Assignment 1

Analysis and Design Document

Car Rental System

(using GAIA methodology)

Project Group 1

Jacob Idoko

Aakash Sorathiya

Table of Contents

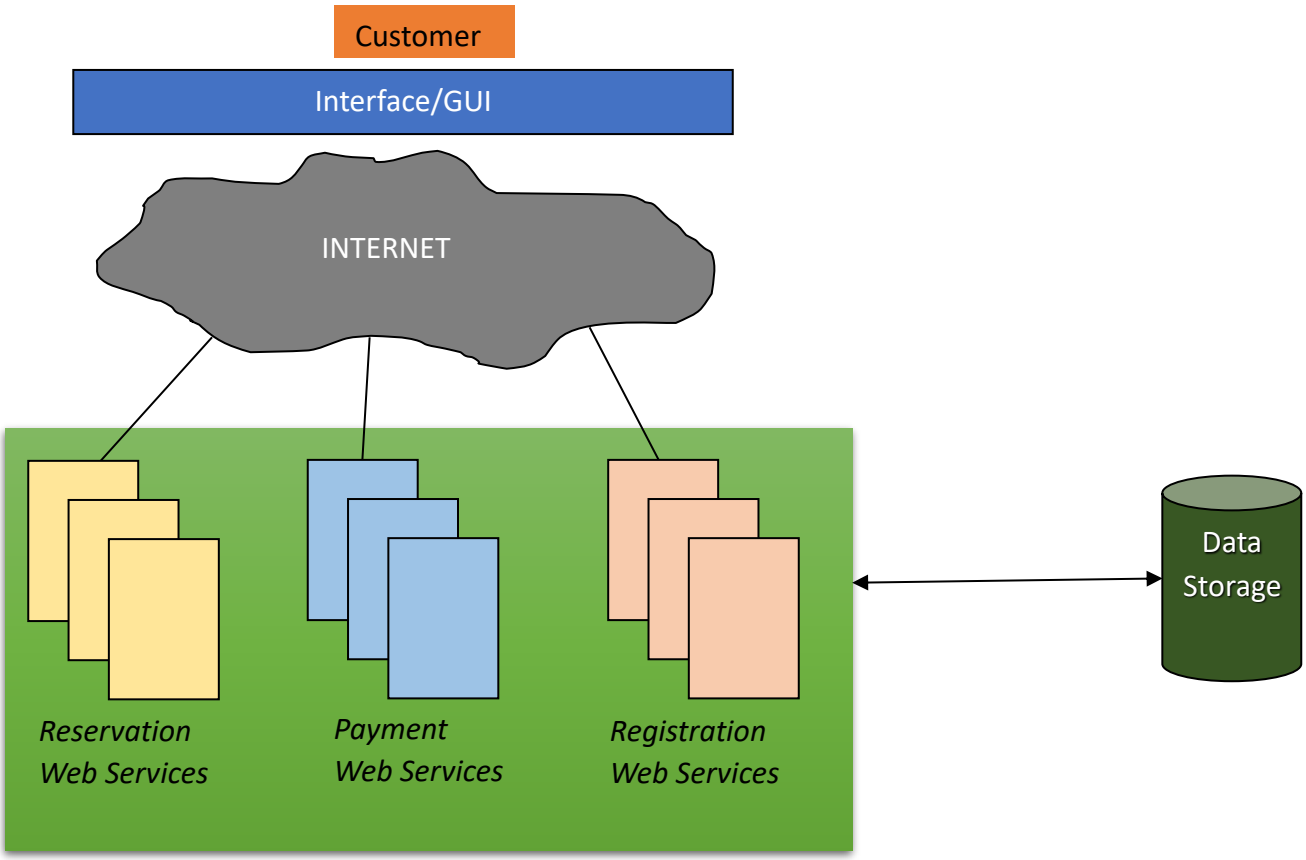
1 System Specifications

1.1 Problem Statement

- The car rental industry has seen a shift towards digital solutions, requiring a more efficient and user-friendly system for managing rentals.
- Customers need a convenient way to browse, book, and manage car rentals online.
- Car rental companies need to optimize their fleet management and streamline the rental process.
- There's a need for a system that can handle various aspects of car rental, including reservations, payments, and vehicle tracking.

1.2 System Description

- The proposed Car Rental System (CRS) is a multi-agent system designed to facilitate the process of renting cars.
- The system allows customers to register, browse available vehicles, make reservations, and manage their rentals.
- It also helps rental companies manage their fleet, track vehicle status, and process payments.



