PROJECT PHASE 3

SECD2613 - SYSTEM ANALYSIS AND DESIGN

SEMESTER II 2024/2025

LECTURER: DR. ARYATI BINTI BAKRI

SECTION: 04

NAME	NO. MATRIX
GOA WEI SIANG	A24CS0077
ABDULRAHMAN KHALED ABDULLAH SHIEKH	A24CS4091
ZAKY ZULHADI	A24CS4097
MUHAMMAD AIMAN FIKRI BIN ZULKARNAIN	A24CS8008
NUR AFIQA AFIFIE BINTI AZMAN	A24CS0151

Content

1.0 System Analysis and System Specification

Before a system is designed, we have analysed the current system, where the functional requirement and non-functional requirements of the system are identified to ensure the newly designed system fulfils the requirements of the client. Other than that, we have specified what the new system should do by generating a logical data flow diagram to visualise the flow of data and also its function.

1.1 Functional Requirement

This section outlines the key features the system must include to meet user needs. These features include user registration, car browsing with filters, real-time booking and payment, invoice generation, and rental management. It also supports feedback collection, notifications, and loyalty rewards to improve user experience and business efficiency:

- ➔ User Registration & Authentication:
  Customers and staff can register and log in securely. Role-based access: admin, customer, operations staff.

- ➔ Vehicle Browsing & Search:
  Customers may search vehicles by model, price, availability, or destination. Allow option to filter results (e.g., SUV, manual/automatic).

- ➔ Online Booking & Payment
  Customers are allowed real-time booking for car rental with date/time selection.
  Centralised database that may store payments made by customers and automatically produce invoice.

- ➔ Rental Management
  Managers may track active, upcoming, and completed rentals. Managers may also track late return, cancellations, and refunds.

- ➔ Fleet Management

Managers can add, update, or remove vehicle listings. Manager able to track vehicle condition, maintenance, availability through digitalised database.

➔ Customer Feedback & Support

Managers may collect ratings and reviews from customers after rentals for feedback. May provide a chat or ticketing system for support.

➔ Report Generation

System may generate usage statistics, financial summaries (invoice report), and customer reports and reviews.

➔ Discounts & Loyalty System

Provide promo codes, discounts and dynamic pricing for car rental. Track customer history and reward repeat customers.

➔ Notification System

Send emails/SMS for booking confirmations, reminders, or alerts.

➔ Location Services

Integrate Google Maps API to show car pickup/drop-off points.

1.2 Non-functional Requirement

This section deals with the quality aspects of the system like performance, security, and usability. The system must be timely, secure, user-friendly, and web and mobile compliant. It must be reliable, scalable, and industry compliant for long-term use and future growth.

★ Performance

Support at least 500 concurrent users with <3s page load time.

★ Scalability

Must accommodate new features (e.g., car subscription plans) and more users.

★ Security

Use HTTPS, input validation, and role-based access. Encrypt sensitive user and payment data (PCI DSS compliance).

★ Usability

Web-based responsive UI. Simple, intuitive navigation for all age groups.

★ Availability

Ensure 24/7 access with minimal downtime (<0.5% monthly).

★ Maintainability

Use a modular codebase (MVC architecture) for easy updates.

★ Portability

Accessible via web browsers and mobile devices (Android/iOS).

★ Interoperability

Integrate with external APIs (map, vehicle tracking, CRM).

★ Compliance

Adhere to local laws (e.g., driver's license validation, rental terms). GDPR compliance for user data privacy in applicable regions.

★ Agility

Development should follow Agile methods (Scrum sprints, CI/CD pipelines).

1.3 Logical DFD

To visualise the proposed system, we had created a logical data flow diagram that showed the proposed process of the system. In this logical DFD, we had modified the AS-IS DFD to automate some of the processes that can be automated, while ensuring the business processes are still maintained.

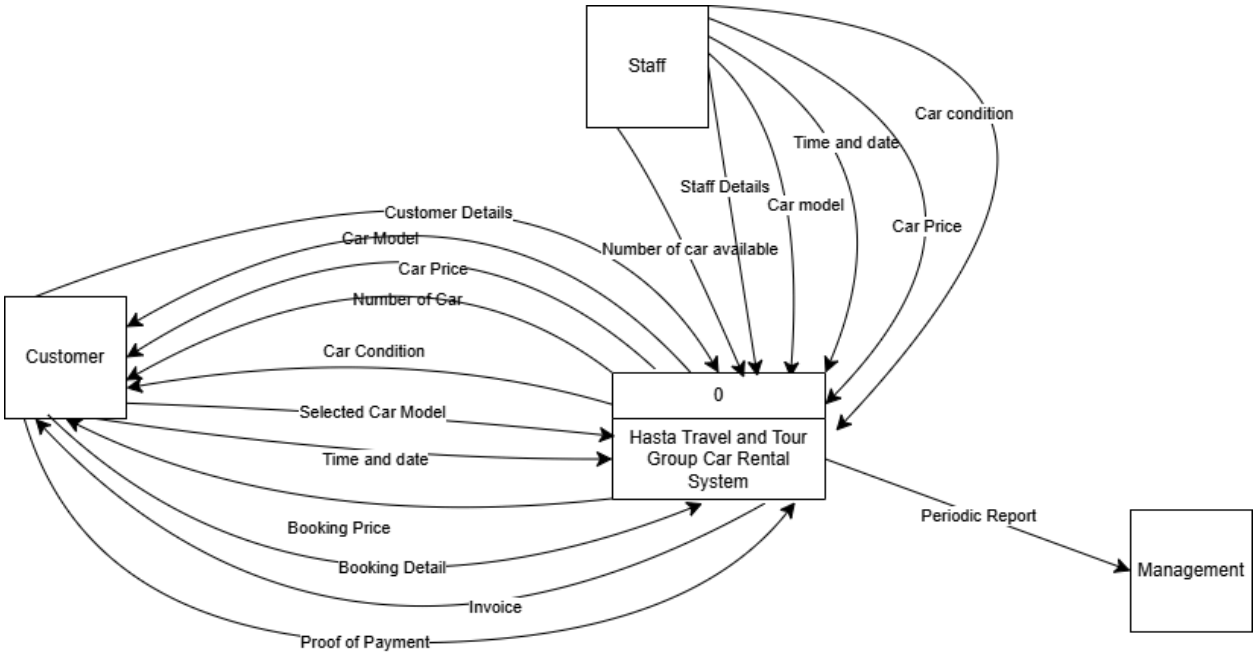


Figure 1 : Logical DFD (Context Diagram)

