# Functional requirements:

**Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR.NO** | **Functional Requirements** | **Comments** | **components** |
| FR1 | Admin Login | Admin should be able to log in into their account using their user name and password | login.php |
| FR2 | View Dashboard | The admin should be able to view a dashboard with information of existing users, new users, total reservations and revenue. | Admin\_dashboard.php |
| FR3 | Create Scheduling Tasks | The admin should be able to run scheduling tasks, appointments, approve purchase orders. | Managetask.php |
| FR4 | View Vehicle Inventory | The admin should be able to manage the inventory of cars, able to add or delete the vehicles. | Edit\_inventory.php |
| FR5 | View Available Vehicles | The admin should be able to see the list of available vehicles. | Available\_vehicles.php |
| FR6 | View Stock Levels | The admin should be able to view the stock level of the vehicle. | Available\_stocks.php |
| FR7 | Generate Notification Alert | The admin will be notified about the stocks when stock falls under the minimum level. | Notification.php |
| FR8 | Manage order | The admin should be able to manage orders, view the order details, updating processing, shipped, delivered and cancel if necessary. | Order\_management.php |
| FR9 | View Bookings | The admin should be able to view all the booking made by users including user information, car details, cost, and payment type. | Booking\_list.php |
| FR10 | Manage users | The admin should be able to manage all signed users with their personal details such as name, address, email and contact number. | Edit\_users.php |
| FR11 | Manage Accessories | The admin able to update available parts, accessories, availability and low stock details. | Edit\_accessories.php |
| FR12 | Manage Shipping | The admin should be able to send notification about shipping information to the users with product details, processing time, delivery date with tracking code. | Shipping\_response.php |
| FR13 | Maintain Records | The admin should be able to identify and store record following company policies, standards and guidelines. | History\_records.php |
| FR14 | Manage Payment | The admin should be able sign in to the payments profile, user record, user access permission type. | Paymemt\_verification.php |
| FR15 | Create Vendor Invoice | The admin should be able to make a vendor invoice, make PO from purchase request, and add import cost on vehicles. | Invoice\_ verification.php |
| FR16 | Manage Refund Request | The admin should be able to approve a refund request and create a return invoice with refund receipt for multiple vehicles. | Refund\_management.php |

**Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR NO.** | **Functional Requirements** | **Comments** | **Components** |
| FR17 | User registration | Allows to users to sign up to the system and create their profile. | Registration.php |
| FR18 | User Login | Allows users to login to their account using registered email and passwords. | User\_login.php |
| FR19 | Car search Option | Allows users to search for cars based on various criteria, such as make, model, price range, and features. | Search\_cars.php |
| FR20 | View Car Categories | Allows users to view the categories of the vehicle by brands, models, accessories parts. | Details\_view.php |
| FR21 | Choose Filter Options | Allows users to select a filter options like choosing vehicle brands, colour, models, in the categories | Filter\_by\_categories.php |
| FR22 | View Vehicle Information | Allows users to view the details like country of origins, location, showroom, brand, models of the selected cars. | Car\_details.php |
| FR23 | Watch Videos | Users will be able to watch a videos on the website about the cars. | Car\_details.php |
| FR24 | Make a Purchase | Allows users to make a purchase a vehicle based on custom sales price per model, interior colour, added accessories and added extra charges by vehicle code. | Purchase.php |
| FR25 | Create Loan Application | Enables users to apply for a loan to finance their car purchase, providing necessary financial details. | Apply\_loan.php |
| FR26 | Create a Booking | Allows users to schedule test drives or book a specific car for purchase, providing flexibility in the buying process. | book\_car.php |
| FR27 | Create Customer profile | Let’s users manage their personal information, preferences, and track their interaction history with the showroom. | Account\_profile.php |
| FR28 | Create a payment | Facilitates the processing of payments for car purchases, loans, or other financial transactions. Also Users will be able to choose a payment option like bank or credit card. | Process\_payment.php |
| FR29 | View Shopping Cart | Allows users to view the shopping cart based on their selected items and be able to proceed to checkout. | Shoppiong\_cart.php |
| FR30 | Create Feedbacks and Reviews | Allows users to provide feedback on their experience, rate purchased cars, and leave reviews for the benefit of other users. | Feedbacks\_page.php |
| FR31 | View Notifications Alert | Sends users alerts and notifications regarding their bookings, payments, or any relevant updates. | Notification.php |
| FR32 | Add to Wishlist | Enables users to create and manage a wish list of cars they are interested in but have not yet purchased. | Edit\_wishlist.php |
| FR33 | Create History And Records | Displays the user's purchase history, including details of past transactions, test drives, and interactions with the showroom. | Purchase\_history.php |
| FR34 | Use Price Calculator | Allows users to calculate sales price for selected vehicles with discount percentage. | Calculate\_total.php |
| FR35 | Make Pre-order/Reserve Request | Allows user to pre-order or reserve the vehicle. | Preorder\_car.php |
| FR36 | Create Order Inquiry | Allows users to create an inquiry with the customer service. | Order\_inquiry.php |
| FR37 | Create a Refund Request | Allows users to able to make a refund request. | Refund.php |
| FR38 | Use AI API Chatbox | Allows users to access the AI chatbox. When asked a question, the chatbot will answer using the knowledge database that is currently available to it. | Chat\_bot.php |

