

Android Application for Digitizing Timetable

(Project Report)



Project Supervisor:

Dr. Arif UrRahman

Submitted by

Hassan Ali

01-134121-032

Syed Mureed Abbas

01-134121-092

BS(CS)

Department of Computer Science
BAHRIA UNIVERSITY, ISLAMABAD

CERTIFICATE

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS(CS)

Dr. FazalWahab
Head of Department

Dr. Arif Ur Rahman
Supervisor

Internal Examiner

External Examiner

DEDICATION

This work is dedicated to our parents and teachers. Without their support we would not have been able to complete this project.

ACKNOWLEDGEMENTS

First of all we would like to thank Al-Mighty Allah, the most Beneficial and the most Merciful. He gave us the strength to complete this major milestone of our degree program. We are thankful to our teachers and parents who supported us both technically and morally at every stage of this project. We would also like to express our gratitude to our project supervisor Dr. Arif Ur Rahman. His technical guidance and support helped us complete this project and achieve its objectives.

ABSTRACT

This project is aimed at developing an Android application that will take image of printed timetable as an input, recognize the input and store for future use. The application will notify user before there class. Therefore, it will ensure that no one will miss his academic activity (class/quizzes/assignments) unintentionally. During the implementation we used different libraries which are OpenCv and the recognition engine Tessract. Effortfor developing a capable android application for digitizing timetable that could work on android based application that prevents users from the tedious work of manually entering.The intended application will mainly focus on recognition of digits and characters. Theapplication is planned to work on the picture which is taken from time table notice board.

Table of Contents

| | |
|--|-----------|
| DEDICATION | 2 |
| ACKNOWLEDGEMENTS | 3 |
| ABSTRACT..... | 4 |
| Chapter # 1 | 9 |
| Introduction..... | 9 |
| 1.1 Overview | 10 |
| 1.2 Project Background..... | 10 |
| 1.3 Problem Description | 11 |
| 1.4 Project Objectives | 11 |
| 1.4.1 Academic Objectives..... | 11 |
| 1.4.2 Product Objectives..... | 11 |
| 1.5 Project Scope | 11 |
| Chapter # 2 | 12 |
| Literature Review | 12 |
| 2.1 My Class Schedule: Timetable..... | 13 |
| 2.2 Timetable:..... | 14 |
| 2.3 Schedule Planner Classic..... | 15 |
| Chapter#3..... | 16 |
| Requirements Specifications | 16 |
| 3.1 Purpose of Document..... | 17 |
| 3.2 Application Overview | 17 |
| 3.4 Business Context | 18 |
| 3.5 General Description | 18 |
| 3.5.1 Product Functions | 18 |
| 3.5.2 Similar System Information..... | 18 |
| 3.5.3 User Characteristics..... | 18 |
| 3.5.4 User Problem Statement..... | 18 |
| 3.5.5 General Constraints..... | 19 |
| 3.6 Functional Requirements | 19 |
| 3.6.1 Capture image | 19 |
| 3.6.2 Save image | 19 |
| 3.6.3 Load image | 19 |

| | |
|--|-----------|
| 3.6.4 Check image data | 19 |
| 3.7 Interface Requirements | 20 |
| 3.7.1 User Interfaces | 20 |
| 3.7.2 Hardware Interfaces | 21 |
| 3.8 Performance Requirements | 21 |
| 3.9 Non-functional Requirements | 21 |
| 3.9.1 Security | 21 |
| 3.9.2 Reliability | 21 |
| 3.9.3 Maintainability | 21 |
| 3.9.3 Portability | 21 |
| 3.9.5 Extensibility | 21 |
| 3.10 Operational Methodology | 21 |
| 3.10.1 OpenCV | 22 |
| 3.10.2 Tesseract OCR: | 22 |
| 3.11 Operational Scenarios | 22 |
| 3.12 Preliminary Budget | 22 |
| Chapter # 4 | 23 |
| System Design | 23 |
| 4.1 Architecture Design | 24 |
| 4.2 Logically Significant Use-case: | 24 |
| 4.3 Use-Cases | 25 |
| 4.3.1 Use Case 1: | 25 |
| 4.3.2 Use Case 2: | 26 |
| 4.3.3 Use Case 3: | 27 |
| 4.3.4 Use Case 4: | 28 |
| 4.3.5 Use Case 5: | 29 |
| 4.4 Sequence Diagram | 30 |
| 4.4.1 Diagram 1 | 30 |
| 4.4.2 Diagram 2 | 31 |
| 4.5 Application Program Interfaces | 32 |
| 4.5.1 OpenCV | 32 |
| 4.5.2 Tesseract OCR Engine | 32 |
| 4.6 User Interface Design | 32 |

| | |
|---|-----------|
| Chapter # 5 | 34 |
| System Implementation | 34 |
| 5.1 Tools and Technology | 35 |
| 5.2 Languages | 35 |
| Chapter # 6 | 36 |
| System Testing & Evaluation | 36 |
| 6.1 Introduction | 36 |
| 6.2 Usability Testing | 37 |
| 6.3 Software Performance Testing | 37 |
| 6.4 Compatibility Testing | 37 |
| 6.6 Installation Testing | 37 |
| 6.7 Test Cases | 38 |
| 6.9 Conclusion | 40 |
| Chapter # 7 | 42 |
| Conclusions | 42 |
| 7.1 Conclusions | 43 |
| References | 44 |

