`

Project report on,

Touchscreen Malayalam Text Input for the Blind

Submitted by

Jaison Jacob | Roll No: 126330008

Under the guidance of

Prof. Anirudha Joshi and Prof. G. V. Sreekumar

Industrial Design Center,

Indian Institute of Technology Bombay

Approval Sheet

Interaction Design

Degree Project entitled

Touchscreen Malayalam Text Input for the Blind

By: Jaison Jacob | 126330008

Is approved in the partial fulfillment of the requirements of the degree of Masters of Design in Interaction Design, December 2013

Guide ______________________________

Chairman ______________________________

External Examiner ______________________________

Internal Examiner ______________________________

Approval Sheet

Interaction Design

Degree Project entitled

Touchscreen Malayalam Text Input for the Blind

By: Jaison Jacob | 126330008

Is approved in the partial fulfillment of the requirements of the degree of Masters of Design in Interaction Design, December 2013

Guide ______________________________

Co-guide ______________________________

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all the principles of academic honesty and Integrity and have not misrepresented or fabricated or falsified any idea/data/fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature:

Jaison Jacob

Interaction Design, IDC, IIT Bombay

Roll no. 126330008

Mumbai , 24th July 2014

Acknowledgement

I take this opportunity to thank my guides, Prof. Anirudha Joshi and Prof. G V Sreekumar, for not only providing with their valuable guidance but also for constant motivation throughout the project.

I would like to thank Prof. Ravi Poovaiah, Prof. Pramod Khambate and Prof. Girish Dalvi for their valuable comments during the stage presentations. I would also like to thank all the users from NAB Trivandrum, NAB Worli and Barrierbreak Technologies who spent their valuable time in giving interviews for my user studies.

And last but not the least, I thank my family members, fellow IDCians and friends for their motivation.

Index

1. Abstract

This project involves the study of the structure of the Malayalam language, alphabets and the effective ways to represent them on a keyboard layout for making the text input process easier for the blind.

Absence of a proper input and feedback mechanisms are the reasons why blind people are not able to use the touchscreen mobile devices for inputting text in Indian languages. This project aims to develop an effective Mobile touchscreen keyboard for the blind users to type in Malayalam.

I studied different keyboards on Indian languages, learned how the blind users use the existing keyboards and designed keyboard layouts based on the insights derived from the study. Here vision is substituted with haptic and voice for the better Interaction between the blind user and the touchscreen, letting the user know where he is, giving two types of feedback to the user: What am I typing, what have I typed.

I tried alternatives that are static, dynamic, circular as well as rectangular and these layouts were theoretically tested by comparing with each other and also with the existing Inscript keyboards that some blind users are experts with.

The input is recorded from a touchscreen keyboard, stored as a Unicode and the output will be a display, haptic or a voice feed.

After performing a theoretical testing on the alternative keyboards (static, dynamic; circular and rectangular), the dynamic circular keyboard is found more effective towards the parameters defined on the design brief (Chapter 7, section 1-3). This keyboard is subjected to further iterations and is prototyped as the final design.

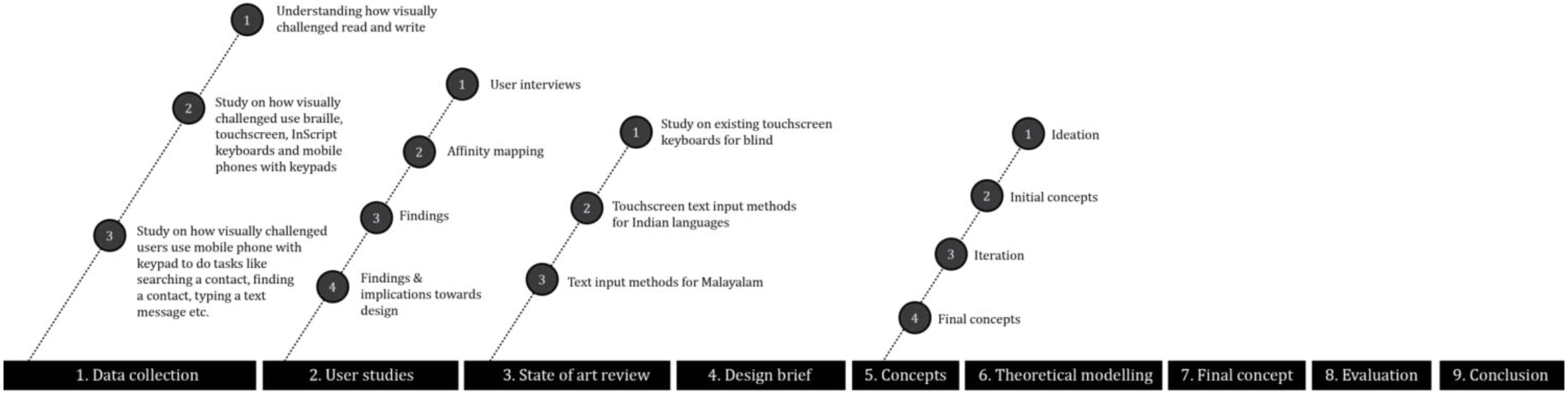
A usability evaluation is performed on the final design for measuring the attributes like: Overall effort, learnability, accuracy and speed. The final design achieved a speed of 1.05 words per minute (speed after the user underwent training for 3 hours with 100 complex Malayalam words) and 0.41 words per minute (mean speed after the users underwent training for 30 minutes with 10 words each).

2. Introduction & Motivation

There are two important issues that need to be addressed with this project. One being the issue that the Malayalis are facing over years: The lack of a text input mechanism to type in Malayalam and enabling touchscreen for the blind to type.

The main aim of this project is to design a touchscreen keyboard for Malayalam that can help any user type without prior training. It also aims at enabling a tool for the blind to perform tasks like saving a contact, searching a contact, typing a text message or typing an email on a touchscreen using Malayalam. The biggest challenge in this project is about enabling blind user to type on a flat screen without the help of physical onscreen markings. Adding physical markings on a touchscreen can disrupt the entire experience of user on the device.

Design process



3. Introduction to Malayalam

Malayalam branched from Classical Tamil that over time gained a large amount of Sanskrit vocabulary and lost its own identity like:. It is generally agreed that by the end of 13th century a written form of the language evolved that was definitely different from Tamil **[17]**. Malayalam's evolution as a modern language dates back to the 16th century. It was heavily influenced by Hindi, Arabic, Urdu, Chinese and many other European languages during its growth. The first book in which Malayalam was used, "*Hortus Malbaricus*" was printed (type based) in 1772. Apart from Kerala, It's also used as a regional language in Lakshadweep, Mahe and Andaman Nicobar Islands. It's currently spoken by 3,30,15,420 users in India and the rest 18,47,902 users outside India**[15].**

In December 2012, Wikipedia had more than 28,030 articles in Malayalam, next to Hindi (1,04,499), Marathi (38,188), Telugu (50,970) and Newari (69,558)**[Figure 3.1][18].** From the table, it is clear that there was a significant growth in the use of Malayalam text input online in past few years (28% growth **[Figure 3.1] [18]**).

Language	Number of speakers (in crores)	Number of Articles		% growth
		2011 December	2012 December	
Hindi	~ 30	1,00,849	1,04,499	3.60%
Bengali	30	22,773	24,958	9.50%
Marathi	9	34,794	38,188	9.75%
Telungu	8	49,670	50,970	2.61%
Tamil	6.6	43,121	50,487	17.08%
Urdu	6	17,380	22,652	30.33%
Kannada	4.7	11,100	12,025	8.33%
Gujarati	4.6	21,866	22,330	2.12%
Sindhi	4.1	394	363	-7.80%
Bhojpuri	3.85	2,698	2,716	0.66%
Malayalam	3.7	21,872	28,030	28.15%
Oriya (Odiya)	3.1	1,951	3,209	64%
Punjabi	2.9	3,362	4,862	44.61%
Assamese	1.3	964	1,926	99.79%
Nepali	1.3	17,889	22,427	25.36%
Kashmiri	50 lakh	238	228	-4.2
Newari	8 lakh	69,801	69,558	-0.348
Bishnupriya Manipuri	4.5 lakh	24,767	24,770	0.01%
Sanskrit	50000	7,015	8,515	21.38%

Figure 3.1: Analysis of the Indic Language Wikipedia Statistical Report 2012 ***[18]***

3.1. Structure of Malayalam

The Modern Malayalam Script consists of 13 Vowels or *Swaraaksharangal*, 36 consonants or *Vyanjanaaksharangal*, 20+ Major Consonant Ligatures or *Kootaksharangal* and 5 Major *Chillu* alphabets (Alphabets independent of vowel sounds) or *Chillaaksharangal*. **[8][9]**

Vowels (13) *with their Devanagari equivalent*

അ	ആ	ഇ	ഈ	ഉ	ഊ	ഋ	എ
अ	आ	इ	ई	उ	ऊ	ऋ	ऎ
ഏ	ഐ	ഒ	ഓ	ഔ			
ए	ऐ	ऑ	ओ	औ			

Consonants (36)

ക	ഖ	ഗ	ഘ	ങ	
क	ख	ग	घ	ङ	
ച	ഛ	ജ	ഝ	ഞ	
च	छ	ज	झ	ञ	
ട	ഠ	ഡ	ഢ	ണ	
ट	ठ	ड	ढ	ण	
ത	ഥ	ദ	ധ	ന	
त	थ	द	ध	न	
പ	ഫ	ബ	ഭ	മ	
प	फ	ब	भ	म	
യ	ര	ല	വ	ശ	
य	र	ल	व	श	
ഷ	സ	ഹ	ള	ഴ	റ
ष	स	ह	ळ	ऴ	ऱ

Conjuncts (20+) frequently in use *(No Devanagari equivalents)*

ക്ഷ	ക്ക	ങ്ക	ങ്ങ		
ച്ച	ട്ട	ന്ത	ന്ദ	മ്പ	ല്ല

Chillu (5) *(No Devanagari equivalents, has separate unicodes)*

ൺ	ൻ	ർ	ൽ	ൾ

The basic characters in Malayalam can be classified into two:

- Vowels (*Svaram*)
  1. Independent vowel letters
  2. Dependent vowel signs

- *Anusvaram*
- *Visargam*
- Consonant Letters(*vyañjanam*)
- Chandrakala (equivalent to *Halant* in devanagari)
- Consonant Ligatures(*Kootaksharangal*)
- *Chillus*

Vowel

A **vowel** is an independent vowel letter is used as the first letter of a word that begins with a vowel.

A dependent vowel is a vowel sign or a vowel modifier, for eg: The corresponding dependent vowel for

"ആ"	is	"◌ാ"
"आ"	is	"◌ा"

"ഇ"	is	"◌ി"
"इ"	is	"◌ि"

Anusvaram

An anusvaram originally denotes the nasalization where the preceding vowel was changed into a nasalized vowel, and hence is traditionally treated as a kind of vowel sign. In Malayalam, however, it simply represents a consonant /m/ after a vowel, though this /m/ may be assimilated to another nasal consonant.

The corresponding dependent vowel for

"അം"	is	"◌ം"
"अं"	is	"◌ं"

Visargam

A *visargam* represents a consonant /h/ after a vowel, and is transliterated as *ḥ*. Like an anusvara, it is a special symbol, and this /h/ is never followed by an inherent vowel or another vowel

The corresponding dependent vowel for

"അഃ"	is	"◌ഃ"
"अः"	is	"◌ः"

Consonant

A **consonant** letter doesn't represent a pure consonant, but represents a consonant + a short vowel.

eg: ക = ക് + അ

Chandrakala

A Chandrakala is attached to a consonant letter to show that the consonant is not followed by an inherent vowel or any other vowel

eg. ക(*ka*) → ക് (*k*)

Chillu

Chillu is a special consonant letter that represents a pure consonant independently without the help of a vowel sound.

eg: "ൻ" is "ന" independent of Vowel sound.

i.e., ൻ = ന്

Consonant Ligatures:

A virama (◌്) is used to kill the inherent vowel of a consonant letter and present a consonant without a vowel. When a dead consonant letter is added with another consonant letter it may result either:

1. A fully conjoined ligature of both the consonants.
2. Half joined ligature: a modified form (half) of the first consonant attached with the second or vice versa.
3. Non-ligated : full forms of both the consonants with a visible *virama*

Common ligatures [13]

	Kka	ṅka	ṅṅa	Cca	ñca	Ññа	ṭṭa	ṇṭa
Non-ligated	ക്‌ക	ങ്‌ക	ങ്‌ങ	ച്‌ച	ഞ്‌ച	ഞ്‌ഞ	ട്‌ട	ണ്‌ട
Ligated	ക്ക	ങ്ക	ങ്ങ	ച്ച	ഞ്ച	ഞ്ഞ	ട്ട	ണ്ട

3.2. Malayalam Unicode

The Unicode is a universal character-encoding standard used for the representation of text for computer processing. It's a consistent way of encoding multilingual plain text; it has enabled users to exchange text files internationally.

The Unicode Standard provides the capacity to encode all of the characters used for the written languages of the world. To keep character coding simple and efficient, the Unicode Standard assigns each character a unique 16-bit value **[Figure 3.2]**.

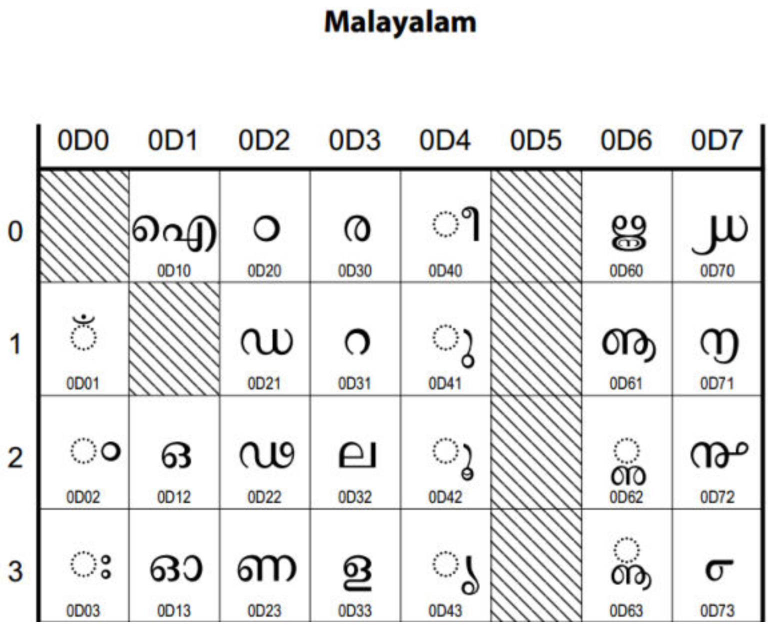


Figure 3.2: Malayalam Unicode chart [23]

Malayalam is allotted code space from U+0D00 to U+0D74 **[23]**.The Unicode standard only addresses encoding and decoding the text elements. It defines how the characters are being interpreted. However, it doesn't define glyphs and doesn't address how the character will be rendered on the screen.

3.3. Text input in Malayalam

Most of the information available on internet is in English (English: 55.7%, German: 6.1%, Japanese: 5%, Chinese: 3.0%, Hindi, Malayalam, Tamil constitutes less than 0.1% individually) **[20]**. With the recent advancement in technology, laptops, smart phones, mobile tablets, internet etc., users have easier access to information and better communication means than what they had 20 years ago. But the only thing that stands in between the users and the recent advancements is the Language barrier. How to overcome this barrier?

The information should be made available for the people in their local language, they should be able to search, access, understand, communicate and modify the information using Malayalam. For this reason, the need for a text-input mechanism is inevitable.

The development of keyboards in Malayalam is as old as 5 decades. A lot of studies where done to design a keyboard layout that can accommodate all Malayalam characters and with high typing efficiency. But most of these studies were limited by technological constraints and assumptions like: *"As the number of characters in a language becomes large, it becomes very difficult to adapt them into devices and is unproductive for typing and printing **[19]**".*

With the introduction of touchscreens, the possibilities to overcome these constraints are endless. But it's very necessary that before designing a keyboard, one should see how these existing keyboards were designed; how it was adopted and what the drawbacks were. Some of the important milestones in Malayalam text input are: Typewriter layout, Malayalam keyboard, InScript keyboard, Malayalam Unicode, Google transliteration etc.

The first keyboard in Malayalam was made for typewriters. After conducting a detailed study, the characters of Malayalam script were modified in 1968 to make it easier to handle in typewriters **[19]**.

3.3.1. Inscript keyboard [19]

Based on the study by Department of Electronics (now ministry of Information Technology, 1983-86), 73 basic characters were decided to be provided in the Malayalam layout. Even though numerals and conjuncts were very important, compromises had to be made in accommodating them. As a result, it is difficult to type the numerals and conjuncts on the InScript Malayalam keyboard (*user has to hold Alt + Cntrl and type the number keys to generate the Malayalam numerals; three key presses are needed to input a conjunct i.e. two consonants added with a chandrakala in between to form a conjunct*). The reason for this compromise was that, the standard QWERTY keyboards were having limited number of keys.

