PARKING MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

AKSHI VISHNOI
(2100290140018)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr Sangeeta Arora
Associate professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(APRIL 2023)

DECLARATION

I hereby declare that the dissertation entitled Parking Management System submitted for the MCA Degree is my original work and the dissertation has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles.

Sign _____

Name of Candidate:

Akshi Vishnoi

University Roll No.:

2100290140018

CERTIFICATE

Certified that **Akshi Vishnoi (2100290140018)** has carried out the project work having "**PARKING MANAGEMENT SYSTEM**" for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: ……

Akshi Vishnoi (2100290140018)

…….

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr Sangeeta Arora
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The project (confidential) brings together industry-leading technology and expertise in the parking sector that benefits both clients and motorists. The platform primarily focuses on enabling users' extensive functionality for parking management, enforcement, and security. Alongside providing the ability for precise facility management ranging from an entire Estate down to individual cameras and devices. The application also utilizes the feed from ANPR cameras to provide greater insight into vehicle data and customer behavior.

The main aim of this project is to reduce the traffic in the parking place. Normally we can see in the multiplexes, cinema halls, large industries, and function halls there is problem they have to go and search which line is empty and which line having place to park the vehicle, for parking then they need workers for parking in correct position it is the money consumed process. So, to avoid this problem the Car Parking System project is implemented. So, the traffic can be reduced in the parking place of the theatres, multiplex, and in large industries and in commercial places.

The front-end of the application is developed on a JavaScript framework (ReactJs) and the back-end is a server-less architecture hosted on GCP and Firestore is used as database storage.

Keywords:

1. Industry leading technology
2. ANPR Camera
3. Traffic

ACKNOWLEDGEMENTS

Success in life is never attained single handed by. My deepest gratitude goes to my thesis supervisor, **Dr Sangeeta Arora** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr Arun Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Akshi Vishnoi

TABLE OF CONTENTS

List of Chapters

LIST OF FIGURES

`

Chapter 1

Introduction

With the rapid development of the economy, vehicles have become an indispensable tool in people's daily lives. However, solving the "difficult parking" problem is now an emerging problem.

Monitoring the status of parking spaces is the most fundamental prerequisite for modern intelligent parking management and guidance systems. In order to obtain accurate parking status information, magnetic field detection technology is used to detect magnetic changes of parking spaces in real time, because ferromagnetic materials are easily magnetized in a magnetic field, and the Magnetized ferromagnetic objects (such as vehicles) can change the surrounding geomagnetic field

The contribution of our work is threefold.

1. We have selected an efficient and innovative sensor to detect the parking state, which can be further applied in the design of an intelligent parking system.
2. We propose a simple but very effective detection algorithm for this parking system.
3. We are carrying out a thorough experimental study of our proposed parking system. Ratings include accuracy, robustness, sensitivity, stability, and more.

Fig.1.1 Parking Management Model

1.1 Objective

The aim of implementing a Parking Management System is to reduce time and increase efficiency of the current Parking Management System. In overpopulated cosmopolitan zones, parking strategies must be well implemented for management of vehicles. The system provides details of the vacant parking slots in the vicinity and reduces the traffic issues due to illegal parking in the vicinity. It is designed with an objective to meet the requirements of controlled parking that offers effortless parking tactics to the authorities.

1.2 Problem Statement

The project (confidential) brings together industry-leading technology and expertise in the parking sector that benefits both clients and motorists. The platform primarily focuses on enabling users' extensive functionality for parking management, enforcement, and security. Alongside providing the ability for precise facility management ranging from an entire Estate down to individual cameras and devices. The application also utilizes the feed from ANPR cameras to provide greater insight into vehicle data and customer behavior.

After the completion of formal training given to interns, I was asked to go through the existing projects and understand the structure and their architecture. From the beginning of August, I was mapped to this project and an overview was given by my colleagues on the working and architecture of the project. Later after the understanding of the project was completed, I was asked to design manual test cases for various modules of the application.

1.3 Tools and Technologies used

When I joined the project, it was already live. All the tools and technologies were already defined and I have to work on them. Mentioned below are the tools and technologies that are being used in our project.

1.3.1 Jira

Jira is a suite of agile work management solutions that powers collaboration across all teams from concept to customer, empowering you to do the best work of your life, together. Jira offers several products and deployment options that are purpose-built for Software, IT, Business, Ops teams, and more. Read on to see which is right for you.

Jira helps teams plan, assign, track, report, and manage work and brings teams together for everything from agile software development and customer support to start-ups and enterprises. Software teams build better with Jira Software, the #1 tool for agile teams. Deliver amazing service experiences across all teams from IT, Dev, Ops, and more with Jira Service Management. Business teams can unlock the power of agile and collaborate better with Jira Work Management. Jira Align is an enterprise agile planning platform that connects work at scale. With templates and solutions crafted for every team and Jira as your common language - work moves fluently and transparently across your organization.

1.3.2 Selenium with Python

Selenium Python bindings provides a simple API to write functional/acceptance tests using Selenium Web Driver. Through Selenium Python API you can access all functionalities of Selenium Web Driver in an intuitive way. Selenium Python bindings provide a convenient API to access Selenium Web Drivers like Firefox, i.e., Chrome, Remote, etc. The currently supported Python versions are 3.5 and above.

