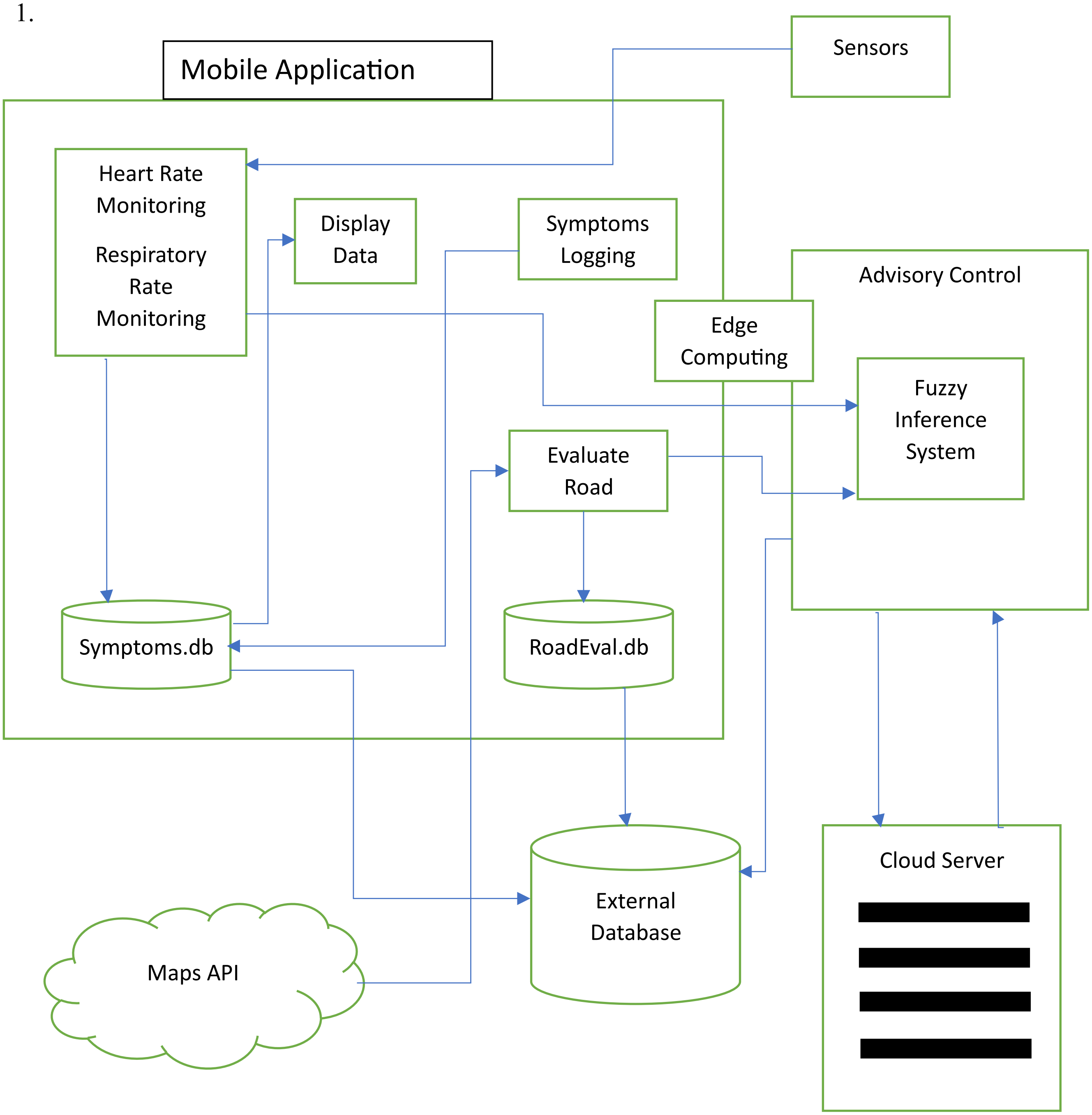
PROJECT 3 ANSWERS

1.



The integration of the mobile application and the advisory control requires communication between the road evaluation in the mobile application and the advisory control. This can be facilitated by either an API or utilizing a common database. Employing an API would be the better approach to mitigating any potential delays in obtaining the information. Furthermore, the road evaluation system retrieves information from Google Maps API. Additionally, to enable real-time decision-making, it is important to utilize external sensors attached to the user to obtain heart rate and respiratory rate information for which another API call is utilized. The advisory control can be deployed on an edge computing resource, or the context information can be sent to a cloud server and retrieve the decision from the cloud server. Employing an edge computing device would be the optimal approach, which would eliminate any potential risk due to delays in retrieving decisions from a cloud server. All the components must have wireless internet connectivity which can be achieved using WiFi modules that are built in with the IoT devices these days. Although there are numerous ways to perform computing and deploying the system, the ideal approach would be to use an edge computing device that processes the information and provides decisions at the edge. Thus, eliminating any possible latencies.

