

**CNG 495 – Cloud Computing**

**Term Project Proposal**

**Hasan Eren Yarar 2457802**

**Ata Kaleli 2385474**

**Serhan Turgul 2453603**

**STAGE 1 REPORT**

**Repair-Shop Car Management System:**

**Overview:**

This system is an all-in-one solution designed to simplify the management of vehicle maintenance and repair tasks for both vehicle owners and repair shop owners. It acts as a strong platform that links vehicle owners, service providers, and administrative staff. The system is specially made to improve the process of organizing, managing, and monitoring vehicle repair and maintenance work.

**Key Features and Functionalities*:***

User (Vehicle Owners):

**Registration & Login:** Users are securely register and log in to the system, ensuring authenticated access to the their vehicle data.

**Maintenance Record View:** Users have access to their past maintenance information.

**User Vehicle Management:** Users can add new vehicles to their profile, including details like model, miles, type, and year.

**Appointment Scheduling:** Users can easily schedule appointments with repair shops, making it more convenient to manage repair and maintenance activities for their vehicles.

Repair Shop Management:

**Access to User Information**: Repair shops can view information about their customers, including contact details and vehicle data.

**Take Maintenance Record:** Shops can add new maintenance records for the services they provide, with detail tracking of all maintenance activities.

**Appointment Management**: Viewing and managing upcoming appointments.

Administrative Functions:

**Repair Shop Statistics**: Administrators can view total income for each Repair Shop.

**Repair Shop Registration**: New repair shops can be added to the system.

**Technologies:**

• Frontend: Python Tkinter

• Backend Python 3.9

• Database: SQLITE3

• Cloud Services: Google Cloud Platform

**Google Cloud Platform as an Infastructure as a Service (IaaS):**

We are utilising Google Cloud Platform as an Infastructure as a Service (IaaS), by using their Compute Engine to run a Virtual Machine(VM). We run Ubuntu 20.04 on the VM, to run our server script and database.

What we are doing is a Software as a Service (SaaS). We have a server script that runs on Google Cloud Platform which listens to incoming details from each client, executes queires depending on what is provided by the client, and then sends the results of the query back to the client. We consider this as SaaS because the client is a piece of software that communicates with the cloud. Any number of repair shops can utilize this software to make their business with their clients easier.