1.3.3 Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter. On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware.

1.3.4 Postman

Postman is an application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated.

- Methods Postman offers many endpoint interaction methods. The following are some of the most used, including their functions:
    - GET: Obtain information
    - POST: Add information
    - PUT: Replace information
    - PATCH: Update certain information
    - DELETE: Delete information
- Response Codes when testing APIs with Postman, we usually obtain different response codes. Some of the most common include:
    - 100 Series: Temporal responses, for example, ‘102 Processing’.
    - 200 Series: Responses where the client accepts the request and the server processes it successfully, for instance, ‘200 Ok’.
    - 300 Series: Responses related to URL redirection, for example, ‘301 Moved permanently.’
    - 400 Series > Client error responses, for instance, ‘400 Bad Request’.

    - 500 Series > Server error responses, for example, ‘500 Internal Server Error.’
- Collections Postman gives the possibility to group different requests. This feature is known as ‘collections’ and helps organize tests. These collections are folders where requests are stored and can be structured in whichever way the team prefers. It is also possible to export-import them.
- Environments Postman also allows us to create different environments through the generation/use of variables; for example, a URL variable that is aimed towards different test environments (dev-QA), enabling us to execute tests in different environments using existing requests.

1.3.5 Firestore

Google Firestore is one of the standout cloud database services preferred by a large number of businesses today. It facilitates advanced data management and real-time functionality for comprehensive application development. Read on to gain a detailed insight into Google Firestore features and advantages.

It enables users to avail themselves options of Unity, Java, C++, Go, and Node.js SDKs and offers support for REST and RPC APIs. The Firestore database enables automatic scaling, enhanced performance, ease-of-use, and also provides a high level of reliability. Firestore helps to sync data across multiple client applications with the use of real-time listeners. It uses the Cloud Identity, and Access Management features from Google for the process of authentication. Firestore performs data storage in the form of documents, with the documents being stored in collections.

1.4 Hardware & Software Requirement:

Hardware Interfaces Requirement

- Processor: i3 11^{th} Gen
- RAM: 8 GB
- Operating System: Windows 11
- System Type: 64-bit operating system
- Hard disk: 256 GB

Software Interfaces Requirement

- Eclipse
- MySQL
- VS CODE

- Apache Tomcat Server

All these types of software automatic configure inside operating system after installation it is having Java, MYSQL, Apache and operating system base configuration file, it doesn’t need to configure manually.

Chapter 2

Feasibility Study

The iterative process begins with a simple implementation of the required software set and re-installs the changes until the entire process has been implemented. With each iteration, innovations are made and new features are added.
Iterative and incremental development is a design, or a combination of iteration and incremental software development design. This combination has been around for a long time and is widely viewed as a powerful growth factor. For example: "During software development, multiple changes can occur simultaneously in the software development cycle." and "This process can be described as an evolutionary" or "gradual development" approach. The relationship between iteration and increment is determined by the entire software development method and software development process. The number and specific nature of the content of incremental design and iterations will be specific to each individual's development.

The lifecycle model does not try to start with needs met. Instead, development begins by introducing and using only a part of the software that can be analyzed to make further decisions. This process is then repeated, creating a new piece of software for each cycle of the model.

2.1 Operational Study

A defined project is only useful if it can be translated into information that meets the needs of the organization. Simply put, this type of efficiency asks whether the system will work once designed and installed. What are the main topics for the application? The question here Technical Feasibility will help assess the effectiveness of Technical Feasibility.

2.2 Technical Feasibility

Technical Feasibility centers on the existing computer system (hardware, software, etc.) and to what extent they can further support demand. For example, if the computer is currently running at 80% capacity (upper limit), running other applications may overload the system or require additional hardware. This includes financial decisions

to support technological development. If the budget is severely constrained, the project is considered unfeasible.

2.3 Economic Feasibility

Businesses may try to weigh the cost of developing and implementing a new system against the benefits of using the new system. This feasibility study provides senior management with a financial investment for the new project. In such cases, a simple business analysis with a realistic comparison of costs and benefits makes more sense. Additionally, this has proven to be a useful tool for comparing actual costs as the project progresses. It can have all sorts of useless benefits to thinking about automation. These can include improved customer satisfaction, improved operational accuracy, better data and storage, faster data retrieval.

2.4 Schedule Feasibility

Schedule Feasibility means that the project can be completed on time. The project does not have a deadline but according to the proposed system the development process is on schedule. Therefore, it is feasible.