Table of Figures

| | |
|--|----|
| FIGURE 1: MY CLASS SCHEDULE: TIMETABLE | 14 |
| FIGURE 2: TIMETABLE | 14 |
| FIGURE 3: SCHEDULE PLANNER CLASSIC | 15 |
| FIGURE 4: APPLICATION INTERFACE | 17 |
| FIGURE 5: A TYPICAL SESSION WITH THE APPLICATION | 25 |
| FIGURE 6: INPUT IMAGE | 25 |
| FIGURE 7: IMAGE PROCESSING | 26 |
| FIGURE 8: RECOGNIZE TEXT | 27 |
| FIGURE 9: STORE DATA | 28 |
| FIGURE 10: DISPLAY RESULT | 29 |
| FIGURE 11: SEQUENCE DIAGRAM FOR CAPTURE IMAGE & GET RESULT | 30 |
| FIGURE 12: SEQUENCE DIAGRAM FOR LOAD IMAGE & GET RESULT | 31 |
| FIGURE 13: GUI | 33 |
| FIGURE 14: TESTING OF INSTALLATION OF APPLICATION | 38 |
| FIGURE 15: TESTING OF LOADING IMAGE | 38 |
| FIGURE 16: TESTING OF CAPTURE AND LOAD IMAGE | 39 |
| FIGURE 17: TESTING OF CHARACTER RECOGNITION FROM IMAGE | 39 |
| FIGURE 18: TESTING OF EVALUATE DATA | 40 |
| FIGURE 19: TESTING OF UNINSTALLATION OF APPLICATION | 40 |

Chapter # 1

Introduction

1.1 Overview

The last two decades have witnessed a tremendous increase in digitization of paper documents making them available online. The concept of digital libraries and distance learning has also gained significant interest in the recent years. Many of the researchers find some ways to recognize text from papers and convert to any digital format. Since the manual transcription of paper documents is tedious and expensive, researchers were attracted to develop automated text and handwriting recognition systems commonly termed as Optical Character Recognition (OCR)[1, 2]. However, with the remarkable increase in the number of hand held devices equipped with a camera, a significant research and development attention has shifted to camera based documents rather than scanned images. Users can simply take a picture of the text using their camera based mobile devices and convert the image into text using a number of recognition tools and applications.

Typical OCR systems are intended to work on alphabets of a particular language. Moreover, they may also account for specific keywords used in a domain.

The proposed project is intended to develop an Android application which will recognize printed timetable through mobile camera. The application will store the course title the Room number where the lecture will be delivered, and the time of the class. The application will have a beep function to notify users a few minutes before the time of the class. Many other applications intend to manually enter the values, but in this application we just have to take a picture of the timetable and it will automatically store the values. There will also be a beep functionality which users may use for notification a few minutes before the class.

1.2 Project Background

This project is aimed at developing an Android based application to recognize time table and store its characters as class name, subject name and time. The project relies on taking an image

of printed copy of time table and applying a series of image analysis operations to segment the individual symbols.

1.3 Problem Description

This project is aimed at developing an Android application that will take image of printed timetable as an input, recognize the input and store for future use. The application will notify user before there class. Therefore, it will ensure that no one will miss his academic activity (class/quizzes/assignments) unintentionally.

1.4 Project Objectives

The main objectives of this project contains following.

1.4.1 Academic Objectives

We applied different concepts which we have learned during different course work to solve different modules in our project and improve skill by developing the project in used tools and technologies also ideas which involved in this project.

1.4.2 Product Objectives

Effort for developing a capable android application for digitizing timetable that could work on android based application that prevent users from the tedious work of manually entering.

1.5 Project Scope

The intended application will mainly focus on recognition of digits and characters. The application is planned to work on the picture which is taken from time table notice board.

Chapter # 2

Literature Review

The main objective of this chapter is to provide overview of familiar applications which recognize timetable. Few of the applications require users to manually enter timetable on the device. The following sections discuss some famous applications.

2.1 My Class Schedule: Timetable

My Class Schedules is an android application that will keep the user informed about their upcoming classes and also reminds the user about their homework. One more feature of My Class Schedule is its timetable that shows schedule for any particular day or week. It notifies the user for upcoming classes and other academic activates which were stored by user. User can interacts with application by entering the name of course and teacher name. User has to enter the time to tell the app at which time the class will takes place [3].Figure 1 shows a sample of My Class Scheduleapplication.

