ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

DESIGN AND IMPLEMENTATION OF VIRTUAL KEYBOARD FOR AMHARIC TEXT ENTRY ON PDA SYSTEM

By

Gedion Tamene

A Project paper submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science

June, 2008

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

DESIGN AND IMPLEMENTATION OF VIRTUAL KEYBOARD FOR AMHARIC TEXT ENTRY ON PDA SYSTEM

By

Gedion Tamene

Name and Signature of members of the Examining Board:

NAME	SIGNATURE
1. Dr. Solomon Atnafu, Advisor	______________________
2.	______________________
3.	______________________

Acknowledgement

First and for most I would like to express my gratitude to my advisor, Dr. Solomon Atnafu, for deep concern, constructive comments and suggestions he has given from the beginning to the end of this project.

I would like also to extend my very sincere and special thanks to Ato Menassie Zewdu, who provided me the necessary assistance during my work.

Lastly, my special gratitude and respects goes to my friends, who supplied me with all the necessary information and valuable comments.

Table of Contents

List of Tables

List of Figures

Abbreviation/Acronym

ADT=Android development Tools

AVKB=Amharic Virtual Keyboard

WPM=Words per Minute

VM=Virtual Machine

PDA=Personal Digital Assistant

OHA=Open Handset Alliance

SDK=Software Development Kit

EICTDA=Ethiopian Information Communication Technology Development Agency

IDE=Integrated Development Environment

OS=Operating System

Abstract

PDAs have become increasingly popular due to advances and availability of wireless services. The desire for effective text entry on portable devices have also increased with the advent of virtual keyboard being as one of the major contributors to the sudden burst of research in handheld device text entry techniques. Virtual keyboard is the one which employs on screen keyboard where programmable graphical buttons are handling the input activity based on the event triggered on them. In order to design virtual keyboard one has to consider nature of the language where by the text entry mechanism is achieved. In this project the Amharic language is chosen for the layout of the virtual keyboard to be designed. Amharic is the working language of the federal government of Ethiopia and is spoken and written as a first or second language in many parts of the country. The abundance of characters in Ethiopic script brings a challenge for developing Amharic virtual keyboard on handheld devices.

The objective of this project is to address the design and implementation of the virtual keyboard for Amharic text entry on PDA devices. For this purpose the layout design for Amharic virtual keyboard is proposed and an Amharic virtual keyboard is developed by taking inherent challenge of Amharic language in to account. The developed virtual keyboard is implemented for a simulator of an Android operating system PDA. Android OS is selected for the implementation considering its possible future popularity and usage for mobile handheld devices such as PDAs, smart phones, mobile phone etc. The system is also designed in such a way that Amharic and English virtual keyboards can be switched to enable multi-script text entry.

Chapter One

Introduction

1.1 Overview

In computing world, there are various methods for rendering text input to the computer. The existing text input methods have five main branches: keyboards, text recognition, unistrokes, speech recognition and gesture recognition. The most popular method for transferring text from human mind into the memory of the computers is the keyboard and two of its branches are the physical and the virtual keyboards. One form of virtual keyboard is a software component that allows a user to enter characters by tapping graphical keys displayed on touch screens with a stylus. Where as the physical keyboard is the one which is commonly used with desktop computers and has set of physical keys where hardware assigns a binary scan code value to each key [17].

These days, computing devices are getting ever-smaller, and the need for having smaller keyboards is increasing. Simultaneously keyboards have shown yet more evidence of their attractiveness since they are very friendly to the users. And this time, like many other things, they have moved to virtual worlds. Of all possible text input methods, virtual keyboards displayed on a touch screen seem to be the best choice in many situations [11].

A list of handheld devices that include Smart phones, PDAs, and Mobile phones uses virtual keyboards as a means of text entry. Virtual keyboard for languages like English, French, Arabic, Japanese etc. has been built and is available on the market. This situation encourages the speakers of the aforementioned language to use the mobile devices, especially PDAs, extensively.

A **PDA** is a handheld computer. PDAs are also known as **small** computers or palmtop computers. They provide services like: calculation, clock, calendar, Internet access, intranets or extranets access via **Wi-Fi**, or Wireless Wide-Area Networks (WWANs) access, sending and receiving E-mails, video recording, typewriting and word processing, address book, writing on spreadsheets, scanning bar codes, radio or stereo access, playing computer games, recording survey responses, and Global Positioning System (GPS)[12]. Newer PDAs also have both color screens and audio capabilities, enabling them to be used as mobile phones (smart phones), web browsers, or portable media players.

Most PDAs manufacturers, such as the Apple Newton and the Palm Pilot, featured touch screens for user interaction, that have only a few buttons usually reserved for shortcuts for frequently used programs. Touch screen PDAs usually have a detachable stylus that can be used on the touch screen. Interaction is then done by tapping the screen to activate buttons or menu choices, and dragging the stylus to, for example, highlight. Text input on a PDA is usually done in one of four ways [12]:

- Using a **virtual keyboard**, where a keyboard is shown on the touch screen. Input is done by tapping letters on the screen.
- Using **external keyboard** or chorded keyboard connected by USB or Bluetooth.
- Using **handwritten letter recognition**, where letters are written on the touch screen, and then "translated" to letters.

- **Stroke recognition** (for example, Graffiti by Palm) is a text entry system with predefined set of strokes that are defined to represent the various characters in English language. The user learns to draw these strokes on the screen or in an input area. The strokes are often simplified character shapes to make them easier to remember.

Among these techniques virtual keyboard and external keyboard do not involve recognition activity, but the other techniques (handwritten letter recognition and stroke recognition) involve recognition process which is time consuming and not completely accurate. However, external keyboard, unlike virtual keyboard, requires connectivity mechanism with the PDA using either Bluetooth or a cable. This indicates that virtual keyboard is by far the best alternative for text entry on PDA environment.

The advantages of virtual keyboards include their simplicity and efficient use of space and when no text entry occur it will disappear, thus freeing the screen space for other purposes. In other words, they are displayed on demand rather than continuously appearing on the screen [13].

1.2 Justification of the Work

This project facilitates the use of Amharic text entry on PDA this in turn pave the way to provide input for Amharic based software application development on handheld devices and as a result Ethiopic software development for handheld devices can be accelerated. The project can also be extended for use on a mobile phone with touch screen capability; there by it will enhance the use of exchanging text massages using local language. Since PDA's can be used in the areas of: schools, hospitals, and in the field of data collection etc. using a localized virtual keyboard will enhance the usage of the device in Ethiopia and its associated benefits.

1.3 Statement of the problem

Virtual keyboard is one of the text entry mechanisms which are widely applicable on handheld devices primarily on a PDA. Having the advantage of this keyboard, several types of virtual keyboard with optimized layout are developed and applied for English text entry. However, no virtual keyboard is available for Amharic text entry. Amharic is the official language of the Federal Government of Ethiopia [15]. It is also a language that is most widely used in Ethiopia.

The effort of Amharic text entry on a PDA has not been made satisfactorily. Except the recent efforts on mobile phones, there is no product on the market. Nevertheless the use of handheld devices such as PDA's is growing and encompassing many other languages. Today's digital technology is being designed by paying due attention to the languages like English, French, Japanese and the like but this has an impact for the speaker of languages like Amharic. So, for users who speak Amharic, the digital technologies having Amharic input method have significant contribution to use the technology more effectively in the local context. On the other hand, due to globalization, newly invented technological devices are able to penetrate into any society with an amazing speed. Since PDAs are among those devices that the digital world contributed to us, it is necessary to design and implement a virtual keyboard for Amharic text entry to PDA devices for making the device more usable for the local people.

1.4 Objectives

1.4.1 General Objective

The general objective of this project is to design and implement a virtual keyboard for Amharic text entry on a PDA platform.

1.4.2 Specific Objective

The specific objectives of the project include:

- Design an optimal ,easy to use, Amharic virtual keyboard layout for PDA
- Select a PDA platform that has a future prospect and that can facilitate the design of Amharic virtual keyboard.
- Select or develop a font for PDA platform to facilitate Amharic text entry
- Implement the designed keyboard layout to demonstrate its effectiveness for Amharic text entry.

1.5 Scope and Limitation

This project focuses on the optimal design of Amharic virtual keyboard layout for PDA environment and involves selection of Amharic font that suits on PDA. In order to have a justifiable layout, standard corpus for Amharic text is needed. But there is no standard corpus yet, due to this previous works by [21] and [4] for Amharic character frequency is considered as bench mark for layout design of the virtual keyboard.

1.6 Methodology and procedures

- Conduct a literature review to explore the available standards and technologies in Virtual keyboard design.
- Select appropriate design for Amharic virtual keyboard layout that satisfies the PDA's constraint and the abundance of Amharic character set.
- Select and use the relevant tools for the developing Amharic virtual keyboard based on the design either on a PDA or PDA emulator.
- Implementing the aforementioned design.

1.7 Organization of the Document

This document contains seven chapters. Chapter one provides the overall introduction about the project. Chapter two describes about the related work on the design of virtual keyboard layout. Chapter three describes about the property of Amharic characters and its associated challenge. In chapters four and five, the analysis and design of the developed system is presented respectively. In the remaining chapters, implementation, and finally conclusion and future works are briefly explained.

Chapter Two

Related Works

2.1 Overview of Virtual Keyboard Design

The criterion for a good virtual keyboard is exactly opposite to the idea behind physical QWERTY keyboard. The idea in physical keyboard is that higher entry rates can be obtained if common digraphs (pair of characters used to write one phoneme) are entered by fingers on opposing hands instead of on the same hand [5]; where as in virtual keyboard layout design, common digraph letters should be close to each other in order to minimize the hand travel with the stylus.

Performance modeling of virtual keyboards focuses on minimizing finger movement on a virtual keyboard, and two factors must be taken into account for this purpose. One is the transitional frequencies from one letter to another in a given language (digraph statistics), and the other is the relative distances between keys. The goal should be to arrange the letters so that the statistical total travel distance is the shortest when tapping on such a keyboard. This means, the most frequent keys should be located in the center of the keyboard and the frequently connected letters, for example in English T and H, should be closer to each other than the less frequently connected letters [6].

In addition, text input requires primary evaluation metrics: speed and accuracy. The simplest way to measure and report speed is to measure the number of characters entered per second during a trial, perhaps averaged over blocks of trials. Accuracy is more problematic. For a simple treatment of accuracy, a metric that captures the number of characters in error during a trial and report these as a percentage of all characters in the presented text can be used [13].

By taking the above virtual keyboard requirement into account, a lot of methods for text entry on a PDA application were developed. The input system design problem for PDA device is to determine the layout that provides the highest text input speed by rendering optimized finger movement for expert users who have memorized the locations of the key. In short, the problem is to find the layout associated with the highest peak expert input rate. However, this criterion is not applicable for novice users as the dominant factor for them is the visual scan time, rather than the movement time [1].

In the next sections, some of the researches that have been made on the layout design for virtual keyboard on stylus operated environment are reviewed.

2.2 ABC Layout

Scott Mackenzie, Shawn X. Zhang and R. William Soukoreff investigated the ABC layout as one virtual keyboard layout that make advantage of grouping the keys as shown in figure 2.1 and by using a space bar that spans the full keyboard[2]. The alphabetic ordering of keys gives novices a good indication of each key's location, and this should reduce the visual scan time. Since the space is the most prevalent character in text-entry tasks, the space bar's size and proximity to the other keys should reduce overall movement time. After entering the key coordinates in their spreadsheet, they made a survey and come up with the prediction rate of >9.6 wpm for novice user and 40.9 wpm for expert users. They indicate `greater than' 9.6 wpm for the novice prediction for the simple reason that the sequenced ordering of keys precludes anyone from being a novice, provided they know the alphabet (a reasonable assumption).

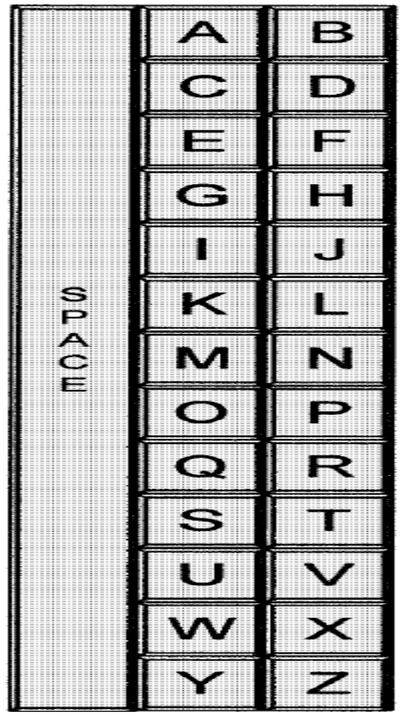

Figure 2-1 The ABC layout [2].

The close proximity of all keys to the space bar is a definite advantage for the ABC layout; however, this appears to be offset by placing letters in two columns. Words like `satisfaction' requires substantial up-down-up pen travel, and this tends to push the efficiency down [2].

2.3. QWERTY Layout

The QWERTY virtual keyboard (figure 2.2) is the other most familiar keyboard layouts for users due to the familiarity with the large full size physical QWERTY keyboard [14]. This layout wouldn't show significant performance, since it was directly adopted from the physical QWERTY layout which is tuned for participation of two hands and placing most frequent digraphs in opposite direction for tapping them in parallel; however this is not the case for stylus operated environment, as frequent digraphs are positioned in such a way that the pen movement for tapping this digraph letters has to be minimized, and virtual keyboards are also limited to serial input because touch screens can only accept a single point of input at a time.

