### **1.5 Assumptions:**

Some potential assumptions are derived from the information outlined in the project description.

1. Stable Internet Connection:

Assumption: Users have a stable and reliable internet connection to access the online system.

Explanation: The system is designed for online transactions and data management, so a stable internet connection is crucial.

2. User Training:

Assumption: Users will receive adequate training to use the system effectively.

Explanation: The assumption is that users may not require extensive training due to the system's user-friendly design, but it's crucial to ensure they understand its functionalities.

3. Data Accuracy:

Assumption: Data entered into the system is assumed to be accurate and validated by the users.

Explanation: The system relies on accurate data for effective management, and assumptions about data accuracy are common in project planning.

4. Security Measures:

Assumption: The implemented security measures are assumed to be sufficient to protect the system from unauthorised access and data breaches.

Explanation: The project assumes that the security measures, such as user authentication and authorisation, are effective.

5. User Availability:

Assumption: Users are assumed to be available for testing, training, and feedback during the development and implementation phases.

Explanation: The project's success may depend on the active participation and cooperation of the end users.

6. Compliance with Regulations:

Assumption: The project assumes the system will comply with relevant laws and regulations governing car showrooms and online transactions.

Explanation: It is crucial to comply with legal requirements to avoid legal issues and ensure the system's ethical use.

7. Budget Constraints:

Assumption: The project assumes that the allocated budget is sufficient for the system's development, testing, and implementation.

Explanation: Budget constraints can impact project timelines, resource allocation, and success.

8. Compatibility:

Assumption: The system assumes compatibility with commonly used browsers and devices.

Explanation: Users may access the system from different devices and browsers, and assuming compatibility can simplify development efforts.

9. System Performance:

Assumption: The system is assumed to perform efficiently under normal operating conditions.

Explanation: Performance assumptions may include expectations about response times, concurrent user handling, and data processing speed.

## **2. Project Deliverables:**

The project aims to develop an integrated and responsive online application system. This report provides a detailed account of the system's functional and non-functional requirements and offers a comprehensive overview of its design and architecture. Below are detailed explanations of the key deliverables for each phase:

1. Planning Phase:

Project Charter:

Description: A document formally authorises the project, outlining its objectives, scope, stakeholders, and high-level requirements.

Purpose: Sets the foundation for the project and aligns stakeholders on the project's goals.

Project Schedule:

Description: A detailed timeline outlining project milestones, tasks, and dependencies.

Purpose: Provides a roadmap for project execution, helping to manage time effectively.

2. Analysis Phase:

Requirements Specification:

Description: A detailed document outlining functional and non-functional requirements for the system.

Purpose: As a foundation for design and development, ensuring alignment with stakeholder expectations.

Feasibility Study Report:

Description: An assessment of the project's feasibility, considering technical, economic, and operational factors.

Purpose: Assists decision-making and resource allocation based on the project's viability.

3. Design Phase:

System Architecture Design:

Description: High-level design detailing the system's architecture, components, and interactions.

Purpose: Guides developers in building a scalable and maintainable system.

Database Design:

Description: Design specifications for the database structure, tables, relationships, and constraints.

Purpose: Provides a blueprint for efficient data storage and retrieval.

User Interface (UI) Design:

Description: Design mockups and wireframes illustrating the user interface.

Purpose: Ensures a user-friendly and intuitive interface aligning with user expectations.

4. Implementation Phase:

Source Code:

Description: The actual codebase was developed to implement the system's functionality.

Purpose: Forms the basis for the system's execution and functionality.

Database Implementation:

Description: Creation and configuration of the database based on the design specifications.

Purpose: Establishes the data storage foundation for the application.

5. Testing Phase:

Test Plans and Test Cases:

Description: Documents outlining the testing strategy, scenarios, and specific test cases.

Purpose: Guides the testing process to ensure comprehensive coverage.

Test Results and Bug Reports:

Description: Records of test execution results and identified bugs or issues.

Purpose: Provides insights into the system's quality and facilitates issue resolution.

6. Deployment Phase:

Deployment Plan:

Description: A detailed plan outlining the steps for deploying the system into a production environment.

Purpose: Ensures a smooth and controlled system transition from development to production.

User Training Materials:

Description: Documentation or training materials to help users understand and navigate the new system.

Purpose: Facilitates user onboarding and adoption of the new system.

7. Maintenance Phase:

Maintenance Plan:

Description: Guidelines for ongoing system maintenance, updates, and support.

Purpose: Ensures the long-term sustainability and performance of the system.

Incident Reports and Resolutions:

Description: Documentation of incidents, their resolutions, and lessons learned.

