TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 491 High-Level Design Report

Team Members:

- Emre Duzcu
- Ozan Muharrem Şahin
- Anıl Aydemir
- Yağız Çakır

Supervisor: Verena Adanova

Jury Members:

- Fırat Akba
- Eren Ulu

Table of Contents

1.Introduction

1.1. Purpose of the system

The CarCare+ system deals with the concept of designing a new, easy-to-use platform that will be able to manage vehicle owners' health and maintenance processes of vehicles effectively. Commonly, vehicle owners face problems in keeping track of the various maintenance activities or periodic needs of a car. Timely procedures of paying taxes, changing oil, and dates of inspection generally stress users and may involve expensive repercussions quite frequently. CarCare+ was built to overcome these challenges by making maintenance easier for vehicles.

What the system mainly tries to implement is facilitating individual car owners to have the ability to monitor the health of their cars on one mobile application. Using the application, users will be able to check their vehicle mileage and get instant critical data on tire pressure and the status of the engine. Additionally, the system will organize the vehicle maintenance history for easy access of past maintenance operations by users.

CarCare+ allows the user to track the vehicle's maintenance, along with sending reminder notifications and planning tools. For instance, the user is notified in time not to miss the upcoming date for paying taxes or even inspecting the vehicle. It will be easier for the car owner to fulfill all legal obligations in consideration of possible delays.

The system also develops an integrated store for further simplification of users' needs in maintaining their vehicles. The application-based store will enable one to view and purchase different products in general for vehicle maintenance products, like oil filters and wipers, thus enabling them to get his needs fast and in a reliable way.

CarCare+ is probable to provide much safer and cheaper driving for the owners in the long run, too. It saves answers because it enhances regular cares for the possibility to predict failures and prolong vehicles' life. CarCare+ will make the process of vehicle maintenance of the users effortless and enjoyable because of its user-friendly interface and high-performance infrastructure.

1.2. Design Goals

Usability / User Friendliness

- It should be developed in such a way that the facility of operation is smooth and instinctive for both older and younger users.
- Predictive maintenance recommendations are granular and actionable.
- The application will ensure fast and easy navigation to real-time data about the vehicle, its maintenance history, and due dates.
- An appealing design will contribute to better engagement and increase satisfaction among the end-users.

Reliability

- This will ensure the system will stand firm during peak loads and seamlessly provide access to vehicle information and functions.
- All the history of maintenance, vehicle metrics, and information regarding the users will be stored securely and recoverable.

Security / Confidentiality

- Similar to other purchase-related data, vehicle information and history of its maintenance should be encrypted to ensure confidentiality.
- Location information shall be utilized only when absolutely necessary regarding integrations of IoT or related services.
- Accounts shall be protected with two-factor authentication, along with strong password policies.

Accessibility

- The app is to be supported on Android 8.0+, iOS, with active Internet.
- Maintenance providers and vehicle product sellers may include their services on the platform, provided the platform's rules are followed.
- The design should consider the principles of accessibility in order to support users with special needs.

Scalability

- The database architecture should support horizontal scaling to meet the growth in user and event numbers.
- The application should efficiently handle concurrent users, for instance, in large-scale IoT data upload.
- AI-powered prediction maintenance algorithms will scale with more vehicles and users.

Extensibility

- Development of modular systems means new features can be added later, in the form of new IoT integrations or advanced analytics.
- The backend and UI should be in a design so that updates are at any time with the least possible downtime.

Performance / Efficiency

- It will ensure that all the main metrics of the vehicle update in less than 3 seconds through response time optimization.
- Notifications and alerts will be delivered in less than 5 seconds.
- This will handle all the algorithms of matching maintenance services and recommendations effectively to produce results in real time.

1.3. Definitions, Acronyms and Abbreviations

Definitions

Vehicle Status:

The general term for the various health and maintenance statuses regarding the vehicle. In this context, the critical information of the vehicle includes tire pressure, engine status, and fuel level.

Notifications:

Notifications of useful system maintenance or renewal dates. Such a date might be given to show when the date of a car's road tax is due or when an oil change is required.

