Swinburne University of Technology

School of Science, Computing and Engineering Technologies

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: <u>SWE30003</u>   Unit Title: <u>Software Architectures and Design</u>

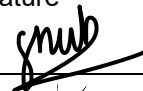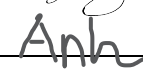Assignment number and title: <u>1, Requirements</u>   Due date: 9th June 2024

Tutorial Day and Time: __Friday 11:00-12:00__   Project Group: _9_____

Tutor: ___Dr. Le Minh Duc_____

To be completed as this is a group assignment
We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature
104222196	Ta Quang Tung	
104169507	Nguyen Quang Huy	
104177513	Tran Hoang Hai Anh	
104072029	Phan Sy Tuan	

Marker's comments:

Total Mark:_____

Extension certification:

This assignment has been given an extension and is now due on _____

Signature of Convener:_____

1 - Introduction

This specification document details the requirements for a Smart Parking information system to be developed at Duy Tan Street. This system will help drivers to book parking slots online conveniently and the owners to better monitor and manage the parking facilities.

This document is structured as follows: **Section 2** discusses the business, the issues with its current operations, and the goals, assumptions, and scope of the new online system. **Section 3** describes the problem domain and domain model. **Section 4** details the functional requirements of the new system with the Task and Support method. **Section 5** illustrates the basic workflows of the system. **Section 6** lists five core quality requirements of the system. **Section 7** discusses additional requirements such as design or product requirements. **Section 8** shows our validation of the requirements using the CRUD check. Finally, **section 9** suggests two possible solutions based on the specified requirements.

2 - Project Background

Overview

Customer	SmartParking company
Background	SmartParking offers paid parking spaces at Duy Tan Street for cars and motorbikes. Under their current system, drivers must physically locate and pay for their parking spots on arrival, resulting in long wait times and the possibility of being turned away. The company also needs an easy way to upgrade and monitor the operation of the parking lot.
Purpose	This project aims to develop an Online Smart Parking Space (OSPS) system for SmartParking. The system will let drivers pre-book slots, manage bookings, handle payments, and receive parking guidance. It will also allow the company owners to better monitor and manage the parking lot.
Project Type	Tender

Pain points

The following are the "pain points" of SmartParking's current system, grouped into two areas:
- Parking services:
  - Drivers have to locate and pay for their parking spots on arrival.
  - Drivers face long waiting times at peak hours and cannot park on time.
  - Some drivers are turned away after all spots are occupied.
  - The parking services are only available offline.
- Administration:
  - The owners currently cannot monitor the statistics of the parking slots.

Goals

Based on the pain points above, the project sets the following goals for the new Online Smart Parking Space (OSPS) system:
- Allow drivers to pre-book parking slots for a faster and more convenient parking experience.
- Provide a seamless parking experience when drivers arrive at the parking lot.
- Provide drivers with a fast and secure payment flow that supports invoice and receipt generation.
- Help SmartParking establish an online presence and broaden its customer base.
- Allow SmartParking to oversee its operations and upgrade its parking lot as new facilities are available.

Assumptions

The following assumptions are made about SmartParking the Online Smart Parking Space (OSPS):
- Currently, SmartParking only has one location in Duy Tan Street and the parking lot has up to 500 slots.
- The number of drivers entering/leaving the parking lot can peak at 300 twice a day, once in the morning from 7:30 AM to 8:30 AM and once in the afternoon from 5:00 PM to 6:00 PM. Outside these peak periods, the average number of drivers per hour is 3/hr.
- Drivers don't need to use the online system to be able to park at SmartParking. They can still arrive at the parking lot and park like before.
- Slot bookings must be pre-paid online. The only allowed payment method for bookings is credit.
- If the driver does not book in advance, they can pay for their parking slot on departure. In this case, they can pay by either cash or credit.
- SmartParking charges for the parking slot based on the type of vehicle and the duration of the parking session.

Scope

The SmartParking company wants the new system to shorten the queuing time for the driver as well as inform all the users about the availability of the parking lots. After creating a new account in the system, the driver can manage their parking slots, pay for their services, and get the invoice and receipt back. The system will also be able to update its information using real-time data and display it over various periods. Besides that, the system also needs to have spare room for future system expansion as they intend to update the current facility in the future.

3 - Problem Domain

Domain vocabulary

This section defines some important terms that will be used throughout this document:
- OSPS: Online Smart Parking Space. This is the system that the SmartParking company wants to develop.
- Customer: Refers to the driver who uses the parking services offered by SmartParking. It is used interchangeably with "driver"; however, "driver" will be preferred whenever possible for clarity.
- Vehicle: The physical vehicle that the customer drives to and parks at SmartParking. Currently, two types of vehicles are supported: cars and motorbikes. In the future, the company may add more types.
- Parking lot: The area owned by SmartParking where drivers can park their vehicles.
- Parking space/parking slot: A portion of physical space in the parking lot where one vehicle can be parked (a car or motorcycle).

Actors

The following are the "actors" who will be interacting with the system:
- Drivers: These are the customers of SmartParking who will be using the parking services.
- Owners: The owners of SmartParking would like to use the system to monitor their parking facilities.
- Guards: Guards are responsible for checking who enters and leaves the parking lot and handling payment.

Domain model and entity descriptions
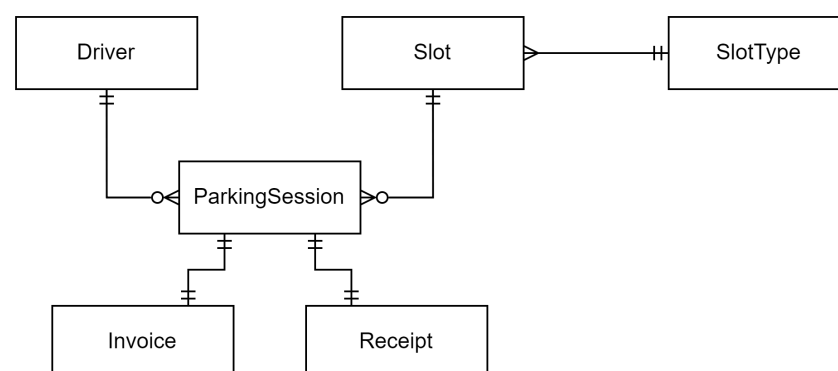
- Driver: This entity represents both the driver and their account on the system.
- Slot: This entity represents a parking slot. A slot is tied to one slot type.
- SlotType: This entity represents the type of a parking slot. Currently, two types are supported corresponding to the two supported vehicles: cars and motorbikes.
- ParkingSession: This entity represents a parking session made by the driver. This parking session can be pre-booked or started on arrival.

