



**MANIPAL UNIVERSITY
JAIPUR**

**SCHOOL OF COMPUTING & INFORMATION
TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING**

SOFTWARE ENGINEERING LAB MANUAL

TANYA GUPTA

189302170

Software Requirements Specification

for

DocEasy

Version 1.0 approved

Prepared by Tanya

Manipal University Jaipur

February 5th, 2020

Table of Contents

Table of Contents	1
Revision History	1
1. Introduction	2
1.1 Purpose	2
1.2 Document Conventions	2
1.3 Intended Audience and Reading Suggestions	2
1.4 Product Scope	2
1.5 References	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Hardware Interfaces	5
3.3 Software Interfaces	5
3.4 Communications Interfaces	6
4. System Features	6
4.1 System Feature 1	6
4.2 System Feature 2 (and so on)	6
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	7
5.4 Software Quality Attributes	7
5.5 Business Rules	7
6. Other Requirements	7
Appendix : Analysis Models	8

Revision History

Name	Date	Reason For Changes	Version
DocEasy_trial	20/12/20	Poor user interface	trial
DocEasy	5/2/21	Final	1.0

1. Introduction

1.1 Purpose

This application aims to digitize handwritten documents using OCR technology. OCR also called Optical Character Reader is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the form. Scanned text documents, pictures stored in mobile phones are the main focus of this application. The purpose of this application is to recognize text in scanned text documents, text images, and any picture taken by an Android based device in order to reuse it later. This application will allow its users to perform many actions in a few minutes, such as copy text from these aforementioned documents and modify it, instead of wasting time on retyping it. This document aims to capture the system requirements and features particularly related to our product.

1.2 Document Conventions

In this complete document, we will mention priority as “low” or “high” throughout the document. All the user entities are written in capitalizations i.e. first letter as capital. Also, any significant term which has been described in the glossary is made bold and italic in the text. On the other hand, those terms which are significant (but not described in glossary) are bold in the text.

1.3 Intended Audience and Reading Suggestions

The purpose of this document is to give a detailed description of the requirements for the “DocEasy” software. It will illustrate the purpose, scope and complete description for the development of the system. It will also explain external interface requirements and system requirements as well as non-functional requirements.

Our software is most useful for students and employees working in software firms. They often face shortage of time for typing long documents. So our product will come as a boon to them using which they can write as many documents or presentations they need by hand without worrying about making them digitized for formal purposes.

1.4 Product Scope

“DocEasy” is a software application developed with the intention of scanning text images and converting them into digitized editable documents. The main purpose is to help the user to have an easy approach to scan any text and edit it with minimum hardware requirements. It maintains a high level of accuracy in recognizing the characters and all the documents scanned can be directly saved in our devices which can easily be accessed. The OCR technology used performs the tasks in less amount of time and more efficiently without the need of retyping the document.

1.5 References

- <https://nanonets.com/ocr-technology>
- <https://pypi.org/project/pytesseract/>

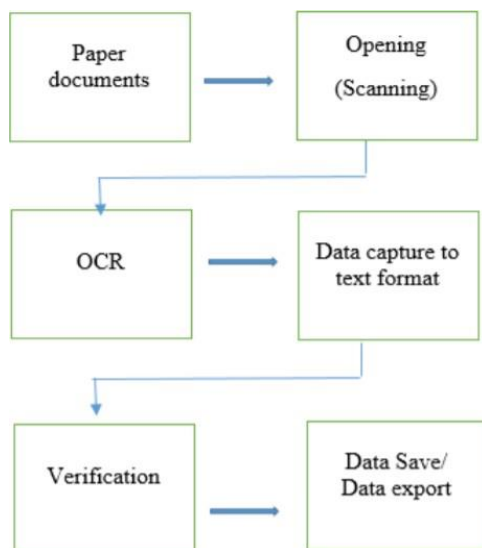
- <https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/>
- <https://flask.palletsprojects.com/en/1.1.x/>

2. Overall Description

Here we provide you with all the minor details about our project, its main features, the specific user categories the app is most useful for, and the operating environment with regards to the hardware and software compatibility among other things.

2.1 Product Perspective

Our product will use an existing technology but with added features. We will integrate our additional features with the existing OCR based technology Tesseract and release our product with the best user interface.



OCR PROCESS

This process generally consists of three stages: Open (Scan/Upload) the document, Capture and recognize data and then Save in a convenient format.