Maintenance History:

Record of the vehicle's previous maintenance operations. All previous maintenance performed can be viewed and tracked, so the user knows the vehicle is maintained regularly.

Reminder:

Event Notification: the ability of the application to notify a user of upcoming events or activities that must not be missed-for example, when inspections on vehicles are due.

Shopping:

A buying section in the app where users can purchase vehicle upkeep products.

Acronyms

IoT (Internet of Things):

IoT basically enables the physical devices to connect with one another and the users over the internet. IoT Integration: Real-time Vehicle Data Monitoring.

API: Application Programming Interface

An interface that allows one application to exchange data with another application. External APIs are used by CarCare+ to gather data from vehicles.

GDPR - General Data Protection Regulation

The General Data Protection Regulation developed by the European Union protects users' data. Correspondingly, any processing of the user data by CarCare+ is in conformity with the latter regulation.

UI (User Interface):

Visual design elements through which the user will interact with the application. The UI design for CarCare+ is quite friendly.

Relational Database Service (RDS)

Fully managed Amazon service that provides a scalable, secure, cloud-based database solution. The data for CarCare+ has been hosted using Amazon RDS.

Abbreviations

Initials OBD-II

Onboard diagnostics A vehicle-based diagnostics system that provides information concerning vehicle performance. CarCare+ is an interface used to retrieve data from an on-board diagnostics.

UX (User Experience)

Represents the experience the user has with the application. CarCare+ does much to provide an intuitive and effective UX.

HTTP

Hypertext Transfer Protocol A standard protocol used to transmit data over the web. CarCare+, in particular, uses it for API requests and responses.

SQL (Structured Query Language)

A programming language used in handling databases. CarCare+ processes data in the database using SQL.

1.4. Overview

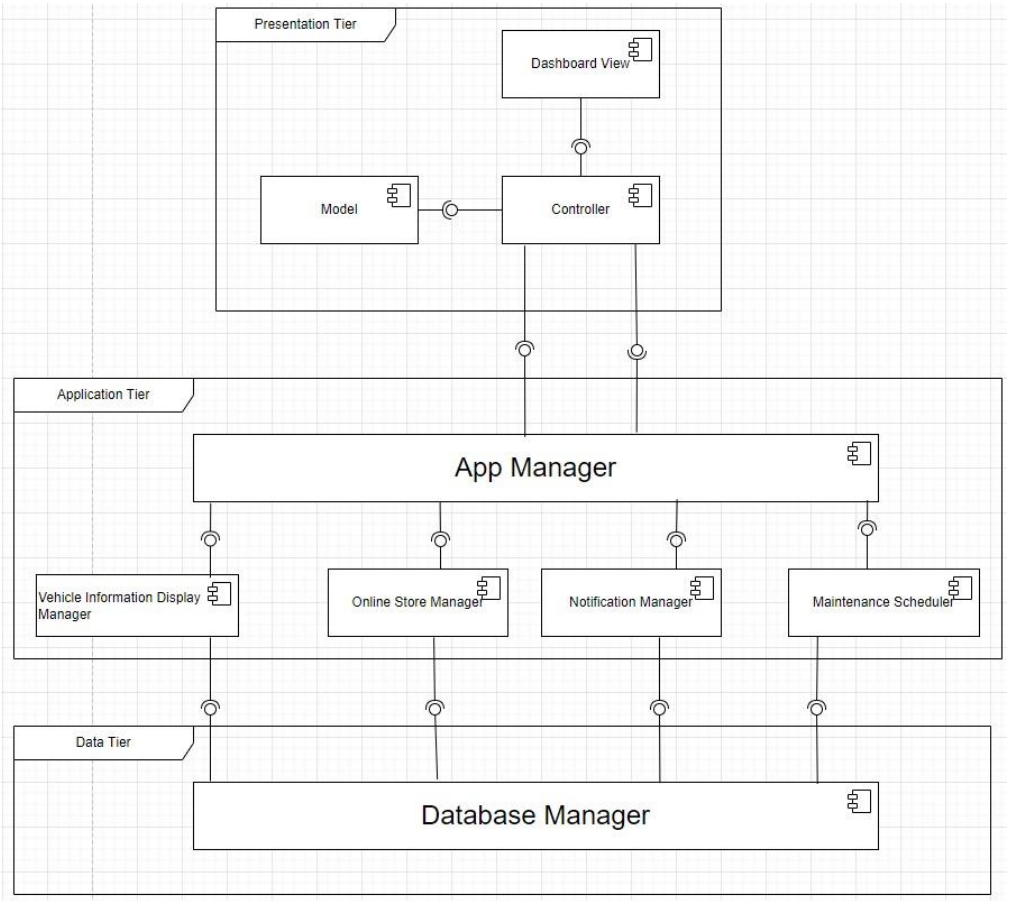
The present report is intended to give a high overview of the CarCare+ project: describing proposed software architecture, system components, main design objectives of the project. The given application will try to provide one-stop assistance for any vehicle owner so as to understand the status of the vehicle and also, in light of that, efficiently carrying out several maintenance activities. Some of the major features that need to be implemented are tracking of vehicle status, notification system, product store, and appointment scheduling. Its core architecture is easily scalable, considering security and usability concerns for the best user experience, adding new features in the very near future with ease. Finally, CarCare+ will safely manage all vehicle information more conveniently within an intuitive and robust system.