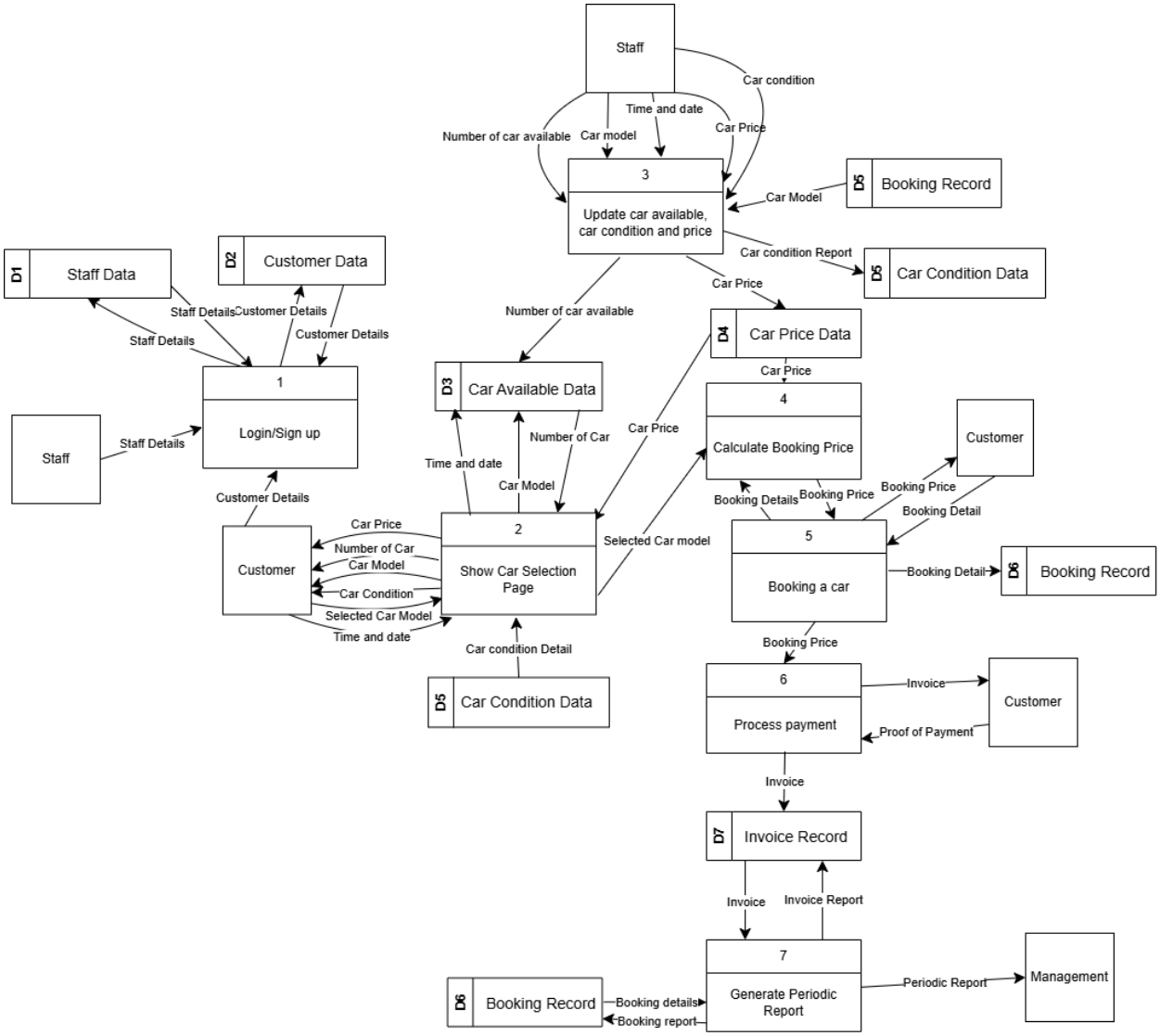


Figure 2: Logical DFD (Diagram 0)

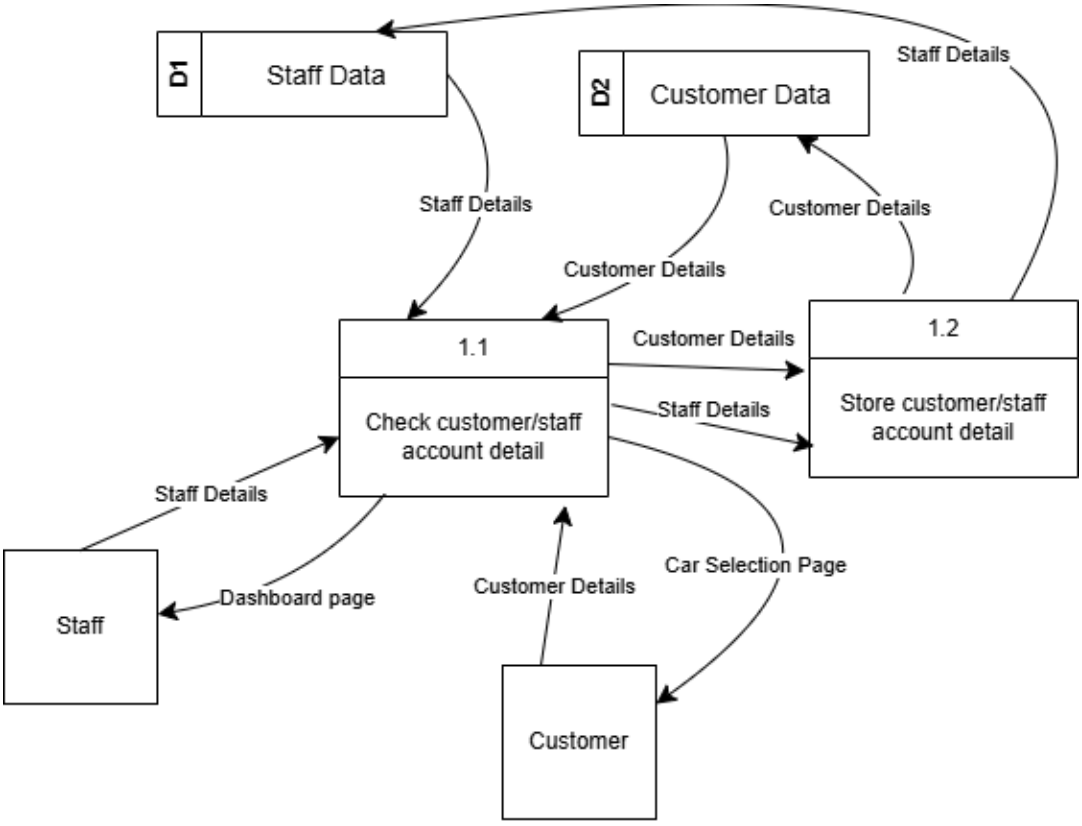


Figure 3: Logical Diagram (Child Diagram-Process 1)

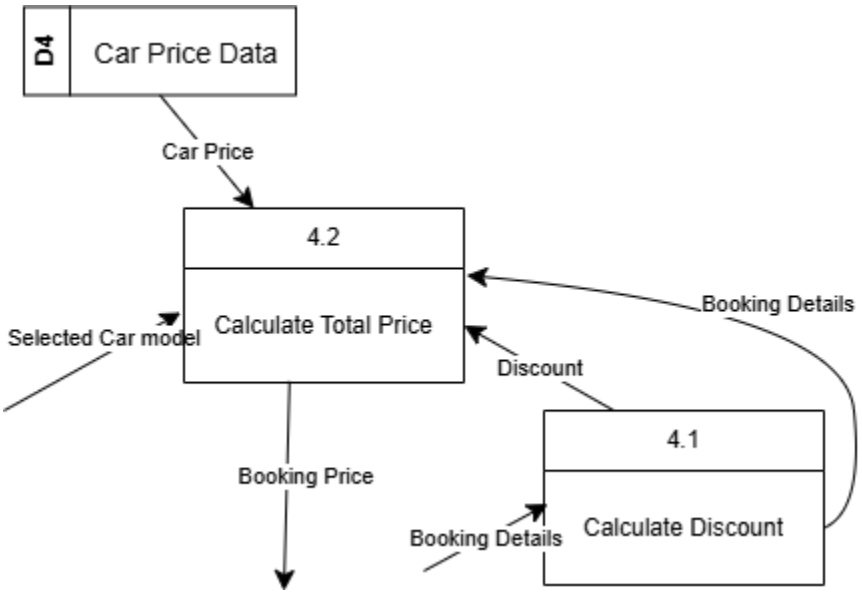


Figure 4: Logical Diagram (Child Diagram-Process 4)