Figure 2-2 The QWERTY virtual keyboard [14].

2.4 Fitaly Layout

It is a replacement keyboard with a layout designed by Textware Solutions Company for a single stylus rather than ten fingers. The most commonly used letters in the English language are clustered in the center of the keyboard, with N and E in the very center along with a spacebar on either side. The layout of the keyword is shown in figure 2.3.

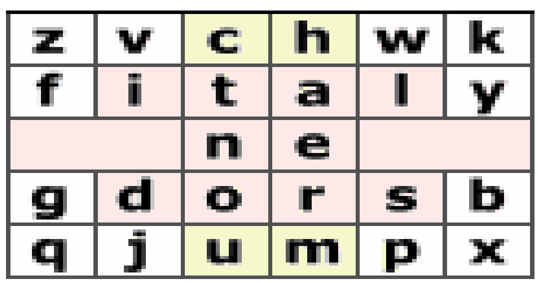

Figure 2-3 Fitaly Keyboard Layout [7].

The main part of fitaly's virtual keyboard design is to arrange letters so that the most commonly used ones are clustered near the center, where they are within three squares' distance of every other letter. According to the company, 73% of the normal English text corpora constitute the letters **i-t-a-l-n-e-d-o-r-s**. Less common letters are positioned outside the center depending on how often they're used and with what letters they're used most often [8]. In addition to this, the layout practically eliminates hand movement when typing text. This means that one can type with the hand rested on the lower area, moving only the fingers holding the pen [7].

The benefits claimed by Textware Solutions Company about Fitaly Keyboard include: minimizing pen/finger movement, eliminating hand movement, the most commonly used letters are placed in the center of the keyboard, letters are arranged to be adjacent or closest to the next letter that one would be entering, and large space keys. They have also made some interesting claims that one will be able to enter 50 words per minute using Fitaly keyboard [9].

2.5 OPTI I and OPTI II Layout

Mackenzie and Zhang [10] designed a new layout, called OPTI I, as shown in the left of figure 2.4. They first placed the 10 most frequent letters in the center of the keyboard. Then assigning the 10 most frequent digraphs to the top 10 keys, and they placed the remaining letters accordingly. The placement was all done by trial and error. They later made a further improved 5X6 layout, as shown in the right of figure 2.4 and they call it the 5X6 layout OPTI II. As can be shown in the right of figure 2.4 there are four space keys in OPTI keyboards that are evenly distributed in the layout. The user is free to choose which one to tap. The optimal choice depends on both the preceding and following key to the space key. For example, for the sequence of M - space – V, the upper right space key is the best choice. However, the upper right space key is not the optimal choice if the tapping sequence is M - space -Y. In practice, the use of the optimal space key ranged from 38% to 47%, depending on the user's experience [10].

In conclusion, the OPTI II performance is between 36 and 40.3 wpm, depending on the optimality of the space key choice. If 38 wpm is taken as a fair, but optimistic estimate, then it will be a 27% improvement over QWERTY performance (about 30 wpm). 38 wpm is beyond what legible handwriting could achieve [3].

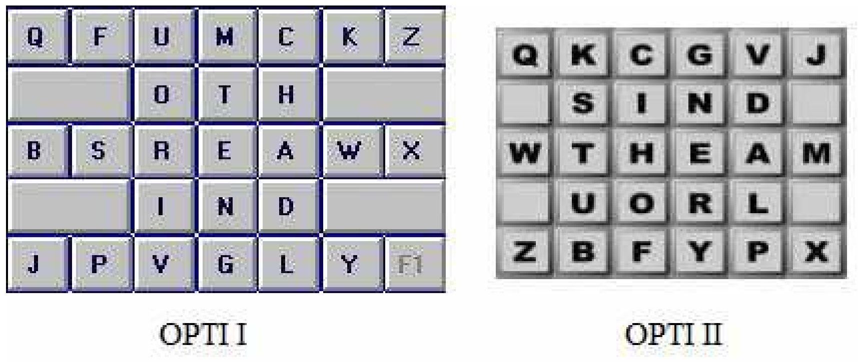

Figure 2-4 The OPTI I and OPTI II Keyboard layouts[14].

2.6 Hooke's Keyboard and Metropolis I & II Keyboards

Zhai, Hunter and Smith [3] applied two physics inspired techniques to generate two optimal virtual keyboard layouts. The first keyboard layout was generated using a mechanical simulation of mesh springs, where springs where stretched between the 26 characters of the alphabet and tensioned proportionally to digraph probabilities in English. This keyboard, figure 2.5, was named Hooke's keyboard after Hooke's law. It yields a predicted expert entry rate of 41.15 words per minute.

The second technique used a Metropolis random walk algorithm guided by a "Fitts energy" objective function producing a keyboard layout called "Metropolis I", in the left of figure 2.6. This keyboard layout generates a 43.1 wpm performance. Further work by Zhai [3] produced the "Metropolis II" keyboard, in the right of figure 2.6, with a predicted word entry rate of 42.94. Metropolis II scored the highest prediction of entry speeds of all the soft keyboards tested.

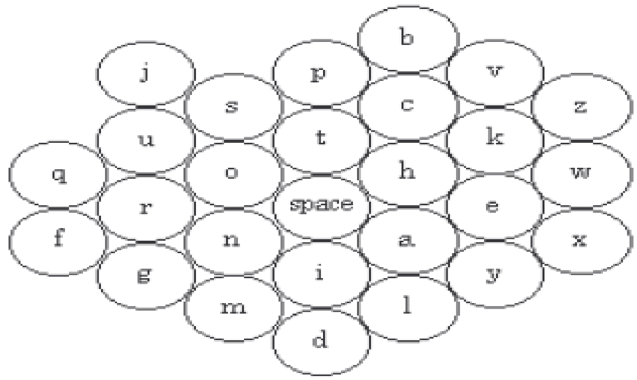

Figure 2-5 **Hooke’s Keyboard [3]**

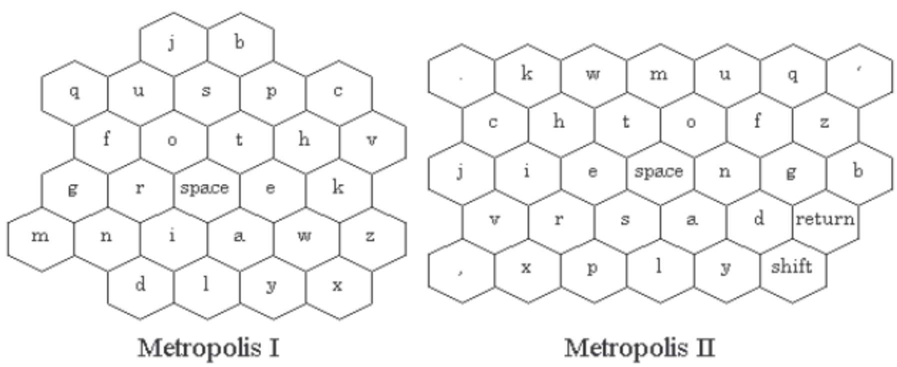

Figure 2-6 The Metropolis I & II Keyboard [14]

Generally all the aforementioned works are performed on English character set, but no endeavor is made to design a virtual keyboard layout for Amharic text entry. However, **EICTDA** had a project on the development of the standardization for Amharic keyboard layout (not virtual keyboard layout) [21].Two popular keyboard layouts have been considered in EICTDA’s project - The Typewriter and Phonetic keyboard layouts. Phonetic keyboard layout is given more priority in this project as it is a much preferred layout by the users and its ease of use.

The draft for keyboard standard pursues the following guidelines to design the Ethiopic keyboard layout [21].

- Applying minimum number of keystrokes per character for speed purpose
- Right and left hand load balance
- Applying Limited deviation from existing keyboard layout
- Avoiding system keys
- Using a single switch key

Then, the most frequently used characters to map into the keyboard based on Character frequency analysis and load balance is determined. For this purpose two public documents are collected - The Ethiopian Constitution and Turnover Tax Proclamation No 380/1995 as text corpora and perform the character frequency analysis. The final result of the analysis show the relative occurrence of every Ethiopic character family (see Appendix A) and the order of Ethiopic characters (see Appendix B).Based on the result of the analysis "የ " families and Sixth order (ሳ ድ ስ) are most frequent and "ኸ " families and eighth order "ኧ ሬ ቤ" are the least frequent.

Worku Alemu [4] has also analyzed the frequency of occurrences of Amharic characters in his thesis work. He collected five thousand words from 'Addis Zemen' newspaper using Microsoft word (Version 7), WashRa (Amharic computer font) then using SAS program he came up with the table of result, (see Appendix C), showing the percentage of occurrence for each Ethiopic character. As can be seen from the table (Appendix C), the most frequently occurring character is ን, which occurs 946 times out of 20,529, accounting for about 4.6695% of the total occurrence of characters and the least frequently occurring characters are ቺ ,ጇ ,ኘ ,ኼ ,ኚ ,ጂ ,ዄ ,ጮ ,ዬ , ዖ ,ፐ ,ጧ ,ፑ each of which occur one times and accounting for about 0.0049% of the total occurrence. He has also tabulated the frequency of occurrence for orders of characters from the collected text corpus (see Appendix B).

Even if no attempt is made to design a virtual keyboard layout for Amharic characters, the project of EICTDA [21] and worku's thesis [4] on analysis character frequency can be used as a bench mark for the Amharic virtual keyboard layout design. Since, the main criteria for designing optimal virtual keyboard layout include the character frequency analysis of a language as a major part.

Chapter Three

Challenges of Virtual Keyboard Design for Amharic Character Set

In designing of the virtual keyboard, the language for which the virtual keyboard is designed has to be studied. Especially when it comes to language like Amharic, that has many characters, the need for analyzing the language will be prominent. In this chapter, the overview of Amharic language and its associated challenges for the design of a virtual keyboard is explained.

Amharic is the working language of the federal government of Ethiopia and is spoken and written as a first or second language in many parts of the country. It is the language that uses characters derived mainly from Ge'ez which was the language of literature in Ethiopia until the middle of the 19^{th} century [4].

As Amharic has borrowed most of its characters from Ge'ez, it uses characters created by consonant vowel fusion (the second version of Ge'ez script representation) and seven vowels are used in Amharic each of which comes in seven different forms (orders) reflecting the seven vowel sounds (ኧ , ኡ , ኢ , ኧ , ኤ , ኧ , አ) [16]. Because of this, Amharic script is categorized under a syllabic type. For example the symbolic representation of the seven forms of the Amharic characters በ (be), ገ (ge), ደ (de) are shown in the Appendix F.

The Amharic writing system consists of a core of 33 characters where each of which occurs in a basic form and six other forms, making a letter to have seven forms. This representation of each core characters in seven different forms raises the number of core characters to 231(33**x**7) as shown in Appendix D. There is also character ቨ , which has also seven forms, used to represent a ‘v’ sound of words from other Latin-based languages. In addition, there are 44 other symbols which contain a special feature usually representing labialization (ሏ ሟ ሯ …). There are also punctuation marks which consist of a basic word-driver (:), a sentence-driver (::), and other marks like equivalent to the English comma (፣), semi-colon (፤), and borrowed symbols like ?, !, ", (,).The numeration system consists of a basic single character for 1 to 10, for multiples of 10(20 to 90),for 100 and 1000.The numeral symbols are shown in Appendix E.

Generally, it can be seen that the total number of symbols used in Amharic script is about 310; it is categorized in the following manner.

Core Symbols	(33 x 7)	=231
Special symbol (ቨ)	(1 x 7)	=7
Labialized symbols (ሏ ሟ ሯ …)		=44
Punctuation marks		=8
Numerals		=20

Furthermore, the Amharic writing system is not suitable for arithmetic computations and numeric representations [4]. In other words, it doesn’t have an efficient way of representing numerical values and operators as well. Consequently, for the representation of numbers the Arabic numerals are commonly used and for operators all symbols in the Latin based scripts are used. The use of these additional symbols raises the number of symbols in the Amharic writing system to more than 330.

Taking the above facts into account, designing Amharic virtual keyboard in PDA environment is more challenging. As it is known, PDA is one of the handheld devices manufactured to operate on limited resources such as small CPU cycle, limited memory, and small screen etc. In addition, the abundance of Amharic characters makes the design of virtual keyboard on the small screen more difficult.

So in designing a virtual keyboard for Amharic language, these challenges should be considered.

Chapter Four

System Analysis

4.1 Overview

In chapter two, the different design of virtual keyboard layout for English language has been considered. Since the numbers of characters for English language are relatively small, all of the layout designs for virtual keyboard are concerned with the arrangement of the characters in such a way that their usability will be more comfortable to the user. In the next subsections the functional and non-functional requirements of the system will be described and modeled using UML modeling notation.

So far there is no Amharic virtual keyboard layout for PDA environment. This will be the first ever endeavor to put corner stone for the design of virtual keyboard layout that can create one alternative way for input method of Amharic text on PDA. Hence there is no any existing system related to this project to consider it as a reference.

4.2 The Envisaged System

4.2.1 Overview of the system

The Amharic virtual keyboard system is the software that is expected to render the Amharic input method on PDA environment. This system should allow the user to provide input as a form of Amharic character on PDA systems. It should also permit English character set entry if the user wanted to insert English texts. The layout of the Amharic virtual keyboard should be designed in such a way that text entries involve the minimum possible hand movements, but for English text entry QWERTY layout is adopted to this system.