2. Proposed Software Architecture

2.1. Overview

CarCare+ app-mobile assists vehicle owners with their own action items and tasks related to vehicle health status management and maintenance needs. The software architecture design must propose a friendly user interface developed in respect of high standards concerning reliability, scalability, and performance. It is intended that the proposed software architecture be modular and flexible in its structure. This approach enables each of them to be developed, tested, and managed separately. A variety of subsystems shall be provided within the system to enable users to track in real time information about their vehicles, remember dates of vehicle maintenance, or make it easy to perform all these necessary operations. In the architecture, there will be a front-end design corresponding to expectations from mobile users and a powerful data processing and integration infrastructure in the background.

2.2. Subsystem Decomposition



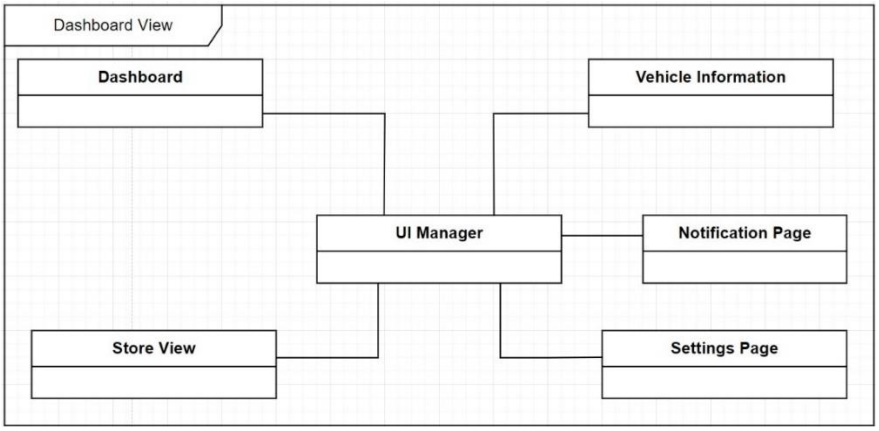
Our system is formed by a 3 Tier Decomposition. The respective tiers are presentation tier, application tier, and data tier. In this manner, tier decomposition provides flexibility in updating the project because tiers are independent of each other.

