Tourist Application

Image Data Fetcher

Submitted By

Arbaz

Student ID 401887

Bachelor of Science in Information Technology

Under the guidance of

REMOVED PROF NAME

2016-2017

Maharashtra College of Arts Science and Commerce

Mumbai - 400008

Project Overview

The next generation open operating systems are not on desktops or mainframes but on the small mobile devices people carry every day. The openness of these new environments leads to new applications and markets and enables greater integration. As the demand grows for mobile phone applications, research in optical character recognition, a technology well developed for scanned documents, is shifting focus to the recognition of text embedded in digital photographs.

Optical character recognition (OCR) is a powerful tool for bringing information from our analog lives into the increasingly digital world. This technology has long seen use in building digital libraries, recognizing text from natural scenes, understanding hand-written office forms etc.

The Personal Computer and the Internet have found revolutionary ways to connect people, to entertain them and let them exchange information. But none of these is able to reach each person anywhere and anytime like the cell phone does. Current global mobile phone usage is 4 billion, which is equivalent to around half of the worlds population. The cell phone has become very important in today’s society

Google has come out with the new open and comprehensive platform for mobile devices called Android. It includes an operating system, middleware, user-interface and applications[7].

1. Introduction

1.1. Theory behind the project concept

It will be very helpful for tourists to understand & adapt local languages with ease.
This Application has an advanced search feature so that recognized as well as translated text can be used to copy, paste, share and search for travel related queries like museums, places, restaurants, books, culture, hotels, etc
The application enables the users to get text translate as ease as a button click. The camera captures the text and returns the translated result in real time.
This application enables people to understand any native country language and also seamlessly fire a travel specific search to get the info regarding the country like museums, restaurants, hotels, culture, temples, books, movies, songs, etc.

1.2. Problem Definition

In Indian market there was always a huge demand of such an Android Application, which would enable a tourist sitting in a restaurant to capture, OCR and translate the hindi language menu in his own country language to order his favourite dish without any need to know the language of menu. This application enables people to understand any native country language and also seamlessly fire a travel specific search to get the info regarding the country like museums, restaurants, hotels, culture, temples, books, movies, songs, etc. Image processing on mobile phones is a new and exciting field with many challenges due to limited hardware, limited CPU utilization, image Quality and connectivity

1.3. Need for Project

Our project, enables travellers and tourists to easily capture the native country language books pages, signboards, banners and hotel menus etc. The built-in OCR converts the text embedded in the captured image into Unicode text format. It also provides translation facility so that Tourists can translate the native language Unicode text into their own country language. This Application has an advanced search feature so that recognized as well as translated text can be used to copy, paste, share and search for travel related queries like museums, places, restaurants, books, culture, hotels, etc.this would prove enormously beneficial with respect to the aspects about localization being a common phenomenon now-a-days. Also android platform has been increasingly being common in accordance with its features like low-cost, customizable, lightweight operating system and more.The application describes the following characteristics:

1) It will be very helpful for tourists to understand & adapt local languages with ease.
2) This Application has an advanced search feature so that recognized as well as translated text can be used to copy, paste, share and search for travel related queries like museums, places, restaurants, books, culture, hotels, etc
3) The application enables the users to get text translate as ease as a button click. The camera captures the text and returns the translated result in real time.
4) This application enables people to understand any native country language and also seamlessly fire a travel specific search to get the info regarding the country like museums, restaurants, hotels, culture, temples, books, movies, songs, etc.
5) It overcomes the existing problems with OCR technology i.e. limited memory and limited processing power challenge moreover also overcome the problem of from networking delay

2.4 Literature Surveyed:

Image data fetcher is an android application that is useful for native tourist and travelers who posses android smart phones. This application enables them to captures images of hoardings, signboards, book pages etc. and convert the text embedded in the captured images into Unicode text format.

We got the idea of this project from a research paper “International Journal of Computer Applications (0975 – 8887)”.

This research paper contains the overall idea of the project. From this concept we thought of developing an application which we will contain additional features like text to speech and websearch

2. Analysis & Design

2.1. Software Development

Android: Android is a Linux-based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005,Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices[7].

Tesseract OCR engine: Tesseract is a free software optical character recognition engine for various operating systems.Tesseract is probably the most accurate open source OCR engine available. Combined with the Leptonica Image Processing Library it can read a wide variety of image formats and convert them to text in over 60 languages. It was one of the top 3 engines in the 1995 UNLV Accuracy test. Between 1995 and 2006 it had little work done on it, but since then it has been improved extensively by Google. It is released under the Apache License 2.0[7].

Bing Translator API: Consists of libraries and classes that can be used for translation coding. Automatic translation enables communication, the exchange of information, and conducting business across language barriers. Microsoft Translator delivers a versatile, instant, and cost-effective automatic translation service to any destination; helping to break the language barrier for businesses, developers and users alike. Translation is a feature, not a destination; it is best delivered within the service or application the user is experiencing. The rich and accessible translation API empowers Developers and Solution Providers to deliver these highly valuable and relevant experiences for their customers where they need them most.

Tesseract Android Tools: Tesseract Tools for Android provides a set of Android APIs and build files for the Tesseract OCR and Leptonica image processing libraries[11].

Android SDK: The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

ADT: Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

Eclipse: In computer programming, Eclipse is a multi-language software development environment comprising a base workspace and an extensible plug-insystem for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by Means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Fortran, Haskell, Perl, PHP, Python, etc.

2.2. Flow of Project

2.2.1. Preliminary Survey:

The language barrier among tourist was one of the major difficulties when traveeling and thus our application helps tourist to overcome this problem.We proposed a system to translate images taken with mobile phone camera from various languages to required language.

In future generation of technology this application can be made very useful as people will be travelling across borders for various purpose.

2.2.2. Feasibility Study:

The very first phase in any system developing life cycle is preliminary investigation. The feasibility study is a major part of this phase. A measure of how beneficial or practical the development of any information system would be to the organization is the feasibility study of the development software can be studied in terms of the following aspects:

1.Operational Feasibility.

2.Technical Feasibility.

3.Economical feasibility.

4.Motivational Feasibility.

5.Legal Feasibility

OPERATIONAL FEASIBILITY :

The site will reduce the time consumed to maintain manual records and is not tiresome and cumbersome to maintain the records. Hence operational feasibility is assured.

TECHNICAL FEASIBILITY :

- At least 166 MHz Pentium Processor or Intel compatible processor.
- At least 16 MB RAM.
- 14.4 kbps or higher modem.

- A video graphics card.
- A mouse or other pointing device.
- At least 3 MB free hard disk space.
- Microsoft Internet Explorer 4.0 or higher.

ECONOMICAL FEASIBILTY :

Once the hardware and software requirements get fulfilled, there is no need for the user of our system to spend for any additional overhead.

For the user, the web site will be economically feasible in the following aspects:

- The web site will reduce a lot of paper work. Hence the cost will be reduced.
- Our web site will reduce the time that is wasted in manual processes.
- The storage and handling problems of the registers will be solved.

LEGAL FEASIBILITY **:**

The licensed copy of the required software is quite cheap and easy to get. So from legal point of view the proposed system is legally feasible

2.2.3. Cost Analysis:

Cost Estimation:

Sr. No.	Expense Description	Value
1	Number of the programmers	1
2	Server Cost	Rs. 1000
3	Documentation Cost	Rs. 500

3	Internet charges and other expenses	Rs. 1000
	Total	**Rs. 3500**

Table 3.2.3.1 :- cost estimation table

2.2.4. Process Model

Process Model Used for the Project:

Software process model deals with the model which we are going to use for the development of the project. There are many software process models available but while choosing it we should choose it according to the project size that is whether it is industry scale project or big scale project or medium scale project.

Accordingly the model which we choose should be suitable for the project as the software process model changes the cost of the project also changes because the steps in each software process model varies.