The user of the PDA can use a stylus to write Amharic characters, using the soft keys (buttons) displayed on the screen. Figure 4.1 shows the high level work flow representation of the system.

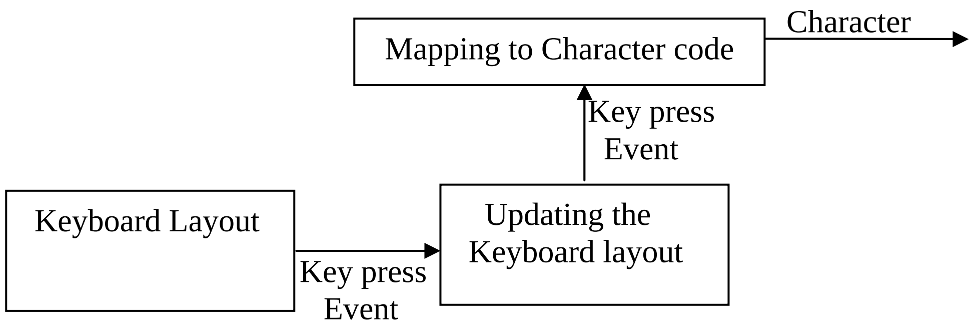

Figure 4-1 Work flow of the Amharic virtual keyboard system

As can be shown in Fig 4.1 ,the system will display the Amharic keyboard layout on the screen showing the basic character selection set which consist of keys(buttons) for the first order letters "ሀ " through "ፐ" and other representatives for Amharic numerals, Arabic numerals and Amharic punctuation marks. When the user presses one of the basic characters set, the keyboard layout will be adjusted so that all the families of the pressed character will be shown in the editable (top or bottom) panel. Now, the user can press the key representing the required character in the editable panel; then this character should be displayed which in turn be consumed by any other application as a user input.

4.2.2 Functional Requirements

The developed system is expected to provide the following functionalities:

- The system should allow the user to tap the Amharic characters shown in the layout and as a result the corresponding character code should be loaded into memory.
- The system should allow the user to delete the character when the need arise
- The system should allow entering English characters if the user needs.

4.2.3 Non-Functional Requirements

There are also non-functional requirements expected from the system and the following lists these requirements.

- The layout appearance should not be complicated in order to remember the keys pattern on it.
- The system must not delay in showing the associated soft keys for non basic characters of the selected basic character.
- The system must be easy to learn and use

4.3 Analysis Model

In this section the system is described by showing its functionality using a use case as shown in figure 4.2, moreover the static and dynamic behaviors are shown using class and sequence diagrams respectively.

4.3.1 Use case Diagram

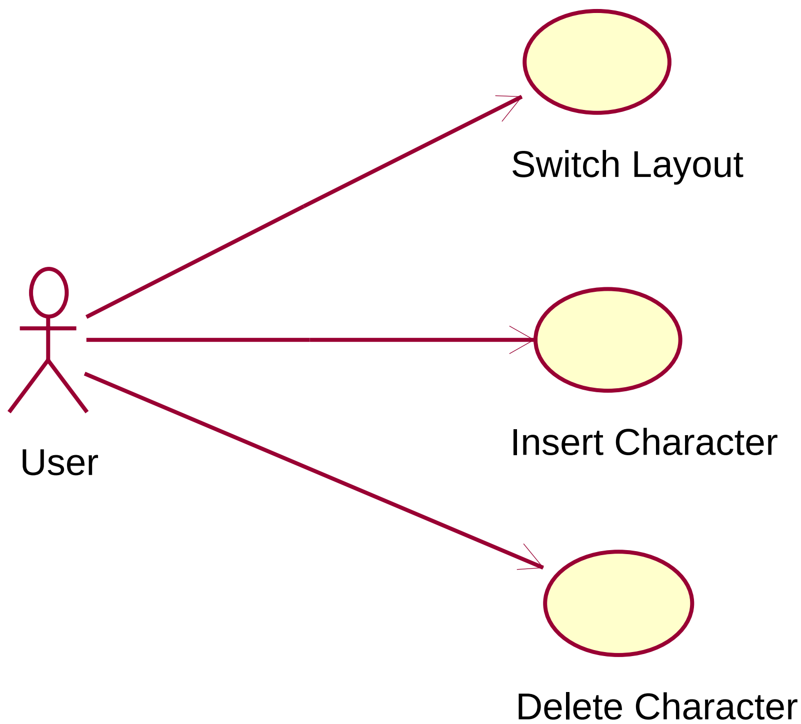

Figure 4-2 **Use case diagram of the system**

4.3.1.1 Use case description

The following three tables (Table 4.1, Table 4.2, and Table 4.3), show the summary of use case description for Switch Layout, Insert Character and Delete Character.

Use Case Name	Switch Layout
Participating Actor:	User
Description:	A use case that allows a user to change the layout between Amharic and QWERTY (English character set).
Pre –condition:	The user initiates the system
Flow of Events:	1. If the system is currently displaying the Amharic layout then pressing the button "**ወደ እንግሊዘኛ**", will change the layout to QWERTY layout. 2. If the system is currently displaying the QWERTY layout then pressing the button "**To Amharic**", will change the layout to Amharic layout.
Post condition:	The layout is changed either "from Amharic to QWERTY" or "from QWERTY to Amharic".

Table 4-1 Use case description for Switch Layout

Use case Name	Insert Character
Participating Actor:	user
Description:	A use case that allows a user to insert a character
Pre-condition:	The layout containing the base characters must be loaded
Flow of Events:	1. The user selects the base family of character, from which his target character is a member, by clicking on it 2. The system displays all the NonBasic families of the pressed character key 3. The user will press the required character key to be inserted 4. The system appends the mapped character to input control box
Post-condition	The inserted character will be displayed in the input control box.

Table 4-2 Use case description for Insert Character

Use case Name	Delete character
Participating Actor:	user
Description:	A use case that allows a user to delete a character from input control box.
Pre-condition:	There must be at least one character in the input control box.
Flow of Events:	1. The user presses the delete char button from base character layout panel. 2. The system removes the last character from the content of the input control box. 3. The input control box will be assigned the updated content.
Post-condition	The input control box will show its content with removed character

Table 4-3 Use case description for Delete Character

4.3.2 Sequence Diagram

The dynamic behavior of the system is depicted by using sequence diagrams. The sequence diagrams facilitate the specification of formalization to accommodate behavioral modeling which is responsible to show and capture the interaction between participating objects in a given use case. They are also helpful to identify the missing objects that are not identified in the analysis object model. The following diagrams describe the direct translation of identified use cases in to the sequence diagram.

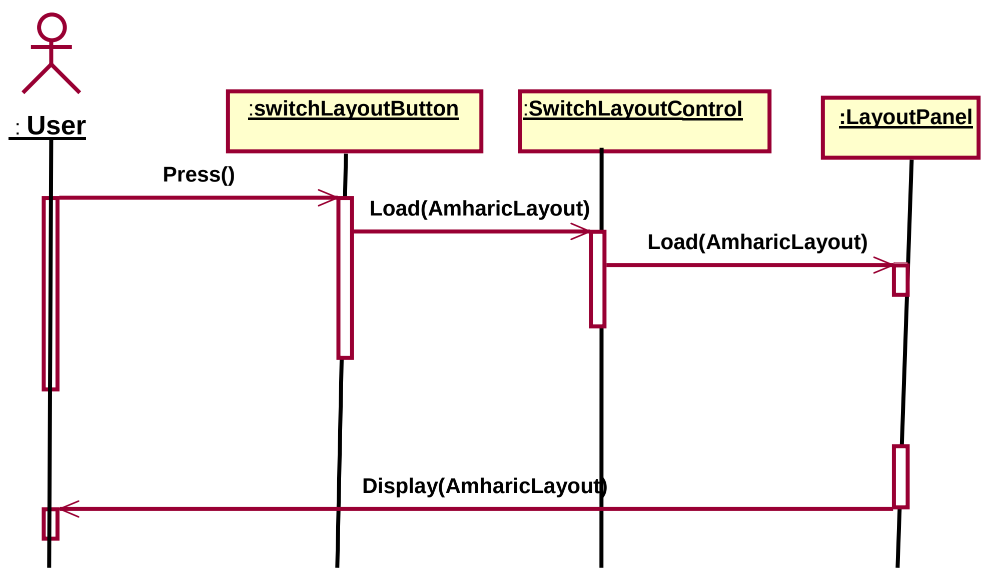

Figure 4.3 Sequence diagram for Switch Layout

As shown in figure 4.3, shows the instance of changing the English layout to Amharic layout by interacting with the boundary object SwitchLayout button, then "SwitchLayoutController" will inform the LayoutPanel object to Load the Amharic layout. Finally, the Amharic layout will be displayed in the LayoutPanel.

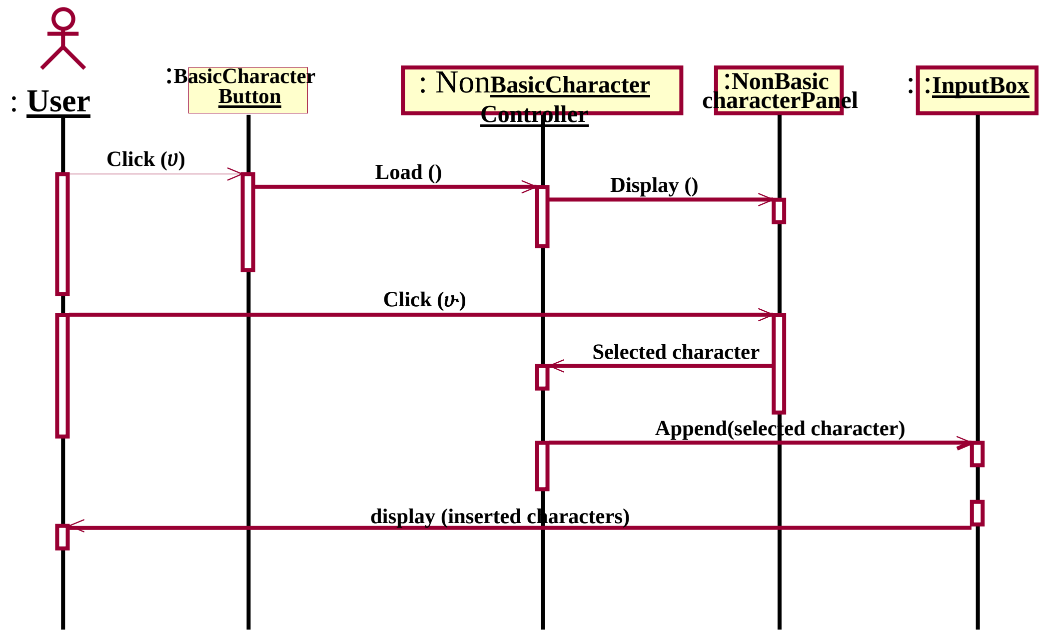

Figure 4-4 Sequence diagram for Insert character use case

As shown in figure 4.4, once the user has activated the basic character panel by interacting with the boundary object "ሀ " button, then "NonBasicCharacterController" displays non-basic character selection panel showing all non basic characters of "ሀ ". After this the control object will wait until the user has to interact with the boundary object "ሁ". When the user presses "ሁ" button character "ሁ" will be mapped as input character and will be appended in input box. Finally, the inserted character, "ሁ", will be displayed as the last character in the input box content.

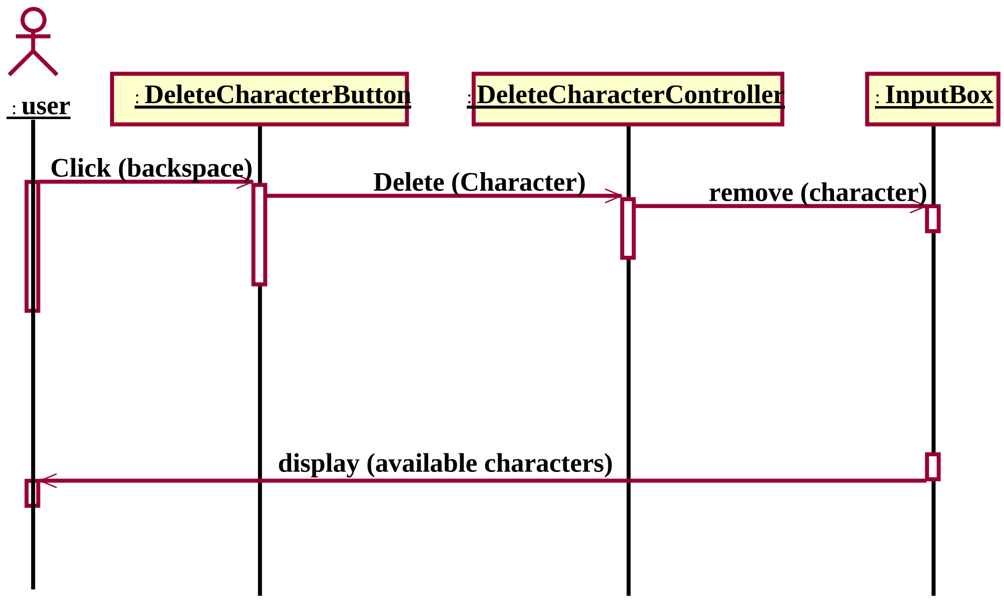

Figure 4-5 Sequence diagram for Delete Character use case

In the figure 4.5, shown above, once the user has activated the basic character panel by interacting with the boundary object "←" button (back space button), then "DeleteCharacterController" will invoke remove method to delete the last character from the content of the input box. Finally, the updated content of the input box will be displayed.

