**Abstract**

In this digital world, where people don’t want to carry tangible objects which could be digital like books , documents, bills, photos, etc. Actually it is difficult to carry or maintain every hardcopies every time in a proper way.

Therefore, people are more interested in converting all their essential documents or personal data into softcopies or in other words, it could be digital.

While we're rapidly headed toward a paperless society, people still love their business cards and enjoy the ritual of exchanging them with their clients and other contacts. Still, it's important to make sure you have all that information where your actually going to use it—like in your smartphone or e-mail account.

This report presents a CardTech using Optical Character Recognition (OCR) system for camera captured image/graphics embedded textual documents for handheld devices. At first, text regions are extracted and skew corrected. Then, these regions are binarized and segmented into lines and characters. Characters are passed into the recognition module.

**Chapter 1**

**Introduction**

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in “storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process”. One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as images. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called document processing. Sometimes in this document processing we need to process the information that is related to languages other than the English in the world. For this document processing we need a software system called character recognition system.

**CardTech** is an automated tool that allows us to read any card by taking its photo thereby extracting the card bearer’s credentials like Name, Phone No, email or Address, etc. and saving it to the users Contact’s list or to a manageable database(here internal or external phone storage) instantly. It uses Optical Character Recognition technique which auto detects and manages text from an image. Thus we can directly make phone calls and send messages directly by storing the card’s image. Not only this, we have an option of using the recognized text to create an e-document (either word or text file). Thus it also provides us with the ability to convert a given image of a printed document into digital document thus saving considerable typing time. CardTech can be conveniently used through a mobile application.

**Objective**

The objective is to make a user friendly android application that helps in maintaining the business card by applying Optical Character Recognition on card and saving the information extracted from them on your smartphone.

* The main objective is let user save the business card information in smartphone as vcf file format i.e. in the form of contact or in any editable text document.
* The user can also save the image of business card.
* The user can share the business card information with their friends by Bluetooth or any mobile messaging application like WhatsApp by sharing vcf.

**Chapter 2**

**Literature Review**

### Optical Character Recognition (Software Based)

Optical Character Recognition (OCR) is a computer science term for the translation of printed characters on physical medium to a form of characters that a computer can handle. Almost all business card scanning devices use optical scanners and OCR software. OCR dates back decades and works well in applications where the text is of consistent font type, size, and structure.

OCR’s historical advantages were speed of processing and price of transcription, as well as the relative ease of dropping in support for different languages. The major disadvantage, however, is poor quality of transcription. Business cards are as much a work of art as they are a medium for exchanging contact data. The creative designs of many business cards lend poorly to OCR transcription, which often misses or incorrectly transcribes the text.

### Human Transcription

Traditionally, human transcription was considerably more expensive than OCR transcription. We built our app in such a way that the price of transcription is now more competitive with OCR. The end result is an app with most of the advantages of both OCR and human transcription.

Until a few decades ago, research in the field of Optical Character Recognition (OCR) was limited to document images acquired with flatbed desktop scanners. The usability of such systems is limited as they are not portable because of large size of the scanners and the need of a computing system. Moreover, the shot speed of a scanner is slower than that of a digital camera. Recently, with the advancement of processing speed and internal memory of hand-held mobile devices such as high-end cell-phones, Personal Digital Assistants (PDA), smart phones, iPhones, iPods, etc. having built-in digital cameras, a new trend of research has emerged into picture. Researchers have dared to think of running OCR applications on such devices for having real time results. An automatic Business Card Reader (BCR), meant for automatic population of relevant contact information from a business card also known as *carte de visite* or visiting card into the contact book of the devices, is an example of such applications.

However, computing under handheld devices involves a number of challenges. Because of the non-contact nature of digital cameras attached to handheld devices, acquired images very often suffer from skew and perspective distortion. In addition to that, manual involvement in the capturing process, uneven and insufficient illumination, and unavailability of sophisticated focusing system yield poor quality images. The processing speed and memory size of handheld devices are not yet sufficient enough so as to run desktop based OCR algorithms that are computationally expensive and require high amount of memory.

