

Parking System Based on PyTesseract

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

**MUVVALA SAI KRISHNA CHAITANYA
(37110475)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE
UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

March - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MUVVALA SAI KRISHNA CHAITANYA (37110475)** who carried out the project entitled "**Parking System Based on PyTesseract**" under my supervision from November 2020 to March 2021.

Internal Guide

Dr. A.C SHANTHA SHEELA, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I **MUVVALA SAI KRISHNA CHAITANYA** hereby declare that the Project Report entitled **Parking System Based on PyTesseract** done by me under the guidance of **Dr. A.C SHANTHA SHEELA, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A.C SHANTHA SHEELA, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The aim is to automate car parking system that can be parked in a given space based on the availability of parking spot and to design a system which takes the image of the number plate of a vehicle and the details are being retrieved Using character segmentation, which is performed by an optical feature extraction character recognition algorithm (OCR). Then the details retrieved from the number plate in the text the format is used to extract all the important information about a vehicle like, the name of the owner, etc. from the database. And the extracted text is used to search in SQLite database whether the user has a slot and can be used for car parking applications. Primarily, in this proposed work a website is being set up using a web framework called Flask which is written in Python. Finally, the objective of this to deploy a web based Optical Character Recognition (OCR) using Python and Python based web tool i.e. Flask and the Tesseract engine. The communication protocol thus used is HTTP.

TABLE OF CONTENT

CHAPTER.NO	TITLE	PAGE.NO
	ABSTRACT	V
	LIST OF FIGURES	VIII
1	INTRODUCTION	1
	1.2 OUTLINE OF PROJECT	2
	1.3 RESEARCH AND SIGNIFICANCE	5
2	AIM AND SCOPE	7
	2.1 AIM OF PROJECT	7
	2.1.1 OBJECTIVES	7
	2.2 SCOPE	7
	2.2.1 ADVANTAGES	8
	2.3 REDUCTION IN MANAGEMENT THROUGH OCR	9
3	SYSTEM DESIGN & METHODOLOGY	12
	3.1 EXISTING SYSTEM	12
	3.2 SYSTEM CAPABILITIES	13
	3.2.1 ADMIN	15
	3.2.2 PUBLIC	15
	3.2.3 REQUIREMENT SPECIFICATION	16
	3.3 SYSTEM FUNCTIONALITIES	16
	3.4 SYSTEM IMPLEMENTATION	18
	3.4.1 OCR SCRIPT	18
	3.4.2 FLASK WEB INTERFACE	19
	3.4.3 CREATING AN APPLICATION	19
	3.4.4 USING HTML TEMPLATES	20

	3.4.5 SETTING UP THE DATABASE	22
	3.5 MODULES	23
	3.5.1 PASSWORD MODULE	25
	3.5.2 SEARCH MODULE	25
	3.5.3 REGISTRATION MODULE	26
4	SOFTWARE DEVELOPMENT	27
	METHODOLOGY	
	4.1 DESCRIPTION OF DIAGRAM	27
	4.2 ARCHITECTURAL DESIGN	28
	4.3 ACTIVITY DIAGRAM	31
	4.4 EMERGING TECHNOLOGY	33
5	RESULTS AND DISCUSSION	36
6	CONCLUSION AND FUTURE WORK	44
	REFERENCES	45
	APPENDIX	46
	SOURCE CODE	46

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	BLOCK DIAGRAM	27
4.2	ARCHITECTURAL DESIGN	28
4.3	ACTIVITY DIAGRAM	31
5.1	LOG-IN-PAGE	36
5.2	REGISTRATION FORM	37
5.3	USER UPLOADING IMAGE	38
5.4	DISPLAY SLOT PAGE	39
5.5	USER REGISTER DASHBOARD	40
5.6	PARKING SLOT DASHBOARD	41
5.7	ACCURACY GRAPH	42
5.8	LICENSE PLATE ACCURACY LIST	43

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Humans can understand the contents of an image simply by looking. We perceive the text on the image as text and can read it. Computers don't work the same way. They need something more concrete, organized in a way they can understand.

This is where Optical Character Recognition (OCR) kicks in. Whether it's recognition of car plates from a camera, or hand-written documents that should be converted into a digital copy, this technique is very useful. And it is very convenient and makes it a lot easier and faster for some people to do their jobs. Optical Character Recognition involves the detection of text content on images and translation of the images to encoded text that the computer can easily understand. An image containing text is scanned and analysed in order to identify the characters in it. Upon identification, the character is converted to machine-encoded text. To us, text on an image is easily discernible and we are able to detect characters and read the text, but to a computer, it is all a series of dots.

Remember the days when you had to fill in the dots of the right answer during an exam? Or how about the aptitude test you gave before your first job? I can vividly recall the Olympiads and multiple-choice tests where universities and organizations used an Optical Character Recognition (OCR) system to grade the answer sheets in droves. Honestly, OCR has applications in a broad range of industries and functions. So, everything from scanning documents – bank statements, receipts, handwritten documents, coupons, etc., to reading street signs in autonomous vehicles – this all falls under the OCR umbrella. OCR systems used to be quite expensive and cumbersome to build a couple of decades ago. But advances in the computer vision and deep learning field mean we can build our own OCR system now.

OCR systems transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible. The sub-processes like text

localization, character recognition etc. steps needed to approach automatic character recognition.

There are two points to note in understanding how OCR software uses technology to read patterns and features. A simple OCR program may be designed to read and recognize many different fonts and patterns. When any of the compatible patterns are presented to the software, it uses technology to convert it to machine-readable text accurately so it becomes searchable. The image is first scanned and the text and graphics elements are converted into a bitmap, which is essentially a matrix of black and white dots. The image is then pre-processed where the brightness and contrast are adjusted to enhance the accuracy of the process. The image is now split into zones identifying the areas of interest such as where the images or text are and this helps kick-off the extraction process. The areas containing text can now be broken down further into lines and words and characters and now the software is able to match the characters through comparison. The final result is the text in the image that we're given.

1.2 OUTLINE OF PROJECT

This system deals with managing, controlling and processing. It is the situation where a person violates the articles/sections under the constitution, which goes against the rules and regulations that is the ethics of the society. For getting rid of these types of problems the security follows many processes like CCTV surveillance, and collecting information. This system will help booking the slot in a web-site and managing the user records through a computerized manner in a database. It will act as the best friend of the admin and also for the citizens. The admin will be able to track the person and can continue required process. The maintenance of the database is very essential as it one of the important factors in vehicle management. The database helps in retrieving a user record or any other information whenever it is required. It provides authentication to keep the database safe and to prevent from any loss or damage of the database. Only the authorized person can access the database and it is provided to the admin, so that the confidentiality and security is maintained.

When the person gets slot the user allocated his license plate number to parking slot table with slot number. The admin update the status of user so that the user gets information of each and everything. They maintain the database and can even

make changes when required. The users using this system need to get access from admin by registering and log in. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document or image. The users can get aware of the slot booking taking place in their society. The user will be updated the status as process proceeds. The user cannot make any change, as they will only be able to access the portal.

In this system we have used DB browser (SQLite) for building the database. The User interface is built by using html. The backend and the main framework of the program are done in Flask. Keeping in mind the security of this application we have assigned the authentication and authorization work to the admin who will be able to make changes and provide access to the users, the user need to first register for log-in and then use it. In the current system it is traditional one where the user needs to visit to the car parking station for lodging a vehicle. And the security maintains the data in a register which contains all the information regarding the data and need to maintain it manually. It's a very lengthy process in which the security does collect data one by one.

This system also shows the status to the user. Thus, this system makes the whole process very easy and increase the flexibility of the system. This system will help to decrease the problem to an extent and making the citizens aware of their society and country. Social and technological changes, so the users will increase. Improved and faster technology and worldwide communications make it easier to organize.

We will use the Python-Tesseract, or simply PyTesseract, library which is a wrapper for Google's Tesseract-OCR Engine. I chose this because it is completely open-source and being developed and maintained by the giant that is Google. We will also use the Flask web framework to create our simple OCR server where we can upload photos for character recognition purposes. Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file. Flask is also extensible and doesn't force a particular directory structure or require complicated boilerplate code before getting started. We are also going to use an environment consists of an interpreter and any number of installed packages. i.e., the Python extension for VS

Code provides helpful integration features for working with different environments. Most people are used to working with programs that provide a graphical user interface (GUI). You can make your Python code accessible to more people by building a GUI for it. Besides those, we'll also use the cv2 library which is a fork of the image processing to handle the opening and manipulation of images in many formats in Python. The advantage of web applications is that they're platform independent and can be run by anyone who has access to the Internet. Their code is implemented on a back-end server, where the program processes incoming requests and responds through a shared protocol that's understood by all browsers. Websites had fixed content that was the same for every user who accessed that page.

These web pages are called static because their content doesn't change when you interact with them. When serving a static web page, a web server responds to your request by sending back the content of that page, regardless of who you are or what other actions you took. Such static websites aren't considered applications since their content isn't generated dynamically by code. While static sites used to make up the entire Internet, most websites today are true web applications, which offer dynamic web pages that can change the content they deliver.

Python-driven web applications use Python code to determine what actions to take and what content to show. Our code is run by the web server that hosts your website, which means that your users don't need to install anything. All they need to interact with your code is a browser and an Internet connection. Serving dynamic content over the Internet involves a lot of different pieces, and they all have to communicate with one another to function correctly. The general process of how content is delivered over the Internet. The programming language used on the server, as well as the technologies used to establish that connection can differ. However, the concept used to communicate across HTTP requests and responses remains the same and is called the HTTP Request-Response Cycle. Python powers many large web applications and is a common choice as a back-end language. Many Python-driven web applications are planned from the start as web applications and are built using Python web frameworks such as Flask. We'll start by developing the Flask back-end layer to serve the results of the OCR engine. From there you can just hit the endpoint and serve the results to the end user in the manner that suits you.

1.3 RESEARCH AND SIGNIFICANCE

The primary aim of this proposed system is to assist the parking system using Optical Character Recognition in such a way that they can get all the necessary and detailed information of the car that took place in a particular area or society which will help the branch department to start the process in a much faster way. The proposed system also provides information or status of the booked slots. There are several intents of this proposed system. Some of the important intents are as follows:

- This system provides information of owner to both the user and the admin.
- This system has SQLite database which helps in retrieving information by the admin whenever it is necessary.
- It makes the whole process efficient.
- This process is based on digitalization.
- It provides authentication and keeps the system secured.