## Non-Functional Requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| NFR No. | Non-Functional Requirements | Comments | Components |
|  | Security: | | |
| 1. | Authentication and Authorization | User authentication through secure mechanisms (e.g., password encryption, multi-factor authentication). | Jwt,auth2.0 |
| 2. | Login | All users must login with a registered email and password. | Md5, jwt |
| 3. | Password | The password must contain over 8 characters, including capital letters, numbers and symbols. | MD5 |
| 4. | Data Encryption | All the data is stored in a database and encrypted using MD5 security. | phpMysql, md5 |
|  | Availability: | | |
| 5. | System Uptime | The system should have a high availability rate, with a minimum of 99.9% uptime. | Apache web server |
| 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. | Mysqldump utility, mysql |
|  | Performance: | | |
| 7. | Response Time | The system should respond to user requests within 2 seconds for optimal user experience. | Apache jmeter and loadrunner |
| 8. | Resource Utilization | Efficient use of system resources such as CPU memory and storage to optimise performance/ | Monitoring tools (ie. Munin, magios) |
| 9. | Scalability | The system should be able to handle increased loads during peak times without a significant drop in performance. | Multiple apache server |
|  | Reliability: | | |
| 10. | Error Handling | The system should have robust error-handling mechanisms to handle unexpected errors. | Try catch mechanism, log error |
| 11. | Functions | All functions should work as expected and meet the user requirements without errors. | phpUnit |
|  | Accessibility: | | |
| 12. | Cross-Browser Compatibility | The system should be accessible from web browsers like Google Chrome, Safari, and Microsoft Edge. | Google chrome, safari |
| 13. | Compliance with Accessibility Standards | Ensure compliance with accessibility standards (e.g., WCAG) for users with disabilities. | “POUR” PRINCIPLES |
|  | Scalability: | | |
| 14. | Horizontal and Vertical Scalability | The system should support adding more servers and upgrading server scalability. | Multiple web servers |
| 15. | Load Balancing | Load balancing to distribute incoming traffic evenly across multiple servers. | Apache jmeter and loadrunner |
|  | Usability: | | |
| 16. | User Interface | The user interface should be intuitive, with straightforward navigation and user-friendly interactions. | Bootstrap css or tailwind css |
| 17. | Instructions/Guide | The instructions should be clear and easy to understand. | Documentation |
| 18. | Response to User Feedback | Establish a mechanism to gather user feedback and continuously improve the systems’ usability. | Iterative model |
| 19. | Responsive Design | The system should be responsive depending on the devices. Able to work well on Smartphones, Tablets, Laptops and the Web. | Bootstrap |

# Logical view of the system

**Presentation Layer:**

Admin and customer interfaces are where users interact. Users use the frontend for actions, like making requests and receiving responses.

**Application Layer:**

The frontend connects with the Application Layer. Application Layer has a Server and Cluster for system tasks. Security Layer adds extra protection. It helps communication between the presentation and data layers.

**Data Layer:**

Databases are vital for the system:

Cars­\_db: Info about available cars.

User\_db: User-related data.

Payment\_db: Deals with payment info.

Supplier\_db: Holds supplier details.

**Flow of Information:**

Admin and customer actions start in the presentation layer.

