Bachelor Thesis Project (BTP) Report

Surisetti Sai Komal

Roll Number: 1903136

Songa Yaniv Nishanth

Roll Number: 1903135

Month of Submission: April, 2023

Supervisor: Dr. Sharad Sinha, Computer Science and Engineering

This report is submitted towards partial fulfillment of the requirements for the award of the Bachelor of Technology (B.Tech) degree in Computer Science and Engineering at the Indian Institute of Technology Goa.

Contents

1. Problem Statement

1.1 Introduction

Vehicle Parking Management system is a web-based application that will manage the records of the incoming and outgoing vehicles in a parking house. It's easy for Admin to retrieve the data if the vehicle has been visited through a number he can get that data. Vehicle parking management system is an automatic system which delivers data processing at a very high speed in a systematic manner.

1.2 Problem Definition

At present all vehicle parking work is done on paper. The whole year vehicle parking record is stored in the registers. We can't generate reports as per our requirements because it takes more time to calculate the vehicle parking report.

Disadvantage of present system:

- **Not user friendly:** The present system is not user friendly because data is not stored in structure and proper format.
- **Manual Control:** All report calculation is done manually so there is a chance of error.
- **Lots of paperwork:** Vehicle records are maintained in the register so lots of paper requires storing details.
- **Time consuming**

1.3 Objective

There is always a need for a system that will provide a way to effectively control records & track vehicle parking traffic.

Thus, there is a big need for developing vehicle parking management systems to computerize the traditional way of parking management. Another need for developing this application is to generate the report automatically.

We provide a way to the users where they can book their parking slots for however long they need and view it anytime they want .

2 . Principal Contributions

2.1 System design

System design is an essential process for building a software application that meets the requirements of users and stakeholders. It involves defining the modules, architecture, components, interfaces, and data needed for the system. The design process must take into account the functional and non-functional requirements, such as performance, scalability, security, and usability.

Web application architecture is a layout that defines the interactions between the various components of a web application, including middleware systems, user interfaces, and databases. The architecture is designed to support multiple applications that work together simultaneously, providing a seamless user experience. The architecture can be designed to be scalable, secure, and easily maintainable, ensuring that the system can adapt to changing requirements and user needs.

System development is the process of creating or altering systems and the processes, practices, models, and methodologies used to develop them. The development process involves analyzing requirements, designing the system, implementing it, testing it, and deploying it. Agile methodologies have become popular for system development, as they promote collaboration, flexibility, and continuous improvement.

Overall, system design and development are critical components of building a successful web application. By carefully defining the system architecture, designing and implementing high-quality components, and following best practices for development, teams can create web applications that meet the needs of users and stakeholders, and can evolve over time to keep pace with changing requirements and technologies.