The proposed model is based on the new technologies which makes the whole process efficient. It is going to be very important and profitable for the whole society especially for the user.

OCR has two parts to it. The first part is text detection where the textual part within the image is determined. This localization of text within the image is important for the second part of OCR, text recognition, where the text is extracted from the image. Using these techniques together is how you can extract text from any image. OCR has widespread applications across industries (primarily with the aim of reducing manual human effort). It has been incorporated in our everyday life to an extent that we hardly ever notice it! But they surely strive to bring a better user experience.

OCR is used for handwriting recognition tasks to extract information. A lot of work is going on in this field and we have made some really significant advancement. Microsoft has come up with an awesome mathematical application that takes as input a handwritten mathematical equation and generates the solution along with a step-by-step explanation of the working.

OCR is increasingly being used for digitization by various industries to cut down manual workload. This makes it very easy and efficient to extract and store information from business documents, receipts, invoices, passports, etc. Also, when you upload your documents for KYC (Know Your Customer), OCR is used to extract information from these documents and store them for future reference.

OCR is also used for book scanning where it turns raw images into a digital text format. Many large scale projects like the Gutenberg project, Million Book Project, and Google Books use OCR to scan and digitize books and store the works as an archive.

The banking industry is also increasingly using OCR to archive client-related paperwork, like on boarding material, to easily create a client repository. This significantly reduces the on boarding time and thereby improves the user experience. Also, banks use OCR to extract information like account number, amount, cheque number from cheques for faster processing.

The applications of OCR are incomplete without mentioning their use in self-driving cars. Autonomous cars rely extensively on OCR to read signposts and traffic signs. An effective understanding of these signs makes autonomous cars safe for pedestrians and other vehicles that ply on the roads. There are definitely many more applications of OCR like vehicle number plate recognition, converting scanned documents into editable word documents, and many more.

One of the major challenges of data entry is inaccuracy. Automated data entry tools such as OCR data entry result in reduced errors and inaccuracies, resulting in efficient data entry. Besides, problems like data loss can also be successfully tackled by OCR data entry. As there is no manpower involved, the issues such as keying in wrong information accidentally or otherwise can be eliminated.

CHAPTER 2

AIM AND SCOPE

2.1 AIM OF PROJECT

The primary aim of this proposed system i.e. Number plate detection using OCR is to assist this parking system in such a way that they can get all the necessary and detailed information where to park and also which will help the system department to store the user data in much easier way. The proposed system also provides information to the user and status of the user to the admin regarding the present slots booked by the user.

2.1.1 OBJECTIVES

The proposed model “Number plate detection using OCR” is based on the new technologies which makes the whole process efficient. It is going to be very important and profitable for the whole society especially for the user and for the management system. This system will provide more accurate information like extracted text of image, details of user, present slots to the admin etc. which will be very helpful for all members to process the procedures and thus it will take less time to complete the process.

2.2 SCOPE

This system is used to develop an efficient car parking system using image processing methods. Automated car parking system allows users to book the available parking slots at the convenience. The system consists of two major sections: Admin and user. A user can book the slots through an web based application. The parking area consists of slots which are uniquely numbered. Optical Character Recognition (OCR) technique is used to detect the text in license plate and allot the available slots in the parking area to the user. These vacant slots are then allocated to the users.

License plate detection for parking is a proposal system aimed at automating the current parking system that is mostly done manually. The other advantage of this project is that, it is cost efficient and secured system as it authorizes every number plate. This system has further scope for improvement in recognition of number plates accurately.

In order to improve the operational efficiency for the public, Smart City uses information, connectivity and technology to help accelerate towards improving the

quality of life for people. Internet of Things (IoT), Automation, and Machine Learning are the new technologies driving the implementation of smart cities. Any city can be considered a smart city initiative by introducing a system such as smart parking using a web based applications to help drivers locate parking slots, smart traffic management track and analyse traffic flows, sharing information electronically, monitoring changes to the environment enabled sanitation, etc. The advancement of technology has led to a lot of changes in the world. But these changes haven't affected the parking system which is still the same and time consuming. It takes a lot of effort and time to find a vacant parking slot. It is known that there is a huge problem of traffic especially in countries where population is high. If population is high, more people are there to park in a limited number of parking slots. Intelligent parking schemes have been significantly improved in urban environments and the associated technologies. Some types of car parking system are also required wherever traffic is relevant. Rapid population growth in the urban areas and increased traffic congestion has increased smart parking demand.

This proposal aims to implement a reservation based automated car parking system to address the rising problems faced due to traffic congestion. The identification of vacant spots is implemented based on image processing techniques thereby overcoming the disadvantages of conventional sensor based methods. The reservation of slots through mobile application can help drivers locate parking slots with minimum effort and in less time. The user authentication is ensured by proper verification of details implemented by database.

2.2.1 ADVANTAGES

- Time saving as it is operated online.
- This system has database which helps in retrieving information whenever it is necessary.
- It provides authentication and keeps the system secured.
- This process is based on digitalization

2.3 REDUCTION IN MANAGEMENT THROUGH OCR

Optical Character Recognition (OCR) is one of the most widely implemented types of data entry methods. OCR involves digitized scanning and recognition of written or printed text. Here the text is first photo scanned, analysed, and is finally translated into character codes. This machine-encoded text can be easily searched and edited electronically.

OCR has greatly improved the process of data entry. This software tool enables quick conversion of scanned documents to searchable text files. Today, the need for the documents to be scanned is on a constant rise as it enables these documents to be viewed conveniently when required. The scanned documents can also be shared easily through the electronic medium.

Although there are numerous advantages of OCR, it mainly helps businesses in increasing the effectiveness and efficiency of the work. Its ability to quickly search through enormous content is extremely helpful, particularly in office settings, which deal with high document inflow and high volume scanning.

OCR software helps businesses to achieve higher productivity by facilitating quicker data retrieval when required. The time and effort which the employees were required to put in for extracting relevant data can now be channelized to focus on core activities. Besides, employees do not have to make numerous trips to central records room to access the required documents, as they can access them without getting up from their desks.

Opting for OCR will help businesses on cutting down on hiring professionals to carry out data extraction, which is one of the most important benefits of OCR data entry methods. This tool also helps in trimming various other costs, such as copying, printing, shipping, etc. Therefore, OCR eliminates the cost of misplaced or lost documents and offers higher savings in the form of reclaimed office space, which would otherwise be used for storing paper documents.

One of the major challenges of data entry is inaccuracy. Automated data entry tools such as OCR data entry result in reduced errors and inaccuracies, resulting in efficient data entry. Besides, problems like data loss can also be successfully tackled by OCR data entry. As there is no manpower involved, the issues such as keying in wrong information accidentally or otherwise can be eliminated.

OCR can scan, document, and catalogue information from enterprise-wide paper documents. This simply means that the data can now be stored in an electronic format in servers, eradicating the need for maintaining huge paper files. In this way, OCR data entry serves as one of the best tools to implement "Paperless" approach across the organization.

Data security is of utmost importance for any organization. Paper documents are easily prone to loss or destruction. Papers can be misplaced, stolen, or destroyed by natural elements such as moisture, pests, and fire. However, this is not the case with data that is scanned, analysed, and stored in digital formats. Furthermore, the access to these digital documents can also be minimized to prevent mishandling of the digitized data.

One of the huge advantages of OCR data processing is that it makes the digitized documents completely text searchable. This helps professionals to quickly lookup numbers, addresses, names, and various other parameters that differentiate the document being searched.

Several inbound contact centres often provide information that their customers seek. While some call centres provide customers with the information they need, the others will have to quickly access certain personal or order-related information of the customers to process their requests. Quick data accessibility becomes extremely important in such cases. OCR helps in systematically storing and retrieving the documents digitally at blazing speeds. With this, the waiting time is drastically reduced for the customers, thereby improving their experience.

Scanned documents need to be edited most of the time, particularly when some information must be updated. OCR converts data to any preferred formats such as Word, etc., which can be easily edited. This can be of great help when there are contents which have to be constantly updated or regularly changed.

Disaster recovery is one of the major benefits of using OCR for data entry. When data is stored electronically in secure servers and distributed systems, it remains safe even under emergency situations. When there are sudden fire breakouts or natural calamity, the digitized data can be quickly retrieved to ensure business continuation.

CHAPTER 3

SYSTEM DESIGN & METHODOLOGY

3.1 EXISTING SYSTEM

In existing recognition system of vehicle license plates are used in monitoring, observing and traffic control systems. In most studies, the main stages of license plate recognition are image pre-processing, plate detection, license plate image

segmentation into symbols and text character recognition. An important and basic step in recognizing vehicle number symbols is the detection and localization (detection) of areas that potentially contain license plates in the image. An error in detecting a license plate will lead to a failure of the algorithms in the subsequent steps and a decrease in the accuracy of character recognition. The procedure for detecting a plate is a difficult task due to the wide variety of the processed image characteristics: size, colour, spatial orientation of the plate, scene complexity. Moreover, the quality of detection significantly depends on the image contrast, the presence of distortions caused by natural artefacts, glare, greases, etc. It should also be noted that the global increase in video data due to an increase in the number of vehicles and video monitoring cameras, creates significant problems for their analysis and real time processing.

The proposed system is designed and developed in such a way so that it becomes user friendly. In this system the user interface build is very attractive and professional. The system has access to the users and the citizens. The admin of the system can only control, make changes and maintain the databases. The admin provides the required security to the users so that the data does not get deleted or leaked. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the site and the important information given. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document. The user will be updated the status as available and proceeds. The user cannot make any change, as they will only be able to access the portal. In this system we have used SQLite for building the database. The User interface is built by using html. The backend and the main framework of the program are done in flask. As flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. The system needs DB browser (SQLite) for changing the changes and it can only done by admin.

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. Thus our need is to develop character recognition software system to perform Document Image Analysis which transforms documents in paper format to electronic format. For this process there are various techniques in the world. Among all those techniques we have chosen Optical Character Recognition as main fundamental technique to recognize characters.