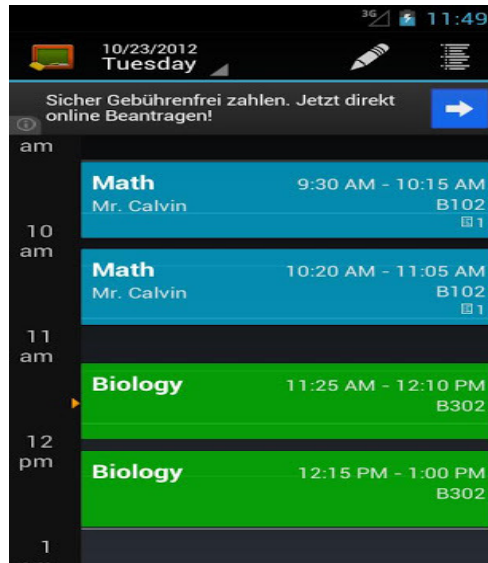


Figure 1: My Class Schedule: Timetable

2.2 Timetable:

Timetable is an android application which allows the user to enter the timetable and all other academic tasks once. Timetable will notify your class time and class room, where the class will takes place. This application has many views so user can check schedule of classes and other events like quizzes, exams and assignments [4]. Figure 2 shows a sample of Timetable application.



Figure 2: Timetable

2.3 Schedule Planner Classic

Schedule Planner Classic is an android application that allows users to make a schedule of their activities. Schedule Planner allows users to modify and edit their scheduled tasks or by batch editing multiple tasks. Users can create notes for more details of their tasks. This app has some more features like calendar view, pie view and bar graph view of app so user can plan in advance and view day by day activities in calendar view [5]. Figure 3 shows a sample of Schedule Planner Classic application.



Figure 3: Schedule Planner Classic

Chapter#3

Requirements Specifications

3.1 Purpose of Document

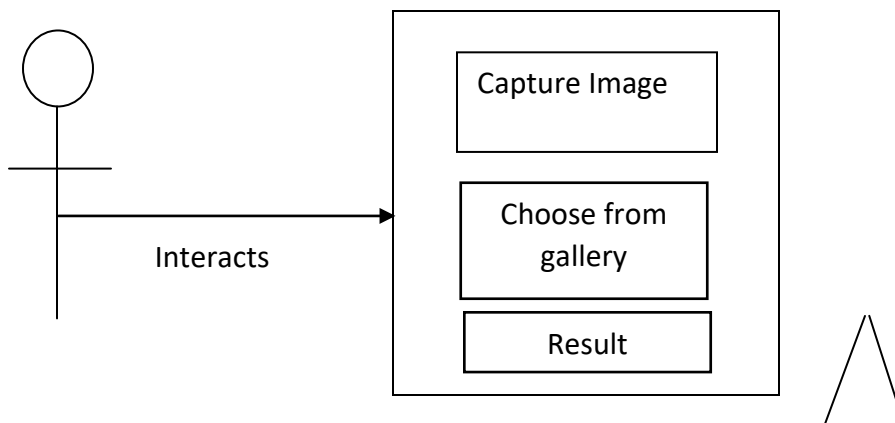
The purpose of this document is to provide full explanation of android application for digitizing timetable. It will explain the purpose, objectives and interface of the purposed system. The application will be and must work under the restrictions which apply by external stimuli. This document is written for both the stakeholder and the developers of the application and will be proposed to the evaluation team for the approval.

3.2 Application Overview

The proposed application is an android based application for digitizing timetable for the students. This application will be designed to maximize the speed and will reduce the effort by receiving a picture of timetable and storing it without doing any paper or manual work. The initial scope of the application is to store the timetable and store required characters.

3.3 System Environment

The global system environment of the proposed application is demonstrated in Figure 4. The application contains three main interactive operations including taking an image from camera or choosing from the gallery and displaying result.



User

Figure 4: Application Interface

3.4 Business Context

The preferred application will facilitates user as a proof of concept which could later be applicable into a complete application that will help and facilitate students (users).

3.5 General Description

3.5.1 Product Functions

User can interact to application with the help of user interface that will let the user to take an image of timetable. The system will then recognize the digits and symbols in timetable. Once the user will take image of timetable, then it will recognize characters and will store them for further process.

3.5.2 Similar System Information

Some of the similar systems that have already been described in chapter 2 [section 2.1- 2.3]. Many of these existing systems allow user for manually entering. In our application, we intended to work on images of printed timetables.

3.5.3 User Characteristics

There is no need for special knowledge to operate or interact with our application. Users are not supposed to do extra effort to interact with android application for digitizing timetable. This application will provide user friendly environment from which users can easily interact with application.