While designing the Inscript layout, the logical structure of the script alphabet derived from the phonetic properties were also taken into account. The vowels were laid out on the left side and consonants on the right hand side. Consonants and vowels were given keystrokes and the consonant-vowel combinations are formed by typing the consonant key followed by the vowel key. Conjuncts are also formed by typing constituent consonants with a *chandrakkala* typed in between.

However there some problems with the InScript layout:

1. **Training:** Adequate training was necessary to use InScript keyboard. Malayalam is taught in alphabetical order, but to learn to type in an Inscript keyboard, users have to learn a new layout where adding vowel modifier, construction of a conjunct involves a number of steps and is hard.

2. **Include Chillu characters:** The chillu characters are extensively used in Malayalam. Based on the frequency of occurrence, There are 5 main chillus in Malayalam,

ൺ	ൻ	ർ	ൽ	ൾ

   Since, it wasn't included in the Inscript keyboard for Windows 7**[Figure 3.5]** (*4 of them are available in CDAC-T New Malayalam Inscript _Keyboard for Windows XP* ***[Figure 3.4]****, Service Pack 2)* ***[21],*** user cannot type the 3 *chillus* – 'ൺ', 'ൽ',' ൾ'. User can type 'ർ' and 'ൻ' by holding the Cntrl + Alt and then typing the numbers 4 and 9.

   In modern script, the sign of *virama* is used to construct a *chillu* character. However it's problematic [22].

For example, “avan” അവൻ (he) is written as:

അ + va വ + chillu-n ൻ,

With the new InScript keyboard (without chillus), I can construct the same word using a *virama*, avanŭ അവന് (“to him”). But, this new word is not having the same meaning as “avan”.

So, the entire idea of removing *chillus* from the keyboard and constructing them using *virama* is a wrong assumption. Hence the *chillus* should be given place in the keyboard.

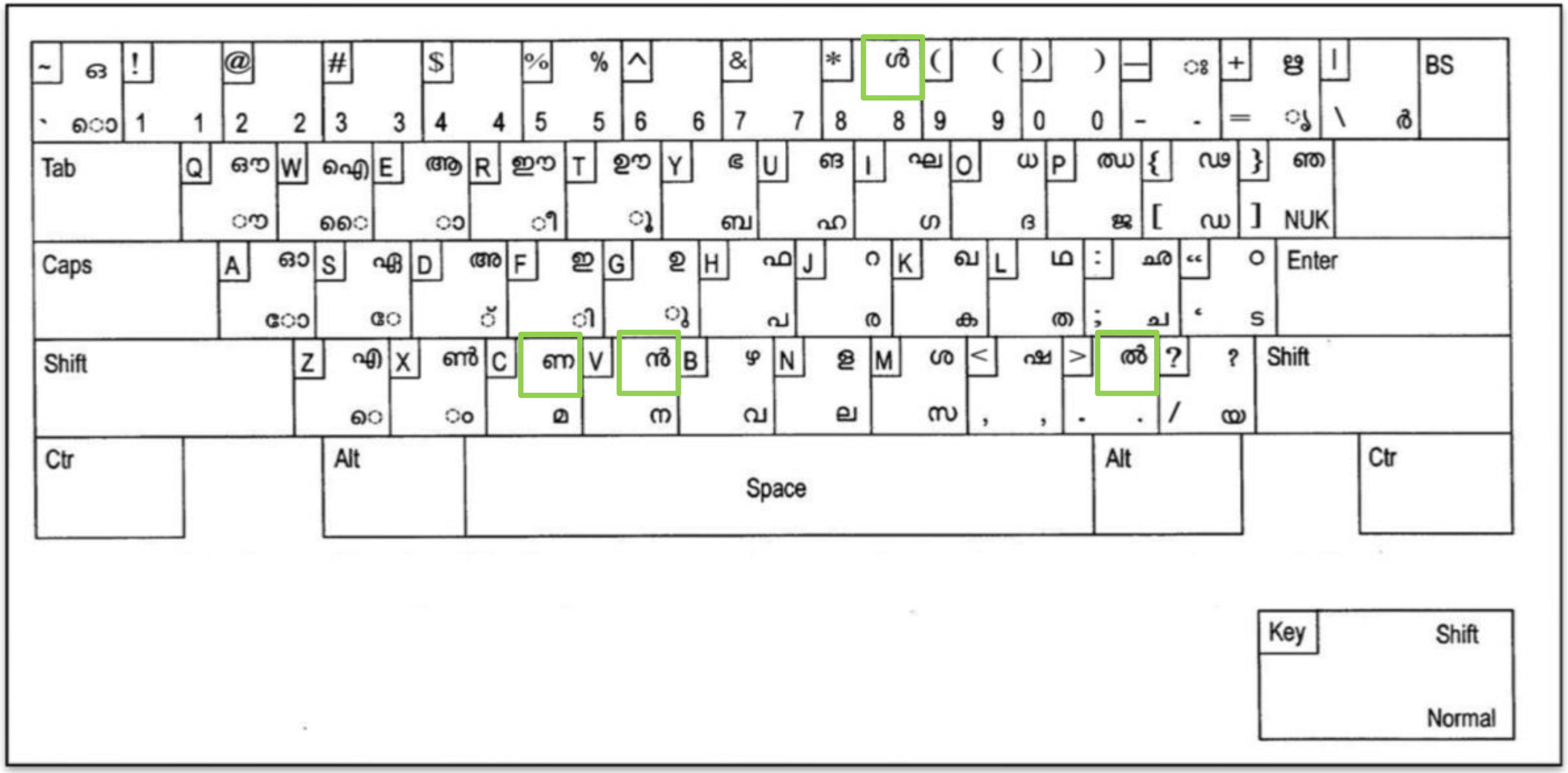


Figure 3.4: The position of *Chillu* characters in Malayalam InScript Layout for Windows XP with Service Pack 2. **[21]**

Figure 3.5: The missing *Chillu* characters in Malayalam Inscript Layout for Windows 7

3.3.2. Google transliteration

Google provides a transliteration feature for Malayalam. It’s based on the idea that user types a Malayalam word using English keyboard, and the tool generates possible Malayalam words that correspond to the input. It’s used for typing Malayalam at a very good speed by people who are familiar with typing in English, but not familiar with any Malayalam keyboards. It also helps user with handwriting recognition **[Figure 3.6]** that helps user in searching, finding and inserting the desired character.

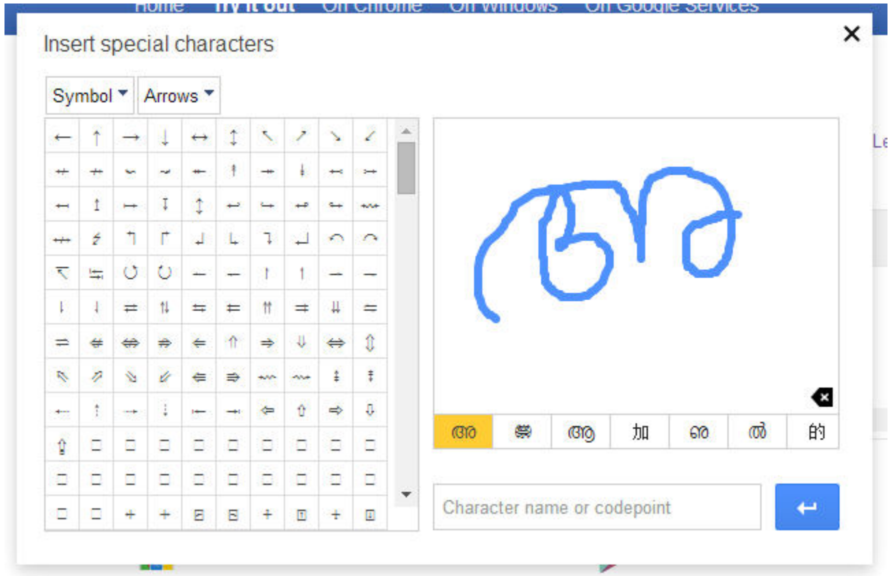


Figure 3.6: Hand writing recognition for inserting special characters

The main problems associated with transliteration are:

1. **Error rate is high as the same English spelling may have multiple Malayalam words**: The accuracy of the prediction is very low. The algorithm searches online for Malayalam documents/contents based on which the word formations are predicted.

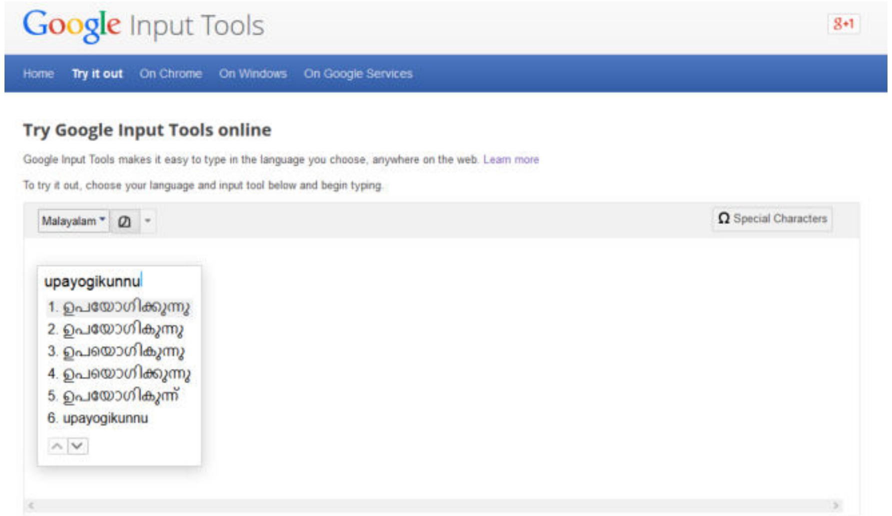


Figure 3.7: Prediction made using transliteration

Since Malayalam doesn’t have many documents/content online. The accuracy of the prediction is very low and the quality of Malayalam written using transliteration tool is very bad.

For example, If I want to type ‘upayogikunnu ഉപയോഗിക്കുന്നു’, I have to type ‘upayogikunnu’ and the tool will generate a list of predictions that matches my input **[Figure 3.7]**. The predictions may differ based on the input that I’ve inputted. The user has to choose the right ‘spelling’ from the list of 5. How will I choose between the predicted list of words and make the right choice? There is a greater chance that the choice made be wrong.

2. **Pain to make it error free**: User should know how the spelling for each word. Even if the prediction is wrong, user can make it right by choosing the characters separately and adding them individually. Few steps are involved in this process and since the accuracy level of prediction is low; user has to do a lot of manual edits.

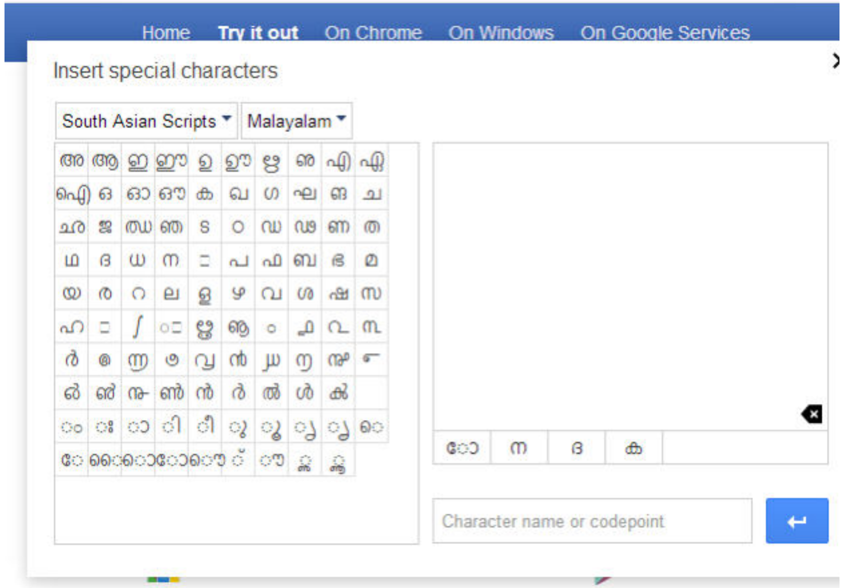


Figure 3.8: if the predicted list doesn't contain any word that spells right and user wants to edit the word, he has to use this interface to add characters individually where he doesn't get a quick feedback of outcome of the combination.

3.4. Touchscreen Mobile Text Input Methods for Indian Languages (Devanagari) [10]

The text Input methods in Devanagari:

- Inscript
- Swarachakra
- Disha

3.4.1. Inscript

The standard Inscript uses a partially logically odered and partially frequency based layout **[Figure 3.9]**. In logical ordering, the vowels are placed on the left and the consonants are placed on the right side of the keyboard. Each dependent vowel are mapped to the same key. Similarly for most consonants, the base consonant and its corresponding aspirate are placed on the same key.

3.4.2. Swarchakra

Swarachakra is a logically ordered design. In this design, the initial layout displays only consonants. They are phonetically arranged and grouped and arranged in a grid similar to be found in the most school textbooks. When the user clicks on a consonant, a wheel with combinations of consonants and frequent dependent vowels pops up around the consonant **[Figure 3.10]**. To select a combination, the user uses a stylus to slide over it or taps on it.

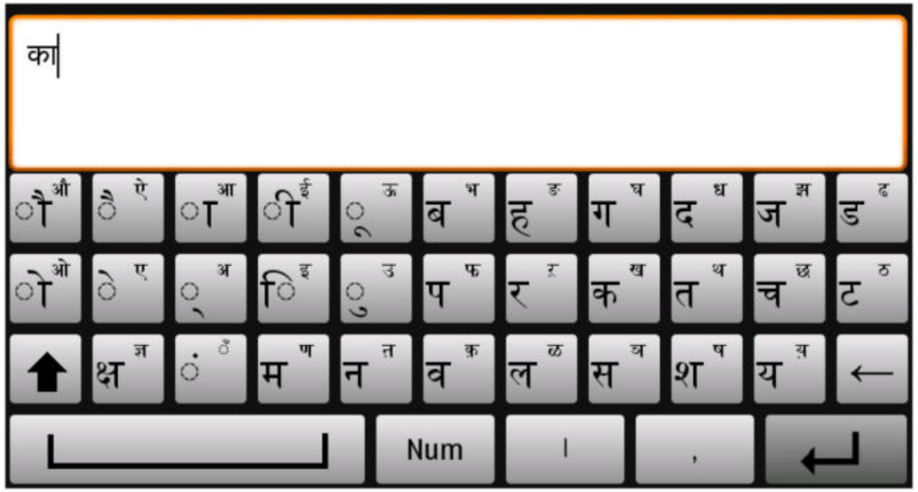


Figure 3.9: Inscript layout

3.4.3. Disha

Disha literally means “direction”. This logically ordered design takes advantage of phonetic grouping of the devanagari characters and the sliding gestures used by the touchscreen. A group of 5 characters with phonetic similarity (क, ख, ग, घ, ङ) are accessible from a single, large button **[Figure 3.11]**. In case the phonetic similarity is not inherent within a group of letters (in case of semi-vowel or vowel groups), the alphabetical order is used for grouping. The first character in the alphabetical (usually also the most frequently used character of the group) is placed at the centre, and the other characters are distributed in a clockwise manner around it.

To select the character in the centre, the user needs to tap a button. To select one of the characters on the side, the user needs to tap and slide the stylus in the direction of the character. While doing so, the user is allowed to extend the slide beyond the button boundary.

Figure 3.10: Swarachakra

Figure 3.11: Disha

3.4.4. Why existing touchscreen keyboards on Indian languages are very hard to be adopted by the blind?

The advantage of InScript touchscreen keyboard over any other keyboard is that it is the only keyboard layout that can be easily be adopted by the blind users, mainly because, they are trained to type over QWERTY layout on computer keyboards. Blind users are more familiar to type InScript over physical keyboards as they have physical markings on keyboards to locate the keys.

As the touchscreen is flat, it might be harder for the user to find a starting point on a flat surface. Adding physical markings on flat surface can spoil the entire experience of the user over the touchscreen. Even if appropriate feedbacks like voice and haptic can be used for locating the keys, finding the starting point will be time consuming and this will slow down the user.

Keyboards that follow logical arrangement of alphabets: Swarachakra and Disha will also face similar problems in locating the keys. Apart from that, the logical way of approach in typing will be more helpful for them in predicting what lies ahead: like “ग” always come after “ख”. It’s more intuitive and user doesn’t have to spend much time in understanding the InScript layout.

3.4.5. To summarize

From the study on different types of touchscreen mobile text input mechanisms, it’s evident that the InScript and Logical layouts have their own advantages as well as disadvantages. It’s very difficult to say which layout is better than the other considering the end users. All blind users are not trained to type in InScript, but both the set of users are taught the language logically or in alphabetical order.

Since we don’t have enough studies supporting which layout is good for the blind users, the best possible way to generate keyboards in both layouts and test them.

3.5. Touchscreen Malayalam Text Input Methods

The text Input methods in Malayalam:

- Nokia Asha Touchscreen Keyboard
- Swarachakra Malayalam keyboard
- Indic Keyboard Prime (Malayalam)

3.5.1. Nokia Asha Touchscreen keyboard

In this method, there are two levels of letters. The top one is about vowels, while the bottom has consonants, these levels can be extended based on the need. It follows a logical order of arranging the alphabets **[Figure 3.12]**.