3.2 SYSTEM CAPABILITIES

The proposed system i.e. Vehicle Number Plate Detection System Based on OCR consist of various features which makes this system efficient for the user and the admin. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the site and the important information given. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the site and the important information given. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related license number plate image. The citizens can get aware of the slot booking taking place in their society and get alert. The user needs to go the apply option for slot booking after that the user needs to fill all the required data and upload the related image. The citizens can get aware of the smart technology taking place in their society and get alert. The user gets all the option to contact the admin in case of any doubt. When the slot gets booked then the admin line automatically updated the status of booking, if it is booked before and after user completed work and get

out then slot booking status changes to available. So that another user gets slot for booking each and everything. Admin maintain the database and can even make changes when required. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the site and the important information given. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document. The user will be updated the status as available and proceeds. The user cannot make any change, as they will only be able to access the portal. In this system we have used SQLite for building the database. The User interface is built by using html. The backend and the main framework of the program are done in flask. As flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. The system needs DB browser (SQLite) for changing the changes and it can only done by admin. Slot booking is a concept that has been applied in a number of ways to the problem of parking. Understanding this system as it requires studying intentions, as well as consequences. The user will be updated the status as booking proceeds. The user cannot make any change, as they will only be able to access the portal. In this system we have used SQLite for building the database. The User interface is built by using html. The features of the proposed system make it a different one from the traditional method. The proposed model has different section. The sections are as follows:

3.2.1 Admin

This section is known as admin page which controls the portal, maintains the database and makes necessary changes whenever required. It gives authentication and authorization to the users. A user form is provided in this page which helps the user to book a slot. Using this system need to get access from admin by registering and log in. The user in the system can go through the feed and the information given. The user needs to go the apply option for book a slot, after that the user needs to fill all the required data and upload the related document or image. The user gets the slot in case of free slots. The user will be updated the status as the process proceeds. The user cannot make any change, as they will only be able to

access the portal. The data or user records can be edited or deleted as per requirement. Different sections have been made, one for the user and the other for the admin. The log-in credentials are also provided by the admin and it is also maintained by admin. Admin can also update the system by providing multi features and accessing can also be provided by the admin.

3.2.2 Public

This section is for the public or any users. In this section, the access is given for the public. Registration and the log-in form are provided where the users need to give their log in credentials (for registered user) or for new user they need to register first. The system needs DB browser (SQLite) for working. Keeping in mind the security of this application we have assigned the authentication and authorization work to the admin who will be able to make changes and provide access to the users, the user need to first register for log-in and then use it. In the current scenario it is traditional one where the user needs to visit to the parking man security for lodging a vehicle. A manual form is also provided in this page which helps the user to search for slots. Using this system need to get access from admin by registering and log in. The user in the system can go through information given. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document or image of license plate. The user will be updated the status as process proceeds. The user cannot make any change, as they will only be able to access the portal. The data or the user records can be edited or deleted as per requirement. Different sections have been made, one for the user and the other for the admin. Here the admin maintain the data in a register which contains all the information regarding the slot information and need to maintain it manually. After logging in, the user or public can reserve the slot and the information will be reached to the admin.

3.2.3 REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer's machines already. So the main need is to introduce appropriate equipment for the product. The requirements are split into two categories, namely:

Software Requirements

The basic software requirements to run the program are:

- i. Microsoft Windows XP, Windows 7, Windows 10
- ii. DB browser, SQLite
- iii. HTML, CSS
- iv. Browser e.g. Mozilla Firefox, Safari, chrome, etc.
- v. Local Server

Hardware Requirements

The basic hardware required to run the program are:

- i. Hard disk of 20 GB
- ii. System memory (RAM) of 512 MB
- iii. I3 processor-based computer or higher.

3.3 SYSTEM FUNCTIONALITIES

This system model has different types of functions. A straightforward working methodology is incorporated so the client can comprehend the distinctive capacities obviously and rapidly. At first as an initial step the executable type of the application is to be made and stacked in the server machine which is available to the whole client and the server is to be associated with a system. Keeping in mind the security of this application we have assigned the authentication and authorization work to the admin who will be able to make changes and provide access to the users, the user need to first register for log-in and then use it. In the current system it is traditional one where the user needs to visit to the park station for lodging a vehicle. The admin can make an alert if there is any kind of free slots. When the slot gets booked then the admin update the status of slot booked, so that the user gets information of each and everything. They maintain the database and can even make changes when required. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the site and the information given. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document or vehicle number plate image. Here the admin maintain the data in a register which

contains all the information regarding the user records and need to maintain it manually.

Some of the functions are given below:

- The user can book slot by filling the form that is provided in the website.
- The user can also provide necessary information like vehicle number, Account number, mail, etc. which will be very helpful for the admin as well as user.
- This system is much secured as it need log in credentials to access the website.
- Admin can also edit or delete information if necessary.

People must be made educated and well aware of the technologies that can help them to get rid of problem in their society. Everyone in the society must be made aware of online booking so that they can use it properly at the time of their requirement. The public must be informed that they can help a lot in reducing time. The people should be made aware of the process that they need to go through after booking a slot. We can see in developed countries like USA, Russia, UAE, etc. where the traffic in parking or where ever the rates are very less due to the high rate of employment and education. It is spatially distributed in patterns some patterns are discovered while others are not.

3.4 SYSTEM IMPLEMENTATION

We are going to implement this project in 2 phases. In the first, we'll create the script, and in the next we'll build a Flask application to act as an interface.

3.4.1 OCR Script

With the setup complete, we can now create a simple function that takes an image and returns the text detected in the image - this will be the core of our project. The function is quite straightforward; we import Image from cv2 and our PyTesseract library. We then create an ocr_core function that takes in a file name and returns the text contained in the image. It is easy for humans to understand the contents of an image by just looking at it. You can recognize the text on the image and can understand it without much difficulty. However, computers don't function similarly. They only understand information that is organized. And this is exactly where Optical Character Recognition comes in the picture.

Tesseract is an open-source text recognition engine that is available under the Apache 2.0 license and its development has been sponsored by Google since 2006. In the year 2006, Tesseract was considered as one of the most accurate open-source OCR engines. You can use it directly or can use the API to extract the printed text from images. The best part is that it supports an extensive variety of languages. It is through wrappers that Tesseract can be made compatible with different programming languages and frameworks. In this blog, I'll be using the Python wrapper named pytesseract. It is used to recognize text from a large document, or it can also be used to recognize text from an image of a single text line.

By default, Tesseract considers the input image as a page of text in segments. You can configure Tesseract's different segmentations if you are interested in capturing a small region of text from the image.

OpenCV (Open Source Computer Vision Library), as the name suggests, is an open source computer vision and a machine learning software library. OpenCV was built to furnish a common infrastructure for computer vision applications. Besides this, it also accelerates the use of machine perception in commercial products. And being a licensed product, OpenCV comes in handy for businesses to utilize and modify the code. The library comprises more than 2500 optimized algorithms that have a comprehensive set of both classic and modern computer vision along with the machine learning algorithms. These algorithms can be used in detecting and recognizing faces, in classifying human actions in videos, extracting 3D models of objects, and many more. Here, I will use it to detect the text from an image file. Tesseract requires a clean image to detect the text; this is where OpenCV plays an important role as it performs the operations on an image like converting a colored image to binary image, adjusting the contrast of an image like bilateral filter and many more.

3.4.2 Flask Web Interface

Flask is a web application framework written in Python. This would make it more user-friendly and versatile. Flask is a web framework that provides libraries to build lightweight web applications in python. For instance, we can upload photos via the website and get the extracted text displayed on the website and perform character recognition on them. We'll now create a templates folder to host our HTML files.

Now integrate our OCR script on the web application. First, we'll add functionality to upload images to our Flask app and pass them to the `ocr_core` function. We will then render the image beside the extracted text on our web app as a result. The function detects the text in the image and returns it. Finally, as a response to the image upload, we render the detected text alongside the image for the user to see the results. Flask application has been able to integrate the OCR functionality and display the text on the browser.

3.4.3 Creating an Application

We first import the Flask object from the flask package. Flask can make use of the SQLite3 module of the python to create the database web applications. You then use it to create your Flask application instance with the name `app`. You pass the special variable `__name__` that holds the name of the current Python module. It's used to tell the instance where it's located—you need this because Flask sets up some paths behind the scenes. Once you create the app instance, you use it to handle incoming web requests and send responses to the user. The concept used to communicate across HTTP requests and responses remains the same and is called the HTTP Request-Response Cycle. Python powers many large web applications and is a common choice as a back-end language. Many Python-driven web applications are planned from the start as web applications and are built using Python web frameworks such as Flask. We'll start by developing the Flask back-end layer to serve the results of the OCR engine. From there you can just hit the endpoint and serve the results to the end user in the manner that suits you. `@app.route` is a decorator that turns a regular Python function into a Flask view function, which converts the function's return value into an HTTP response to be displayed by an HTTP client, such as a web browser. You pass the value `'/'` to `@app.route ()` to signify that this function will respond to web requests for the URL `/`, which is the main URL. As we can see here, the `/` URL is bound to the main function which is responsible for returning the server response. It can return a string to be printed on the browser's window or we can use the HTML template to return the HTML file as a response from the server.

App routing is used to map the specific URL with the associated function that is intended to perform some task. The URL ('/') is associated with the home function that returns a particular string displayed on the web page. In other words, we can say that if we visit the particular URL mapped to some particular function, the output of that function is rendered on the browser's screen. The converter can also be used in the URL to map the specified variable to the particular data type. The flask URL function is used to build a URL to the specific function dynamically. The first argument is the name of the specified function, and then we can pass any number of keyword arguments corresponding to the variable part of the URL. We have returned the simple string as the response from the view function. Flask facilitates us to return the response in the form of HTML templates. This function is useful in the sense that we can avoid hard-coding the URLs into the templates by dynamically building them using this function. We can change the URLs dynamically instead of remembering the manually changed hard-coded URLs. The generated paths are always absolute, avoiding unexpected behaviour of relative paths in browsers.

3.4.4 Using HTML templates

Currently your application only displays a simple message without any HTML. Web applications mainly use HTML to display information for the visitor, so you'll now work on incorporating HTML files in your app, which can be displayed on the web browser. Flask provides a `render_template()` helper function that allows use of the template engine. This will make managing HTML much easier by writing your HTML code in `.html` files as well as using logic in your HTML code. You'll use these HTML files, (templates) to build all of your application pages, such as the main page where you'll display the current blog posts, and so on. We'll import the Flask object to create a Flask application instance. Also import the `render_template()` helper function that lets you render HTML template files that exist in the templates folder. The file will have a single view function that will be responsible for handling requests to the main `/` route. The `login()` view function returns the result of calling `render_template()` with `login.html` as an argument, this tells `render_template()` to look for a file called `index.html` in the templates folder.

