JES
Journal of Engineering Sciences

Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

PARKING SLOT INDICATOR

[1]R. Surender Reddy,[2]Murali Babu
[1]asst Prof,[2]Asso Prof [1,2]lords Institute Of Engineering And Technology, Hyderabad

ABSTRACT: Locating a parking spot during peak hours in most populated areas like shopping malls, universities, exhibitions or convention centers is difficult for the drivers. The difficulty rises from not knowing where the available spots may be at that required time. Smart parking is a solution to metropolitan cities to reduce congestion, cut vehicle emission totals and save persons' time by helping them in finding a spot to park. Smart Parking is a parking system, usually a new one that is equipped with special structured devices (things) to detect the available parking slots at any parking area. This is an application based on Internet of Things (IoT) that in Real-Time environment have sensors and devices embedded into parking spaces, transmitting data on the occupancy status; and the vehicle drivers can search for parking availability using their mobile phones or any infotainment system that is attached to the vehicle. Hence the driver would know where there is an available spot to park his vehicle in less time, reducing the energy consumption and air pollution. The Client or the sensor posts the parking slot occupancy status to a web service URL. The Java based web service is built using Spring and Hibernate to connect to the backend system. The web service (.war) file is deployed on Apache Tomcat Server and the backend used is MySQL database.

INTRODUCTION

Project Description

Seeking a vacant parking space during peak hours in areas like Hospitals, Hotels & Shopping Centers, Airports, Universities, and Exhibitions & Convention Center has always been frustrating for many drivers. Surveys says that traffic generated by cars searching for vacancies in Parking Spaces is up to 40% of the total traffic [1]. Now that is a serious issue to look after, and Smart Parking System is one of the best available solutions to at least reduce the traffic congestion caused due to the above problem. This application gives information about the occupancy status of the spaces in the parking lot equipped with sensors that detect the presence of vehicles.Smart Parking is an Internet of Things (IoT) based application, used to detect the available parking slots. This app uses ultrasonic sensor to detect the presence of a vehicle (whether the parking slot is occupied or not). Based on the parking slot occupancy, the status (occupied/unoccupied) is displayed on the web application dashboard. In real time, the environment have sensors and devices embedded into parking spaces, transmitting data on the occupancy status; and the vehicle drivers can search for parking availability using their mobile phones or any infotainment system that is attached to the vehicle. Hence the driver would know where there is an available spot to park his vehicle in less time, reducing the energy consumption and air pollution.

The second part in this application is doing analysis on parking trends in a parking lot. The analysis gives information about which parking space is most occupied and least occupied and at what times of the day. This information is helpful in choosing one parking space when there are

multiple available, keeping in mind the history of that space. For example, when there are more than one vacant slots the driver will want to choose the one that has less occupancy rate because the high occupancy rated slot might be wanted by many other drivers and you don't want to waste your time reaching that slot.

Motivation

A Smart Parking System like this helps drivers make smart decisions which will reduce congestion and make the most of available spaces. Finding a parking space has become a daily concern these days, and that is where the motivation for this project came up from. With the evolution of technology, we have smartphones, sensors that detect the presence of any object and my idea is having a system where parking spaces are equipped with these ultrasonic sensors that tells about the occupancy status of the parking spaces and a central management system that posts this occupancy status to a web application to guide the drivers in finding a vacant slot.

REQUIREMENT ANALYSIS

Requirement Gathering

As soon as the project idea is confirmed, I have started working on the requirements for the implementation of the project. The idea is to develop a web service that can receive information about the occupancy status of the parking space from Client (here the database) and post that information to the web application. Also passing information to database through web service URL and updating the changes in the system. I did some research on current technologies that are used in industry and decided on understanding how Spring Framework works, how to connect to database with Hibernate instead of JDBC, also seen some best practices in writing JavaScript.

Requirement Specification

These are the technical requirements to develop Smart Parking System web application.