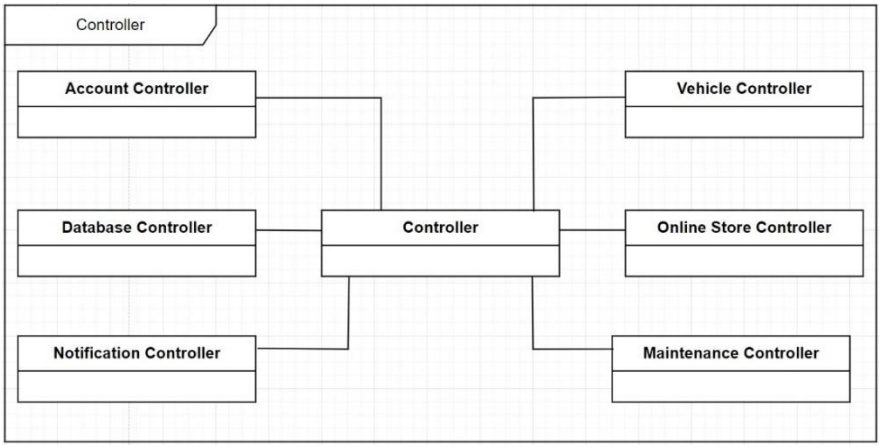
The Presentation Tier contains the different User Interfaces. It contains Dashboard View, Controller, and Model modules. It is in this tier that the user will interact with the system and all the data presented to him. Each one of these different modules has several classes and

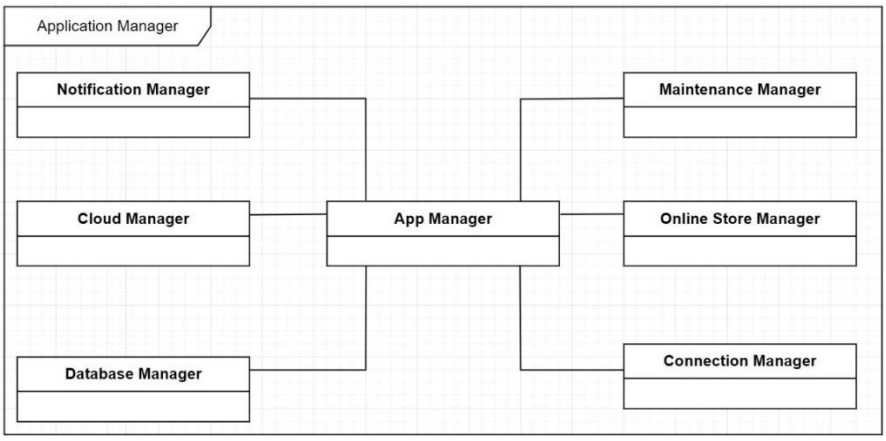
responsibilities handling user input, showing vehicle data, and also communicating with the application layer.

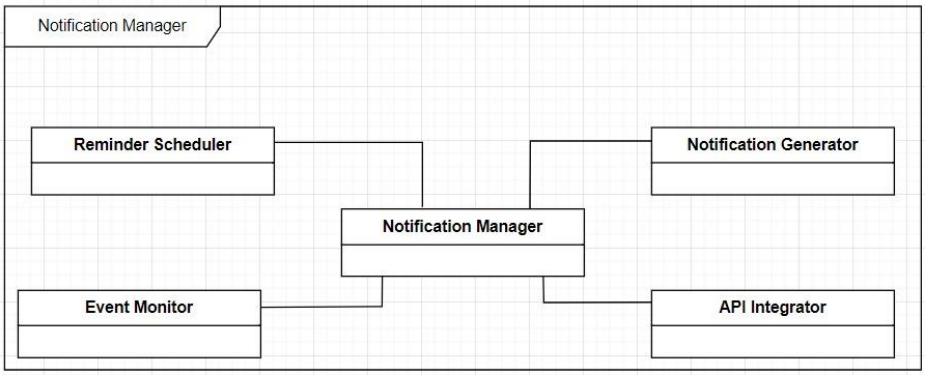
The main logics will be provided from the application tier, such as an app manager, notification manager, Vehicle Information Display Manager, Online Store Manager and maintenance scheduler. They ensure that the user will be notified about certain events, such as changing oil, paying taxes, and checking up on the car.