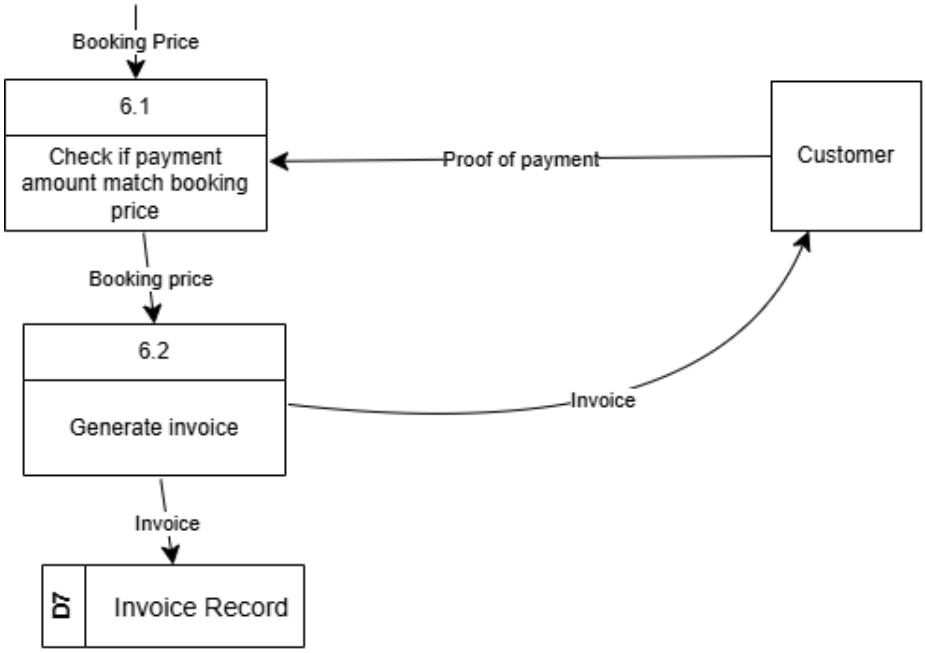


Figure 5: Logical DFD (Child Diagram-Process 6)

1.4 Process Specification

For the process specification of the proposed system, we had decided to use structured English to specify the processes in the proposed system. This is due to our system had more sequential process which is better to be present in Structured English. The process specification for each of the process of the proposed system are as followed:

PROCESS 1: Login/Sign up

```
DO
IF sign up
        IF staff account
                THEN enter staff details
                STORE staff account details
        ELSE IF customer account
                THEN enter customer details
                STORE customer account details
        ENDIF
IF login
        DO WHILE account not match
        ENTER Account details
        CHECK Account details
        IF match username and password
                THEN IF Staff Account
                                THEN show dashboard page
                        ELSE IF Customer Account
                                THEN show car selection page
                        ENDIF
        ELSE show account not found
        ENDIF
        ENDDO
ENDIF
ENDDO
```

PROCESS 2: Show Car Selection Page

```
DO
SHOW car selection page
GET Time and Date
CHECK car availability
Show available car model
IF select car model
        THEN show selected car page
                show car price
                show car condition
        IF booking car
                THEN show booking page
        ENDIF
ENDIF
ENDDO
```

PROCESS 3: Update car available, car condition and price

```
DO
SELECT car model to update
UPDATE car price
UPDATE car available count
DO WHILE car available count != 0
        IF car of the car model is book
                THEN decrement car available count
        ENDIF
ENDDO
IF change car condition
        THEN select car
                UPDATE car condition
ENDIF
ENDDO
```

PROCESS 4: Calculate Booking Price

```
DO
GET date of booking
CHECK day of date of booking
IF weekend
        THEN discount = 5
ELSE discount = 0
ENDIF
GET duration of booking
IF duration > 3 days
        THEN booking discount rate = 0.85
ELSE booking discount rate =0
ENDIF
GET location
IF location outside UTM
        THEN delivery fee = 10
ELSE delivery fee = 0
ENDIF
CALCULATE booking fee = (car price - discount)*booking discount rate + delivery fee
SHOW booking fee
ENDDO
```

PROCESS 5: Booking a car

```
DO
GET car model
GET car price
ENTER date of booking
ENTER duration of booking
ENTER location
CALCULATE booking fee
SHOW booking fee
IF confirm booking
        THEN store booking detail
                show payment page
ENDIF
ENDDO
```

PROCESS 6: Process Payment
DO
SHOW QR for payment
GET proof of payment(Receipt)
CHECK payment amount
IF total payment amount match booking fee
    THEN generate e-invoice
ENDIF
ENDDO

PROCESS 7: Generate Periodic Payment
DO
GET invoice
GET booking details
IF end of month
GENERATE monthly invoice report
GENERATE monthly booking report
ENDIF
IF end of year
GENERATE yearly invoice report
GENERATE yearly booking report
ENDIF
ENDDO

2.0 System Design

After analysing and specifying the requirement and function of the new system, we have designed the system by identifying how the system may carry those requirements and functions, whether by computerised method or still through manual method. Other than that, we also identified the type of computer or device that will be used for the system and also the system architecture to ensure the workings of the system when it is implemented.

2.1 Physical DFD

After creating the logical DFD for the proposed system, the logical DFD has to be converted into physical DFD to aid in the implementation of the system. In the physical DFD, the processes are more detailed and validation of data will also be carried out to ensure data integrity in the system.

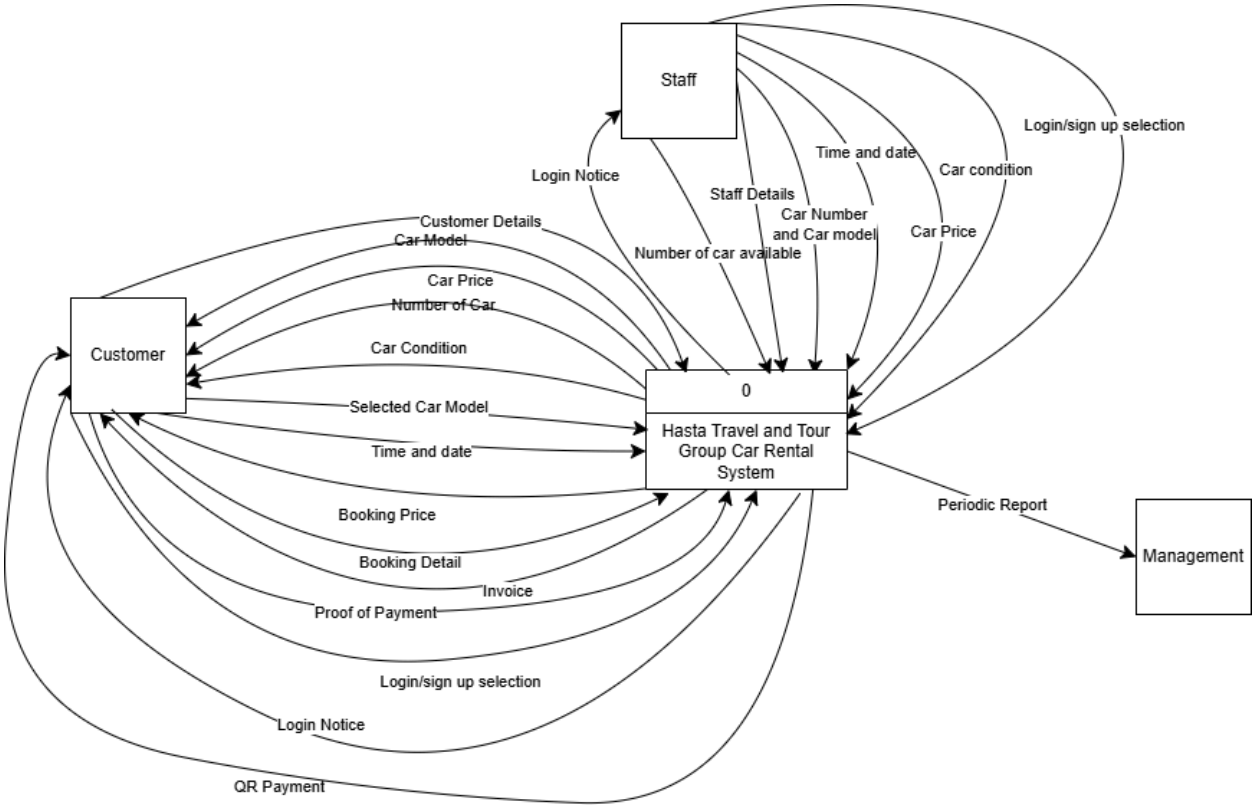


Figure 6: Physical DFD (Context Diagram)