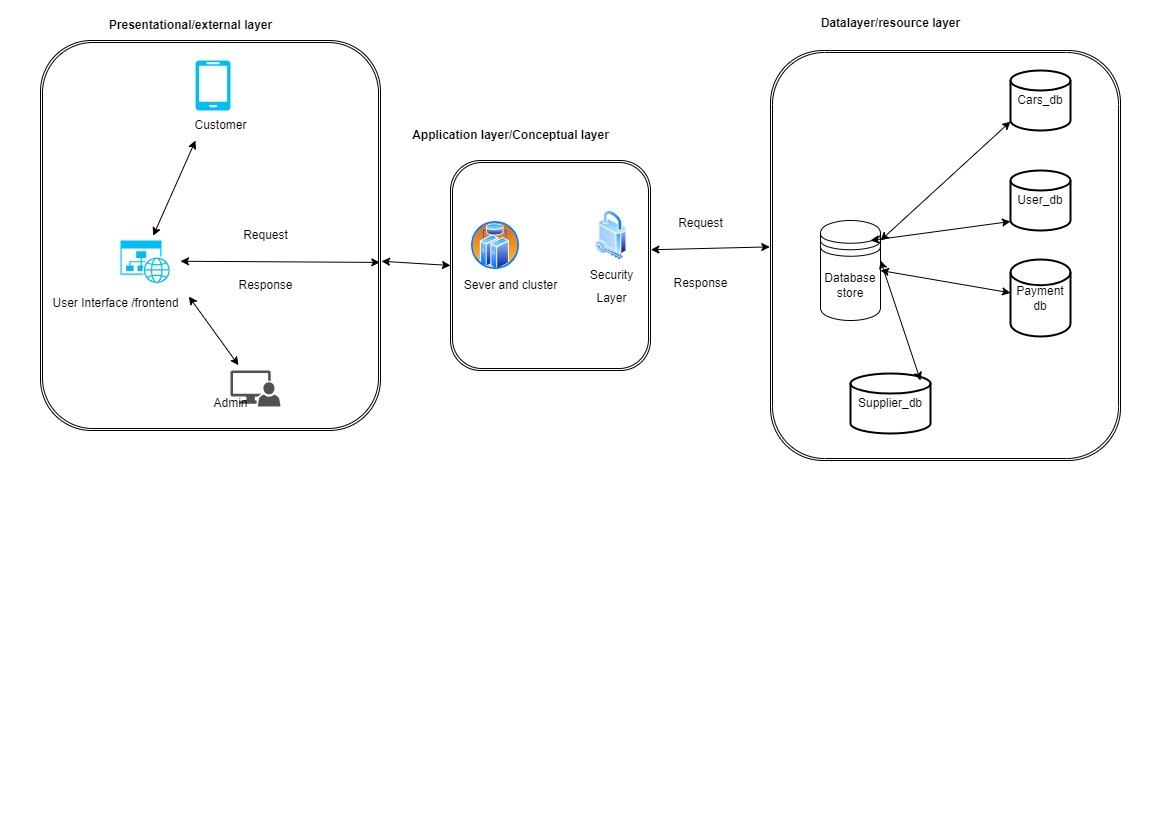
Requests and responses move through the frontend.

The Application Layer (server, cluster, security) handles communication.

Data Layer stores essential databases (cars, user, payment, and supplier).

Requests and responses smoothly move between these layers, meeting user needs and keeping the system running.