This software is build using the waterfall mode. This model suggests work cascading from step to step like a series . It consists of the following steps in the following manner

Waterfall model:

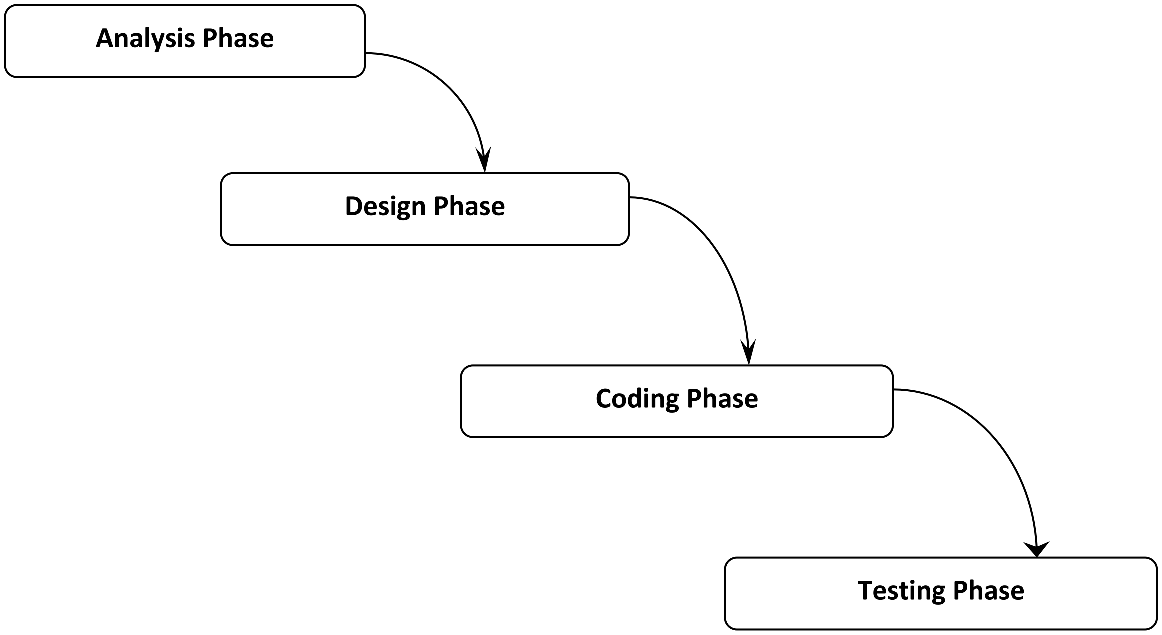


Fig 3.2.4:- WaterFall model

The construction phase builds increments of the system. Each increment is developed using a waterfall approach. This includes detailed analysis and design for the use cases in the increment and coding and testing of the event processors that implement the sequence of events defined by the use casesTransition

The transition phase (not shown in the figure) is the last phase in the project. This may include such things a performance tuning and rollout to all users.The Incremental development helped us because our team size was small, just two members to be precise, as compared to the vastness and complexity of the project. It helped us meet with business deadlines by providing customers with prototypes, thereby enabling partial functionality without inordinate delay.

We delivered our first prototype, the core product, at no charge to the organization for a trial period. This helped the customer get the feel of the system and moreover gave us useful feedback that helped us plan what could be implemented in the next prototype. The feedback helped us modify the core products to better meet the needs of the customer and the delivery of additional features & functions

Analysis Phase:

To attack a problem by breaking it into sub-problems. The objective of analysis is to determine exactly what must be done to solve the problem. Typically, the system’s logical elements (its boundaries, processes, and data) are defined during analysis.

Design Phase:

The objective of design is to determine how the problem will be solved. During design the analyst’s focus shifts from the logical to the physical.□ Data elements are grouped to form physical data structures, screens, reports, files and databases.

Coding Phase:

The system is created during this phase. Programs are coded, debugged, documented, and tested. New hardware is selected and ordered. Procedures are written and tested. End-user documentation is prepared. Databases and files are initialized. Users are trained.

Testing Phase:

Once the system is developed, it is tested to ensure that it does what it was designed to do. After the system passes its final test and any remaining problems are corrected, the system is implemented and released to the user.

2.2.5. Data Flow Diagrams:

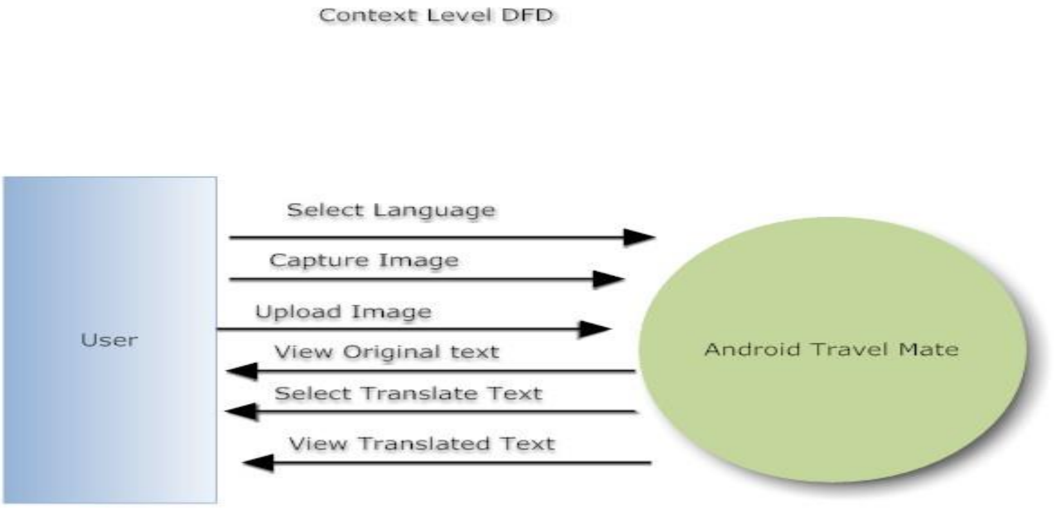


Fig 3.2.5.1:-level 0 -DFD

Level 1

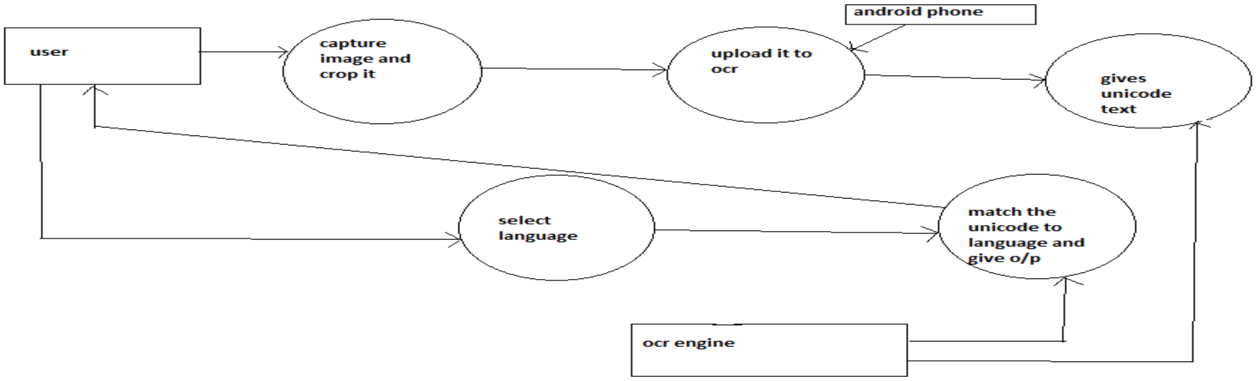


Fig 3.2.5.2 :level 1-DFD

Level 2

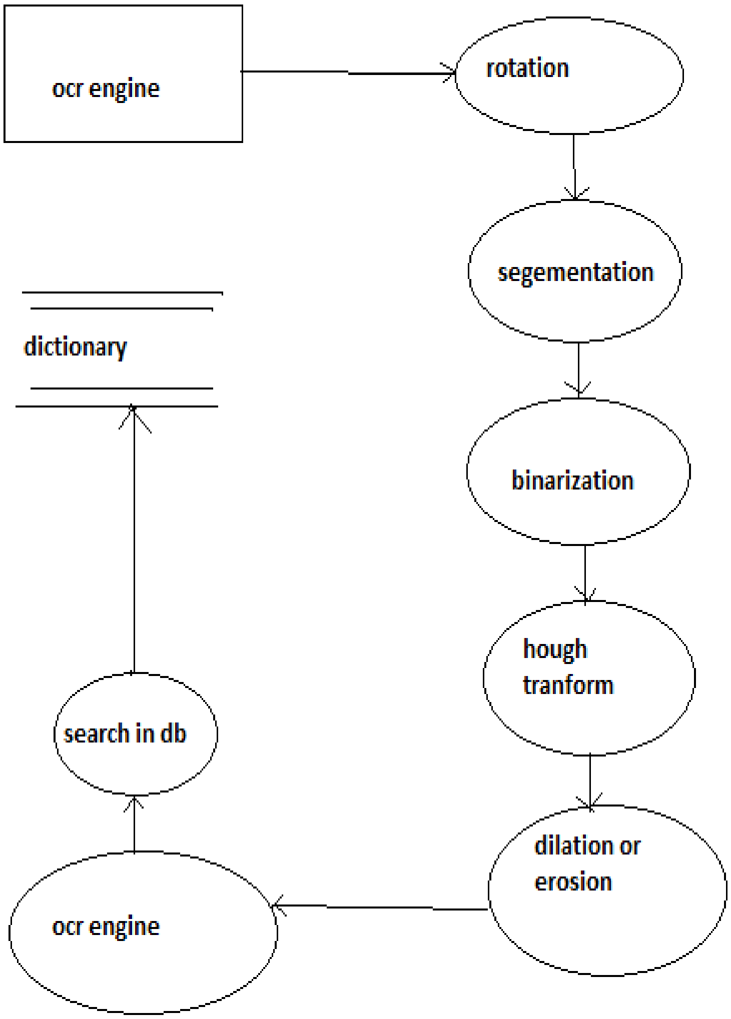


Fig 3.2.5.3:-level 2-DFD

2.3. UML Diagrams

Use case:

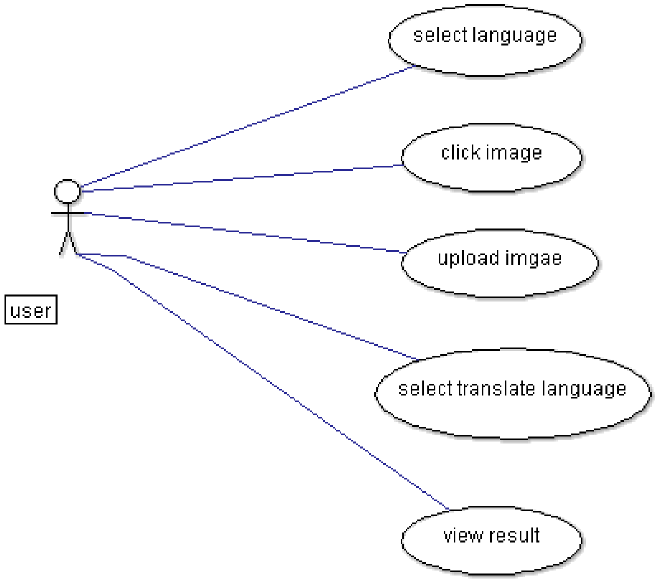


Fig 3.3.1-use case

Collaboration:

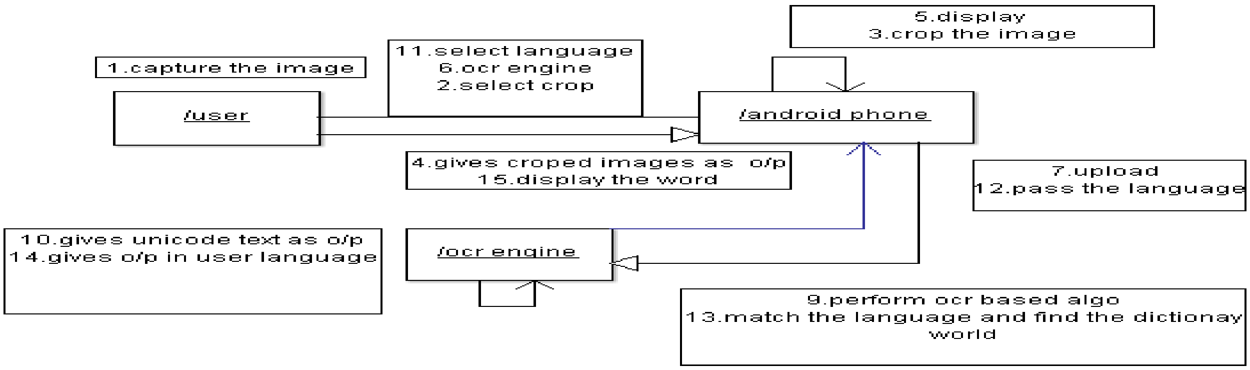


Fig 3.3.2:-collaboration diagram

System architecture

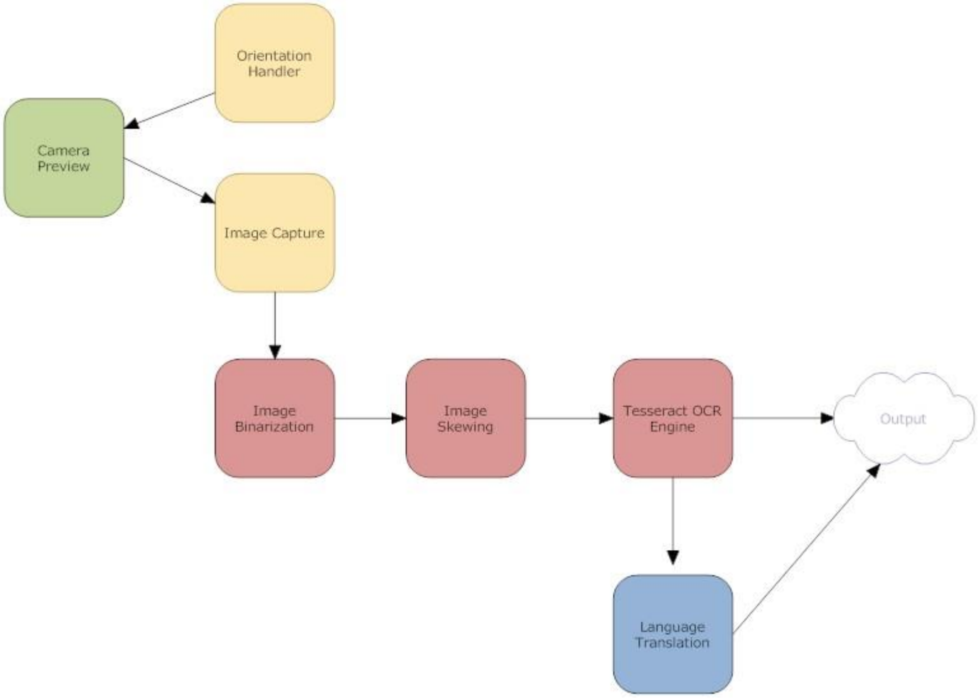


Fig 3.3.3:-system diagram

Actitvity:

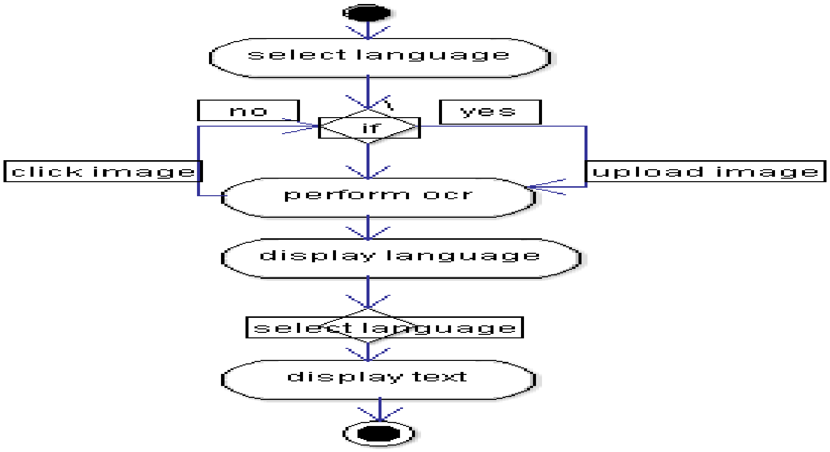


Fig 3.3.4:-activity diagram

2.4. Technologies Used

2.4.1. Hardware & Software Requirements:

Hardware:

- 256 MB RAM.
- 8 GB HDD.
- Intel 1.66 GHz Processor Pentium 4
- GPRS enabled Mobile Phone with Android.

Software :

- Windows XP s
- SDK for Android
- Bing translator

2.4.2. Introduction to Programming Tools:

Android for Mobile Application:

Android is an operating system for mobile devices such as smart phones and tablet computers. It is developed by the Open Handset Alliance led by Google. Google purchased the initial developer of the software, Android Inc., in 2005. The unveiling of the Android distribution on November 5, 2007 was announced with the founding of the Open Handset Alliance, a consortium of 84 hardware, software and telecommunication companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache License, a free software license. The Android Open Source Project (AOSP) is tasked with the maintenance and further development of Android. Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run compiled Java code. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. Developers write primarily in a customized version of Java. There are currently more than 520,000 apps available for Android[2].