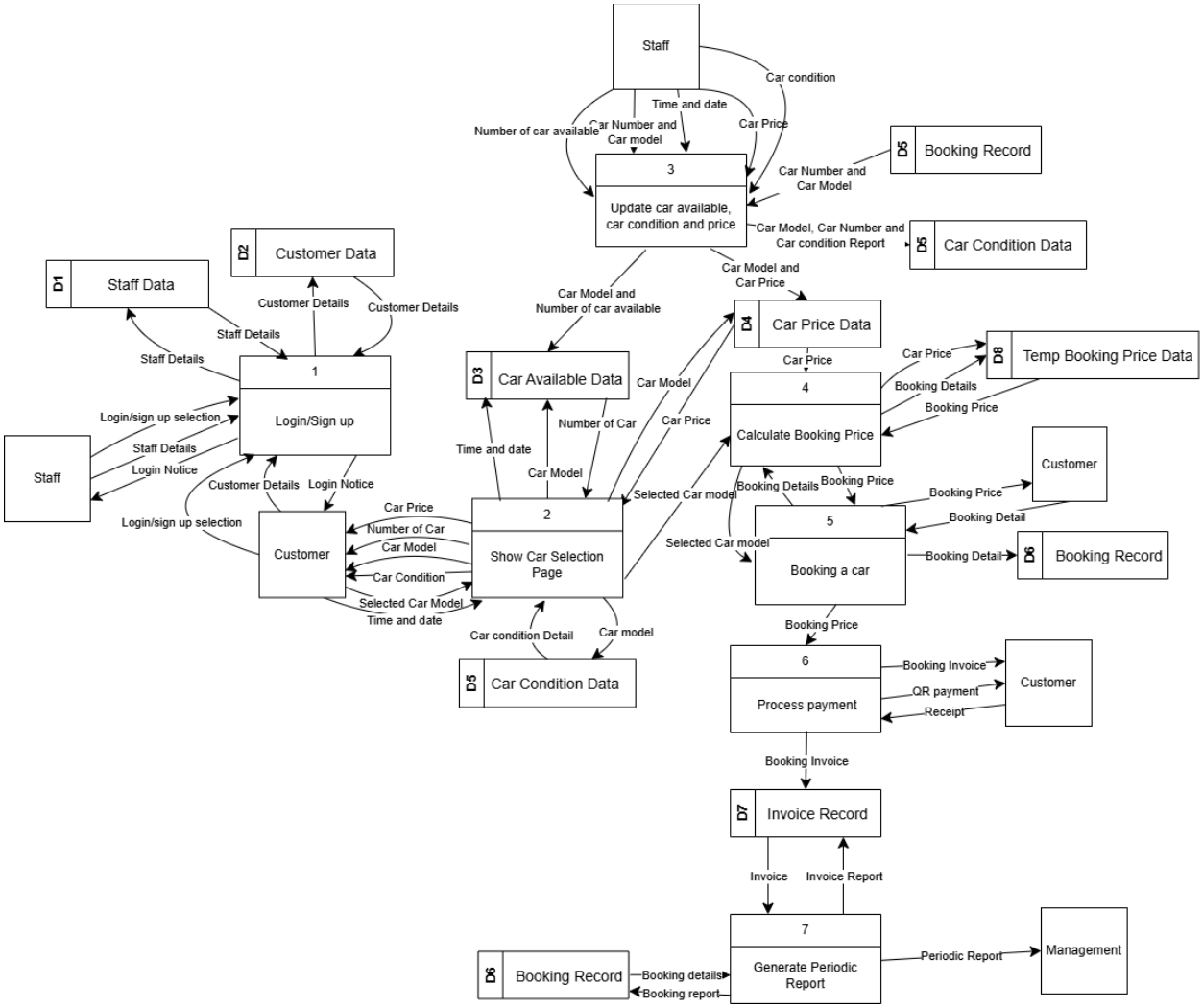


Figure 7: Physical DFD (Diagram 0)

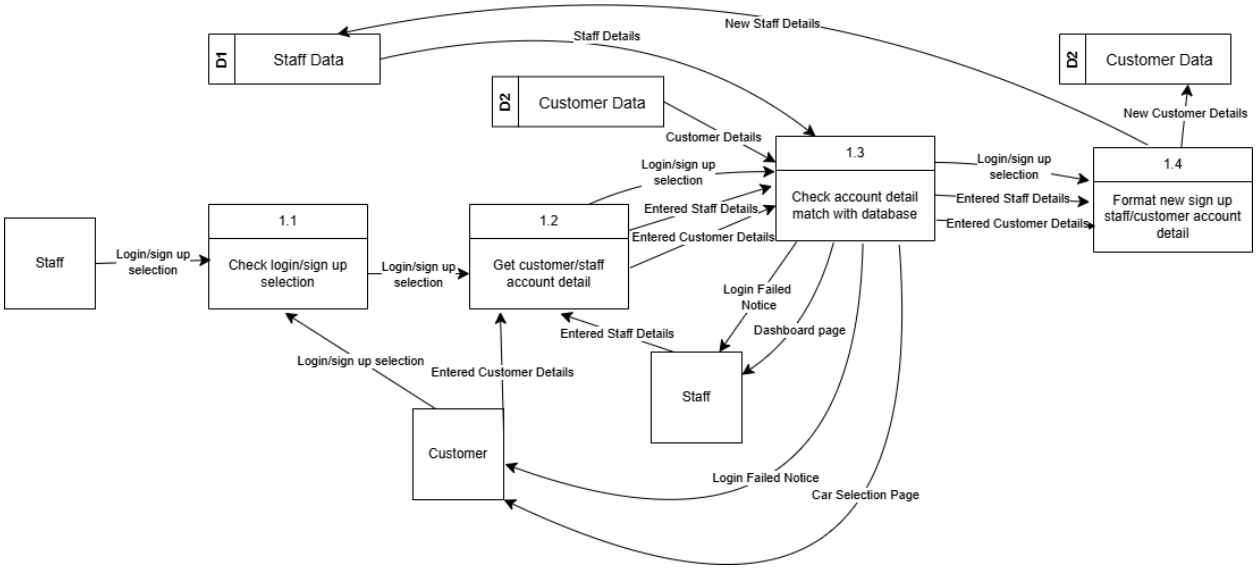


Figure 8: Physical DFD (Child Diagram-Process 1)