Figure: Logical view of the system



Functional view of the system

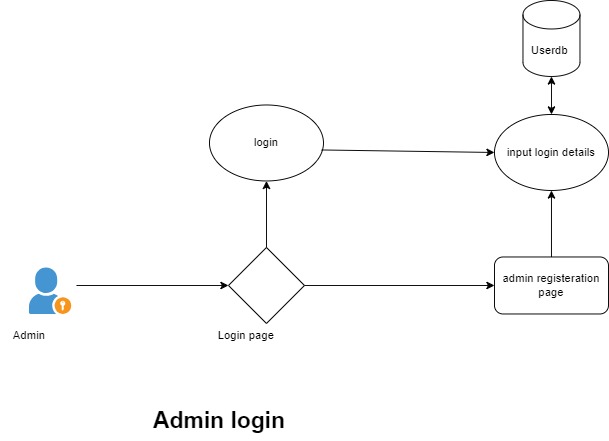


Figure 1: Admin login

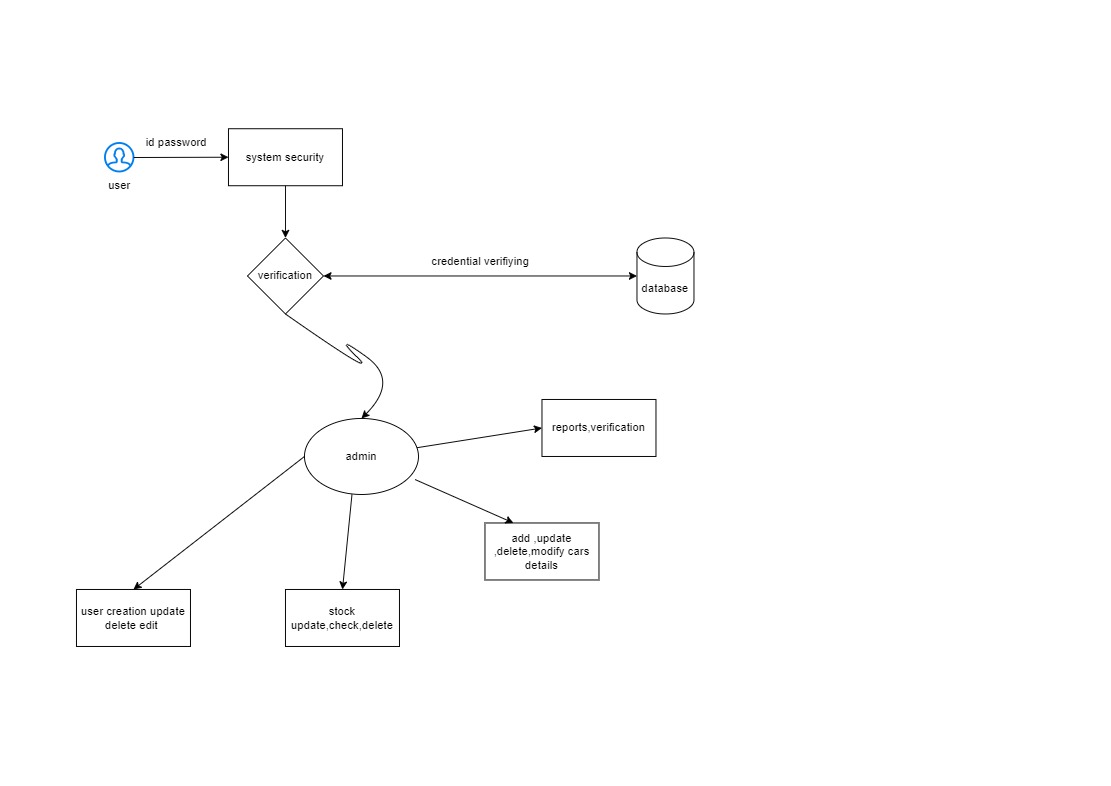