- Invoice: This entity represents an invoice tied to a particular booked parking session. Invoices are not generated for walk-in parking sessions as it is unnecessary.
- Receipt: This entity represents a receipt tied to a particular parking session. It is proof that the driver has paid for their parking time at SmartParking.

Tasks

These are the tasks that the OSPS must support, divided into two work areas:
1. Parking service
    1.1. Create driver account
    1.2. Update driver account
    1.3. Pre-book a parking slot
    1.4. Change a booked slot
    1.5. Park vehicle
    1.6. Pay for booking
    1.7. Pay for walk-in
2. Parking lot administration
    2.1. View booking statistics of the parking lot
    2.2. Change existing parking slot
    2.3. Add a new parking slot

4 - Functional Requirements

Work area 1 - Parking service

General	This area covers the core business activities of SmartParking. SmartParking drivers can either pre-book a slot or get allocated one on arrival. Choosing the former requires the driver to have an account on the system but lets them reserve a spot and conveniently pay online. Choosing the latter may result in the driver being turned away when the lot is out of space. Activities in this work area can take place at any time of day, although the bulk happens during the morning and afternoon rush hours and in the evening.
Actors involved	1. **Drivers** that want to use the parking services. They may have diverse IT backgrounds, but all are assumed to be average mobile/computer users. 2. **Guards** are involved in some of the tasks. They have very low IT skills.

Task 1.1 - Create driver account

Purpose	Grant the driver access to the system, a prerequisite for activities such as booking slots and updating bookings.
Trigger	A new driver would like to use the system and access it for the first time.
Frequency	● 30 registrations/day during the first week of deployment. ● Average 5 registrations/day afterward.
Critical	A large influx of new users (approximately 200) at the system launch.

Sub-tasks	Example solution
1 - Provide the driver's signup credentials.	The system provides a user interface through which the driver can type in their credentials such as email, phone number, and password.
2 - Verify signup credentials.	After the driver submits their credentials, the system checks their validity. The system could compare them against an existing database to ensure the email or phone number has not been used, for example.
3 - Provide the driver's personal information.	Similar to the example solution of subtask 1.
4 - Verify personal information.	Similar to the example solution of subtask 2.
5 - Record the new account information.	The system saves the newly created account into the database.

Variants	Example solution
2a - Signup credentials are invalid.	The system displays an error message and prompts the driver to enter new valid credentials.
4a - The driver's personal information is invalid.	Similar to the example solution of variant 2a.

Task 1.2 - Update driver account

Purpose	Change the credentials and/or personal information of the driver's account.
Trigger/precondition	A driver wants to update his account information. A driver has already logged into the system.
Frequency	10 updates/month.
Critical	None

Sub-tasks	Example solution
1 - Provide new credentials (optional).	The system provides a user interface through which the driver can type in their new credentials such as password.
2 - Verify credentials (optional, only if 1 was performed).	After the driver submits their new credentials, the system checks their validity. The system could compare them against an existing database to ensure the email or phone number has not been used, for example.
3 - Provide new personal information (optional).	Similar to the example solution of subtask 1.
4 - Verify personal information (optional, only if 3 was performed).	Similar to the example solution of subtask 2.
5 - Record the change.	The system saves the new changes into the database.

Variants	Example solution
2a - Signup credentials are invalid.	The system displays an error message and prompts the driver to enter new valid credentials.
4a - The driver's personal information is invalid.	Similar to the example solution of variant 2a.

Task 1.3 - Pre-book a parking slot

Purpose	Let the driver book a parking slot in advance to ensure it is available on arrival.
Trigger/precondition	A driver wants to book a parking slot. The driver has already logged into the system.
Frequency	200 bookings/day.
Critical	Drivers are most likely to book slots the evening before arriving at the parking lot, resulting in more booking traffic at this time (about 150 requests).

Sub-tasks	Example solution
1 - Select a parking slot.	The system displays a map of the parking lot showing selectable slots. The driver then clicks on a slot to select it.
2 - Provide the arrival and departure times.	The system provides an interface through which the driver can specify their parking time.
3 - Pay for the booking (see subtask 1.6).	After confirming the slot location and parking time, the user proceeds to the payment screen. See subtask 1.6 for an example solution.
4 - Record the booking.	The system adds the booking information to the database and marks the slot as booked for the specified duration.
Variants	**Example solution**
2a - The slot is already occupied at the specified time.	The system displays an error message and prompts the driver to select a different time.

Task 1.4 - Change a booked slot

Purpose	Update the time and location of the booked slot or cancel it.	
Trigger	The driver has already successfully booked a slot and wants to update it.	
Frequency	2 updates/day.	
Critical	None.	
Sub-tasks	**Example solution**	
1 - Select the booked slot.	The system displays a list of bookings made by the driver. The driver then selects the booking that they want to change.	
2 - Update slot time/location (optional).	The system provides an interface through which the driver can specify their new parking time and/or location.	
3 - Cancel the booking (optional).	The driver clicks on the "Cancel" button on the interface.	
4 - Record the update.	The system updates or deletes the booking entry in the database.	
Variants	**Example solution**	
2a - The desired time or location is not available.	The system displays an error message and prompts the driver to select a different slot or time.	

Task 1.5 - Park vehicle

Purpose	Let the driver park their vehicle.
Trigger	A driver arrives at the parking lot.
Frequency	400 times/day.
Critical	The number of drivers arriving at the parking lot can peak at 300 between 7:30 AM - 8:30 AM.

