**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**

**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 **Josh Technology Group** submitted for the B.Tech 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 Applications 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:

**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 behaviour.

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 handedly. My deepest gratitude goes to my thesis supervisor, **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. Ajay Kumar Shrivastava, 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**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Certificate | | | | I |
|  | Abstract | | | | ii |
|  | Acknowledgements | | | | iii |
|  | Table of Contents | | | | iv |
|  | List of Abbreviations | | | | ix |
|  | List of Tables | | | | xii |
|  | List of Figures | | | | xiii |
|  |  | | | |  |
| 1 | Introduction | | | | 1-16 |
|  | 1.1 | Overview | | | 2 |
|  |  | 1.1.1 | ……………. | | 4 |
|  |  | 1.1.2 | …………………….. | | 10 |
|  |  |  | 1.1.2.1 | ………………… | 10 |
|  | 1.2 | ………………………… | | | 12 |
|  |  | 1.2.1 | ………………………………… | | 13 |
|  |  | 1.2.2 | ……………………………….. | | 13 |
|  | 1.3 | ………………………………. | | | 14 |
|  | 1.4 | ……………………………… | | | 15 |
|  | 1.5 | ……………………………….. | | | 16 |
| 2 | Feasibility Study | | | | 17-36 |
|  | 2.1 | …………………………. | | | 18 |
|  | 2.2 | ……………………… | | | 18 |
|  | 2.3 | ……………………… | | | 35 |
|  | 2.4 | ………………… | | | 35 |
|  | 2.5 | …………………… | | | 36 |
| 3 | Design | | | |  |
|  |  |  | | |  |
|  | Bibliography | | | |  |
|  |  | | | |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Table** | **Page** |
| 2.1 |  | 32 |
| 2.2 |  | 34 |
| 5.1 |  | 81 |
| 5.2 |  | 83 |
| 5.3 |  | 84 |
| 5.4 |  | 85 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** | |
| 1.1 |  | | 2 | |
| 1.2 |  | | 5 | |
| 1.3 |  | | 5 | |
| 1.4 |  | | 6 | |
| 1.5 |  | | 7 | |
| 1.6 |  | | 8 | |
| 1.7 |  | | 9 | |
| 1.8 |  | | 11 | |
| 1.9 |  | | 12 | |
| 3.1 |  | | 40 | |
| 3.2 |  | | 45 | |
| 4.1 |  | | 54 | |
| 4.2 |  | | 55 | |
| 4.3 |  | | 57 | |
| 4.4 |  | | 58 | |
| 4.5 |  | | 59 | |
| 4.6 |  | | 60 | |
| 4.7 |  | | 62 | |
| 4.8 |  | | 63 | |
| 4.9 |  | | 63 | |
| 5.1 |  | | 87 | |
|  |  | |  | |

**INTRODUCTION**

* 1. **OVERVIEW**

With the rapid development of 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.

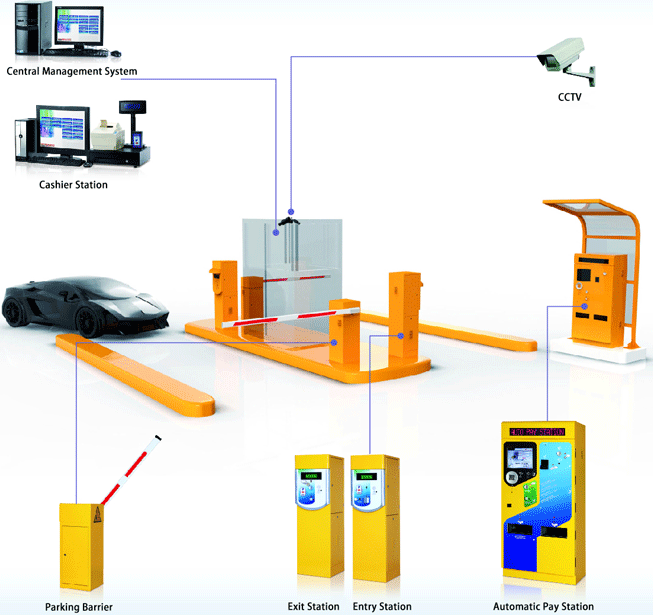


Figure 1.1 Parking Management Model

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.
   1. **OBJECTIVE**

The aim of implementing 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. **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. **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. **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. **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. **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. **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. **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.