Software Requirement

Operating System: Windows XP, Windows 7 or Windows 8

IDE: Eclipse with STS Plugin

Application Server: Apache Tomcat 7.0.67

Front End: HTML5, JavaScript, jQuery, AJAX and CSS

Frameworks and APIs: Spring and Hibernate

Web Service: RESTful web services

Database: MySQL

Browser: Chrome or Firefox or Internet Explorer

Hardware Requirement

Processor: Intel Core 2 Duo or Higher

RAM: 2GB

Users who want to use the application can directly deploy the war file in Tomcat server and run the app using localhost with the port number that is configured during installation of the web server. The path to the web application is passed along with localhost to the browser.

DEVELOPMENT BACKGROUND AND APPROACH

In order to understand our choice of frameworks we have to describe Smart Parking System's particular application requirements. This project has two primary components; a web service that receives the occupancy status from the client or the sensor and a web application that displays the parking slot occupancy status and also the parking history trends. The Java based web service is built with Spring and Hibernate to connect to the backend system which is MySQL database. The web service is deployed in Apache Tomcat Server. And the web application has front end designed using HTML5, CSS, JavaScript and jQuery. And has Ajax call to the web service URL.

Spring Framework [10]

The Spring Framework [2] is an application framework and inversion of control container for the Java platform. It has become popular as a replacement for, alternative to, or even addition to the EJB model. The Spring Framework provides about 20 modules which can be used based on an application requirement. The below figure shows typical Spring Framework which can be categorized into 4 groups: Core Container, Data Access/Integration, Web and Miscellaneous.

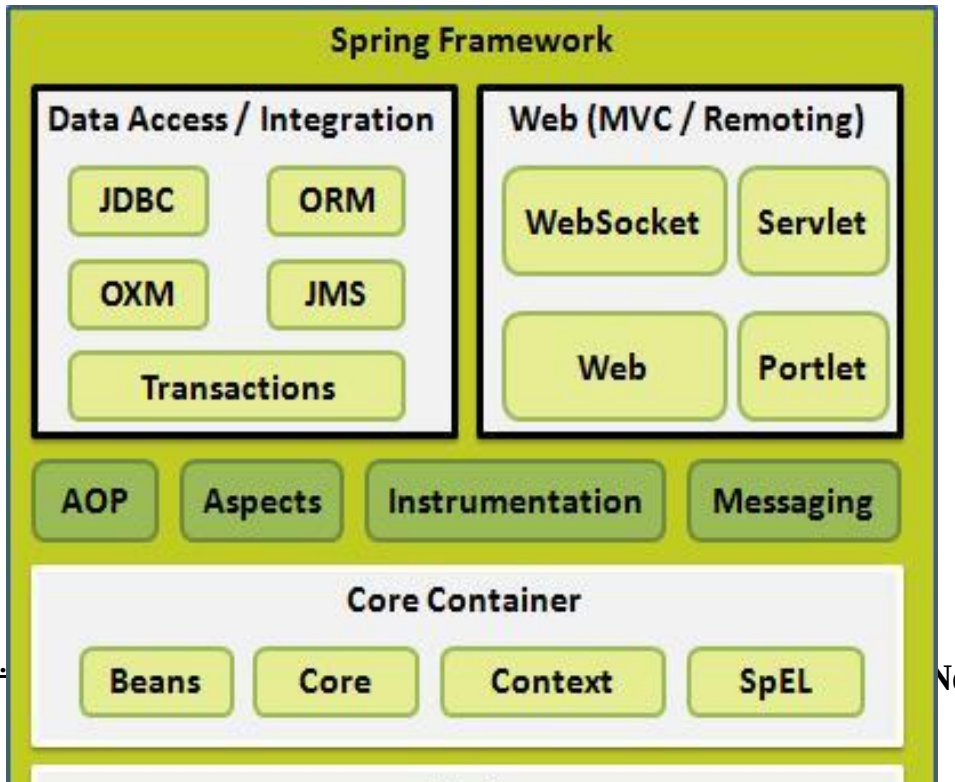

Figure - Spring Framework [3]

SYSTEM DESIGN

System Design

UML diagrams are used to explain the design of the system. Once the requirement gathering is completed, system design is done using Unified Modeling Language (UML). UML plays an important role in designing object oriented software by using graphical notations to depict the design of the system.

