ParkingSlot

CZ2006 Project (Team 3,App Monsters)

Done By: Jia Wei, Chun Foong, Emmi, Jun Le, Andrew, Sabrina

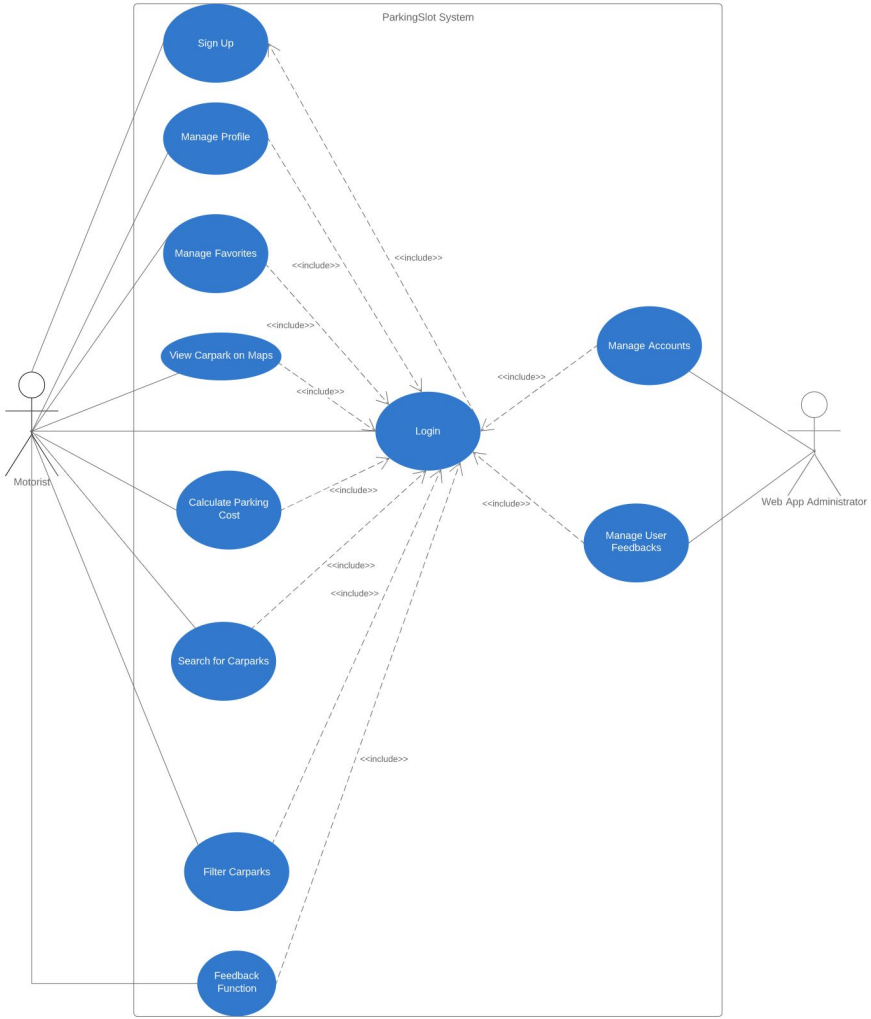
Overview

1. Introduction
2. Use Case Model
3. Main Functionalities
4. Demonstration of Working Features
5. Traceability of Program
6. Good Software Engineering Practices

Introduction

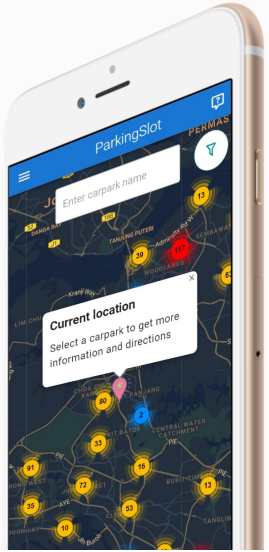
- Project Background
  - Used carpark data from data.gov.sg as part of a Smart Nation initiative
  - LTA, URA and HDB APIs to fetch up-to-date parking rates and real-time availability of lots
- ParkingSlot aims to solve the frustrating daily parking problems of motorists :
  - Time wasted for finding parking space
  - Uncertainty over parking charges
- Expected Users:
  - People who own a vehicle
  - Friends/relatives of people who own a vehicle