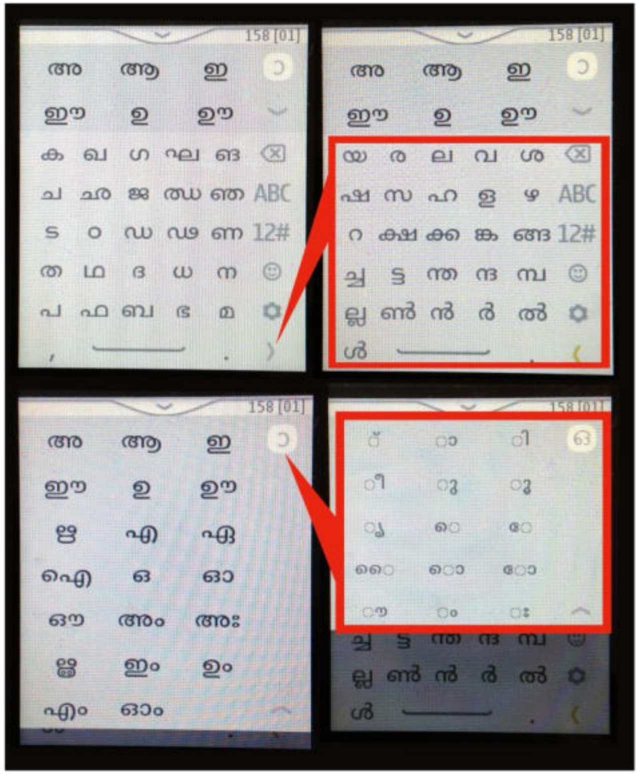


Figure 3.12: Nokia Asha /Malayalam touch keyboard

3.5.2. Swarachakra Malayalam

Swarachakra Malayalam has a similar layout like Swarachakra Hindi. It is a logically ordered design. In this design, the initial layout displays only consonants. When the user clicks on a consonant, a wheel with combinations of consonants and frequent dependent vowels pops up around the consonant **[Figure 3.13]**. User can select the desired combination from the wheel thereby reducing the number of steps to add a consonant and a vowel modifier. To insert a conjunct, user has to select the consonant with *virama* symbol from the wheel and the entire layout remaps into a list of conjuncts that are based on the consonant. Swarachakra Malayalam also supports most frequent *chillus.*

The chakra cannot accommodate the entire list of Malayalam vowel modifiers due to lack of space. The remaining vowels from the chakra can be found in '◌ാ' and 'ര്'.

Figure 3.13: Swarachakra Malayalam Keyboard for Touchscreen mobile phone

3.5.3. Indic Keyboard Prime (Malayalam) [24]

Indic keyboard prime follows Malayalam InScript based layout **[Refer chapter 3, section 3.3.1] [Figure 3.14]**. The vowels are placed on the left and the consonants are placed on the right side of the keyboard. Each dependent vowel is mapped to the same key. User should use shift key (🡅) to switch between these two. It also provides word suggestions.

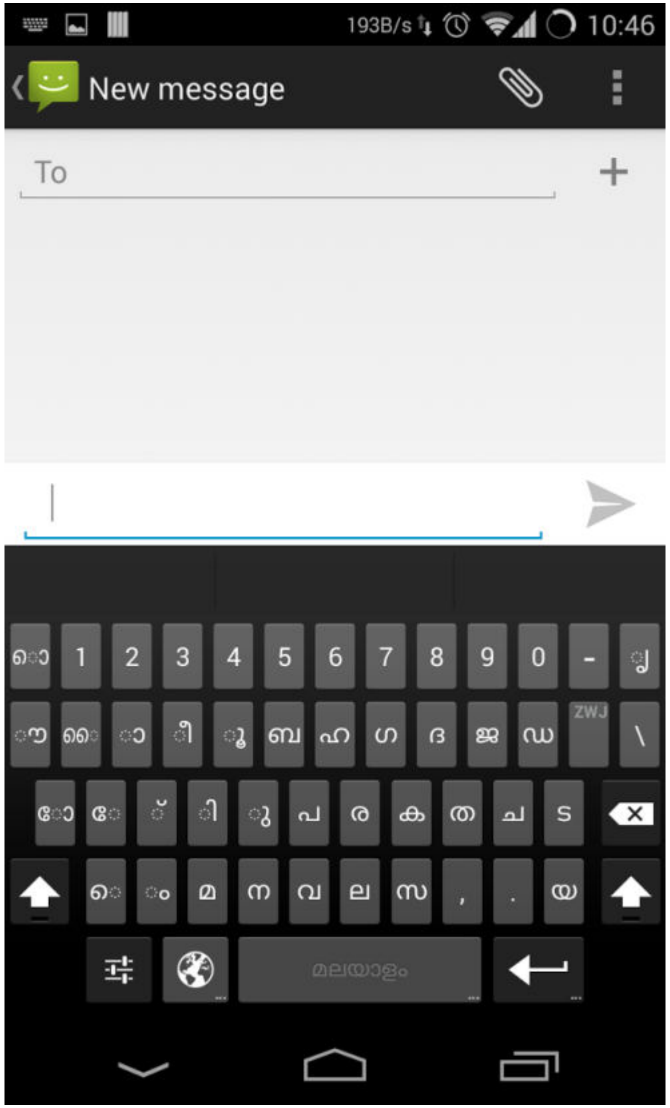


Figure 3.14: Indic Keyboard Prime (Malayalam) [24]

3.5.4. To summarize

From the review of text input methods for Malayalam, it's evident that the problems associated with text input in Malayalam are: large number of alphabets; arrangement of letters on the screen; constructing a consonant ligature etc.

The Nokia Asha touchscreen keyboard is an example of a logical approach based keyboard. The main disadvantage of the design is that, a lot of characters are forced to fit into this small screen irrespective of the size of the button. Listing down the entire letters only serves to fill the screen and reduce the size of each button. The minimum size of a button on a touchscreen should be around 45 to 57 pixels i.e. the minimum size should be 45 X45pixels **[8][6].** None of the touchscreen keyboard follows this. This reduces the visibility of the alphabet on the button and also causes errors as the keys are arranged very close to each other. Hence the keys become too small to be typed by a finger. This problem generally happens with touchscreen phones that has small screen resolution.

When it comes to the blind, the challenges are more. User needs to be given appropriate feedbacks like where he is, what he has typed etc. The interactions should be more intuitive to the blind user and the time required for training a user should also be less.

Due to the absence of haptic feedback mechanisms, Voice support and TTS support (Text to Speech), A blind user cannot use these keyboards. The size of the buttons makes it error prone and hence, we can arrive at a conclusion that there are no touchscreen keyboards on an Indian language that can support a Blind user.

4. Studies on Blind

In India, Visual Acuity and Field of Vision are considered as the criteria for Blindness. It refers to a condition when a person stuffers from the following three conditions [3]: Total absence of sight, Visual acuity not exceeding 6/60 or 20/200 or Limitation of the field of vision subtending an angle of 20 degree or worse.

Visual Acuity is acuteness or clearness of vision. Visual Acuity is tested by requiring the person whose vision is being tested to identify characters on a Snellen chart [Refer **Figure 4.1**] from a set distance. Normal visual Acuity is commonly referred as 20/20 vision.

Visual acuity not exceeding 6/60 or 20/200 means the distance in feet between the Chart and Subject (20 feet away), the same chart can be read by a person with normal vision 200 feet away [4]

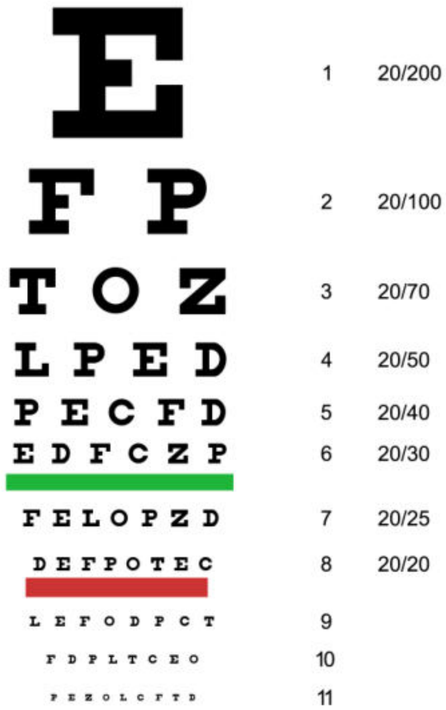


Figure 4.1: Snellen chart [14]

Figure: 4.2 Normal field of vision [4]

Normal field of vision is 60 degrees up, 75 degrees down, 60 degrees nasally (towards nose), 100 degrees temporally (away from nose, outward). **[Figure 4.2].**

The current prevalence of blindness in India is estimated to be 15 million. Nearly 1/3 of the total blind population in the world resides in India (if the cut of point of < 3/60 in the better eyes according to the Snellen chart[1] is considered) [1].

It is known that only less than 1% of these blind are literate [2]. This means that the blind education has not yet reached the major part of the blind population. The number of blind schools in each states are limited and they are not accessible by the students due to their geographical locations, long distance travel and travelling with disability.

Literacy among blind is considered on the basis of the ability of the Blind to read and write using Braille. Braille is a tactile writing system used by the blind. It is written over an embossed paper. A blind user can write braille with an original slate and stylus or type it on a Braille writer such as a portable braille note taker **[Figure 4.3**], or on a brailler/braille typewriter **[Figure 4.4].**

[1] 3/60 according to Snellen Chart means the distance in feet between the chart and the subject(3 feet away), the same chart can be read by a normal person standing 60 feet away

Figure 4.3: Portable note taker with a slate and stylus.

Figure 4.4: Brailler or a Braille type writer

Braille characters are small rectangular blocks called cells that contain raised dots. The number of arrangement of these dots distinguishes one character from another. They have braille characters defined for all Indian languages **[Figure 4.5].**

With the advent of screen readers, and talking softwares in computers, the usage of braille method is declining. There are not much braille presses in India to provide enough braille material for the whole blind population. However, the braille education still remains important for developing the reading skills among the blind children as it increases their touch sensitivity and concentration.

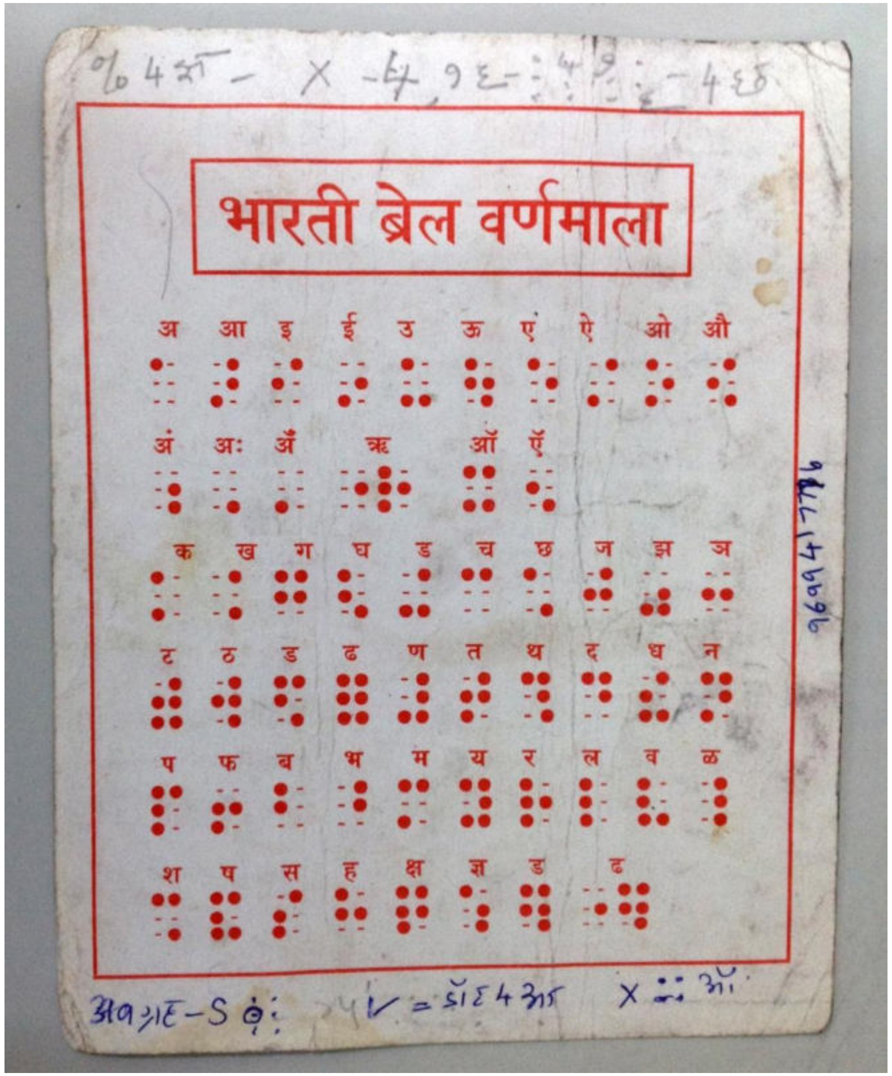


Figure 4.5: Braille code for Devanagari

The blind schools in Kerala train the students how to use the Inscript keyboard to type. Almost all the students interviewed at NAB Trivandrum could type in Malayalam as well as English and they were found to type even faster than the sighted ones on the computer. The number of people who are getting trained to use Inscript keyboards is very small when compared with the total blind population. So the lack of training and proper keyboards holds back the visually challenged from using Malayalam for typing in existing Mobile devices (both Touchscreen as well as Mobile phones with keypad).

4.1. How visually challenged users read and write

The visually challenged users have several assistive devices to carry out their daily activities. These assistive devices can broadly be classified into the following categories: education, mobility, vocational, daily living devices, low-vision devices etc. In this project we are concerned more about the devices that are being used in education, i.e. devices that are used for reading and writing. (*Based on the artifacts collected from Barrierbreak s and NAB, Trivandrum*)

4.1.1. Educational devices

Braille writers: *Brailler* and thermoform machines are used to convert the materials into braille.
Brailler is a device that is used for printing braille on paper. A special kind of thick paper is used for this process, the typewriter press the paper against the dye containing braille characters, thereby leaving a tactile impression on the paper. The user can read the printed words by touching and feeling the braille projections on the paper.

Thermoform paper expands when heated; this property of this paper is used to create images that pop-up from the surface of the paper. Hence the blind user can feel the images rather than seeing them **[Figure 4.6].**

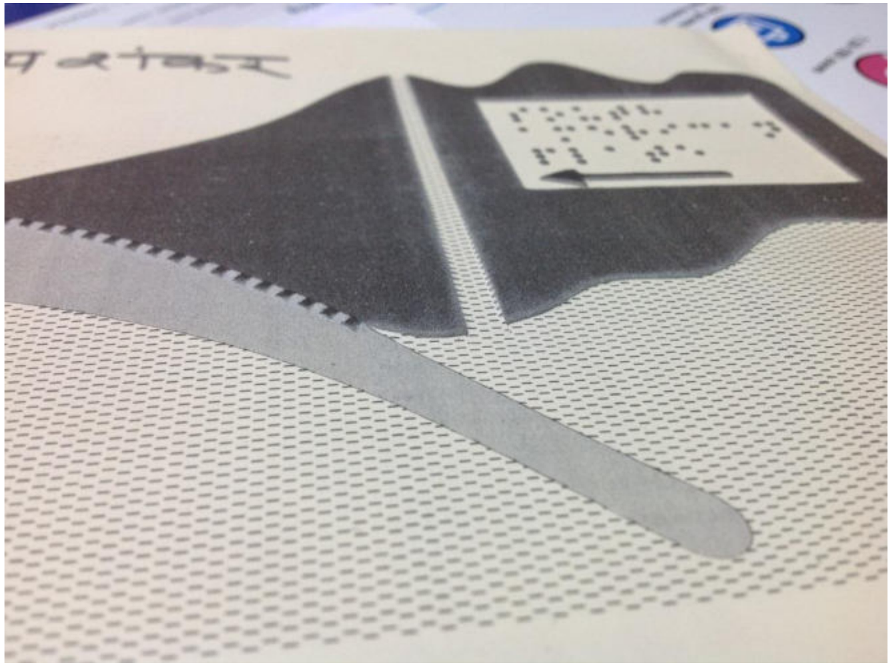


Figure 4.6: Thermoform paper with pop-up illustration

- **Writing devices:** Braille slates, Braille frame, Braille paper etc. are the devices used by the visually challenged users to write in braille. The braille slate contains a braille frame with 3 curves on either sides of it. User can use a stylus to punch holes on the paper following the curves in the braille slate. User can open the braille slate, remove paper and read it once the notes are taken**.**

- **Talking books**: Materials recorded over devices has emerged as the most popular mode of imparting education **[Figure 4.7]**.

Figure 4.7: Talking books

- **Talking software:** While working on computers, the users will get voice feedback on “where user is” and “what user has entered”. JAWS, TTS etc. are some of the most commonly used talking softwares.