Use Case Diagram

In its simplest form, a use case can be described as a specific way of using the system from a user's (actor's) perspective. A more detailed description might characterize a use case as:

A pattern of behavior the system exhibits A sequence of related transactions performed by an actor and the system Delivering something of value to the actor

Use Cases provide a means to:

Capture system requirements Communicate with the end users and domain experts Test the system. The User of the system is a vehicle driver who would be searching for a vacant parking space, and the use cases are the sequence of actions that provide something of measurable value to the user like checking the parking lot for vacant spaces, checking the parking history to find the most occupied and less occupied parking slots. And finally after finding a vacant space, he can park his vehicle.

The below figure depicts the Use Case Diagram for the Vehicle Drivers:

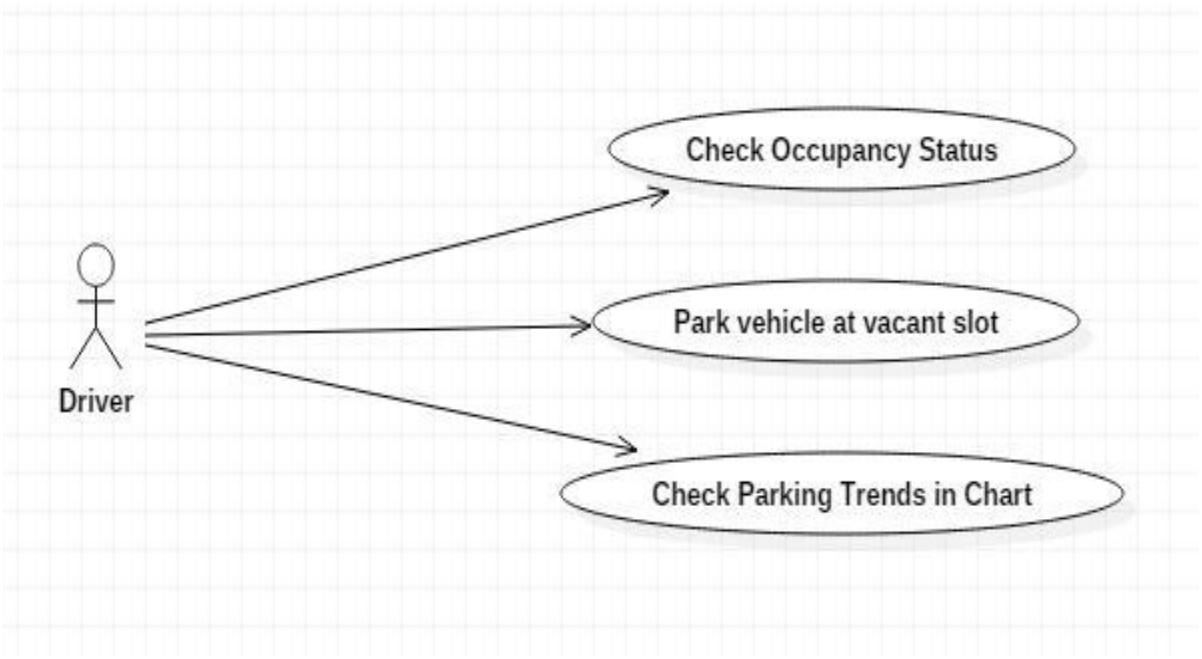

Figure - Use Case Diagram for Vehicle Driver

Class Diagram

A class diagram is a picture for describing generic descriptions of possible systems. Class diagrams and collaboration diagrams are alternate representations of object models. This contain icons representing classes, interfaces, and their relationships. We can also create one or more class diagrams to depict classes contained by each package in our model; such class diagrams are themselves contained by the package enclosing the classes they depict; the icons representing logical packages and classes in class diagrams.

Below is the main class diagram for the entire application. The diagram clearly explains about the relationships between two or more classes and also between classes and interfaces.