Use Case Model

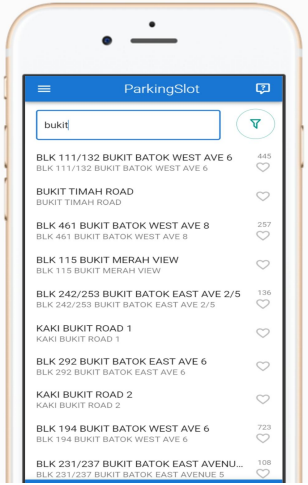


Main Functionalities

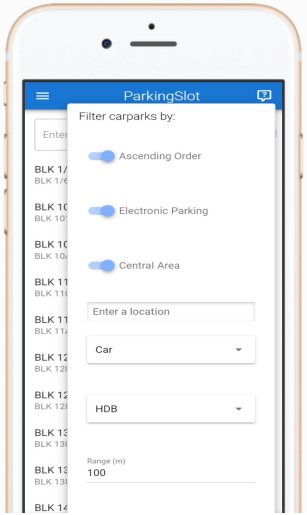
Carpark information at your fingertips

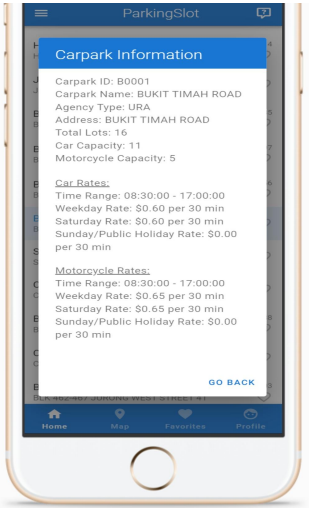


Search for carparks you wish to go to



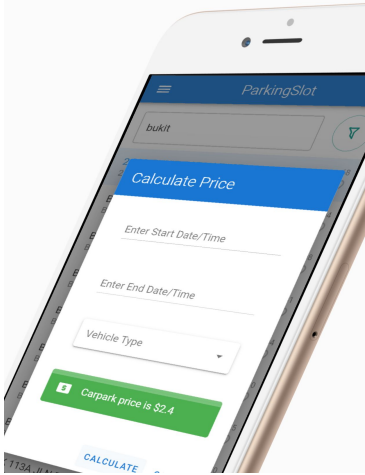
Filter search with your desired choice





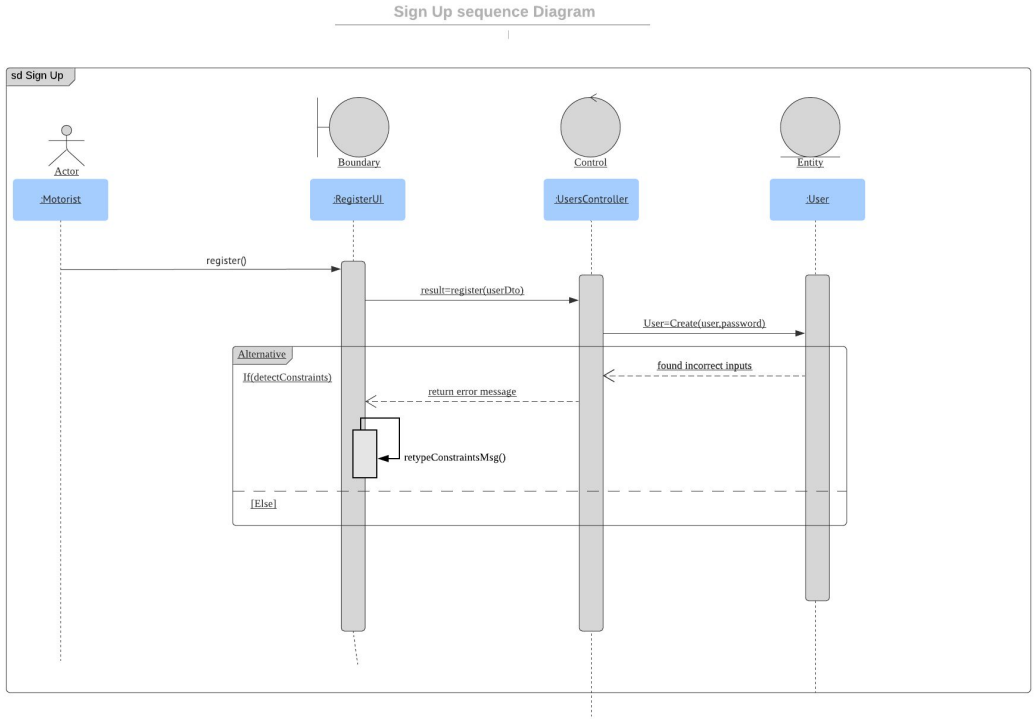
View real-time availabilty and parking rates