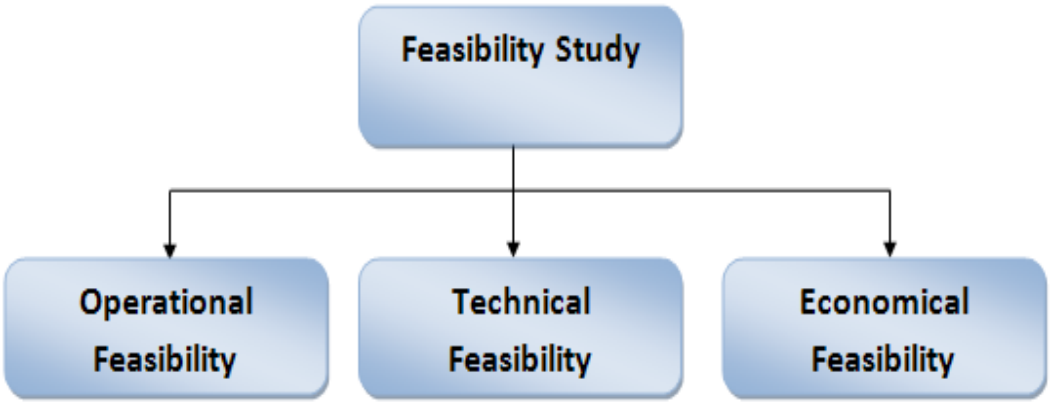

Fig. 2.1 Feasibility Study

Chapter 3

System Analysis Phase

3.1 User Requirements

- Need for an application that makes communicating easy and comfortable.
- An application that enables users to park a vehicle safely and securely.
- Need for an application that is easy to use and widely available and hence a web application.
- Handling all functions done with organization in a computerized manner.
- Allowing the user to park the vehicle directly

3.2 Functional Requirements

3.2.1 Performance Requirements

- **Flowless Payment –** User can make flowless payment with hight secure gateways.
- **Notifications –** User gets notified for each status of the permit application and payment status via email.
- **Debugging –** Easy debugging for unidentified error through Firestore log explorer.
- **User Satisfaction**: The system is such that it stands up to the user's expectations.
- **Response Time**: The response of all operations (Permit page, Permit list page load time) is good.
- **Multiple Roles** – Platform has multiple roles (Admin, End Users) to keeps user permissions discrete.
- **Error Handling**: Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- **Safety and Robustness**: Data stored as hash value to avoid data leakage to make sure user and payment details are secured.
- **Portable**: The software should not be architecture specific. It should be easily transferable to other platforms if needed.

- **User Friendliness**: The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

3.2.2 Design Constraints

- **Standard Compliances:** This specifies the requirement for standards the system must follow. The standards may include the report format and accounting properties.
- **Hardware Limitations:** Hardware limitations can include the types of machines to be used, operating system available on the system, language support and limits on primary and secondary storage.
- **Reliability and Fault Tolerance:** Fault tolerance requirements can be placing a constraint on how the system is to be designed. Recovery Requirements are often an integral part here, detailing what the system should do if some failure occurs to ensure certain properties.

3.3 Non-Functional Requirements

- **Security Requirements and Privacy**: Application has 20 min inactivity auto-logout feature to provide security and keep data secure.
- **Availability**: The Application should be available all the time i.e., the user can access it using a web browser, only restricted by the down time of the server on which system runs. A customer friendly system which is in access to the people around the world should work 24 hours.
- **Performance**: The Permit list page loading time is found to be ½ seconds for 10 permits, that make it faster and more reliable. It must be able to perform in adverse conditions like; slow internet speed, low memory and RAM on the device, low battery and should provide uninterrupted connections and must have a high data transfer rate.
- **Maintainability**: In case of a failure, a re-initialization of the system will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.
- **Supportability**: The code and supporting modules of the system will be well documented and easy to understand.
- **Operational Requirements**: The application must work on all mobile and tablet devices as well. The user interface must be consistent on all devices

Chapter 4

Project Module's

4.1 Estate Management

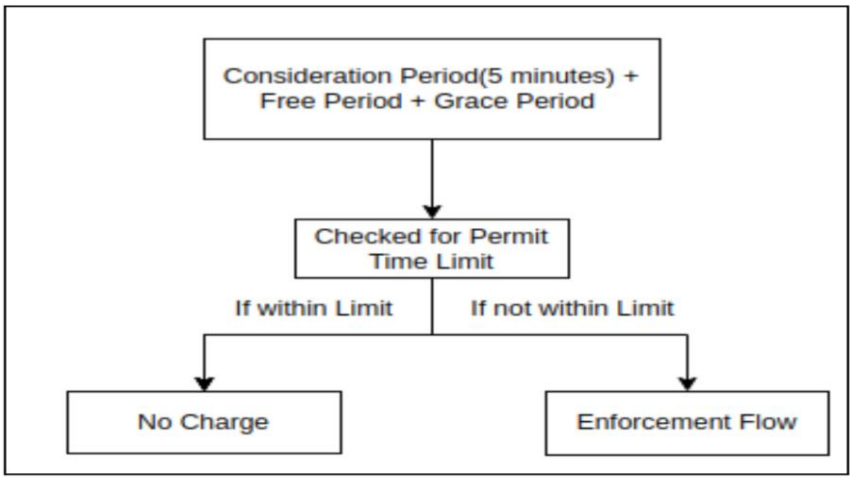

Fig. 4.1 Estate Management Consideration Period

A module where the activities of Manco, Clients, Sites, Car Parks, Devices can be monitored and handled. These sub-sections of estate management are the organizing entities who organize the parking related activities/facilities of Users. All these sections work in a hierarchical way that will be:

Manco → Client → Site → Car Park

4.2 Permit Management

Provides users to apply for the permit using multiple permit form templates. On user application admin has a permission to approve or reject the application. He can assign the application to the permit group defined at a particular site and car park.