HTTP is the hypertext transfer protocol which is considered as the foundation of the data transfer in the World Wide Web. All web frameworks including flask need to

provide several HTTP methods for data communication. We can specify which HTTP method to be used to handle the requests in the route () function of the Flask class. By default, the requests are handled by the GET () method. We can also check the URL which also contains the data sent with the request to the server. This is an important difference between the GET requests and the POST requests as the data sent to the server is not shown in the URL on the browser in the POST requests.

Flask facilitates us to render the external HTML file instead of hardcoding the HTML in the view function. Here, we can take advantage of the jinja2 template engine on which the flask is based. Flask provides us the render template () function which can be used to render the external HTML file to be returned as the response from the view function. In the client-server architecture, the request object contains all the data that is sent from the client to the server. We can retrieve the data at the server side using the HTTP methods.

File uploading is the process of transmitting the binary or normal files to the server. Flask facilitates us to upload the files easily. All we need to have an HTML form with the encryption set to multipart/form-data.

The server-side flask script fetches the file from the request object using files[] Object. On successfully uploading the file, it is saved to the desired location on the server.

The uploaded file is saved to the temporary directory of the server for a while before it is saved to some desired location.

Since we are using the development server on the same device, hence the file will be uploaded to the directory from where the flask script is executed.

In the next step, we'll set up a database that will store your application data.

3.4.5 Setting up the Database

We'll set up a database to store data. And use a SQLite database file to store your data because the sqlite3 module, which we will use to interact with the database, is readily available in the standard Python library. First, because data in SQLite is stored in tables and columns, next, create database car parking, that is used to

create the user register table. Open a file called carparking.db inside your project folder. User register table contains following columns mainly are:

Name: name of the user

Pwd: An integer that represents a primary key, this will get assigned a unique value by the user for each entry.

Vnum: It is vehicle number which user has.

SQLite is an in-process library that implements a self-contained, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite exposes methods to manage a SQLite database. SQLite Database has methods to create, delete, execute SQL commands, and perform other common database management tasks.

SQLite is a relational database system that uses the SQL query language to interact with the database. Each database can have tables and each table can have records.

The SQLite database stores all data in a single file. You can create a SQLite database from Python code. The program creates a SQLite database 'carparking.db' where the student tables are created.

Import the sqlite3 module and then open a connection to a database file named carparking.db, which will be created once you run the Python file. Next you execute its contents present in the code that executes multiple statements at once, which will create the database as well as table. Finally, you commit the changes and close the connection. Once the file finishes execution, a new file called carparking.db will appear in your project directory. This means you've successfully set up your database.

3.5 MODULES

This system model consists of mainly two actors. They are admin and the user. The primary task of the admin is to maintain the database records and provide authentication to the user. The user uses the credentials to access the website so that they can book a slot to park. The user needs to make the login and will receive

an acknowledgement with the slot number. Then the user needs to visit the park area to park. Even for any updates of the system the user needs to visit to admin. The admin can make a statement if there is any kind of information. When the user gets slot the user can park the car. The admin update the status of parking slot table so that the admin gets information of each and everything. They maintain the database and can even make changes when required. If all slots are full then if user login for book the slot then admin inform that parking slots are over.

We will use the Python-Tesseract, or simply PyTesseract, library which is a wrapper for Google's Tesseract-OCR Engine. I chose this because it is completely open-source and being developed and maintained by the giant that is Google. We will also use the Flask web framework to create our simple OCR server where we can upload photos for character recognition purposes. Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file. Flask is also extensible and doesn't force a particular directory structure or require complicated boilerplate code before getting started. We are also going to use an environment consists of an interpreter and any number of installed packages. i.e., the Python extension for VS Code provides helpful integration features for working with different environments. Most people are used to working with programs that provide a graphical user interface (GUI). You can make your Python code accessible to more people by building a GUI for it. Besides those, we'll also use the cv2library which is a fork of the image processing to handle the opening and manipulation of images in many formats in Python. The advantage of web applications is that they're platform independent and can be run by anyone who has access to the Internet. Their code is implemented on a back-end server, where the program processes incoming requests and responds through a shared protocol that's understood by all browsers. Websites had fixed content that was the same for every user who accessed that page.

These web pages are called static because their content doesn't change when you interact with them. When serving a static web page, a web server responds to your request by sending back the content of that page, regardless of who you are or what other actions you took. Such static websites aren't considered applications since

their content isn't generated dynamically by code. While static sites used to make up the entire Internet, most websites today are true web applications, which offer dynamic web pages that can change the content they deliver.

Python-driven web applications use Python code to determine what actions to take and what content to show. Our code is run by the web server that hosts your website, which means that your users don't need to install anything. All they need to interact with your code is a browser and an Internet connection. Serving dynamic content over the Internet involves a lot of different pieces, and they all have to communicate with one another to function correctly. The general process of how content is delivered over the Internet. The programming language used on the server, as well as the technologies used to establish that connection can differ. However, the concept used to communicate across HTTP requests and responses remains the same and is called the HTTP Request-Response Cycle. Python powers many large web applications and is a common choice as a back-end language. Many Python-driven web applications are planned from the start as web applications and are built using Python web frameworks such as Flask. We'll start by developing the Flask back-end layer to serve the results of the OCR engine. From there you can just hit the endpoint and serve the results to the end user in the manner that suits you.

There are four important modules. They are mentioned below:

3.5.1 Password Module

In this section, the user enters password and therefore the portal checks its validity. This section checks the authentication and if the log in credentials is correct then the user can log in to book slot. This module keeps the whole system secure and it doesn't allow any unauthorized users to access the system. This section checks the authentication and if the log in credentials is correct then the user can log in for slot. Sometimes we must need to wait in search of place it becomes a chaotic environment. The security is also not able to perform their work properly due to the rush in the parking area. They become puzzled and cannot concentrate on a single person. The user needs to provide the documents manually; after which the admin needs to keep it very safely so that the data does not get lost. The admin maintain a database where all the complaints are filed and it need to be kept safely

until process is done. This module keeps the whole system secure and it doesn't allow any unauthorized users to access the system.

3.5.2 Search Module

This module helps in searching of the records from the database and also helps in retrieving information from the database whenever necessary. In the Search section, authorized member will search for matching records of database records. The admin update the status of booking slot so that the user gets information of each and everything. They maintain the database and can even make changes when required. The citizens using this system need to get access from admin by registering and log in. The user in the system can go through the information. The user needs to go the apply option for slot booking, after that the user needs to fill all the required data and upload the related image. The citizens can get aware of the slot booking taking place in their society and get aware. The user will be updated the status as process proceeds. The user cannot make any change, as they will only be able to access the portal. This section also helps the admin to update the status of the current investigation so that the user can see which slot is allocated i.e. the user can see the status of the slot only by logging in the website of this system.

3.5.3 Registration Module

In this section, the user has to register first if he or she is new in this website. After successful registration, they get their log in credentials which are provided by the admin. They use the log in credentials to log in for booking a slot in the website. Societies all over the world expand and develop continuously, as such human relationship and activities also expands. The results of dynamism of the growing population can cause traffic. These societies therefore strive to establish and develop institutions that can ensure peace as well as security of lives and property of its citizenry. They become puzzled and cannot concentrate on a single person. The user needs to provide the documents manually; after which the security needs to keep it very safely so that the data does not get lost. Here the admin maintain a register where all the complaints are filed and it need to be kept safely until process is done. The user cannot even see the status of the filled slots and its processing. The user does not get any information of slot booking taking place.

CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

4.1 DESCRIPTION OF DIAGRAM

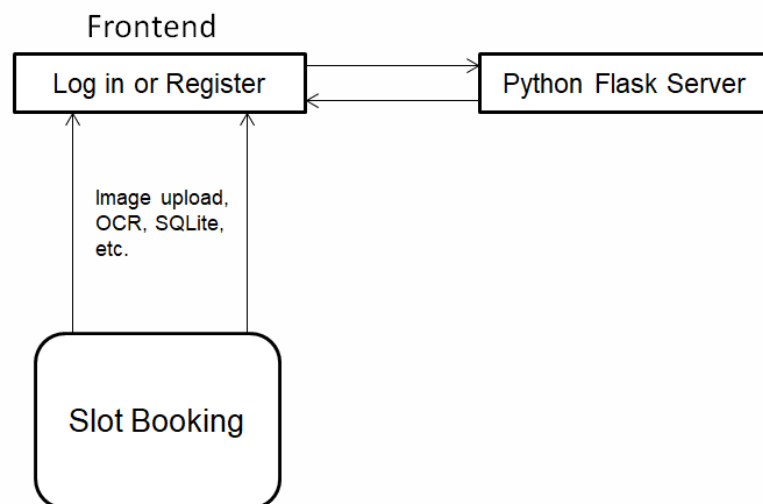


Fig 4.1: BLOCK DIAGRAM

Block diagrams derive their name from the rectangular elements found in this type of diagram. They are used to describe hardware and software systems as well as to represent processes. Block diagrams are described and defined according to their

function and structure as well as their relationship with other blocks. Client, server, and database are the three components shown in the architectural diagram. If a user is new to the system, they must first establish an account before logging in. Here, the server checks for errors, validates the user, and grants portal access. The database is updated with the user information. The user must then provide feedback to the device. The input picture is a licence plate number for a car. After that, the server transforms the image into a bilateral filter to improve image clarity. The machine would then extract text from the image using OCR (optical Character Recognition). If the text matches the current text as it appears in the database, the slot will be assigned to the user. If the user has already slotted, his slot number will be shown. And it is here that the administrator is in charge of the SQLite-based database.

A block diagram is an essential method used to develop and describe hardware or software systems as well as represent their workflows and processes. Block diagrams are used in electronics to represent systems and their shifting. Admin has set up a dashboard that shows which users are assigned to which slots. The admin is also shown and displayed the person's information, pictures, and id. Admins can use SQL queries to extract information about users from this page. And the administrator can make any modifications he wants, such as removing a user or modifying a licence plate number if the user requires it. Block diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. These innovations can ensure that society's law and order is maintained. Technology may be a blessing if we use it correctly, but when it is misused, it causes damage and harm to society. The technology can be used to avoid issues such as traffic and security, as well as to store data. The new method is a conventional one, in which the user must go to a car parking station to park their vehicle. And the data is held in a registry by protection, which contains all of the information about the data and eliminates the need to manage it manually. It's a time-consuming mechanism in which protection collects data one by one.