4.3.3 Activity Diagram

The description of the system in terms of activities is shown in figure 4.6 using activity diagram of the system.

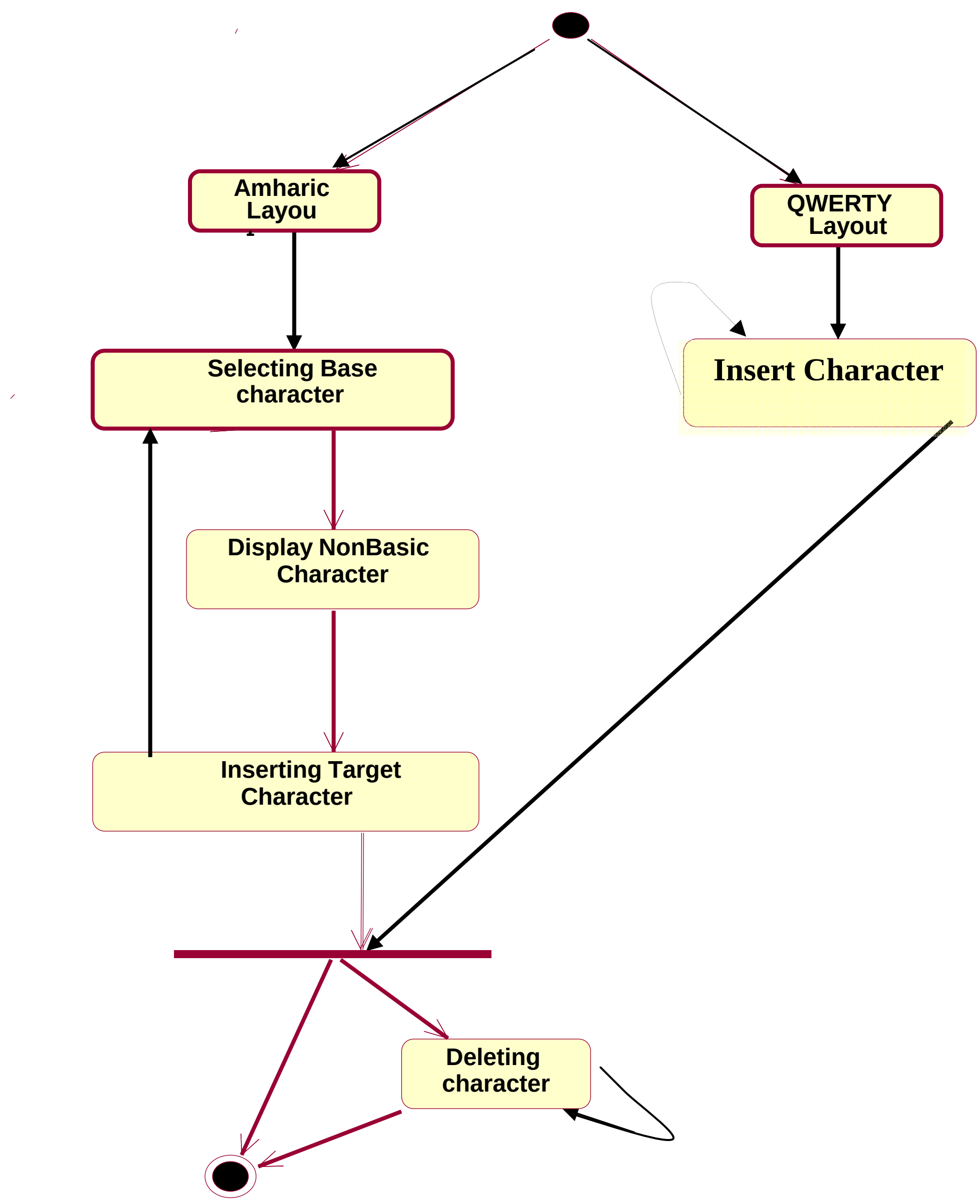

Figure 4-6 Activity diagram of the system

4.3.4 Class Diagram

The Class diagram is used to identify domain objects or conceptual classes, associations between conceptual classes, attributes and operations of conceptual classes. Figure 4.7 depicts the class diagram for the set of classes that are identified in our system.

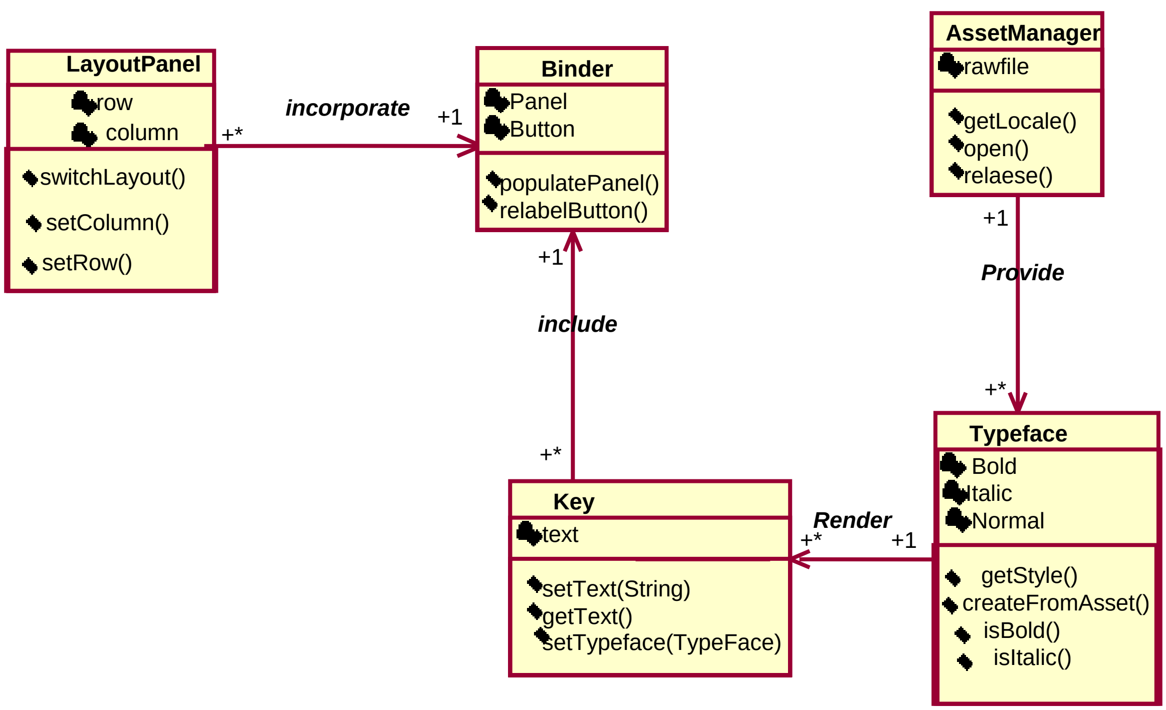

Figure 4-7 Class diagram of the system.

Chapter Five

System Design

In chapter 4, the functional and nonfunctional requirements of the system were considered and analysis model was constructed. This chapter presents the design of the system by setting the design goals, the architecture of the system followed by subsystem decomposition.

5.1 Design Goals

The design goals represent the desired qualities that the system should have and provide a consistent set of criteria that should be taken into account when making design decisions. It describes the qualities of the system that developers should optimize.

When one needs to design software for handheld devices, like PDAs, the inadequacy of processing speed, memory size, and size of the screen should thoroughly be considered. In order to achieve the design goal one should balance the conflicting requirements like time space trade-off. It is possible to satisfy the response time or throughput requirement by consuming more space; however space is also a constraint for handheld devices. Therefore it is necessary to consider the following design issue in detail while the system is being designed.

- Design issue 1: Limited Computational Capacity
- Design issue 2: Limited Screen Size
- Design issue 3: Limited Memory Size
- Design issue 4:Limited battery life

5.1.1 Performance Criteria

Performance may include the speed and memory requirements of the system. Processing speed and available memory are the main constraints of handheld devices. Hence, the following performance issues should be considered in designing the system having in mind the constraints of such devices.

Response time: the system should map the machine character code to the pressed key in the layout without noticeable delay.

Memory Requirement: the system should be designed so that it can fit the memory restrictions of PDAs.

Throughput: The system must arrange the layout based on the pressed key in the layout and notify the user for the input entry before the next key is tapped by the user.

5.1.2 Maintenance criteria

The maintenance criteria deal with the difficulty of changing the system, adding new functionality, revising existing functionality, portability of the system.

Extensibility: the system should enable the addition of new layout for other languages like Tigrigna , Guragigna and other local languages, without affecting its current functionality.

Modifiability: the system should enable the change in layout configuration with minimum effort if the need arise.

5.1.3 Dependability criteria

The dependability criteria deal with the effort to minimize system crashes and their consequences in connection with security risk and safety issues; it also deal with the availability of the system to the user.

Reliability: The system shall be tested in order to manage the difference between specified and observed system behavior; in other words, it should give consistent and correct character with specified font to the pressed key in the layout.

5.1.4 End User Criteria

The end user criteria includes qualities that are desirable from a users' point of view and have not yet been covered under the performance and dependability criteria

Usability: The system shall be developed so as to be easy for user understanding. By providing a layout that requires minimum learning curve for novice users, it is possible to enhance usability of the system.

5.2 Architecture of the System

The system is well suited to be implemented by a **Model View Controller** design pattern.

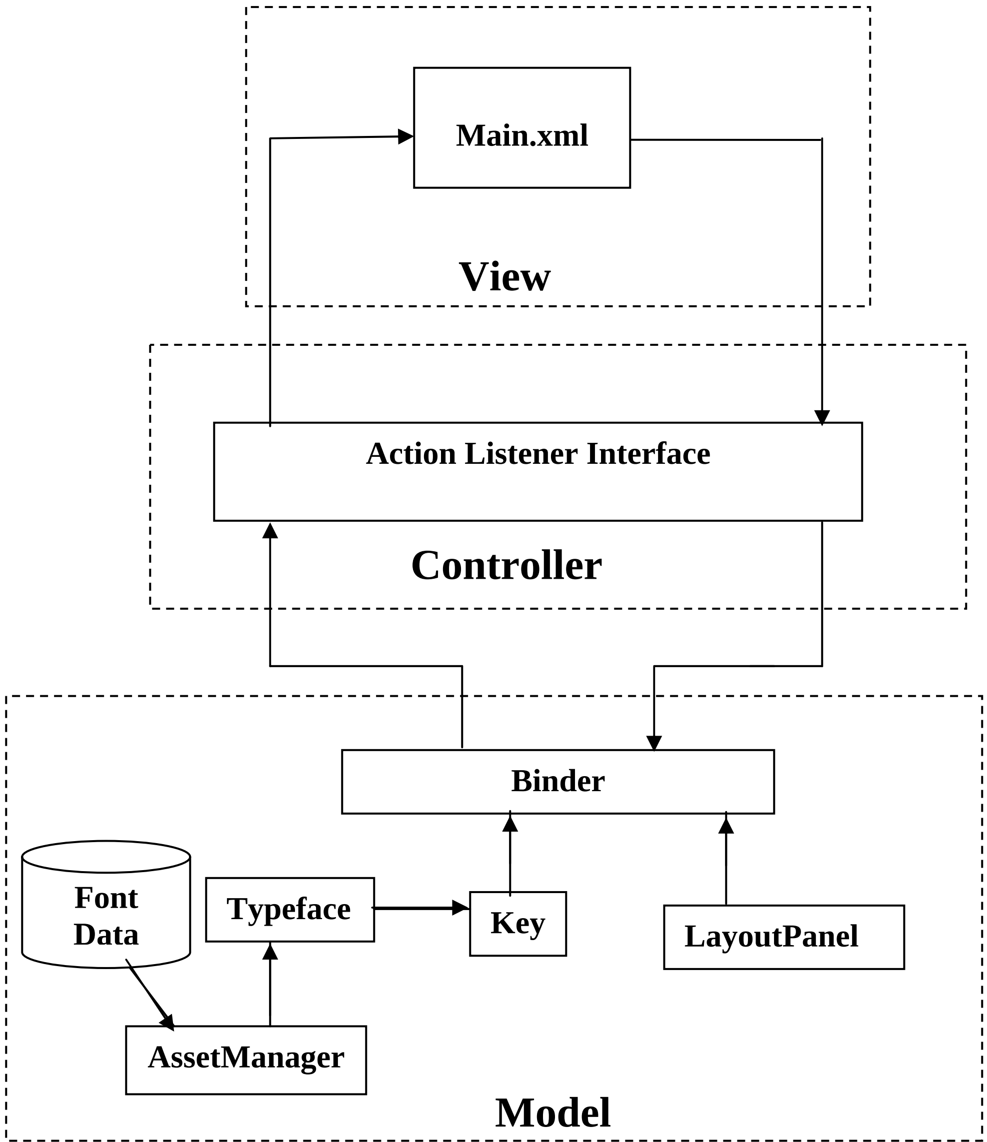

Figure 5-1 The general architecture of the Amharic virtual keyboard system

From the architecture of the system one can observe that the system has three layers: the view, the controller and the model.