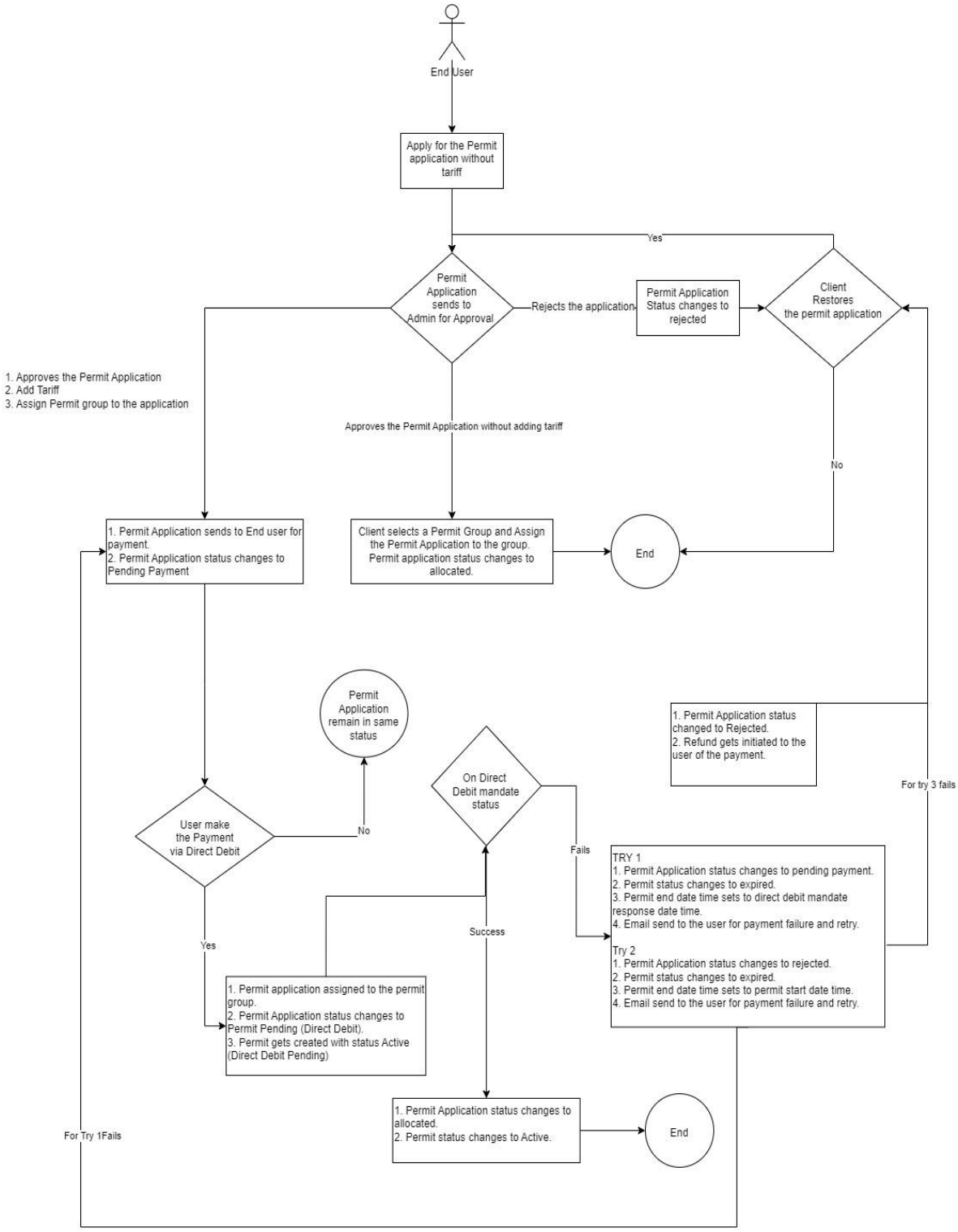

Fig. 4.2 Permit Application Creation Flow

Creation of the Permit application form:

- Admin can create the permit application form.
- Every site can have multiple active forms, but to create more than one form per site, the user needs allow it in Estate management else only single form is allowed.
- Also, while creating the form, the user can have the choice whether he wants one VRM (Vehicle registration number) or three VRMs in the application.
- Users get the choice to create the application form for a particular site and the car park.
- Tariffs can be added while the creation of the form by the admins.

Filling of the application form by End user:

- End users can fill the form and apply for the permit.
- If multiple forms are available for a site, the user gets the option to select the form and then fill it.
- Once the user submits the form, the form is under waiting approval and it is visible in the application tab to the user.
- Once it gets approved by the admins, it will go to the approved state and then it will be visible to the user in the permits tab.
- If the permit tariff is added in the form, the user will be asked to pay the amount and until the user pays.
- The form can be rejected by the admin, which will then be in rejected state and it will be visible to the user in the applications tab

Permit Approval Flow via Admin

- Admin has an access to approve the application.
- There is also an option to bulk approve, only point is that all the permits selected for the bulk approve must be of similar form.
- If the permit tariff is not applied on site, admin will get the pop-up to add the tariff amount on that site/car park.
- If the admin wants to add the tariff, can simply enter the amount and click on continue.
- If the admin doesn't want to add the tariff, he can click on the continue button without entering the amount.
- Now if the tariff is already added while the creation of the form , no such pop-up to add the tariff will be displayed to the user.
- Once the user pays the tariff the application will again come in waiting approval state and then again, the admin needs to approve it. Upon approving it again, the admin will get the pop-up modal to select the group and then after selecting the group he can click on approve.