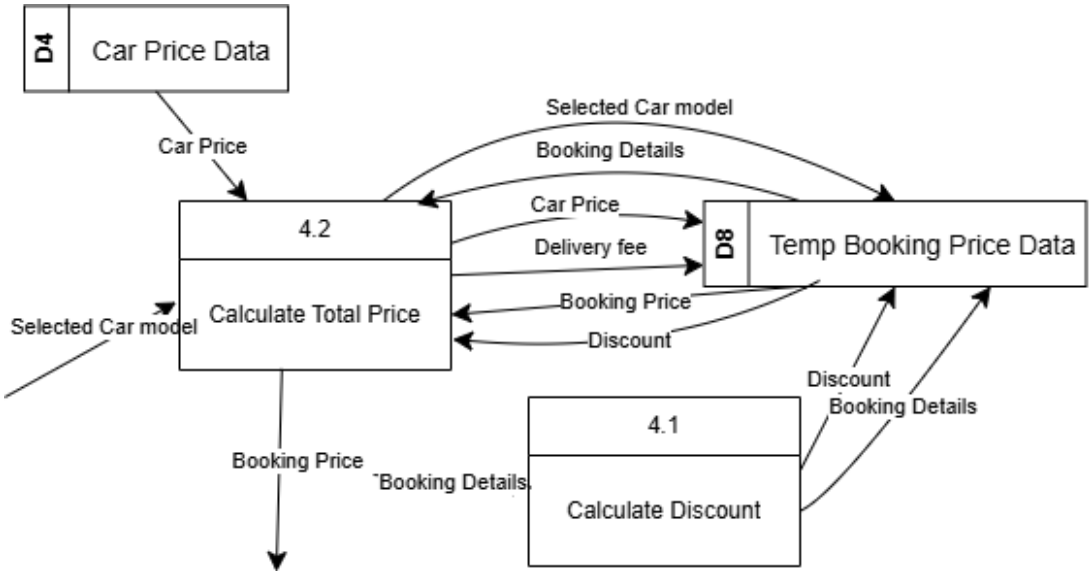


Figure 9: Physical DFD (Child Diagram Process 4)

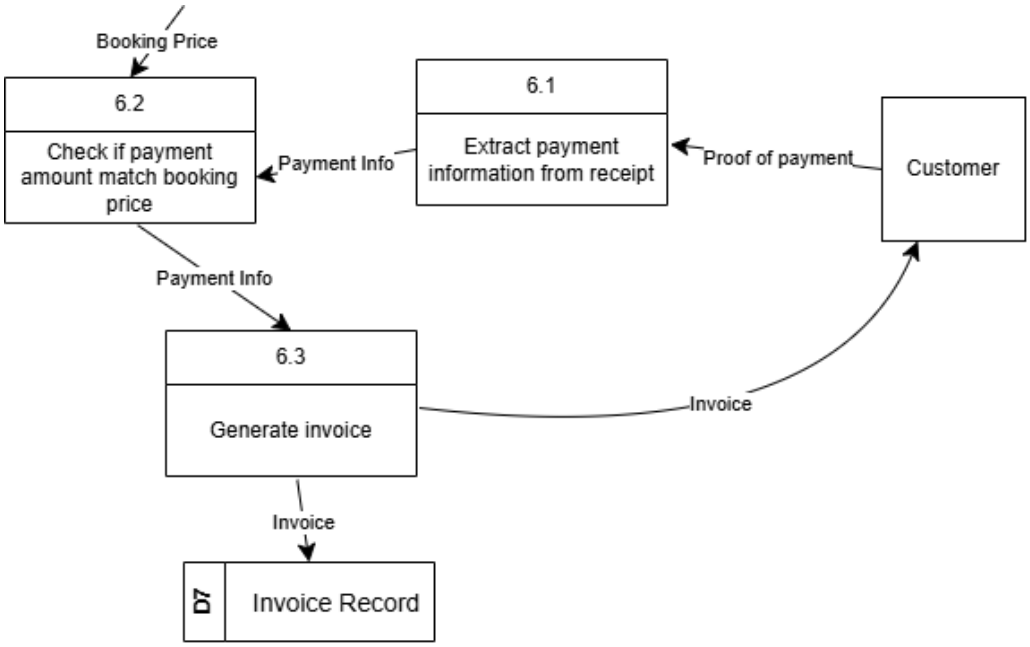


Figure 10: Physical DFD (Child Diagram-Process 6)

2.2 CRUD Matrix

To identify each activity functional process in a system, we had created a CRUD matrix table. The CRUD table may ease us in identifying the process and also how the data is processed in each activity.

Activity	Customer Details	Staff Details	Car Price	Car Count	Car Condition	Booking Detail	Invoice Data
Login/Sign Up	CR	CR					
Update car price,car count, car condition		R	CRUD	CRUD	CRUD		
Show Car Selection			R	R	R		
Booking car	R		R	R	R	CU	
Calculate Booking Price	R		R			RU	
Process Payment							CU
Generate Report						RU	RU

Table 1: CRUD Matrix table

2.3 Event Response Table

To understand the flow of the activity of the system, an Event Response Table has been made to show the source, trigger, activity and response for each event that is included in the system. With the event response table, we may visualise the flow of process more clearly as the input and output of the process are identified.

Event	Source	Trigger	Activity	Response	Destination
Customer/staff log in/sign up	Customer, Staff	Login/sign up selection, Customer account detail, Staff account detail	Check login/sign up selection, Check customer/staff account detail match customer/staff record, Format customer/staff account detail	Car Selection Page, New Customer Details, Dashboard page, New Staff details	Customer, Staff
Customer browse car	Customer	Selected car model	Show car selection page Show selected car page	Car selection page Selected car page	Customer
Customer book car	Customer	Booking detail	Store data in booking record Calculate booking price	Booking price Payment page	Customer
Obtain customer payment	Customer	Proof of payment(receipt)	Check payment amount with booking price Generate invoice	Invoice	Customer

Update car price, car count and car condition	Staff	Car price Car count Car condition	Set car price Update car count Update car condition	Updated car price Updated car count Updates car condition	
Generate periodic report		Monthly,Yearly	Generate monthly report Generate yearly report	Periodic report	Management

Table 2: Event Response Table

2.4 Partitioning

To divide the processes into collections of computer programs or manual procedures, we had partitioned the processes of the physical DFD. The processes inside the dashed line means that the processes should be created in a single program.