Purpose: Supports continuous improvement and informs future decision-making.

## **3. System Requirements:**

### **3.1 Functional Requirements:**

### **3.2 Non-Functional Requirements:**

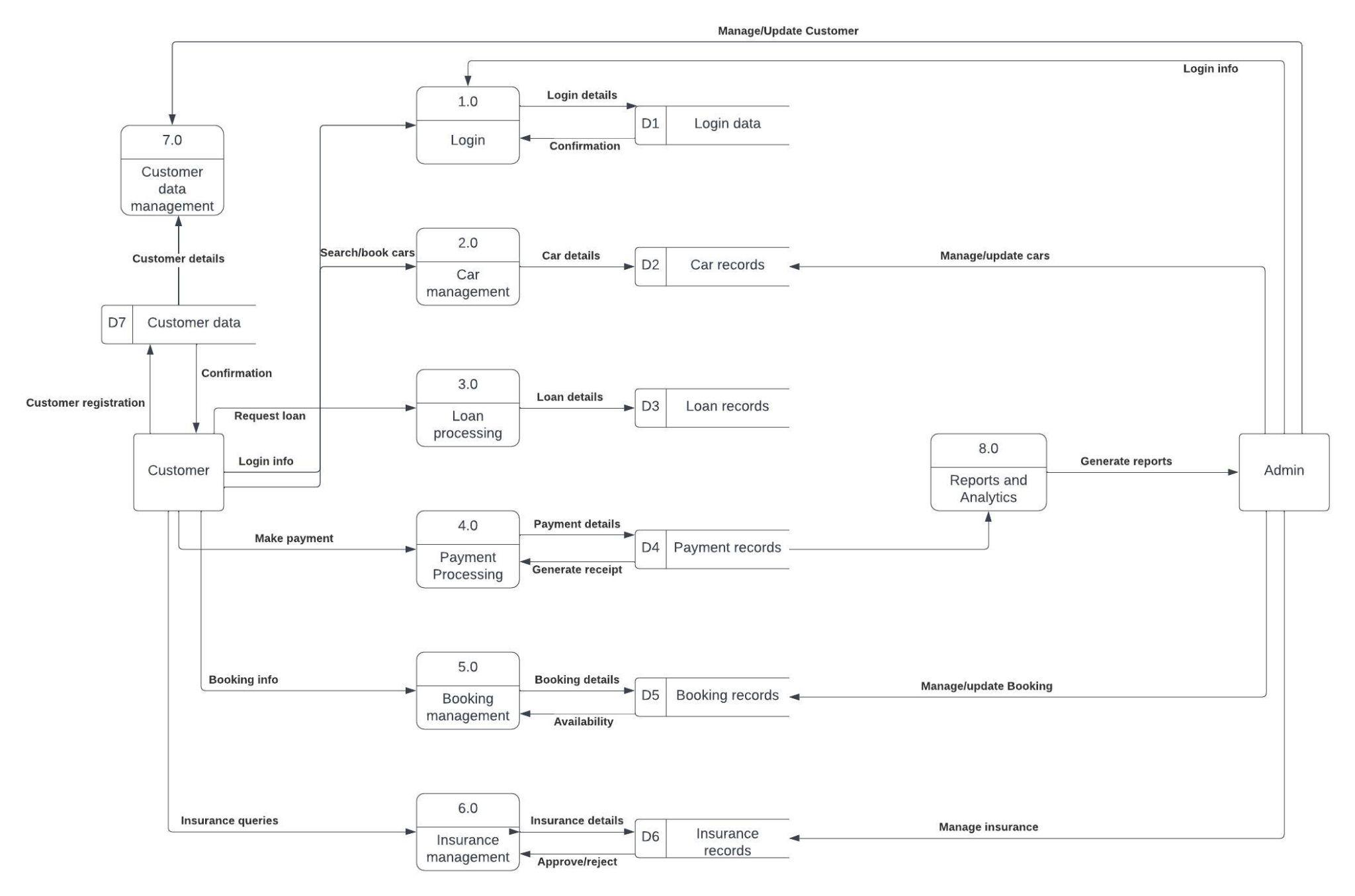
|  |  |  |
| --- | --- | --- |
| NFR No. | Non-Functional Requirements | Comments |
|  | **Security:** | |
| 1. | Authentication and Authorization | User authentication through secure mechanisms (e.g., password encryption, multi-factor authentication). |
| 2. | Login | All users must login with a registered email and password. |
| 3. | Password | The password must contain over 8 characters, including capital letters, numbers and symbols. |
| 4. | Data Encryption | All the data is stored in a database and encrypted using MD5 security. |
|  | **Availability:** | |
| 5. | System Uptime | The system should have a high availability rate, with a minimum of 99.9% uptime. |
| 6. | Backup and Disaster Recovery | Regular automated backups and disaster recovery of the database and system configurations to minimise downtime in case of system failures. |
|  | **Performance:** | |
| 7. | Response Time | The system should respond to user requests within 2 seconds for optimal user experience. |
| 8. | Resource Utilization | Efficient use of system resources such as CPU memory and storage to optimise performance/ |
| 9. | Scalability | The system should be able to handle increased loads during peak times without a significant drop in performance. |
|  | **Reliability:** | |
| 10. | Error Handling | The system should have robust error-handling mechanisms to handle unexpected errors. |
| 11. | Functions | All functions should work as expected and meet the user requirements without errors. |
|  | **Accessibility:** | |
| 12. | Cross-Browser Compatibility | The system should be accessible from web browsers like Google Chrome, Safari, and Microsoft Edge. |
| 13. | Compliance with Accessibility Standards | Ensure compliance with accessibility standards (e.g., WCAG) for users with disabilities. |
|  | **Scalability:** | |
| 14. | Horizontal and Vertical Scalability | The system should support adding more servers and upgrading server scalability. |
| 15. | Load Balancing | Load balancing to distribute incoming traffic evenly across multiple servers. |
|  | **Usability:** | |
| 16. | User Interface | The user interface should be intuitive, with straightforward navigation and user-friendly interactions. |
| 17. | Instructions/Guide | The instructions should be clear and easy to understand. |
| 18. | Response to User Feedback | Establish a mechanism to gather user feedback and continuously improve the systems’ usability. |
| 19. | Responsive Design | The system should be responsive depending on the devices. Able to work well on Smartphones, Tablets, Laptops and the Web. |