- **Mathematical devices:** For teaching geometry, geometrical devices specially marked with dots are used **[Figure 4.8]**.

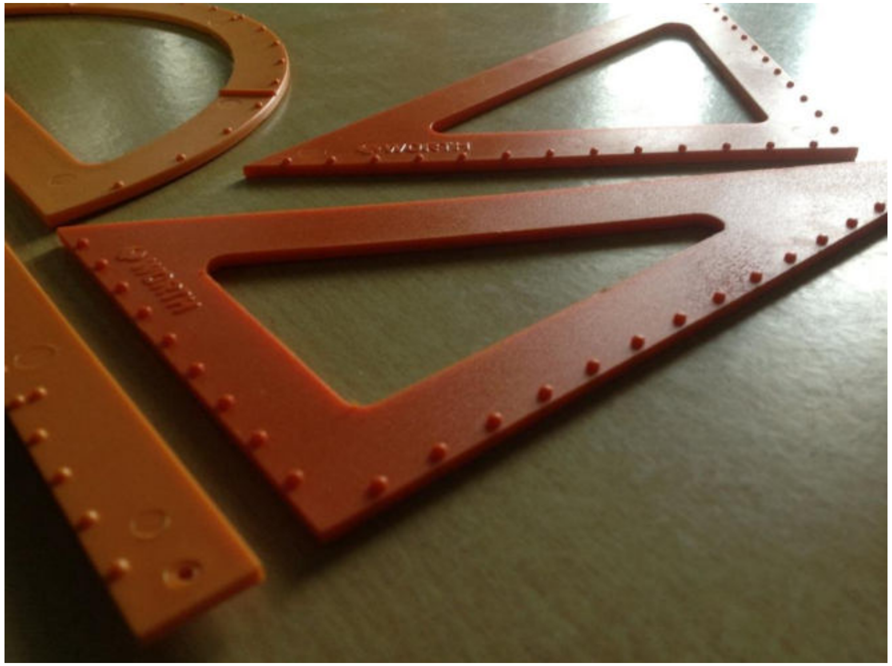


Figure 4.8: Geometrical devices marked with dots

4.2. How visually challenged users use physical keyboards

The visually challenged users can use a computer or a mobile phone with physical keypad as easy as a sighted person. The government of India standardized **The Inscript Layout**, over the QWERTY keyboard and is available on most operating systems. In spite of standardization and wide availability, InScript or any other keyboard on PC's has not been widely adopted by the users till date **[5]**.

The large number of characters and fitting them in a small Touchscreen Interface is another challenge. However keyboards like Swarachakra[5], Keylekh[11] also addresses the same issue. But how could we make a keyboard that could help visually challenged to type in Touchscreen Indian language keyboards?

From the user study, it's clear that the literate visually challenged are trained in braille to read and write hence they only know the sound and feel of the alphabets, they don't know how an alphabet look like. Here for a completely Blind user, an alphabet meant a tactile feel of a unique arrangement of dots or a particular sound for it. Here the visual representation of an alphabet is substituted by a tactile feel or a sound. Therefore to type, finding location of the keys, having tactile feedback and voice feedback through Screen readers was necessary. Typing on Inscript keyboard was enabled using the tactile markings on keys "F" and "J"**[Figure 4.9]**, number "5"**[Figure 4.10]** on the number pad, and also with screen reading softwares like JAWS[2] and ILTTS[3](in Malayalam). It reads out what the user have entered. This led to the hypothesis that by providing the tactile and Voice feedback, the visually challenged can be able to use the Touchscreen Keyboards in Indian Scripts.

In my initial study, I found none of the users I interviewed were using a touchscreen keyboards for inputting text in Indian Scripts.

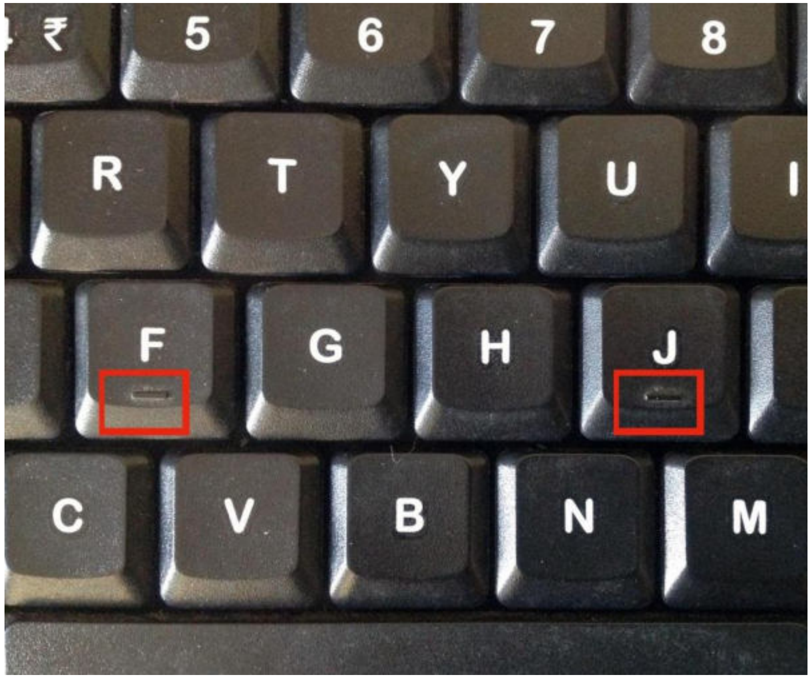


Figure 4.9: Markings over "F" and "J"

[2] JAWS or Job Access With Speech is a screen reading software in English
[3] ILTTS is a Text To Speech System in Indian Languages, developed by C-DAC

Figure 4.10: Marking over “5” on Number pad

4.3. How visually challenged users use mobile phone in daily life

Almost all the phones that are available in market with physical keypads can be used by the blind. User have to first find the starting point i.e. “5” **[Figure 4.11]** based on which user creates a mental map in his mind about the keypad layout. The voice feedback supports the user in using the number pad. A visually challenged user can dial a number without any trouble. When it comes to searching a contact, or saving a contact, users first have to navigate and find the application from the menu. Since, the menu button, start button and cancel button are different in size and shape; it is very easy to differentiate the menu button from the other keys in the keypad. This gives a quick access to the menu. Now user has to navigate through the menu items using the directional keys and the voice feedback lets the user know where he is. To save a number in the contact list, user has to add a name for the contact. The user can still use the same keypad to type in English. Here 1 stands for ‘a, b, c’. If user wants to type ‘b’, he has to press the key “1” thrice. For ‘c’, he has to press the same key four times.

To search for a contact, user generally finds the contact, and then starts typing the user name. The voice feedback helps user navigate and arrive at the right contact. Typing a text message is also as simple as the above tasks.

Figure 4.11: Markings over '5' on keypad

4.4. To summarize

*From the above study on how the visually challenged users use the mobile phone keypad, it was clear that, since the number of Indian script characters over a button is too high that the number of steps involved in entering a character is too many **[Figure 4.11]**. User has to keep pressing the same button a number of times to enter the same character. This is one reason why typing an Indian language on mobile phones with keypad is difficult for the blind users.*

During the data collection phase, none of the blind users were found to be using a touchscreen phone. The reason why touchscreen is not yet usable for blind is because the screen is flat and no physical markings can be applied on the screen as it may affect the entire user's experience on the touchscreen badly. So, how will the user find a starting point on a flat touchscreen surface? This is the biggest challenge of the project.

To validate the above findings and gathering more data, there was a need for further user studies.

5. User Studies

To validate the data from the initial data collection, there was a need for a detailed user study. User studies were done at three places, Braille Press: National Association for Blind, Mumbai; Barrierbreak Technologies, Goregaon, Mumbai and National Association for Blind, Trivandrum.

5.1. User interviews

The study at NAB, Worli and Barrierbreak technologies aimed at getting a big picture of how visually challenged used Inscript keyboards and the Mobile devices. The second stage of user studies at NAB, Kerala aimed at validating the earlier findings and concentrating more on how the visually challenged users input text on a mobile phone.

There were a certain set of questions for which the answers were sought.

- Name, Age, Education
- Do you read and type in Malayalam? If so, How?
- Tasks you do with your mobile phone?
- Do you use a computer?
- Have you used a touchscreen mobile phone before?

All the answers were validated by short demonstrations, where the user was asked to perform tasks based on the answers. A master apprentice model was followed throughout the interviews.

5.2. Affinity mapping

The data collected as a part of the user studies were in the form of voice record files, quick notes, observations etc. these data needed to be organized in a meaningful way for the better understanding of the problem. All the interviews were converted into excel sheet transcripts in a Question – Answer pattern. All the conversations were given a code, and an affinity mapping was conducted **[Figure 5.1].**

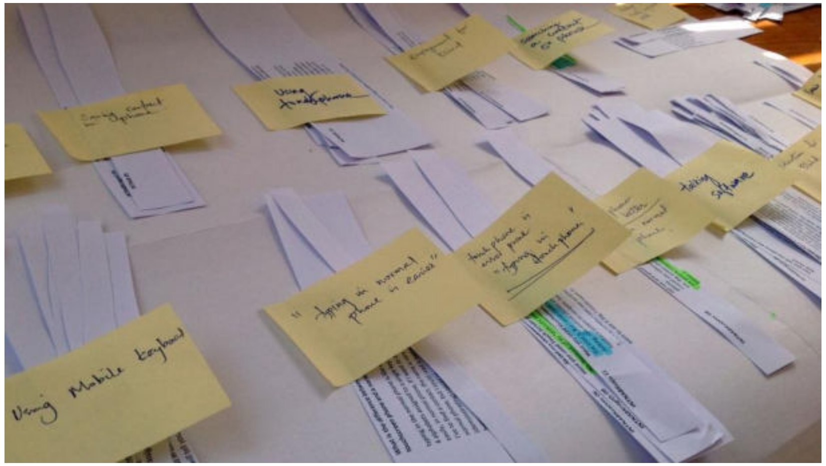


Figure 5.1: Affinity Mapping

5.2.1. Users

2 Female users from Mumbai, of age group 28-35, working at NAB Worli and Barrierbreak Technologies, Goregaon.

7 users from Trivandrum, 5 Males and 2 Females of an age group 26-38, all of them were students, apart from studies, they were also working with NAB, Kerala and helping other new students with training process.

5.2.2. Education

All the users were educated, 7 out of 9 users were degree holders. 2 out of 9 users were working as full-time software testers. 5 out of 9 users were students and out of which 2 were trainers.

All 8 out of 9 users were trained, and are able to type in computer using InScript Keyboard (Both Malayalam and English)

6 out of 9 users could use software's like Excel; Word etc. 1 user was using Audacity to edit sound clips

5.2.3. Tech Savvyness

All users could use computer without anyone's help, as they were using shortcuts from keyboard for all the tasks. They used a "Talking Software" by name "JAWS"- Job Access with Speech and ILTTS in Malayalam (Text To Speech System in Indian Languages, developed by C-DAC). 4 users were using email services and could type emails in English as well as using InScript keyboard

5.2.4. Mobile phone usage

All 9 were Mobile phone users. One user was using Touchscreen phone (iPhone) and she was using Voiceover in iPhone for navigating through the menu, Typing a text message in English etc . Voiceover is a screen reader built into iPhone and other iOS operating system, By using Voiceover, the user can access their device based on spoken descriptions. For navigating through the interface, the user can either slide through the touchscreen or just by swiping on any fixed part of the screen to move to the next application. As the particular application is selected, it speaks out the name of the app, followed by "*Double tap to enter*". For entering the text, the user can slide over the touchscreen keyboard, it reads out the keys and double tap is used as a gesture for entering the alphabet. There is a difference in tone when the voiceover speaks out where the user is, and what the user has entered.

5.3. Findings

6 out of 9 users were able to search a contact, save a new contact and type a text message in English using the phones with keypad. 4 users talked about Touchscreen, They got exposed to Touchscreens either from their relatives or their friends.

5.3.1. If trained, Blind users can use Computers and Inscript Keyboards

In Blind Schools, Blind students are taught Braille, they read and write using Braille hence they don't know how an alphabet looks like, and they only knew the sound and feel of the alphabets. A small number of them learned to use physical keyboards like InScript keyboards, and they can type faster than the sighted people, they use "Windows screen reader", "JAWS", "ILTTS" for reading out the screen content. Windows Screen reader and JAWS use English while ILTTS gives feedback in Malayalam. This clearly tells that once the Blind user is given training, he could use the Computer as well as type in InScript keyboards

5.3.1.1. Issues in using an Inscript keyboard for Text entry.

Navigating through keyboard and finding the right alphabet, adding two alphabets for the construction of a glyph or a Conjunct was difficult on an InScript keyboard.

5.3.1.2. Switching languages on an Inscript Keyboard

"I've to remember the Malayalam layout over the English keys every time I change the language to Malayalam". In Blind schools, when the visually impaired students are trained on Computer Keyboards, they are taught to type in English first, then on the InScript layout. The user had to memorize all the time which English alphabet is over which key on an InScript keyboard and while switching over to Malayalam, he had to map the new set of alphabets(almost twice than English) over the same layout. This puts a lot of cognitive load on the user.

5.3.2. Mobile phones with keypad are more usable than a Touchscreen phone

From the user studies, it was found that one blind user used Touch phone. She was using iPhone with the help of Voiceover (An inbuilt screen reader in iOS devices). For entering the text in English using Voiceover, the user can slide over the touchscreen keyboard, it reads out the keys and double tap is used as a gesture for entering the alphabet.

Phones like iPhone and with android OS have accessibility softwares (Voiceover, Talkback[4]) but They found using a mobile phone with hardware keypads in English and voice feedback more usable. Mobile phones like Nokia C5, E Series, N70, N73 etc., had hardware keypads and talking software and 8 out of 9 users were found to be using these phones.

Reasons why blind users find mobile phones with keypads usable?

[4] TalkBack is an Accessibility Service in android that helps blind and vision-impaired users interact with their devices more easily. This application adds spoken, audible, and vibration feedback to your device. It is a system application that was pre-installed on most devices and is updated when the accessibility service is improved.

1. Users can dial, save and search a number
2. User could type a message in English
3. Adequate feedbacks to help a blind user: Talking software; Physical keypad with markings.

Still, the mobile phones with keypad are not usable for blind users to type in Indian languages. ***(Refer chapter 4, Section 4.4)***

The reason why Mobile phones with keypad are more usable than a touchscreen phone is that the blind people can't create a mental model of how to use a touchscreen based phone.

5.3.2.1. Existing Mobile keyboards (Touchscreen and Hardware keypad) on Indian Scripts are not for Blind

One user said "We don't need a new keyboard, we need guides for training blinds to use the existing keyboards", this implies that there are a lot of keyboards available in Mobile phones with keypad as well as in mobile phones with touchscreen, But from user studies it was clear that majority of these users (8 out of the 9 users) didn't know or were able to use these keyboards for typing Malayalam. The reason being:

1. The users couldn't figure out a mental model of how to use it.
2. Number of keys is less and the number of alphabets is high.
3. The user had to memorize the Malayalam layout over the existing number layout every time he switched the languages
4. Each button holds 5 letters, user have to keep pressing the same button a number of times to enter a desired character. This increases the number of steps; key presses to enter a word.

Eg: button '4' contains 5 letters:

ക	ഖ	ഗ	ഘ	ങ

If I want to type 'ങ', I've to press '4', five times.
If I want to enter a word with just two letters- 'ആന', I've to go through 7steps (2+5).

5.3.2.2. Touchscreen is a very bad Interface for Blind

Touchscreen is flat; there are no tactile objects or markings on the screen that can help user to differentiate the elements that appear on the Touchscreen. "I touched somewhere and it got clicked, but I didn't feel anything - It has a bad interface and it's difficult to use. " One of the user complained that touchscreen is difficult to use since he can't get any tactile feel of the buttons on the screen. Blind user first tries to locate the keys that are having the physical markings- both on a Computer Keyboard ("F" and "J") **[Refer Chapter 4, section 4.2]** as well as a Mobile phone with a physical keypad ("5" on Number pad) **[Refer Chapter 4, section 4.3**].

Then he tries to create a map inside his mind, if he wants to type "D" on a Computer keyboard, He finds "F" first, place his fingers on the Keyboard, then create a Map of Layout inside his mind, and then finally figures that "D" comes under the middle finger, next to the "F" key. This implies the blind user need a starting point on a keyboard to begin with. However, touchscreen cannot substitute this tactile feedback; hence the visually challenged user feels that Touchscreen has a very bad interface. However, there was another user who was using a touchscreen phone. She used accessibility software like

Voiceover for operating the phone. This implies that if trained, the other blinds could also use the touchscreen easily.

The blind users feel touchscreen is not usable for them due to their lack of training and inexperience.

5.3.3. Voice feedback can make the user annoyed in long term use.

Without voice feedback, it's very hard to use the computer as well as the mobile phones but it's very hard to listen to Screen reader for more than 10-12 hrs. "It keeps shouting out all the things on the screen, sometimes, It's annoying"- This implies that The Screen reader software reads out everything that appears on the screen, both wanted as well as the unwanted contents. It creates frustration in the user in long term usage.