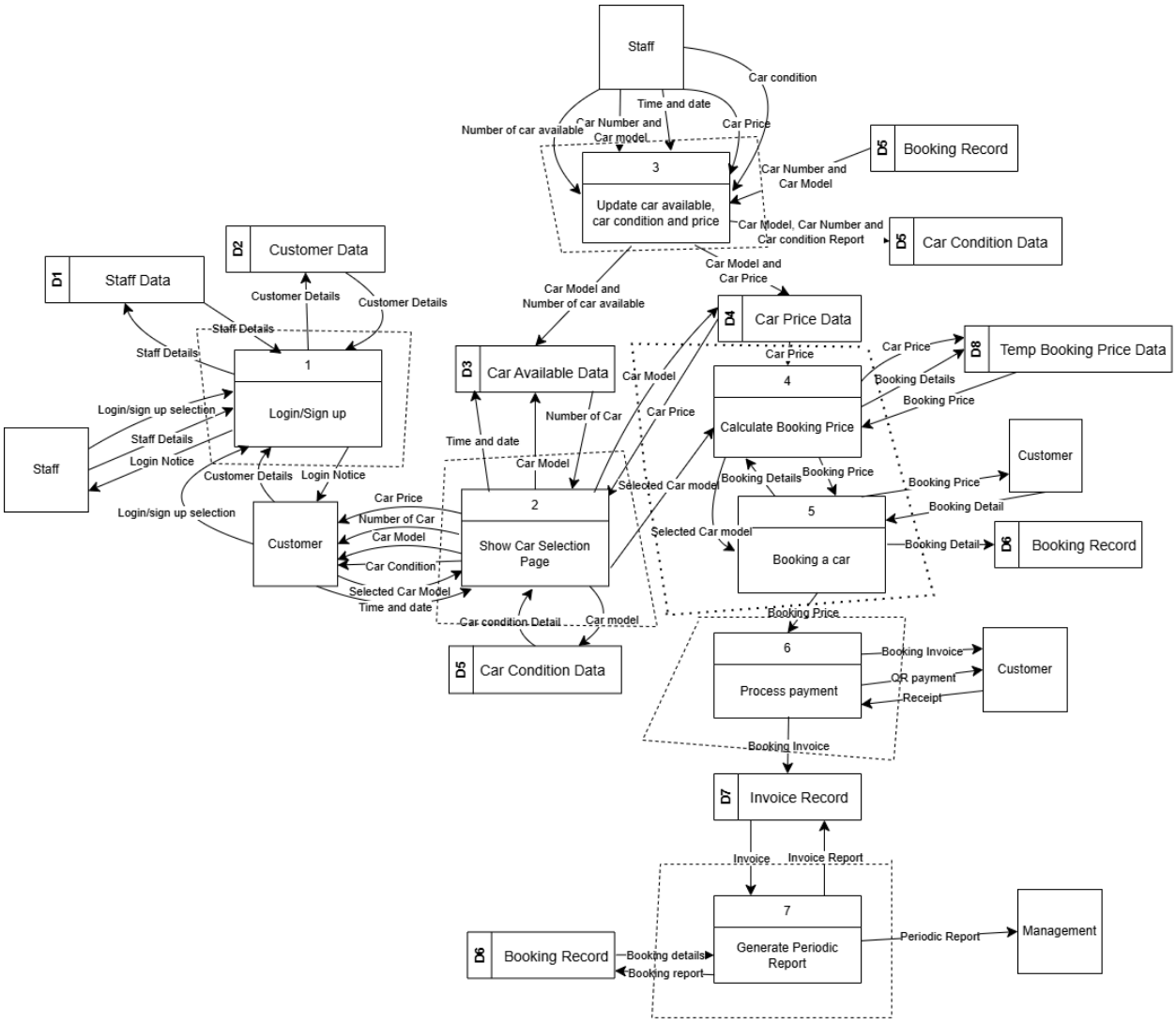


Figure 11: Partitioned Physical DFD

2.5 Structure Chart

To show the functional decomposition of the proposed system, we had created a structure chart to show the relationship between each process module in the proposed system.

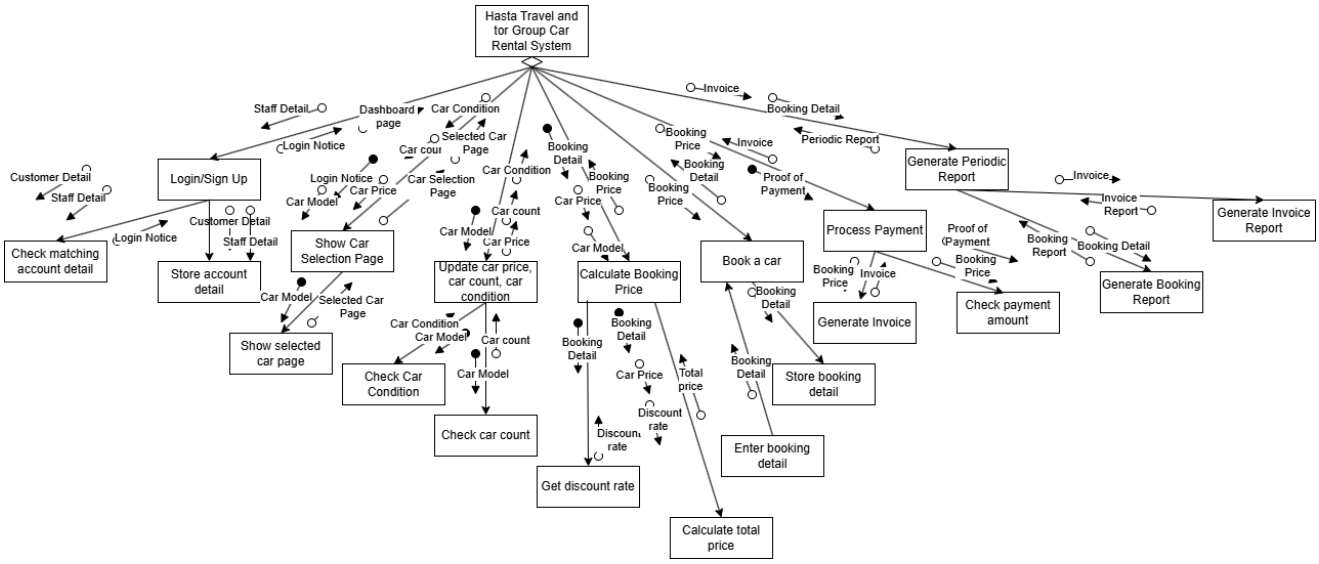


Figure 12: Structure Chart

2.6 System Architecture

For the system architecture of the system, we will be using 2-tier architecture, that is architecture consisting of 2 tier, namely client and web server.

Since this system architecture requires a web server, we will require hardware that is able to manage at least 100 page requests per second so that the performance of the system is not affected.

Clients, which may include staff or customers will access the web server through the webpage link to access the booking system.

The booking system will also be connected to a database to store data required for the booking system processes such as customer detail, staff detail, booking detail and more.

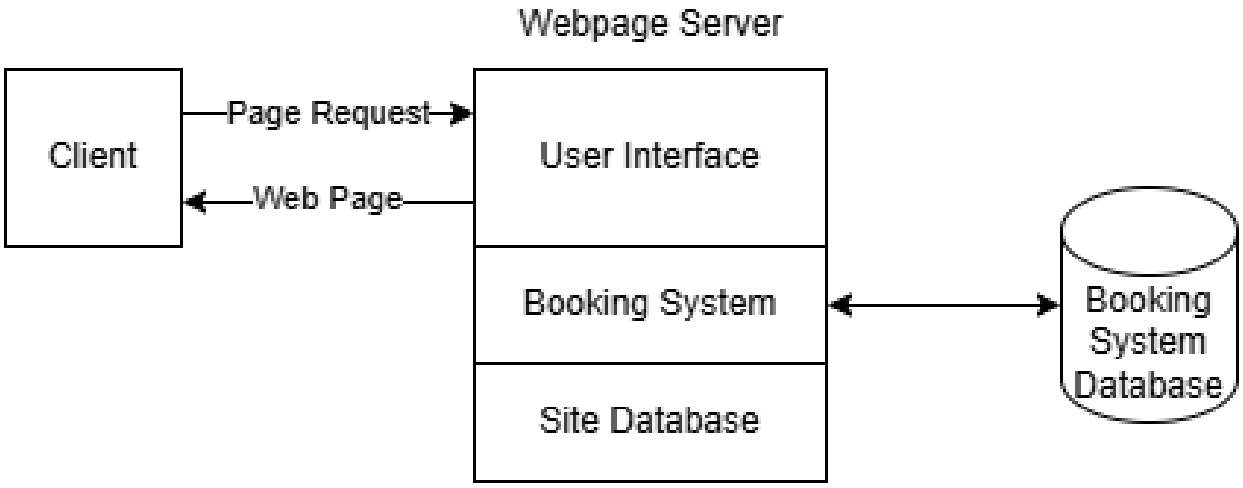
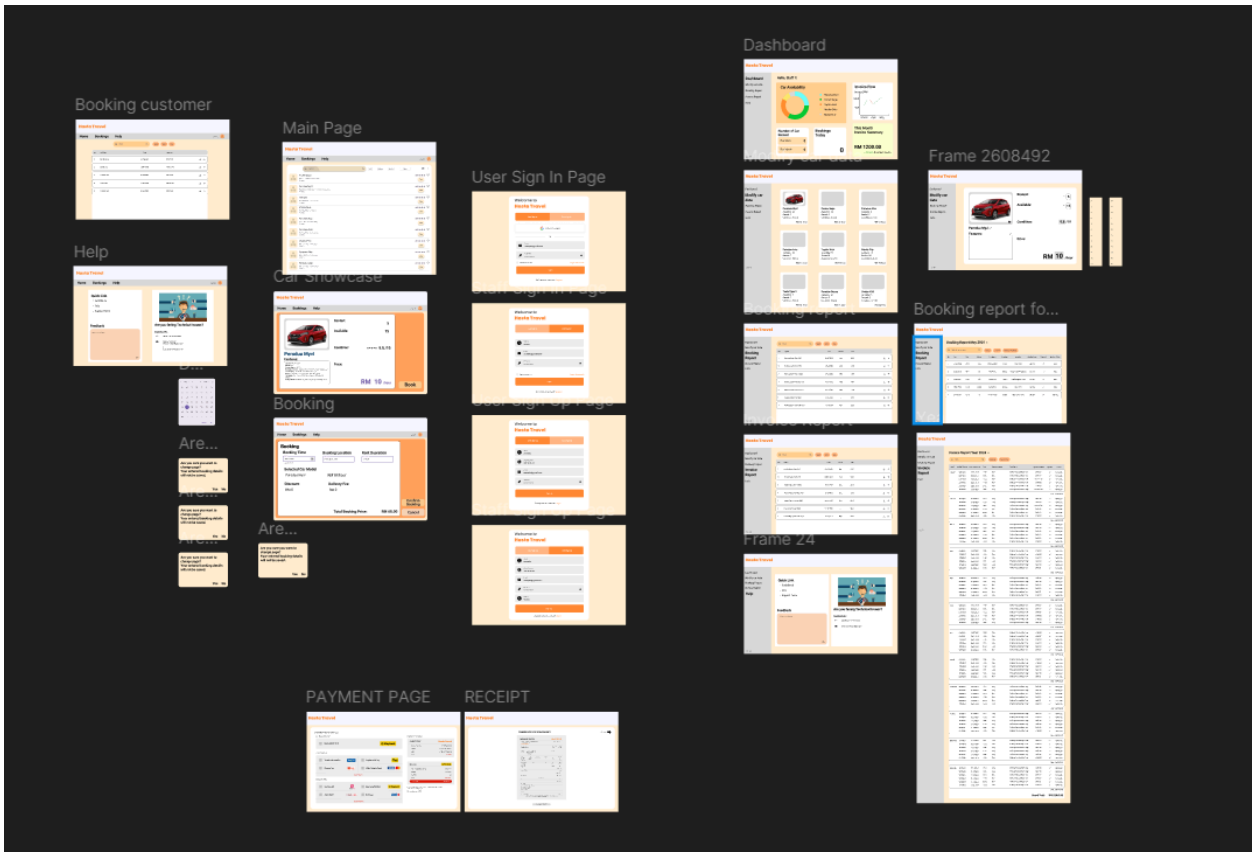