The view is responsible for the visualization of the state of the model, and for the mapping of graphics onto a device. A view typically has a one to one correspondence with a display surface and knows how to render to it. It also attaches to a model and renders its contents to the display surface. In other words, the view is capable of rendering the contents of the model to the display surface by managing the graphical and textual output portion of the bitmapped display allocated to the application. For our system there is an xml specification about the view of the user interface that is written in the main.xml.

The controller is the means by which the user interacts with the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. In effect, the controller is responsible for mapping end-user action to application response by interpreting the mouse and key press input from the user, and commanding the model and/or view to change as appropriate. The user interaction in this system is button click that contains the desired character font. Based on the user interactions and the outcome of the model actions, the controller responds by selecting an appropriate view.

The model manages the behavior and the data of the application domain, it responds to requests for information about its state and instructions to change its state. It is the layer that contains all the business rules and algorithm of the application to manage the state of the application and conduct all transformations. For this system the five classes Binder, Layout panel, Key, Type face and AssetManager classes are part of the model for the Amharic Virtual keyboard system. The font data is stored in the asset folder of the application.

Generally, the Model-View-Controller design pattern maintains links between model and views and notifies the views when the model changes state. The view is the piece that manages the visual display of the state represented by the model. A model can have more than one view. So the model part of the system will be considered for subsystem decomposition.

5.2.1 Subsystem Decomposition

The objective of subsystem decomposition is to reduce the complexity of the system in such a way that the subsystems are loosely coupled to each other and strongly cohesive internally. In our project the system is decomposed into two subsystems. These are layout subsystem and map subsystem as shown in the figure 5.2.

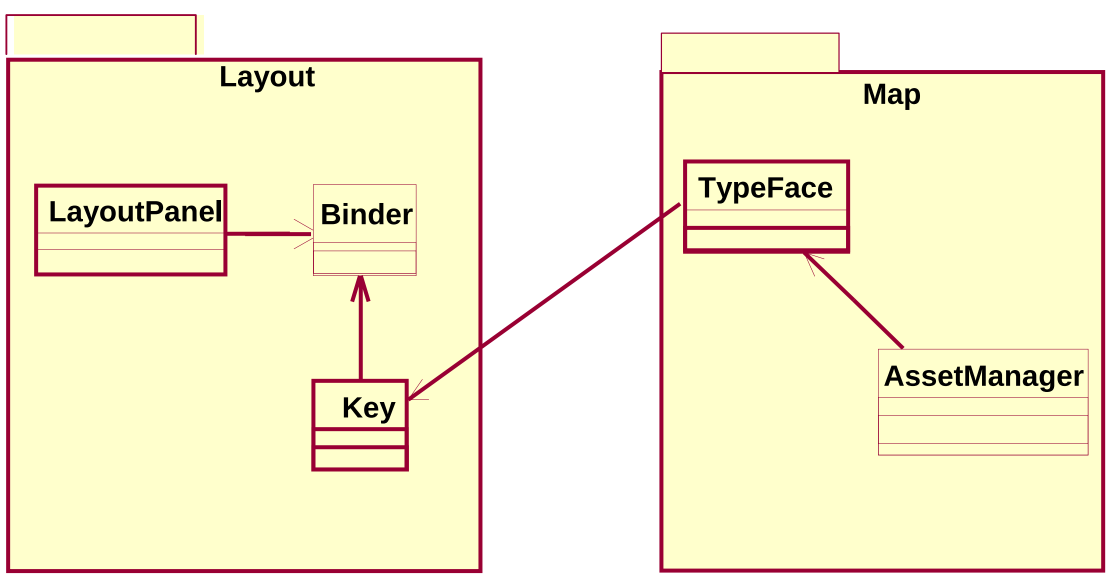

Figure 5-2 Subsystem Decomposition

5.2.1.1 Layout subsystem

The layout subsystem provides the appropriate interface for the input method of the system. The display configuration of the virtual keyboard is determined by binder class using array of keys, and the layout panel manages the keyboard appearance.This subsystem packages the bind, key (button) and layoutpanel classes together.

This subsystem is the major component of the project that incorporates the proposed layout design for Amharic virtual keyboard and the description of the layout design is stated as follows.

The proposed layout for Amharic virtual keyboard is composed of three panels.

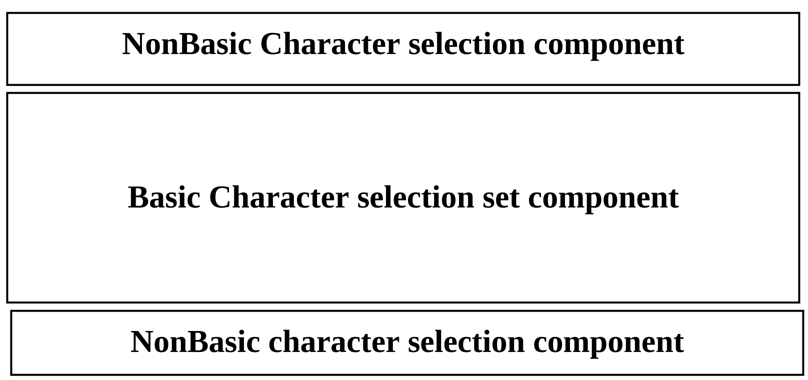

Figure 5-3 Proposed Layout for Amharic virtual keyboard

This layout is designed by considering the abundance of Amharic character set and the limitation of screen size of the PDA. In PDAs screen, the rule of thumb for the width and height of the layout is 3:1 aspect ratio [14]. Since it is unfeasible to have one to one mapping for each key in the layout and each Amharic character code, we come up with the idea of partitioning the keyboard panel into three parts as shown in the figure 5.3. The middle panel is the main panel, the top and bottom panel are dependent on this panel. The middle panel, Basic Character selection set panel, is designed to accommodate the first order character of the 34 Amharic character family ,Arabic numeral selection key, Arithmetic operators selection key, Amharic digit selection key, Amharic numeral (more than one digit) selection key ,delete character key. In addition to this, it is enabled to load QWERTY configuration when English layout is needed. The upper panel, NonBasic character selection component, is intended to display the entire order including the eighth order “ኧ ሯ ቧ” of the clicked basic (first order) character in the middle panel. The bottom panel, NonBasic character selection component, is intended to display complete Amharic, Arabic numerals and Arithmetic operators in addition to NonBasic characters. The bottom panel is introduced in order to reduce the pen travel distance for the NonBasic characters of the basic characters at the third and fourth row of the middle panel.

The major criterion for designing layout of the virtual keyboard is to develop a layout that will facilitate pen (stylus) travel minimization of the user while entering input. This in turn entails us to consider the character frequency of the language. In order to analyze the frequency of Amharic character, Amharic text corpus is needed. However, no standard Amharic corpus is available as specified in the scope of the project. So the work on the draft of standard keyboard layout [21] and Worku’s thesis [4] on character frequency analysis will be considered as baseline for the proposed layout. The table in (Appendix A) shows the character frequency table for Amharic character family and is used for the purpose of arranging the first order characters in the middle panel of the layout according to their frequency weight.

The middle panel has four rows and ten columns as shown in the figure 5.4. This panel contains the first order of the 34 Amharic characters, Arabic numeral selection key, Arithmetic operator's selection key, Amharic digit selection key, Amharic numeral (more than one digit) selection key and delete character key. The placement of the first order characters are made by using their corresponding frequency count based on the works in [21] .This frequency table is shown in Appendix A.

የ	መ	ነ	ተ	ለ	በ	ረ	ወ	ሰ	ገ
ሠ	ኘ	ጀ	ዐ	ጸ	ፀ	ሸ	ጨ	ፐ	ኀ
ኸ	ጰ	ዠ	ቨ	#	+	፩	፳	፡	<-
ከ	ደ	አ	ቸ	ቀ	ጠ	ፈ	ሀ	ዘ	ሐ

Figure 5.4 The configuration of the first order letters in the middle panel of the layout.

The frequency of letter "ኸ" is not specified in the project of EICTDA (see Appendix A).Approximation is made for the frequency of the letter "ኸ" by taking the worku's thesis[4] into account and it is placed below the letter "ሠ".

For the first two rows of characters in the middle panel, the pen travelling distance is measured from the character in the middle panel to the target character in the top panel. For the last two rows of characters in the middle panel, the pen travelling distance is measured from the character in the middle panel to the target character in the bottom panel. Obviously, the distance will be shorter for characters placed on the top row and bottom row of the middle panel of the layout and it is longer for characters placed in the second and third row of the middle panel of the layout.

As can be seen from figure 5.4, the arrangement of the characters is made in a manner that the character with highest frequency to take the top and bottom rows of the middle panel. This is made deliberately; by making the most frequent families at the top and bottom row, it is possible to minimize the pen travelling distance as NonBasic characters are going to be shown in the top and bottom panels of the layout.

The top panel has one row and eight columns. This panel is dedicated to display the target characters that are going to be used as input. The appearance of this panel is dependent on the chosen characters in the middle panel. In other words, the role of characters in the middle panel is to dictate which family of the character to be displayed in the top panel of the layout. Here, once again the frequency of character's order is considered. That means the non basic characters will be displayed with respect to the base character in the middle panel such that the most frequent character is closest to the base character as compared to others, and the same will be applied to the remaining NonBasic characters. This will make the most frequent characters to be typed with minimum pen travel distance. The order of frequency for Amharic characters is also analyzed by [21] and [4], and they come up with the same result as shown in Appendix B. As an example one can consider the situation that will happen if the button containing "የ " is pressed. The layout in the top panel will appear as shown in the table 5-1 in response to the pressed key "የ ".Similarly the appearance of the top panel when "ገ " is pressed, in the middle panel of the layout, is shown in table 5-2.

ይ	የ	ያ	ዪ	ዩ	ዬ	ዮ	ዯ

Table 5-1 Appearance of top panel of the layout, when "የ " is pressed in the middle panel.

ጓ	ጔ	ጕ	ጎ	ጒ	ጐ	ገ	ጘ

Table 5-2 Appearance for the top panel of the layout, when “ገ ” is pressed in the middle panel.

፨	፣	:-	‘	¨	?	፧	፤	፡	።

Table 5-3 Appearance of top panel of the layout, when pressing “:” in the middle panel.

The bottom panel is responsible for displaying Arabic digits (0 through 9), Amharic digits (፩ through ፲), Amharic numerals (፳ through ፻), and NonBasic characters of the base character in the middle panel. The advantage of incorporating the bottom panel in the layout is three fold. First is to change the direction of hand movement for half portion of the characters in middle panel and this makes reduction of statistical (or Average) pen movement distance by half. Second is to reduce visual search of Characters. Third is to make the position memorable (learning process of keyboard layout easier). The appearance of the bottom panel when “:” is pressed, in the middle panel of the layout, is shown in table 5-3. The appearance of the bottom panel when “#” is pressed, in the middle panel of the layout, is shown in table 5-4. The appearance of the bottom panel when “፩” is pressed, in the middle panel of the layout, is shown in table 5-5.

0	1	2	3	4	5	6	7	8	9

Table 5-4 Appearance of bottom panel of the layout, when “#” is pressed in the middle panel.

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲

Table 5-5 Appearance of bottom panel of the layout, when “፩” is pressed in the middle panel.

5.2.1.2 Map subsystem

The map subsystem is responsible for mapping the user action into the appropriate Amharic character font, and sending the character as output. This character can be seen as output of the system. This subsystem packages typeface and AssetManager classes together.

Chapter Six

Implementation

An implementation of the designed Amharic virtual keyboard layout is developed for Android platform PDA Emulator. This chapter will present the developed system.

6.1 The system development environment

In this section, the development environment of the implemented system and the tools used are presented

6.1.1 The Android OS

The Android operating system is a complete software stack for mobile devices such as cell phones, PDAs and high end MP3 players, etc. It is based on open source software, and managed by Google and OHA project [18, 19]. It is optimized for mobile devices with limited memory and horsepower. Its anatomy [20] is split into four layers as shown in figure 6.1:

- The applications layer
- The application framework layer
- The libraries and runtime layer
- The kernel layer

Figure 6-1 Android Architecture [22]

The Android SDK (Software Development Kit) provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. For this project, the implementation is developed using this platform (Android SDK) with Eclipse IDE together with ADT (Android Development Tool) plug-in.

The Applications Layer

This layer contains a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language. There are four building blocks to an Android application: Activity, Intent-receiver, service and content provider.

Not every application needs to have all four, but will be written with some combination of these. Once the needed components are decided for the application, then it should be listed in a file called AndroidManifest.xml. This is an XML file that specifies where the declaration of components for the application together with their capabilities and requirements.

The Application Framework Layer

This is a section for accessing the framework APIs and core applications. Generally, all applications are composed of set of services and systems, including: views for graphical stuff, content provider for data sharing, resource manager for accessing non-code resource (layout files), notification and activity manager to provide information and to control life cycle of the application.

The Libraries and Run time Layer

In this layer a set of C/C++ libraries that are used by various components of the Android system are included. The set of core libraries that provide most of the functionality available in the core libraries of the Java programming language is also adopted. These capabilities are exposed to developers through the Android application framework. Some of the core libraries include: system C and media libraries, Graphics libraries, **free bitmap and vector font rendering**, and SQLite for relational database engine. This project uses the free bitmap vector font rendering engine to support the Amharic font (nyala).

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.

