Application: ParkIT!

Mission Statement: ParkIT aims to make parking more efficient by guiding users to find suitable car parks near their destinations, saving them the time needed to search for available lots. Our platform enables users to plan their journeys, track parking expenses, and have a stress-free parking experience.

Functional Requirements

1	The web application must allow the first-time user to create a new account.
1.1	The web application must require the user's credentials for account creation
1.1.1	The web application must require the user's name.
1.1.2	The web application must require the user's email.
1.1.3	The web application must require the user's password.
1.2	The web application must validate the uniqueness of the email address to ensure that each email is associated with only one account.
1.3	The web application must ensure that the password entered meets security criteria.
1.3.1	The password must have a minimum of 8 characters.
1.3.2	The password must have at least 1 lowercase letter.
1.3.3	The password must have at least 1 uppercase letter.
1.3.4	The password must have at least 1 special character.
1.4	Once an account is created, the web application must log the user in.
2	The web application must allow the user to login.
2.1	The web application must allow existing users to log in using their registered credentials.
2.1.1	The web application must require the user's registered email.
2.1.2	The web application must require the user's registered password.
2.2	The web application must validate the login credentials and grant access provided email and password are correct.
2.2.1	The web application must display an error message when the login credentials are incorrect.
2.3	The web application must allow the user to log out of their account.

2.3.1	The web application must redirect them to the login page upon logout.
3	The web application must present a tutorial to the new user upon their first successful account creation and login.
3.1	The web application must provide an option to skip the tutorial for the user who does not wish to view it.
3.2	The tutorial must provide visual and text-based instructions to help new users navigate the application.
3.3	The web application must track whether the user has completed the tutorial.
3.3.1	The web application must not prompt the user to view the tutorial again upon subsequent logins after the completion of the tutorial.
3.3.2	The user must be directed to the home page after skipping or finishing the tutorial.
4	The user must be able to input a destination using the web application.
4.1	The web application must use destination and search for car parks within a set distance (1km by default).
4.1.1	The web application must query the details of car parks within the set distance.
4.2	The web application must display information about the nearby car parks.
4.2.1	The web application must show car parks available as icons on the map, and details must be shown when selected.
4.2.1.1	The car park details displayed must consist of car park availability.
4.2.1.2	The car park details displayed must consist of parking rates.
4.2.1.3	The car park details displayed must consist of the distance from the destination.
4.2.1.4	The car park details displayed must consist of service time.
4.2.1.5	The car park details displayed must consist of vacancy.
4.3	Users can find a suitable car park by using the filter.
4.3.1	Users can select one, some, all, or none of the filter options.
4.3.1.1	One of the filter options must include car park availability.
4.3.1.2	One of the filter options must include parking rates.
4.3.1.3	One of the filter options must include distance from the destination.
4.3.1.4	One of the filter options must include service time.

4.3.1.5	One of the filter options must include vacancy.
4.4	The icons of each car park must display a different colour based on its vacancy.
4.4.1	A green icon indicates that the car park still has vacancies.
4.4.2	A red icon means that the car park is full.
4.4.3	Vacancies of car parks are tracked live.
4.5	The user can select their preferred choice of car park by clicking on that car park.
4.5.1	The web application will redirect the user to Google Maps to find the best route to the car park
5	The user can check in and out of the web application when at a car park.
5.1	When safely parked, the user can check in the web application to create a time log.
5.1.1	The web application must add a new parking detail on “history” with the button “checkout” which indicates that this parking event has not been checked out.
5.2	If the vacancy of the car park is full, a notification must appear recommending another car park.
5.2.1	Clicking on the notification must redirect the user to Google Maps to find the best route from the current car park to the recommended one.
5.3	The user can check out of the web application when exiting the car park to end the time log.
5.3.1	The web application must calculate the expense based on duration (Time In and Time Out) and car park rate when the user clicks on “checkout”.
5.3.2	The web application must save expenses in the “Expense Dashboard”.
5.3.3	The web application must update the “checkout” button in the “history” to the time the parking event was checked out.
6	The user must be able to view individual car park expenses on the “Expense Dashboard”.
6.1	The dashboard must display the total current expenditure up to date.
6.2	The dashboard must display the average expenditure for some range of time.
6.2.1	One option is to display the average expenditure in a day.
6.2.2	One option is to display the average expenditure in a week.
6.2.3	One option is to display the average expenditure year.

7	The user must be able to view individual car park history on a “history” dashboard.
7.1	The “history” dashboard must keep records of the parking event details for a year.
7.1.1	Each event detail on the “history” dashboard must contain the name of the car park
7.1.2	Each event detail on the “history” dashboard must contain the address of the car park
7.1.3	Each event detail on the “history” dashboard must contain a “check in” time.
7.1.4	Each event detail on the “history” dashboard must contain a “check out” time.
7.1.4.1	If the parking event has not checked out yet, the event shows a “check out” button for the user to check out.
7.1.5	Each event detail on the “history” dashboard must contain expenses.
8	The user must be able to add car parks to a list of favourites.
8.1	The user must be able to see a list of car parks when they search for the desired car park.
8.2	The web application must display an “Add to Favourites” icon for each car park in the list.
8.3	The user must be able to select the "Add to Favourites" icon for a car park.
8.4	The web application must save the car park to the user's favourites list if the user selects the "Add to Favourites" icon for a car park.
9	The web application must allow the user to submit feedback.
9.1	The user must be able to click on the “Feedback” icon to submit feedback.
9.2	The web application must direct the user to a page to submit feedback.
9.3	The user must be able to go back to the previous page even if they do not submit feedback.
9.3.1	The web application must display a “go back” icon on the feedback submission page
9.3.2	If the user clicks on the “go back” icon, the web application must redirect the user back to the previous page.
9.4	The web application must display a feedback form on the feedback page for the user.
9.4.1	The user must be able to type in characters in the form.

9.4.2	The user must be able to attach images in the form.
9.4.3	The user must be able to attach video files in the form.
9.5	The user must be able to click on “Submit” to submit the feedback.
9.6	The user must be able to go back to the web application after submitting feedback.
9.6.1	After submission, the web application must display a “Feedback Successfully Submitted!” message upon successful submission.
9.6.2	The web application must display a “go back” icon to redirect the user back to the web application page after submission.

Non-Functional Requirements

Usability	80% of first-time users must be able to view information of the car park and its vacancies within 2 minutes of starting to use the web application.
	The user must be able to operate the web application safely with one hand, while on the road.
Performance	When the user opens the web application or inputs their destination or redirects the user to Google Maps, the system must respond and/or display the correct information within 2 seconds.
Reliability	The web application must provide accurate and up-to-date information on parking lot vacancies, parking rates, service times, distance and duration of user to car park.
	The web application must have an uptime of 99.9%, ensuring that it is available to users at all times.
Accuracy	The web application must update the information on car park vacancies and duration to car park within 20 seconds of the API updating its information, even when a user is currently viewing an image.
	The web application must display parking rates accurate to the current hour.
Security	The web application must encrypt user data into the database.
	The web application must follow best practices for secure authentication.
Supportability	The web application must support multiple languages based on the user’s locale.
	The web application must be compatible with the latest versions of web browsers.

Maintainability	The web's codebase must be modular and well-documented to facilitate easy updates and bug fixes.
	The web is designed so that testing can be performed easily and efficiently which does not introduce new bugs when any changes are made.

Data Dictionary

Term	Definition
User	An individual who owns a vehicle in Singapore and is making use of the web application to find availability of parking slots.
Account	A unique identity created by a user consisting of a name, email, and password. No two accounts can have the same email.
HDB	Housing Development Board. It is a statutory board under the Ministry of National Development responsible for public housing in Singapore. They provide a real-time API for the availability of the parking slots updated every minute.
Car Park	A location that is legally designated for parking.
Car Park Details	Information about a car park, including address, parking lot availability, parking rates, distance from current location, and service time.
Check In	The action a user takes to log their parking start time at a car park.
Check Out	The action a user takes to log their parking end time at a car park, which calculates the parking expense.
Favourites	A list of car parks that a user has marked for easy access and future reference.
Parking Rate	The fee that must be paid upon exit of a parking space. This fee is usually metered based on the duration of use and varies between parking spaces and days of the week.
Parking Expense Dashboard	A visual display showing the user's total expenses up till date, and average monthly spending. Bar charts must be shown for easier visualisation for trends.
Login Credentials	A User ID and password combination that allows users to access a website or application.
Query	A request for data from database.