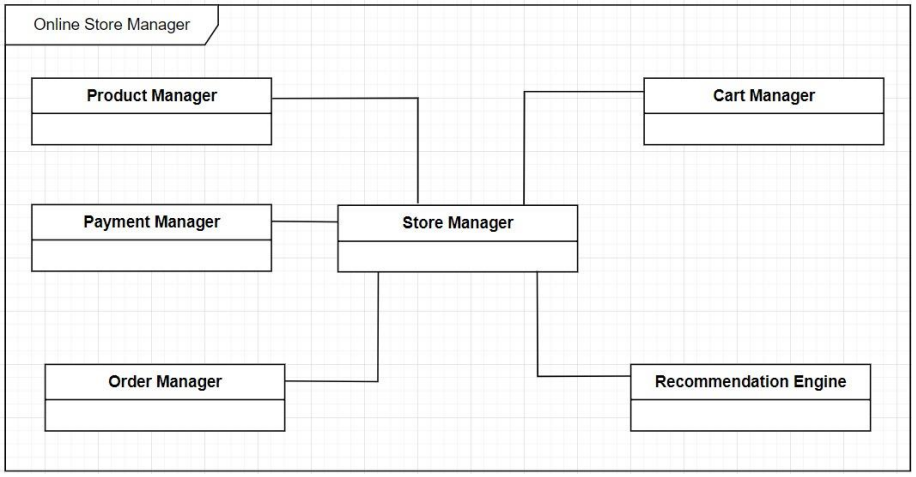
Data tier: The level at which, through the database manager, all data about the users and the vehicles are controlled. Ensuring that whatever information concerning storage, access, and retrieval is securely done to make the transactions successful. The specifics of its components, including dashboard view, controller, and application manager, are provided in the following diagrams:



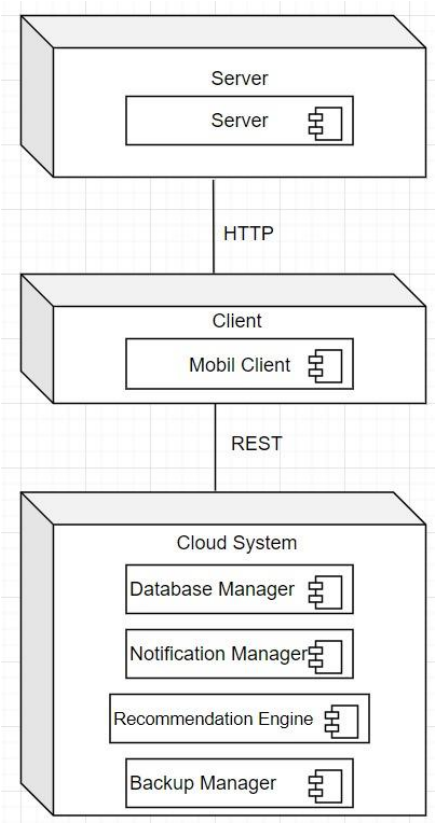








2.3. Hardware/Software Mapping



The tiers of the CarCare+ system are separated into server, client, and cloud system as shown in the diagram. This architecture ensures scalability, efficient data processing, and a seamless user experience.

The server tier represents the back-end part of the whole system; it is in charge of core functionalities, such as managing API requests and performing all the background models whenever required. This also facilitates communication between the client and cloud system in a request/reply model over HTTP via TCP. All the data interchange is made with the server acting as a bridge that ensures a request from a client will be processed safely and forwarded to any of the cloud components if needed.

The client tier is the mobile application which can be used in Android platforms through the creation of the application in React Native. It plays a role of the user view layer through which the user is able to have his or her interaction with the application. This layer is for presenting the vehicle data, misc. notifications and to enter the store. The mobile client uses HTTP protocol to interact with the server and data are shared using RESTful APIs.

The cloud system tier includes the following components in order to support data storage, notifications, recommendations, and backups:

Database Manager: This particular module is aimed at keeping users' information, vehicle specifications, and the maintenance records safely. Such cloud-based relational databases as Amazon RDS make it retention of large data sets to be more efficient.