Linux Kernel Layer

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model[22]. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

6.1.2 Why Android

The basic reasons to choose Android platform, for this project, are the following:

- It uses Freetype, a free and open source bitmap and vector font engine, which supports Unicode text layout and glyph rendering. Ethiopic Unicode fonts, such as Microsoft's Nyala and GFZ's Geez Free Zemen, can be used to render text using Freetype on the Android platform. Other platforms such as the Palm OS lack the internationalization support required to fully meet the needs of Ethiopic software development.
- It is based on open source product, Linux kernel, which brings a big change from the current oligopoly of mobile OS makers that charge handset manufacturers various licensing fees.
- It comes with immediate availability to mobile versions of Google search, Google Maps, and Google Products that allow independent developers to create their own array of applications at a much less restrictive cost

As a show case the system is implemented on Eclipse IDE and tested with Android Emulator. This emulator is described in section 6.1.3.

6.1.3 Development Tools

The Android SDK includes custom tools that help to develop mobile applications on the Android platform. The most important of these are the *Android Emulator* and the *Android Development Tools plug-in for Eclipse*.

The Android Emulator

The Android SDK includes a mobile device emulator — a virtual device that runs on the computer as shown in figure 6.2. The emulator lets us prototype, develop, and test Android applications without using a physical device. The Android emulator imitates all of the typical functions and behaviors of a mobile device, except that it can not receive or place phone calls [23].

Figure 6-2Android Emulator [23]

Android Development Tools Plug-in for the Eclipse IDE

The ADT plug-in adds powerful extensions to the Eclipse integrated environment, making creating and debugging Android applications easier and faster. If **Eclipse** is used as a development environment, the ADT plug-in gives an enhancement for developing Android applications. It gives access to other Android development tools from inside the Eclipse IDE. For example, ADT lets access for taking screenshots, managing port-forwarding, setting breakpoints, and viewing thread and process information directly from Eclipse. It also provides a New Project Wizard, which helps to quickly create and set up all of the basic files needed for a new Android application. In addition it automates and simplifies the process of building Android application by providing code editor that help to write valid XML for Android manifest and resource files.

In general, the ADT plug-in handles the all packaging and installation of the application for using the Android Emulator on Eclipse IDE. Figure 6.3 shows the screen shot taken while implementation is done for this system.

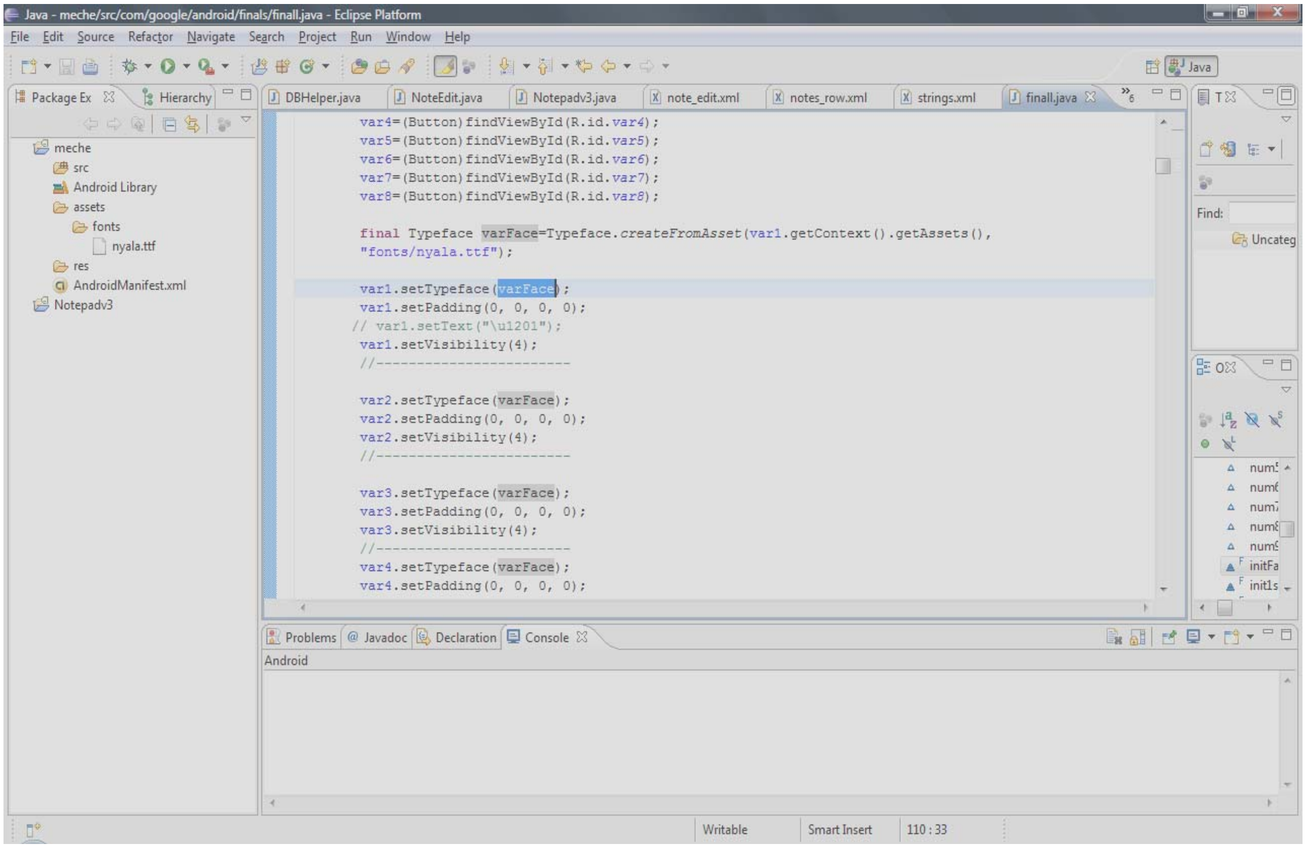

Figure 6-3 Eclipse IDE with ADT plug-in embedded in it.

6.2 The Amharic Virtual Keyboard System (AVKB)

Based on the proposed layout, designed in the previous chapter, implementation is done and the implemented system is named as “Amharic Virtual Keyboard System”. AVKB is implemented in such a way that Amharic characters, punctuations, Arabic, Arithmetic operators and Ethiopic numerals to be incorporated in the layout systematically. In addition to this, the system is capable of permitting English text entry. Initially, the system will display all the first order of the 34 Amharic characters, Arabic numeral selection key, and Arithmetic operators’ selection key, Amharic digit selection key, Amharic numeral (more than one digit) selection key, punctuation selection key and delete character key. The user has to tap the key of his interest from which the character family belongs. In other words, if the user wants to input the character “ሂ”, then he has to press a key containing the first order form of the character he wants to provide; In this case the user has to press the base character “ሀ”, after that the top part of the keyboard will show all the NonBasic families of the base character. Then the user has to press the key labeled with “ሂ”.Similarly this procedure will be applicable to all characters, digits, operators, punctuation and numbers.

After deploying the system onto the Android Emulator, AVKB will be available on the desktop of the PDA as shown in Figure 6.4.

Figure 6-4 Image of Android Emulator device with AVKB System installed.

Once the **AVKB** system is installed, you can run it by clicking the icon in the desktop of Android PDA then the following interface containing the basic character selection sets and others as specified in the layout design will appear as shown in figure 6.5.

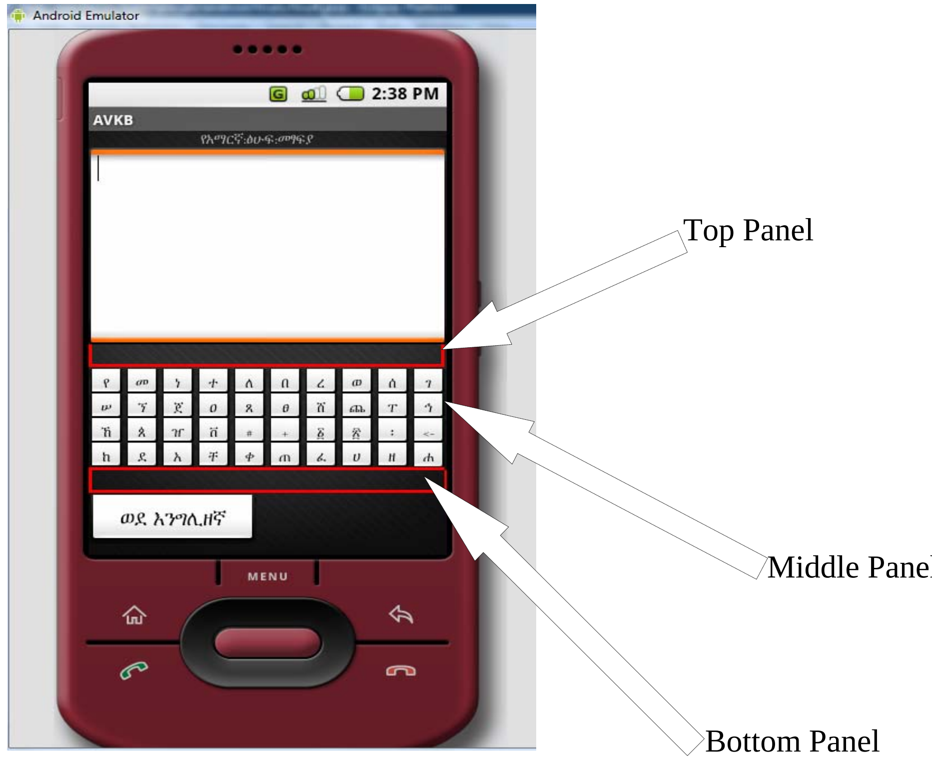

Figure 6-5 The initial layout after running AVKB.

The system consists of the proposed keyboard layout, and textbox for the purpose of monitoring the characters inserted by the user. When a user wants to insert a character, he/she has to select (press) the base character of the target character in the layout. Then the NonBasic characters for the selected character will be shown in either top or bottom panel of the layout. The decision to select the panels (top or bottom) is made based on the proximity of the pressed base character in the middle panel of the layout. If the user wants to insert a character “ይ”, he has to press “የ”, and then the top panel will display all the NonBasic characters of “የ” based on their frequency of occurrence (see Appendix B). This case is shown in Fig 6.6

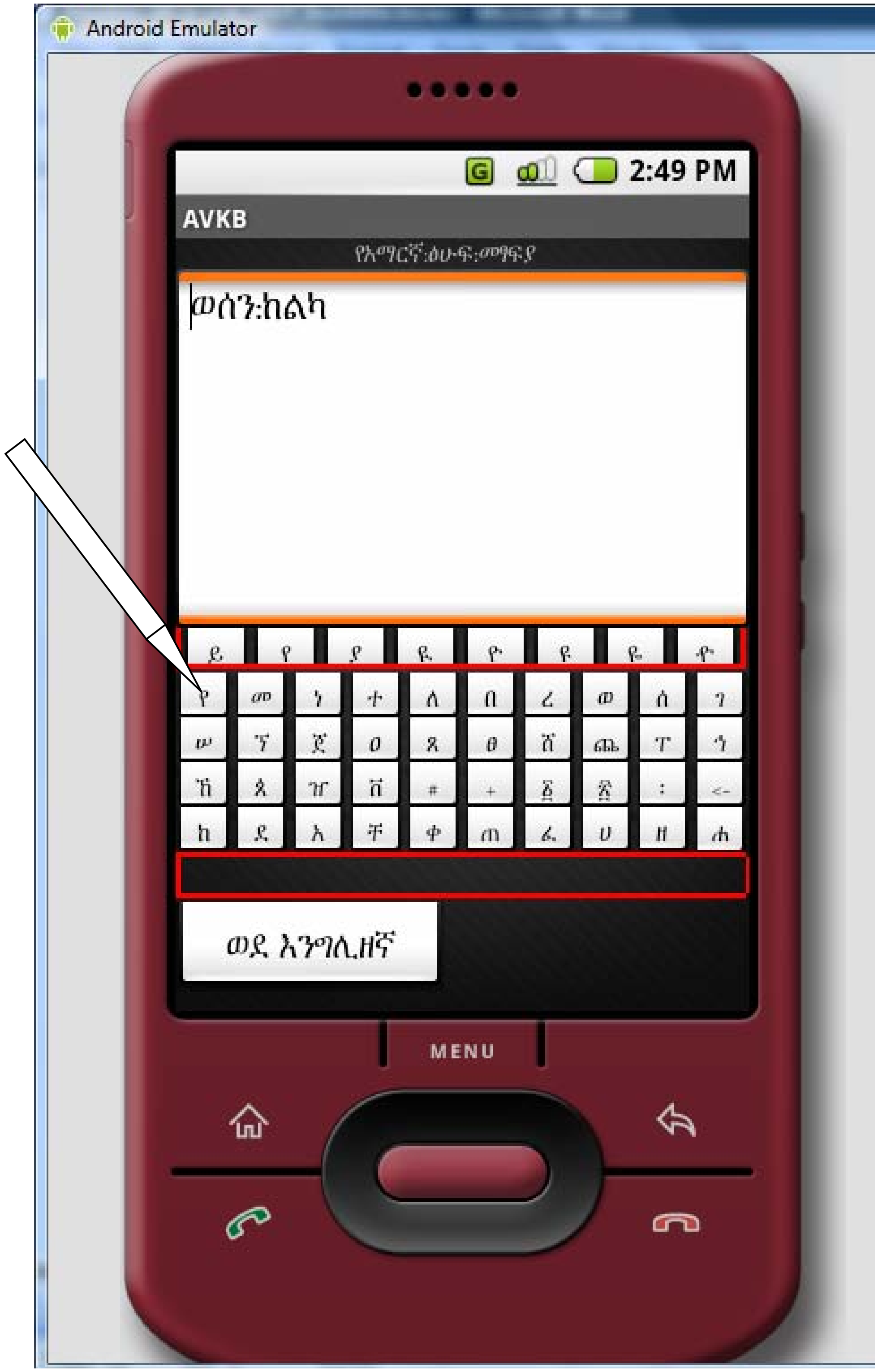

Figure 6-6 The appearance of the layout when pressing “የ”

The status of the system after pressing "ይ" from the NonBasic characters is shown in the figure 6.7

Figure 6-7 The appearance of **AVKB** after pressing "ይ"

Generally the direction of appearance for NonBasic (dependent) characters of the selected first order character is optimized so that the first two upper row base characters have NonBasic characters that will be displayed on the top panel of the layout and the last two rows base characters have NonBasic characters that will be displayed on the bottom panel. For instance, if the user has pressed on the key labeled with “፩” at the third row of the middle panel of the layout, then the appearance of the layout will look as shown in the figure 6.8.