4.2 ARCHITECTURAL DESIGN

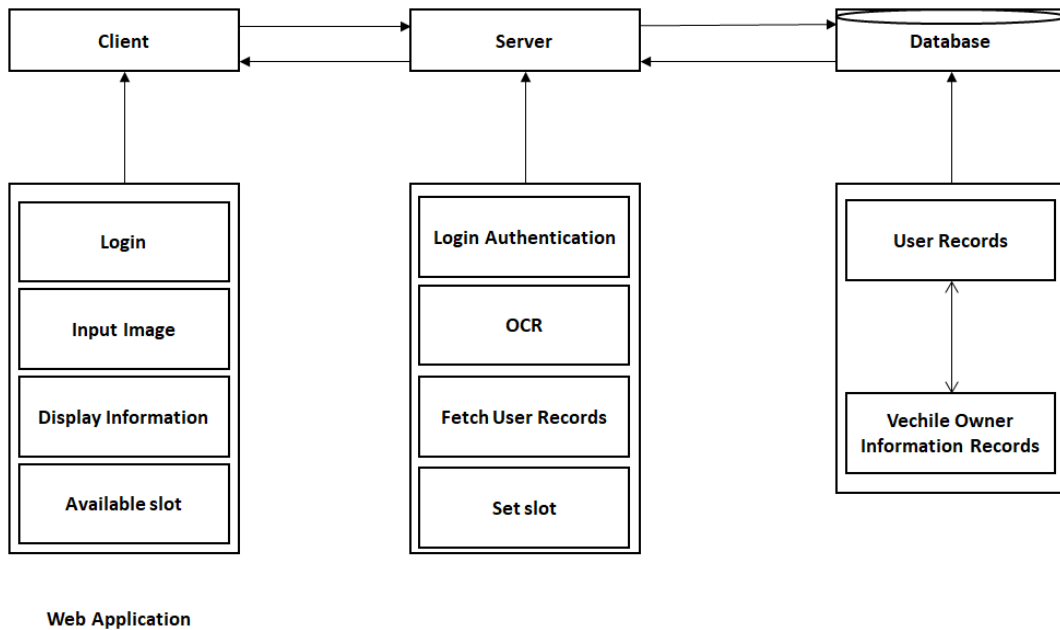


Fig 4.2: ARCHITECTURAL DESIGN

In the Fig 4.1, the closed boxes represent the set of sources and sinks of data. Within the system, we will observe that the user interacts with the appliance through a graphical interface (GUI). The inputs to the proposed system are the license plate image, etc. In this architectural diagram, user has to log in or register in the website in order to book a slot. The user interface contains the log in or registers form. A user can book a slot by filling the form. The proposed system has been divided into two sections: Admin and User.

In this way we can represent complex algorithms or flows of information or communication among individual components within a large system. The above representation is often easier to understand than a textual representation. Here there are three modules in architectural diagram, client, server and database. In the client, if user is new to the system then user has to first register for account and then login into the system. Here the server checks if there are any problem or validate the user and give access to the portal. The user records are uploaded in database. And next step is user has to give input to the system. The input image is vehicle license plate number. Then after uploading, the server takes the image and

converts into bilateral filter to get image clarity. Then the system will use OCR (optical Character Recognition) to get text from the image. If the text is matched with the present text as this is presented in database then it will allocate the slot for the user. If the user has already slot then it will display his slot number. And here the admin handle the database which is built by SQLite.

The proposed system shows the status of the slots and that can be seen by only admin. User can see his slot only and when he login again to book the slot it will show his allocated slot only. Along with the slot booking, the user has to fill the data in form which helps in providing more detail information. The arrows represent the information received or generated by the application. The closed boxes represent the set of sources and sinks of information. In the system, we can observe that the user interacts with the application through a graphical user interface.

In this world of technology the world is reaching to newer heights through innovation and technology day-by-day. Technology has changed the lives of the people and made it easier. So we must take the advantage of the technology and apply it for the well-being of the society. So we have decided to make use of the cutting edge technologies to build this application which will help the system and the users and make them feel secured and comfortable. In our system HTML, Flask framework for routing the URLs, DB browser (SQLite) and cv2 filters has been used to make the application more dynamic and user friendly. Keeping in mind the security of this application we have assigned the authentication and authorization work to the admin who will be able to make changes and provide access to the users, the user need to first register for log-in and then use it.

Admin has provided with a dashboard which helps to see which user is allocated to which slot. Even the details of the persons, images and id are shown and displayed to the admin. Here admin can have a SQL queries to retrieve information regarding users. And admin can make changes whatever he want like deleting the user and changing any license plate number if user need. These technologies will ensure to maintain the law and order in the society. Technology is a blessing if we use it in a right way and misuse of it leads to the destruction and harm to the society. The technology can be used to get away from problems like traffic and security to store data etc. In the current system it is traditional one where the user needs to visit to the car parking station for lodging a vehicle. And the security maintains the data in a

register which contains all the information regarding the data and need to maintain it manually. It's a very lengthy process in which the security does collect data one by one.

The admin of the system can only control, make changes and maintain the databases. The admin provides the required security to the users so that the data does not get deleted or leaked. The citizens using this system need to get access from admin by registering and log in. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data and upload the related document or image. The users can get aware of the slot booking taking place in their society. The user will be updated the status as process proceeds. The user cannot make any change, as they will only be able to access the portal.

In this system we have used DB browser (SQLite) for building the database. The User interface is built by using html. The backend and the main framework of the program are done in Flask framework and cv2 filters. Technology has changed the lives of the people and made it easier. So we must take the advantage of the technology and apply it for the well-being of the society. So we have decided to make use of the cutting edge technologies to build this application which will help the system and the uses and make them feel secured and comfortable. We have made a dashboard which gives each and every information regarding the current number of users registered and how many users booked a slot. All these data can be retrieved by admin using SQL queries. The details of the persons and id are shown and displayed to the admin.

4.3 ACTIVITY DIAGRAM

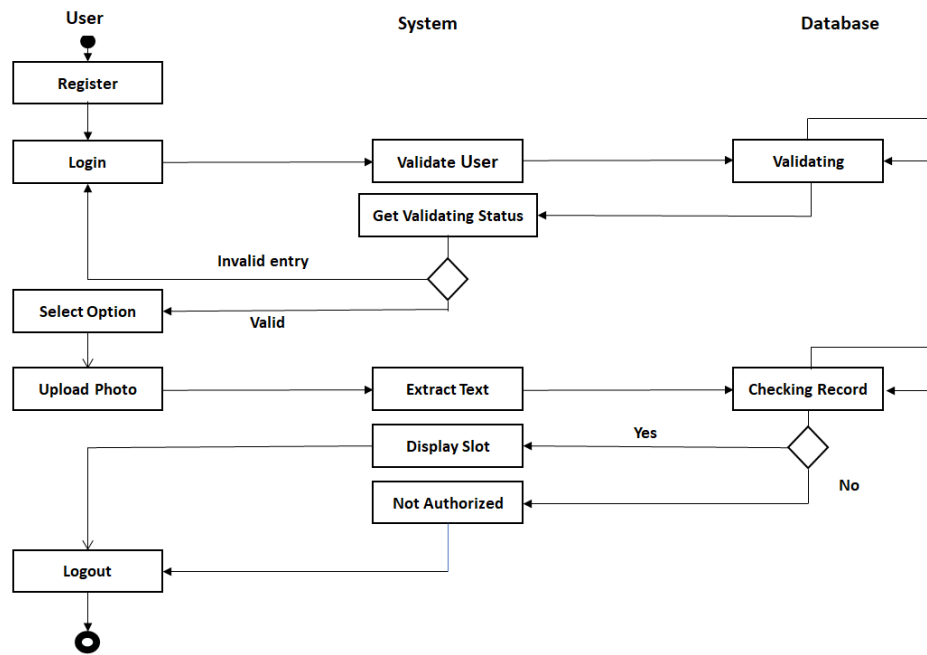


Fig 4.3: Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart. A small filled circle followed by an arrow represents the initial action state of User registration. And then the transitions from one action state to another login state. Thus system validate user that has been sent to database for validating and sends back the acknowledgement as success if the user credentials are correct. If not then the state goes back to login state as the credentials are incorrect. After completing the login state user has to upload license number plate image. We will use the Tesseract OCR an Optical Character Recognition Engine (OCR Engine) to automatically recognize text in vehicle registration plates. Py-tesseract is an optical character recognition (OCR) tool for python. That is, it'll recognize and read the text embedded in images. The extracted text is used to check the same user record if the user vehicle number is matching in database. If matching then the system will display slot through UI. If user record does not match with vehicle number then it shows "license number is not matched and not authorized to park".

We need to upload the image of the license plate of car, so that it can be passed on for further processing and all the further processes take place. This includes converting the image from RGB to grey scale. This is done because differentiating

such images requires less information that is needed to be provided for each pixel. It also reduces the complexity of the image. Another process that is taking place is the removal of noise from the image. Image noise is an undesirable by-product of image capture that influences or obscures the desired information of the image. Removal of such noise from the image improves the quality of image, which will be helpful in further processing.

Each parking slot will be allotted a unique number, which is drawn manually one time, in the parking area. This whole process is done using Optical Character Recognition (OCR) techniques in image processing. The image that has been captured will be processed, so that its quality can be increased. This involves the conversion of the image to grey scale from RGB. It also involves improving the contrast of the image. Processing of the image also involves removal of unwanted noises that might be there in the image. After the processing, we will get a more enhanced and clearer image which can be used for further processing. This is one of the main techniques which is used for optical character recognition system in image processing. Here, it basically involves the process of identifying the available slots in the parking area. DB browser (SQLite) is used to verify whether the parking slot has been occupied by the same person who has booked that specific slot. Usually, there are systems which require the driver to wait in the queue to get a token, the token is used to know how much time the driver has parked his car in the parking area, so on return he needs to submit.

OCR function is used to process the image, which as a result gives the characters present in the image. Pythontesseract is an optical character recognition tool for python. This function takes the threshold image and the configuration file for recognizing digits, as arguments thereby returning a string of characters in the image. These characters will be then search in database. The extracted slots are then assigned to the user. . In result figure shows the extracted available slots from the parking lot using OCR technique.