Notification Manager: This module also sends alerts for maintenance time, tax to be paid and other announcements in regard to the status of the vehicle.

Recommendation Engine: The application analyzes data coming from the car and from the interaction with the user's and provides recommendations. For instance, it might suggest buying some of the vehicle products or booking necessary service.

Backup Manager: This module ensures that critical user and system data gets backed up regularly, which minimizes the risk of data loss and allows for restoration in case of problems. It offers its service with communication enabled by protocols of secure HTTP and RESTful API. The client can fetch and send data only to the server, and then the server communicates further with the cloud system to achieve the desired request. Thus, the three-tier architecture allows the CarCare+ app to be quite effective for offline and online environments, providing a better solution to manage vehicles reliably.

2.4. Persistent Data Management

User information is stored permanently within the CarCare+ application:

Local Data Storage

SQLite/Room Database (for Android):

General data will be stored in a local database within the device itself: vehicle information, mileage, tire pressure, user preferences, and maintenance history among others.

Advantages:

- It enables offline access.
- This would increase privacy in that user information would remain on the device.

Cloud-Based Data Storage (Optional):

- If server-side integration is required, this could also be performed using Amazon RDS MySQL or Firebase Realtime Database.
- Cloud Storage Purposes: To let the users use all their devices to access the same data. In data backup so that the data should not be lost.

Data Management Policies

- Processing of data in CarCare+ is based on the following policies.
- Information is kept secure by using encryption both on the device itself and in the cloud. This can be achieved using an encryption standard such as AES-256.

Backup:

- The application automatically backs up the data on your device.
- It also provides an optional backup of usual data through syncing it to the cloud system, like Firebase.

Data Update and Deletion:

Users can also edit the vehicle information, account information, and delete the same if user want to. It gives a functionality that allows the application to be compliant with data protection regulations, such as the General Data Protection Regulation.

Data Types

The main types of data stored in the CarCare+ application are:

User Data:

- Basic account information such as username, email, password.
- User preferences (notification settings, language selection, etc.).

Vehicle Data:

- Vehicle model, mileage, last maintenance date, and maintenance history.
- Vehicle health parameters such as tire pressure and engine condition.

Maintenance and Notification Data:

- Scheduled maintenance dates completed maintenance history.
- Notifications and reminders sent to the user.

Store Data (Optional):

- User’s order history and shopping cart information.

Database capability

Database systems perform the following functions:

Insert Data:

- The system will be updated regarding this new user, vehicle, and maintenance information.

Update Data:

- This will automatically update whenever a user edits the information for a given vehicle.

Query Data:

- Allows the user to query the status information of a vehicle, its maintenance history, or shop data.

Delete Data

- Removed after closure of the account or when the owner changes the particulars of a particular vehicle.

2.5. Access Control and Security

The CarCare+ system is designed to ensure that sensitive user and vehicle data is secure and protected. Our aim is to create a reliable environment for our users. Some of the key aspects are:

User Authentication:

- To prevent someone to join another user's account we will use a two-factor authentication. This method ensures that only the authorized user can reach that account by requiring a secondary verification method.

Authorization Levels:

- Users shouldn't be able to reach every piece of information. A regular user should only be able to see their user and vehicle information. An admin might need to see more information and should be able to manage the store's products. Different access levels allow this to happen.

Data Encryption:

- CareCare+ provides data privacy for their users. We will use encryption to prevent someone intercepting the data being sent from your phone and CarCare+ servers. All the information between the client, server and cloud will be encrypted.

Password Policies:

- There are some rules to make sure that our users have strong passwords. The passwords have a minimum length, and they are enforced to have character variety.

2.6 Global Software Control

The CarCare+ system needs an efficient architecture in order to guarantee coherency, reliability, and maintainability.

Centralized Database Management System:

- All data could be stored in one central database. User profiles, all records of the car-such as maintenance-and items the store has in stock can be managed with ease to avoid misunderstandings over updated material.

API (Application Programming Interface):

- The mobile app will be communicating with a centralized database with the API. In this respect, the program will be in proper order, and it is going to be easier to perform upgrades.