5.3.4. The Blind user needs assistance for setting up his mobile device

Talking software, Speed dial, Key shortcuts, different ringtones for different contacts and Voice tags (Set by taking help from a sighted person) are used by Blind user to operate Mobile phones.

5.4. Findings and Implications to Design

From, the initial studies and the user studies, the following implications towards design are derived.

5.4.1. How a blind person use Mobile phone with Keypad

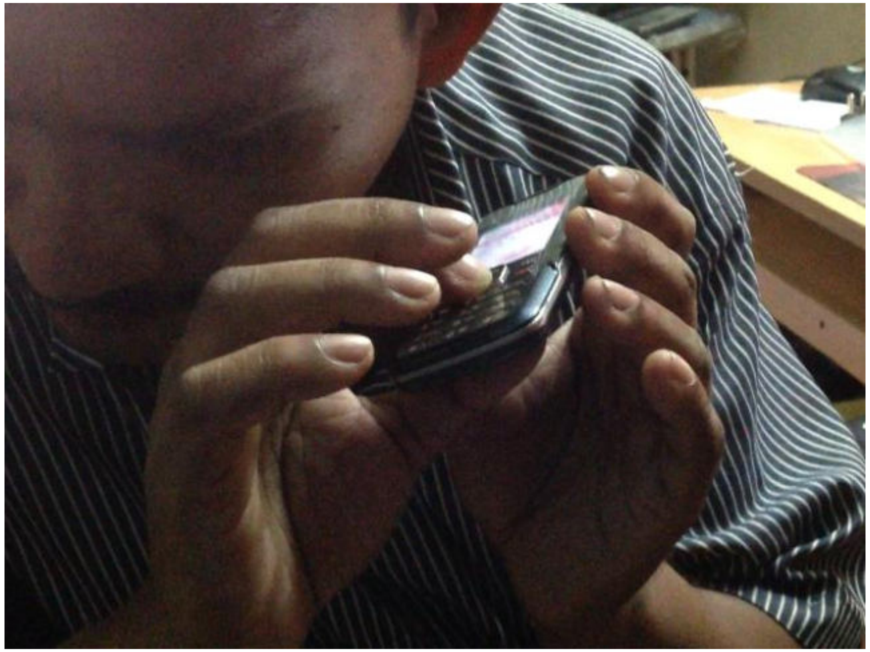


Figure 5.2: Blind user searching a contact by using keyboard shortcuts.

The blind user normally uses Mobile phones with keypad that has defined keys for accepting a call, disconnecting a call, navigating through the main menu etc. **[Figure 4.11].** Keys differ with each other in shape and sizes. So, it makes the blind user identify the keys easily. Key shortcuts are used for accessing different Applications like

Contact page, text message, Missed calls etc. The key shortcuts reduce the effort and number of steps to perform a desired task. The shortcuts are generally set around the navigation button and the other buttons around (select button, Back/cancel button, call button etc.). Apart from the different key sizes and shapes, these phones also has talking software for Audio feedback, it reads out where the user is, and what he has entered.

Normally different ringtones are set for different contact names, so that the user can quickly identify the caller. The changes in settings are normally done with the help of a sighted person. Different sounds can help the Blind recognize different entities.

Voice tags are used for quick dialing where the user presses a button for a few seconds and a voice input interface appears where you can enter the caller's name by speech and also by speed dial where different keys have different contacts assigned to it, a call will be triggered as the pre-defined key is pressed for a few seconds. Again, these shortcuts are mostly set with the help of a sighted person. It has its limitations like, not many speed dials can be set due to limited number of keys.

Phones with keypad are used normally for entering the text, keywords, The mobile phones have physical markings on "5" on the keypad **[Figure 5.2]**, that helps the user find the starting point, and then he figures out the keys around it and so the alphabets on each key, i.e. the physical markings provide the tactile feedback and help the user differentiate the particular key from all other keys.

Some people use phones without talking software, but most likely to end up in errors. For e.g. two users were found using mobile phones without talking software. From interview it was clear that they had to change their handsets since the earlier phone stopped working, and they couldn't find phones with talking softwares anymore as according to them, the companies stopped their production and these phones weren't available anymore in the Market(Nokia C5, E Series, N70, N73, Supernova etc). They could only use the new phones without talking software just for dialing a number, receiving calls as well as disconnecting them. They had to seek help from the sighted ones for reading their personal messages. There are many tasks that are done without the help of talking software, e.g. Simple navigations using the central navigation button on a keypad based phone can be given a small sound as a feedback.

[Refer figure 4.11: Markings on button "5"**]**

This implies that the tactile, sound feedback of the buttons, voice feedback – telling the user where he is and what he has entered, reducing the number of steps to access different applications can make a blind user's phone experience better.

5.4.2. How a blind person use Physical Keyboards on a Computer

The visually challenged user first finds the physical markings on the keyboard, these markings act as the starting point for the user (dots on "F" and "J" on an InScript Keyboard, dots on "5" on Number pad), then he places his fingers on the keyboard with respect to "F" and "J", as he places his finger on the keyboard, he can create a model of where the other keys around.

Figure 5.3: A blind user navigating through the Desktop using different shortcut keys from the Keyboard.

Apart from the physical markings on keyboard, it also has voice assistance softwares like JAWS(English Screen reader software), ILTTS(Text To Speech in Indian Languages)that gives feedback in Malayalam to the user while typing on a physical keyboard. It reads out the screen as well as the user has entered. The user also uses some Keyboard shortcuts to open different applications like Microsoft Word, Excel etc. for eg. Alt + Cntrl + N for Microsoft Word, Insert + Cntrl + V for JAWS, Insert + Cntrl + S for ILTTS, Alt + O for formatting font for a Microsoft Word document etc. **[Figure 5.3]**

Each key is assigned with alphabets also, the user memorizes the alphabets on each key, also when the layout is changes to Malayalam, the user again has to memorize the new set of alphabets over the current keyboard. Each English alphabet key will be assigned with two Malayalam alphabets. When they change the language into Malayalam on an InScript keyboard, the user again has to memorize which alphabet is on what key. SHIFT is used to switch between the two Malayalam alphabets on the same key.

This implies that the visually challenged users should be given a starting point/ physical markings onscreen for helping them locate the keys easily, also voice feedback should be given regarding what the user is going to enter, as well as what the user has entered.

5.4.3. Challenges in using Touch phones

From the user studies, it was clear that there was only one user out of 9 who was using an iPhone. She used Voiceover by apple to navigate between the applications. She typed using English touchscreen keyboard available in iPhone.

*From **chapter 5(section 5.3.2.), chapter 4(section 4.2),** we can conclude that the reason why other users weren't using touch phone are that the user can't get a tactile feedback from a Touchscreen, The screen is all flat; everything is the same throughout the screen. Thus the user can't create a mental model of how to operate the touch phone; he doesn't know where to start. Other reason is that the Voice support is annoying, it reads out everything on the screen: Both the wanted and the unwanted. Virtual keys on the Touch phones are too small, and very sensitive. The probability of making an error is too high for a blind.*

This implies that if I can provide the tactile feel on a keyboard over a touchscreen, using the haptic feedback as well as the necessary Voice feedback for where the user is, What the user has entered, A starting point where the user can create and follow a mental model of the keyboard, The Touchscreen can be made usable for blind.

6. State of art review

There are many existing touchscreen keyboards that are made for the blind

6.1. Existing touchscreen keyboards for blind

6.1.1. Fleksy[11]

Fleksy is an input method for touchscreen devices which offers a traditional tap typing interface coupled with some gestures for common functions such as space, delete and word correction **[Figure 6.1]**. It uses error-correcting algorithms that take into account the exact locations where the user touches the screen and, coupled with a language model, tries to guess the intended word. Fleksy is designed for use on touchscreen devices with a traditional QWERTY keyboard, and provides the ability to touch type on a touchscreen device. Completely eyes-free typing is achieved by leveraging the user's muscle memory, making the software popular particularly among blind and visually impaired users.

For typing user has to create a mental model of a QWERTY keyboard in their mind, and type on the screen with the same model. The algorithm picks up the typed word using an algorithm by calculating the most probable words that can come out of that combination of buttons pressed. Here user doesn't have to be accurate in finding the right button while typing. Based on the input, most probable words will be predicted, and user can select them. For adding space, user has to slide their finger on the screen towards right and for deleting; user has to slide their finger on the screen towards the opposite direction.

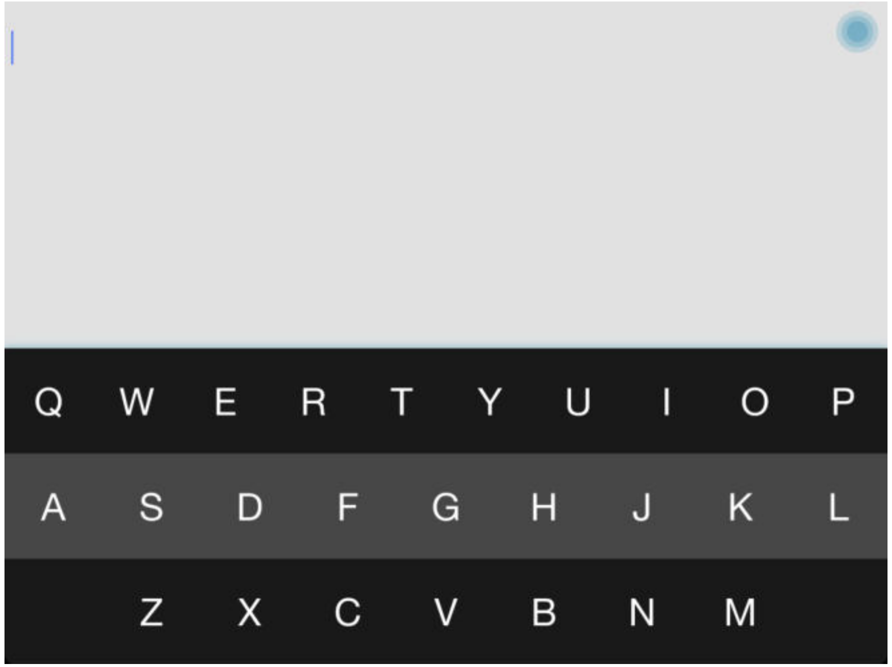


Figure 6.1: fleksy keyboard

Every time space is added while typing, the application reads out the last word/predicted list of possible words with that combination. User can slide their finger vertically down or up to navigate through the predicted list.

6.1.2. VoiceOver[12]

VoiceOver on iOS interacts with the user by using various "gestures", - different motions one makes with one or more fingers on the display. Many gestures are location-sensitive - for example, sliding one's finger around the screen will reveal the visual contents of the screen as the finger passes over them. This enables blind users to explore the actual on-screen layout of an application. A user can double-tap - similar to double-clicking a mouse - to activate a selected element, just as if a sighted user had tapped the item. VoiceOver can also turn off the display but leave the touch screen sensitive to touch, saving battery power. Apple calls this feature "Screen Curtain". The device can also be configured so that VoiceOver can be toggled by a triple-click of the Home button on the device.

For inputting the text, the user can slide over the touchscreen QWERTY keyboard **[Figure 6.2]**, it reads out the keys and double tap is used as a gesture for entering the alphabet. There is a difference in tone when the voiceover speaks out where the user is, and what the user has entered.

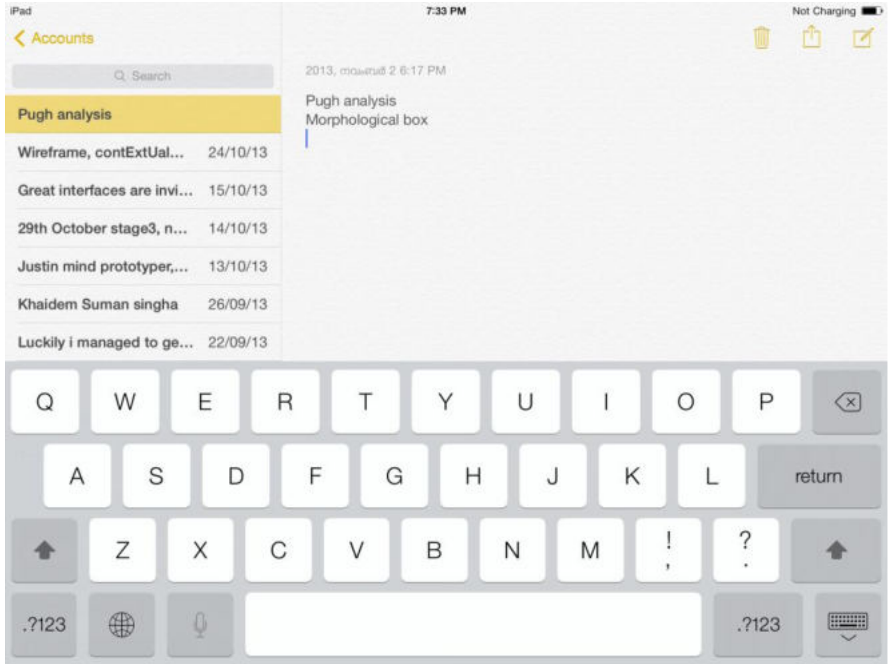


Figure 6.2: Inputting text using Voiceover

6.2. Comparing the State of art with findings from user studies

The two keyboards above are considered as the best rated touchscreen keyboards available in the market. Fleksy is available for both android as well as iOS devices whereas VoiceOver is only available for iOS. The main difference between these two keyboards is, Fleksy calculates the keys pressed and gives predicted list of the most probable words that can come from those key locations & combinations whereas, VoiceOver is a normal keyboard which speaks out the alphabet when user slides over it. Double tap is used as a gesture to input the character.

Fleksy doesn't tell the user where he is or what he has typed. But once user adds space, the last entered word/predicted list of words will be read out; whereas VoiceOver only helps user in letting him know where he is (On which button, user's finger is) and what he has entered. The kind of gestures used in fleksy are more intuitive and easier to use but the main disadvantage for fleksy is that it won't work that efficiently in small screen resolution phones.

Both the keyboards encourage thumb based typing but the main disadvantage of using VoiceOver is that, it reduces the speed of typing as user has to make sure he is about to type the right button all the time.

In this project, to build a touchscreen keyboard for blind, I should strictly follow on the following principles:

- **Giving appropriate feedbacks to the user**- Where he is, what he has typed. E.g.: voice, haptic etc.
- **Use interactions that are more intuitive to blind users**- for example, 'delete' and 'space' gestures used in Fleksy.
- **Screen resolutions**-The keyboard should run on any screen resolutions without any hassle. The letters should be visible and the number of alphabets onscreen should be placed with enough space in between to avoid errors.
- **Follow logical way of approach**- The layout should be more meaningful to the user; user should always know what lies ahead in the next button. This can dramatically increase the user's speed in typing.
- **Learn to type with less training**- User doesn't have to go through a very hard and rigorous training that consumes a lot of time to learn how to type on the keyboard. It should be easy and any blind user should be able to pick it up in an hours' time.

7. Design Brief

Any Malayalam user, who is visually challenged, should be able to use this input system. The system should be in such a way that the user needs only lesser keystrokes to input, edit text. The following goals were set after interviewing the visually challenged users and observing their interaction with the mobile devices.

1. **Reduced cognitive load while inputting the text**
   The layout should be more meaningful to the user; user should always know what lies ahead in the next button. This can dramatically increase the user's speed in typing.
   a. **Intuitive interactions**
      The interactions should be intuitive; user should be able to perform actions like 'deleting' or adding a 'space' without an effort.
   b. **Finding starting point on the keyboard**
      User should be able to find the starting point on the keyboard easily. **[Chapter 4, section 4.3]**
   c. **Avoid errors due to virtual key sizes**
      The minimum size of the buttons should be around 45 to 57 pixels i.e. the minimum size of button should be 45X45 pixels. **[Chapter 5, section 5.3.2.3]**

2. Providing necessary feedbacks for better interaction with the visually challenged user and the touchscreen
   a. **Tactile feedback**
      Adequate haptic feedback should be given to the users while interacting with the interface. This helps user in locating the starting point
   b. **Voice feedback**
      Voice feedback should be given regarding what user is going to enter, what user has entered, deleting a character/word and adding space
   c. **Sound feedback**
      For supporting user with haptic feedback, adequate sounds should be given when user moves from one button to another For example, if user moves from one button to another, the corresponding sound feedback can add to the tactile feedback of the button along with haptic feedback.