Permit Rejection Flow via Admin

- Admin can reject the permit application
- If a single permit is rejected, it will directly go to the rejected application.
- Single permit can be rejected either by selecting it from the outside or by going into the permit detail page.
- Admin can do the bulk reject by selecting the multiple application from the waiting approval list page.
- Once the permit rejection is confirmed from the rejected pending tab, it moves to the rejected application form.
- If the payment was done for the permit application and then the permit has been rejected by the admin, then the payment will be refunded.

4.3 Enforcement

The process directed towards collection of the payments and entails issuing documents to contravention based on the regulatory and client requirements is called Enforcement.

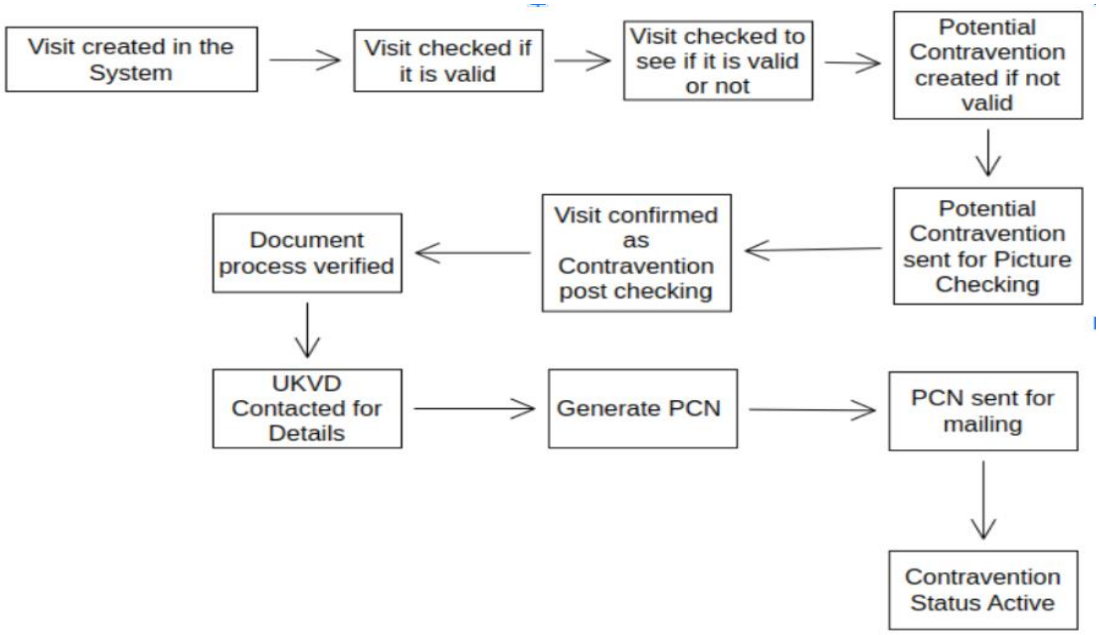

Fig. 4.3 Enforcement Management Model

CHAPTER 5

Software Requirement Specification

5.1 Use Case Diagrams

5.1.1 Permit Management Use Case diagram

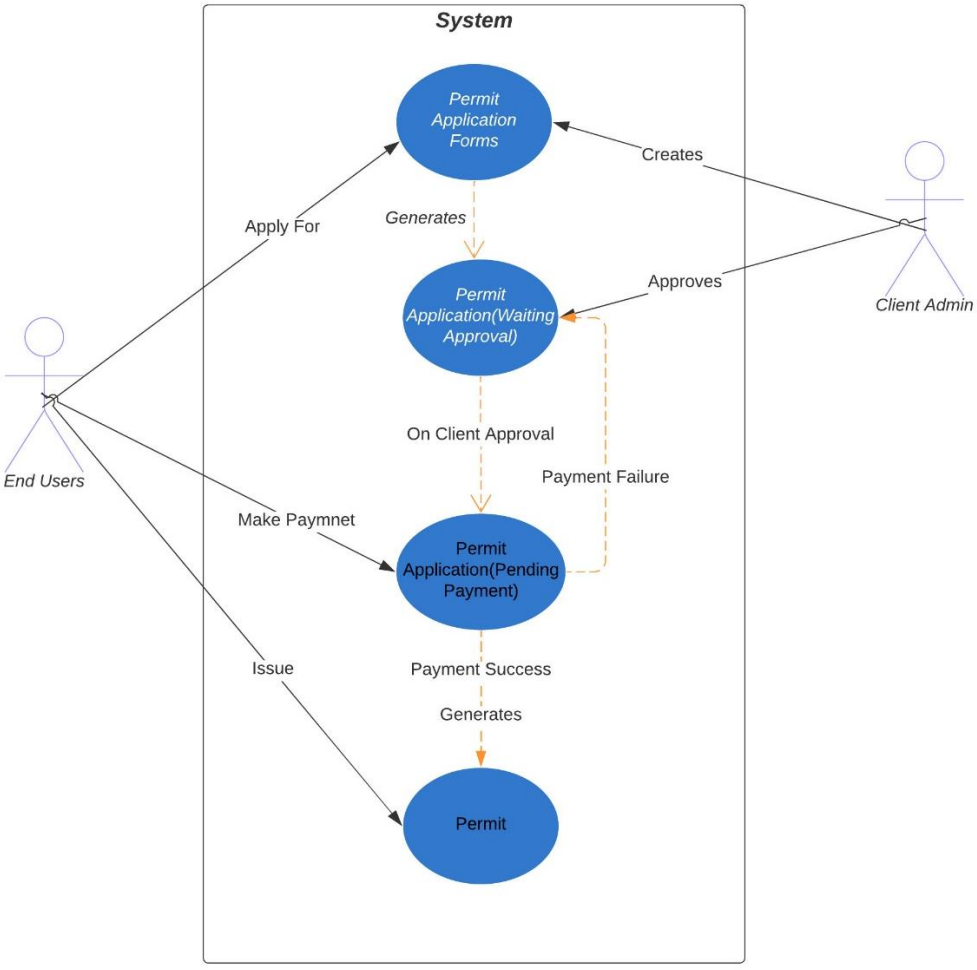

Fig. 5.1 Permit Management Use Case Diagram

5.1.2 Enforcement Use case diagram

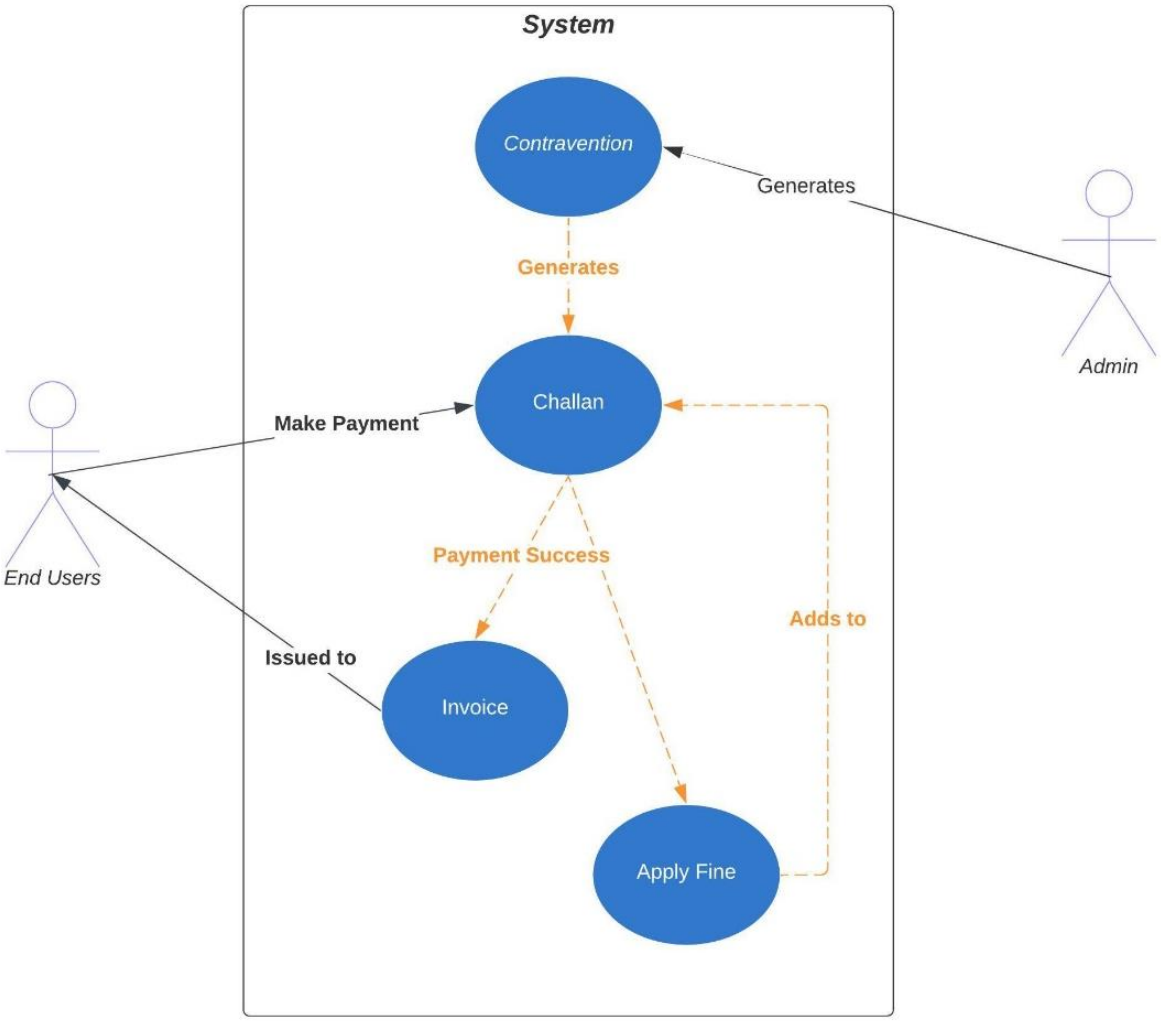

Fig. 5.2 Enforcement Use Case Diagram

5.2 Sequence Diagrams