1.3 Assumptions

- The system maintains customer data, vehicle data, and rental transaction data.
- A Customer has two options:
    - A registered customer can log in using their email and password to access the system.
    - An unregistered customer can create a new account by entering their details.
- The customer can access the system to:
    - Browse available vehicles.
    - Make reservations.
    - Manage existing rentals.
    - Make payments.
- The Fleet Manager manages the vehicle inventory and availability.
- The Rental Agent processes reservations and handles customer inquiries.
- The Payment Processor handles all financial transactions.

1.4 Requirements

- The CRS shall provide access to the list of available vehicles.
- The CRS shall allow customers to make and manage reservations.
- The CRS shall process payments securely.
- The CRS shall register, de-register, and update customer data.
- The CRS shall keep track of vehicle availability and maintenance schedules.

1.5 Wishlist (Not Implemented)

- The CRS shall integrate with external mapping services to provide location-based recommendations.
- The CRS shall implement a loyalty program for frequent customers.
- The CRS shall provide real-time tracking of rented vehicles.
- The CRS shall offer a mobile app for on-the-go rentals and management.

2 System Analysis

2.1 Role Model

We have identified the following roles for our multi-agent Car Rental System:

- Verification
- Registration
- Vehicle Management
- Reservation Handling
- Payment Processing

2.2 Role Schema

Role Schema		**Verification**
Description		Verifies customer identity
Protocols and Activities		AuthenticateCustomer
Permissions		read customerdata write customerdata
Responsibilities	Liveness	authenticate = (Authenticate, Customer)
	Safety	successful authentication of a customer

Role Schema		**Registration**
Description		Registers/Deregisters a customer
Protocols and Activities		RegisterCustomer DeregisterCustomer UpdateCustomerData
Permissions		read customerdata write customerdata
Responsibilities	Liveness	register = (Register, Customer) deregister = (De-register, Customer) update = (Update.CustomerData, Customer)
	Safety	successful register/de-register a customer keep the data storage updated

Role Schema	**Vehicle Management**
Description	Manages vehicle inventory and availability

Protocols and Activities		AddVehicle RemoveVehicle UpdateVehicleStatus
Permissions		read vehicledata write vehicledata
Responsibilities	Liveness	add = (Add.Vehicle, Vehicle) remove = (Remove.Vehicle, Vehicle) update = (Update.VehicleStatus, Vehicle)
	Safety	maintain accurate vehicle inventory keep the data storage updated

Role Schema		**Reservation Handling**
Description		Manages customer reservations
Protocols and Activities		CreateReservation CancelReservation ModifyReservation
Permissions		read reservationdata write reservationdata read vehicledata
Responsibilities	Liveness	create = (Create.Reservation, Reservation) cancel = (Cancel.Reservation, Reservation) modify = (Modify.Reservation, Reservation)
	Safety	ensure reservation integrity update vehicle availability accordingly

Role Schema		**Payment Processing**
Description		Handles financial transactions
Protocols and Activities		ProcessPayment IssueRefund
Permissions		read paymentdata write paymentdata
Responsibilities	Liveness	process = (Process.Payment, Payment) refund = (Issue.Refund, Refund)
	Safety	secure transaction processing maintain accurate financial records