Documents support a wide variety of data types, such as nested objects, numbers, and strings. Firestore enjoys integration with Google Cloud Platform and Google Firebase. Businesses prefer Firestore for the level of security and reliability it offers.

Key Features of Firestore:

* Automatic scaling Firestore has been designed to scale automatically depending on user demand. It retains the same level of performance irrespective of database size. Firestore database size does not impact query time.
* Server less development Client-side SDK of firestore handles networking and authentication while reducing the requirement for coding. Its backend security rules enable swift access to data and help you apply sophisticated validation logic on data.
* Offline Usage Cloud Firestore from Google facilitates convenient offline usage through a robust database on users’ devices. Offline data access ensures that applications run seamlessly even if the user gets disconnected from the Internet. Offline functionality can be availed by users across the web, iOS, and Android platforms.
* Datastore Mode Cloud Firestore offers Datastore API support. Developers do not need to make any changes to current Datastore applications. This feature helps users get uniform performance and benchmark stability while maintaining cost-efficiency.
* Robust Query engine Google Firestore has a sturdy and high-performance query engine. This critical feature helps developers execute complex queries against NoSQL data. It helps to do so without causing any compromises in performance. Users get greater flexibility when it comes to data structuring.
* ACID transaction ACID (atomicity, consistency, isolation, and durability) transaction is an essential feature of Cloud Firestore from Google. This feature helps to support transactions. In the case of operational failures in a transaction, the transaction will fail entirely.

**SYSTEM ANALYSIS PHASE**

* 1. **INFORMATION GATHERING**

Information gathering is done by interviewing the users and reviewing the existing documents. For the development of Parking management system, a lot of research and important input from various website and application user was needed. Hence the following questionnaires were provided to them and hence the need for our website arises.

* + Interviewing the users:
    - What are the difficulties you are facing in the existing system?
    - What all new things you want to be included in the proposed system?
    - In what way you are storing your information?
    - Who all are the users of the system?
  1. **USER REQUIREMENT**
* Need for an application that makes communicating easy and comfortable.
* An application that enables user to park a vehicle with safe and secure.
* 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
  1. **FUNTIONAL REQUIREMENTS**
     1. **Performance Requirements**
     + **User Satisfaction**: The system is such that it stands up to the user expectations.
     + **Response Time**: The response of all operations is good.
     + **Error Handling**: Response to user errors and undesired situation has been taken care of to ensure that the system operates without halting.
     + **Safety and Robustness**: The system is able to avoid or tackle disastrous action.
     + **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.
     1. **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 on integral part here, detailing what the system should do if some failure occurs to ensure certain properties.
     1. **Hardware Requirements**

Hardware requirements are like Memory Restrictions, Cache Size, the processor, RAM Size etc. that are required for software to run.

Preferred Hardware Requirements are –

* Processor Core i3 or greater.
* Hard Disk Drive with minimum storage capacity of 500 GB.
* RAM minimum 4 GB.
  + 1. **Software Requirements**
    - Any windows-based operating system is the primary Requirement for the Software Development.
    - Windows 7 and up are required.
    - The system must be connected to the Internet through any mode.

* 1. **NON-FUNCTION REQUIREMENTS**
     + **Performance**: 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.
     + **Security** **Requirements and Privacy**: Application should be automatically logout all customers after a period of inactivity. The system should not leave any cookies on the customer’s computer containing the user’s password. The system’s backend server shall only be accessible to authenticated management.
     + **Availability**: The Application should be available all the times 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.
     + **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. Online user documentation and Help system requirements will be provided.
     + **Operational** **Requirements**: The application must work on all mobile and tablet devices as well. The user interface must be consistent on all devices.