3.5.4 User Problem Statement

With the increase in the number of hand held devices equipped with a camera, the norm of taking notes is being replaced with taking pictures of the required information. Scenes of students taking pictures of notice boards, date sheets, time tables and white board notes are frequent and common. Although this saves a lot of time and effort, the major drawback with these camera based notes is that all the captured information is in the form of images. This makes it impossible

to edit, search or analyze the textual content of these images. Automated algorithms which convert these images into text could be handy and could result in a wide variety of interesting applications. The focus of this study is for digitizing timetable. The proposed application will recognize the tokens in timetable.

3.5.5 General Constraints

Catering with huge diversity in timetable styles of different individuals would be a difficult task. All styles captured from printed timetable may not be recognized by the system. The application will mainly focus to recognize digits, symbols and characters.

3.6 Functional Requirements

In this document we will describe all functional requirements that were gathered.

3.6.1 Capture image

User need to capture image from the camera by using option ‘capture image’ in the application.

3.6.2 Save image

In order to make use of the system, user must need to save the image on a mobile device.

3.6.3 Load image

- User need to load the image in the desired application.
- First user need to capture and save the image in order to load it.
- User has to save image.

3.6.4 Check image data

To get the accurate result, user has to take image clearly to recognize text.

3.6.5 Communication Channel

Our application will provide user the functionality to digitize timetable using android based smart phone.

I. Technical issues

the device should have a camera and should have required memory to load and execute the application.

II. Cost

The product has only development cost, the application does not have any functioning

cost.

III. **Risks**

No particular risks can be identified for the given scenario.

3.6.6 Smart Phone Selection

The smart phone selected should be an Android based device.

I. **Criticality**

The smartphone compatibility is the most basic functionality requirement that should be completed.

II. **Technical issues**

The hardware and software of the smart phone, if having some technical issues should be replaced or repaired for full function of the application.

III. **Cost**

User need to pay the cost of smart phone.

3.7 Interface Requirements

It describes how the system's graphical user interface is and how it operates with users for input and output.

3.7.1 User Interfaces

This portion shows how user will interact with product interface.

I. **GUI**

The GUI is very simple and comprises three main buttons including 'Capture', 'Select from photos' and 'Load'. The user will have to take picture of timetable or choose from the gallery.

II. **Tools Used**

The interface has been developed using the Android SDK while the backend computations and algorithms are implemented using OpenCV, Tesseract OCR engine.

3.7.2 Hardware Interfaces

The only hardware interface is the Android based device which will allow the users to interact with the application.

3.8 Performance Requirements

The required android smart phone should have minimum specifications that are needed to run the proposed application. The processor, memory and the type of android version should match with the application requirements.

3.9 Non-functional Requirements

Following are non-functional attributes required by the application are listed.

3.9.1 Security

Due to the nature of the application as such no security issue is involved. In this application there is no need for any user's personal information or data.

3.9.2 Reliability

The application should be reliable in the sense that the recognition engine should provide suitably. In other hand as the user's expectations for execution, the applications should not crash in exceptional cases.

3.9.3 Maintainability

Developer will be responsible for the maintenance if it is required.

3.9.3 Portability

This application is android based so it is portable because android based application itself portable.

3.9.5 Extensibility

Application is extensible by the developer whereas not by the users. We will enhance the functionality of our application if required.

3.10 Operational Methodology

During the implementation we used different libraries which are OpenCv and the recognition engine Tesseract.

3.10.1 OpenCV

OpenCV is a well-known computer vision and image processing library supporting a variety of standard operations and algorithms. Different versions of OpenCV are available for various platforms. The current project will be employ **opencv4** Android library. This library will mainly be helpful in document preprocessing and segmentation/extraction of printed components from the image for further processing[6][7].

3.10.2 Tesseract OCR:

Tesseract OCR engine is a trainable recognition engine that can be exploited to classify the input symbols (digits, characters etc.) into one of the learned classes. The available library is trained to recognize printed text only. The components extracted from the image will be fed to the Tesseract engine for training and subsequent recognition [8].

3.11 Operational Scenarios

This version of proposed application is capable for recognizing the printed timetable. The user could be a student or teacher. The user can take a picture of timetable. The application will recognize the characters and symbols from timetable.

3.12 Preliminary Budget

The budget is only needed in the development phase which will include

- Smart phone.
- Additional libraries if required.

Chapter # 4

System Design

Main objective of this chapter is to document and tracks the compulsory information essential to efficiently define the architecture and system design. The intended audience of this chapter includes the project manager, project team and development team. Some portions of this document such as the user interface (UI) may infrequently be shared with the client/user and other stakeholders whose input/support for the UI is needed.

4.1 Architecture Design

The application is proposed to run on the Android smart phone. Accordingly, the basic interaction of the application with the hardware devices includes the following.

- Camera for taking the input image of the required timetable.
- Memory for accessing the previously saved image as an input.
- Processor for executing the algorithms and carrying out the required results.

The interaction between different modules within the application includes the OpenCV library which processes the image and pass the output to the Tesseract engine.