2.2 Product Functions

We aim to make an application which converts handwritten documents into digitized format. Upon successful conversion, it lets the user specify which format does he want the output into like a .pptx, .docx, .xlsx, etc. Our app will contain the following tabs for best user experience:-

- Tutorial/Help
- Upload Picture
- Import document
- Open with another application
- Edit
- Share

2.3 User Classes and Characteristics

This section describes how the different kinds of users can make the best out of our app.

2.3.1 Students

Students can write their assignments by hand and then use our app to digitize it. They no longer have to spend a great deal of time typing out their assignments.

2.3.2 Business Class

The reports and presentations for the business firms can be digitized after preparing a draft by hand. They can later be edited by the features our app provides. It speeds up such processes and is an example of efficient use of time and hard work.

2.3.3 Forensics

Illegible pieces of documents acquired at a crime spot may be digitized to convert it into a legible format. This may generate many clues to catch the culprits.

2.3.4 General Public

The app is equally suitable for the general public, according to their needs.

2.4 Operating Environment

This application doesn't require any network connection. It will work as a device inhouse application. This is very useful since sometimes people need to extract data from documents while the internet is not working, so this application will allow them to do so. It can be used by college students to scan documents, retrieve text from books in order to edit it and to convert any images such as slides into notes. They can write as many documents and presentations by hand without worrying about making them digitized for formal purposes. The application can also help its users such as foreigners to take a picture of any text image written in any language, retrieve it in order translate it to their native language in few minutes.

2.5 Design and Implementation Constraints

This application has been developed for devices with high camera resolution.

2.6 User Documentation

An online help tutorial would be given to the user prior to product installation.

2.7 Assumptions and Dependencies

The product internally uses the Pytesseract tool for running Optical Character Recognition on an image. Pytesseract is a wrapper for Tesseract-OCR Engine. So for proper functioning, it

requires the installation of tesseract-ocr executable file prior to use. It also uses standard python libraries like PIL to open and manipulate images. It is based on Flask for rendering web-based services to the user.

3. External Interface Requirements

3.1 User Interfaces

Users must choose picture from the phone's directory. The OCR will ignore the non-textual region of the picture and will print only the text. Also, the user has to follow the required steps in order to avoid any error while using the application.

The application will work as follow:

- *Choosing a picture from the phone's directory.*
- *Recognition of the text.*
- *Retrieving the text and making it editable.*

With a simple click, the user can take advantage of this application and perform many actions in a few minutes. By using the mobile's camera, different text images can be scanned, copied, and saved.

3.2 Hardware Interfaces

This application requires an android device with high camera resolution. It does not require any external hardware interface.

3.3 Software Interfaces

The product has a simple and easy to understand structured flow. A clear target image with least background details will act as an external input to the system. The system synchronises with the internal logical files (upload directory) for image storage and retrieval. Image will be further passed on to the Pytesseract module for recognition of text, which will be displayed as an external output on the monitor screen. The system also synchronises with the user's downloads folder to download the file in required extension specified by the user.

3.4 Communications Interfaces

The communication architecture follows the client-server model. The web services are rendered to the user over hypertext transfer protocol secure.

4. System Features

Following are the major challenges that DocEasy would solve when it is run the end user's computer / laptop.

4.1 Basic OCR and Download format

4.1.1 Description and Priority

User will be able to upload an image and run ocr on it (highest priority job). He/she could perform editing i.e. copy, paste or modify on the retrieved text (mid priority job). Further, they can also download the output of OCR in a file of their chosen extension (lowest priority task).

4.1.2 Stimulus/Response Sequences

User action → DocEasy Response

- Upload image → Store the image in uploads folder
- Run OCR → Call the Pytesseract API to render text recognition
- Select extension → Select the required file extension
- Download image → Store the image in downloads folder

4.1.3 Functional Requirements

REQ-1: The system will show option of choosing the desired text image. The user must use a camera of typical resolution and choose image from existing ones in his/her phone.

REQ-2: The text will be recognized from the image and it will be ready to use.

REQ-3: User will be able to copy, edit or modify the retrieved text and store it.

REQ-4 : Users will be able to choose the document type/format during storage.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The application response time shall be adequate and sufficient enough, that's why the time required for this application to respond to its user's actions has to be managed and controlled. But in order to maintain the performance of the application, the user has to follow the required steps to get the desired result.

5.2 Safety Requirements

This application should protect the confidentiality of the user's personal information and any personal data stored on his/her mobile phone.