Under the current work, a character recognition system is presented for recognizing English characters extracted from camera captured image/graphics embedded text documents such as business card images.

There are many different apps in market for Card Reader like CamCard,Office Lens, Abbyy Card Reader, etc. These have been developed with an accuracy of around 70%. These apps take images from gallery or camera, prepocessed them and apply OCR on image and then segregate information and store in the app database or has a option to store in contacts. It has been found often that data extracted is not clear despite having high quality image.

**Chapter 3**

**Problem Statement**

The problem is for the software system to recognize characters when information is scanned through paper documents especially cards. Whenever we scan the documents through the scanner, the documents are stored as images such as .jpeg, .gif, etc. in the system. These images cannot be read by the user. To use this information it is very difficult to read the individual contents and searching the contents from these documents line-by-line and word-by-word.

CardTech enables the user to take a picture of a card and update this information in the user’s phone contacts list. The user may later on readily access the information in the form of his personal contact. He can also take and store a snap shot of the paper document and create a digital document instantly eliminating the need to type word-by-word and later on using it.

**Chapter 4**

**Requirements**

**Standard Development Tools**

* The system shall be coded and tested using Android Development Tools (ADT) a plugin for Eclipse IDE.
* Most suitable versions of Eclipse to be used are 4.2(Juno) or higher and 3.7(Indigo).
* NDK plugins allowing Eclipse to compiler native code such as C/C++ in android development.
* A fork of Tesseract Tools for Android which is a set of Android API’s and builds files for the Tesseract OCR and Leptonica image processing libraries must be installed as libraries.
* The OCR natively has highest ratio of recognition with images of 300 dpi of resolution and converted to linear (1 bit colour).
* A v3.02 trained data file for a language is needed.
* The project is set up to build on Android SDK Tools r22.3+ and Android NDK r10+. The build works on Linux, Mac OS X, and Windows 7/8.
* Response time for Optical Character Recognition should not exceed 5 seconds and can vary from one image to another.
* The Android Virtual Device must have a minimum 25 Mb of external storage for successfully building the application.
* The Alpha release has its compatibility unto Android 4.4(Kitkat) Version.
* A jdk of version 6 or higher needs to be preinstalled.

**Chapter 5**

**System Functionallity**

Our software system CardTech can be divided into three modules based on its functionality. The modules are classified as follows:-

**Document Processing Module**

This module perform certain activities such as scanning documents, storing documents, storing them as images, recognizing characters in images to transfer

them into word format. During the recognition process, this module uses the OCR (Optical Character Recognition) methodology. The control of this module is directly in the hands of the user. The module supports the following services:-

* Scanning printed documents as snapshots through camera or images through phone’s internal or external storage.
* Processing the image-based documents.
* Converting these image-based documents into e-documents (also called structured documents).
* Recognizing the characters in documents through tesseract (Tess-Two) library.

**Document Editing Module**

Once the scanned document is recognized, they are stored inside the system database (preferably internal storage of the phone). Thus the user has the information in editable form. The control of this module is directly in the hands of the user. There are two modes:-

* **Contact Updating**
* Information segmentation in which information is segmented to user details such as contact’s name, phone-no, address, email.
* Updates the recognized information to the user’s contact list.
* **General Use**
* Conversion of the document to an editable form (MS-Word, Text or specified by user).
* Any other modification of the document.
* Saving the finalized document inside the internal phone storage.

**System Training Module**

This module can be accessed by both the administrator and the end-user. Before converting the printed document in to editable document, the first and mandatory step is providing training to the system. Here training in the sense that the language and most of the fonts in the scanned document should be identified by the OCR engine. Then the scanner recognizes the characters from the image file. The user can train the module for different languages. It supports

* Training the system with pre-defined languages.

Training the system with new languages that are not present in the system and cannot be identified by the system.