Figure 2: View Dashboard

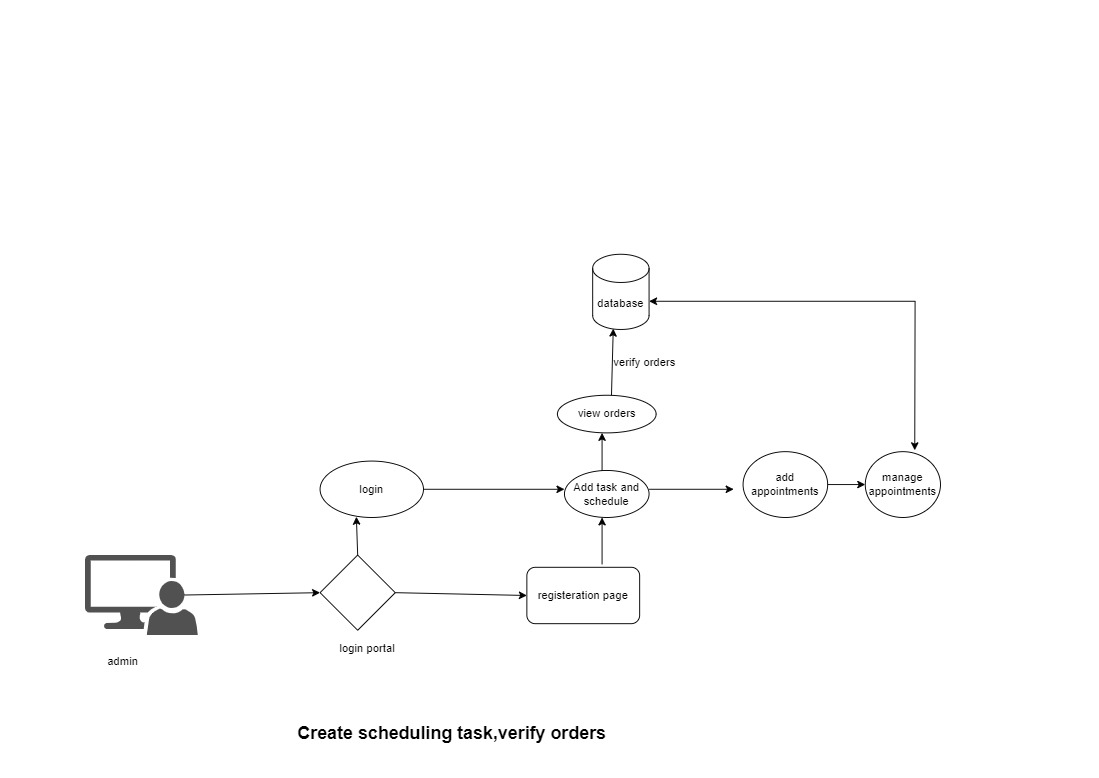


Figure 3: Create Scheduling task

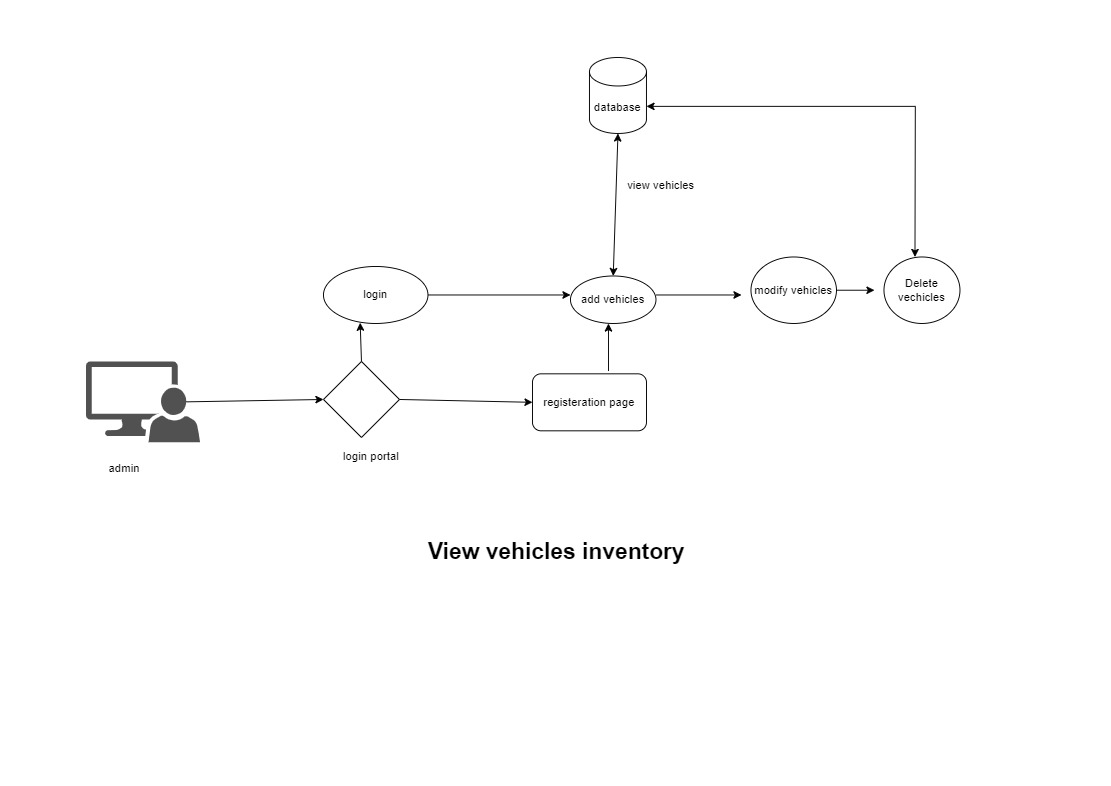