### 

### 

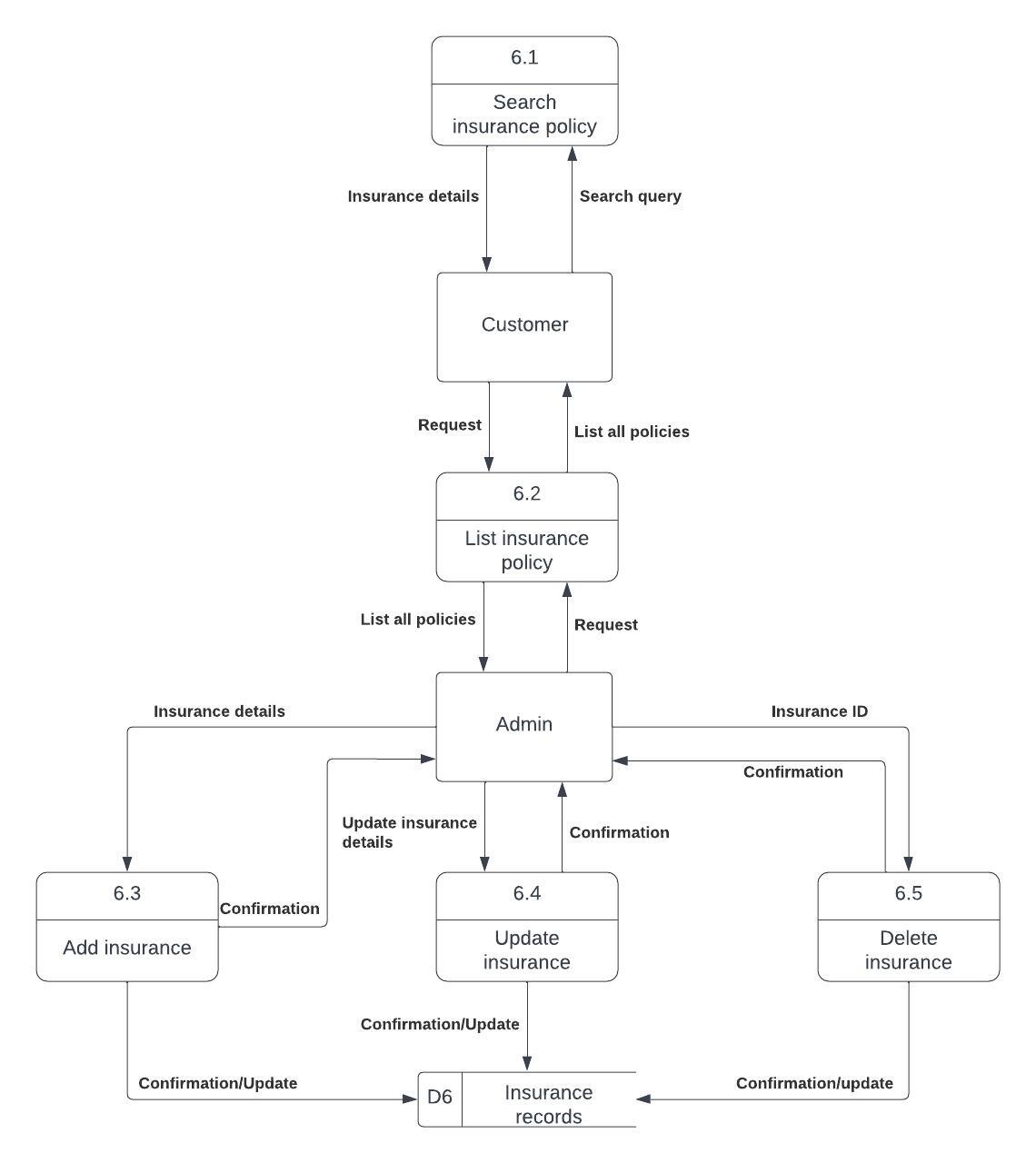
## **7. Data Flow Diagram:**

### **7.1 DFD Level 1:**

****