5.3 Security Requirements

The security signature and certificate of the application is required as in any mobile application.

5.4 Software Quality Attributes

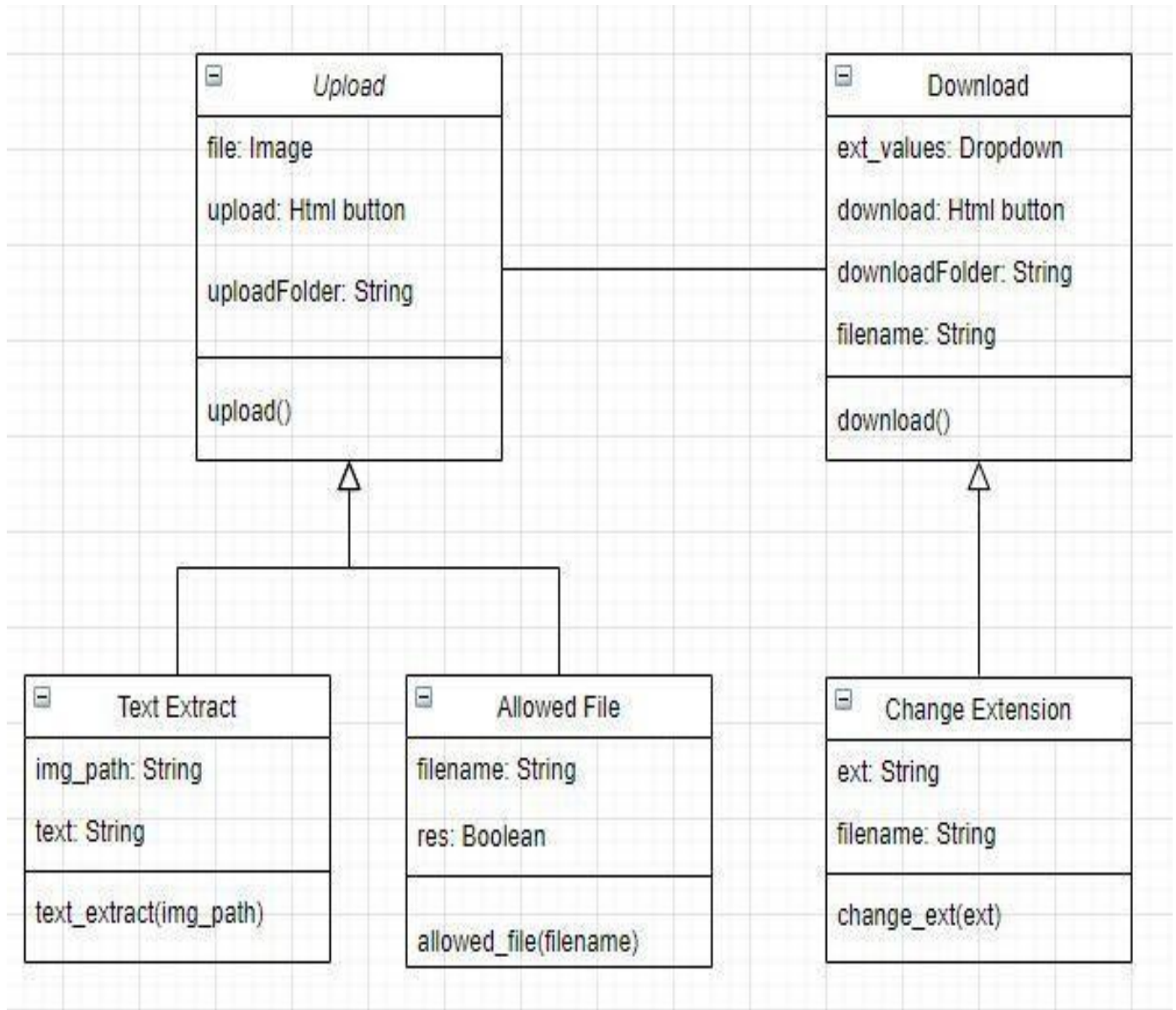
- Maintainability : *The application shall respond to any change on the requirements.*
- Adaptability : *The application shall be compatible to any OS.*
- Availability: *The application shall be available on the store whenever users want to download it.*
- Flexibility : *The architecture shall be flexible to any change of the requirements.*

6. Other Requirements

The product is very flexible can be extended and customised for a large organisation as per their needs with set legal procedures and bindings.