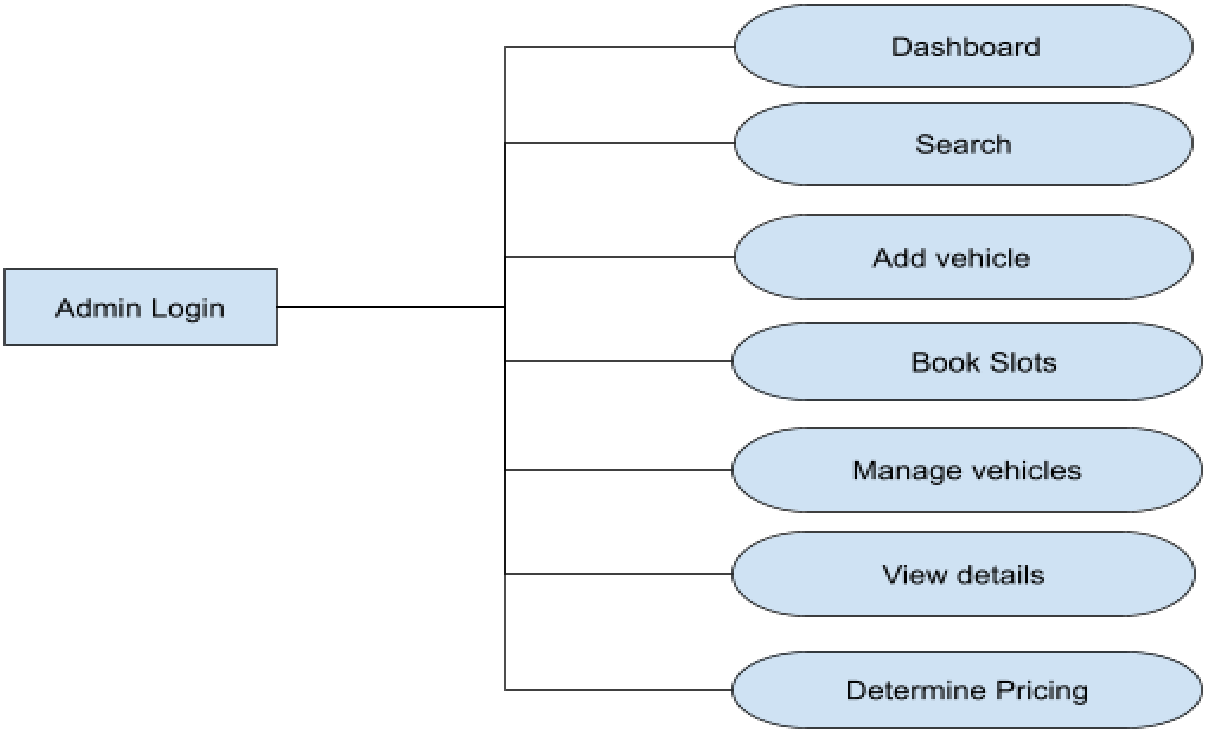
2.2 User Interface

A web user interface or web application is a software application that enables users to interact with content or software hosted on a remote server through a web browser. The web page containing the content is downloaded from the server and displayed in the user's web browser, which serves as the client. The user can then interact with the content via the browser's interface, enabling them to carry out tasks, access data, or manipulate software from anywhere with an internet connection. Web UIs have become ubiquitous, with everything from social media platforms to enterprise-level software now accessed via web interfaces.