**SYSTEM PLANING PHASE**

* 1. **PROCESS MODEL**

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 analysed to make further decisions. This process is then repeated, creating a new piece of software for each cycle of the model.

* 1. **FEASIBILITY TEST**
     1. **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.

* + 1. **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

* + 1. **OPERATIONAL FEASIBILTY**

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.

* + 1. **TECHNICAL FEASIBILITY**

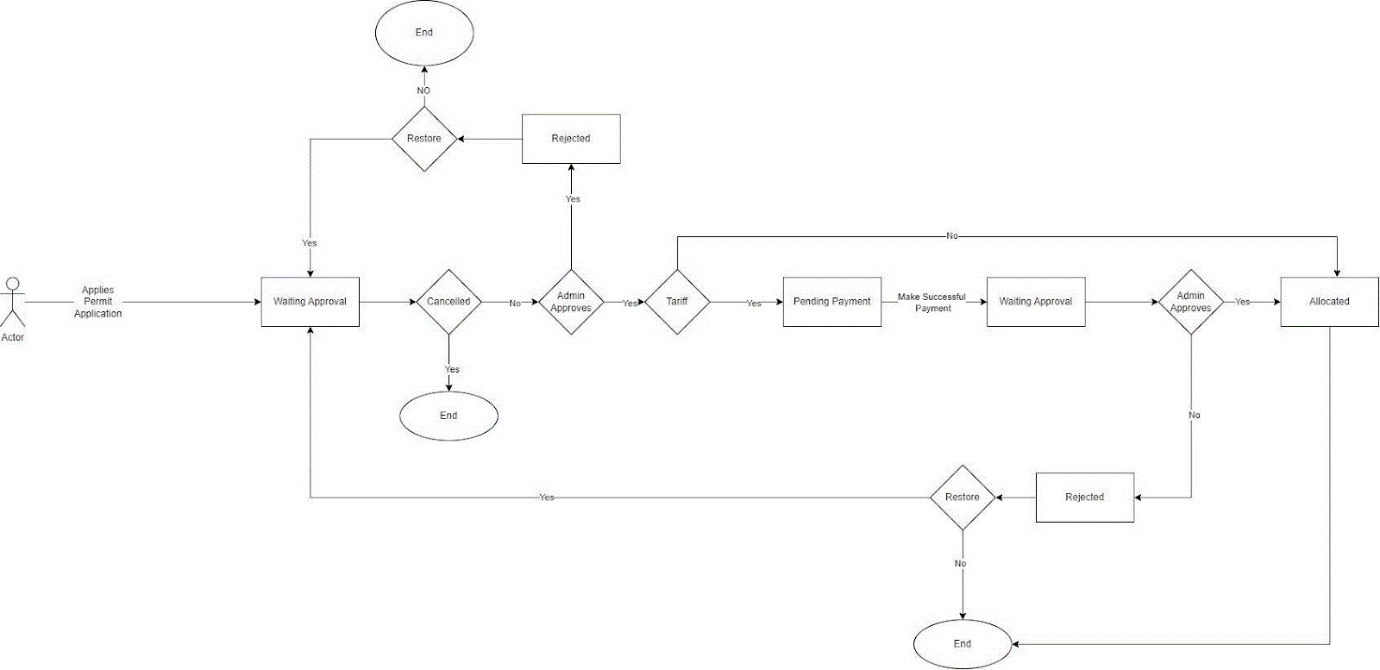
Technical Feasibility centers on the existing computer system (hardware, software, etc.) and to what extent they can further support demand. For example, if the computer is currently running at 80% capacity (upper limit), running other applications may overload the system or require additional hardware.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

**SOFTWARE REQUIREMENT SPECIFICATION**

* 1. **PERMIT APPLICATION**

Provides user 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 particular site and car park.