Tesseract OCR Engine version 3.01

Today, Tesseract is considered one of the most accurate open source OCR engines available. Tesseract OCR Engine was one of the best 3 engines in 1995 UNLV Accuracy Test. Between 1995 and 2006 however; there was little activity in Tesseract , until it was open sourced by HP and UNLV in 2005. It was again re-released to the open source community in August of 2006 by Google. Tesseract has ability to train for newer language and scripts as well. A complete overview of Tesseract OCR engine can be found in. While Tesseract was originally developed for English, it has since been extended to recognize French, Italian, Catalan, Czech, Danish, Polish, Bulgarian, Russian, Greek, Korean, Spanish, Japanese, Dutch, Chinese, Indonesian, Swedish, German, Thai, Arabic, and Hindi etc. Training the Tesseract OCR Engine for Hindi language requires in-depth knowledge of Devanagari script in order to collect the character set. Moreover, Tesseract OCR Engine does not just require training of the collected dataset but also to tackle the character segmentation and clubbing issues based on the script specific features i.e. Shirorekha, maatraetc.To prepare the customized training data. It has following phases described below:

*Training image generation:*It involves the sufficiently spaced out single font specific text image creation. For each new font Tesseract OCR Engine suggests preparation of a new image file.

*Box file generation:*The information about the Bounding Boxes for all the characters present in the training image is generated for specifying language specific components in the box file. The default generated Bounding boxes can easily be edited using box file editors i.e. cowboxer tool etc.

*Train file generation:*Box file editors also allow editing the corresponding Unicode characters against appropriate Bounding boxes.

Character set file generation:Character set file is required to specify the information like uppercase, lowercase, digits, punctuation marks etc. about the Unicode characters. Since Devanagari does not distinguish upper and lower case characters, only digits and punctuation marks have to be specified.

*Font properties selection:*Font properties like italic, bold, fixed, serif etc. are required to be specified before training the data. In this work only normal fonts have been considered.

*Feature extraction:*This phase extracts the features of the shape of characters from the Training Data Image.

Clustering:This phase clusters the character shape features into prototypes.

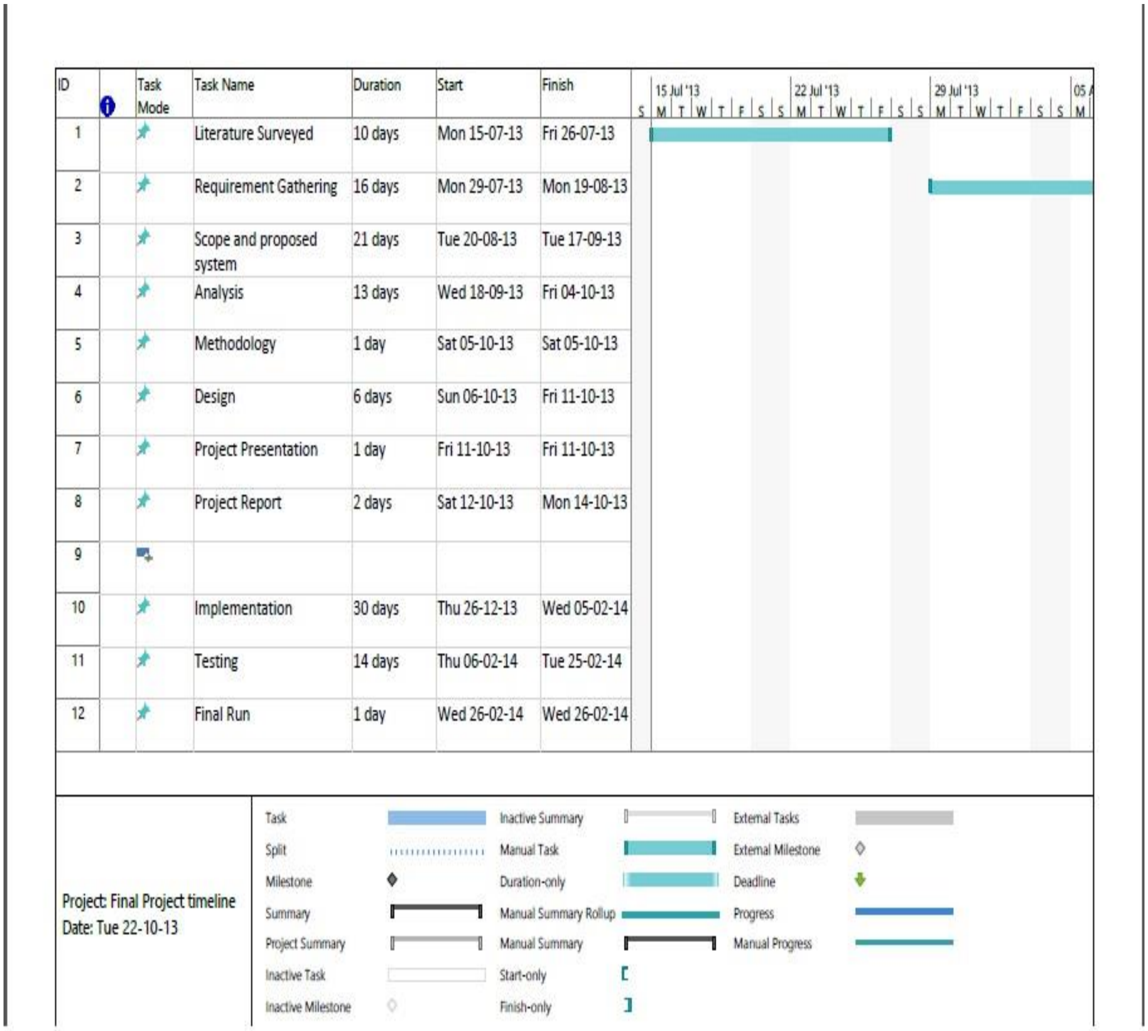
*Dictionary data preparation:*Tesseract may use up to 5 types of Dictionary files which are converted into Directed Acyclic Word Graph (DAWG) files.

*Post processing ambiguity removal:*Editing the unicharambigs file allows removing the intrinsic ambiguity between two similar looking characters or their combinations by using a substitution rule.

3.Project Time & Task Distribution

3.1. Timeline Chart:

Chart 1



ID		Task Mode	Task Name	Duration	Start	Finish
1			Literature Surveyed	10 days	Mon 15-07-13	Fri 26-07-13
2			Requirement Gathering	16 days	Mon 29-07-13	Mon 19-08-13
3			Scope and proposed system	21 days	Tue 20-08-13	Tue 17-09-13
4			Analysis	13 days	Wed 18-09-13	Fri 04-10-13
5			Methodology	1 day	Sat 05-10-13	Sat 05-10-13
6			Design	6 days	Sun 06-10-13	Fri 11-10-13
7			Project Presentation	1 day	Fri 11-10-13	Fri 11-10-13
8			Project Report	2 days	Sat 12-10-13	Mon 14-10-13
9						
10			Implementation	30 days	Thu 26-12-13	Wed 05-02-14
11			Testing	14 days	Thu 06-02-14	Tue 25-02-14
12			Final Run	1 day	Wed 26-02-14	Wed 26-02-14

Project: Final Project timeline
Date: Tue 22-10-13

Chart 2 :