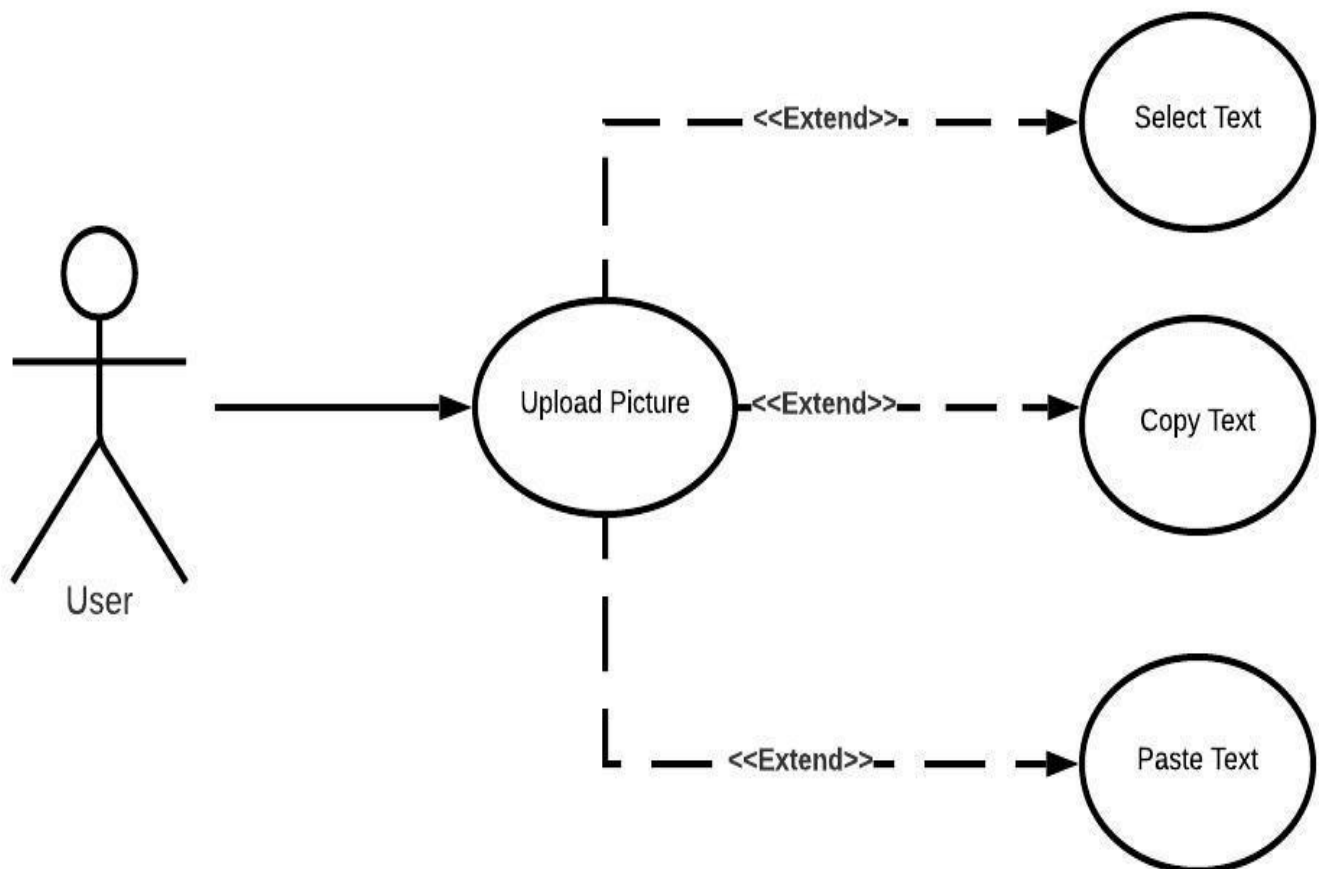
UML DIAGRAMS

Class Diagram

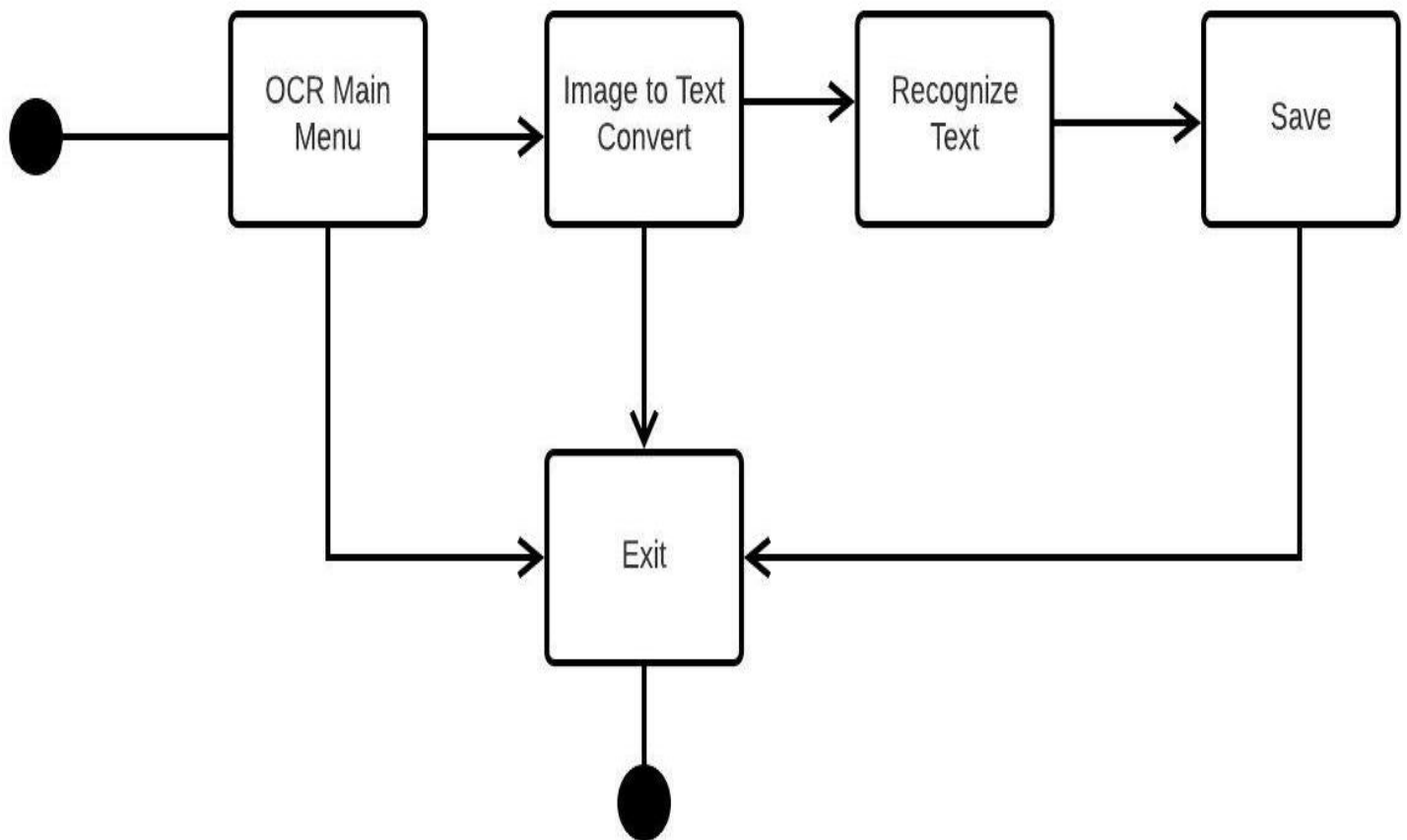