4.2 Logically Significant Use-case:

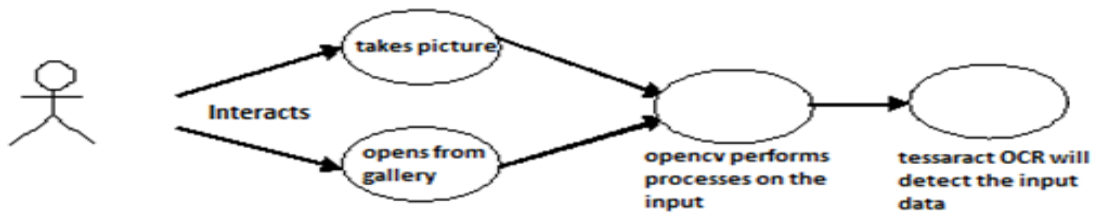


Figure 5:A typical session with the application

4.3 Use-Cases

4.3.1 Use Case 1:

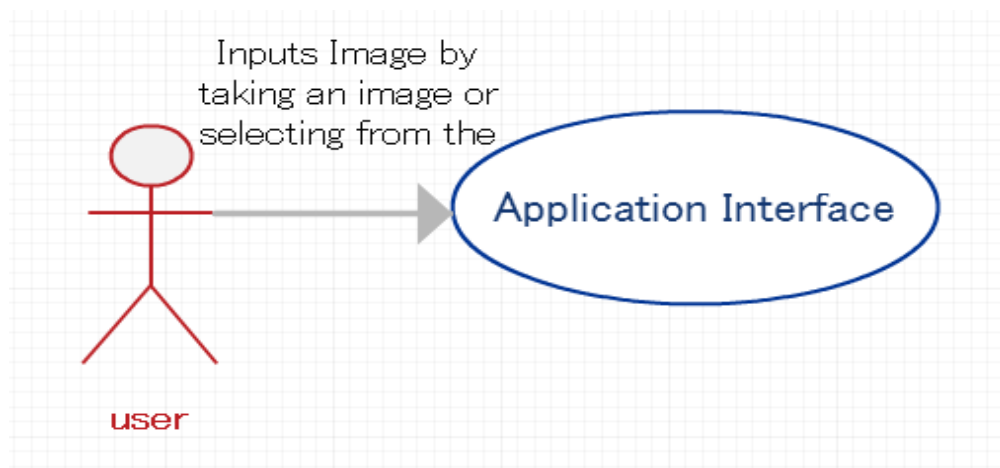


Figure 6:Input Image

| | |
|----------------------|----------------|
| Use Case Id | UC 1 |
| Title | Input Image |
| Description | Load the image |
| Primary Actor | Any User. |

| | |
|-----------------------|--|
| Pre-Condition | User have to the open application first and then mobile gallery. |
| Post-Condition | Image as input is given to the application. |

4.3.2 Use Case 2:

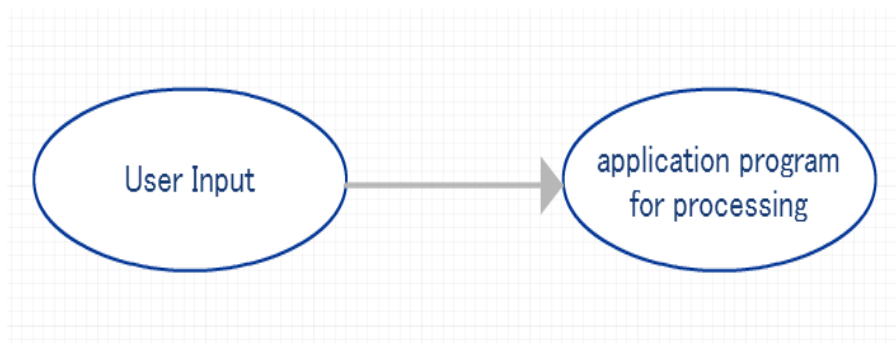


Figure 7:Image processing

| | |
|----------------------|--|
| Use Case Id | UC 2 |
| Title | Image processing |
| Description | User have to load the image for processing |
| Primary Actor | Any User |
| Pre-Condition | Load Image. |

| | |
|-----------------------|---|
| Post-Condition | User can verify the data from image loaded. |
|-----------------------|---|