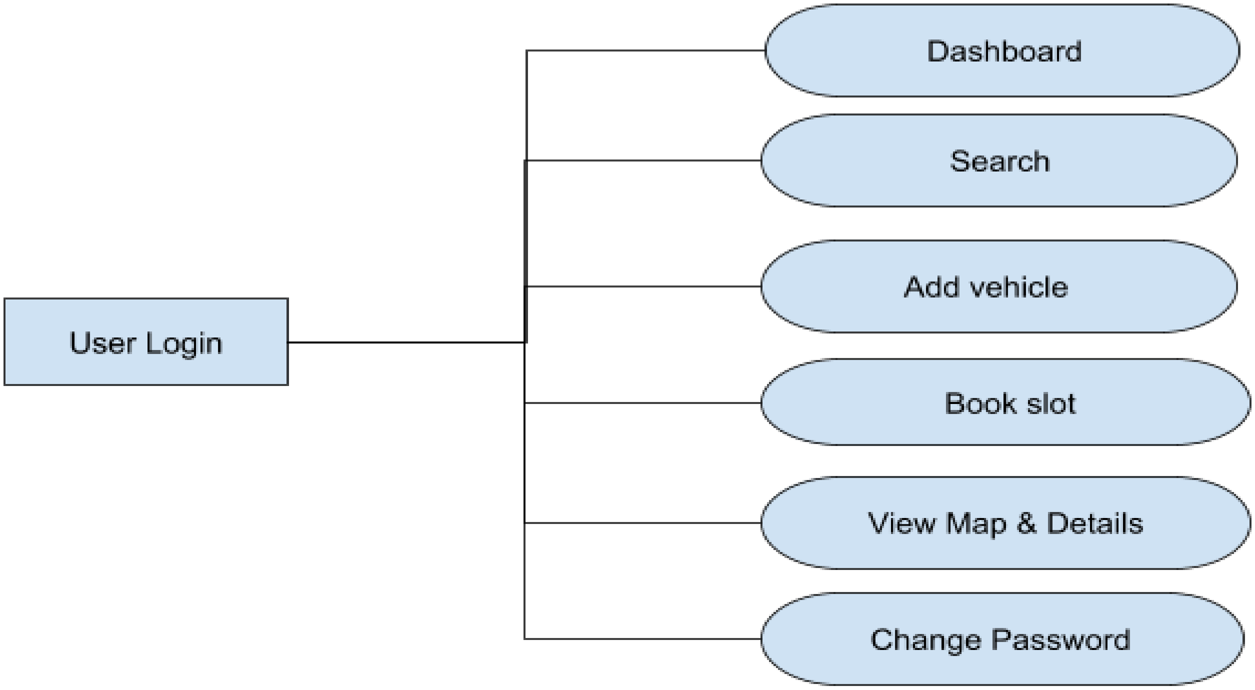
2.3 Use Case Diagram

The Use case diagram illustrates what the user and admin can do.The diagram can help to provide a high-level overview of the application's functionality and how it is intended to be used, which can be useful for both development and documentation purposes. The following are Use case Diagrams for User and Admin.

a. Admin - Use Case Diagram

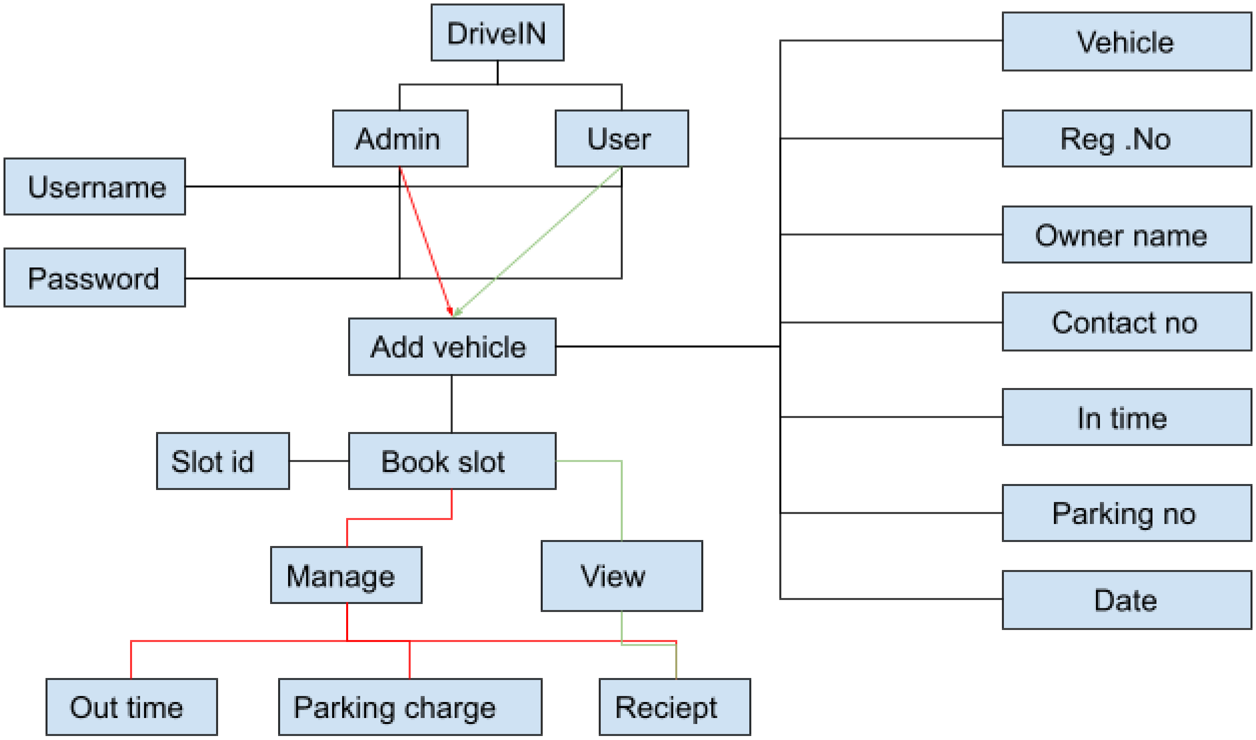


b. User - Use Case Diagram:



2.4 ER Diagram

This is the ER diagram for the website.