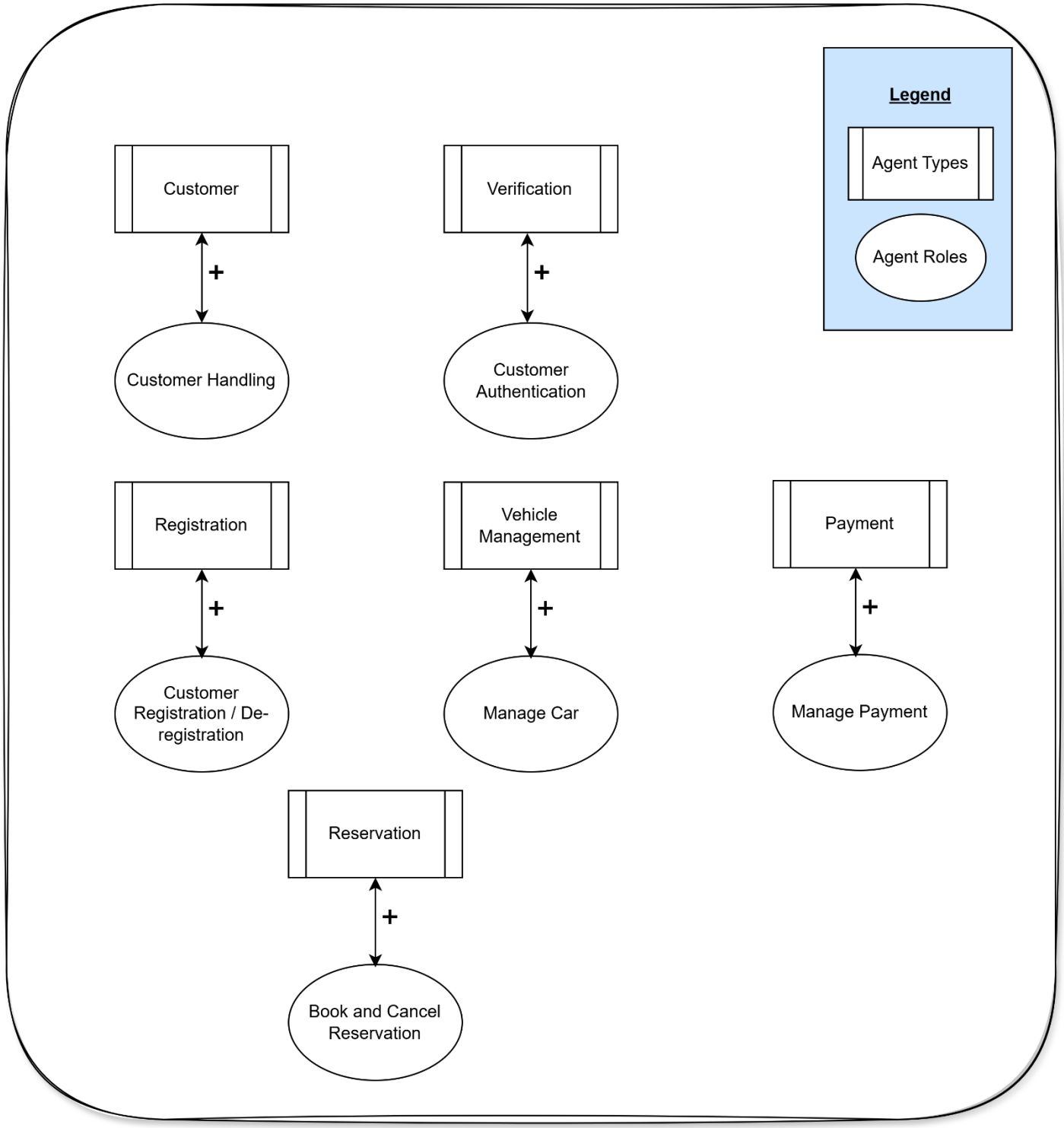
2.3 Interaction Model

Protocol	RegisterCustomer	DeregisterCustomer	AuthenticateCustomer
Purpose/ Parameters	Register a customer	Deregister a customer	Authenticate a customer
Initiator	Customer	Customer	Customer
Receiver	Registration	Registration	Verification
Responding Protocol	Customer registration successful	Customer de-registration successful	Customer authenticated successfully

Protocol	CreateReservation	CancelReservation	ProcessPayment	UpdateVehicleStatus
Purpose/ Parameters	Create a new reservation	Cancel an existing reservation	Process a rental payment	Update vehicle availability
Initiator	Customer	Customer	Customer	Vehicle Management
Receiver	Reservation Handling	Reservation Handling	Payment Processing	Data Manager
Responding Protocol	Reservation created successfully	Reservation cancelled successfully	Payment processed successfully	Vehicle status updated successfully