4.3.3 Use Case 3:

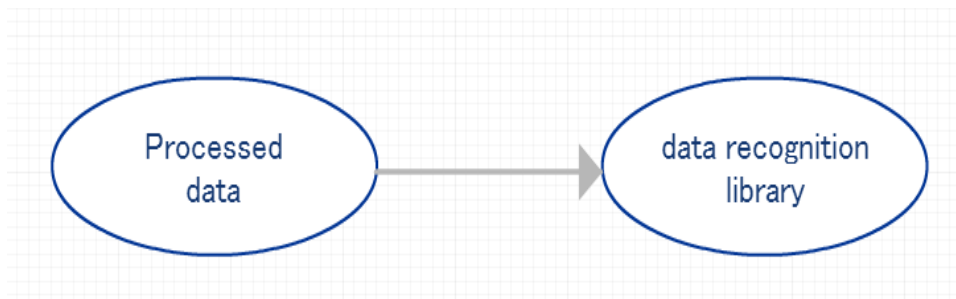


Figure 8:Recognize text

| | |
|----------------------|--|
| Use Case Id | UC 3 |
| Title | Recognized text |
| Description | User will obtain the recognize text. |
| Primary Actor | Any User |
| Pre-Condition | Image must go through the processing step. |

| | |
|-----------------------|--|
| Post-Condition | User can 'extract characters' from text. |
|-----------------------|--|

4.3.4 Use Case 4:



Figure 9:Store data

| | |
|----------------------|--|
| Use Case Id | UC 4 |
| Title | Store data |
| Description | User will check the text and go with the option 'submit' |
| Primary Actor | Any User |
| Pre-Condition | Text must be recognized and it has to be in a valid form |

| | |
|-----------------------|--|
| Post-Condition | User can go for option 'allow or disallow' |
|-----------------------|--|

4.3.5 Use Case 5:

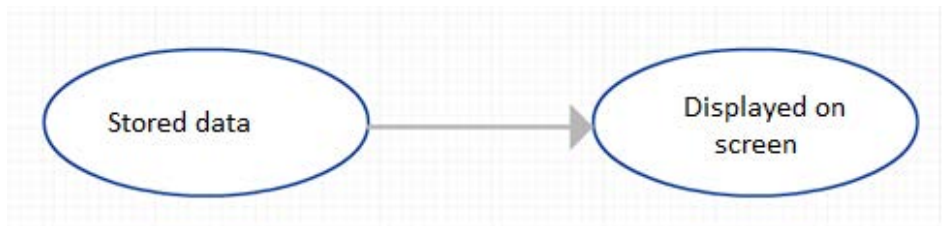


Figure 10: Display Result

| | |
|--------------------|---|
| Use Case Id | UC 5 |
| Title | Display the result |
| Description | Result of the given image will be displayed |

| | |
|-----------------------|-----------------------------------|
| Primary Actor | Any User |
| Pre-Condition | Must be a valid form of timetable |
| Post-Condition | Result displayed to user |