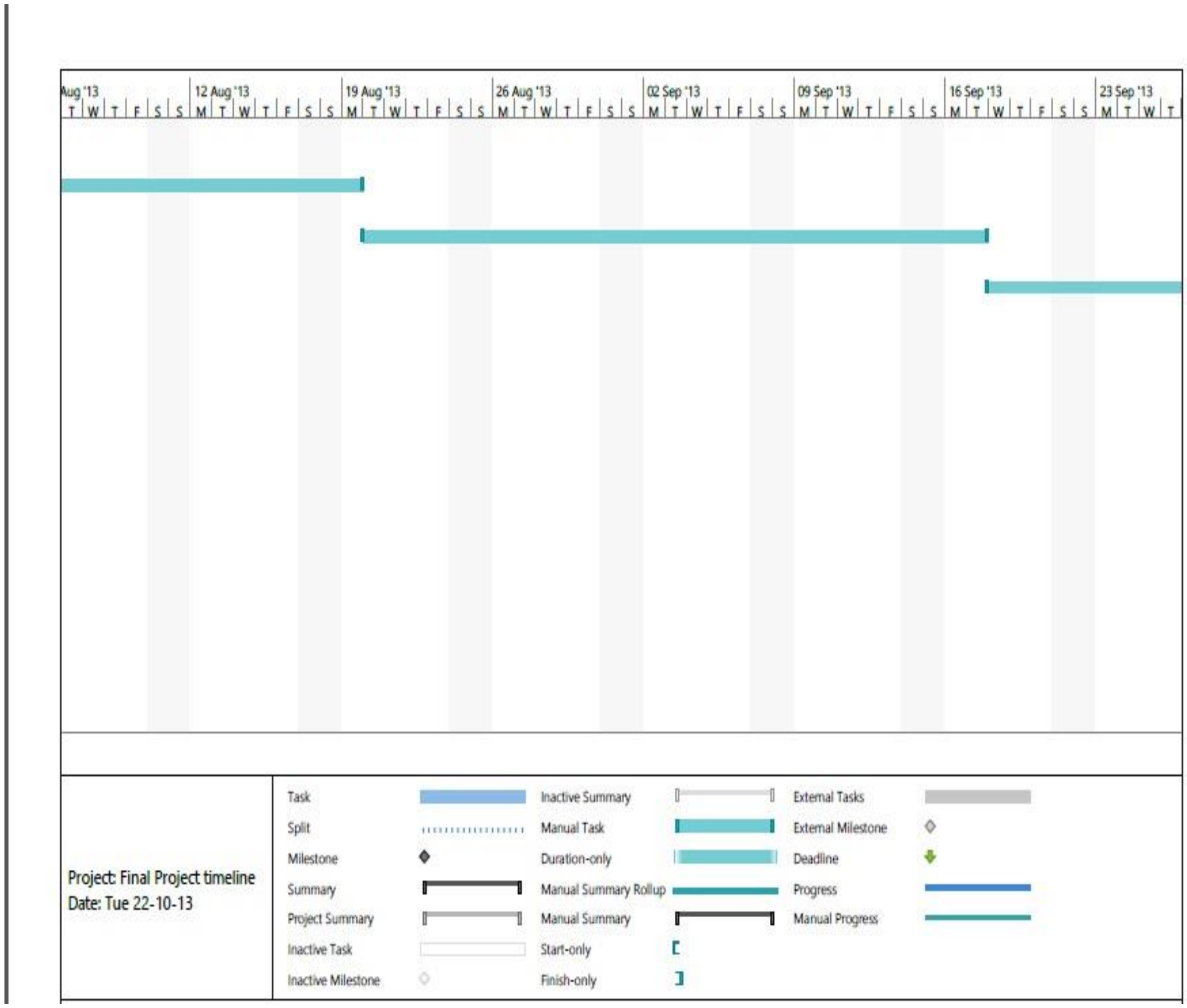

Aug '13
12 Aug '13
19 Aug '13
26 Aug '13
02 Sep '13
09 Sep '13
16 Sep '13
23 Sep '13
T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T
Project: Final Project timeline
Date: Tue 22-10-13
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

Chart 3:

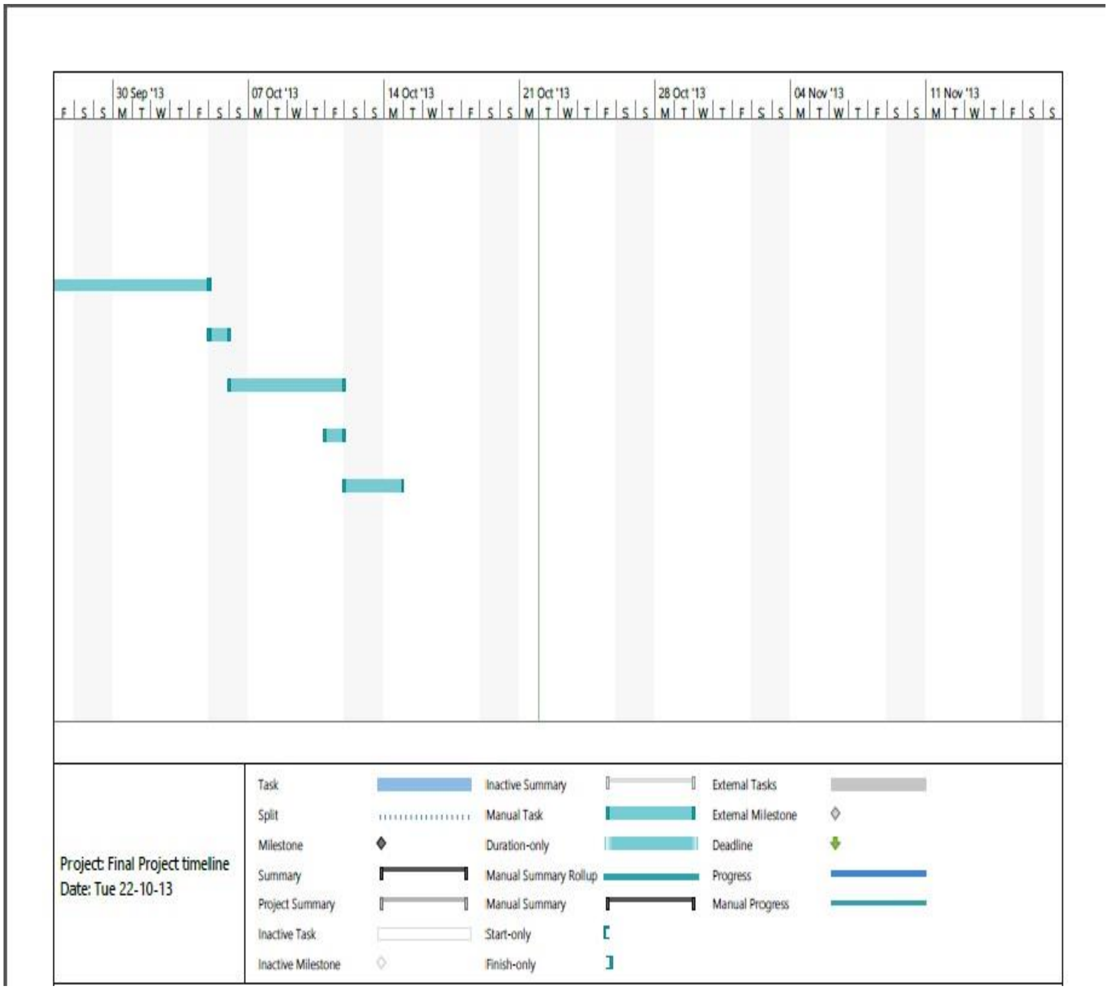

30 Sep '13
07 Oct '13
14 Oct '13
21 Oct '13
28 Oct '13
04 Nov '13
11 Nov '13
Project: Final Project timeline
Date: Tue 22-10-13
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

Chart 4 :