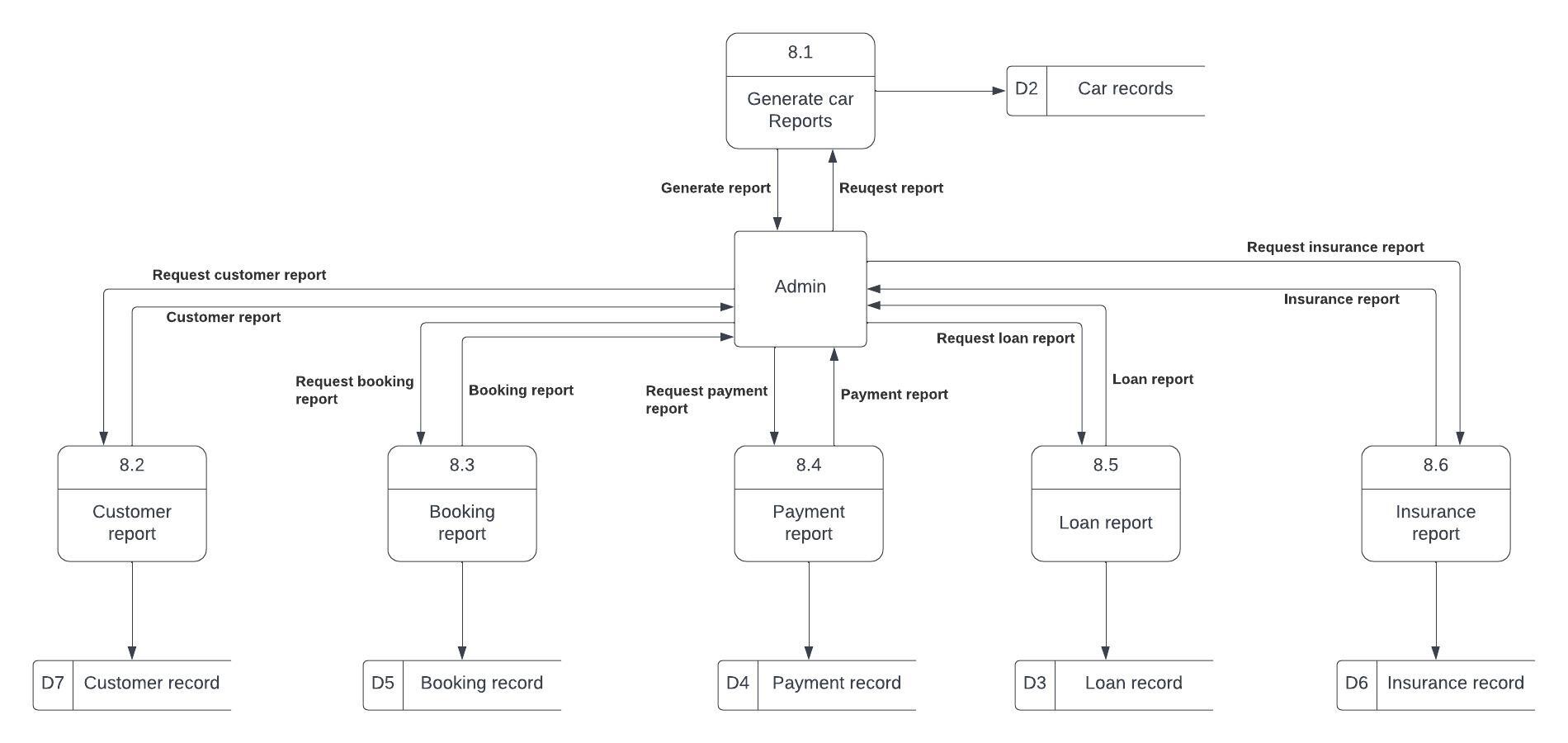
### **7.2 DFD Level 2:**

**Insurance Management:**

****