4.4 Sequence Diagram

4.4.1 Diagram 1

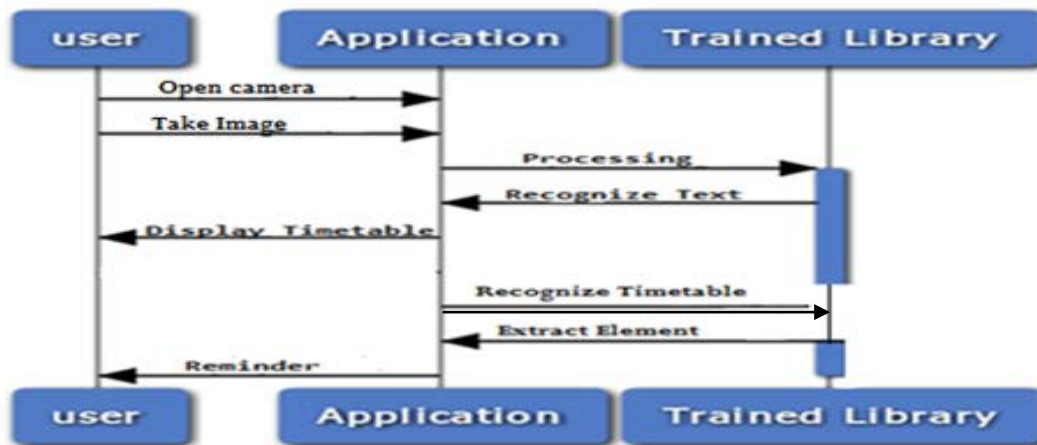


Figure 11:Sequence Diagram for Capture Image & Get Result

4.4.2 Diagram 2

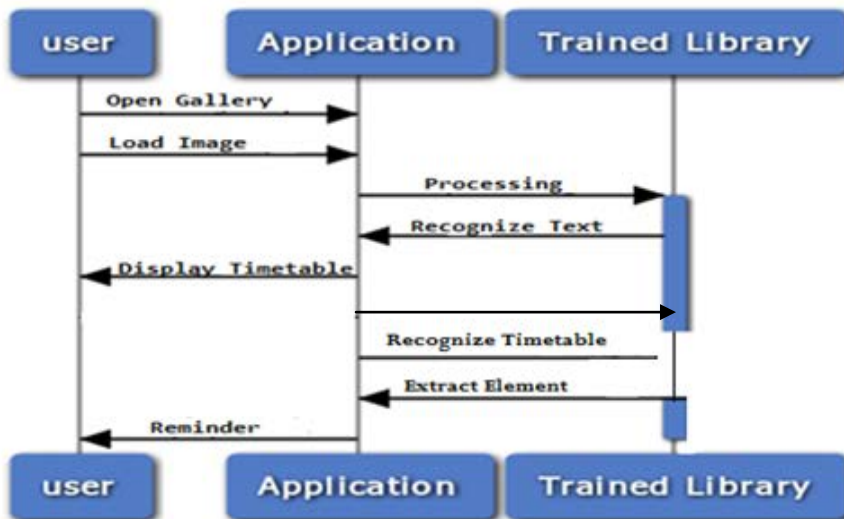


Figure 12: Sequence Diagram for Load Image & Get Result

4.5 Application Program Interfaces

4.5.1 OpenCV

As discussed earlier, the OpenCV library provides implementation of a number of standard image processing and image analysis operations. Among the various versions of OpenCV, this project will employ 'opencv4android' library. Operations like image Binarization and extraction of connected components are implemented using the aforementioned library.

4.5.2 Tesseract OCR Engine

The Tesseract OCR engine will be investigated to perform recognition of digits and other symbols. Images of individual symbols and digits will be used to train the recognition engine which will later classify the query symbols into one of the learned classes [9].

4.6 User Interface Design

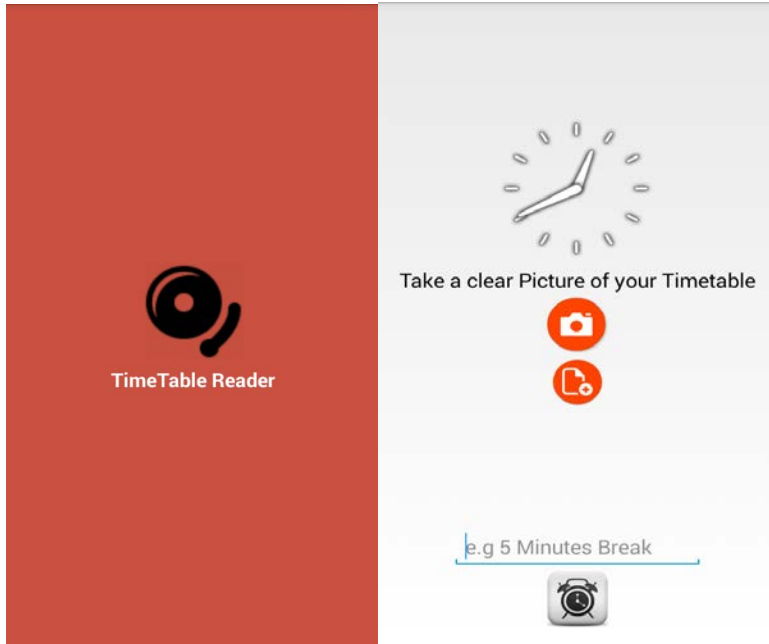


Figure 13: GUI

Figure 14: GUI

Chapter # 5

System Implementation

This chapter provides the basic image of the algorithmic description of the implemented methodology, the tools used and the detail of applied operations on the system which is used during the development process.

5.1 Tools and Technology

Followings are the tools which are used to develop this application.

I. Eclipse

Eclipse is a commonly used IDE i.e. integrated development environment. It is an open source software which is used to create different applications. Eclipse supports many programming languages including Java. Developers also have the flexibility to add required plugins in their implementations

II. OpenCV

OpenCV is the library of different image processing operations.

III. Tesseract

It is an open source OCR (Optical Character Recognition) implementation mainly developed for recognition of printed text. It can read many types of image formats and fonts and converts these images into text. It is the most reliable open source Optical Character Recognition available to the best of author's knowledge. Tesseract does not offer any graphical interface and is executed from the command line interface.

5.2 Languages

The project has been developed in Java a popular and widely used object oriented programming language. It supports integration with a vast number of libraries. It also provides many predefined operations for the convenience of the developers.

Chapter # 6

System Testing&Evaluation

6.1 Introduction

Software testing is a process that guaranteed that every phase of product is correctly working and it is providing results as required by user. Application's each module is tested separately and later on, full application testing is also done. The primary goal of software testing is to check whether the proposed application fulfill mandatory quality criteria or not. Testing of software is targeted at determining, either the proposed product is providing result as required by user or not.

6.2 Usability Testing

Software usability testing is targeted at determining the easiness with which system may be used. Software usability testing is accomplished by taking a sample of primary users and giving them opportunity to use proposed application. So later on, we can get feedback from user to recognize usability issues and try to fix them. It shows how user experienced our application.

6.3 Software Performance Testing

Performance testing of software is a process in which the performance, reliability and efficiency of product is measured to check whether it meets the user requirements or not. Software performance testing is conducted to determine the performance goals of proposed application in terms of effectiveness, reliability and efficiency.