**Data Types**

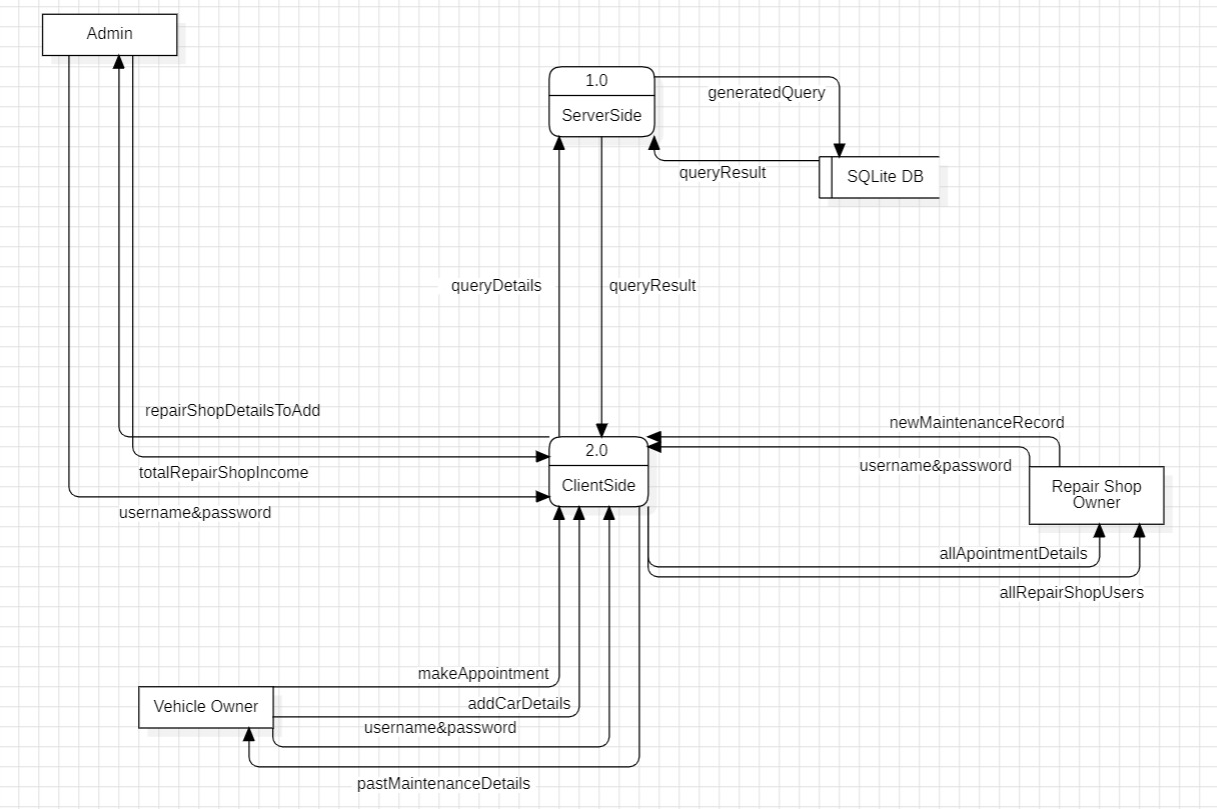
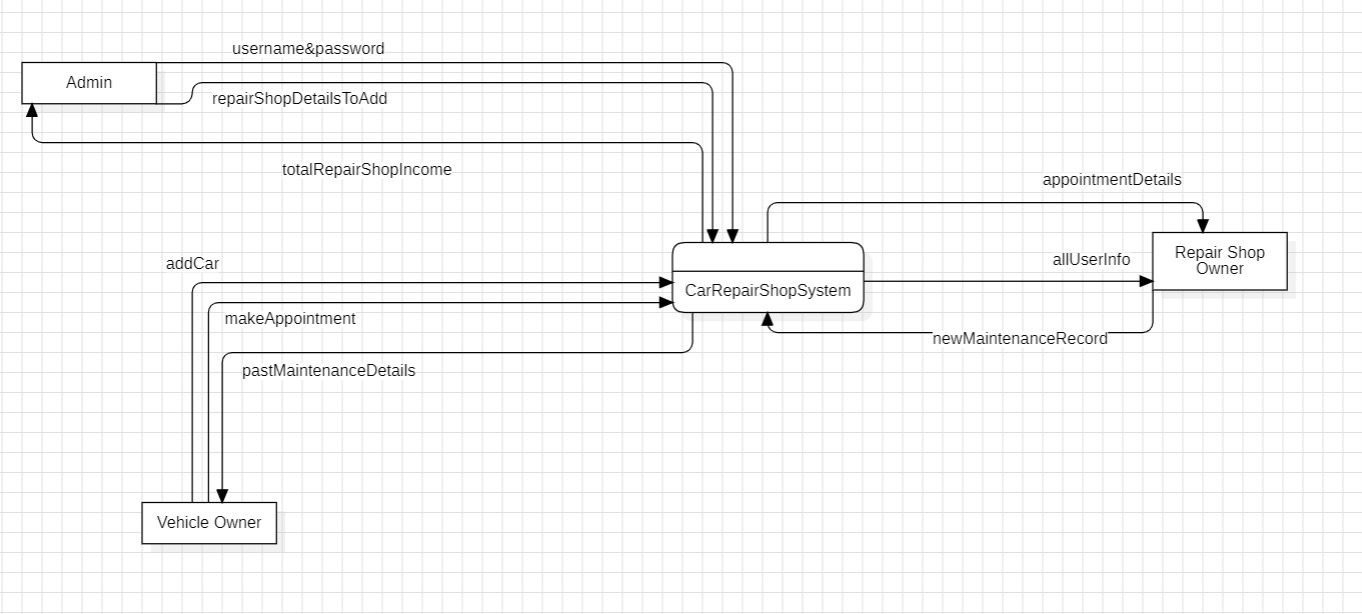
**Text Data:**

* Maintanance descriptions and details.
* User information such as email name address
* Admin inormation such as email pasword name
* Repair Shop information such as email password