5.2.1 Permit Management Sequence Diagram

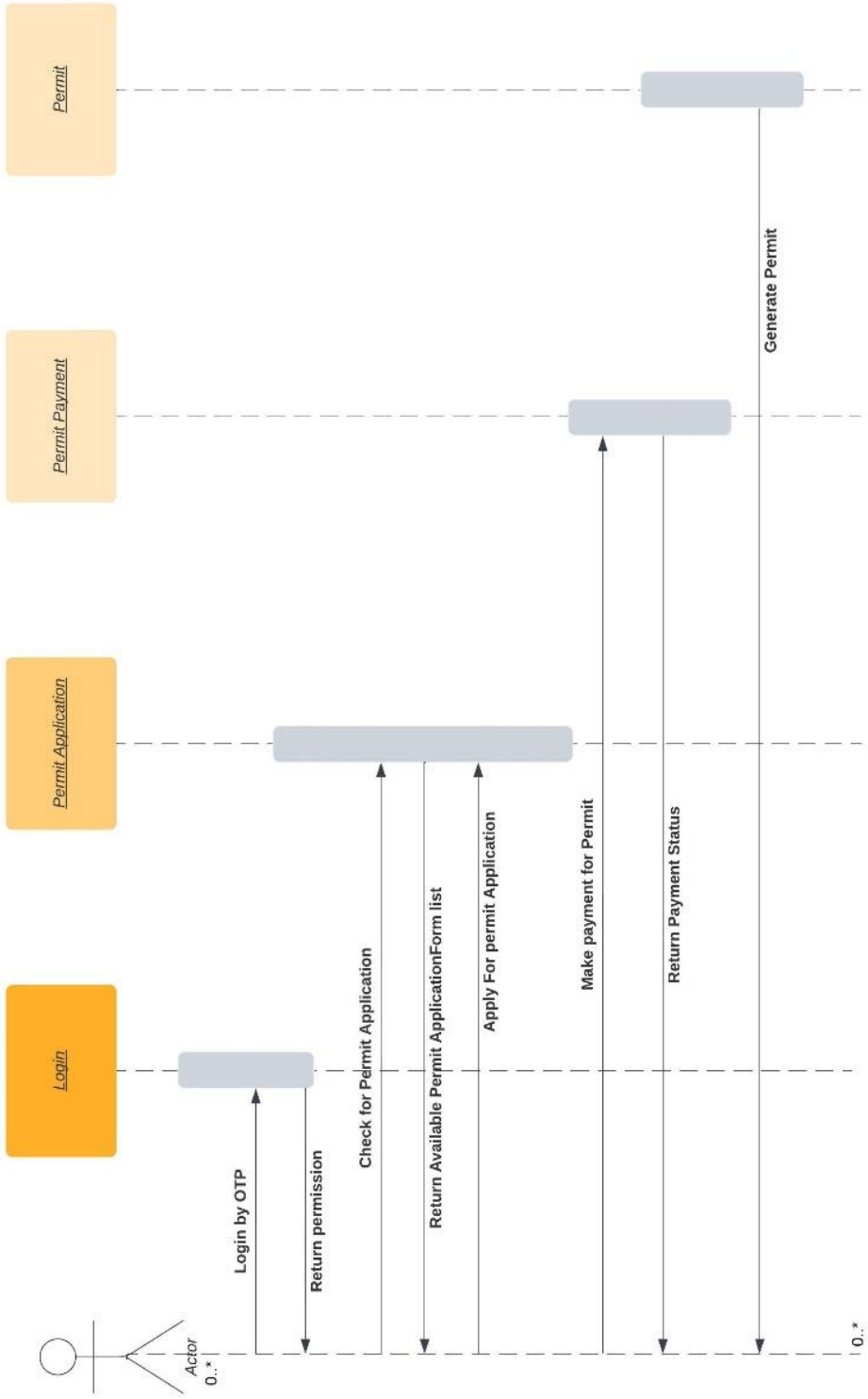

Fig. 5.3 Permit Management Sequence Diagram

5.2.2 Enforcement Sequence Diagram

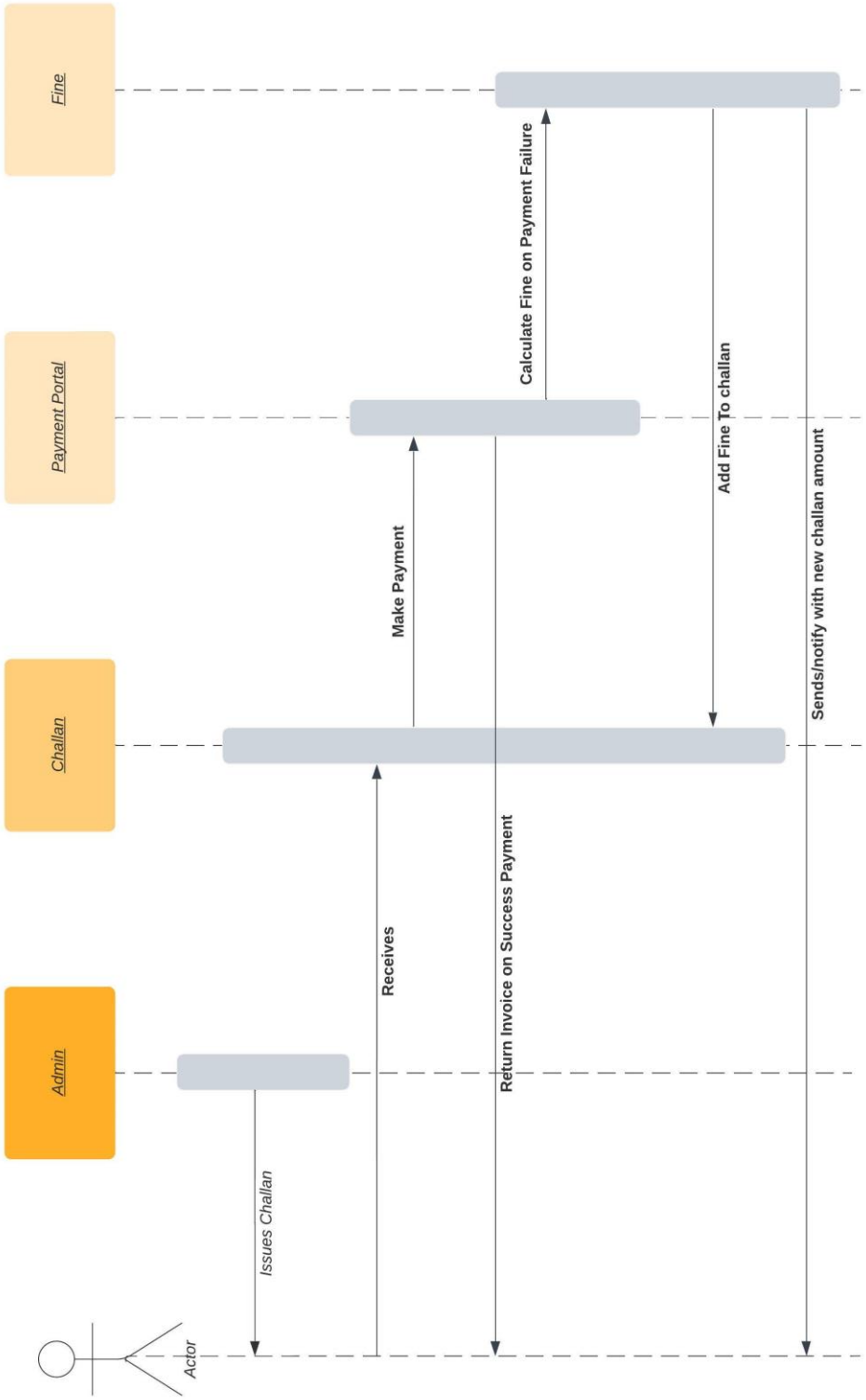

Fig. 5.4 Enforcement Sequence Diagram

CHAPTER 6

Software Testing

5.1 Testing Process

Testing is a process to show the correctness of the program. Testing is needed to show completeness, it improves the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to reduce the testing costs and operation time. Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing, we can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

5.1.1 Black box testing

Black box testing, also called behavioral testing, focuses on the functional requirements of software. This testing approach enables the software engineer to derive the input conditions that will fully exercise all requirements for a program. Black box testing attempts to find the errors like Incorrect or missing functions

- Interface errors.
- Errors in data structures or external database access
- Initialization and termination errors

In Black box testing software are exercised over a full range of inputs and outputs are observed for correctness.

5.1.2 White box testing