3. The time taken should reduce as the Novice user transforms into an expert.

8. Concepts

8.1. Ideation

8.1.1. Design idea #1

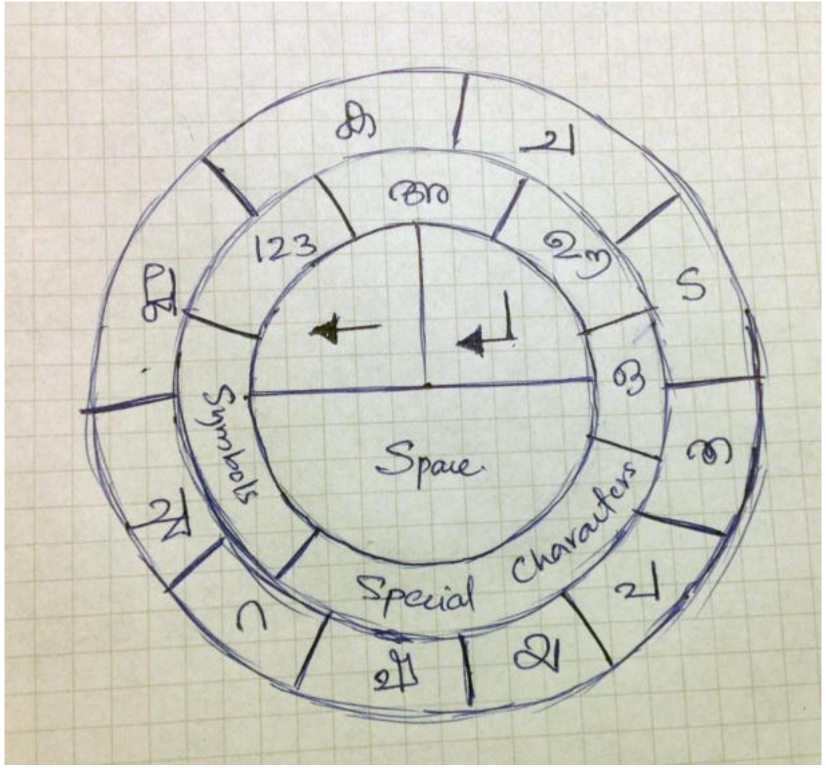


Figure 8.1: Idea 1

The alphabets were chunked on the basis of their *vargam* in circular menus. Here ക stands for ക, ഖ, ഗ, ഘ and ങ. It gives voice feedback as the user slides over each of the alphabets *[Design Brief, Chapter 7, section 2.b.]*. The user can double tap to initiate the submenu. The user has to move in circular path to access the alphabets in the sub menu. Spacebar, backspace and enter buttons were located to the center of the keyboard Layout. The vowel symbols, special characters and numbers were also represented in the same layout and could be activated using a double tap on the respective button leading to a submenu with the list of symbols or special characters or the numbers**[Figure 8.1]**.

The only advantage of this layout is that the alphabets are chunked logically based on their *vargam. [Design Brief, Chapter 7, section 4].* This reduces size and the number of alphabets on the interface reduce significantly*[Design Brief, Chapter 7, section 1.c.].*

The disadvantage of using this keyboard was, the blind user is subjected to a lot of ambiguity and uncertainity as he starts navigating through the interface. For the blind user, moving through the narrow circular paths will be a tedious task and this will invite errors.

8.1.2. Design idea #2

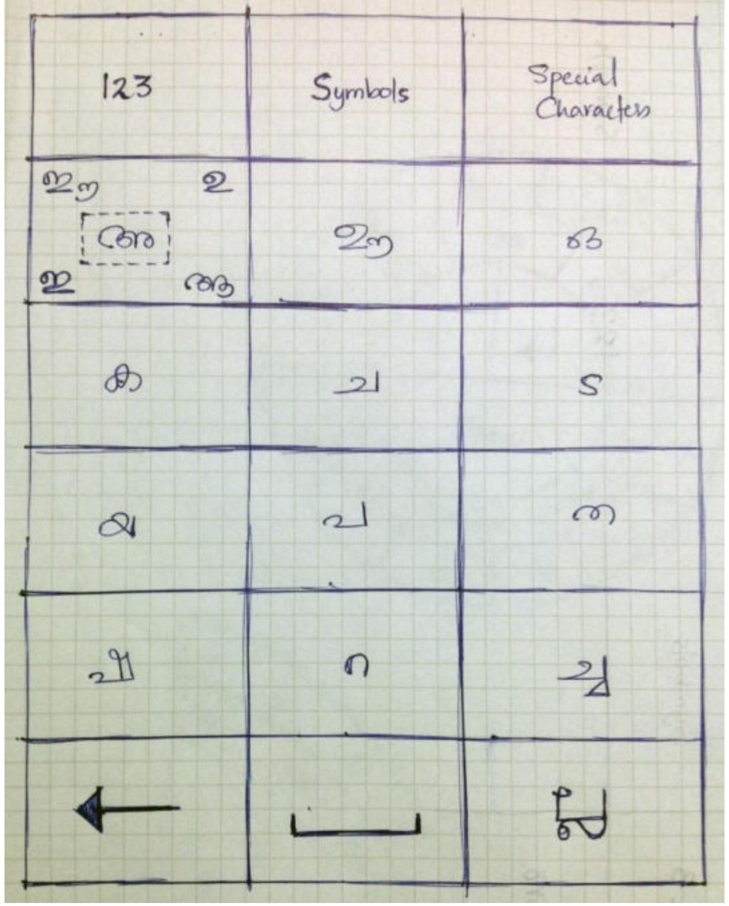


Figure 8.2: Idea 2

The second idea followed a box layout with bigger keys and Alphabets. Each key had five alphabets in it and they are arranged in a sequential Zigzag manner thereby helping the user create a mental model of the arrangement of keys **[figure 8.2]**. Each key had 5 alphabets in it, with the frequently used alphabet as parent; four other alphabets were arranged to the edges of the button. They get activated as the parent alphabet is tapped twice.

Backspace and space buttons were located towards the end of the interface. There are menu buttons on the top for switching the keyboard layout into numbers, symbols and special characters.

This design also followed the same interactions as the Idea 1. It reads out the alphabet as the user slides over the key, after entering the alphabet the voice replies the alphabet has been entered [*Design Brief, Chapter 7, section 2.b.]*.

The advantage of this design was large keys and sequential arrangement of alphabets can help the blind user to build a mental model. The disadvantage was every time the user had to type a letter; his finger had to travel a long distance over the keyboard.

8.1.3. Design idea #3

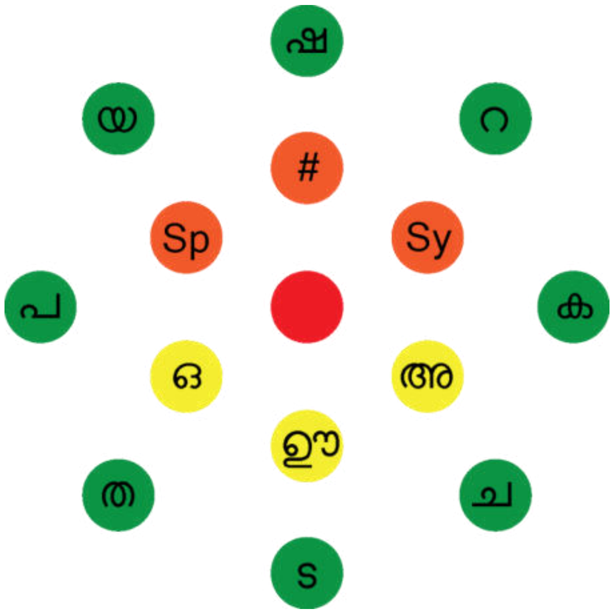


Figure 8.3: Idea 3

Instead of user finding the keyboard on a non-tactile keyboard, why can't the keyboard find the user? *[Design Brief, Chapter 7, section 1.b.].* Why can't the point where the user touches become the starting point? All the alphabets are located in proximity with the starting point. This was a dynamic layout and it floats to where the user touches the screen. **[Figure 8.3]**

The vowels and consonants and were arranged in a circular manner around the centre of the keyboard. The blind user could move in both ways, circular path as well as the in different directions from the center of the keyboard. This layout also follows the chunking of alphabets based on its *vargam.* The advantage of this layout was it could float with the finger and the disadvantage was that the navigation through the circular menu was limited due to the smaller area of the buttons.

8.1.4. Design idea #4

Figure 8.4: Idea 4

This layout also follows Box layout, with Physical markings over the touchscreen that can enhance the tactile experience of a user. Here all the alphabets are logically listed down, the vowels, consonants, numbers and special characters are all chunked in a menu that comes on the top of the layout *[Design Brief, Chapter 7, section 4].* Advantage of using this design is that the keys are large and no submenu is required for the text entry. The disadvantage is that the finger has to travel a long distance **[Figure 8.4]**.

8.1.5. Design idea #5

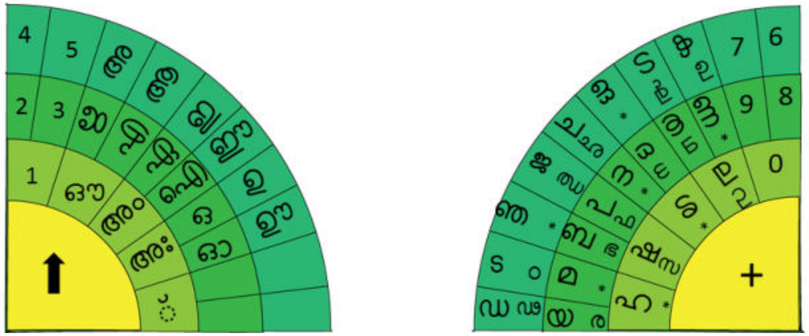


Figure 8.5: Idea 5

This layout has split keyboards, vowels on left and consonant on the right side. It follows a frequency based (arranging the alphabets on the layout on the basis of its frequent use in the language) as well as logical approach *[Design Brief, Chapter 7, section 4]* **[Figure 8.5]**. Shift keys as well as key for the construction of consonant ligatures are also added on the keyboard. Space and backspace are gestures like swiping across the touchscreen in different directions *[Design Brief, Chapter 7, section 1.a.]*. Voice feedback and sound feedback are given to find the key and after entering the key. The advantage is that, reducing the number of alphabets per finger **[Figure 8.4]**, the disadvantage of this design is that the load on the thumb finger is high.

8.2. Initial Concepts

After ideation, three Ideas were selected for theoretical modeling.

8.2.1. Concept #1

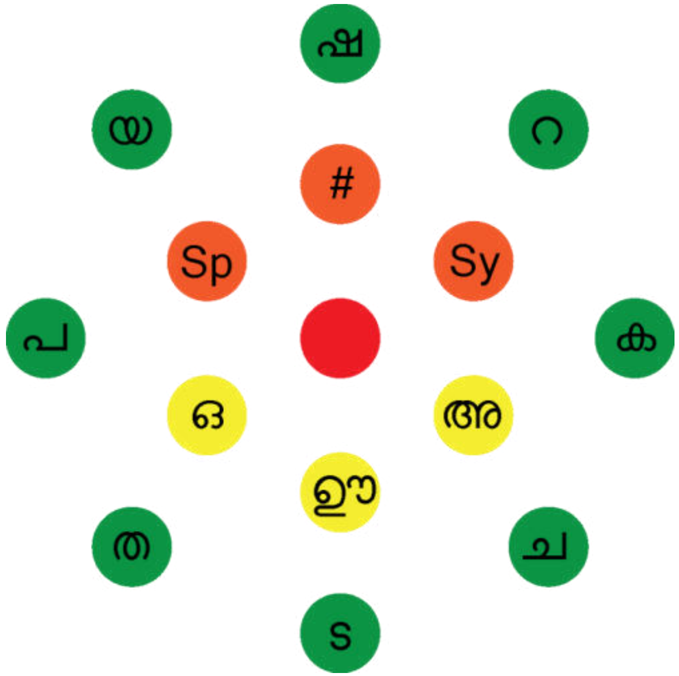


Figure 8.6: Concept 1

Sp: Special Characters, #: Numbers, Sy: Symbols

The keyboard **[Figure 8.6]** consists of two menus, the vowel and consonant buttons, the user has to touch anywhere on the screen for 5 seconds to activate the keyboard.

The keyboard will be activated with a vibration feedback. Then the user has to short drag from the center towards any direction to activate the vowel menu. Likewise, long drag for the consonant menu.

Each menu has a different vibration feedback to make sure that the user knows where he is.

Each key on the main menu had a submenu with 5 alphabets. The alphabets were chunked in such a way that the number of alphabets on the screen will be reduced and the user doesn't have to go through all the keys to find the desired alphabet.

Voice feedback and haptic feedback were given to assist the blind user in navigating throughout the keyboard. As soon as the user slides over the key, it reads out the alphabet.

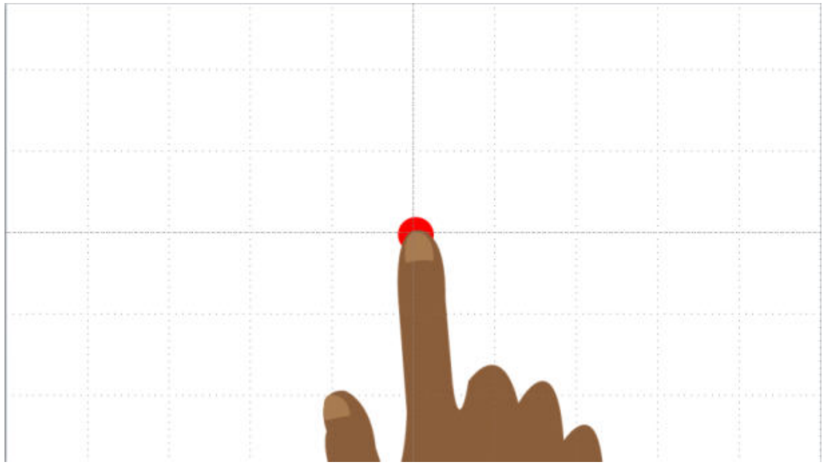
The user had to perform a double tap to go into a sub menu.

The space can be entered by swiping across the screen towards right, and backspace in the opposite direction.

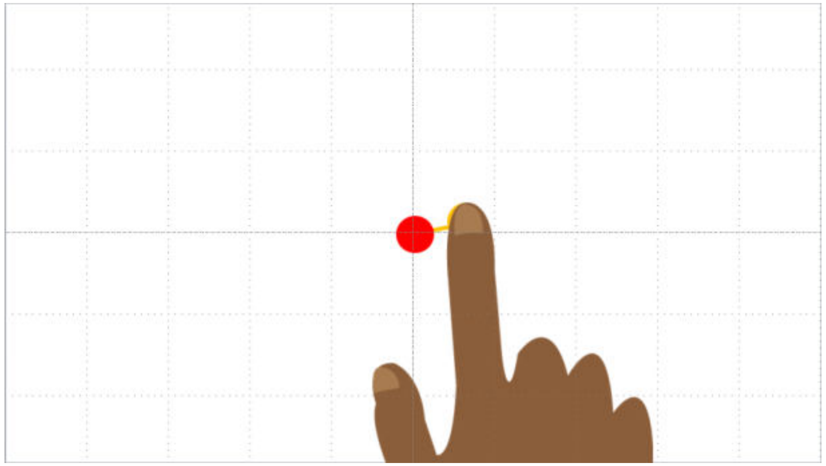
Activating the menu

Activating the menu can be better explained in the following steps.