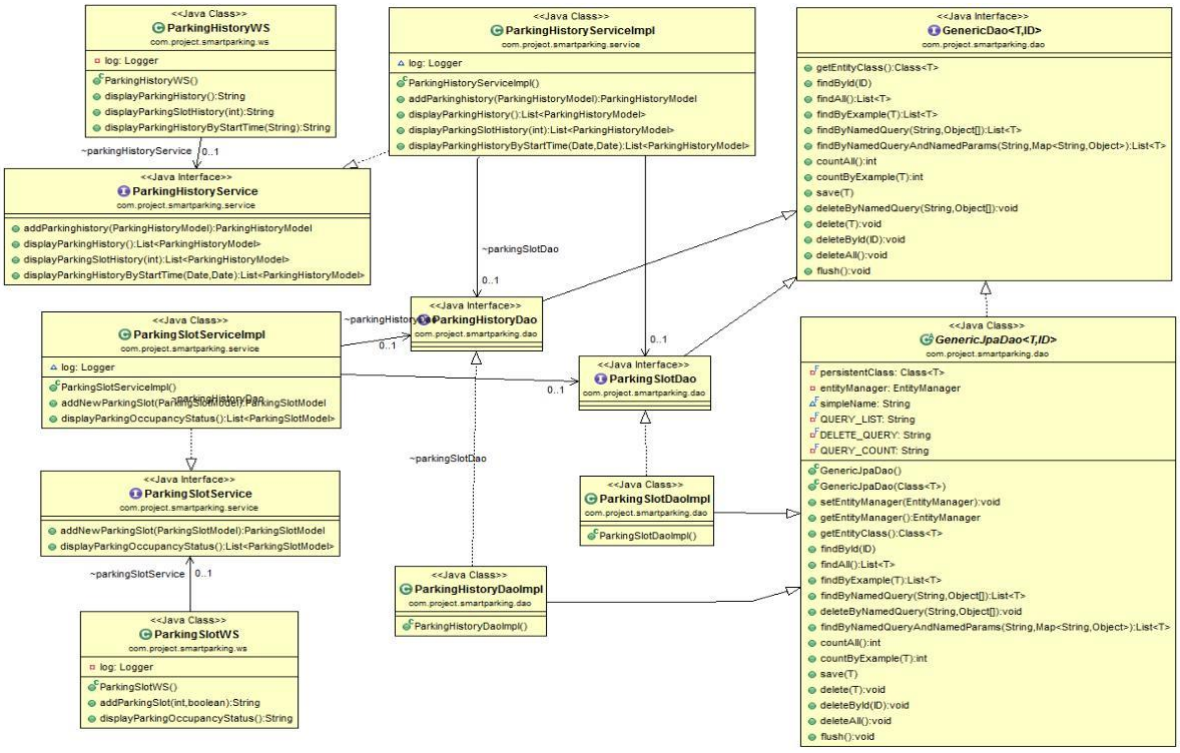

Figure - Class Diagram for Smart Parking System

IMPLEMENTATION

Smart Parking System is a web application that is developed to help drivers in finding a vacant parking space during any time of day. The main objective of the application is to provide the occupancy status (vacant/occupied) of the parking spaces and also to get information about

the parking history data on any particular date. The user is allowed to choose any date and can get the information related to parking spaces occupancy throughout that day.

The main features of the application are:Parking Slot Occupancy: This is the first part of the application where there is a dashboard that has occupancy statuses of the parking slots. The dashboard is refreshed every 2 sec and the latest information is displayed for the user.

Parking History Analysis: This is an extension to the application where User is able to see the occupancy rate of each parking lot on any particular day in the form of charts. Using this data, driver can choose the less occupied slot when there are more than one slot available. Also the parking lot owners can rent the spaces by charging based on their occupancy rates at any particular time of the day. The Smart Parking Application is developed on Eclipse IDE with STS Plugin using Java 1.7. The user interface is designed using JavaScript and the business logic I coded in Java. The application also makes use of annotations in Java, especially Spring Framework and Hibernate annotations. The backend database is MySQL and I have used MySQL Query Browser in order to manage the database. The web application developed is archived to a war file and then deployed in Tomcat Server. The total number of lines of code breakdown with respect to language is listed below:

Language	Number of LOC
Java	1677
XML	132
HTML5, CSS	169
JavaScript, jQuery	164
Total	2142

Table Lines of Code

Output Screens

The Graphical User Interface for this application is designed by keeping in mind that the application's user interface is everything that the user can see and interact with. Technologies used to develop this user interface are mainly JavaScript, jQuery, HTML5 and CSS. To interact with backend web service written in Java, jQuery methods get() and post() are mainly used. Ajax calls are made every 2 seconds by passing URL to be hit, and the required data from the database is retrieved and passed back to the caller. Accordingly the Occupancy Status Message and

Occupancy Indicator in the front end are updated every 2 seconds. Similar thing is with the chart showing the parking history. JavaScript FusionCharts are used to show the occupancy of parking spaces starting from zeroth hour to 24th hour. Here we can see the parking history of any day, by just setting the required date by using the calendar provided.