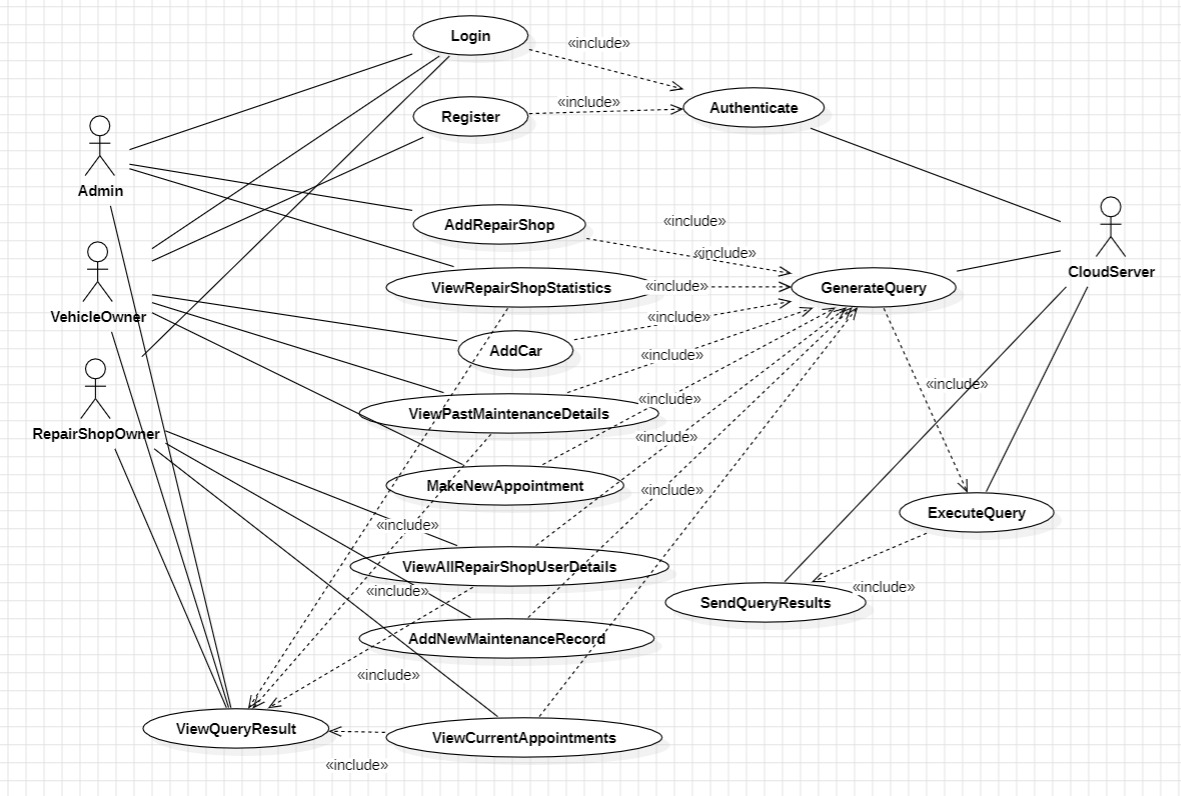
**Numerical Data:**

* Miles of car
* Price of the maintanance

**Data Flow Diagram of The System:**



**Client & Service Interactions Diagram:**



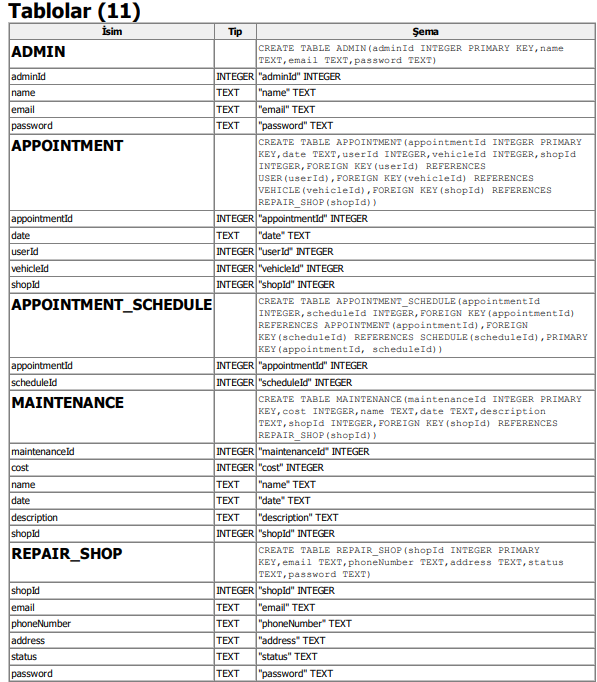
***• State the expected contribution for each project group member:***