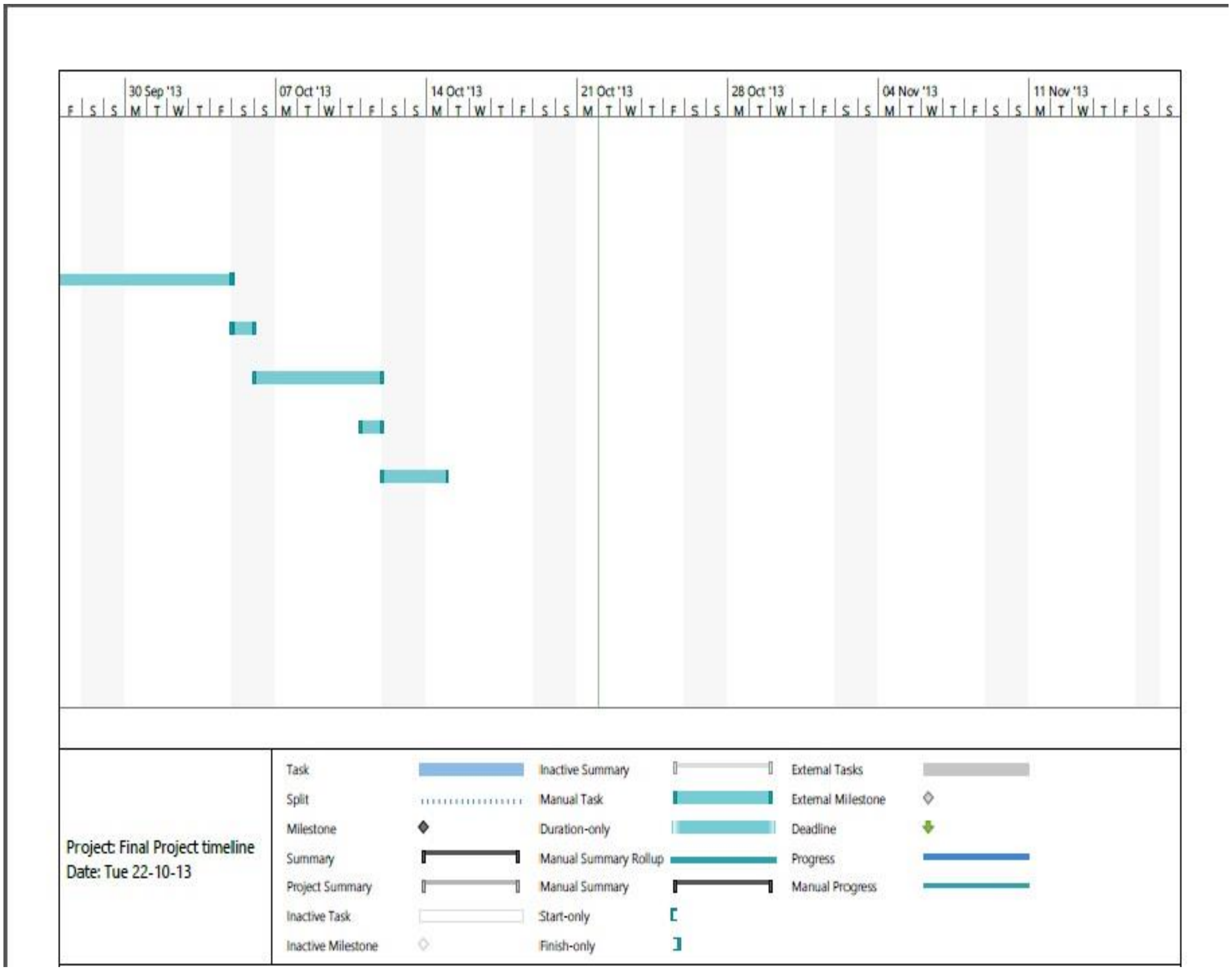
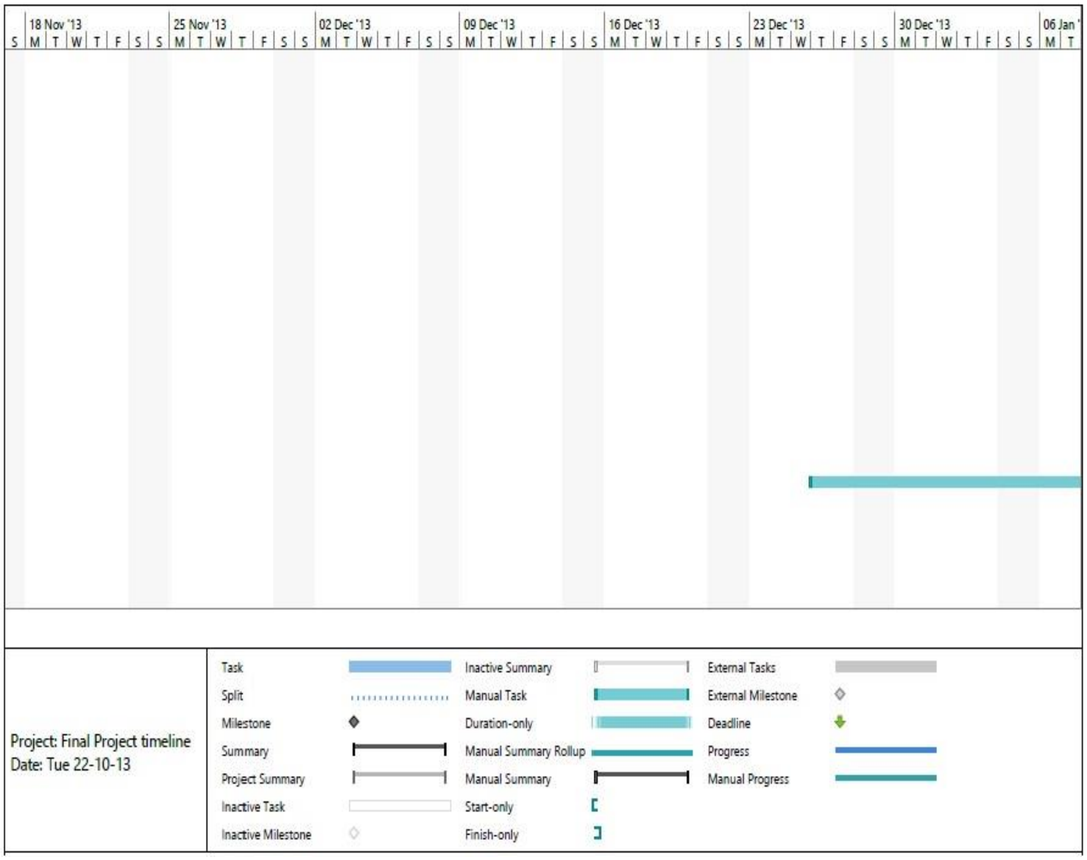
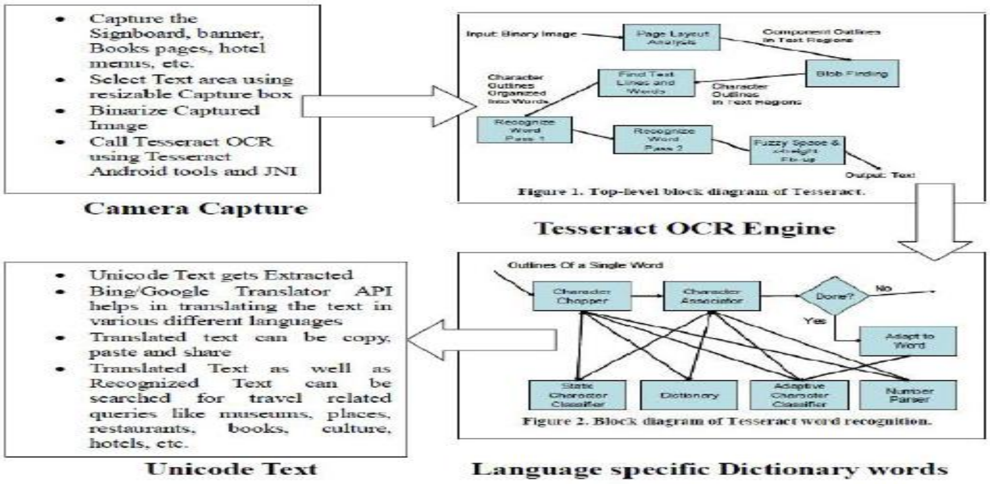

30 Sep '13
07 Oct '13
14 Oct '13
21 Oct '13
28 Oct '13
04 Nov '13
11 Nov '13
F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Project: Final Project timeline
Date: Tue 22-10-13
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

Chart 5:


18 Nov '13
25 Nov '13
02 Dec '13
09 Dec '13
16 Dec '13
23 Dec '13
30 Dec '13
06 Jan '
Project: Final Project timeline
Date: Tue 22-10-13
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

4. Implementation

Code has been removed for privacy and confidentiality concerns.



- Capture the Signboard, banner, Books pages, hotel menus, etc.
- Select Text area using resizable Capture box
- Binarize Captured Image
- Call Tesseract OCR using Tesseract Android tools and JNI

- Unicode Text gets Extracted
- Bing/Google Translator API helps in translating the text in various different languages
- Translated text can be copy, paste and share
- Translated Text as well as Recognized Text can be searched for travel related queries like museums, places, restaurants, books, culture, hotels, etc.

Camera Capture
Input: Binary Image
Page Layout Analysis
Component Outlines in Text Regions
Blob Finding
Character Outlines in Text Regions
Find Text Lines and Words
Character Outlines Organized Into Words
Recognize Word Pass 1
Recognize Word Pass 2
Fuzzy Space & x-height Fix-up
Output: Text
Figure 1. Top-level block diagram of Tesseract.
Tesseract OCR Engine
Outlines Of a Single Word
Character Chopper
Character Associator
Done?
No
Yes
Adapt to Word
Static Character Classifier
Dictionary
Adaptive Character Classifier
Number Parser
Figure 2. Block diagram of Tesseract word recognition.
Unicode Text
Language specific Dictionary words

Fig 5.1:-working of software[7]

- **Camera Capture Module 1 :**

In this module the user is allowed to resize the camera capture box by touching the box corners on the screen so as to capture the only concerned text image from signboard, banner and book pages. The camera keeps continue auto focusing the image automatically throughout the session. Once the capture button is pressed the beep sound plays and the captured image is sent to Tesseract OCR engine module.

