Bilkent University

Department of Computer Engineering

Senior Design Project

ParkHound

Project Specifications Report

Group Members:	Arda Türkoğlu
	Ege Turan
	Berkin İnan
	Görkem Yılmaz
	Ata Coşkun
Supervisor:	Varol Akman
Jury Members:	Uğur Doğrusöz, Ercüment Çiçek
Innovation Expert:	Nezir Ertürk

Table of Contents

1. INTRODUCTION

With the constant increase in population and owned cars in cities, finding a parking spot becomes a more prominent problem. Drivers spend hours around a parking lot to find an available spot to find their cars [1]. A 35% travel during rush hours looking for free parking spots that are hard to find. This makes parking very important to regulate vehicle flow and reduce atmospheric pollution [2]. This search causes drivers to waste time and gas. Therefore, parking systems are very important for regulating the traffic flow and sustaining transportation of the city. Because of the increasing number of personal vehicles, many foundations cannot provide enough parking slots or their parking areas face several parking problems. Since foundation can have many different and large-scale parking areas.

In addition to parking availability, there are two other aspects of outdoor vehicle parks. One of them is the security of the parking area which includes car security and protecting the area of the property. It is not always easy to control a large area separated for cars and this uncontrolled area can be a place for illegal activities. Secondly, some outdoor vehicle parks demand some regulations about parking, such as placing the car between the white lines. Normally, this kind of problem is handled by the security personnel in the foundation. They are walking around and making controls but again, this is not easy and has a high percentage of error-prone in the large outdoor vehicle parks.

For almost a decade many foundations have used sensor-based control parking slot maintenance systems in closed car parking spaces. However, the management of the outdoor car parking spaces is not easy for sensory systems due to the safety of the devices and the placement difficulties faced by the producers. Besides, these sensory systems cannot handle many security issues and regulations about parking. Therefore, we came up with an idea of creating a computer vision system to maintain parking spots in outdoor vehicle cars. Our system is based on the image processing unit and machine learning system that uses predetermined dataset for understanding the status of the parking slot in the open air and maintaining the specified regulations and the

security concerns. Our system will be implemented on the camera system and can be used for the large-scale open vehicle parks.

1.1 Description

ParkHound is a cross-platform application that will help users to find available parking spots in an outdoor parking lot. Currently, this can be done with sensors in an indoor parking lot. ParkHound will provide a similar service without having to install sensors near each of the parking spots using object detection and share this information via an application, so users can see the available parking spots without having to drive around the parking lot.

Normally, maintenance of the outdoor vehicle park is handled by the regular auto park workers or security in some situations. However, many foundations have a large scale of outdoor vehicle parks, and maintenance of the large parking system is very hard for the security people. In addition to that, regular users of these parks spend their time finding empty slots. Especially in the rush hours, people are waiting for the empty spots and this creates traffic jams in the outdoor vehicle parks therefore people lost more time in the rush hours. ParkHound solves this problem by detecting empty parking spots and providing an abstraction of the parking area for regular users.

Our system will first create an abstract map of the parking lot using the parking lines. On this map, users will be able to see which parking spots are available in the parking lot. This map will be available to users from their mobile app.

Our system will provide information for two types of users. One of them is a regular user who uses the parking spots in specified areas. Our system will have a browser-based web application and as well as mobile applications and regular users will be registered to the system using their email address and password which is provided by their company or organization. Information that they see will be restricted because of security concerns. Our systems will show the map and route of the parking lot and the

unoccupied parking spots in that parking lot. Regular users (car owners) can only see that parking spot's location and information about it. They will be notified if they want to get notifications about that park. No further information will be shared by the application.

Another type of user is the security user, who is the one normally responsible for the outdoor vehicle park. Park security is normally responsible for viewing cameras and observing any anomalies. ParkHound helps park securities by notifying them about the wrong parking or fully occupied parks and showing them a live view of the parking spot. With this automation, we get rid of manual check on parking spots and more accurate detections. ParkHound will help security members by sharing car information and location of the problem park spot. Only registered user information and live view of the parking spot will be shown to the security user. They can also send a notification to the regular users about parking and traffic problems.

1.2 Constraints

1.2.1 Implementation Constraints

- The platform of ParkHound will be mobile and web application.
- ParkHound will use MySQL for the database.
- Object-Oriented Programming will be pursued for ParkHound.
- Some open-source libraries and datasets will be used for image analysis.
- GitHub will be used for collaboration among members of the project.
- Some creative UI designs and tools will be examined to improve the application.
- The server-client architecture will be used.
- Main high-level programming language for the project will be Python.
- The mobile application will be developed with React Native [3].
- A REST API will be used to communicate between server and application.