Server Infrastructure:

- We will be requiring some reliable and powerful servers so we can host the database as well as the API on them. The servers constantly need to be looked upon to ensure they keep running all the time and there is no downtime in them.

Version Control:

- In doing updates on CarCare+, we are going to make sure we keep track of all our changes so that when there is a problem, we can always revert to our last version.

2.7 Boundary Conditions

There should be limits to what a system can handle. We will set this limits to make sure our system doesn’t crash.

Number of Users:

- If too many people use the app at the same time, the servers can be overloaded, which can crash the system. Therefore, in order to keep the server stable, a certain number of users will be allowed to log in.

Data Storage Capacity:

- As more users join our application we will need to hold more data. We need to consider how much data our system can store and make plans for expending it in the future.

Network Connectivity:

- The app uses internet connection. If the user doesn’t have a stable internet connection or even no internet connection, the app should display an error. Storing some datas locally also can be an option.

Device Compatibility:

- CareCare+ is a mobile app so we need to make sure it works fine in different types of phones. We will test it on different devices to make sure it works for a variety of devices.

3. Sub-System Services

--Presentation Tier--

Dashboard View

- **Services:**
    - Provide the rendering of UI components to display Vehicle Details, Notifications, and Maintenance Schedule.
    - Display Information of Vehicle: mileage, fuel level, engine temperature, tire pressure, oil level.
    - Navigate to Store, Map, Notifications, Settings, etc.
    - Quick access to Maintenance Schedule, Traffic Fine Enquiries, and Tax Enquiries.

Model

- **Services:**
    - Defines the data structure of Vehicle Information and User Preferences.
    - Scrubs and prepares the data before sending it to the Controller.
    - Synchronize data between UI components and Application Tier.

Controller

- **Account Controller**
    - Authenticate user credentials at the time of login.
    - Maintain updates of user profile and application preferences.
- **Notification Controller**
    - Maintain reminders about maintenance, insurance renewals, and tax payments.
    - The final one is the push notification and alerts to the user interface.
- **Vehicle Controller**
    - Retrieve and present vehicle-specific information such as mileage, fuel level, and tire pressure.
    - Gather real-time data on vehicle maintenance and status updates.
- **Online Store Controller**
    - Interact with the product interface of the store, for instance by ordering a product and placing it in the cart.

    - Allow to recommend goods and set filters by car accessories and spare parts.

- **Maintenance Controller**
    - Maintain and share schedules and records of auto maintenance.
    - Synchronize communication of the planned activities concerning maintenance.

--Application Tier--

App Manager

- **Services:**
    - Orchestrate the interactions among all the components of the application.
    - Act as a common interface between the Presentation and Data tiers.
    - Maintain system-wide settings such as user preferences and dark mode.
    - Vehicle Information Manager.

Vehicle Information Display Manager

- **Services:**
    - Retrieve and format the details of the vehicle: mileage, fuel level, tire pressure, and engine temperature.
    - Communicate with both the Vehicle Controller and the Database Manager to obtain and replicate vehicles information.
    - Be able to supply instantaneous information regarding the state of a vehicle.

Online Store Manager

- **Product Manager**
    - Ensure the proper record and updating of motor vehicle supplies (lubricants, parts, accessories etc.) for sale.
    - Track product availability and other information related to stock.
- **Cart Manager**
    - Control the interaction with the shopping cart, to be able to add / cancel items.
    - Total each amount to an order or purchase that may include any discounts.
- **Payment Manager**
    - Able to safely make payments for store products that customers want to pay for.
    - This can be done by linking with other external payment processors to facilitate the transactions.
- **Order Manager**
    - Track user order history and statuses. Provide functionality for cancellations and refunds. Recommendation Engine

- **Recommendation Engine**
    - Suggest related products regarding user purchase history and data of the vehicle.
    - Analysis of user interactions to better recommendations.