Sub-tasks	Example solution
1 - Allocate an unoccupied slot to the driver.	The guard checks the system, which displays a map of parking slots and their statuses (empty, occupied, booked), and informs the driver of an empty slot.
2 - Guide the driver to the slot.	The driver clicks on the slot on their mobile phone, and the system will give them directions.
3 - Record the slot as occupied.	A sensor at the slot registers when the driver arrives and informs the system to mark the slot as occupied.
Variants	**Example solution**
1a - No unoccupied slots are available.	The guard will have to turn the driver away in this case.
1b - The driver has pre-booked a slot.	Slot allocation on arrival is no longer necessary, and the system will guide the driver to their booked slot on their mobile phone.
2a - The driver does not park at their allocated slot.	When a slot is allocated or booked, the system will record the driver's license plate. If the system finds it at a different slot, it will notify the guard, who will then come and guide the driver to the correct slot.

Task 1.6 - Pay for booking

Purpose	Charge the driver for their booking.
Trigger	The driver has selected their location and time slot and needs to pay for the booking.
Frequency	200 times/day.
Critical	Drivers are most likely to book slots the evening before arriving at the parking lot, resulting in more booking traffic at this time (about 150 requests).

Sub-tasks	Example solution
1 - Generate invoice.	The system calculates the parking fee based on the vehicle type and duration and then sends the invoice to the driver's phone.
2 - Handle payment.	The system provides an interface through which the driver can pay for the booking with their bank account.
3 - Generate receipt.	The system prints or sends a receipt to the driver's phone as a confirmation of payment.
4 - Record payment.	The system records the receipt in the database as proof of payment.
Variants	**Example solution**
2a - The payment method is invalid.	The system shows an error message and prompts them to enter a different bank account.
2b - The driver has insufficient funds.	Similar to variant 2a.

Task 1.7 - Pay for walk-in

Purpose	Charge the driver for their parking slot.	
Trigger/precondition	● A driver leaves the parking slot. ● This driver **had not** booked their slot.	
Frequency	400 times/day.	
Critical	The number of drivers leaving the parking lot can peak at 300 between 5:00 PM - 6:00 PM.	
Sub-tasks	**Example solution**	
1 - Determine the slot to be paid for and calculate the fee.	The system shows the guard the parking slot and duration corresponding to the driver's license plate. It then calculates the fee that has to be paid based on the vehicle type and parking duration.	
2 - Handle payment.	The guard informs the driver of the amount and asks them to pay either in cash or by credit.	
3 - Generate receipt.	The system prints a receipt as a confirmation of payment.	
Variants	**Example solution**	
2a - The payment method is invalid.	The guard will ask the driver to choose another payment method and prevent the driver from leaving until they have paid successfully.	
2b - The driver has insufficient funds.	Similar to variant 2a.	

Work area 2 - Parking lot administration

General	This work area covers administrative tasks such as monitoring parking statistics and making changes to the parking lot. It is often done periodically, once or twice a month depending on the needs of the business.
Actors involved	The business owners, who have moderate to high IT skills.

Task 2.1 - View booking statistics of the parking lot

Purpose	Monitor how frequently different parts of the parking lot are being booked to optimize resources and make potential upgrades.
Trigger	It is the end of the month and the owners want to check the booking statistics to review their performance.
Frequency	Once a month.
Critical	None.

Sub-tasks	Example solution
1 - Select the desired time period (i.e. day, week, or month).	The system provides an interface where the owner can specify the time range they want to monitor.
2 - Retrieve the statistics within the specified time.	The system queries data for the specified period from the database.
3 - View the statistics.	The system formats the queried data for easier reading. For instance, it can present the data as a heat map of the parking lot to show the most frequently booked areas. When the owner clicks on a particular slot, detailed statistics for that slot will be shown.
Variants	**Example solution**
None.	

Task 2.2 - Change existing parking slot

Purpose	Let the owners modify an existing parking slot to accommodate new facility changes.	
Trigger/precondition	● The owner wants to change an existing parking slot. ● The chosen slot must not be occupied or booked.	
Frequency	5 times/year	
Critical	None.	
Sub-tasks	**Example solution**	
1 - Select the slot to change.	The system displays a map of the parking lot and lets the owner click on a slot to select it.	
2 - Update the slot's information (optional).	The owner types in the slot's new information and click submit.	
3 - Remove the parking slot (optional).	The owner clicks the "Remove" button to delete the slot from the parking lot.	
4 - Record the changes.	The system updates or deletes the slot entry in the database.	
Variants	**Example solution**	
1a - The slot to change is currently occupied or booked.	The system displays an error message notifying the owner that this slot is currently in use.	

Task 2.3 - Add a new parking slot

Purpose	Expand the parking lot by adding a new parking slot.	
Trigger	SmartParking's facilities have expanded and there is room for a new parking slot.	
Frequency	15 times/year	
Critical	None	
Sub-tasks	**Example solution**	
1 - Allocate physical space for the new slot.	The owners manually find a space for the new slot.	
2 - Specify the slot information.	The system provides an interface where the owner can specify the new slot's information such as vehicle type supported and location on the map.	
3 - Record the new slot.	The system records the new slot in the database.	
Variants	**Example solution**	
None.		