- **Tesseract OCR Engine Module 2:**

In this module, The Binarization of Captured Image takes place, after that the text layout is analyzed, Blobs are detected and finally words and lines are detected. The words are sent to a number of passes. In these passes each word is chopped into characters and characters are checked for the need of joining the broken characters or the breaking of associated characters. Finally chopped characters are recognized with the help of inbuilt fuzzy features matched to language specific trainingdata of Unicode characters. After each pass the words are matched back and forth with the Language specific Dictionary words[4].

- **Dictionary words Matching Module 3:**

In this module each group of sequential characters is searched for a dictionary based word match, which helps in identifying the word more accurately rather than just giving a meaningless word as result. Finally the recognized text is transferred to Unicode text Post processing Module.

- **Unicode Text Post processing Module 4 :**

In this module, the recognized characters are displayed as Unicode characters and the user is allowed to translate the recognized text into his desired language available in the drop down list from settings. From there user may choose one of the two available translators from the drop down list including Bing Translator and Google translator. Moreover the user can use the advanced search feature to search the travel specific related queries like museums, books, videos, songs, culture, images, places and hotels etc. related to recognized or translated text.

Activity	**Description**	**Effort in person weeks**	**Deliverable**	Month and year
Phase 1				
P1-01	Requirement Analysis	2 weeks	Requirement Gathering	October 2013.
P1-02	Existing System Study & Literature	3 weeks	Existing System Study & Literature	October – November 2013
P1-03	Technology Selection	2 weeks	>.Net and Android	November 2013
P1-04	Modular Specifications	2 weeks	Module Description	November 2013.
P1-05	Design & Modeling	4 weeks	Analysis Report	January 2014

Activity	Description	Effort in person weeks	Deliverable	
Phase 2				
P2-01	Detailed Design	2 weeks	LLD / DLD Document	January 2013
P2-02	UI and user interactions design	Included in above	UI document	January - February 2013
P2-03	Coding & Implementation	12 weeks	Code Release	January- March 2013.
P2-04	Testing & Bug fixing	2 weeks	Test Report	January
P2-05	Performance Evaluation	4 weeks	Analysis Report	Feb 2014
P2-06	Release		System Release	March –april

Table 5.1:-Implementation table

5. Testing

Software Testing

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Beta Testing

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

5.1. Graphical User Interface

- Home screen module1:

The application is provided with two input(source) language Hindi and English

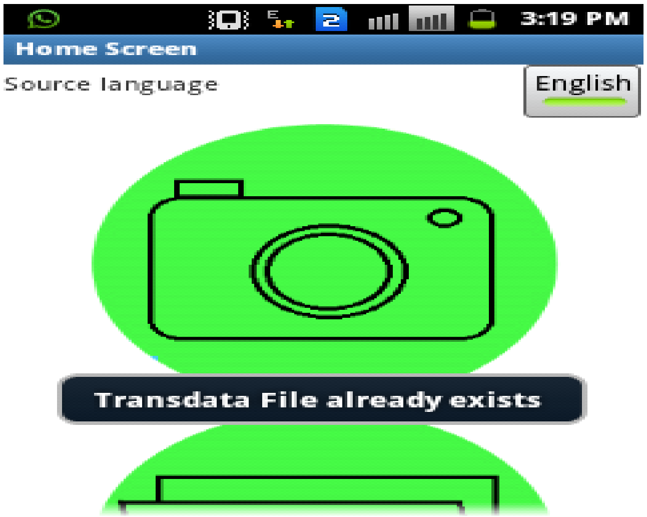


Fig 6.1.1:Home Screen module

- Camera module :

Two option capturing the image by camera or browsing the image from database

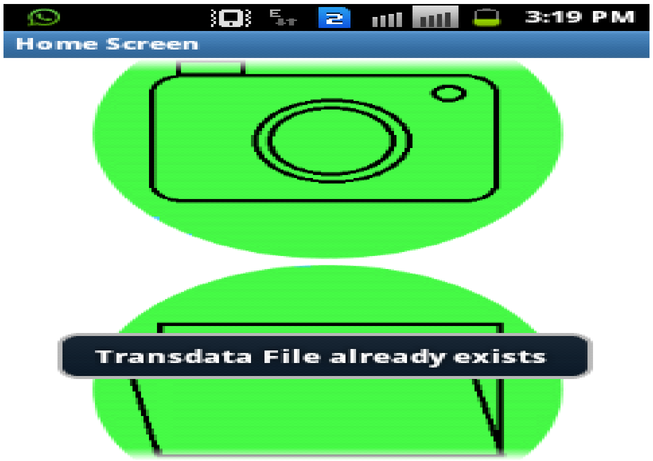
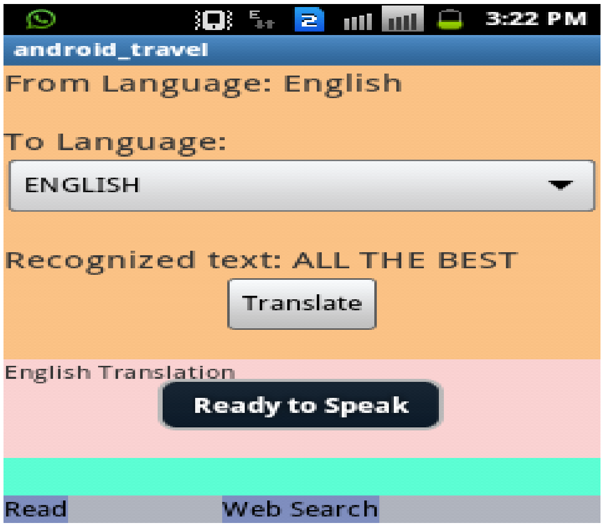


Fig 6.1.2:camera module

- Recognised text screen:

The image captured by camera or browse image go through OCR and text is recognised



From Language: English

To Language: ENGLISH

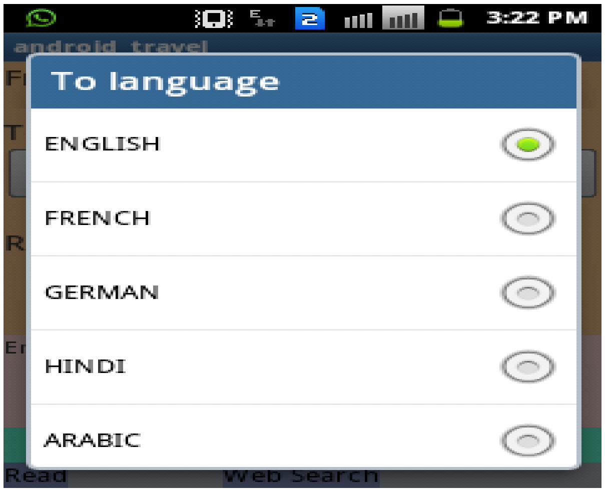
Recognized text: ALL THE BEST

English Translation

Fig 6.1.3:Recongised Text screen

- Output language screen:

Application is provided with nine language



To language

● ENGLISH

○ FRENCH

○ GERMAN

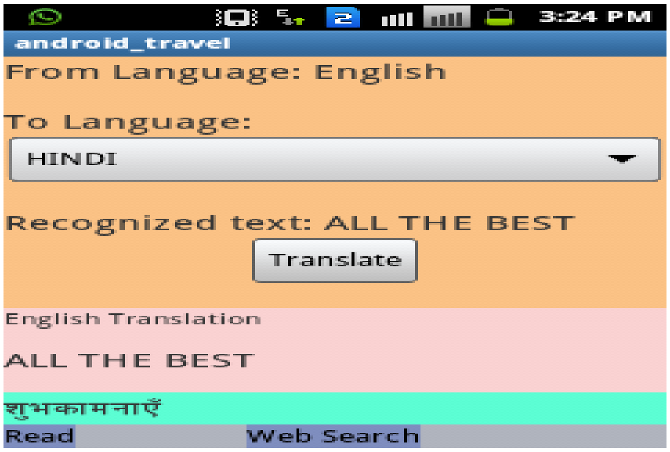
○ HINDI

○ ARABIC

Fig 6.1.4 language screen

- Translated language screen:

  The text is translated as per user choice language



android_travel

From Language: English

To Language:

HINDI

Recognized text: ALL THE BEST

English Translation

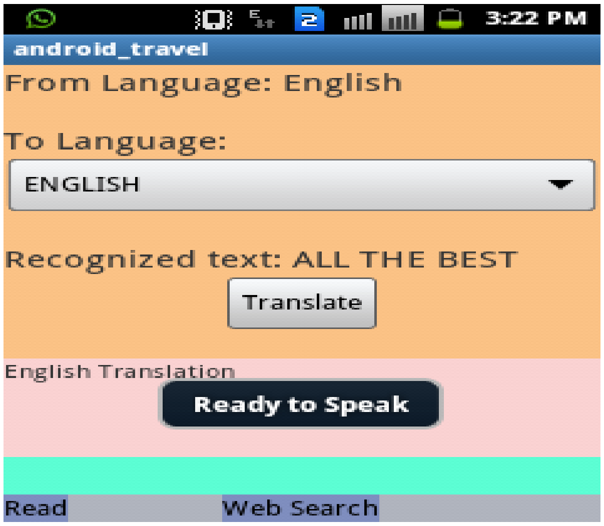
ALL THE BEST

शुभकामनाएँ

Fig 6.1.6:translated language screen

- Read and web search screen:

  The application is provided with READ and WEBSEARCH option



android_travel

From Language: English

To Language:

ENGLISH

Recognized text: ALL THE BEST

English Translation

Fig 6.1.7:Read and WebSearch module

5.2. Test Cases

For Mobile Application in Android

Test Case Id: 01.

Test Objective: To test OCR & Translation

Item No	**Test Condition**	**Operator Action**	**Input Specification**	**Output Specification(Expected Results)**	**Pass or Fail**
1	Successful output of OCR	Perform Optical Character Recognition	Start the camera click the image	Load the tesseract library. If character recognize then character Displayed on screen	Pass
2	Unsuccessful Output of OCR	Perform Optical Character Recognition	Start the camera click the image	If any problem to load tesseract library then character will not recognize	Pass
3	Successful Language translation	Perform language translation	Select the translation language	If there is sufficient Internet connectivity then the application will connect to the Bing translator	**<u>Pass</u>**

				Engine & return the output	
4	UnSuccessful Language translation	Perform language translation	Select the translation language	If there is not sufficient Internet connectivity then the application will not perform the translation & no output is generated	**Pass**

Table 6.2.1 :- Test Cases for ocr

Test Case Id: 02

Test Objective: To test text to speech

Item No	**Test Condition**	**Operator Action**	**Input Specification**	**Output Specification(Expected Results)**	**Pass or Fail**
1	Successful output of tts	Perform read operation	Click on read button	Load library. If character recognize then there would be speech heard	Pass
2	Unsuccessful Output of tts	Perform read operation	Click on read button	If any problem library then speech would not heard	Pass

Table 6.2.2 :- Test Cases for text to speech

7.Conclusion and future scope

Conclusion :

This Application provides fast, robust and extremely highQuality performance because of having improved Auto focusbehavior, continuous dynamic preview, improved noisetolerance feature and no remote computing overhead.This application already has features like text translation ro required language and text to speech featured is also included.Scalability is the basic characteristic of this application as later other features can also be included for improvement in the application.In future generation of technology this application can be made very useful as people will be travelling across borders for various purpose.

Future scope:

Image data fetcher an android application is very useful in todays era of technology.With ease of use and beneficial to all users our application can also be extended in future by adding features like navigation.It can also be improvised and made more attractive and useful by implementing speech to text.

Appendix A:User manual

Procedure :

- Image data fetcher is android application
- First install from playstore the android application
- The application is useful for language translator.

1. Select the input language from the toggle provided.
    a. Input language provided are English and Hindi.
    b. After selecting language again they are two option provided
    - Capture the image
    - Browse the image
2. Capture the image or browse the image already existed.
3. Click on yes button for confirmation of the image.
4. Image text will be recognised by the ocr.
5. That text will be sent to the translator used.
6. select the output language from the options provided.
7. Get the derired translated output.
8. you can also use the read option for hearing the translated text.
9. translated text can also be google searched for getting more information on the text for better understanding.

Appendix B:Classes and Libraries

Libraries used in Image Data fetcher

- **Android support v4 library**

The Android Support Library package is a set of code libraries that provide backward-compatible versions of Android framework APIs as well as features that are only available through the library APIs. Each Support Library is backward-compatible to a specific Android API level. This design means that your applications can use the libraries' features and still be compatible with devices running old android flavors.

Including the Support Libraries in your Android project is considered a best practice for application developers, depending on the range of platform versions your app is targeting and the APIs that it uses. Using the features the libraries provide can help you improve the look of your application, increase performance and broaden the reach of your application to more users.

- **Microsoft-translator-java-api-0.6.1-jar-with-dependencies.jar**

Download the "microsoft-translator-java-api-0.6.1-jar-with-dependenciesjar" file provided in "library.rar" above.

Include this jar as a library in the Eclipse project for your language translator.

Classes used in project :

- Camera.size**:** Sets the dimensions for pictures.
- Toast :A toast is a view containing a quick little message for the user. The toast class helps you create and show those.
- Drawables : Drawable is stored as individual files in one of the *res/drawable* folders. Typically you would store bitmaps for different resolutions in the *-mdpi*, *-hdpi*, *-xhdpi*, *-xxhdpi* subfolders of *res/drawable*. The

ADT project creation wizard creates these folders by default. If these bitmaps are provided in different folder, the Android system selects the correct one automatically based on the device configuration

- Style: A style is a collection of properties that specify the look and format for a View or window. A style can specify properties such as height, padding, font color, font size, background color, and much more. A style is defined in an XML resource that is separate from the XML that specifies the layout..
- Spinner: Spinners provide a quick way to select one value from a set. In the default state, a spinner shows its currently selected value. Touching the spinner displays a dropdown menu with all other available values, from which the user can select a new one.

Appendix C:Input and Output for Test Cases

- Test cases for OCR
  - Input < Start the camera click the image>
  - Output <if character is recongised by ocr proper text is given as output>
  - Input <start the camera click the image>
  - Ouput <if there is any obstacles in capturing the image or if there image is not clear then text will not be recongised properly>

- Test case for language translation
  - Input <select the language desired >
  - Output <if there is proper connectivity ,the output for translation in other language is provided properly>
  - Input <select the language desired>
  - Output <if there no data connectivity or language selected is same as input language the language translation is not provided properly>

- Test cases for Text to speech
  - Input <click on read button>
  - Output <if there is noise in envoirnment then it would not be heard properly>

- Test cases for websearch
  - Input <click on websearch option>
  - Output<if proper data connectivity output will be sucessfull>

REFERENCE

1. Download the Android SDK. Available at: http://developer.android.com/sdk/
2. Android Developers Homepage. Available at: http://developer.android.com/index.html
3. Eclipse IDE for JAVA Developers. Available at: http://eclipse.org/downloads/packages/eclipse-ide-java-developers/galileosr2
4. Bansal, V. and Sinha, R.M.K. “A Complete OCR for Printed Hindi Text in Devanagari Script”, Sixth International Conference on Document Analysis and Recognition, IEEE Publication, Seatle USA, 2001, Page(s):800-804.
5. Microsoft Translator Java API. Available at: http://code.google.com/p/microsoft-translator-java-api
6. Bansal, V. and Sinha, R.M.K. “A Complete OCR for Printed Hindi Text in Devanagar i Script”, Sixth
7. International Conference on Document Analysis andRecognition, IEEE Publication, Seatle USA, 2001,Page(s):800-804.

8. Saba, T., Sulong, G. and Rehman, A. “A Survey on Methods and Strategies on Touched Characters
9. Segmentation”, International Journal of Research andReviews in Computer Science (IJRRCS)
10. Vol. 1, No. 2,June 2010.

11. Pal, U., Chaudhuri, B. B. "Indian Script Characterrecognition: A survey", Pattern Recognition,
12. vol. 37, pp.1887-1899, 2004

13. Jindal, M.K., Sharma, R.K., lehal, G.S. “A Study of Different Kinds of Degradation in Printed Gurmukhi Script”, Proceedings of the International Conference on Computing: Theory and
14. Applications (ICCTA'07), 2007.

15. Google code : http://googlecode.blogspot.com/2006/08/ announcing-tesseract-ocr.html (last accessed 8 January,2012)