Notification Manager

- **Reminder Scheduler**
    - Schedule alerts for maintenance and tax/insurance due dates.
    - Warn users about approaching deadlines through their vehicle data.
- **Event Monitor**
    - Tracking user activities as well as vehicle conditions in order to trigger an appropriate notification.
    - External factors like weather or location-based events are monitored for awareness in context.
- **Notification Generator**
    - Formatting of notification messages to be sent for maintenance, renewal of insurance, and state of fuel.
    - Send notifications as in-app notifications (in-app notifications).

Maintenance Manager

- **Services:**
    - Tracking and listing of vehicle maintenance history, tasks performed
    - Generation of maintenance schedule by mileage and condition
    - Reminders to users about approaching deadlines.

Map Manager

- Services:
    - Show user current position on map
    - Gas stations and car washes in vicinity, using location
    - User searching for particular places or services, like repair shops
    - Navigation to selected places using third-party API, if necessary.
    - Allow the user to toggle on/off map layers, such as displaying gas stations or car washes.

Settings Manager

- Services:
    - Allow editing of profile information: profile pictures and personal information.
    - Manage vehicle appointment schedules and insurance dates.
    - Provide enable/disable options for dark mode.
    - Display an FAQ section with answers to common questions about the app.
    - Functionality for rating the app, with user feedback.
    - Allow secure logout and account deletion options.

- Keep system-wide settings. Ensure user preferences are saved and applied across the board.

--Data Tier--

Database Manager

- **Services:**
  - Provide CRUD (Create, Read, Update, Delete) operations for all subsystems.
  - Ensure data consistency and backup.
  - Optimize query performance for the application.

4.Glossary

Acronyms & Abbreviations

- **API:** Application Programming Interface - allows different applications to share data.
- **GDPR:** General Data Protection Regulation - General Data Protection Regulation—EU regulation that ensures data privacy and security.
- **IoT:** Internet of Things -the connection of physical devices to the internet for real-time data exchange.
- **OBD-II:** On-Board Diagnostics - on-Board Diagnostics—a system within a vehicle that provides data on its performance.
- **RDS (Relational Database Service):** Managed cloud database service; used here for scalability and secure management of data.
- **SQL (Structured Query Language):** Programming language designed for the management and querying of databases.
- **UI (User Interface):** Visual interface with which users interact in an application.
- **UX (User Experience):** Overall experience of a user with an application, focused on an intuitive design.

Key Concepts

- **Dashboard View:** Major UI displaying vehicle information, notifications, and links to other functionality such as tax payment, appointment and traffic ticket inquiry.
- **Notification System:** Notify and alert system about an upcoming event regarding a vehicle-inspection, oil change.
- **Recommendation Engine:** Recommend sales of products based on user and vehicle data.
- **Reminder Scheduler:** Schedule and remind the user about vehicle-related tasks and deadlines.
- **Vehicle Controller:** Responsible for handling and representing in real time vehicle telemetry: tire pressure, engine health.

Subsystems & Modules

- **App Manager:** High level error handling and central component managing system interactions.
- **Cart Manager:** It handles total calculations and discounts for shopping cart functions.

- **Notification Manager:** It coordinates the delivery of notification.
- **Online Store Manager:** Helps with vehicle maintenance product purchases in the app.
- **Vehicle Information Display Manager:** Display Manager: It collects and formats vehicle data for the dashboard.

Technologies & Infrastructure

- **Cloud-Based Data Storage:** Provides a scalable way to store user and vehicle data for access and backups remotely.
- **Encryption (AES-256):** The measure of security for the protection of user and system data while in storage and transport.
- **RESTful API:** Used as a medium of communication between the mobile app, server, and the cloud.
- **Three-Tier Architecture:** The idea of System design – Operations as module for presentation (UI), application (logic), and data (storage).

5.References

Figure 3: Subsystem decomposition, shown using a UML component diagram. **(n.d.). ResearchGate. https://www.researchgate.net/figure/Subsystem-decomposition-shown-using-a-UML-component-diagram_fig1_311634500**

Sam Saber. (2024, October 3). What is a Subsystem? Reqi. https://reqi.io/articles/what-is-a-subsystem