All team members contributed to both backend and frontend coding. Specifically, Hasan Eren Yarar focused on designing and coding the database, Ata Kaleli handled the graphical user interface and server connection, and Serhan Turgul specialized in integrating the program with Google Cloud Services.

**STAGE 2 REPORT**

**PART 1 & 2**

**First of all, Eren dealed with database of our system with sqlite3 in pyhton. Below are the screenshot of our system’s database.**



A screenshot of a document

Description automatically generated

**Then he code the necessary queries for our system.**

**Ata dealed with client side of our system in Python. Client side has a connection with server side. In client side, we have several GUIs. They are as follows:**

A screenshot of a computer screen

Description automatically generated

Şekil 1:Login GUI

A screenshot of a computer error

Description automatically generated

Şekil 2: Invalid login

A screenshot of a computer screen

Description automatically generated

Şekil 3: User(Car Owner) Menu

A screenshot of a message

Description automatically generated

Şekil 4: User Menu - Show my Past Transactions (To be implemented)

A screenshot of a computer

Description automatically generated

Şekil 5: User Menu- Make Appointment(To be implemented)

A screenshot of a computer

Description automatically generated

Şekil 6: User Menu- Add New Car (to be implemented)

A screenshot of a message

Description automatically generated

Şekil 7: User Menu- Show Car Status (to be implemented)

A screenshot of a computer

Description automatically generated

Şekil 8: Repairshop Menu

A screenshot of a message

Description automatically generated

Şekil 9: Repairshop Menu- Show all Car Owners (to be completed)

A screenshot of a message

Description automatically generated

Şekil 10: Repairshop Menu - Show All Appointment (to be implemented)

A screenshot of a computer

Description automatically generated

Şekil 11: Add new maintanance (to be implemented)

A screenshot of a computer

Description automatically generated

Şekil 12: Update Maintanance(to be implemented)

A screenshot of a computer program

Description automatically generated

Şekil 13: System Admin Menu

A screenshot of a computer screen

Description automatically generated

Şekil 14: Show Repairshow Anaysis

A screenshot of a computer

Description automatically generated

Şekil 15: Add new repairshop

A screenshot of a computer error

Description automatically generated

Şekil 16: It is added successfully

A screenshot of a computer

Description automatically generated

Şekil 17: Server waits for connections

A screenshot of a computer

Description automatically generated

Şekil 18: login query coming from client

A screenshot of a computer

Description automatically generated

Şekil 19: show repairshop analysis query coming from client

A screenshot of a computer

Description automatically generated

Şekil 20: add repairshop query coming from client

PART3

Google Cloud Platform

Google Cloud Platform (GCP) is a suite of cloud computing services provided by Google. It offers a variety of services, including computing power, storage, databases, machine learning, and networking, among others. GCP allows businesses and developers to build, deploy, and scale applications and services on Google's infrastructure.

**For this Cloud technology, we watched a video on youtube, which is the following link:**

<https://www.youtube.com/watch?v=RFPlXmgKCtk&ab_channel=AdamTMedia>

Virtual Machine

A virtual machine (VM) is a software-based emulation of a physical computer. It runs an operating system and applications just like a physical machine but is hosted on a physical computer known as a host. Virtual machines are commonly used for server consolidation, testing environments, and running multiple operating systems on a single physical machine. They provide flexibility, resource isolation, and ease of management in computing environments.

**We created a virtual machine in Google Cloud Platform, and we run our server.py in this machine on Ubuntu. Moreover, to connect virtual machine outside, we got the external IP from Google Cloud platform.**

PART 4:

**Explations and difficulties we achieved:**

For database, in server side, we got an error regarding the table creations and inserting the data. To solve this problem, Eren found a solution which implements operating system module.

For client side, we got an error to connect with server. Later, Ata changed the socket implementation, so the problem is solved.