3. Learnings

3.1 Frontend

For the frontend of this web application, we have used:
HTML: Which is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

CSS: Cascading Style Sheets (CSS) are a collection of rules we use to define and modify web pages. CSS are similar to styles in Word. CSS allows Web designers to have much more control over their pages' look and layout. For instance, you could create a style that defines the body text to be Verdana, 10 points. Later on, you may easily change the body text to Times New Roman, 12 points by just changing the rule in the CSS. Instead of having to change the font on each page of your website, all you need to do is redefine the style on the style sheet, and it will instantly change on all of the pages that the style sheet has been applied to. With HTML styles, the font change would be applied to each instance of that font and have to be changed in each spot.
CSS can control the placement of text and objects on your pages as well as the look of those objects.
HTML information creates the objects (or gives objects meaning), but styles describe how the objects should appear. The HTML gives your page structure, while the CSS creates the “presentation”. An external CSS is really just a text file with a .css extension. These files can be created with Dreamweaver, a CSS editor, or even Notepad.

Javascript: JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a web page has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

3.2 Backend

For the backend of this web application, we have used SQLite and Django. SQLite helps in storing all the data in the backend, and Django, as a framework, helps in working with all the data that comes in the backend. All the data that is collected from the user can be accessed through the backend of the web application and sent to the admin in this case

4. System Overview

4.1 Working

Vehicle Parking Management system is a web-based technology that will manage the records of the incoming and outgoing vehicles in a parking house.
The key features required in the system are as follows:
Dashboard: In this section, admin can briefly view the number of vehicle entries in a particular period.
Add Vehicle: In this section, admin and users can add a vehicle which is going to park.
Parking Map: In this section, admin can book slots for the vehicles.
Manage Vehicle: In this section, admin can manage incoming and outgoing vehicles and admin can also add parking charges and his/her remarks.
In/Out Status: In this section, users can view their status of the vehicle and also check their parking spot anytime by clicking on the parking number in the In Status page.
Reports: In this section admin can generate vehicle entries reports between two dates.
Search: In this section, admin can search a particular vehicle by parking number.
Admin and users can also update and change the password.

The application features a login page that allows users and admins to access the system. Upon accessing the login page, users can choose to sign up for a new account. The login page includes two fields, namely, username and password, which will be stored in the database. The register page, on the other hand, has three fields, namely, username, email, and password, which will also be stored in the database. This setup ensures that users and admins can securely access the application, and their credentials are stored in a secure database for easy retrieval and management..

Upon entering the login credentials, the system authenticates the user's response with the data stored in the database. If the entered details match the data on the server, access is granted to the specific user. During this authentication process, the server validates the user's response to ensure the user is authorized to access the application. Once the authentication is successful, the user is directed to the dashboard, where they can access various features of the application. This process ensures that only authorized users can access the system, maintaining the security and integrity of the application and the data stored within.

Upon successfully logging into the application, the user is directed to the dashboard, which is the main interface of the application. The basic structure of the dashboard is created using HTML, while CSS is used for advanced structuring, coloring, and alignment of buttons and divisions. The dashboard typically contains various elements such as menus, charts, tables, and other features specific to the application.

The top left corner of the dashboard displays the website logo, providing visual identification of the application. Moving to the right, a user icon is displayed along with the user's image. This icon is used to identify the user's login type and provides options to log out of the current session and change the password.

On the left side of the dashboard, a sidebar is used to navigate to different sections of the application. This sidebar typically contains menus or links to various features and sections of the application, making it easy for users to access the required features.