Figure 13: 2-tier system architecture

3.0 Wireframe

Before implementing the coding for the system, we had to design the system interface first. Hence, we had designed the prototype design for the car rental system by using Figma. The design for the interface for input design and output design are carefully considered to ensure a comfortable and intuitive interface for the user.

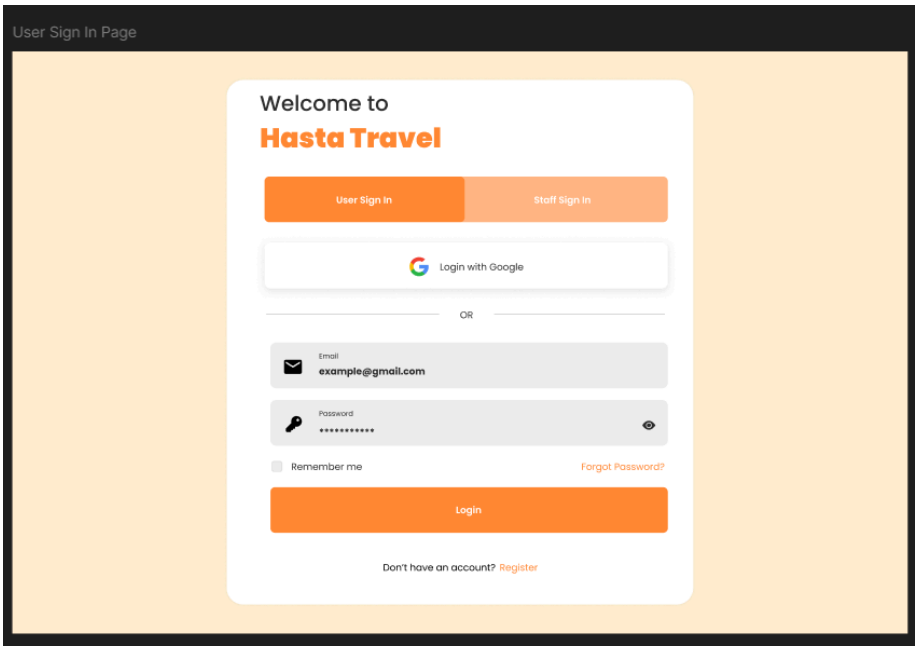


Picture 1: Wireframe(Figma)

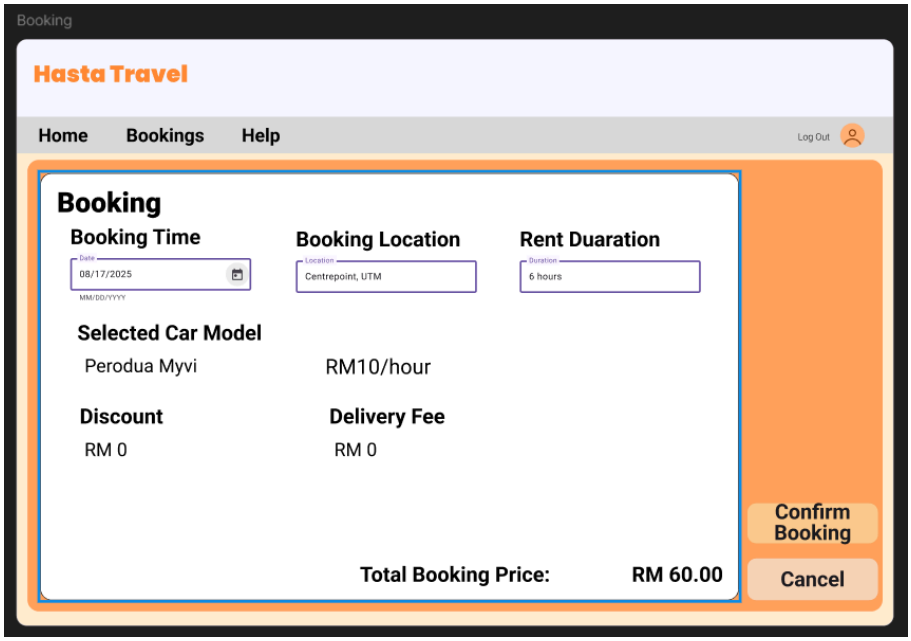
Figma link:
https://www.figma.com/design/l9hfGZSk0qiZGivDo8KbyJ/Project-SAD?node-id=0-1&t=gNsRH05Kurt46kyJ-1