For server, we really stuck several hours to connect the server with Cloud. Then Serhan watches a youtube tutorial regarding the connection of our server code with Cloud platform we used, then we successfully managed the overall connection.

**Part 5**

1. Complete the not completed parts of GUI, such as RepairShop functionalities (ATA)
2. Add new GUIs to our system (ATA)
3. Better optimization for our database, especially “Scheduling” (EREN)
4. Create new queries and databases for GUIs that will be added (EREN)
5. Adding new functionalities for queries coming from client on server side. (Serhan)
6. Implementing a notification system that is connected with Cloud. For example, Cloud system will send a notification message to Car owner whose car is almost repaired. (SERHAN)

WEEK1:

Ata planning to complete first part of his task (option1)

WEEK2

EREN planning to complete first part of his task (option3)

SERHAN planning to complete first part of his task (option5)

WEEK3

EREN planning to complete second part of his task (option4)

SERHAN planning to complete second part of his task (option6)

Final WEEK

ATA planning to complete second part of his task (option2)

Finally, we will delivery the code for client and server. Also we will provide a backup of our database.

References:

<https://www.youtube.com/watch?v=RFPlXmgKCtk&ab_channel=AdamTMedia>

<https://www.vmware.com/topics/glossary/content/virtual-machine.html>

<https://cloud.google.com/gcp/?utm_source=google&utm_medium=cpc&utm_campaign=emea-emea-all-en-dr-bkws-all-all-trial-e-gcp-1011340&utm_content=text-ad-none-any-DEV_c-CRE_500228034474-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+~+GCP+~+General%23v22-KWID_43700071659798940-kwd-281541840951-userloc_9070307&utm_term=KW_cloud.google-NET_g-PLAC_&&gad_source=1&gclid=Cj0KCQiA7OqrBhD9ARIsAK3UXh2-qcaWIfdBAq8v6qBmiB4wKv5yFGp144Mp8SC6cJtahMCchEhGF1AaAoNKEALw_wcB&gclsrc=aw.ds>

GITHUB LINK

<https://github.com/SerhanTRGL/CNG495.git>



**CNG 495 – Cloud Computing**

**Term Project Proposal**

**Repair-Shop Car Management System**

**Hasan Eren Yarar 2457802**

**Ata Kaleli 2385474**

**Serhan Turgul 2453603**

**Overview**

This system is an all-in-one solution designed to simplify the management of vehicle maintenance and repair tasks for both vehicle owners and repair shop owners. It acts as a strong platform that links vehicle owners, service providers, and administrative staff. The system is specially made to improve the process of organizing, managing, and monitoring vehicle repair and maintenance work.

**Benefits**

* Strong platform that links vehicle owners, service providers, and administrative staff
* Improve the process of organizing, managing, and monitoring vehicle repair and maintenance work by holding the data in the system database.
* Ease the process of keaving and bringing car as a car owner, and ease the process of maintaining those records for repairshops

**Novelties and novel idea of our Project**

In our world, as a car owner, when you have some issues with your car, such as tire problem or oil change etc. , you need to leave your car to maintanance services. The main problem is that you can not assume how much time does the maintanance will takes . So our system eases the process of this big problem.

**Same Projects similar with ours**

**AutoShop Manager:** Comprehensive management software for auto repair shops, handling tasks from appointments to inventory.

**GaragePro**: A professional tool aiding garages in streamlined operations, from customer management to service tracking.

**CarCare** Pro: User-friendly software designed to simplify car maintenance, ensuring efficient service and customer satisfaction.

**Github link for the first system**

<https://github.com/alexiusacademia/autoshop-management-system>

**STRUCTURE OF OUR PROJECT**

**Key Features and Functionalities*:***

User (Vehicle Owners):

**Registration & Login:** Users are securely register and log in to the system, ensuring authenticated access to the their vehicle data.

**Maintenance Record View:** Users have access to their past maintenance information.

**User Vehicle Management:** Users can add new vehicles to their profile, including details like model, miles, type, and year.

**Appointment Scheduling:** Users can easily schedule appointments with repair shops, making it more convenient to manage repair and maintenance activities for their vehicles.

Repair Shop Management:

**Access to User Information**: Repair shops can view information about their customers, including contact details and vehicle data.