Overall, the dashboard provides a user-friendly interface for accessing the various features and functionalities of the application, making it easier for users to interact with the system and achieve their objectives efficiently.

In the center of the web-page, the dashboard displays different information based on the user's role. For the admin, the number of vehicles present in the parking lot is displayed. In contrast, for the user, general instructions on how to navigate the website are displayed.

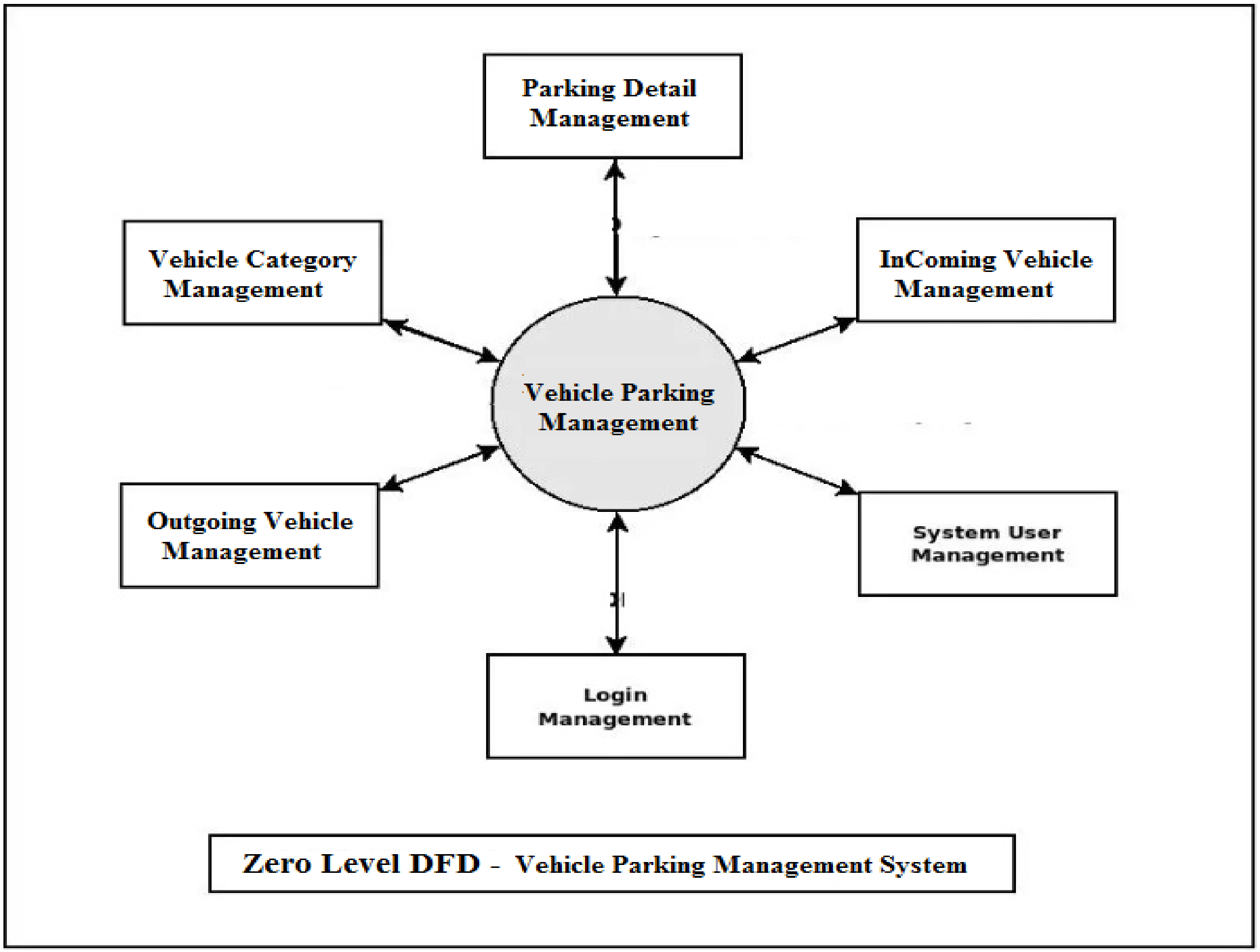
To add a new vehicle, both the admin and user click on the "Add Vehicle" button, which takes them to a new page where they need to enter the vehicle details and click on "Next." Upon clicking the button, the user is directed to the parking map where they can see the available slots. The slots that are filled are displayed with an image of the registered car. Users can book one of the available slots and click on "Submit." For the admin, clicking on the button directs them to the "Manage Incoming Vehicle" page, where they can view all the registered vehicles that are still in the parking lot.