**Chapter 6**

**Project Designing And Architecture**

**DATA FLOW DIAGRAM**:-

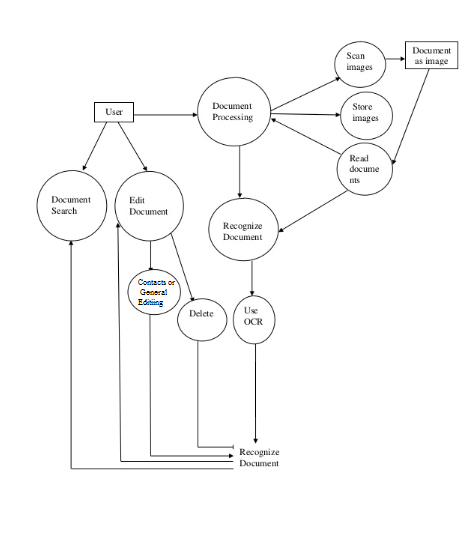


Fig1 Data Flow Diagram

**Use-Case Diagram**

1. **Use Case Name:** Document Processing
2. **Description**: The system is the only thing who participates as background process. Here the scanning of image takes place (pre-post processing). The scanned documents are read as finalized image.
3. **Actors:** System
4. **Flow of Event:**

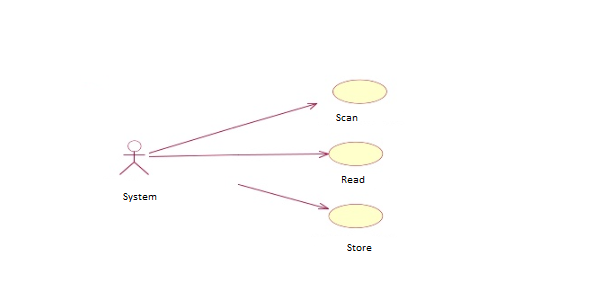
* The system scans the image which he wants to edit
* The scanned images are read as processed images
* Finally the images that are read are stored in system memory for future reference. 

Fig2 Document Processing

1. **Use Case Name:** Document Editing
2. **Description**: The system can perform the document editing. Here the scanning of image takes place (pre-post processing). After selecting the edit action editing operation is performed and finally save the information(Contact’s update or general use) that has been edited
3. **Actors:** End-user
4. **Flow of Event:**

* The user opens the image which he wants to edit
* He selects the edit action. The action consists of contact update or editing for general motive.
* After selecting the edit the action is performed.
* Finally the edited data is permanently saved.

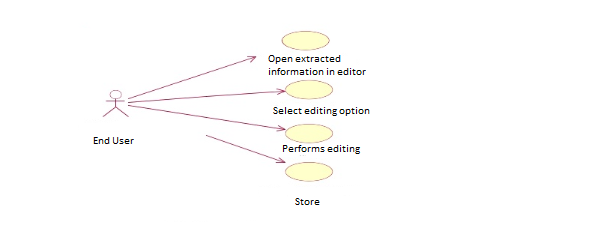
****

Fig3 Card Editing

1. **Use Case Name:** Document Recognition
2. **Description**: The administrator or user trains the system according to the given language or characters are recognized the system is trained
3. **Actors:** User or System
4. **Flow of Event:**

* The user trains the system to recognize the characters.
* After the system is trained it recognizes the data.