Figure 4: View vehicle inventory

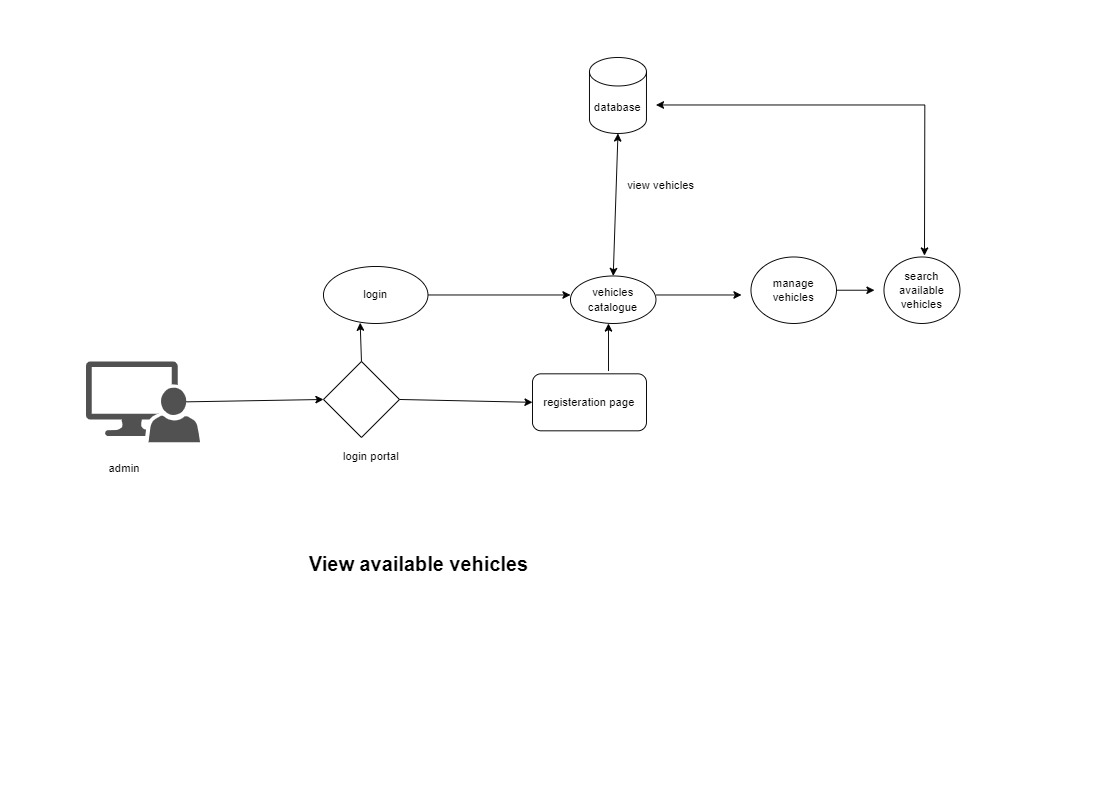


Figure 5: View available vehicles

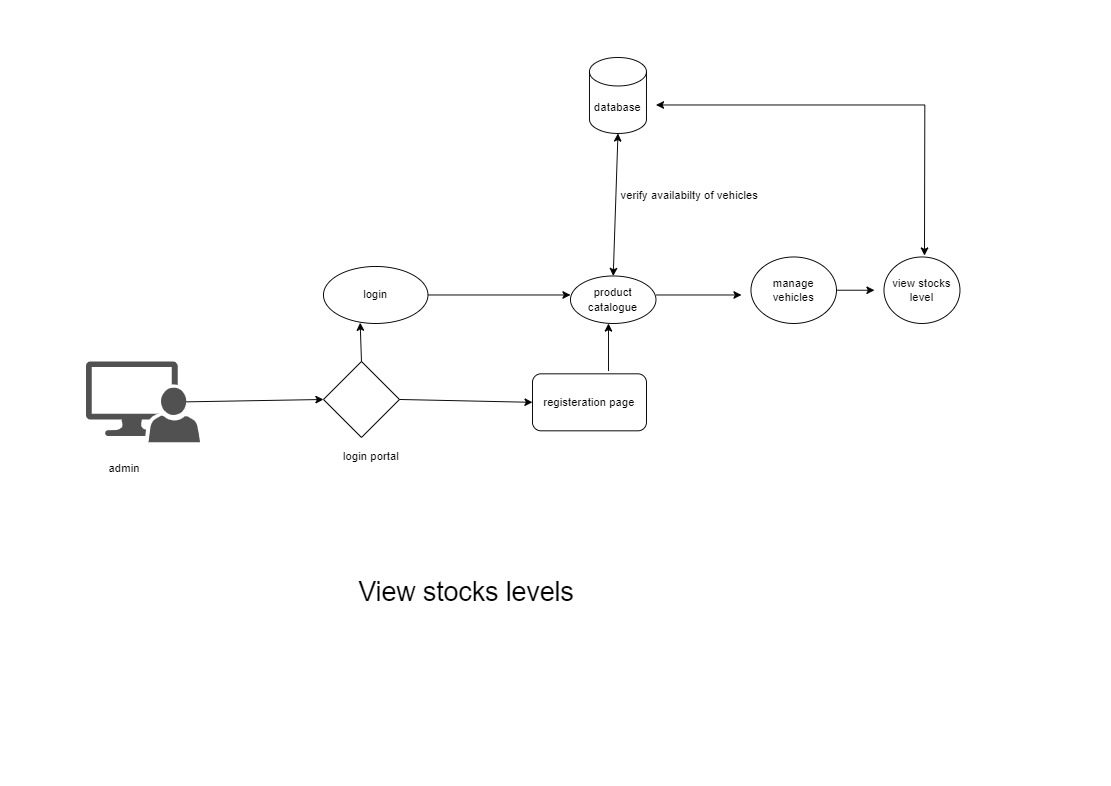