6.4 Compatibility Testing

Software compatibility testing is a testing process in which application is tested through different stages those are supports application. As our proposed application is an Android application, so it is compatible with different Android devices.

6.5 Load Testing

The load testing of software is takes place to check the product behavior by providing a detailed predictable load. User can select only one image at a given point in time, application runs efficiently, no matter how much data is there in image. Load testing of our product is done by providing an image of timetable which contains the characters, digits and symbols.

6.6 Installation Testing

Installation testing is takes place on user device. Install proposed application on user device which has application's compatible android version. We have tested our application on different devices those runs different versions of android. Our application installed correctly on android device and performed functionality successfully[10].

6.7 Test Cases

Test Case # 1: Installation of Application

| | |
|----------------------|---|
| Test Case ID | TC#1 |
| Unit of Test | Verifying the application is installing. |
| Steps to be Executed | Open the application apk from where the file was saved. |
| Expected Result | Application should be installed. |
| Actual Result | Application installed successfully. |
| Status | Success |

Figure 15: Testing of Installation of Application

Test Case # 2: Load Image

| | |
|----------------------|--|
| Test Case ID | TC#2 |
| Unit of Test | Test to check if the image is loading successfully. |
| Steps to be Executed | Click on the load image from the application screen and select the image from the phone storage or from the external phone memory. |
| Expected Result | Image should be loaded. |
| Actual Result | Image loaded successfully. |
| Status | Success |

Figure 16: Testing of Loading Image

Test Case # 3: Capture & Load Image

| | |
|----------------------|---|
| Test Case ID | TC#03 |
| Unit of Test | Test to verify application can capture the image. |
| Steps to be Executed | Click on the capture image button. Device camera will be opened. Capture the image and press save. |
| Expected Result | Image should be captured and must be loaded if user press save. |
| Actual Result | Image capture and loaded successfully. |
| Status | Success |

Figure 17: Testing of Capture and Load Image

Test Case # 4: Character Recognition

| | |
|----------------------|---|
| Test Case ID | TC#04 |
| Unit of Test | Test to verify application recognize text |
| Steps to be Executed | Load the image. |
| Expected Result | The system must be able to recognize the symbols. |
| Actual Result | Symbols recognized with acceptable recognition rates. |
| Status | Success |

Figure 18: Testing of Character Recognition from Image

Test Case # 5: Evaluate the data given in image

| | |
|-----------------------|---|
| Test Case ID | TC#5 |
| Test case description | Test to verify if the application correctly perceives the data from image. |
| Steps to be Executed | After perceiving the data, select the solve option. |
| Expected Result | Display the class time and subject name and location where the class will take place. |
| Actual Result | Class time, subject name and location displayed to the user. |
| Status | Success |

Figure 19: Testing of Evaluate Data

Test Case # 6: Uninstallation of application

| | |
|----------------------|---|
| Test Case ID | TC#6 |
| Unit of Test | Test to verify if the application uninstalls from the phone. |
| Steps to be Executed | Go to the Settings->Application manager and select the application. |
| Expected Result | Application must be uninstalled. |
| Actual Result | Application is uninstalled successfully. |
| Status | Success |

Figure 20: Testing of Uninstallation of Application

6.9 Conclusion

In this chapter we applied different test cases on different modules to verify the working condition of our application. This application is also tested on different mobiles (android based), it work good on different mobiles. We also obtain quantitative results carried out to assess the correctness of recognition system.

Chapter # 7

Conclusions

7.1 Conclusions

The aim of this project was developing an Android based application that could take images of printed timetable, recognize the symbols and operators. The methodology relies on binarizing the image and extracting connected components from the binarized image. These components are then fed to the Tesseract OCR library for training. Once the library is trained, the components of the query document image are recognized and the recognized data is presented to the user. After this, the system evaluates the data to compute required output. The project employs a combination of different libraries including OpenCV and Tesseract recognition engine.

References

1. Due Trier, Ø., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. *Pattern recognition*, 29(4), 641-662.
2. Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1), 63-84.
3. My Class Schedule: Timetable
<https://play.google.com/store/apps/details?id=de.rakuun.MyClassSchedule.free&hl=en>
4. Timetable
<https://play.google.com/store/apps/details?id=com.gabrielittner.timetable&hl=en>
5. Schedule Planner Classic
<https://play.google.com/store/apps/details?id=com.intersog.android.schedule&hl=en>
6. BernátGábor “Introduction to opencv android”.18th Oct 2015
http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/android_dev_intro.htm
7. “Opencv for android Sdk” 21th Oct 2015
http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/O4A_SDK.html
8. Rav Smith “Tesseract Overview:” Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society (2007)2th Jan2015
9. Ray Smith “Tesseract OCR” 6th Jan 2015
<https://code.google.com/p/tesseract-ocr/>
10. “Testing Technique” 7th Sep 2015
<http://www.softwaretestinghelp.com/types-of-software-testing/>