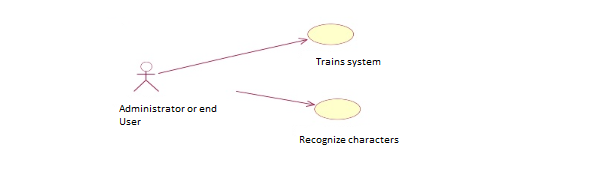


Fig4 Card Recognition

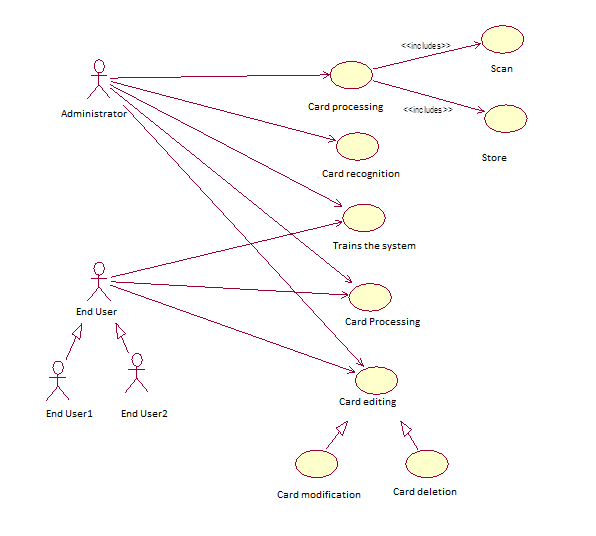


Fig5 Overall Use Case Diagram

**Chapter 7**

**Testing And Deployment**

For developing our project we are using the traditional SDLC(software development life cycle) and thus we have developed our project in several phases and ensured that errors occurring in a particular phase our taken care at that moment only and not passed on to the next phase. To ensure that this doesn’t happen , we have developed our project in modules and testing each and every module by passing all types of parameters and also we have also executed our project on worst configured systems to ensure that there is no performance lag experienced by the user and our proposed system is able to perform at its maximum efficiency.The following section describes the various phases our team has agreed upon regarding the execution of our project.

**Deliverables**

**Project plan** (Phase I)

* Project purpose and scope.
* All steps/tasks, review and revision time frames for documentation including system documentation, technology research and infrastructure preparation.
* Estimated number of hours of work required from partner staff.
* Project GANTT charts to reflect project status.

**Requirement specification document** (Phase II)

* System Description including functional requirements including system features accompanied by various requirement models to be used for detailed design (UML diagrams).
* Non-functional or technical requirements including interface requirements, performance, scalability and usability.

**Executable code module** (Phase III)

* Transform the design of a system into code in a high level language.
* Code review after successful compilation and elimination of all syntax errors.
* Code walk through and code inspection.
* Providing internal documentation in the source code and stating assumptions specific to each testing phase are documented.

**The Alpha Release** (Phase IV)

* Identifying defects through functional as well as structural testing.
* All System Integration Test Plans executed through successful completion.
* All User Acceptance Test Plans executed through successful completion.
* Unit Test Log free of errors.
* System Integration Test Log free of errors.
* User Acceptance Test Log free of errors.
* Unit Test Evaluation complete.
* System Integration Authorization is approved.

**User Manuals** (Phase V)

* Providing external documentation through end user training documents.
* Detailed training plan and user training sessions.

**Project Closeout Report** (Phase VI)

* Intended to provide an objective assessment of how the project evolved and assist future project team with learned lessons.
* It documents the favourable and unfavourable aspects of the project.
* It provides summary of the project team and the roles they played on the project.
* Quality Evaluation of criteria versus results.
* Maintaining test scripts that could ease and speed some repetitive testing effort and activities and releasing future versions of the product.

**Release to manufacturing (RTM)** (Phase VII)

* The software product ready to be delivered or provided to the customer for installation or distribution to the related hardware end user computers or machines.

Overview

The main highlight of this section is to highlight the work that has been accomplished at present in the form of executable code module. Thus this section provides a brief description required to compile and run the OCR along with Eclipse which forms the part of Document Processing module.

**1. Setting up Tess-two Library for Android**

* Download the NDK from the Android Development site.
* Then extract the NDK to a folder in your computer and extract to “”C:\Apps\android-ndk”
* Configure Eclipse to use the NDK
* Start by right clicking on your android project (named hello-neon in the below screenshots) with JNI resources, and select Properties. In the resulting dialog, choose the Builders entry in the list to the left and press the New Button.