Use Case Diagram

USE CASE DIAGRAM

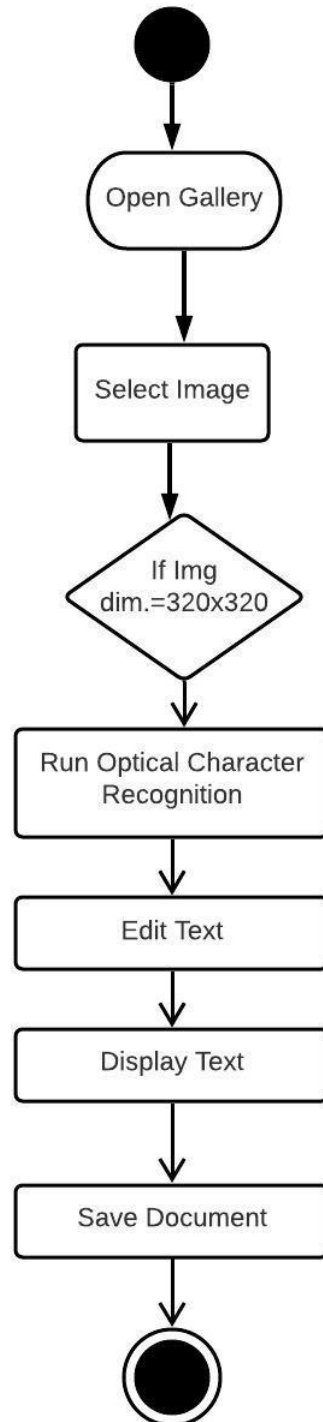


STATE CHART DIAGRAM