4.4 EMERGING TECHNOLOGY

In this world of technology the world is reaching to newer heights through innovation and technology day-by-day. Technology has changed the lives of the people and made it easier. So we must take the advantage of the technology and apply it for the well-being of the society. So we have decided to make use of the cutting edge

technologies to build this application which will help this system and the citizens and make them feel secured and comfortable. In our system HTML, SQLite, DB browser, Flask Framework and Optical Character Recognition has been used to make the application more dynamic and user friendly. Keeping in mind the security of this application we have assigned the authentication and authorization work to the admin who will be able to make changes and provide access to the users, the user need to first register for log-in and then use it. We have a dashboard which gives each and every information regarding the number of users, details of users using SQL queries, etc. Even the vehicle plate number along with their slots are shown and displayed to the admin. These technologies will ensure to maintain the law and order in the society. Technology is a blessing if we use it in a right way and misuse of it leads to the destruction and harm to the society.

Technology can be used to get away from the traffic at crowd places etc. The actors in the system are the users, admin and the booking system consisting of form processing, records generation. There are two points to note in understanding how OCR software uses technology to read patterns and features. A simple OCR program may be designed to read and recognize many different fonts and patterns. When any of the compatible patterns are presented to the software, it uses technology to convert it to machine-readable text accurately so it becomes searchable. The Immense Impact of OCR can make life easier in many ways.

Printed material can be easily edited once it is in digital form. Searching through books for a specific passage or even a single word simply aren't practical solutions when working with printed matter. A computer can do this in seconds when the data is stored as text. It then becomes searchable. For research purposes, OCR is an incredible tool that has simply revolutionized this field of work.

OCR is a computer technology that helps machines to understand the text in a given sample. The sample could be an image, a written or a printed document. OCR is mostly used to differentiate between any printed or handwritten text characters inside a digital image of physical texts such as a scanned document or a vehicle registration plate. It is one of the earliest addressed computer vision tasks and when the data is normal and simplified only normal algorithms are sufficient. However, for large and complex datasets, deep learning is required. There are a lot of areas where OCR plays a vital role. The data available nowadays is in terabytes &

petabytes and it requires deep learning to analyse such large data for valuable outputs. OCR functions on a very basic process that involves examining the text of any particular document and then translating the characters into code to be used for data processing

In businesses, OCR technology brings about significant solutions in efficiency. Statistically, of the 12 minutes required by an employee to create and process a document, 9 minutes are spent in locating and getting it ready for use. A document scan that is already in readable form in the computer can be accessed with a few clicks in seconds, improving productivity immensely.

Enhanced accessibility is another advantage of OCR. A book may be accessible to those who are in physical proximity to it. However, material that is in digital form can be accessed by or sent to people located anywhere in the globe. What's more, many users have the rights to access the same document at the same time.

Safeguarding vital documents becomes much easier when the material is in digital form. Books, manuscripts, and files may all be damaged beyond repair due to a myriad of unforeseeable reasons—water leakage, fire, poor handling, etc. Having these documents in a digital format allows you to store them in more than one safe location as backup solutions so that critical data is never permanently lost or damaged beyond use.

Lastly, substantial savings can be achieved by opting for OCR and paperless document management. Using advanced technology to convert and scan print data into digital images lets you avoid spending on filing, archiving, and storage. There is no need for supplies such as paper, ink, or files to store these documents, neither is there any need for staff members to carry out these tasks. The costs saved here can work out to thousands of dollars over a year.

In order to improve the operational efficiency for the public, Smart City uses information, connectivity and technology to help accelerate towards improving the quality of life for people. Internet of Things (IoT), Automation, and Machine Learning are the new technologies driving the implementation of smart cities. Any city can be considered a smart city initiative by introducing a system such as smart parking using a web based applications to help drivers locate parking slots, smart traffic management track and analyse traffic flows, sharing information electronically,

monitoring changes to the environment enabled sanitation, etc. The advancement of technology has led to a lot of changes in the world. But these changes haven't affected the parking system which is still the same and time consuming. It takes a lot of effort and time to find a vacant parking slot. It is known that there is a huge problem of traffic especially in countries where population is high. If population is high, more people are there to park in a limited number of parking slots. Intelligent parking schemes have been significantly improved in urban environments and the associated technologies. Some types of car parking system are also required wherever traffic is relevant. Rapid population growth in the urban areas and increased traffic congestion has increased smart parking demand. This proposal aims to implement a reservation based automated car parking system to address the rising problems faced due to traffic congestion. The identification of vacant spots is implemented based on image processing techniques thereby overcoming the disadvantages of conventional sensor based methods. The reservation of slots through mobile application can help drivers locate parking slots with minimum effort and in less time. The user authentication is ensured by proper verification of details implemented by database.

CHAPTER 5

RESULTS AND DISCUSSION

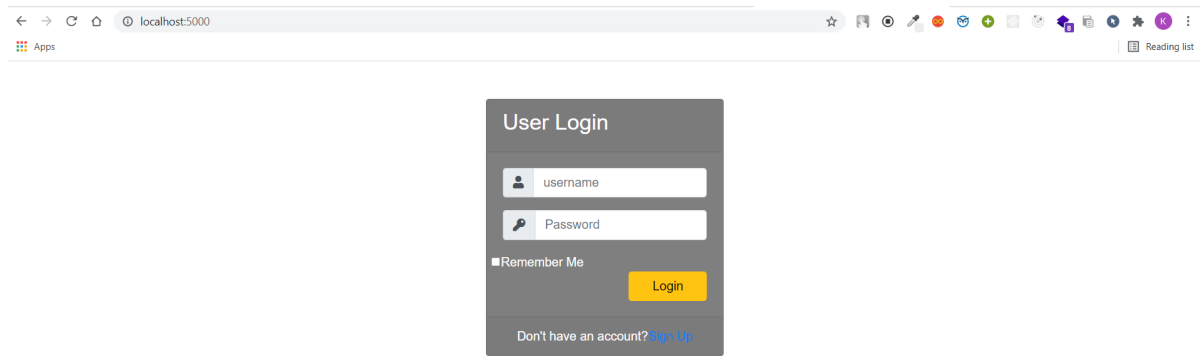


Fig 5.1: Log-In Page

The Fig 5.1 is the log-in page which generally is the entry or authentication of every secured portal. This portal can be logged in by the user. The admin can check the working of the portal and to maintain the database. The user can log-in to access the portal and to book the slot. The admin provides the required security to the users so that the data does not get deleted or leaked. This section checks the authentication and if the log in credentials is correct then the user can log in and book the slot. It is the frontend of the project or it can also be termed as user interface. Here the user gets the multiple options to execute or access their task as per need. This page will appear once the user or the admin make a log-in into the portal. It is the frontend of the project or it can also be termed as user interface. The admin of the system can only control, make changes and maintain the databases. This module keeps the whole system secure and it doesn't allow any unauthorized users to access the system.

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/reg'. The browser's toolbar includes back, forward, refresh, and home buttons, along with various extension icons. On the left side of the page, there is a blue 'Home' button. The main content area features a 'Registration' form with the following fields: 'Username', 'Password', 'Enter your phone number', 'Email', a dropdown menu currently showing 'Two Wheeler', 'Vehicle Number', 'Account number', and 'Enter IFSC'. A green 'Submit' button is located at the bottom of the form.

Fig 5.2: Registration Form

The Fig 5.2 shows that the user has to register first if he or she is new in this website. After successful registration, they get their log in credentials which are provided by the admin. They use the log in credentials to log in and to book the slot in the website. Societies all over the world expand and develop continuously; the results of dynamism of the growing population give room for traffic. The user interface contains the log in or registers form. This portal can be logged in by both admin and the user. The admin can log-in to check the working of the portal and to maintain the database. The user can log-in to access the portal and to booking a slot. The admin provides the required security to the users so that the data does not get deleted or leaked. It is the frontend of the project or it can also be termed as user interface. The admin of the system can only control, make changes and maintain the databases. This section checks the authentication and if the log in credentials is correct then the user can log in.

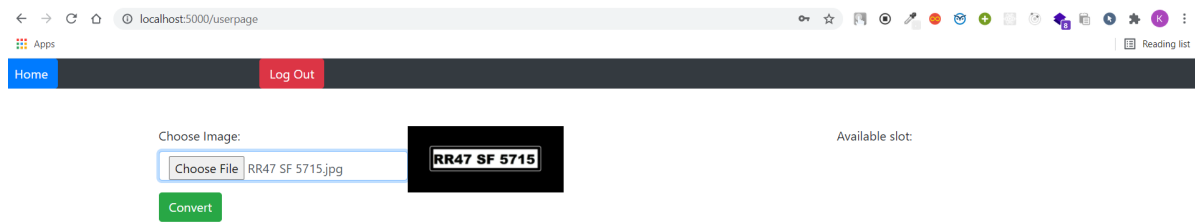


Fig 5.3: User Uploading Image

The Fig 5.7 is the information which is uploading the image in the portal. So that after uploading the convert button which is in green colour is being clicked by the user. The uploaded image is user car license number plate. After this, using OCR the text in the image is being extracted and search in the database whether the slots are free to park. In the block diagram, user has to log in or register in the website in order to book a slot. The user interface contains the log in or registers form. The users can fill the form and after login upload Image in the portal to book a slot. This portal can be logged in by both admin and the user. If the user login to book the slot then the slot number and the user vehicle license number are view by the admin in this dashboard. The admin can log-in to check the working of the portal and to maintain the database. The user log-in is to access the portal and slot booking. The admin provides the required security to the users so that the data does not get deleted or leaked. The user interface contains the log in or registers form. The admin of the system can only control, make changes and maintain the databases. The admin provides the required security to the users so that the data does not get deleted or leaked.