Calculate your estimated parking costs



Demonstration of ParkingSlot App

Traceability of Program - Register Sequence Diagram



Traceability of Program - Register Black Box Test Cases

-

Test ID	Description	Expected Result	Actual Result
1	Precondition: user click “Register” button 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in Email 5. Key in phone number 6. Key in Password 7. Key in Confirm Password 8. Click “Sign Up” button	Successful account registration. User will be on Login Page.	Pass
2	Precondition: user click “Register” button Precondition: existed username 1. Key in First Name 2. Key in Last Name 3. Key in existed Username 4. Key in Email 5. Key in phone number 6. Key in Password 7. Key in Confirm Password 8. Click “Sign Up” button	Unsuccessful account registration. Display “ Username “(username)” is already taken”	Pass
3	Precondition: user click “Register” button Precondition: existed email 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in existed Email 5. Key in phone number 6. Key in Password 7. Key in Confirm Password 8. Click “Sign Up” button	Unsuccessful account registration. Display “Email “(email)” is already taken”	Pass
4	Precondition: user click “Register” button 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in Email without ‘@’ 5. Key in phone number 6. Key in Password 7. Key in Confirm Password 8. Click “Sign Up” button	Unsuccessful account registration. Display “Please include an ‘@’ in the email address. ‘(email)’ is missing an ‘@’.”	Pass
5	Precondition: user click “Register” button Precondition: existed phone number 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in Email 5. Key in phone number 6. Key in Password 7. Key in Confirm Password 8. Click “Sign Up” button	Unsuccessful account registration. Display “Phone number “(phone number)” is already taken”	Pass

Traceability of Program - Register Black Box Test Cases

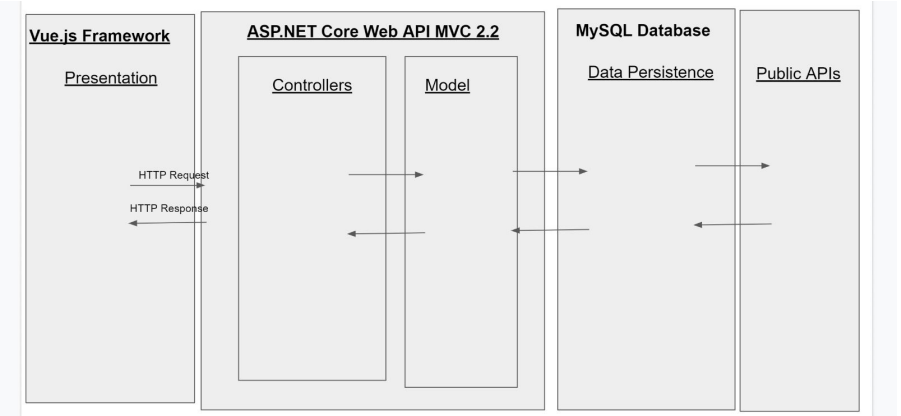
6	Precondition: user click "Register" button 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in Email 5. Key in phone number with length less than 8 6. Key in Password 7. Key in Confirm Password 8. Click "Sign Up" button	Unsuccessful account registration. Display "Please match the requested format."	Pass
7	Precondition: user click "Register" button 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in existed Email 5. Key in phone number 6. Key in Password 7. Key in Confirm Password with different characters than previous password 8. Click "Sign Up" button	Unsuccessful account registration. Display "Password does not match"	
8	Precondition: user click "Register" button 1. Key in First Name 2. Key in Last Name 3. Key in Username 4. Key in existed Email 5. Key in phone number 6. Key in Password with length less than 8/ no uppercase/no lower case/ no special characters/ no digit 7. Key in Confirm Password 8. Click "Sign Up" button	Unsuccessful account registration. Display "Passwords must be at least 8 characters and contain at 3 of 4 of the following : upper case (A-Z), lower case (a-z), number (0-9) and special character (e.g. !@#$%^&*)"	

Good Software Engineering Practices (Frontend)

- Vue Javascript Framework used to create user interface
  - Decomposing components for **reusability**
    - Navigation Bar, Tab Bar, Search, FIlter
- Consistent and responsive design
  - Material Design component framework Vuetify
- Efficient page loading
  - Lazy loading components
- **Separation of concerns**
  - Admin, Auth, Layout, Plugin, Store
- User Experience
  - Validation and feedback

Good Software Engineering Practices (Backend)

- 3 Main Components
  - Public API, Database and Private API
- Population of data into database from Public API
  - Data cleaning and storing in database for ease of access for CRUD operations
- Private API built by ASP.NET Core Web API Framework
  - **Model-View-Controller Pattern**
- Database Manipulation
  - Object Relational Mapping technique
- Data Access Object pattern
  - **Isolates application layer from the persistence layer**



Thank you! :)