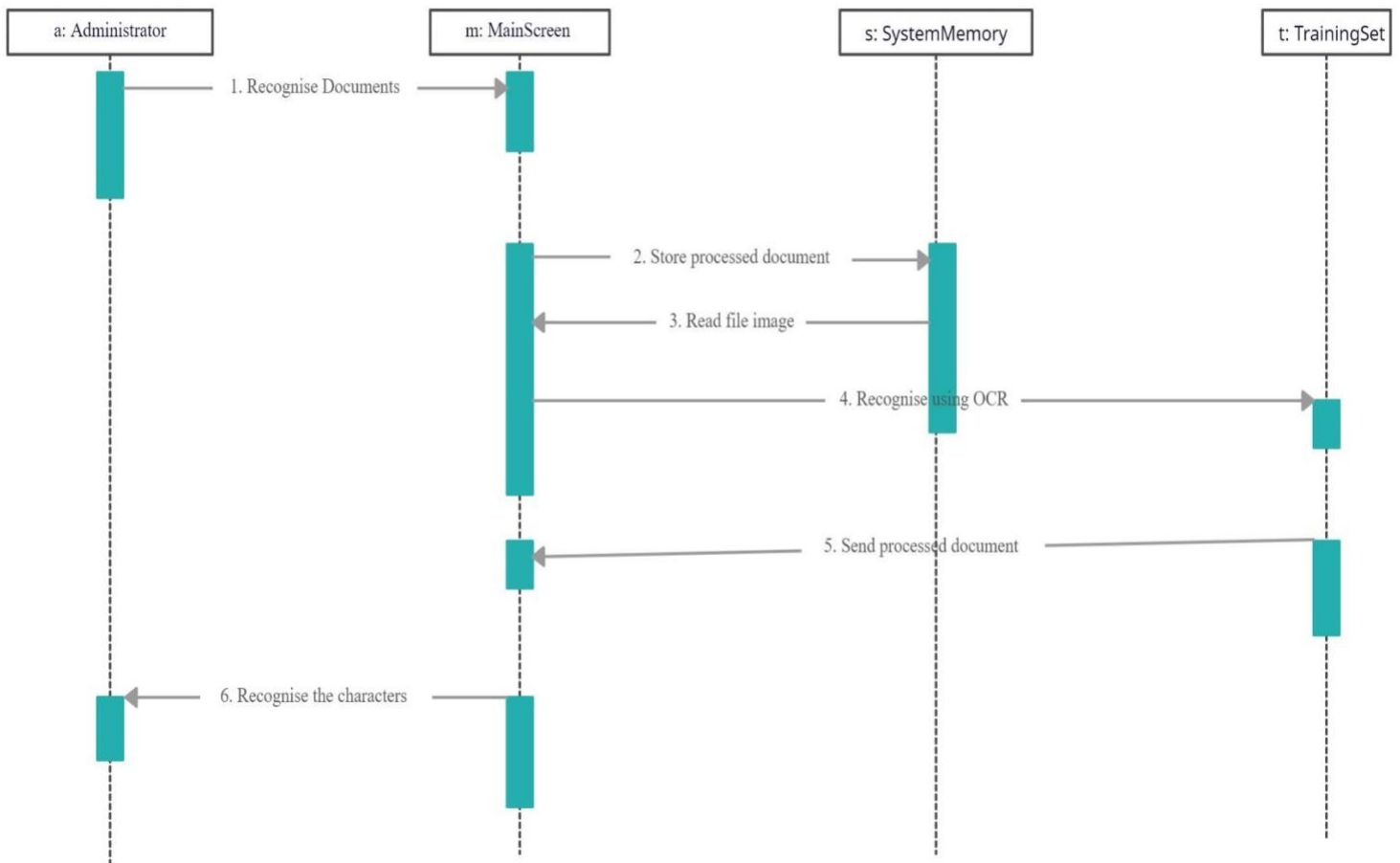
ACTIVITY DIAGRAM

ACTIVITY DIAGRAM



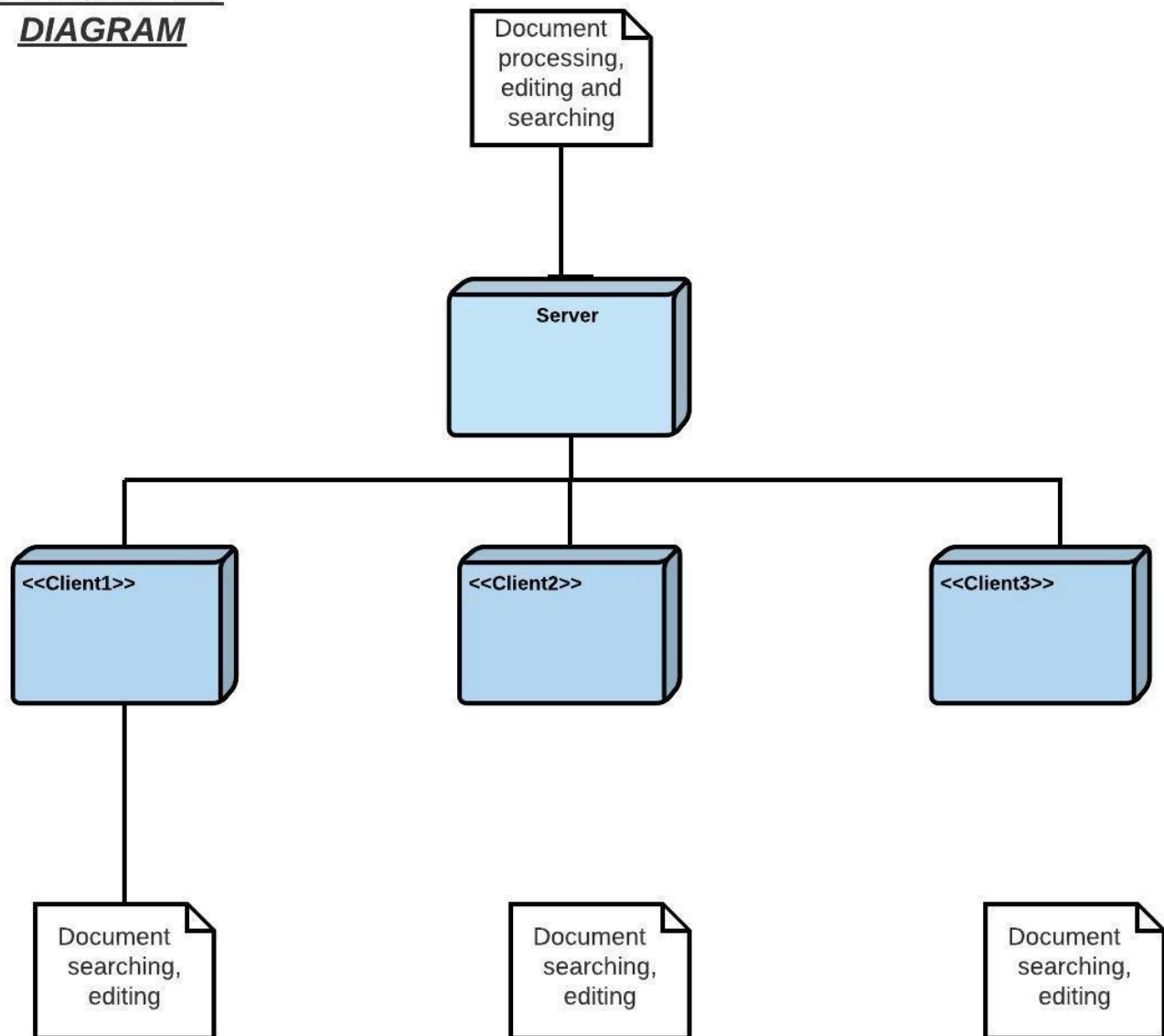
SEQUENCE DIAGRAM

SEQUENCE DIAGRAM



DEPLOYMENT DIAGRAM

DEPLOYMENT DIAGRAM



COMPONENT DIAGRAM

COMPONENT DIAGRAM