1.2.2 Economic Constraints

- The application will be cross-platform and it will be downloadable from Apple Store and Google Play Store.
- The IDE's, libraries, and datasets that will be used will be free.

● The server which contains data will be rented. To gain better profit, all options will be analyzed.

● Because ParkHound will be developed for mobile and desktop, users should have mobile devices or computers to use it. Also, they need an internet connection.

1.2.3 Data Constraints

● For testing and training of the application, some example data and dataset will be used for better progress. Datasets will be created by our data and will be received from open sources.

● All data coming from the user will be protected in a private database.

1.2.4 Language Constraints

● The main language of the ParkHound will be English.

● It will also provide a Turkish interface for users who do not speak English. Because security users may not be required to know English.

1.2.5 Privacy Constraint

● Personal data and privacy is an important issue and there are regulations about private data. So, ParkHound will not share any image and personal data with third-party applications. We will use images and private data to create our dataset. The main purpose of collecting data is to enhance our dataset for future analysis. For instance, to broaden the datasets of image analysis algorithms used. However, none of the gathered data should be associated with a user and others. We will get the required permission from the necessary authorities to take images.

1.3 Professional and Ethical Issues

ParkHound intends to show empty and occupied parking spots to users and wrong parking, traffic status to traffic control members of the customer of the app. To achieve this, our application requires data on the park spaces and user information. However, this might cause some concerns about the protection of gathered information. Professionalism requires data protection and privacy. Therefore, all data from the cameras will be used to enhance our learning algorithms and will not be shared. User information is required to create an account for the application and log in to the application. User information will be stored in a server and all information is private.

Thus, it will be made sure that the user information is not shared with third-party organizations. The main ethical issue is taking images of the auto parks and parking slots of the private zones. We will take permission from the owners of the properties. This will enable our team to create a neutral parking slot information system that will not lead to any misunderstanding of privacy and ethical issues. After these problems are solved, our application focuses on creating accurate algorithms on parking issues and informing the customers with a user-friendly interface.

2. REQUIREMENTS

2.1 Functional Requirements

2.1.1 Data Management

● Users can register to our application by providing their email addresses or company email addresses.

● There are 2 types of users. One of them is maintainers such as park securities and line drawers to standardize the park features. The other type of users are regular users.

● Regular users and securities should be informed when regular users park incorrectly.

● Regular users should not see information about other cars. Only they can see available slots in the parks.

● Users should be able to see which park slots are available for that park.

● Park spot detection system should consider the weather and time of the day.

2.1.2 User Network

● Users should be able to register to our application's system by filling the required information.

● Users should be able to update their user information.

● Users should be able to register to our application's system through their company emails for the associated organization.

2.2 Non-Functional Requirements

2.2.1 Usability

● ParkHound should provide two types of interface and for the regular users, it should have a user-friendly interface that only specifies necessary information for parking to empty slots.

● ParkHound should provide more information offering interface for security users.

● The application should include an explicit user manual that demonstrates how to use ParkHound.

2.2.2 Reliability

● The application needs to be stable and avoid any interruptions/crashes.

● The application needs to provide high accuracy in determining the parking status of the vehicles.

● The application needs to provide real-time data of the available parking spots.
●

2.2.3 Security

● The application needs to secure user information from any possible threats.

2.2.4 Performance

● The load time of the application should be low.

● The image recognition and connecting to server processes should be optimized so that users can quickly and easily look for the information they can access.

2.2.5 Extensibility

● To improve and modify our project, the datasets that we are going to use will be extendable.

● The application should be available on multiple platforms.

2.2.6 Portability

● Any personal computer with a browser and devices with an Android or iOS operating system will be able to run ParkHound.

3. REFERENCES

[1] "News." *Motorists Spend Nearly Four Days a Year Looking for a Parking Space*, www.britishparking.co.uk/News/motorists-spend-nearly-four-days-a-year-looking-for-a-parking-space.

[2] "The parking of the future: problems, challenges and solutions" https://circontrol.com/the-parking-of-the-future-problems-challenges-and-solutions/ [Accessed: 22 February 2020]

[3] "Is there a worldwide parking problem? "https://www.bbc.com/news/magazine-17271118 [Accessed : 22 February 2020]

[4] "React Native · A Framework for Building Native Apps Using React." *React Native Blog ATOM*, facebook.github.io/react-native/. [Accessed : 22 February 2020]