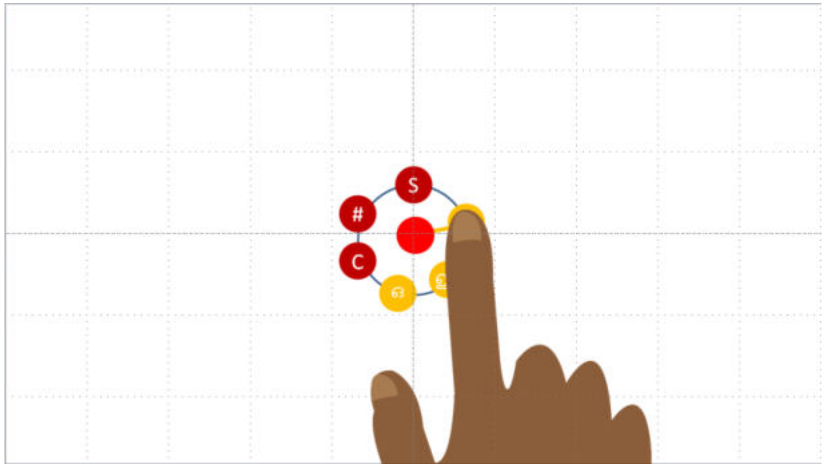
Step 1: Touch anywhere on the screen for more than 2 seconds



Step 2: shortdrag in any direction to activate the vowel menu



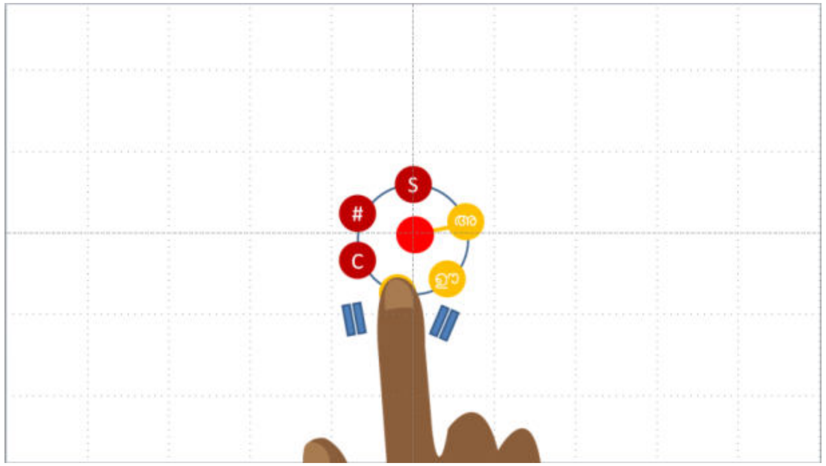
Step 3: Move in circular paths to navigate



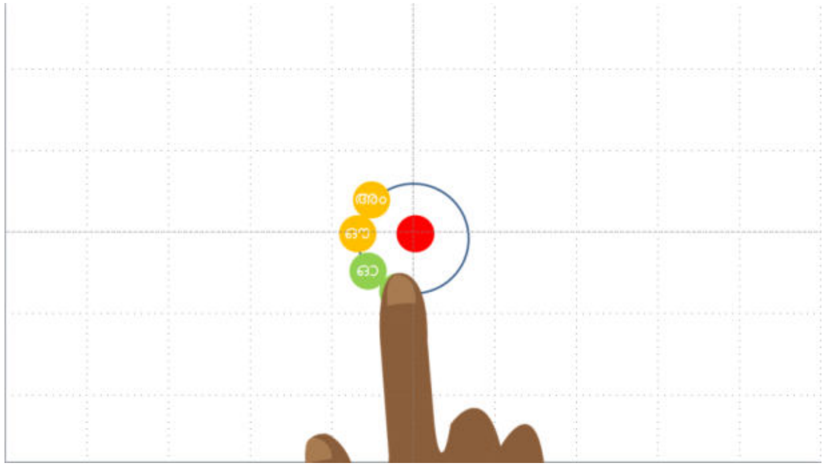
Note: S stands for Symbols(Sy), C stands for Special Characters(Sp), and # stands for numbers

Activating the sub menu

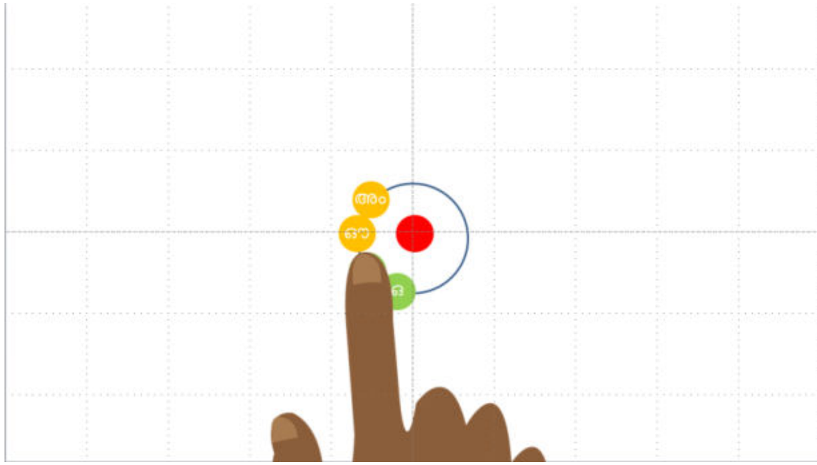
Step 1: Double tap to enter



Step 2: Entering the submenu



Step 3: Release to enter the vowel.



To activate the vowel modifiers, the user have to double tap at the starting point itself, the vowel menu changes into Vowel modifier menu.

Advantages

- Wherever the user touches, that point will become the starting point of the keyboard.
- All alphabets are at close proximity around the starting point
- Alphabets are arranged in a sequential manner (since it's in a circular shape, after the last key, user moves to the first key), so the navigation becomes easier

Disadvantages

- The user had to create a mental model where each alphabet was located.
- There was some ambiguity or disconnect formed while the user navigates through the menus because there were many empty spaces between each alphabet.
- Area for each button key was less. Since the behavior of the blind user towards the interface can't be predicted, there is a need for more button area.
- Won't facilitate the horizontal and vertical movement of finger as the keyboard is circular in shape.

Figure 8.7: Concept 1

8.2.2. Concept #2

In this keyboard, the logical way of arranging the keys were followed. All the alphabets were listed out in the order of their occurrence. The same way students are taught Malayalam at school.

The vowels, vowel modifiers, Numbers, Symbols etc. were chunked separately on a menu. The menu and the keys were differentiated using physical markings over the screen for the tactile feedback **[Figure 8.7]**.

Voice feedback and haptic feedback were also provided for supporting the user navigation.

Scenario

Step 1: The user should follow the physical markings onscreen for locating the keys.

Step 2: The user can follow any path to locate the alphabet

Advantages

- Big buttons are used; hence it's less error prone.
- As the alphabets are listed out in a logical way, it will be easier for a blind user to use the keyboard.

Disadvantages

- Too many keys, too much feedback. Chances that the user can get annoyed is high
- User has to move a lot through the interface to switch between the menus.

8.2.3. Concept #3

This is a split keyboard for typing with the thumb **[Figure 8.8]**. It follows a layout similar to the Inscript but this keyboard is also logically based **[Figure 8.7]**. It has two alphabets on a single key and a shift key to switch between them. *Chillus* and Conjuncts are removed and hence there is a new operation that is included "+", this operation triggers the addition of two alphabets, to form a conjunct, a glyph, or a *chillu*. When the particular combination is entered using the "+" operator, the system will detect the combination, then replace it with a conjunct, glyph or a *chillu.*

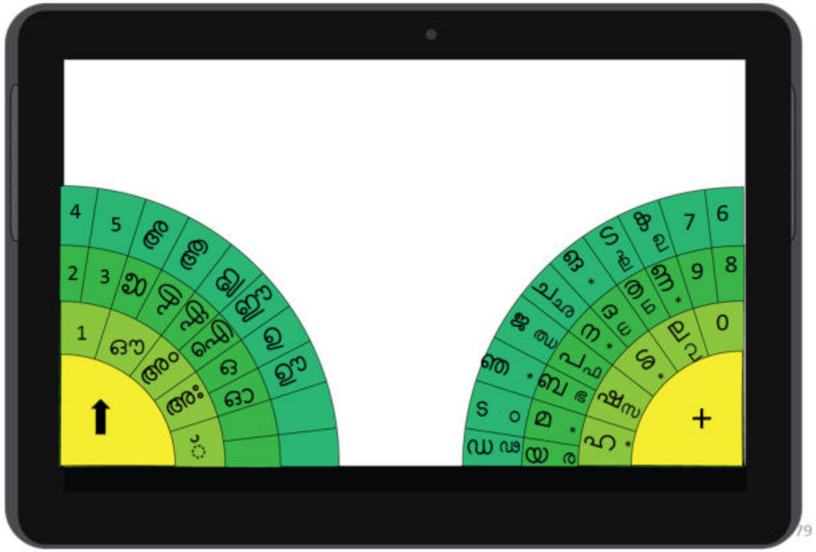


Figure 8.8: Concept 3

Advantages

- Lesser keys than the other two designs
- Easier to use as two thumbs are involved in typing
- Shift button to switch to the both alphabets on the same key

Disadvantages

- Load on thumb is very high.

8.3. Arranging alphabets on Keyboard

To represent the number of alphabets on the Keyboard, three concepts were formulated.

- All alphabets can be entered.
- Remove *chillu* and conjuncts, replace the combinations with *chillu* and conjuncts after the particular combinations are entered.
- Provide the first letter of each *vargam,* thereby reducing the number of alphabets on the keyboard for easy navigation.

8.3.1. Concept #1

The Concept#1 **[Figure 8.9]** was chosen, and iterated further for removing the identified flaws.

The identified problems were:

- The user had to create a mental model where each alphabet was located.
- There was some ambiguity or disconnect formed while the user navigates through the menus because there were many empty spaces between each alphabet.

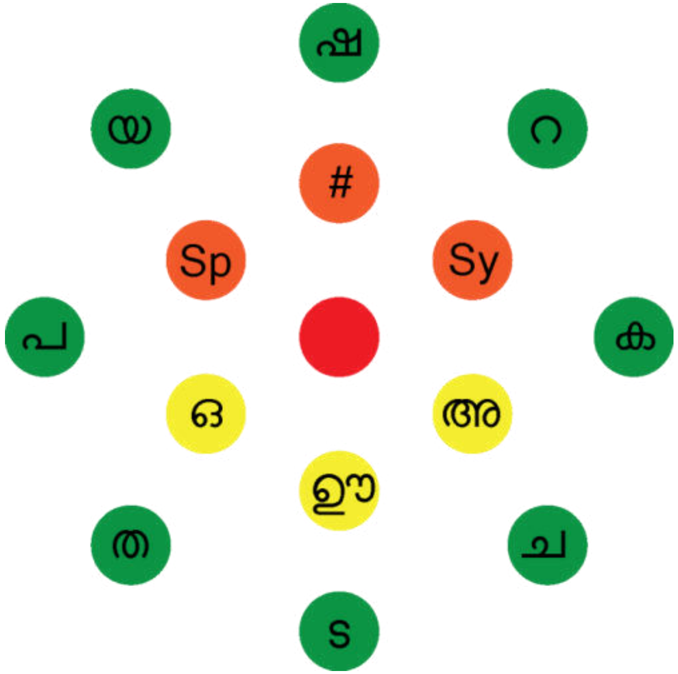


Figure 8.9: Concept 1

- Area for each button key was less. Since the behavior of the blind user towards the interface can't be predicted, there is a need for more button area.
- Won't facilitate the horizontal and vertical movement of finger as the keyboard is circular in shape.

Step 1:

It was found that the area of button was less in the initial concept. So to retain the same layout and increasing the area, a chakra layout **[Figure 8.10]** was put forward. *The circular shape is the most optimum shape for the keyboard as the keyboard elements will be equidistant from the center i.e. the elements will be equally arranged around the center or the starting point.*

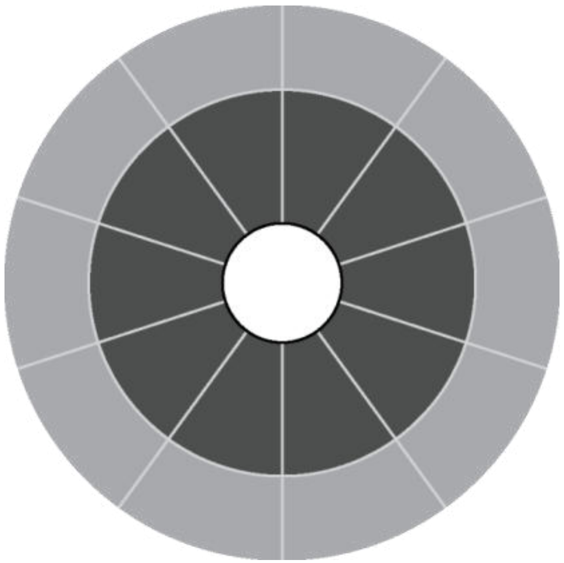


Figure 8.10: Chakra

Step 2:

In order to facilitate the horizontal and vertical finger movement, the circles were resized **[Figure 8.11]**.

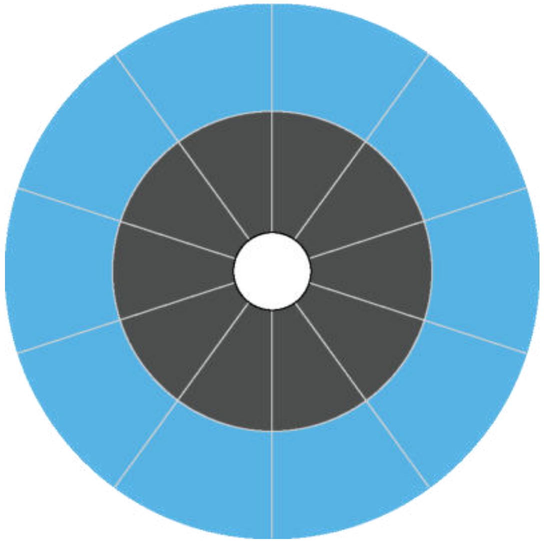


Figure 8.11: Resized Wheel layout

Step 3:

Studies say that the minimum diameter of a human finger pad is 16mm. when converted to pixels, it was 45X45pixels **[Figure 8.12] [6].** So this measurement was used as the minimum key size inside the layout.

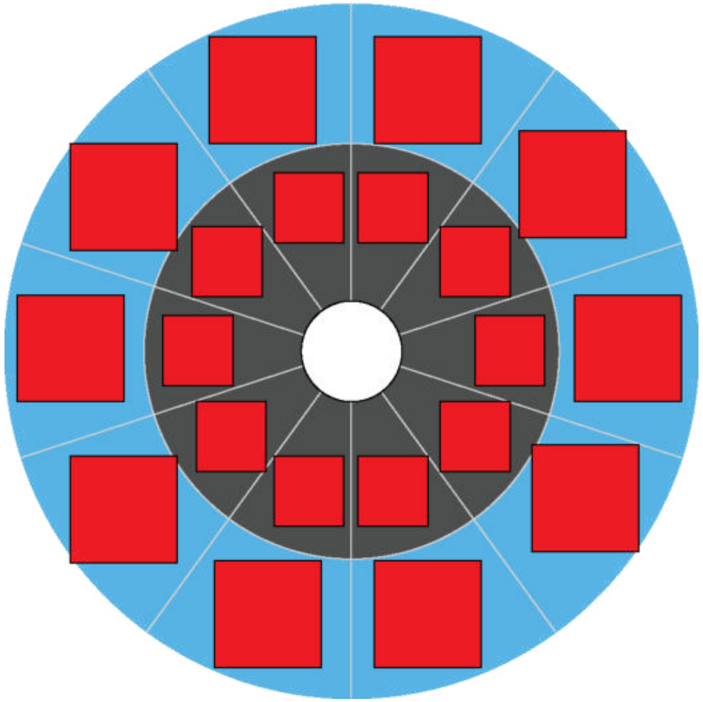


Figure 8.12: The small squares represent the minimum button size.

The layout was tilted a little bit so as to make sure that if user swipe vertically to the top, He can get access to the numbers '123'. Since numbers are much frequently used, it's necessary that there should be a unique gesture to get access to the numbers. So to access the consonants and vowels, user has to move towards right, and to top for accessing the numbers.

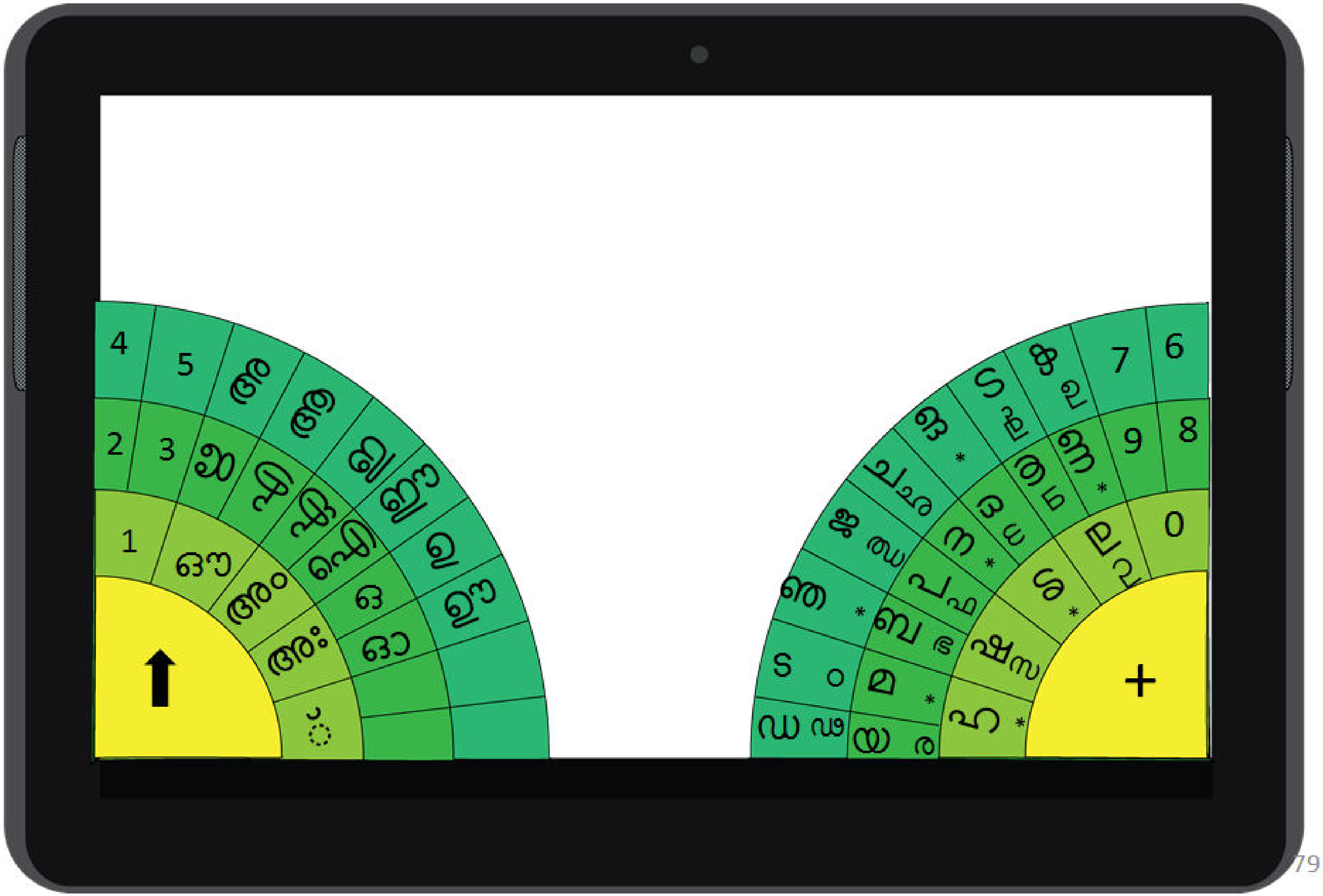
Figure 8.13: Final UI design for the Keyboard

8.4. Final Concepts:

8.4.1. Concept#1

8.4.2. Concept#2

8.4.3. Concept#3



9. Theoretical Modeling

The concepts were subjected to a theoretical testing where the three concepts were compared to the Malayalam Inscript Keyboard(Windows XP) using four words from Malayalam.