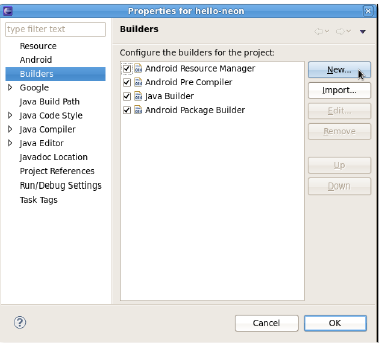
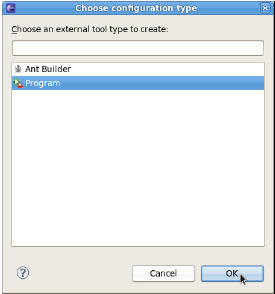
 

Fig6 Setting Tess-two Library

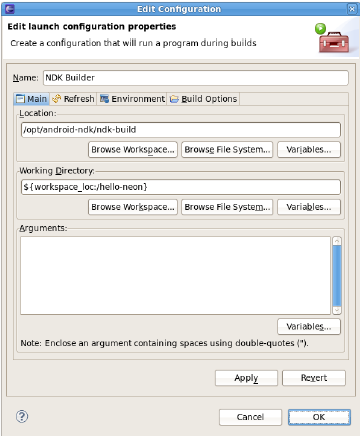
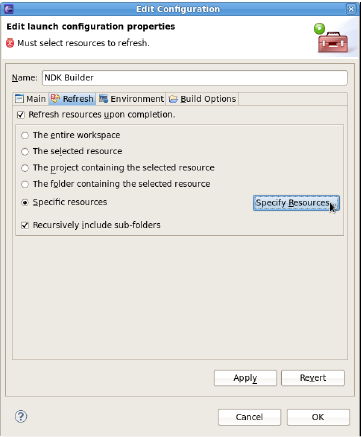
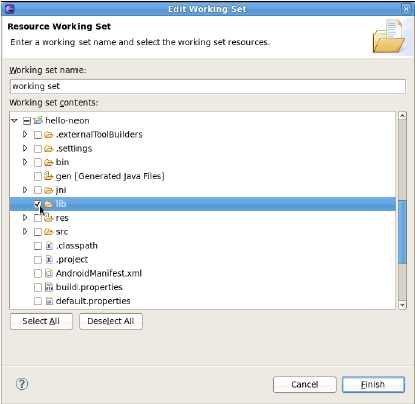
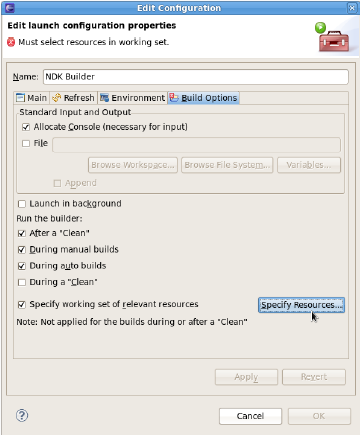
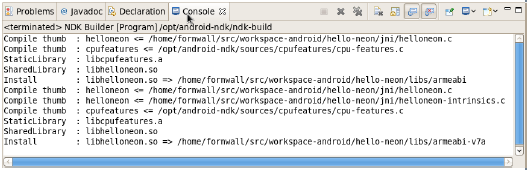
 

Fig7 Setting Tess-two Library



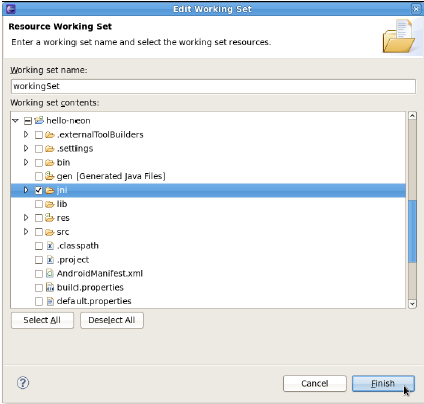


Fig8 Setting Tess-two Library

* Now continue with the refresh tab. Make sure the two checkboxes Refresh resources upon completion and recursively include sub-folders are checked.
* Choose the Specific resources radio button and press the specify Resources button.
* Since the NDK-build process will generate files in the lib folder, we want Eclipse to discover changes made there without having to refresh manually. So select the lib folder in the project (create one if necessary) and press the

Finish button.