Figure 6-8 The appearance of **AVKB** after pressing “፩”

The status of the system after pressing “፱ ” from the NonBasic characters is shown in the figure 6.9

Figure 6-9 The appearance of **AVKB** after pressing “፱”

This Amharic virtual keyboard is capable of providing input method to the user for all Amharic characters in the similar manner shown in the above figures (From fig.6.1 through fig. 6.9).

If a user wants to insert English characters, for example the word “Good Bye”, then he has to press the “**ወደ እንግሊዘኛ**” button, and the appearance of the layout after the response of the pressed button is shown in figure 6.10; he can then insert the required word by pressing the keys containing the letters of the word. Similarly if the user wants to switch the virtual keyboard layout to Amharic mode, then he has to press the “**To Amharic**” button as shown in figure 6.11.

Figure 6.10 The appearance of **AVKB** after pressing “ወደ እንግሊዘኛ” button

Figure 6.11 The appearance of **AVKB** after pressing “To Amharic” button

Chapter Seven

Conclusion and Recommendation

On screen keyboard is one form of virtual keyboard which is frequently available in handheld devices with touch screen capability, especially on PDA environment. The keys appearing on the screen are actually component of the software application that is programmed to map the corresponding character code based on the event triggered on key. In this project, a virtual keyboard is designed and implemented for Android OS PDA platform.

The layout design for languages having abundant characters is challenging, because it is not possible to make one to one mapping of key with character code. So, judicious technique is mandatory. In this work, a virtual keyboard layout is designed for Amharic language so that all Amharic characters, Amharic digits, Amharic numerals, Arabic numerals and Arithmetic operators are represented in usable manner. The designed layout is composed of three panels: the top, the middle and the bottom panel. The middle panel contains all the basic (or first order) Amharic characters, a key for Amharic punctuation characters selection , a delete character key, a key for Amharic digit selection, a key for Arithmetic operators selection, keys for Amharic and Arabic numerals selection. Both the top and the bottom panels are dedicated to display all the Non-Basic forms of the selected character in the middle panel.

After rigorous review of related work in the literature, analysis was made to optimize the arrangements of keys in all the three panels of the virtual keyboard to reduce the overall hand movement for Amharic text entry. For the implementation purpose, Android platform is used which is product for handheld devices by Google's OHA project. The Android Emulator together with Eclipse IDE and ADT plug-in were used for the implementation of the system. Android was selected for the implementation of the project as it's capable of supporting free type, a free and open source bitmap and vector font engine. Since no Amharic virtual keyboard layout is available so far, it is our belief that the output of this project will pave the way for Amharic language based application development on handheld devices.

It is a common practice to see many Amharic text writers to have the need to combine Amharic text with some English words in their text. To accommodate this requirement, the system is designed and implemented in such a way that users can switch between our Amharic Virtual keyboard and the QWERTY virtual keyboard. Thus, this gives the option for multi-text character set text entry on the Android OS PDA environment.

Future works of this project can be listed as follows:

- Currently, the proposed virtual keyboard layout was designed based on the character frequency analysis of [21] and [4]. The designed layout can be improved by using a large size corpus, to design a more optimal layout based on a better and effective character frequency analysis.
- Extending the Amharic virtual keyboard to incorporate the character for other local languages such as Tigrigna, Guragigna etc. that use the Ethiopic character set.
- Analyze relationship(correlation) between characters to improve the appearance of the layout
- Providing a module for editing any character at any location

References

1. Patent storm, US Patent 7259751-Input systems and methods for the evaluation and optimization, http://www.patentstorm.us/patents/7259751/description.html, Last accessed may 20, 2008.

2. Scott Mackenzie, Shawn X. Zhang and R. William Soukoreff, "Text entry using soft keyboards", Behavior & Information Technology, 1999.

3. Zhai, S., Hunter, M., & Smith, B. A., "The Metropolis Keyboard -- An Exploration of Quantitative Techniques for Virtual Keyboard Design", Proc. UIST'2000 – the 13th Annual ACM Symposium on User Interface Software and Technology, 2000.

4. Worku Alemu, "The application of OCR Techniques to the Amharic Script", An MSc thesis at Addis Ababa University Faculty of Informatics Department of Information Science, 1997.

5. Lewis, J. R., Potosnak, K. M., and Magyar, R. L., "Keys and keyboards", in M. Helander, T. K. Landauer, and P. V.Prabhu (eds) Handbook of human-computer interaction, 1997.

6. Shumin Zhai, Michael Hunter, and Barton A. Smith, "Performance Optimization of Virtual Keyboards", Human-COMPUTER INTERACTION, 2002.

7. The Fitaly Keyboard for Windows CE, Fitaly for the Pocket PC–CE, http://www.pocketpcfitaly.htm, Last accessed May 16, 2008.

8. Everything, Fitaly keyboard, http://www.index.pl.htm, Last accessed May 16, 2008.

9. The gadgeteer, Fitaly, http://www.fitaly_review.htm, Last accessed May 16, 2008.

10. Mackenzie, I.S. and S.X. Zhang. "The design and evaluation of a high-Performance Soft keyboard", in Proc. of CHI'99: ACM Conference on Human Factors in Computing Systems, 1999.

11. Existing text input methods,keyboards,http://www.cs.uta.fi/~poika/g/node6.html, Last accessed May 20, 2008,

12. Wikipedia, PDA, http://www.wikipedia.org/PDA.html, Last accessed February 28, 2008.

13. Scott Mackenzie and R. William Soukoreff, "Text Entry for Mobile Computing: Models and Methods, Theory and Practice", York university, 2000.

14. Andrew Alexander Patterson, "Dynamically Resizing Keys for PDA Text Entry", An MSc project at University of Strathclyde Department of Computer and Information Sciences, September 2005

15. Yonas Hailu, "Kirtas: Ethiopic Online Handwriting Recognition System Using Simplified Ethiopic Script", An MSc thesis at Addis Ababa University Faculty of Informatics Department of Computer Science July, 2007.

16. Yohannes Afework, "Automatic Amharic News Text Categorization", An MSc thesis at Addis Ababa University Faculty of Informatics Department of Computer

Science March, 2007.

17. IBM, physical keyboard layout, http://www-306.ibm.com/software/globalization/topics/keyboards/physical.jsp, Last accessed June 15, 2008.

18. The pragmatic programmers, Hello, Android: Introducing Google's Mobile Development Platform, http://www.pragprog.com/titles/eband/hello-android, Last Accessed May 18, 2006.

19. Linux decices.com, Android Developer Challenge announces first-round winners, http://www.linuxdevices.com/news/NS3168326017.html, Last Accessed May 18, 2006.

20. Linux devices.com, A developer's perspective on Google's android, http://linuxdevices.com/articles/AT9900056470.html, Last Accessed June 16, 2008.

21. Official Website of EICTDA, Standard keyboard layout for Ethiopic script, http://www.eictda.gov.et, Last Accessed May12, 2008.

22. Android - An Open Handset Alliance Project, What is Android, http://code.google.com/android/what-is-android.html, Last accessed June 13, 2008.

23. Android - An Open Handset Alliance Project, Android Emulator, http://code.google.com/android/reference/emulator.html, Last accessed June 13, 2008.

Appendix A

Character Family Frequency

Document: Ethiopian Constitution (የኢትዮጵያ ህገመንግስት) [21].

Character Family	Frequency
የ የ ሃረግ (የ ዩ ዪ ወዘተ)	3209
የ መ ሃረግ (መ ሙ ሚ ወዘተ)	3100
የ ነ ሃረግ (ነ ኑ ኒ ወዘተ)	3097
የ ተ ሃረግ (ተ ቱ ቲ ወዘተ)	3037
የ ለ ሃረግ (ለ ሉ ሊ ወዘተ)	2811
የ በ ሃረግ (በ ቡ ቢ ወዘተ)	2726
የ ረ ሃረግ (ረ ሩ ሪ ወዘተ)	2278
የ ወ ሃረግ (ወ ዉ ዊ ወዘተ)	1817
የ ሰ ሃረግ (ሰ ሱ ሲ ወዘተ)	1537
የ ገ ሃረግ (ገ ጉ ጊ ወዘተ)	1482
የ ከ ሃረግ (ከ ኩ ኪ ወዘተ)	1343
የ ደ ሃረግ (ደ ዱ ዲ ወዘተ)	1266
የ አ ሃረግ (አ ኡ ኢ ወዘተ)	1232
የ ቸ ሃረግ (ቸ ቹ ቺ ወዘተ)	1163
የ ቀ ሃረግ (ቀ ቁ ቂ ወዘተ)	855
የ ጠ ሃረግ (ጠ ጡ ጢ ወዘተ)	798
የ ፈ ሃረግ (ፈ ፉ ፊ ወዘተ)	605
የ ሀ ሃረግ (ሀ ሁ ሂ ወዘተ)	578
የ ዘ ሃረግ (ዘ ዙ ዚ ወዘተ)	563
የ ሐ ሃረግ (ሐ ሑ ሒ ወዘተ)	555
የ ሠ ሃረግ (ሠ ሡ ሢ ወዘተ)	455
የ ኘ ሃረግ (ኘ ኙ ኚ ወዘተ)	307
የ ጀ ሃረግ (ጀ ጁ ጂ ወዘተ)	226
የ ዐ ሃረግ (ዐ ዑ ዒ ወዘተ)	223
የ ጸ ሃረግ (ጸ ጹ ጺ ወዘተ)	195
የ ፀ ሃረግ (ፀ ፁ ፂ ወዘተ)	195
የ ሸ ሃረግ (ሸ ሹ ሺ ወዘተ)	165
የ ጨ ሃረግ (ጨ ጩ ጪ ወዘተ)	150
የ ፐ ሃረግ (ፐ ፑ ፒ ወዘተ)	117
የ ኀ ሃረግ (ኀ ኁ ኂ ወዘተ)	76
የ ጰ ሃረግ (ጰ ጱ ጲ ወዘተ)	71
የ ዠ ሃረግ (ዠ ዡ ዢ ወዘተ)	7
የ ቨ ሃረግ (ቨ ቩ ቪ ወዘተ)	3

Appendix B

Output of Character Frequency Analysis

1. Document: Ethiopian Constitution (የኢትዮጵያ ህገመንግስት) [21].

Character Order	Frequency	%
ሳድስ (ህ፣ ል፣ ም ወዘተ)	14526	40.05
ግእዝ (ሀ፣ ለ፣ መ ወዘተ)	10824	29.84
ራብእ (ሃ፣ ላ፣ ማ ወዘተ)	5285	14.57
ሳልስ (ሂ፣ ሊ፣ ሚ ወዘተ)	1581	4.36
ሳብእ (ሆ፣ ሎ፣ ሞ ወዘተ)	1498	4.13
ካእብ (ሁ፣ ሉ፣ ሙ ወዘተ)	1225	3.38
ሃምስ (ሄ፣ ሌ፣ ሜ ወዘተ)	1170	3.23
አረቤ (ኋ፣ ሏ፣ ሟ ወዘተ)	159	0.44
	36268 *	**100.00%**

** Spaces and punctuations not counted*

2. Document: Turnover tax proclamation 380/1995 EC

Character Order	Frequency	%
ሳድስ (ህ፣ ል፣ ም ወዘተ)	6167	37.52%
ግእዝ (ሀ፣ ለ፣ መ ወዘተ)	5683	34.57%
ራብእ (ሃ፣ ላ፣ ማ ወዘተ)	2589	15.75%
ሳልስ (ሂ፣ ሊ፣ ሚ ወዘተ)	737	4.48%
ሳብእ (ሆ፣ ሎ፣ ሞ ወዘተ)	542	3.30%
ካእብ (ሁ፣ ሉ፣ ሙ ወዘተ)	448	2.73%
ሃምስ (ሄ፣ ሌ፣ ሜ ወዘተ)	247	1.50%
አረቤ (ኋ፣ ሏ፣ ሟ ወዘተ)	25	0.15%
	16438	100.00%

Appendix C

FREQUENCY OF OCCURRENCE OF AMHARIC CHARACTERS [4].