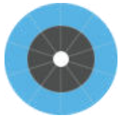
1. **മലയാളം** : A simple word with two vowel modifiers

2. **മുന്നറിയിപ്പ്**: A word with three vowel modifiers, one *Chandrakala* and two conjuncts

3. **അതിർത്തിക്കുള്ളിൽ**: A complex word with one vowel, four vowel modifiers, three conjuncts and two *chillu* alphabets

4. **അംഗഭംഗം**: A complex word with one vowel, four vowel modifiers, three conjuncts and two *chillu* alphabets

The number of key presses was calculated.

Key press	Concept 1	Concept 2	Concept 3	Inscript Malayalam (Windows XP)
മലയാളം	17	18	14	7
മുന്നറിയിപ്പ്	32	46	38	13
അതിർത്തിക്കുള്ളിൽ	36	52	50	21
അംഗഭംഗം	17	22	18	9
Total	**102**	**138**	**120**	**50**
Average	**25**	**34**	**30**	**12**

Figure 9.1: Finger movement across three concepts

10. Final concept

Based on theoretical modeling **[Figure 9.1]**, the number of key presses was calculated for each concept against four Malayalam words of different levels of complexity. The concept 1 was proved to be the fastest keyboard next to the InScript layout for physical keyboards.

Concept 1 is prototyped **[Figure 10.1]** for further evaluation with the blind users.

[Refer DVD for the demo video and prototype]

Figure 10.1: User Interface for the prototype.

11. Evaluation

11.1. Evaluation plan

Based on theoretical modeling, the three concepts were compared on the basis of the number of key presses. It is certain that the Concept 1 is faster than the other concepts. Now the following attributes needs to be measured:

1. **Overall effort,** i.e. the number of keys to be pressed, finger load.
2. **Learnability**, How fast a user can learn to type
3. **Accuracy**, i.e. the number of errors, the number of accurate attempts, Key Accuracy etc.
4. **Speed:** words per minute.

The other goals of this evaluation are to test the prototype with the blind Malayalam users for usability and document the findings for future reference.

11.2. Evaluation process

Step1: Training task: One user was given a task to type 100 complex Malayalam words (with *chillu,* consonant ligatures, Vowel modifiers, all of them together) in 3 hours, blind folded. Other users were given training only for 30 minutes.

Step 2: Speed and Accuracy: 3 users were asked to use the keyboard blind folded, to type a paragraph in Malayalam containing 21 words.

Step 3: Think aloud: one completely blind user was recruited to do a think aloud test on the design.

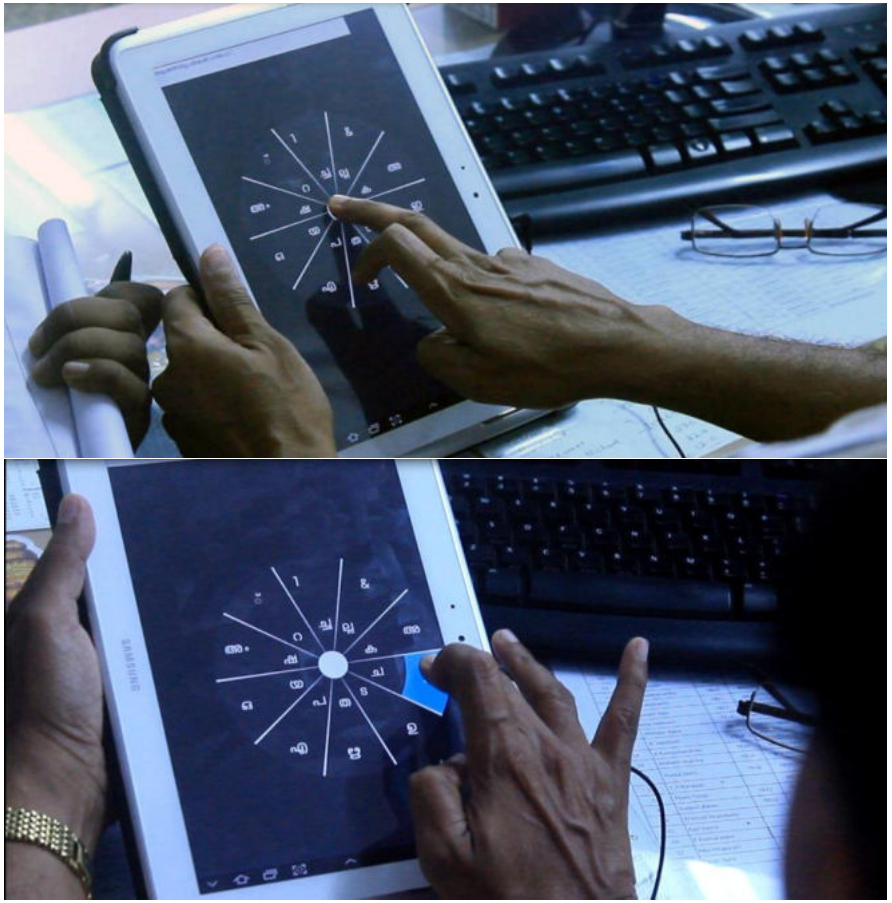


Figure 41: Testing with blind folded Malayalam users

11.3. Results

The following table shows the summary of the tests done.

User	Training time	Result	Speed	TTS support	Errors
User 1	3+ hrs	21 words in 19 : 43	1.05 word/minute	No	6
User 2	30 mins	11 words in 21 : 06	0.52 word/minute	No	9
User 3	30 mins	11 words in 34 : 21	0.3 word/minute	yes	0

User who got trained for 3+ hours with 100 words could type at a speed of 21 words in 19:43 minutes with 6 errors. i.e. 1.05 words per minute.

Other users who got trained for 30 minutes each could only finish the task with 0.52 words per minute and 0.3 word per minute.

The third user was provided with 'text to speech' manually by reading aloud what user had just entered.

The main cause of errors that happened during the testing process is:

- **Lack of TTS (text to speech) support**
  The TTS or text to speech has an important role to play in this keyboard. Unless the user is given adequate feedbacks on what he has entered, and the user understands whether his input is error free, he cannot proceed any further. During the testing process, most users faced this difficulty and the error rate was also high. Users were given feedback on what they have entered in the later testing process. User 3 achieved error free typing based on the TTS provided through oration.

- **User couldn't get feedback what he has deleted**: When user enters a character and he wants to delete it, he needs to swipe the screen towards left side. User gets a feedback telling "deleted last letter", but it doesn't tell user what he has deleted.

11.4. Think aloud results

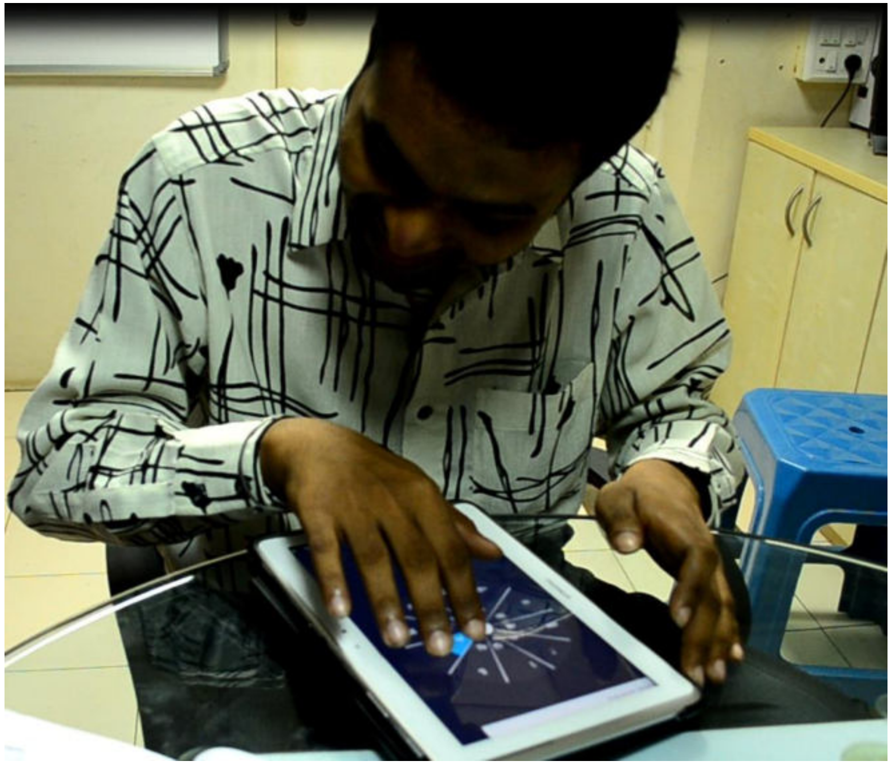


Figure 42: Testing the keyboard with a completely blind user

The think aloud test with a completely blind user revealed the following problems:

- **Hints for the interactions:**
  User complained that there should always be a "help" button that can give hints about how to interact with the keyboard and what are all the necessary interactions user needs to know while typing on the keyboard.

- **Circular layout:**
  Getting familiar with circular keyboard alone takes time; Blind users who are blind from birth cannot create a mental model how a circle looks like. Hence they have to follow the haptic feedback closely while moving through the outer circle to stay inside the keyboard.

- **"tak", "tak" sound feedback sound for buttons:**
  User complained that while moving through the letters on the keyboard, they should have a sound feedback like "tak", "tak" as they felt that the vibration feedback is not enough to create that tactile feel.

- **Need feedback for the last letter entered:**
  After adding space, there should be a voice feedback that tells what the user has entered.

- **Need feedback for the last letter that is deleted:**
  The last deleted letter should also be notified while deleting. This tells user that what is deleted, what is left to be deleted and what should not be deleted.

- **Switch between different modes:**
  User should be able to switch between different modes like consonant mode, vowel mode, number mode etc. If so, user doesn't have to worry about having two layers with consonant on one and vowels on the other one.

12. Conclusion

From the evaluation, it's clear that the keyboard will not work without the help of TTS (Text To Speech) output. If TTS can be added to Mudra, it's a good tool for the blind users to type in Malayalam.

The users who are completely blind from birth cannot visualize how a circle looks like. So, following a circular keyboard to type in Malayalam needs a good amount of training. User has to follow the haptic feedback to move through the interface. Mudra can fit any screen sizes.

12.1. Future scope

Mudra can be used by both sighted and non-sighted users alike since it follows the logical layout and has enough feedbacks to support a blind user. If the blind can type, this opens the doors to the possibility that the illiterate in India can also type using the same keyboard. The third option is that even sighted users can type without looking at the touchscreen.

13. Reference

1. Eye problems - major concerns by Dr. Anuja. U, Assistant Professor of Community Medicine, Medical College, Trivandrum.
   Date: 9/11/2013
   Source:http://www.commedtvm.org/phus/phu09_session%201.html

2. Blind literacy by National Initiative for the Blind(NIB), Tamil Nadu
   Date: 16/11/2013
   Source: http://www.ashanet.org/projects/project-view.php?p=503

3. Definition: visual impairment by Ms. Karin van Dijk, New Vision Consultant
   Date:9/11/2013
   Source: http://www.bpaindia.org/VIB%20Chapter-I.pdf

4. Visual Acuity: What is 20/20 Vision? By American Optometric Association
   Date: 9/11/2013
   Source: http://www.aoa.org/patients-and-public/eye-and-vision-problems/glossary-of-eye-and-vision-conditions/visual-acuity

5. Design and Evaluation of Devanagari Virtual Keyboards for Touch screen Mobile Phones by Anirudha Joshi, Girish Dalvi, Manjiri Joshi, Prasad Rashinkar and Aniket Sarangdhar
6. 3-D Finite-Element models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense by Kiran Dandekar, Balasundar I. raju, Mandayam A. Srinivasan*, The Touch Lab, MIT
   Date: 9/11/2013
   Source: http://touchlab.mit.edu/publications/2003_009.pdf

7. Keypad for Large letter-set Languages and Small Touch-Screen Devices(Case Study: Urdu) by Asad Habib, Masakazu Iwatate, Masayuki Asahara, Yuji Matsumoto.
   Date: 11/11/2013
   Reference: http://ijcsi.org/papers/IJCSI-9-3-3-47-58.pdf

8. Malayalam script: Adoption of New Script for Use-Orders Issued by Government of Kerala
   Source: Date: 11/11/2013
   http://www.malayalamresourcecentre.org/Mrc/order.pdf

9. Malayalam alphabet: by Wikipedia
   Date: 11/11/2013.
   Source: http://en.wikipedia.org/wiki/Malayalam_alphabet

10. Design and Evaluation of Devanagari Virtual Keyboards for TouchScreen Mobile Phones by Anirudha Joshi, Girish Dalvi, Manjiri Joshi, Prasad Rashinkar and Aniket Sarangdhar

11. Voiceover by Wikipedia
Date: 18/11/2013
Source:http://en.wikipedia.org/wiki/VoiceOver

12. Fleksy by Wikipedia
Date: 18/11/2013
Source:http://en.wikipedia.org/wiki/Fleksy

13. Consonant Ligatures by Wikipedia
Date: 18/11/2013
Table reference:
http://en.wikipedia.org/wiki/Malayalam_alphabet

14. Snellen Chart image used under Creative Commons Attribution-Share Alike 3.0 Unported license.
Date: 18/11/2013
Image Reference:
http://en.wikipedia.org/wiki/File:Snellen_chart.svg

15. Third stage in the development of Malayalam as a language: **Yavvana avastha** / AD 1625 onwards by Prof. A R Rajaraja Varma,
Date:10/7/2014
Source: http://ml.wikipedia.org/wiki/Malayalam

16. Beginning and initial growth phase of Malayalam Wikipedia
Date:10/7/2014
Source: http://en.wikipedia.org/wiki/Malayalam_Wikipedia

17. A Comparative Grammar of the Dravidian or South Indian Languages by Robert Caldwell, 1875
Date: 11/7/2014
Source: http://en.wikipedia.org/wiki/Malayalam#cite_note-Clad-17

18. Analysis of the Indic Language Wikipedia Statistical Report 2012 by Shiju Alex (from Malayalam Wikipedia, http://meta.wikimedia.org/wiki/User:Shijualex)
Date: 11/7/2014
Source: http://shijualex.in/analysis-of-the-indic-language-statistical-report-2012

19. Malayalam standardization by Standardization Committee, 2001, Princeton University, NJ, USA.
Date: 11/7/2014
Source:
http://www.cs.princeton.edu/~mp/malayalam/keyboard/malayalam%20standardization%20report.pdf

20. Usage of content languages for websites, Survey by W3techs,
Date: 16/7/2014
Source:
http://w3techs.com/technologies/overview/content_language/all

21. New Malayalam InScript Keyboard by C-DAC
Date: 17/7/2014
Source:
http://malayalamresourcecentre.org/malayalamkeyboard/index.html

22. Chillus in Unicode by Wikipedia
Date: 17/7/2014
Source:
http://en.wikipedia.org/wiki/Malayalam_script#Chillus_in_Unicode

23. Malayalam Unicode Chart
Date: 20/7/2014
Source: http://www.unicode.org/charts/PDF/U0D00.pdf

24. Indic keyboard Prime Malayalam keyboard by androidtweak.in
Date: 23/7/2014
Source:
https://play.google.com/store/apps/details?id=in.androidtweak.inputmethod.indic

Special Thanks,

Prof. Anirudha Joshi and Prof. G. V. Sreekumar *for the guidance and Motivation*

Prof. Girish Dalvi *for teaching me how to do the Theoretical Modeling*

Manjiri Joshi *for writing Java code for Mudra*

Tanushree Paul and Shantanu *for recording the Malayalam voice*

Sujit Devkar, Researcher at TCS Innovation lab *for sharing his experience in interviewing the blind*

Mr Josy Joseph, General Secretary of National Association for Blind, Trivandrum, Kerala

Annie Teacher, NAB, Trivandrum *for introducing me to her students*

Susamma Alexander *for teaching me Structure of Malayalam Script*

All the blind students at NAB, Kerala *for their cooperation*

Shwetha Kamble, Snr VC *for assisting me to NAB and BB, Mumbai*

Manisha Swarnkar, Snr Animation *for editing voice clips for my initial prototypes*