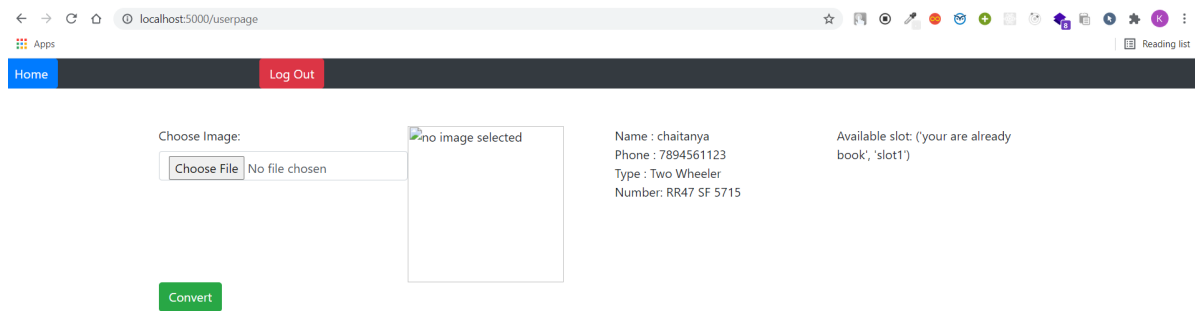


Fig 5.4: Display slot Page

The Fig 5.7 is the information came from the database after uploading the vehicle license plate image by the user. So that after uploading the convert button which is in green colour is being clicked by the user. The uploaded image is user car license number plate. After this, using OCR the text in the image is being extracted and search in the database whether the slots are free to park. In the block diagram, user has to log in or register in the website in order to book a slot. The user interface contains the log in or registers form. The users can fill the form and after login upload Image in the portal to book a slot. The extracted text from image is search in database and assigns a slot and also displays user information. This portal can be logged in by both admin and the user. If the user login to book the slot then the slot number and the user vehicle license number are view by the admin in this dashboard. The admin can log-in to check the working of the portal and to maintain the database. The user log-in is to access the portal and slot booking. The admin provides the required security to the users so that the data does not get deleted or leaked. The user interface contains the log in or registers form. The admin of the system can only control, make changes and maintain the databases. The admin

provides the required security to the users so that the data does not get deleted or leaked.

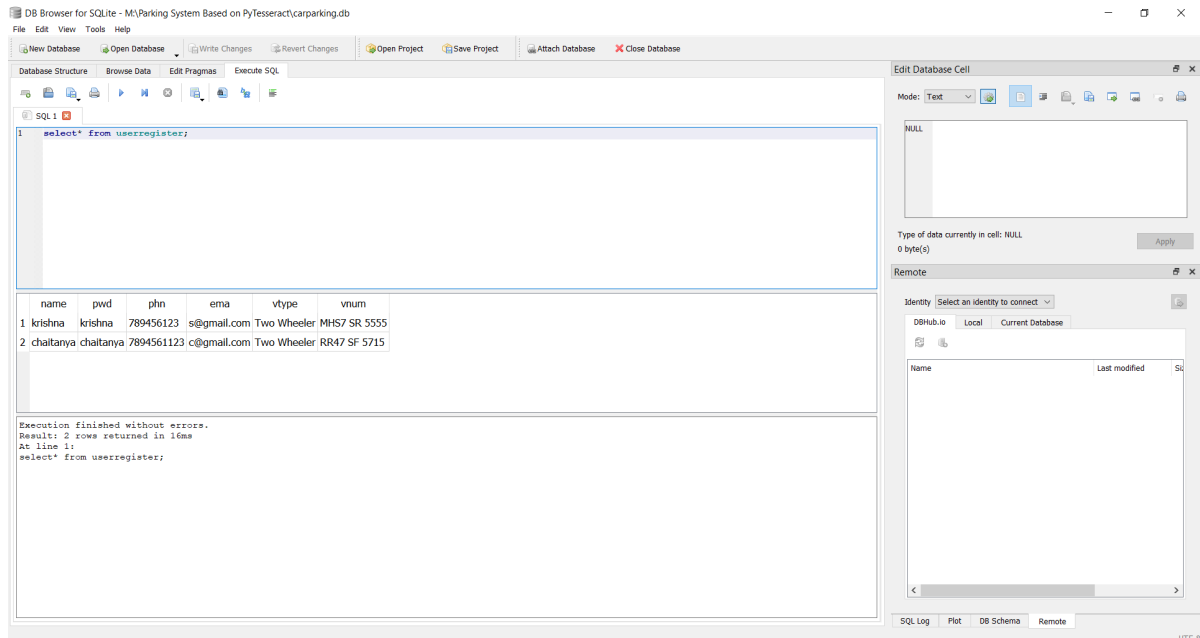


Fig 5.5: User Register Dashboard

The Fig 5.9 is all about the dashboard which can only be accessed by the admin. Here the admin can observe and can have a clear idea of the number of users, details of users, etc. The user needs to go the apply option for booking the slot, after that the user needs to fill all the required data. All the data is shown in this dashboard or DB browser (SQLite). The user cannot make any change, as they will only be able to access the portal. All these data can be retrieved by admin using SQL queries. Here all the details which were being filled by the user that are being uploaded can be accessed by the admin to take further actions. The admin can change data by the contact of user and give inform to the users for the further processing.

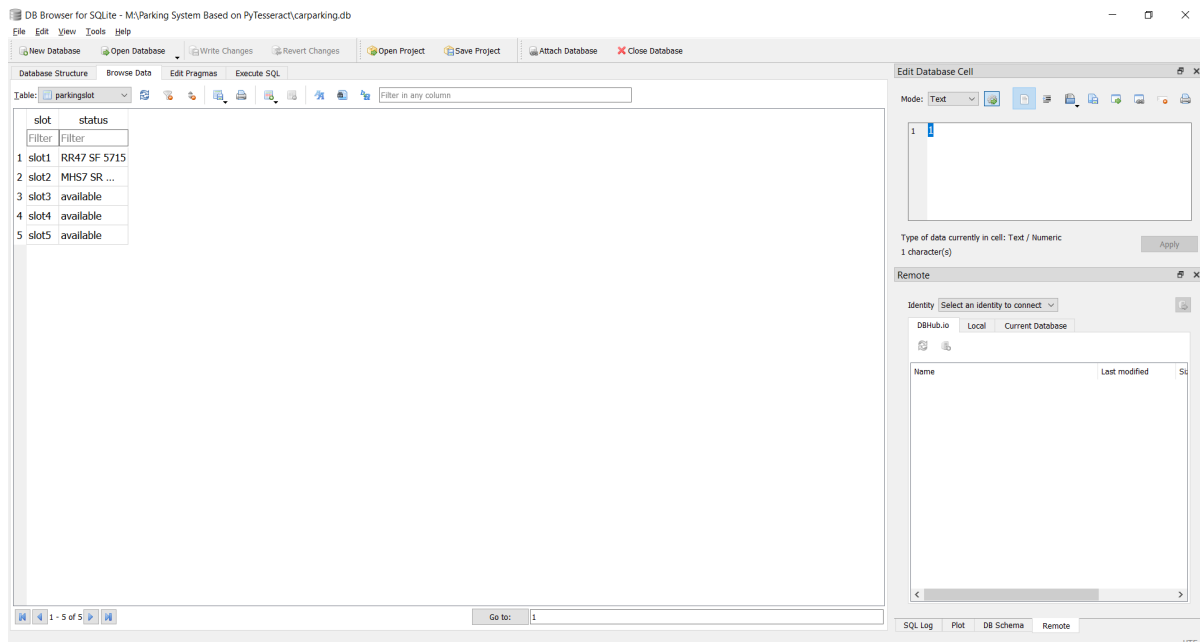


Fig 5.6: Parking Slot Dashboard

The Fig 5.7 is the information which is being booked by the user in portal. In the block diagram, user has to log in or register in the website in order to book a slot. The user interface contains the log in or registers form. The users can fill the form and after login upload Image in the portal to book a slot. This portal can be logged in by both admin and the user. But here this is the dashboard which means only admin can open the database. If the user login to book the slot then the slot number and the user vehicle license number are view by the admin in this dashboard. The admin can log-in to check the working of the portal and to maintain the database. The user log-in is to access the portal and slot booking. The admin provides the required security to the users so that the data does not get deleted or leaked. The user interface contains the log in or registers form. The admin of the system can only control, make changes and maintain the databases. The admin provides the required security to the users so that the data does not get deleted or leaked.

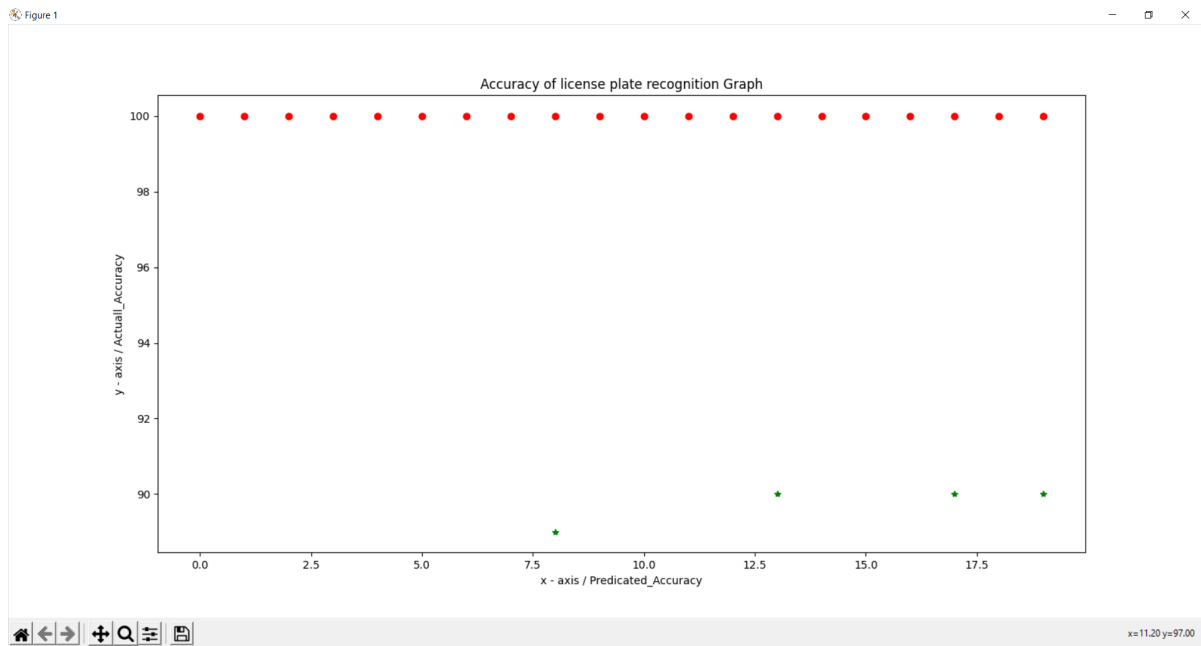


Fig 5.7: Accuracy Graph

To graphing in python with Matplotlib, which is arguably the most popular graphing and data visualization library for Python.