5 - Workflows

Task 1.1 - Create driver account
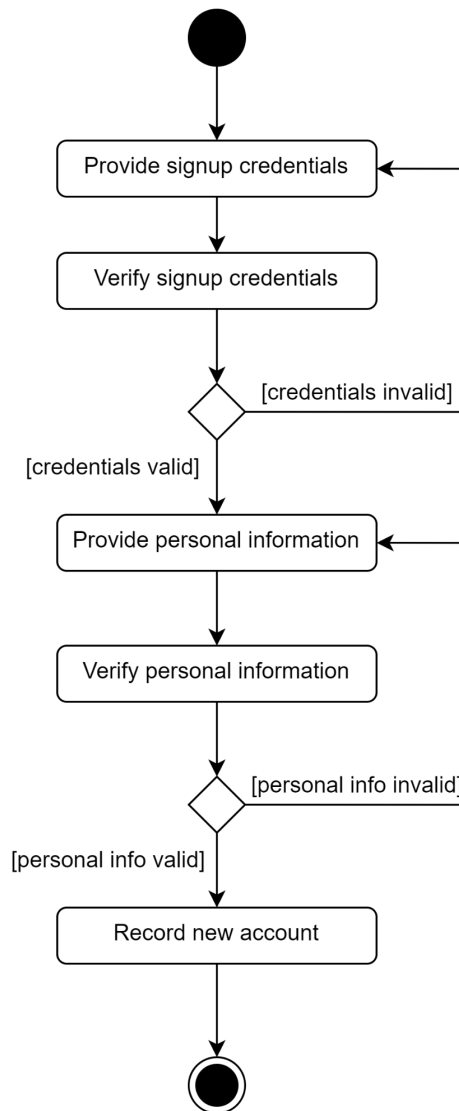
Task 1.2 - Update driver account
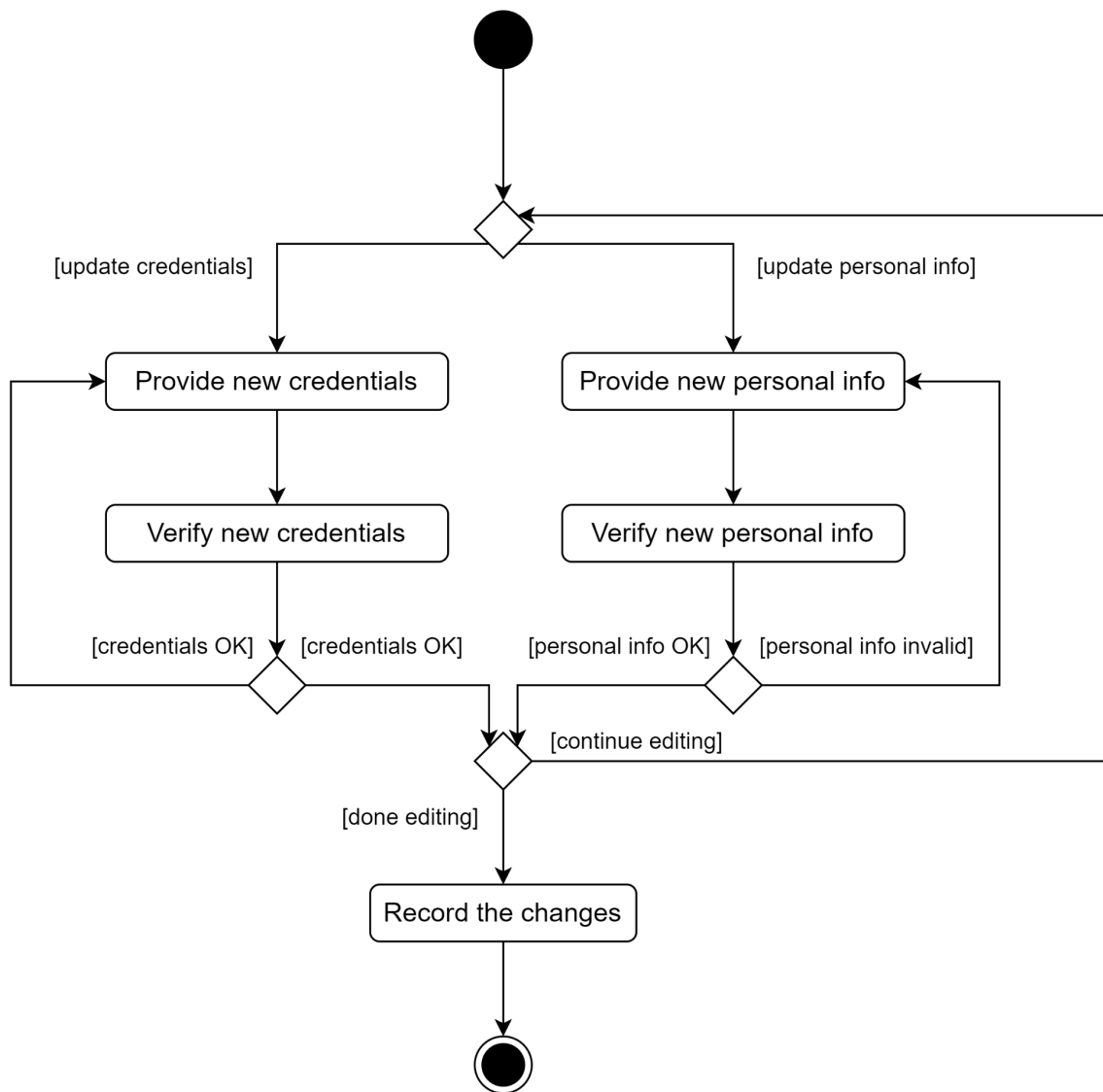
Task 1.3 - Pre-book a parking slot
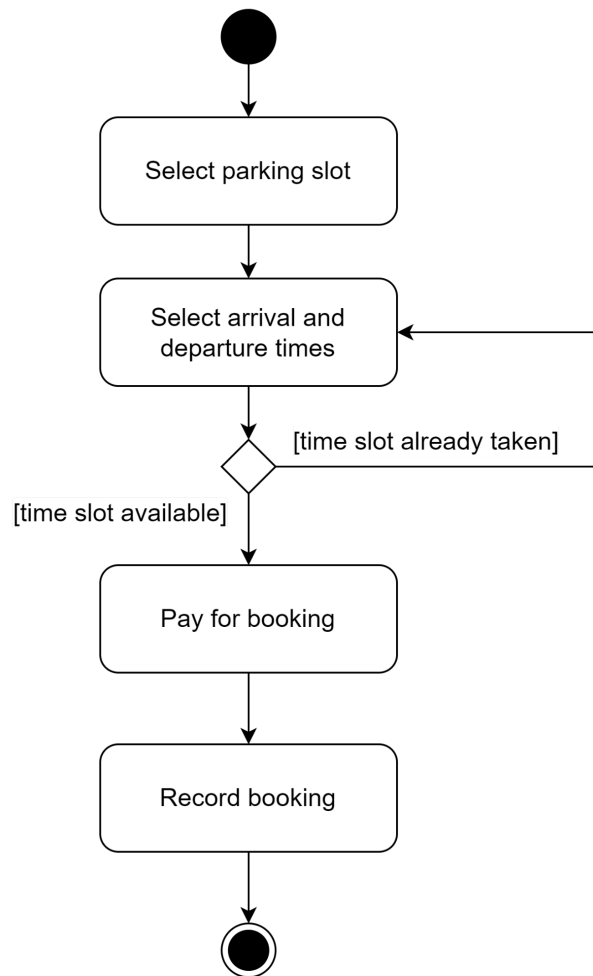
Task 1.4 - Change a booked slot
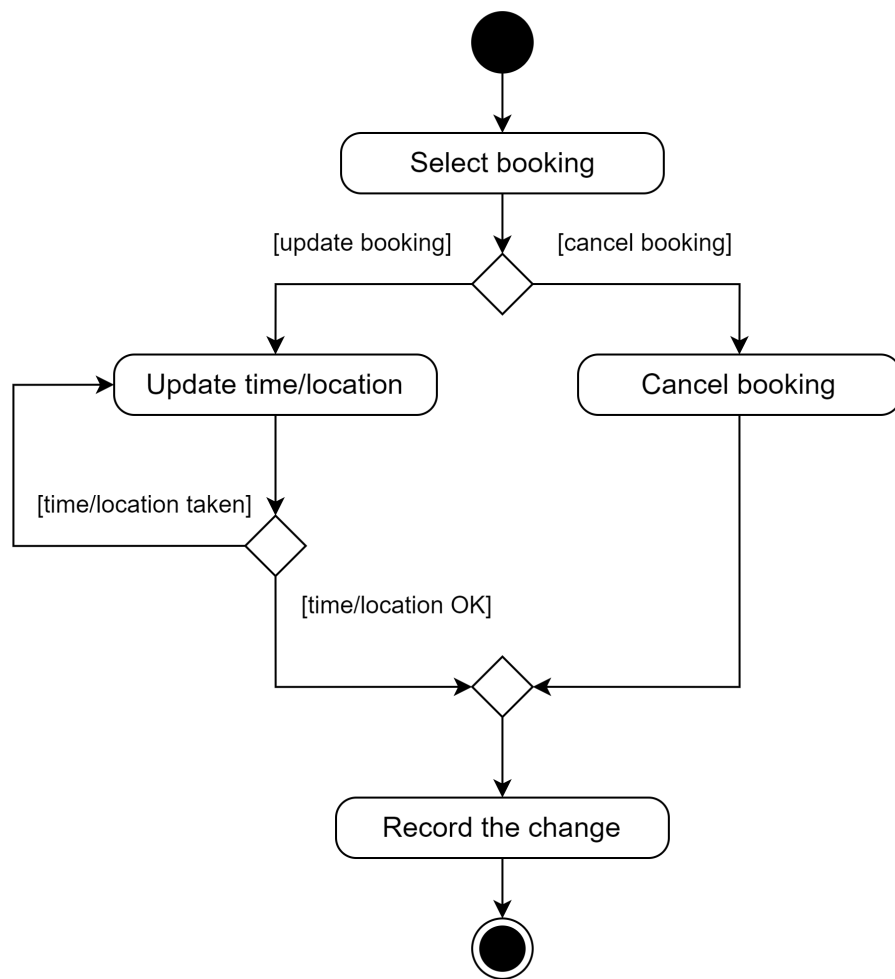
Task 1.5 - Park vehicle
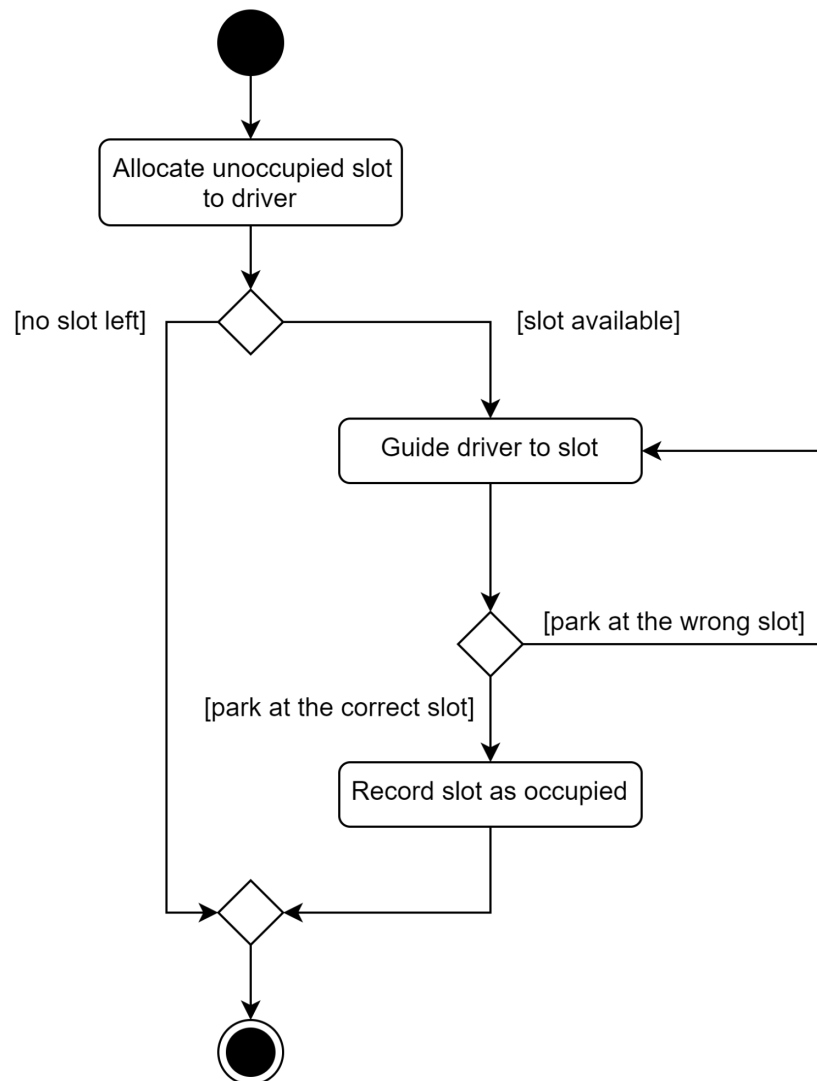
Task 1.6 - Pay for booking
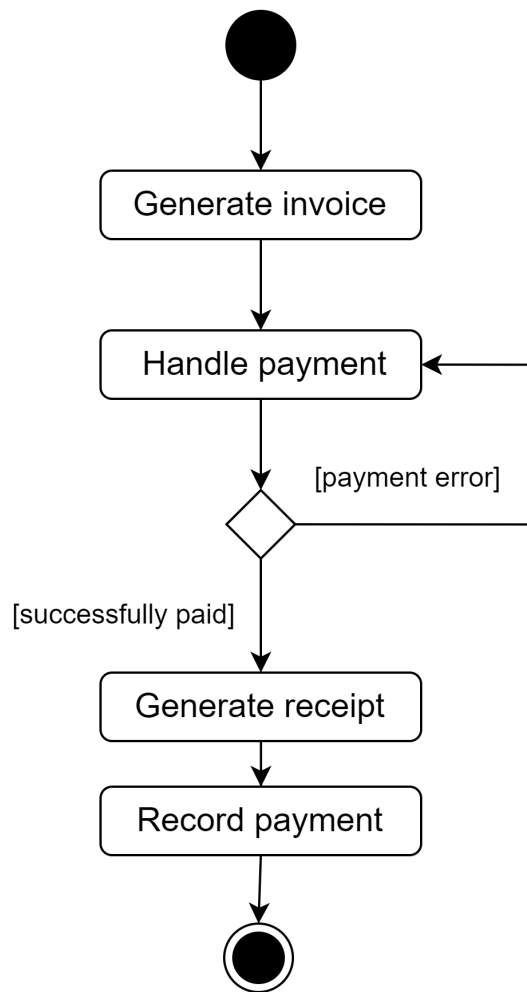
Task 1.7 - Pay for walk-in
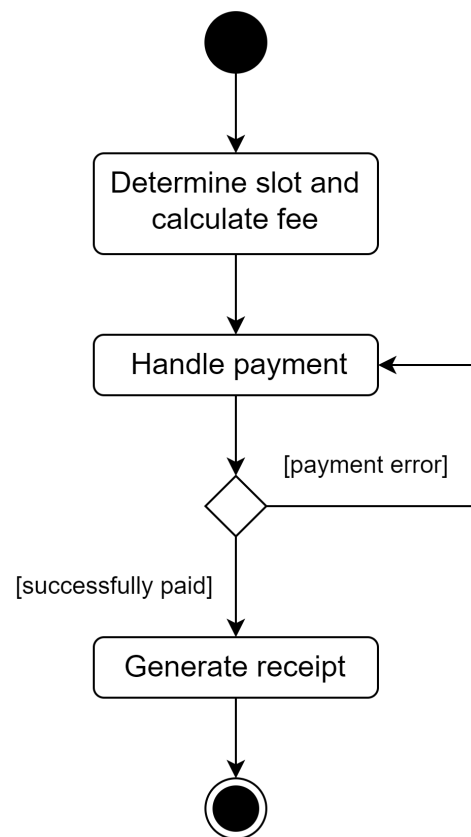
Determine slot and calculate fee

Handle payment

[payment error]

[successfully paid]

Generate receipt

Task 2.1 - View booking statistics of the parking lot

```
                    ●

                    │
                    ▼
        ┌───────────────────────┐
        │ Select desired time period │
        └───────────────────────┘
                    │
                    ▼
        ┌───────────────────────┐
        │ Retrieve data for selected period │
        └───────────────────────┘
                    │
                    ▼
        ┌───────────────────────┐
        │    View the statistics    │
        └───────────────────────┘
                    │
                    ▼
                   ◉
```

Task 2.2 - Change existing parking slot

```
                    ●
                    │
                    ▼
        ┌───────────────────────┐
        │  Select a parking slot │
        └───────────────────────┘
                    │
                    ▼
  [in use]         ◇────────── [not in use]
        │                         │
        │              [update]  ◇  [remove]
        │                 │           │
        │                 ▼           ▼
        │         ┌──────────────┐ ┌──────────────┐
        │         │ Update the slot│ │ Remove the slot│
        │         └──────────────┘ └──────────────┘
        │                 │           │
        │                  ◇
        │                  │
        │         ┌──────────────────┐
        │         │ Record the changes │
        │         └──────────────────┘
        │                  │
        └────────◇◄────────┘
                 │
                 ▼
                 ◉
```

Task 2.3 - Add a new parking slot
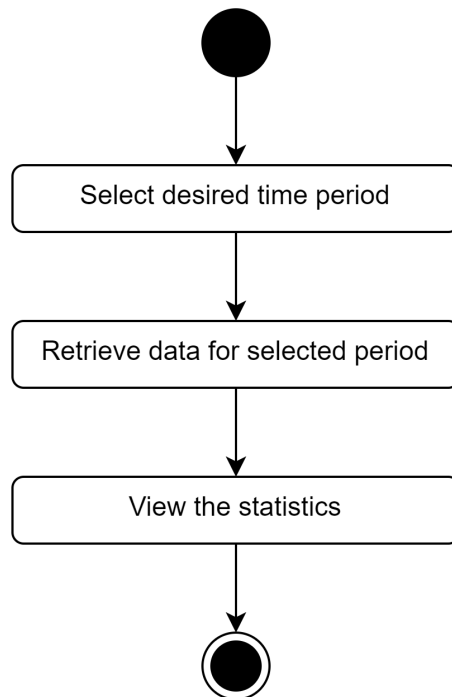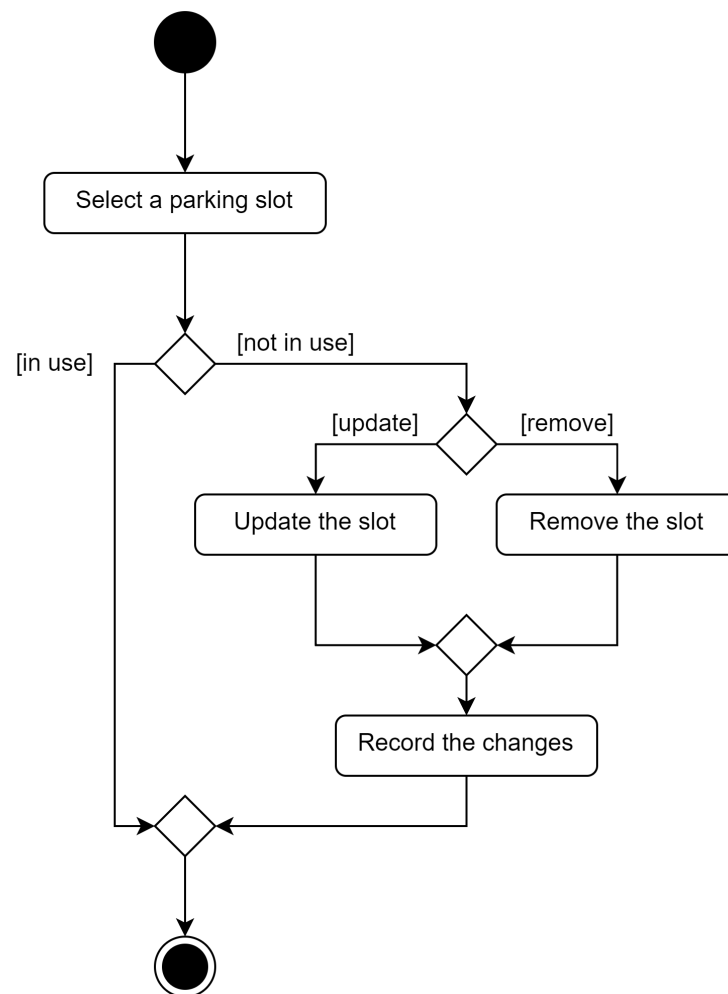
```
          ●
          │
          ▼
  ┌─────────────────────┐
  │ Allocate physical space │
  └─────────────────────┘
          │
          ▼
  ┌─────────────────────┐
  │ Specify slot information │
  └─────────────────────┘
          │
          ▼
  ┌─────────────────────┐
  │  Record the new slot  │
  └─────────────────────┘
          │
          ▼
         ◉
```
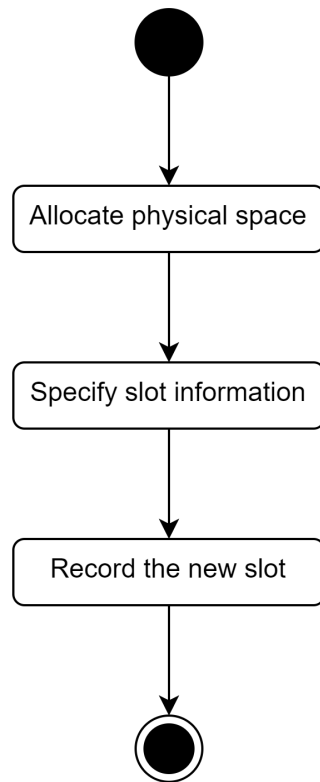
6 - Quality Requirements

Security

Security is one of the most important qualities of the system as it needs to store drivers' personal information and handle payments. Therefore, it must be secured against unauthorized access as well as prevent users' data from being exposed. To achieve these requirements, the system must:
- Be configured securely by removing any unnecessary services and setting up appropriate users' permission.
- Perform regular backups and updates to keep the system free from vulnerabilities and data loss.
- Implement a secure payment gateway to perform transactions.

Performance

Performance is an important factor in smart parking software since it directly affects user experience and operational efficiency. High-performance software detects available parking spaces quickly and accurately, minimizing the amount of time vehicles spend looking for them. This reduces congestion, lowers pollutants, and increases customer pleasure. Furthermore, dependable and quick performance enables seamless interaction with other systems, such as payment processing and navigation, resulting in a more enjoyable overall parking experience, the system must meet the following requirements:
- The app should respond to user inputs and questions in 1-2 seconds.
- The response time for loading available parking spaces should be less than three seconds.
- The parking guidance should be at least 95% accurate.
- Should be stable when receiving around 1000 queries at the same time.
- Login time should be less than 3 to 4 seconds.

Scalability

Scalability is a crucial quality attribute for the SmartParking Online Smart Parking Space (OSPS) system, ensuring that the system can handle increased load and expanded functionality over time without compromising performance or user experience. To ensure scalability, the system must meet the following requirements:

- Handle the creation of 30 new user accounts per day initially, reducing to 5 per day after the first week. Manage a large influx of new users, approximately 200, at system launch. Accommodate an annual growth rate of 15%, starting from 300 accounts in the first year, over the next 5 years.
- Support peak traffic of 300 drivers between 7:30 AM - 8:30 AM, with additional peaks of 200 requests between 5:00 PM - 6:00 PM, and 150 requests between 8:00 PM - 10:00 PM. Maintain a smooth user experience during average traffic loads throughout the day.

- Efficiently manage growing data volumes related to user accounts, bookings, parking slots, and transactions. Must support up to approximately 2GB of data each year. This amount is not expected to grow too significantly every year.
- Process 200 payment transactions per day, with scalability to accommodate increased volumes. This amount will grow by 15% each year.

Usability

Usability is an important quality of the system as it will be used by drivers, guards, and owners with varying levels of IT expertise. A system that is difficult to use will slow down operations and deter drivers from using the services, damaging SmartParking's revenue. To ensure usability, the system must meet the following requirements:

- Drivers shall successfully perform the tasks of booking a slot and updating the booking in at most 5 minutes.
- Guards shall be able to use the system after 12 hours of training.
- Owners shall be able to use the system after 12 hours of training.
- The system must show users error messages and provide clear guidance when an error happens.
- The system must follow the guidelines established in the Swinsoft Consulting UI/UX document.

Portability

Portability is critical in smart parking software because it guarantees that the program runs smoothly across several devices and operating systems. This broad compatibility improves user convenience by providing access to parking information from any device. It also facilitates maintenance and upgrades, resulting in a consistent and dependable user experience, which is critical for wider adoption. the system must meet the following requirements:
- Available on both Android and IOS platforms.
- Available as a web application for computer users.

7 - Other Requirements

Product-level requirements

- Create Driver Account:
  - The system should provide an interface for drivers to create a new account. This should include fields for the driver's name, contact information, and vehicle details.
- Update Driver Account
  - The system should allow drivers to update their account information. This includes changing their contact information, password, and vehicle details.
- Pre-book a Parking Slot
  - The system should allow drivers to pre-book a parking slot. The system should display available slots for the driver to choose from.
- Change a Booked Slot
  - The system should allow drivers to change their booked slot. This includes canceling a booking and making a new booking.
- Park Vehicle
  - The system should allow drivers to check in their vehicles once they arrive at the parking lot. The system should confirm the booking and guide the driver to the booked slot.
- Pay for Walk-in
  - The system should allow walk-in drivers to pay for their parking. The system should calculate the parking fee based on the duration of parking.
- View Booking Statistics of the Parking Lot
  - The system should provide an interface for administrators to view booking statistics. This includes the number of bookings, revenue generated, and occupancy rate.
- Change Existing Parking Slot
  - The system should allow administrators to change the details of an existing parking slot. This includes changing the slot size, location, and availability.
- Add a New Parking Slot
  - The system should allow administrators to add a new parking slot. The system should prompt the administrator to enter the slot details such as size, location, and availability.

Design-level requirements

Besides the product-level requirements that have been mentioned in the previous sections, the system must also follow these design-level requirements:
- The UI/UX of the system must be accessible to users by using appropriate font style and color contrast.
- The system should be responsive to ensure the user interface works with a wide range of devices and screen sizes.
- Visualize the statistics of the booked slots nicely to the owners of the system for the purpose of monitoring the parking lot.
- The system may need a notification system to remind the users about their services.

8 - Validation

CRUD Check

Entities / Tasks	Driver	Slot	SlotType	ParkingSession	Invoice	Receipt
1.1 - Create driver account	C					
1.2 - Update driver account	R, U					
1.3 - Pre-book a parking slot		R	R	C		
1.4 - Change a booked slot		R	R	R, U, D		
1.5 - Park vehicle		R	R	C, U		
1.6 - Pay for booking		R	R	R	C, R, D	C, R
1.7 - Pay for walk-in		R	R	R		C, R
2.1 - View booking statistics of the parking lot		R	R	R		
2.2 - Change existing parking slot		R, U, D	C, U, D			
2.3 - Add a new parking slot		C	C, U, D			
Missing	*D*				*U*	*U, D*

Explanation of missing operations:
- ***Delete*** for ***Driver***: We assume that SmartParking does not allow the deletion of Driver accounts.
- ***Update*** for ***Invoice*** and ***Receipt***: It does not make sense to update an invoice or receipt.
- ***Delete*** for ***Receipt***: We assume that SmartParking may want to calculate the historical profits generated from their parking lot, requiring the retention of receipts for long periods. Also, given their small size, it is acceptable to store all old receipts.

9 - Possible solutions

Solution 1 - Mobile app for OSPS

A Smart Parking mobile app is intended to streamline the parking experience by providing extensive and user-friendly features. It has strong user account management, allowing users to register with their email, password, and personal information and manage their profiles securely.

The software displays real-time parking slot availability, allowing users to choose slots, specify arrival and departure times, and receive booking confirmations and reminders. Payment processing is safe, and it accepts a variety of payment methods such as credit cards and PayPal, with automatic invoice generation. Administrators benefit from a comprehensive administration interface that allows them to control slots and monitor real-time activity. Reporting tools offer insights into usage trends, financial activities, and system performance data.

The software is designed for scalability, processing large numbers of reservations effectively and providing quick response times. Security is the top priority, with strong data protection and access control procedures in place. The software is mobile-compatible, with an easy layout for a seamless user experience. interaction with third-party services such as Google Maps and payment gateways improves functionality, whilst API access enables more extensive system interaction. Comprehensive support and maintenance services ensure that the app stays functional and up to date. This mobile app should be available on both Android and IOS platforms for versatility.

Solution 2 - IoT-Enabled OSPS

An IoT-enabled smart parking system leverages the power of the Internet of Things (IoT) to provide a highly automated and efficient parking management solution. This system integrates IoT sensors, mobile apps, and web interfaces to deliver real-time data and automated control over parking operations. Users can register with their email, password, and personal information, managing their profiles securely through mobile and web interfaces, with strong authentication and role-based access controls ensuring data security and user privacy.

IoT sensors installed in parking lots provide real-time data on slot availability, allowing users to book slots, specify arrival and departure times, and receive booking confirmations and reminders. The system supports both advance and immediate reservations, with real-time updates on slot status. Secure payment processing is available with various payment options like credit cards and PayPal, integrated into both the mobile app and web interface, with automatic generation of invoices and receipts for completed transactions.

Administrators have access to a comprehensive dashboard that integrates data from IoT sensors. Reporting tools provide insights into usage trends, financial activities, and system performance data, enhanced by real-time IoT data. Designed to handle large volumes of data from IoT sensors, the system ensures scalability and reliability through cloud hosting (e.g., AWS, Azure), with load balancing and auto-scaling features managing varying traffic loads effectively, maintaining quick response times and high availability.

Security is a top priority, with robust data protection measures and access controls in place, complemented by regular security audits and compliance with industry standards. Enhanced security protocols for IoT devices and data transmission protect user data and system integrity. Integration with third-party services such as Google Maps for navigation and payment gateways for secure transactions enhances functionality, while API access allows for further system interaction and integration capabilities. Comprehensive support and maintenance services ensure the system remains functional and updated, providing a seamless user experience across all devices and allowing users to access the system from smartphones, tablets, desktops, laptops, and through automated IoT interfaces.