**Take Maintenance Record:** Shops can add new maintenance records for the services they provide, with detail tracking of all maintenance activities.

**Appointment Management**: Viewing and managing upcoming appointments.

Administrative Functions:

**Repair Shop Statistics**: Administrators can view total income for each Repair Shop.

**Repair Shop Registration**: New repair shops can be added to the system.

**Technologies we used**

• Frontend: Python Tkinter

• Backend Python 3.9

• Database: SQLITE3

• Cloud Services: Google Cloud Platform

**Google Cloud Platform as an Infastructure as a Service (IaaS):**

We are utilising Google Cloud Platform as an Infastructure as a Service (IaaS), by using their Compute Engine to run a Virtual Machine(VM). We run Ubuntu 20.04 on the VM, to run our server script and database.

What we are doing is a Software as a Service (SaaS). We have a server script that runs on Google Cloud Platform which listens to incoming details from each client, executes queires depending on what is provided by the client, and then sends the results of the query back to the client. We consider this as SaaS because the client is a piece of software that communicates with the cloud. Any number of repair shops can utilize this software to make their business with their clients easier.

**Data Types**

**Text Data:**

* Maintanance descriptions and details.
* User information such as email name address
* Admin inormation such as email pasword name
* Repair Shop information such as email password

**Numerical Data:**

* Miles of car
* Price of the maintanance

**Data Flow Diagram of The System:**

A diagram of a computer

Description automatically generated

**A diagram of a software system

Description automatically generated with medium confidence**

**Client & Service Interactions Diagram:**

**A diagram of a network

Description automatically generated**

**User Manuel**

**Home Page**

A screenshot of a computer

Description automatically generated

This is our system’s home page. If you dont have an account, you need to create first by clicking the register button.

A screenshot of a computer

Description automatically generated

If you have already created your account, you need to login to our system by clicking the login button

A screenshot of a computer

Description automatically generated

**Registration Page**

A screenshot of a computer login

Description automatically generated

This is registration page. You need to enter your name, email, password and phoneNumber when you are trying to register. Please remember that you need to enter an email which is not existing in our database. After you enter your information, you need to click the register button

A screenshot of a computer screen

Description automatically generated

If your registration is success, you will be directed to login page. This means that your registration is succesfully done! Else, you will get an error message that your registration is invalid!

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer login

Description automatically generated

This is our login page. You need to enter your email and password, and select your role. Then you need to click the login. Please remember that you need to enter a valid information, otherwise your login will be unsuccessful!

A screenshot of a computer screen

Description automatically generated

USER (CAR OWNER) SIDE

This is the screen after you login successfully

A screenshot of a computer

Description automatically generated

**In this menu, you can do:**

1. Show your past transactions(car maintanances)
2. Add new appointment
3. Add new car
4. Show my cars

1)

A screenshot of a computer

Description automatically generated

This screen will occur when you click the show button

2)

A screenshot of a computer

Description automatically generated

This menu will occur when you click add button

To add a new appointment, you need to enter your carId, repairshopId and appointmentDate.

A screenshot of a computer

Description automatically generated

After entering these values, you need to click add button. When your appointment is added, you will be modified by an email thanks to Google mail api.

A screenshot of a computer

Description automatically generated

Şekil 21:server success message

A screenshot of a computer

Description automatically generated

Şekil 22: mail succeess message

3) To add new car, you need to enter the details of your car’s model,miles,type and year. After you enter these informations, you will be get successful message if your car is added to the system

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

4) When you click show my cars, you can see the cars that are registered to the system. If you dont have any car, you will get a message saying that you dont have any car in the system.

A screenshot of a computer

Description automatically generated

REPAIR SHOP SIDE

A screenshot of a computer

Description automatically generated

This is the menu that will shown to you. You can do the followings from this menu:

1. Show all car owners
2. Show all appointments
3. Add new maintanance
4. Update maintanance

1)When you click show, you can see the car owners that will register to you

A screenshot of a computer

Description automatically generated

2)When you click show, you can see all the appointments done by car owners to you

A screenshot of a document

Description automatically generated

3)when you click add to add a new maintanace, you will see this screen

A screenshot of a computer

Description automatically generated

After entering