* Now skip the Environment tab and go to the final Build Options tab. Make sure the Run the builder: During auto builds checkbox is checked.
* Since the NDK build only needs to happen when editing files in the JNI folder, check that folder and press the Finish button.
* Now finally press OK in the builder configuration dialog - the new NDK builder should now be up and running.
* Right click the project, Android Tools->Fix Project Properties. Right click->Properties->Android->Check Is Library.



Fig9 Snapshot of App

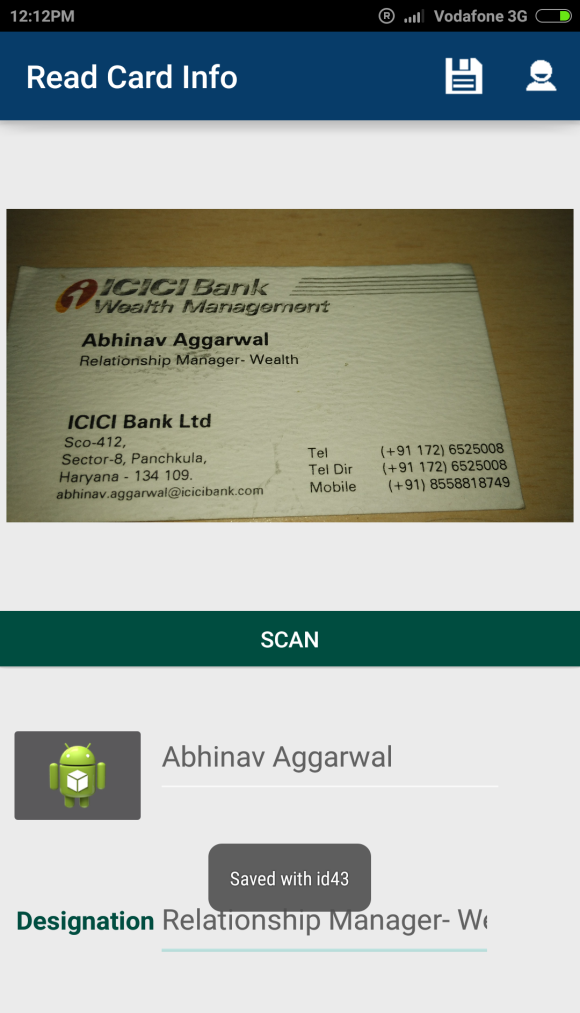


Fig10 Snapshot of App



Fig11 Snapshot of App

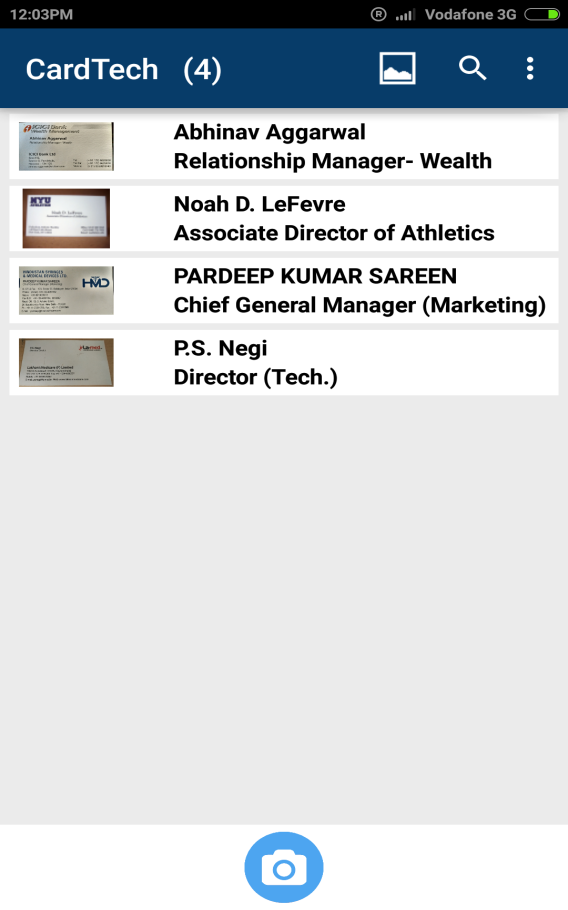


Fig12 Snapshot of App

**TimeLine**

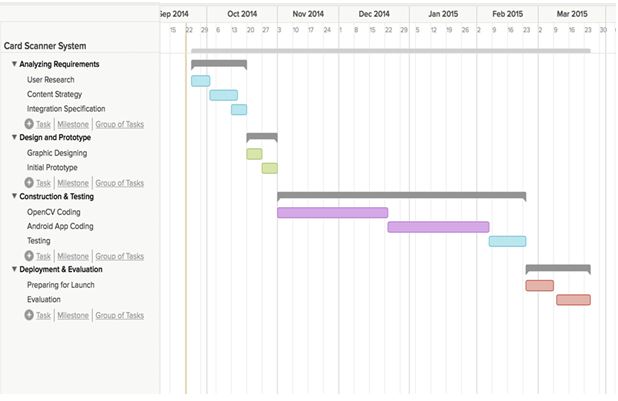


Fig13 Gantt Chart

**Chapter 8**

**Conclusion And Future Enhancements**

**Conclusion**

**CardTech** is an Android application that has been developed to carry cards with them not physically but digitally in our smartphones as in recent time cards is distributed among clients/customers/staff members. So it is difficult to keep cards with them. There are chances that card they require at any particular instant are not with them because of misplacement or that might have been lost. What happens is that they won’t able to communicate resulting in time waste and opportunity.

This application helps by storing cards in one place hence saves time required for searching. Information from cards can be saved into vcf format i.e. in contacts which can be shared among peers.

**Future Enhancements**

Today in market there are many players providing Business card reader application and each has its own advantages and disadvantages , so does our system. Following are some features which we can incorporate into our system for future needs :

* Our application accuracy is around 50% at this stage which by means in not so good. We want our app to increase its accuracy by 70-80% which is excellent.