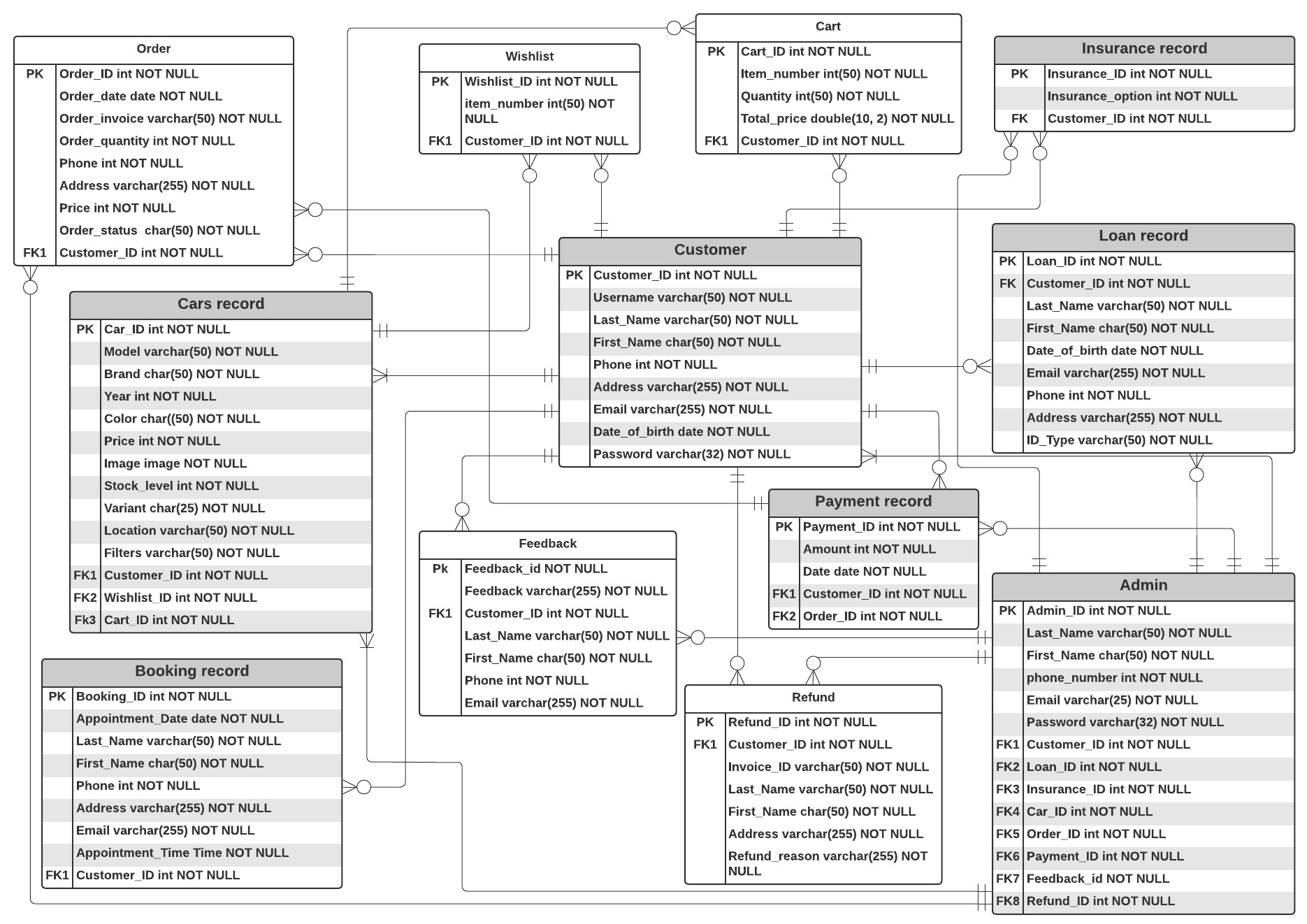
## 

**Report:**



## 

## **10. Entity Relationship Diagram:**



## 9.1 User Interface:

