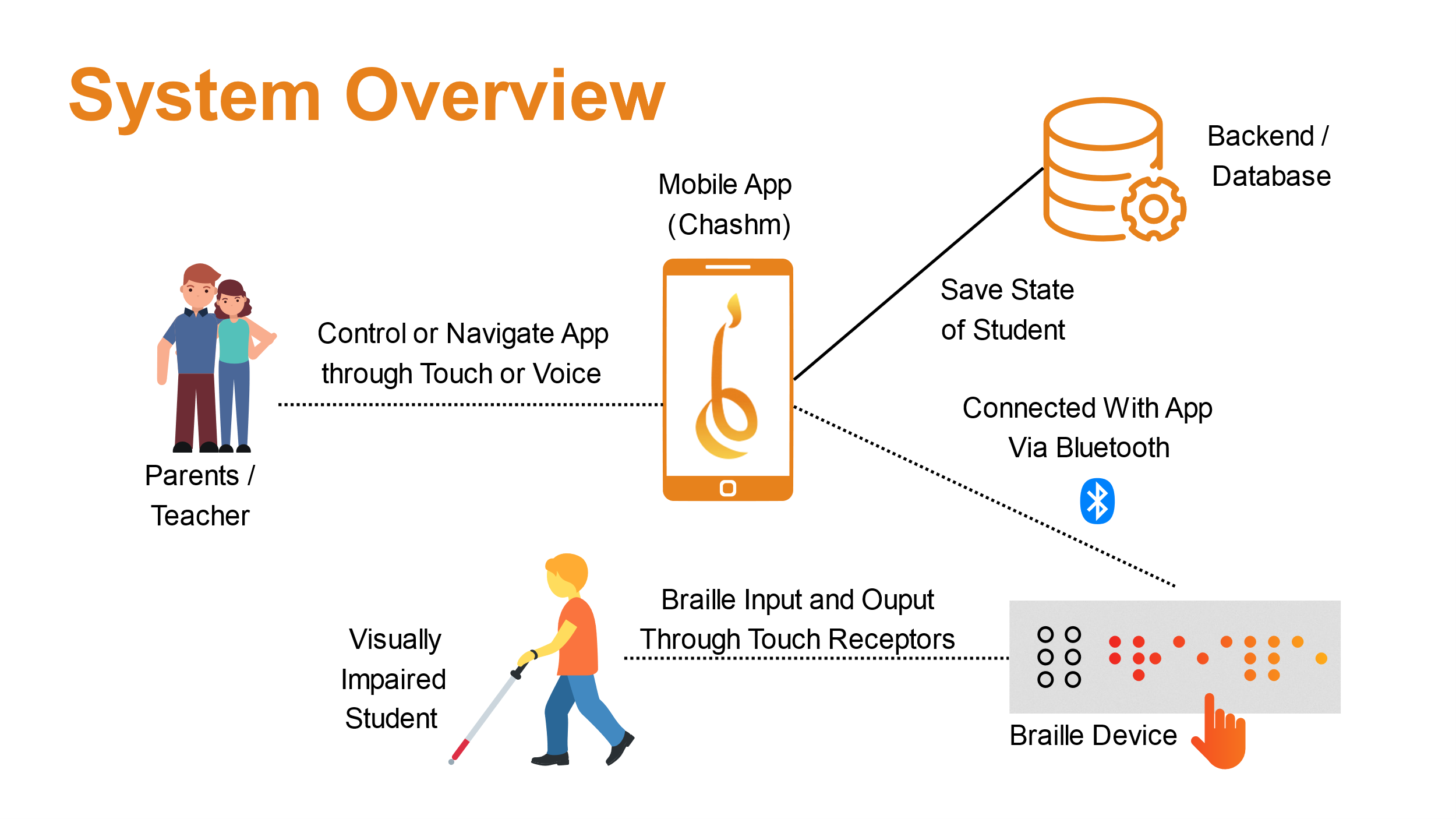
**Software:**

* Users (students or parents) should have mobile devices supporting Android version 6.0 or above.
* In the case of iOS handsets, the minimum version for the application to operate is ios 11 or above.
* The data of the test or assessments and the results of the activities are loaded from the Database on the server side. The working of the mobile app and BKD is dependent on the availability of the server.

# External Interface Requirements

## User Interfaces





User Interface for our assistant are mobile application or desktop application which we built using flutter or WRF.

Multiple buttons and hovering affects are been implementated on the system.

## Hardware Interfaces

The Hardware interfaces in our Assistant are Tablet or Mobile or it can be personal computer with minimum specifications to run our application are i3 3rd generation/AMD Ryzen 1000 series or above. The graphic card is not necessary but it will be cherry on the cake, minimum ram required is 2 GB. The recommended specifications i5 7th Generation/AMD Ryzen 3000 series or above, Nvidia GTX 750 Ti/AMD RX 380 or above, and 4 GB of RAM. The storage usage should be less than 200 MB. Hardware used in it is aurduino, keyboard etc.

## Software Interfaces

The program is designed to run for a minimum of Windows 7/Ubuntu 10.0/MacOS 9.0. The recommended specifications are Windows 10/Ubuntu 15.0/MaxOS 11. The technologies used in this assistant are flutter, python ,Autocad, tensorflow. There will be multiple tabs specific to each levels and student progress.

## Communications Interfaces

We do not have much communications between interfaces but there will be hardware connection with software technologies just like python or tensorflow or flutter will be interacting with Arduino and keyboards etc.

# System Features

## Mobile App

4.1.1 Description

A mobile app to house the dashboard features. It includes student login, student dashboard.

4.1.2 Stimulus/Response Sequences

App Launch is the starting point of the system.

4.1.3 Functional Requirements

REQ 4.1.1: The system should include an interface between the hardware and software.

REQ 4.1.2: The system should summarize all aspects of a student’s learning in one place.

REQ 4.1.3: The system should display test results.

REQ 4.1.4: The system should display student progress.

## Input/Output through BKD

4.2.1 Description

The BKD allows all sorts of input or output operations as needed. This includes:

* Reading braille from the keyboard
* Recording voice answer of student
* Navigating through buttons on BKD
* Sound output through speaker/attached headphones.

4.2.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection -> Take Lesson/Test -> I/O through BKD

4.2.3 Functional Requirements

REQ 4.2.1: The student should be able to read Braille from a hardware device.

REQ 4.2.2: Student’s actions/responses should be received by the device

REQ 4.2.3: The device should play system messages through built-in/external sound device.

## Connection to BKD

4.3.1 Description

The mobile app establishes a connection to the BKD before any operations. It can be wireless (via Bluetooth or over Wi-Fi) or wired (through USB cable). Connectivity is mandatory to move to next stage.

4.3.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection

4.3.3 Functional Requirements

REQ 4.3.1: The learner should be able to communicate with the central system.

## Navigation

4.4.1 Description

It includes all kinds of navigation to operate the system which includes app navigation, BKD navigation and system navigation.

* App can be navigated through touch and voice.
* BKD can be navigated through touch and voice
* System navigation is done by commands originating from app or from the BKD.

4.4.2 Stimulus/Response Sequences

It is required at every stage in all system features.

4.4.3 Functional Requirements

REQ 4.4.1: The learner should be able to access all features/options of the system.

REQ 4.4.2: The facilitator should be able to use the app via touch or voice.

REQ 4.4.3: The learner should be able to use the app via voice.

REQ 4.4.4: The facilitator should be able to operate the braille device controls.

## Student Login

4.5.1 Description

This allows the learner to login to the system by providing credentials. If the learner is new to the system, it allows him/her to register to get his unique credentials for logging in.

It also allows another learner to use the system, after the previous one logs out. This means that a separate system is not needed for every student.

4.5.2 Stimulus/Response Sequences

App Launch -> Login

4.5.3 Functional Requirements

REQ 4.5.1: The system should maintain a unique instance of every learner.

REQ 4.5.2: Multiple users should be able to use a single system.

## Take Lessons

4.6.1 Description

The learner can select a lesson from the lessons window on the app as planned by the teacher. The lessons window is queried from the database and lessons uploaded by the teacher are displayed on the app. The lessons window is accessible/readable through voice commands.

4.6.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection -> Take Lesson

4.6.3 Functional Requirements

REQ 4.6.1: The learner should be able to select a lesson through voice commands.

REQ 4.6.2: The facilitator should be able to select a lesson for the learner by touch or voice commands.

REQ 4.6.3: List of all available courses should be displayed.

REQ 4.6.4: Updates form teachers should be visible to the learner/facilitator.

## Take Tests

4.7.1 Description

The learner can select a test from the tests window on the app as planned by the teacher. The tests window is queried from the database and tests uploaded by the teacher are displayed on the app. The tests window is accessible/readable through voice commands.

4.7.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection -> Take Lesson

4.7.3 Functional Requirements

REQ 4.7.1: The learner should be able to select a test through voice commands.

REQ 4.7.2: The facilitator should be able to select a test for the learner by touch or voice commands.

REQ 4.7.3: List of all available tests should be displayed.

REQ 4.7.4: Updates form teachers should be visible to the learner/facilitator.

## Save Student Progress

4.8.1 Description

The progress of the learner (test or lesson) is saved and synced to the respective logged in user.

4.8.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection -> Take Lesson/Test -> I/O through BKD -> Save Student Progress

4.8.3 Functional Requirements

REQ 4.8.1: The learner’s responses must be recorded.

REQ 4.8.2: The learner’s lesson progress must be recorded.

REQ 4.8.3: The learner’s progress report should be synced accordingly.

## Display Results

4.9.1 Description

The results of a test can be displayed on the app and overall progress report can be downloaded as a CSV file

4.9.2 Stimulus/Response Sequences

App Launch -> Login -> BKD connection -> Take Lesson/Test -> I/O through BKD -> Save Student Progress -> View Results

4.9.3 Functional Requirements

REQ 4.9.1: The learner’s progress must be displayed.

REQ 4.9.2: A progress report should be maintained for all tests taken.

REQ 4.9.3: The progress report should be downloadable.

# Other Nonfunctional Requirements

## Performance Requirements

5.1.1 Voice navigation should provide instant feedback.

5.1.2 Test results should be visible right after completion of tests.

5.1.3 Student progress and results should be synced with student profile in real time.

## Portability Requirements

5.1.1 The system should be completely portable to be used anywhere anytime.

## Scalability Requirements

5.1.1 The system must employ a dynamic question bank that can be updated from the backend.

5.1.2 The system should be able to teach languages other than Urdu as well such as regional or international languages.

## Security Requirements

5.1.1 Student credentials must be secure from threats.

5.1.2 Test results must be visible to the candidate or his/her parents/teacher only.

## Software Quality Attributes

5.1.1 The mobile app must be cross-platform.

5.1.2 The braille device should be able to connect to any device that runs the mobile app.

5.1.3 The mobile app should integrate any registered and compatible braille device.

# Other Requirements

**Appendix A: Glossary**

* *BKD: Braille Keypad Device*

**Appendix B: To Be Determined List**

* *World Health Organization, “Fact Sheet: Visual impairment*
* *and visually impaired”, October 2013, Web, July 01, 2014*
* [*http://www.who.int/mediacentre/factsheets/fs282/en/*](http://www.who.int/mediacentre/factsheets/fs282/en/)

* *Xu, Cheng, et al. "Tactile display for the visually impaired*
* *using TeslaTouch." CHI'11 Extended Abstracts on Human*
* *Factors in Computing Systems. ACM, 2011.*
* *Nicolau, Hugo, et al. "HoliBraille: multipoint vibrotactile*
* *feedback on mobile devices." Proceedings of the 12th Web*
* *for All Conference. ACM, 2015.*
* *D. Kendrick. (July 2009). Product evaluation lowering the price of Braille.*
* *Retrieved from* [*http://www.afb.org/afbpress/pubnew.asp?DocID=aw100405*](http://www.afb.org/afbpress/pubnew.asp?DocID=aw100405)
* *S. Kelly Braille. Technology: What's New and Emerging? Retrieved from*
* [*http://www.visionaware.org/info/everyday-living/essential-skills/reading-*](http://www.visionaware.org/info/everyday-living/essential-skills/reading-)

[writing-and-vision-loss/braille-technology-7770/1235](http://www.visionaware.org/info/everyday-living/essential-skills/reading-)