**User flow diagram:**

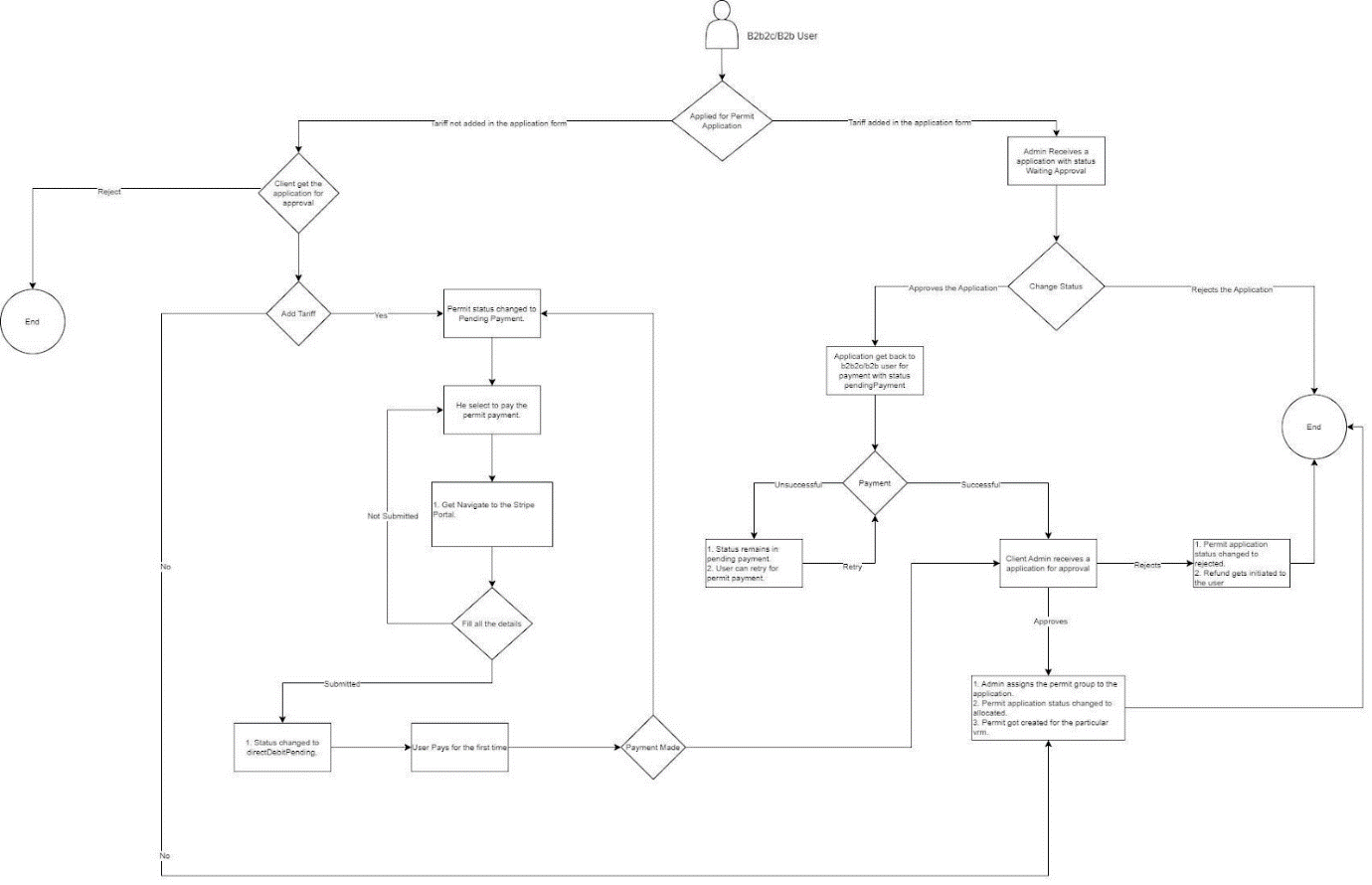
****

* 1. **PERMIT PAYMENT**

Platform provides three different types of permit payments -

* + Pay by card
  + Pay by Google pay/ Apple pay
  + Direct debit (BACS model): Subscription based model

**User flow diagram:**

****

**SYSTEM TESTING PHASE**

**5.1 TESTING PROCESS**