3.1 Input Design

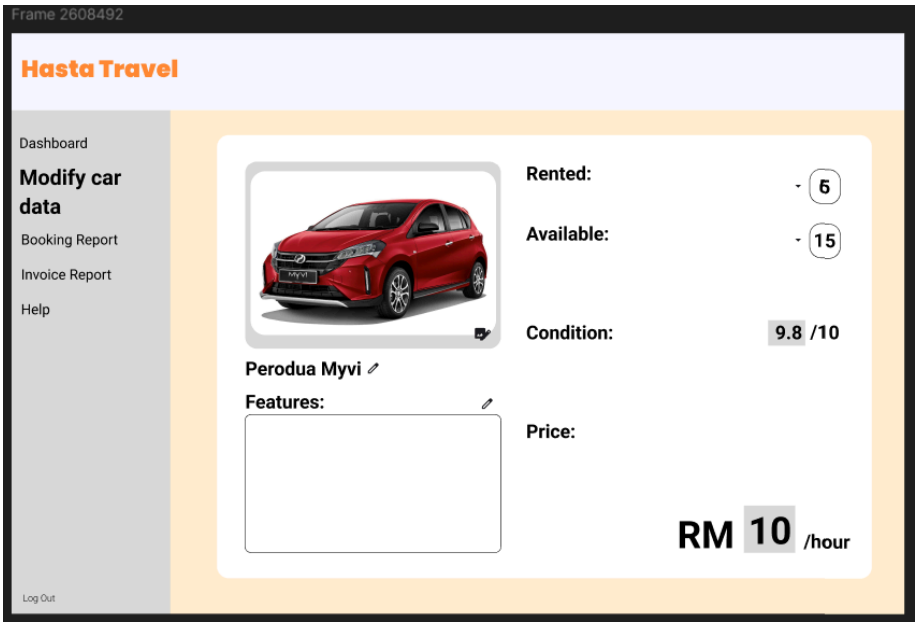
For the input design, we had made the interface more intuitive by identifying regions of input where the main inputs are grouped together, while secondary inputs are grouped together and placed in a lower region. This will ensure that the user may recognise what to input when given the interface for input.



Picture 2: Sign In Input Design



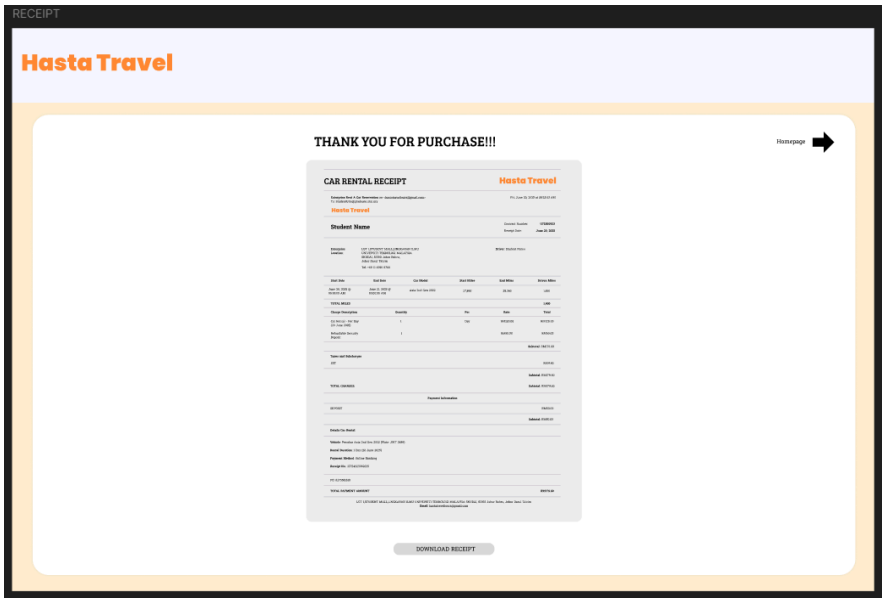
Picture 3: Booking Input Design



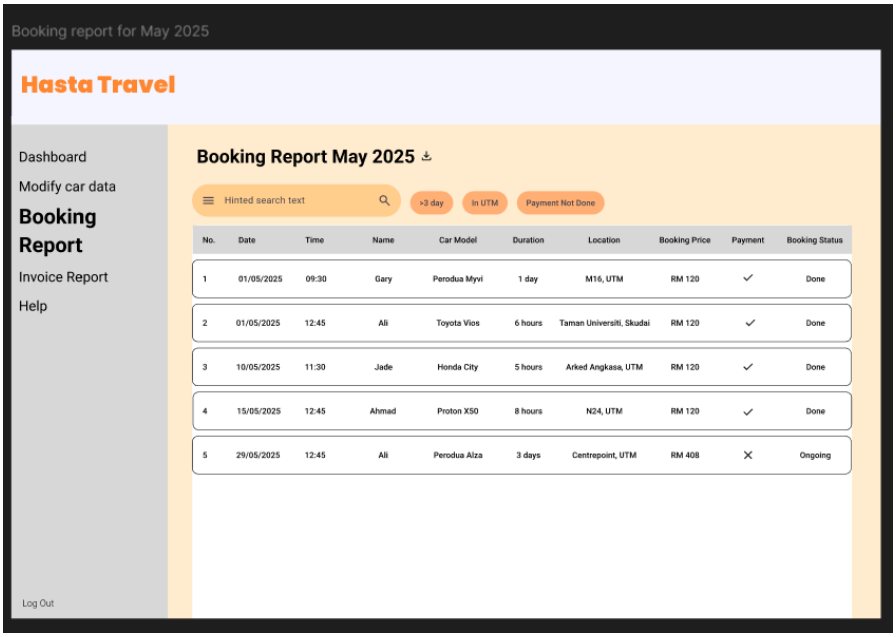
Picture 4: Update car Input Design

3.2 Output Design

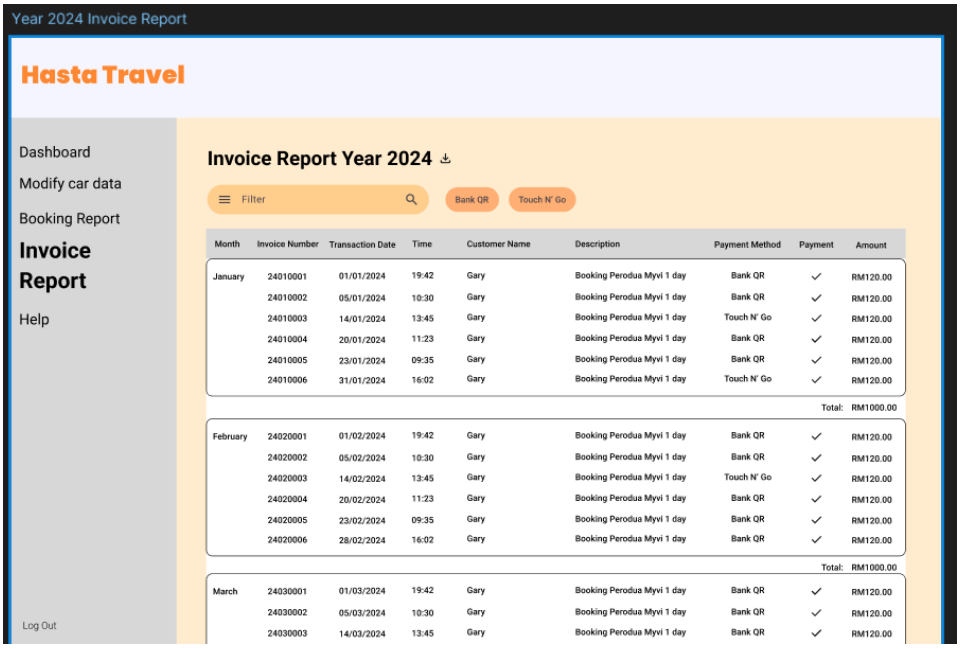
For the output design, we had decided to group together data of similar groups and to display the more important details at the left and right so that users may easily locate the required information.



Picture 5: Invoice Output Design



Picture 6: Monthly Report Output Design



Picture 7: Yearly Report Output Design

4.0 Summary

To ensure the design of the new system fulfills the requirements of the client, we have analysed the requirement and designed a system that may help in reducing the business loads and business cost. The system is also designed with a user-friendly interface to ease the user in adapting to the new system and also to avoid the new system from interfering with the business process.

After producing the wireframe of the new system, we may propose the design of the new system to the client to ensure the design is accepted by the client before implementation of the new system. After the design are approved, the implementation process may start and the production of the new system will begin.