No	Fidel	Frequency	Cumulative Frequency	Percent	Cumulative Percent
1	ን	946	946	4.6695	4.6695
2	ት	828	1774	4.0871	8.7566
3	በ	730	2504	3.6033	12.3599
4	የ	713	3217	3.5194	15.8794
5	ው	618	3835	3.0505	18.9299
6	አ	565	4400	2.7889	21.7187
7	ር	559	4959	2.7593	24.4780
8	መ	540	5499	2.6655	27.1435
9	ል	506	6005	2.4977	29.6411
10	ም	495	6500	2.4434	32.0845
11	ለ	489	6989	2.4137	34.4982
12	ስ	467	7456	2.3051	36.8034
13	ተ	442	7898	2.1817	38.9851
14	ያ	370	8268	1.8263	40.8115
15	ይ	333	8601	1.6437	42.4552
16	ች	329	8930	1.6240	44.0792
17	ና	315	9245	1.5549	45.6340
18	ኑ	307	9552	1.5154	47.1494
19	እ	292	9844	1.4413	48.5907
20	ገ	281	10125	1.3870	49.9778
21	ማ	280	10405	1.3821	51.3599
22	ድ	275	10680	1.3574	52.7173
23	ከ	271	10951	1.3377	54.0550
24	ደ	262	11213	1.2933	55.3482
25	ግ	259	11472	1.2784	56.6267
26	ረ	252	11724	1.2439	57.8706
27	ላ	250	11974	1.2340	59.1046
28	ራ	250	12224	1.2340	60.3386
29	ብ	247	12471	1.2192	61.5578
30	ባ	243	12714	1.1995	62.7573
31	ሚ	233	12947	1.1501	63.9074
32	ወ	232	13179	1.1452	65.0526
33	ታ	214	13393	1.0563	66.1089
34	ሰ	197	13590	0.9724	67.0813
35	ክ	173	13763	0.8539	67.9352
36	ጥ	171	13934	0.8441	68.7793
37	ቀ	156	14090	0.7700	69.5493
38	ህ	149	14239	0.7355	70.2848
39	ሪ	149	14388	0.7355	71.0203
40	ዳ	145	14533	0.7151	71.7360
41	ጠ	136	14669	0.6713	72.4073

42	ካ	132	14801	0.6516	73.0589
43	ዋ	128	14929	0.6318	73.6907
44	ሳ	118	15047	0.5825	74.2732
45	ቅ	116	15163	0.5726	74.8457
46	ቶ	115	15278	0.5676	75.4134
47	ሁ	111	15389	0.5479	75.9613
48	ጋ	111	15500	0.5479	76.5092
49	ኒ	107	15607	0.5282	77.0374
50	ሆ	106	15713	0.5232	77.5606
51	ዘ	106	15819	0.5232	78.0838
52	ሉ	105	15924	0.5183	78.6021
53	ዓ	103	16027	0.5084	79.1105
54	ቻ	102	16129	0.5035	79.6140
55	ዎ	97	16226	0.4788	80.0928
56	ፊ	97	16323	0.4788	80.5716
57	ቱ	95	16418	0.4689	81.0405
58	ፍ	94	16512	0.4640	81.5045
59	ሮ	89	16601	0.4393	81.9438
60	ሩ	81	16682	0.3998	82.3467
61	ሥ	81	16763	0.3998	82.7435
62	ኛ	79	16842	0.3900	83.1334
63	ጣ	79	16921	0.3900	83.5234
64	ዲ	76	16997	0.3751	83.8985
65	ሌ	72	17069	0.3554	84.2539
66	ጅ	71	17140	0.3505	84.6044
67	ዚ	67	17207	0.3307	84.9351
68	ቃ	65	17272	0.3208	85.2559
69	ጉ	63	17335	0.3110	85.5669
70	ዝ	61	17396	0.3011	85.8680
71	ሲ	58	17454	0.2863	86.1543
72	ዊ	57	17511	0.2814	86.4357
73	ፌ	56	17567	0.2764	86.7121
74	ቢ	55	17622	0.2715	86.9836
75	ሎ	55	17677	0.2715	87.2550
76	ዮ	55	17732	0.2715	87.5265
77	ዜ	54	17786	0.2665	87.7931
78	ጃ	54	17840	0.2665	88.0596
79	ኢ	52	17892	0.2567	88.3163
80	ቤ	51	17943	0.2517	88.5680
81	ሞ	51	17994	0.2517	88.8198
82	ቡ	51	18045	0.2517	89.0715
83	ኮ	49	18094	0.2419	89.3134
84	ሽ	48	18142	0.2369	89.5503
85	ህ	48	18190	0.2369	89.7873
86	ፋ	47	18237	0.2320	90.0193
87	ሙ	45	18282	0.2221	90.2414

88	ዩ	45	18327	0.2221	90.4635
89	ሻ	42	18369	0.2073	90.6708
90	ሊ	41	18410	0.2024	90.8732
91	ዛ	41	18451	0.2024	91.0765
92	ጊ	40	18491	0.1974	91.2730
93	ሬ	38	18529	0.1876	91.4606
94	ቦ	38	18567	0.1876	91.4682
95	ኖ	38	18605	0.1876	91.8357
96	ጭ	38	18643	0.1876	92.0233
97	ኔ	37	18680	0.1826	92.2059
98	ሀ	36	18716	0.1777	92.3836
99	ጨ	36	18752	0.1777	92.5613
100	ቁ	35	18787	0.1728	92.7341
101	ዕ	35	18822	0.1728	92.9069
102	ጃ	34	18856	0.1678	93.0747
103	ቻ	33	18889	0.1629	93.2376
104	ኩ	33	18922	0.1629	93.4005
105	ጎ	32	18954	0.1580	93.5584
106	ፐ	30	18984	0.1481	93.7065
107	1 and ፩	30	19014	0.1481	93.8546
108	ሱ	30	19044	0.1481	94.0027
109	ጵ	29	19073	0.1431	94.1458
110	ኃ	29	19102	0.1431	94.2890
111	ቆ	27	19129	0.1333	94.4222
112	ቱ	26	19155	0.1283	94.5506
113	ሄ	26	19181	0.1283	94.6789
114	ዱ	26	19207	0.1283	94.8072
115	ቋ	26	19233	0.1283	94.9356
116	ሃ	25	19258	0.1234	95.0590
117	ሜ	25	19283	0.1234	95.1824
118	ሽ	25	19308	0.1234	95.3058
119	ኝ	25	19333	0.1234	95.4292
120	ኤ	24	19357	0.1185	95.5477
121	ሶ	23	19380	0.1135	95.6612
122	ጬ	23	19403	0.1135	95.7747
123	ኅ	23	19426	0.1135	95.8882
124	ሔ	23	19449	0.1135	96.0018
125	9 and ፱	22	19471	0.1086	96.1104
126	ቴ	22	19493	0.1086	96.2190
127	ኙ	22	19515	0.1086	96.3276
128	ኪ	22	19537	0.1086	96.4362
129	ጸ	22	19559	0.1086	96.5447
130	2 and ፪	21	19580	0.1037	96.6484
131	ሴ	21	19601	0.1037	96.7521
132	ቂ	21	19622	0.1037	96.8557
133	ፖ	21	19643	0.1037	96.9594

134	ሐ	21	19664	0.1037	97.0630
135	ቲ	20	19684	0.0987	97.1618
136	ዶ	20	19704	0.0987	97.2605
137	ዴ	19	19723	0.0938	97.3543
138	ጽ	18	19741	0.0888	97.4431
139	4 and ፬	15	19756	0.0740	97.5172
140	ዘ	15	19771	0.0740	97.5912
141	ሥ	15	19786	0.0740	97.6652
142	ኳ	15	19081	0.0740	97.7393
143	ኘ	14	19815	0.0691	97.8084
144	ቄ	14	19829	0.0691	97.8775
145	ኞ	14	19843	0.0691	97.9466
146	ፓ	13	19856	0.0642	98.0108
147	ጡ	13	19869	0.0642	98.0749
148	ጠ	13	19882	0.0642	98.1391
149	ሳ	13	19895	0.0642	98.2033
150	ዓ	12	19907	0.0592	98.2625
151	ኦ	12	19919	0.0592	98.3217
152	ፌ	12	19931	0.0592	98.3810
153	ፒ	12	19943	0,0592	98.4402
154	ቷ	12	19955	0.0592	98.4994
155	6 and ፮	11	19966	0.0543	98.5537
156	ጤ	11	19977	0.0543	98.6080
157	፻	11	19988	0.0543	98.6623
158	ፀ	10	19998	0.0494	98.7117
159	ጁ	10	20008	0.0494	98.7610
160	ጆ	10	20018	0.0494	98.8104
161	ፉ	9	20027	0.0444	98.8548
162	0	9	20036	0.0444	98.8993
163	ኺ	9	20045	0.0444	98.9437
164	ጂ	9	20054	0.0444	98.9881
165	ጌ	9	20063	0.0444	99.0325
166	8 and ፰	9	20072	0.0444	99.0770
167	ጪ	5	20077	0.0247	99.1016
168	ሂ	8	20085	0.0395	99.1411
169	5 and ፭	8	20093	0.0395	99.1806
170	ኽ	8	20101	0.0395	99.2201
171	፳	8	20109	0.0395	99.2596
172	ኾ	7	20116	0.0346	99.2941
173	ዙ	7	20123	0.0346	99.3287
174	ቨ	7	20130	0.0346	99.3632
175	ቾ	6	20136	0.0296	99.3929
176	ኋ	6	20142	0.0296	99.4225
177	ፁ	5	20147	0.0247	99.4472
178	ዐ	5	20152	0.0247	99.4718
179	ሏ	5	20157	0.0247	99.4965

180	፲	5	20162	0.0247	99.5212
181	ቼ	4	20166	0.0197	99.5409
182	ዬ	4	20170	0.0197	99.5607
183	ፎ	4	20174	0.0197	99.5804
184	ሦ	4	20178	0.0197	99.6002
185	ጻ	4	20182	0.0197	99.6199
186	ሯ	4	20186	0.0197	99.6397
187	ቪ	4	20190	0.0197	99.6594
188	ዓ	4	20194	0.0197	99.6792
189	ሟ	4	20198	0.0197	99.6989
190	ኗ	4	20202	0.0197	99.7186
191	3 and ፫	3	20205	0.0148	99.7335
192	ሹ	3	20208	0.0148	99.7483
193	ዌ	3	20211	0.0148	99.7631
194	ቧ	3	20214	0.0148	99.7779
195	ጳ	3	20217	0.0148	99.7927
196	ዖ	3	20220	0.0148	99.8075
197	ጇ	3	20223	0.0148	99.8223
198	ሏ	3	20226	0.0148	99.8371
199	፶	2	20228	0.0099	99.8470
200	፹	2	20230	0.0099	99.8569
201	ኜ	2	20232	0.0099	99.8667
202	ኡ	2	20234	0.0099	99.8766
203	ጢ	2	20236	0.0099	99.8865
204	ጬ	2	20238	0.0099	99.8963
205	ፁ	2	20240	0.0099	99.9062
206	ፔ	2	20242	0.0099	99.9161
207	ኸ	2	20244	0.0099	99.9260
208	ሒ	2	20246	0.0099	99.9358
209	ቺ	1	20247	0.0049	99.9408
210	፺	1	20248	0.0049	99.9457
211	ቫ	1	20249	0.0049	99.9506
212	ቬ	1	20250	0.0049	99.9556
213	ኚ	1	20251	0.0049	99.9605
214	7 and ፯	1	20252	0.0049	99.9654
215	ሼ	1	20253	0.0049	99.9704
216	ጮ	1	20254	0.0049	99.9753
217	ቄ	1	20255	0.0049	99.9803
218	ፆ	1	20256	0.0049	99.9852
219	ፐ	1	20257	0.0049	99.9901
220	ጧ	1	20258	0.0049	99.9951
221	ፑ	1	20259	0.0049	100.0000

Appendix D

The following table shows list of Amharic core characters [4].

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

Appendix E

The numeral symbols of Amharic [4].

1	፩
2	፪
3	፫
4	፬
5	፭
6	፮
7	፯
8	፰
9	፱
10	፲
20	፳
30	፴
40	፵
50	፶
60	፷
70	፸
80	፹
90	፺
100	፻
1000	፼

Appendix F

Seven forms of Amharic characters for (በ , ገ , ደ) [16].

consonant	1st Order	2nd Order	3rd Order	4th Order	5th Order	6th Order	7th Order	
በ	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	The seven forms of በ
	ብ ኧ	ብ ኡ	ብ ኢ	ብ ኧ	ብ ኤ	ብ ኧ	ብ ኦ	Consonant – Vowel representation
	bä	Bu	bi	ba	be	B	Bo	Represented sound
ገ	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	The seven forms of ገ
	ግ ኧ	ግ ኡ	ግ ኢ	ግ ኣ	ግ ኤ	ግ ኧ	ግ ኦ	Consonant- Vowel representation
	gä	Gu	gi	ga	ge	G	Go	Represented sound
ደ	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	The seven forms of ደ
	ድ ኧ	ድ ኡ	ድ ኢ	ድ ኧ	ድ ኤ	ድ ኧ	ድ ኦ	Consonant- Vowel representation
	dä	Du	di	da	de	D	Do	Represented sound

Declaration

I, the undersigned, declare that this project is my original work and has not been presented for degree in any other university, and that all sources of materials used for the project have been acknowledged.

Declared by:

Name: ______________________________

Signature: ______________________________

Date: ______________________________

Confirmed by advisor:

Name: ______________________________

Signature: ______________________________

Date: ______________________________

Place and date of submission: Addis Ababa, July 2008.