3 System Design

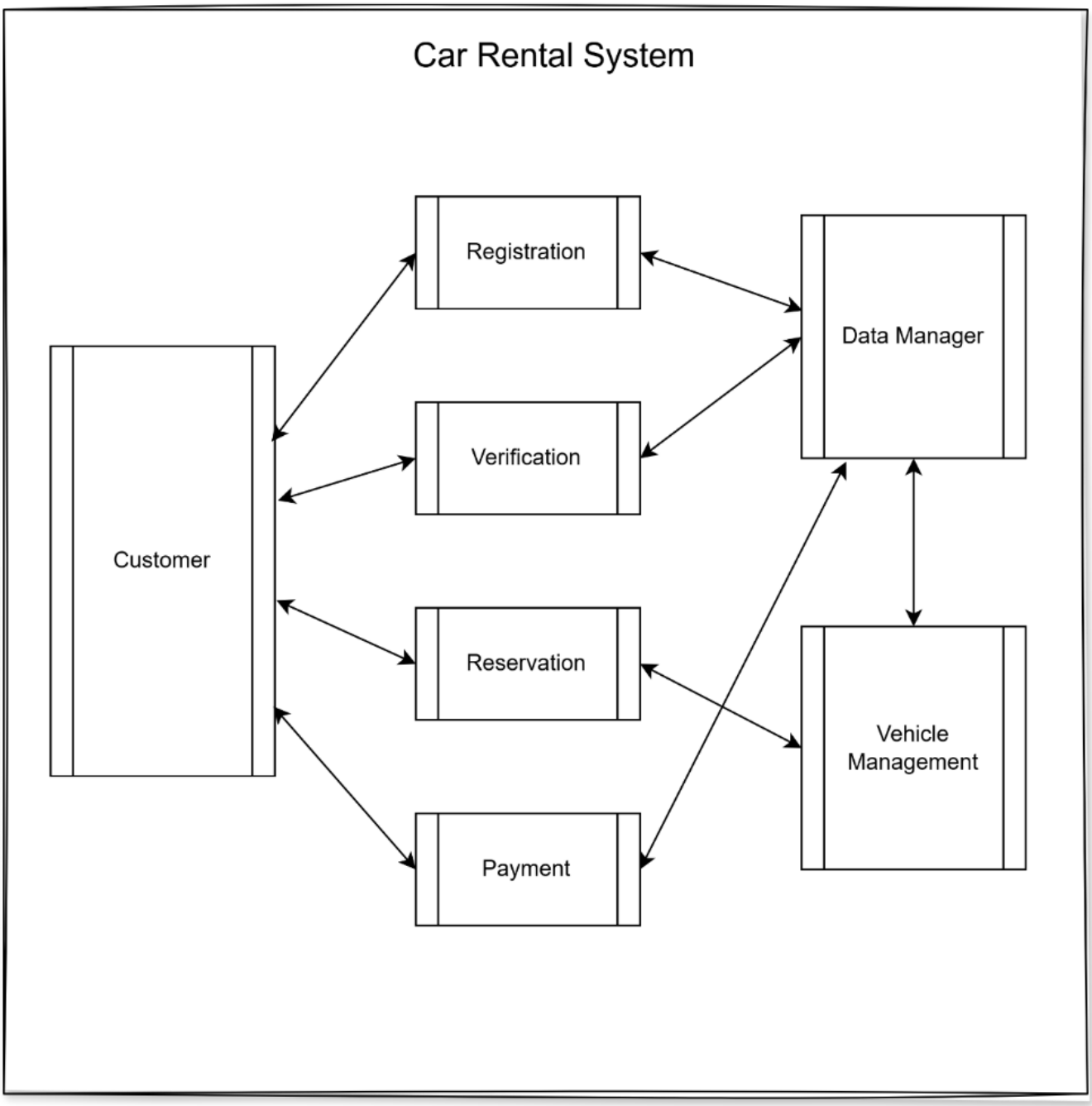
3.1 Agent Model



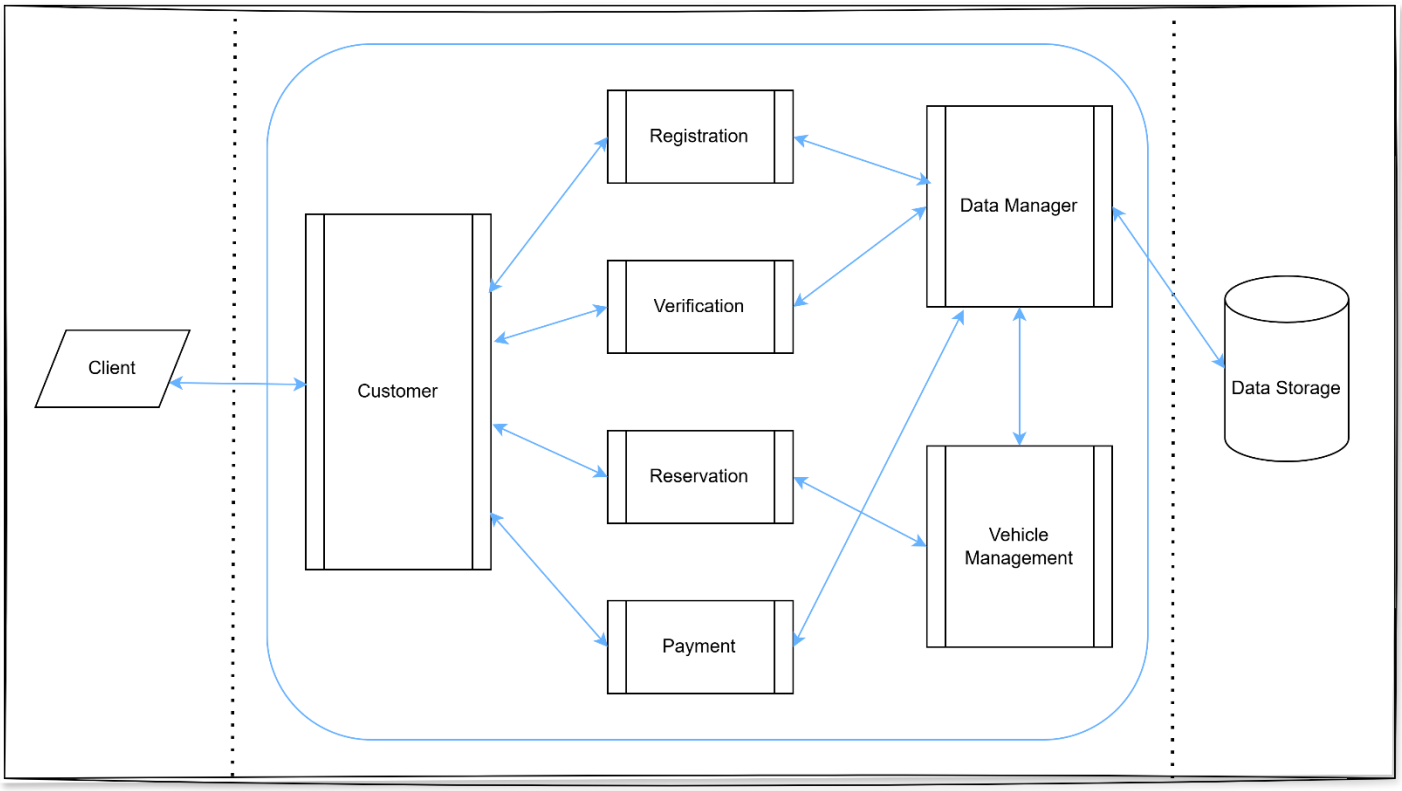
3.2 Services Model

Service	Inputs	Outputs	Pre-conditions	Post-conditions
Customer Authentication	Customer credentials	Authentication status	Launch the Customer GUI	Send a request to the data storage to verify customer data
Vehicle Reservation	Vehicle details, rental period	Reservation confirmation	Go to the Reservation GUI	Send a request to update vehicle availability and create reservation record
Payment Processing	Payment details	Payment confirmation	Go to the Payment GUI	Send a request to process payment and update financial records
Vehicle Management	Vehicle details	Vehicle status update	Access Vehicle Management GUI	Send a request to update vehicle data in storage

3.3 Acquaintance Model



4 Multi-Agent System Architecture



The CRS is designed as a multi-agent system where various agents interact to provide a comprehensive rental service:

- The Customer Agent serves as the primary interface for users, handling all customer interactions and relaying requests to appropriate specialized agents.
- The Verification and Registration Agents manage customer accounts and authentication.
- The Reservation Agent coordinates with the Vehicle Management Agent to handle bookings and ensure vehicle availability.
- The Payment Agent securely processes all financial transactions.
- All agents communicate with the Data Manager to maintain consistent and up-to-date system information.

5 Agent Description

5.1 Customer Agent

- Acts as the main point of contact for customers through the web interface or mobile app.
- Coordinates with other agents to fulfill customer requests (authentication, registration, reservations, payments).
- Provides a unified interface for all customer interactions with the system.

5.2 Verification Agent

- Handles customer authentication processes.
- Communicates with the Data Manager to verify customer credentials and maintain security.

5.3 Registration Agent

- Manages customer account creation, updates, and deletion.
- Ensures customer data integrity and consistency in the system.

5.4 Reservation Agent

- Processes reservation requests, modifications, and cancellations.
- Coordinates with the Vehicle Management Agent to check and update vehicle availability.

5.5 Vehicle Management Agent

- Maintains the vehicle inventory and tracks vehicle status (available, rented, under maintenance).
- Updates vehicle information and availability in real-time.

5.6 Payment Agent

- Handles all financial transactions, including rental payments and refunds.
- Ensures secure and accurate processing of payments.

5.7 Data Manager

- Central repository for all system data.
- Provides data access and update capabilities to all other agents, ensuring data consistency across the system.