2.

The fuzzy system controller can be optimized by designing a better

- Membership functions
- Fuzzy rules
- Appropriate weights

However, in the video, there is a different and more efficient way to optimize the fuzzy system controller. The fuzzy logic controller is optimized by representing it in the form of a neural network and this neural network is trained using the Back Propagation algorithm, Genetic Algorithms, or any other nature-inspired optimization tool.

Fuzzy logic can be combined with neural networks in two different ways.

a. **Fuzzy neural network:** Neurons are designed using the concept of fuzzy set theory. It can have three kinds of inputs and weight combinations.
    i. Real inputs but fuzzy weights
    ii. Fuzzy inputs but real weights
    iii. Fuzzy inputs and fuzzy weights

    However, this approach is not widely used.

b. **Neuro-Fuzzy System (NFS):** This utilizes the Mamdani approach. NFS modifies the network to train the Mamdani approach. The neural network comprises of five layers.
    i. Layer 1 – Input Layer
    ii. Layer 2 – Fuzzification Layer
    iii. Layer 3 – AND operation layer
    iv. Layer 4 – Fuzzy Inference layer
    v. Layer 5 – Defuzzification layer

This neural network is tuned using the Batch Mode of training, Back Propagation algorithm, Genetic algorithm, or any nature-inspired optimization tool, thus optimizing the fuzzy system controller.

3.

The mobile application that we have built has the following functionalities:

- Monitor heart rate and respiratory rate using the flash of the camera and the accelerometer sensor of the mobile.
- Log the user's symptoms to the database.
- Measure the traffic conditions of the route.

The advisory control determines the deceleration of the autonomous vehicle so that it doesn't crash into any other vehicle. The fuzzy inference system in the advisory control utilizes the information from other cars (velocity). This whole system comprising the mobile application and the advisory control can be deployed in a Vehicular Ad-hoc NETwork (VANET) using the following strategies.

- The application can sense the alertness of the driver using the information available from the mobile application. Specifically, the heart rate and respiratory rate along with the symptoms of the user can be used to assess the health condition and the alertness of the driver. However, it is not feasible to use camera flash, accelerometer, and other sensors to measure this data from the driver when in the car. So, the user can wear a smart device that has sensors to capture this information, or the user must wear sensors so that the application can have real-time access to the data. Having access to this real-time data enables the application to assess the alertness and health condition of the driver in real time and share it over VANET. This information shared over the VANET can be utilized by the advisory control to determine the deceleration or braking force of the vehicle.
- VANET collects information from other vehicles connected, this enables access to real-time information about the drivers in other vehicles which can be utilized in the advisory control to determine the deceleration and braking force for all the connected cars independently. Additional information such as conditions of the cars, and proximity of other cars also aid in improving the advisory control.

Now, for the deployment of this application integrated into VANET, we would require real-time information from the sensors, compute resources, and network resources. This is made possible by employing the Internet of Things (IoT). All the applications and sensors are connected over a wireless network. The information collected can be processed using Edge Compute resources for minimal computations or can be sent to the Transportation Control System which is a cloud server that collects all the context information necessary to make the decision. The decisions are propagated back to the applications integrated into the VANET which then communicates it to the vehicle software allowing. Thus, allowing all the systems to work effectively without compromising on the user's safety. Deployment of edge computing devices is extremely useful in these applications to mitigate the processing delays (although accounted for in advisory control) over a cloud server, making the risk almost none.

4.

The input conditions given are:

- Driver Alertness – (Alert, Mentally Fatigued)
- Traffic Conditions – (Low Traffic, Medium Traffic, High Traffic)
- Vehicle Proximity – (Close, Far)

Rules:

IF the Driver is **Alert** AND Traffic is **Low** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Alert** AND Traffic is **Low** AND Proximity is **Close** THEN **No Alarm**

IF the Driver is **Alert** AND Traffic is **Medium** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Alert** AND Traffic is **Medium** AND Proximity is **Close** THEN **No Alarm**

IF the Driver is **Alert** AND Traffic is **High** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Alert** AND Traffic is **High** AND Proximity is **Close** THEN **Alarm**

IF the Driver is **Fatigued** AND Traffic is **Low** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Fatigued** AND Traffic is **Low** AND Proximity is **Close** THEN **Alarm**

IF the Driver is **Fatigued** AND Traffic is **Medium** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Fatigued** AND Traffic is **Medium** AND Proximity is **Close** THEN **Alarm**

IF the Driver is **Fatigued** AND Traffic is **High** AND Proximity is **Far** THEN **No Alarm**

IF the Driver is **Fatigued** AND Traffic is **High** AND Proximity is **Close** THEN **Alarm**