* We are lacking in good image processing so in future we would like to improve that.
* We would be extending our app not just to card reader but also to document reader and document scanner. This means we would like to go deeper into OCR(Optical Character Recognition).
* Initially our app is available only in English language but in near future we would like to expand to different languages.
* We can expand our range by developing application for other platforms like IOS, Microsoft mobile Windows and many other growing platforms.

**Chapter 9**

**List Of figures**

|  |  |
| --- | --- |
| Fig1 | Data Flow Diagram |
| Fig2 | Card Processing |
| Fig3 | Card Editing |
| Fig4 | Card Recognition |
| Fig5 | Overall Use case Diagram |
| Fig6 | Setting up Tess two Library |
| Fig7 | Setting up Tess two Library |
| Fig8 | Setting up Tess two Library |
| Fig9 | Snapshot of App |
| Fig10 | Snapshot of App |
| Fig11 | Snapshot of App |
| Fig12 | Snapshot of App |
| Fig13 | Gantt Chart |

**Chapter 10**

**References**

* <https://code.google.com/p/tesseract-ocr/>
* <https://github.com/rmtheis/tess-two/blob/master/tess-two/src/com/googlecode/tesseract/android/TessBaseAPI.java>
* <http://gaut.am/making-an-ocr-android-app-using-tesseract/>
* <http://jagadish-android.blogspot.in/>
* <http://developer.android.com/guide/index.html>
* <http://stackoverflow.com/questions/20324155/get-filepath-and-filename-of-selected-gallery-image-in-android>
* <http://shodhganga.inflibnet.ac.in:8080/jspui/bitstream/10603/4166/10/10_chapter%202.pdf>
* <http://arxiv.org/ftp/arxiv/papers/1109/1109.3317.pdf>
* Design of an Optical Character Recognition System for Camera-based Handheld Devices by Ayatullah Faruk Mollah, Nabamita Majumder, Subhadip Basuand Mita Nasipuri. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011
* A Literature Review On Hand Written Character Recognition By Mansi Shah And Gordhan B Jethava Indian Streams Research Journal Vol -3 , ISSUE –2, March.2013 ISSN:-2230-7850.