Figure 6: View stocks levels

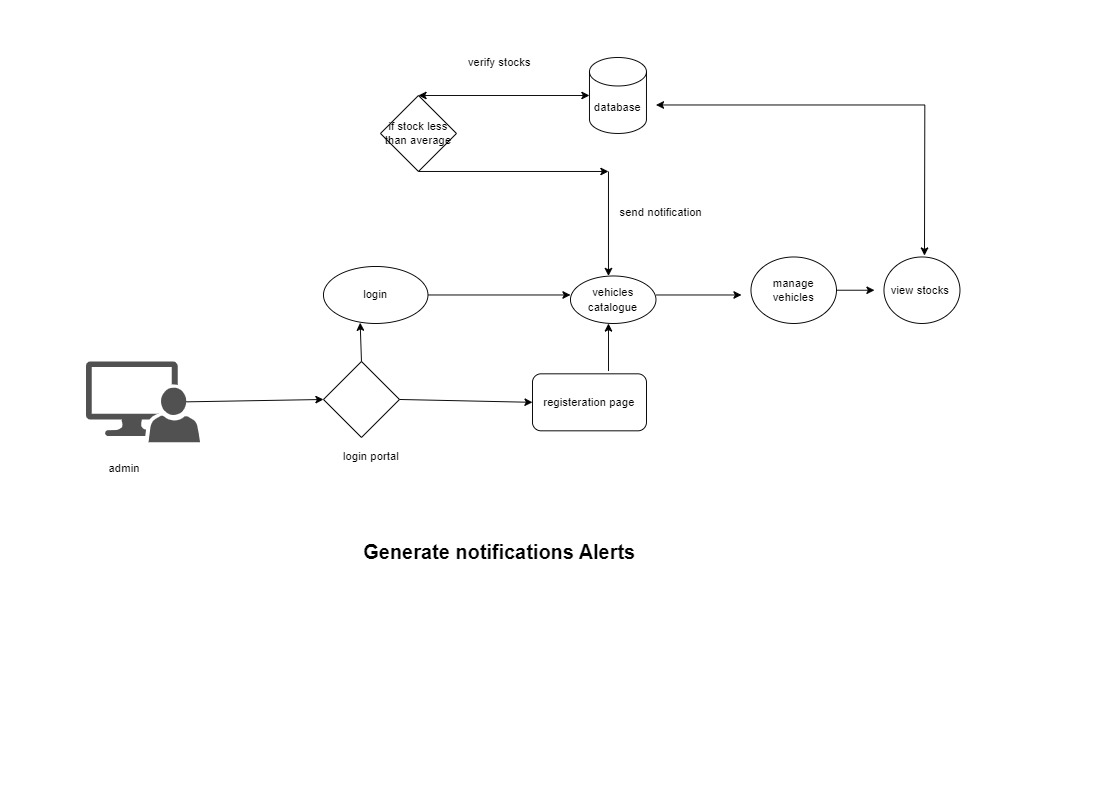


Figure 7: Generate notifications Alert

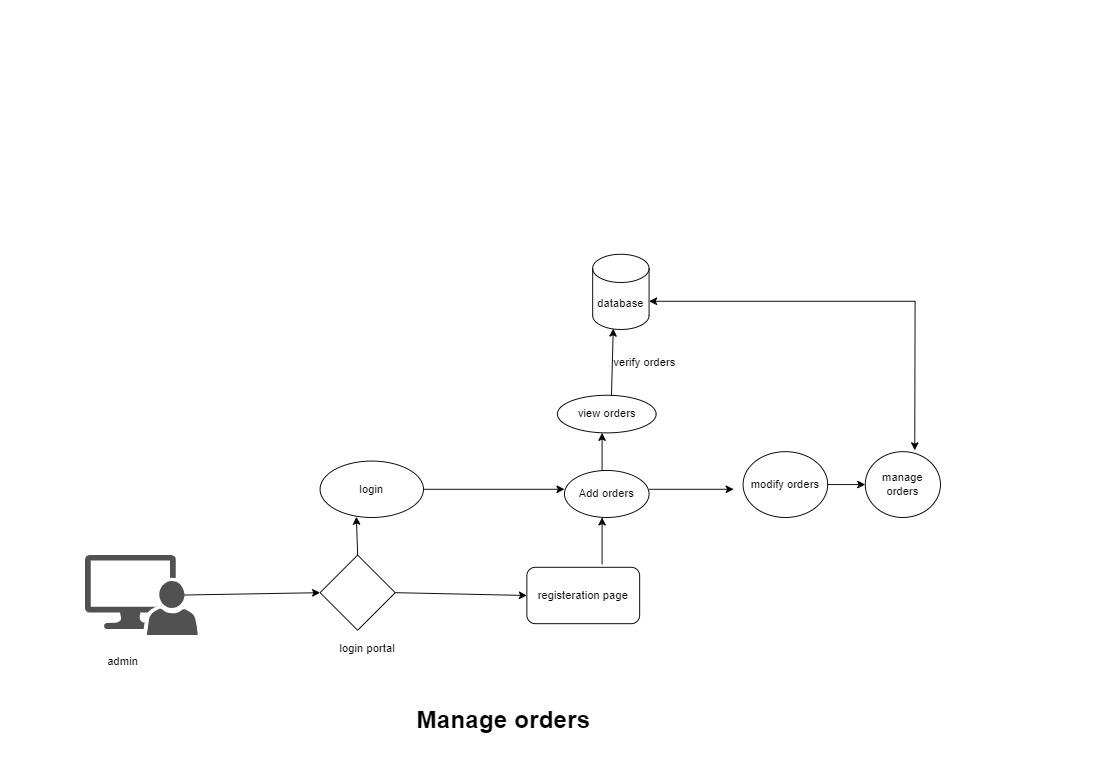