White box testing is also called Glass box testing is a test case design control; structure of the procedural design to derive test cases using White box testing method, the software engineer can derive the test cases that guarantee that all independent paths within the module have been exercised at least once. Exercise all logic decisions on their true or false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structure to ensure their validity.

5.2 Software Testing Strategies

Testing involves Unit testing Integration testing Acceptance testing To create your account, Google will share your name, email address and profile picture with academia.edu. See academia. Edu's/ privacy policy and Terms of Service. The first level of the test is unit testing. The purpose of unit testing is to ensure that each program is fully tested.

The second step is integration testing. In this individual program units or programs are integrated and tested as a complete system to ensure that the software requirements are met. Acceptance Testing involves planning and the execution of various types of tests in order to demonstrate that the implemented software system satisfies the requirements. Finally, our project meets the requirements after going through all the levels of testing.

Chapter 7

Conclusion

The project saves data by paying and saying few details about customers and vehicles, thus reducing parking tasks. In this case, the vehicle parks safely.

Adopting a parking management system significantly reduces the amount of time consumed in seeking the parking space, renders valuable data upon the availability of the parking area, accurate mapping of the parking space, offers guidance and suggestion for proper vehicle parking.

.

Chapter 8

References

Riley, D.B. and Weyman, A.K., 1995, "A survey of permit-to-worksystems and forms in small to medium sized chemical plant, MajorHazards Onshore and Offshore II", Symposium Series No. 139, 367(IChemE, Rugby, UK).

Abhirup Khanna, Rishi Anand, "IoT based Smart Parking System", Proc., In 2016 International Conference on Internet of Things and Applications (IOTA), 22 Jan - 24 Jan 2016.

Anusha, Arshitha M, S, Anushri, Geetanjali Bishtannavar "Review Paper on Smart Parking System," International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume 7, Issue 08, Special Issue – 2019.

S. Senthil, M. Suguna, J. Cynthia, "Mapping the Vegetation Soil and Water Region Analysis of Tuticorin District Using Landsat Images", IJIEST ISSN (2455-8494), Vol.03, No. 01, Jan 2018.