- Define the x-axis and corresponding y-axis values as lists.
- Plot them on canvas using `.plot ()` function.
- Give a name to x-axis and y-axis using `.xlabel ()` and `.ylabel ()` functions, x-axis as predicted accuracy and y-axis as actual accuracy
- Give a title to your plot using `.title ()` function.
- Finally, to view your plot, we use `.show ()` function.

```
C:\Windows\System32\cmd.exe - python test.py
M:\Parking System Based on PyTesseract>python tes.py
python: can't open file 'tes.py': [Errno 2] No such file or directory

M:\Parking System Based on PyTesseract>python test.py
Actual License Plate      Predicted License Plate      Accuracy
-----
AB12CDE                  AB12CDE                      100
ABC123                   ABC123                       100
AP12JK5327              AP12JK5327                  100
AP04AS4321              AP04AS4321                  100
BR05E5000               BR05E5000                   100
CAG12345                 CAG12345                    100
DL54RH5711              DL54RH5711                  100
DL88255454              DL88255454                  100
HR33C0001               HR33C0001                   89
HR45124689              HR45124689                  100
KL25FJ4848              KL25FJ4848                  100
KL55MJ6713              KL55MJ6713                  100
KR77FG1171              KR77FG1171                  100
MH57SR5555              MHS7SR5555                  90
MH77UU7722              MH77UU7722                  100
MP77RF2121              MP77RF2121                  100
RR47SF5715              RR47SF5715                  100
TN22UA3517              TN22UA3517                  90
TN44TE1155              TN44TE1155                  100
TS1IHJ7732              TS1IHJ7732                  90
Predicted_Accuracy list : [100, 100, 100, 100, 100, 100, 100, 100, 89, 100, 100, 100, 100, 100, 90, 100, 100, 100, 90, 100, 90]
Actual_Accuracy list : [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100]
```

Fig 5.8: License Plate Accuracy List

To view the data and prediction we do a bit of visualization as shown above. we are also calculating the accuracy of prediction without using any built-in function.

We see that the Tesseract OCR engine mostly predicts all of the license plates correctly with 100% accuracy. For the license plates, the Tesseract OCR Engine predicted incorrectly (i.e. HR33C0001, MH57SR5555, and TN22UA3517).

Applying the image processing techniques would increase the accuracy of the Tesseract Engine for the license plates.

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION

As conclusion, this system will make the system of booking the slots more efficient and flexible for the users. The users no need to worry about the security and the confidentiality of the documents submitted. And for the admin he needs not to worry about the information to be stored because it is all maintained in a database and can be edited or retrieved whenever needed. The objective of system have been achieved, it saves user time in search of parking space available in such a long parking area. We have been able to scan images and extract text from them. This is Optical Character Recognition and it can be of great use in many situations. Project is aimed at making the system more cost effective and user friendly. Future works will extend the system to provide a graphical view of the parking spaces to confirm their slot and other different technologies such as interlink with smart phones.

FUTURE WORK

In future, system will provide mail facility to members. A successful implementation of this project would result in less traffic in crowded parking spaces like malls and business buildings where many people share a parking area. Future works will extend the system to provide a graphical view of the parking spaces to confirm their slot and communication between admin and users thus increase the efficiency of the parking system.

REFERENCES

1. Tejas, B., Omkar, D., Rutuja, D., Prajakta, K., & Bhakti, P. (2017, June). Number plate recognition and document verification using feature extraction OCR algorithm. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1317-1320). IEEE.
2. Kaur, E. K., & Banga, V. K. (2013). Number plate recognition using OCR technique. *International Journal of Research in Engineering and Technology*, 2(09), 286290.
3. Qadri, M. T., & Asif, M. (2009, April). Automatic number plate recognition system for vehicle identification using optical character recognition. In *2009 International Conference on Education Technology and Computer* (pp. 335-338). IEEE.
4. Jain, K., Choudhury, T., & Kashyap, N. (2017, February). Smart vehicle identification system using OCR. In *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)* (pp. 1-6). IEEE.
5. Bansal, S., Gupta, M., & Tyagi, A. K. (2020, July). A Necessary Review on Optical Character Recognition (OCR) System for Vehicular Applications. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 918-922). IEEE.
6. Jondhale, A., Das, G., & Sonavane, S. (2015, January). OCR and RFID enabled vehicle identification and parking allocation system. In *2015 International Conference on Pervasive Computing (ICPC)* (pp. 1-4). IEEE.

APPENDIX

A. SOURCE CODE

```
from flask import Flask,url_for,request,render_template,redirect,session
import sqlite3
from werkzeug.utils import secure_filename
import os
import pytesseract
from PIL import ImageFilter
from PIL import Image
from cv2 import cv2

import datetime

app=Flask(__name__)
app.config['UPLOAD_FOLDER'] = './media'

app.secret_key="Rudy"

vnumlist=[]
parkingslot=[]
slotname=None
slotnamelist=[]
pytesseract.pytesseract.tesseract_cmd=r"C:\Program Files\Tesseract-OCR\tesseract.exe"
@app.route('/',methods=["GET","POST"])
def login():
    username=None
    password=None
    err="Invalid username and password"
    if request.method=='POST':
        session['username']=request.form['uname']
        session['vid']=request.form['pwd']
        username=request.form['uname']
        password=request.form['pwd']
        conn = sqlite3.connect("carparking.db")
        r=conn.cursor()
        r.execute("select name,pwd from userregister where name=? and pwd=?",
(username,password))
        rows=r.fetchall()
        if username=='Admin' and password=='Admin':
            return redirect(url_for('Admin'))
        else:
            if len(rows)!=0:
                for i in rows:
                    if i[0]==username and i[1]==password:
                        return redirect(url_for('userpage'))
                    #return render_template('user.html',username=username)
            else:
```

```

        return render_template('login.html',err=err)
    else:
        return render_template('login.html',err=err)
    return render_template('login.html')

@app.route('/reg',methods=["GET","POST"])
def reg():
    uname=None
    pwd=None
    phn=None
    ema=None
    vtype=None
    vnum=None
    slot=None

    msg="Register successfully"
    errmsg = "Please check your car number and register again"
    if request.method=='POST':
        uname=request.form['txt']
        pwd=request.form['passw']
        phn=request.form['phn']
        ema=request.form['ema']
        vtype=request.form['vtype']
        vnum=request.form['vnum']
        # slot='avaliabile'
        try:
            import sqlite3
            table_name = 'userregister'
            conn = sqlite3.connect("carparking.db")
            c = conn.cursor()
            c.execute('create table if not exists ' + table_name + ' (name varchar(50),pwd varchar(50) primary key,phn varchar(50),ema varchar(50),vtype varchar(50),vnum varchar(50))')
            c.execute('insert into '+table_name+' values (?,?,?,?,?,?)',(uname,pwd,phn,ema,vtype,vnum))
            conn.commit()
            conn.close()
            #flash("Register successfully")
            return render_template('registration.html',msg=msg)
        except:
            return render_template('registration.html',msg=errmsg)
        # return render_template('registration.html')
    else:
        return render_template('registration.html')
    return render_template("registration.html")

@app.route('/userpage',methods=["POST","GET"])
def userpage():
    err="Your not authorized to park here."
    sloterr="Parking slot was full"

```

```

if request.method=='POST':
    image = request.files['ocrImage']
    text = ''
    filename = secure_filename(image.filename)
    image.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
    img = Image.open(os.path.join(app.config['UPLOAD_FOLDER'], filename))
    '''img = os.path.join(app.config['UPLOAD_FOLDER'], filename)

    dummy = cv2.imread(img)
    cv2.imshow("dummy image", dummy)
    cv2.waitKey(0)
    # rgb to gray scale conversion
    gray = cv2.cvtColor(dummy, cv2.COLOR_BGR2GRAY)
    cv2.imshow("Gray scale conversion", gray)
    cv2.waitKey(0)
    # noise removal
    gray = cv2.bilateralFilter(gray, 11, 17, 17)
    cv2.imshow("bilateral filter", gray)
    cv2.waitKey(0)'''
    # print(os.path(img))
    text = pytesseract.image_to_string(img)
    print(text)
    x = session["vnumm"]=text.strip()
    print(x)
    conn=sqlite3.connect("carparking.db")
    r=conn.cursor()
    l=conn.cursor()
    l.execute("select name,phn,vtype,vnum from userregister")
    user =l.fetchall()
    for name in user:
        print(name)
    r.execute("select name,phn,vtype,vnum from userregister where vnum=?",
(session["vnumm"],))
    # ss.execute("select slot from parkingslot where status=?",("available",))
    # slotrow=ss.fetchall()
    # if len(slotrow)!=0:
    #     for i in slotrow:
    #         all_slot.append(i)
    # r.execute("select name,phn,vtype,vnum from userregister ")
    rows=r.fetchall()

    print('hi')
    print(rows)
    num = []
    if rows == []:
        print('wrong number plate')
        book = 'your number plate is in correct'
        return render_template("user.html",message = book,rows=rows)

```

```

v_num = rows[0][3]
# print(num)
li=[]
if len(rows)!=0:
    print('hi')
    if v_num != x :
        print('your number plate is in correct')
        message = 'your number plate is in correct'
        return render_template("user.html",message = book,rows=rows)

    elif rows[0][3]==session["vnumm"]:
        ss=conn.cursor()
        ss1=conn.cursor()
        ss.execute("select slot,status from parkingslot ")
        slotrow=ss.fetchall()

        for i ,j in slotrow:
            if v_num == j:
                print('your already booked in ',i)
                book = 'your are already book',i
                return render_template("user.html",message = book,row
s=rows)

                break
            else:
                if j == 'available':
                    print(i)
                    sql="""update parkingslot set status = ? where sl
ot = ?"""

                    ss1.execute(sql,[v_num,i])
                    conn.commit()
                    book = 'your allocated slot number',i
                    return render_template("user.html",message=book,r
ows=rows)

                    break
                continue

        else:
            return render_template("user.html",rows=rows,err=sloterr)

    else:
        print("noo")
else:
    return render_template("user.html",err=err)

return render_template('user.html',text=session["vnumm"])
else:
    return render_template("user.html")
return render_template('user.html')

```

```
if __name__ == '__main__':  
    app.run(debug=True)
```