Figure 8: Manage order

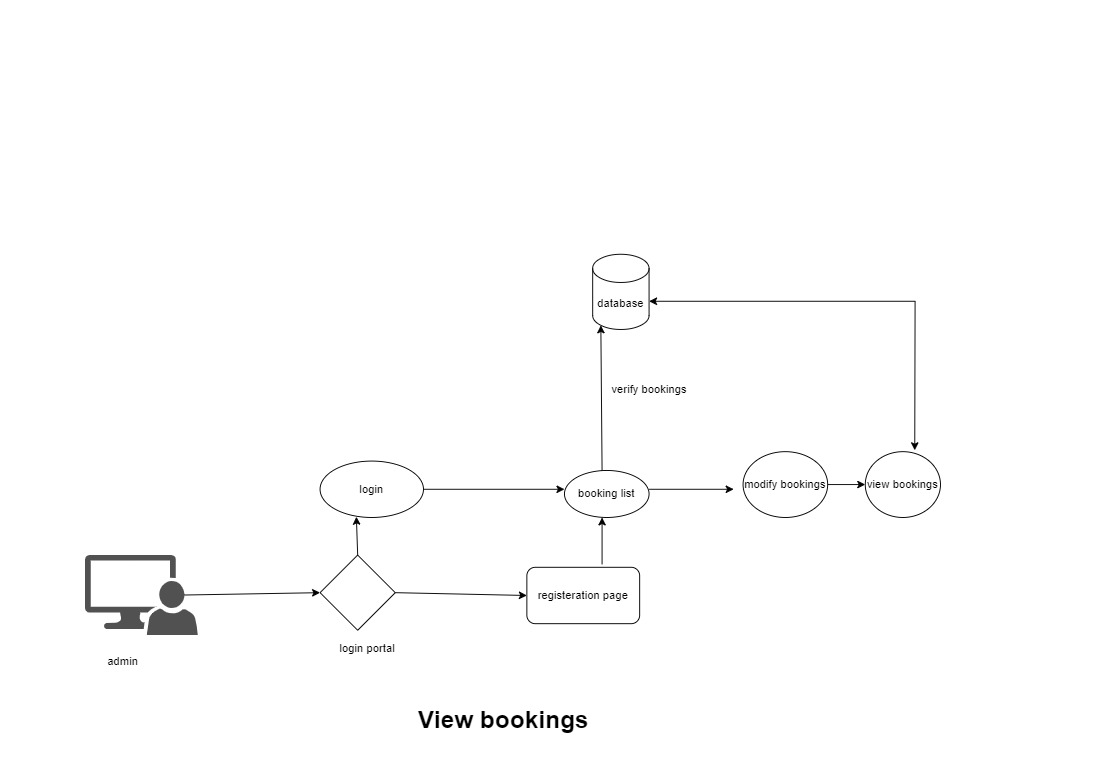


Figure 9: View bookings

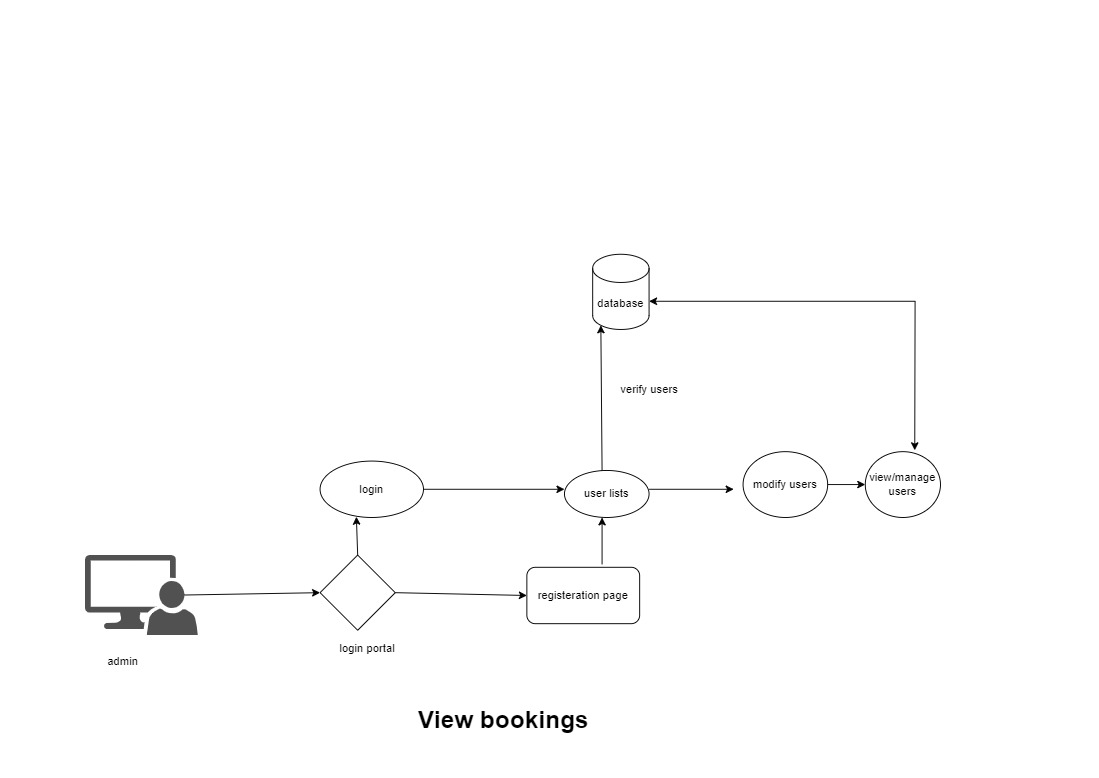


Figure 10: View bookings