Test Case	Expected Result	Result
After the start of Tomcat Server, on the load of html file	Display correct occupancy status messages for all the parking slots	Pass
After the start of Tomcat Server, on the load of html file	Display correct occupancy status indicators for all the parking slots	Pass
After the start of Tomcat Server, on the load of html file	Display Red indicator for 'Occupied' Status Message	Pass
After the start of Tomcat Server, on the load of html file	Display Green indicator for 'Vacant' Status Message	Pass
Upon receiving occupancy status change from client for parking slot 1	Change in the parking slot 1 only, and should not affect other parking slot status messages.	Pass
Upon receiving occupancy status change from client for parking slot 1	Change in the parking slot 1 only, and should not affect other parking slot status indicators.	Pass
Upon receiving occupancy	Change in the parking slot 2 only,	Pass

status change from client for parking slot 2	and should not affect other parking slot status messages.	
Upon receiving occupancy status change from client for parking slot 2	Change in the parking slot 2 only, and should not affect other parking slot status indicators.	Pass
Upon receiving occupancy status change from client for parking slot 3	Change in the parking slot 3 only, and should not affect other parking slot status messages.	Pass

CONCLUSION AND FUTURE WORK

Conclusion

Smart Parking System is a solution to the existing traffic congestion, to reduce drivers' frustration by providing information about the occupancy status of the parking spaces. The project development went smoothly while teaching me many best practices in programming using the current trending technologies like Spring Framework, Hibernate ORM and REST APIs. I could see that all the initial requirements of the project are achieved and also I tried doing minor data analysis on the parking spaces occupancy statuses. The web application is user friendly; any user can easily find the status (vacant/occupied) of the parking space and also can set the required date to see the parking history. Starting from coming up with project idea, understanding the requirements and choosing the best technologies for the implementation, all this gave me very good experience and exposure in development of a full stack web application. Upon completing this project successfully, I got familiar with Eclipse IDE Shortcuts, Database tools and Tomcat server usage.

Future Work

This application is an initial step in reaching the effective solution for the daily concern.

This project can be extended in multiple ways: To provide a central management system that make sure only authenticated information is sent to the Client, i.e. dealing with the security issues. More analysis can be done using the parking history data by which User can get

recommendations or suggestions on parking spaces and their availability trends. And this analysis can be used while reserving a parking space by User or while renting a space, to decide the price of the parking space. We could also do a mobile application through which driver can get the occupancy statuses of the parking spaces.

BIBLIOGRAPHY

1. A Reservation – based Smart Parking System by Hongwei Wang and Wenbo He – April 01, 2016 http://cse.unl.edu/~byrav/INFOCOM2011/workshops/papers/p701-wang.pdf

2. Spring Architecture – April 01, 2016 https://sites.google.com/site/sureshdevang/why-spring framework

3. Spring Framework – Architecture – April 01, 2016 http://www.tutorialspoint.com/spring/spring_architecture.ht
4. Hibernate Architecture – April 02, 2016 http://www.tutorialspoint.com/hibernate/hibernate_architecture.ht

5. Hibernate ORM – April 02, 2016 http://www.javabeat.net/hibernate-interview-questions/

6. JavaScript Basics – April 04, 2016 http://www.w3schools.com/js/default.asp

7. JQuery Methods – April 04, 2016 http://api.jquery.com/jquery.post/

8. Apache Tomcat – April 04, 2016 https://en.wikipedia.org/wiki/Apache_Tomca

9. Apache JMeter – Performance Testing – April 9, 2016 http://jmeter.apache.org/

10. Spring Framework Introduction – April 01, 2016

11. https://docs.spring.io/spring/docs/current/spring-framework-reference/html/overview.html