Each vehicle is assigned a unique parking number that is associated with a URL. Clicking on the parking number displays the slot where the car is parked. The admin can view everyone's parking spot since they have access to all the registered vehicles. This feature makes it easy for both the admin and the users to manage their vehicles and parking spots efficiently.

In the "Manage Incoming Vehicle" page, the admin can take action on the parked vehicles, such as adding out-time, parking fees, and remarks, if any. Upon clicking on "Submit," a new page opens, displaying the details of all outgoing vehicles. The admin can view and print the receipt and even delete the entry completely.

Once the admin has managed the incoming details and sent the vehicle outgoing, the users can view their receipts in the "Out Status" page. This page displays the details of the vehicle, including the parking fee and out-time. Users can also print the receipt for their records. This feature enables users to keep track of their parking details and make payments quickly and easily, enhancing the overall user experience. The system also provides the admin with a convenient way to manage the parking lot efficiently and keep track of all incoming and outgoing vehicles, making it a valuable tool for both the admin and the users.

4.2 Dataflow diagram


Parking Detail Management
Vehicle Category Management
InComing Vehicle Management
Vehicle Parking Management
Outgoing Vehicle Management
System User Management
Login Management
Zero Level DFD - Vehicle Parking Management System

5. Advantages and Limitations

5.1 Advantages of Vehicle Parking Management system

“Vehicle Parking Management System” provides various features, which complement the information system and increase the productivity of the system. These features make the system easily usable and convenient. Some of the important features included are listed as follows:

- Intelligent User Forms Design
    - Data access and manipulation through same forms
    - Access to most required information
- Data Security
- Restrictive data access, as per login assigned only.
- Organized and structured storage of facts.
- Strategic Planning made easy.
- No decay of old Records.
- Exact financial position of the business.

5.2 Limitations of Vehicle Parking Management System

Besides the above achievements and the successful completion of the project, we still feel the project has some limitations, listed as below:

- It is not a large-scale system.
- Only limited information provided by this system.
- Since it is an online project, customers need internet connection.

6. Conclusion

The Vehicle Parking Management System is a reliable and efficient solution for managing company visitor parking. With its user-friendly interface and backend technologies such as Python Django and Sqlite database, the system streamlines the parking process, making it easier for visitors to register their vehicles and for the company to manage parking slots. Overall, the system improves the efficiency and accuracy of company visitor parking management.

7. Bibliography

7.1 Resources

https://www.w3schools.com
https://stackoverflow.com
https://www.geeksforgeeks.org/django-tutorial/
https://www.javatpoint.com/django-tutorial/
https://www.sqlitetutorial.net/

7.2 References

Two scoops of Django for 1.11 by *Daniel Greenfeld's and Audrey Greenfield*

Lightweight Django by *Elman and Mark Lavin*

8. Appendix

To run this code you need to have at least Python 3.8 version.

This project is built using django version 3.1.3. To install django run this command in the terminal:

```
pip install django==3.1.3
```

After installing django, navigate to the directory where the project folder is located. Once you get there, run the following command:

```
python manage.py migrate
```

For windows, it may be python3 manage.py migrate.

This will load the models in models.py file to run the project.

Finally to start the backend and open the website, run the following command:

```
python manage.py runserver
```