CS5551 Advanced Software Engineering

Problem Set 3 (PS-3)

Deadline: March 13 (T)

Submit a hard copy of your solutions to the instructor during the class

Name: Sneha Mishra
Class ID: 21

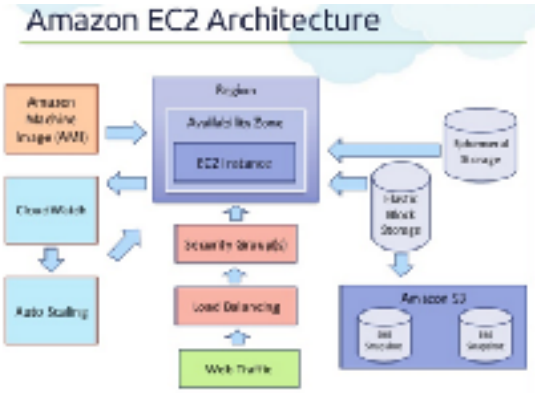
Our taxi mobile solution will maximize your revenue with ease by using our apps. We help you manage and maintain our advanced technology. You have time to grow and dominate your competitors. Attracting new and retaining regular customers has never been so easy.
TaxiMobileSolutions is the solution you have been looking for. Within a few short clicks using our app, available drivers will be able to view and accept pickup requests through our app. Customers will know the vehicle type and driver that is coming to pick them up.
With this innovative technology, you can seamlessly connect drivers to riders by using our apps. We help make cities more accessible for riders, while generating more revenue for drivers.
TaxiMobileSolutions is available for your customers and drivers on both iOS and Android platforms.
TaxiMobileSolutions will white-label the solution for your taxi company. In branding the solution for your company, the solution will have your company's logo and name, not TaxiMobileSolutions.
You also have the option to further customize our product, based on your unique needs. With our solution, you need not look any further, because our product includes everything that you need to operate your business smoothly, efficiently and profitably.
We have developed this technology in such a way that customers will happily avoid the typical inconveniences of:

- Calling a taxi service company and being put on hold.
- Waiting an extended period of time for service.
- Hailing a taxicab in unpleasant weather conditions.
- Riding with unknown drivers they might not be comfortable with.
- Fumbling around in a purse or wallet for payment and tips.

1. Draw a software architecture for TaxiMobileSolutsions.

2. Explain how a MVC pattern is applicable to your software architecture of "TaxiMobileSolutsions" application (Adroid and iOS platforms)

3. Draw a sequence diagram for Calling a taxi service company



Reference:

- AmazonEC2 software Architecture Diagram

  https://aws.amazon.com/architecture/

- MVC Architectural pattern

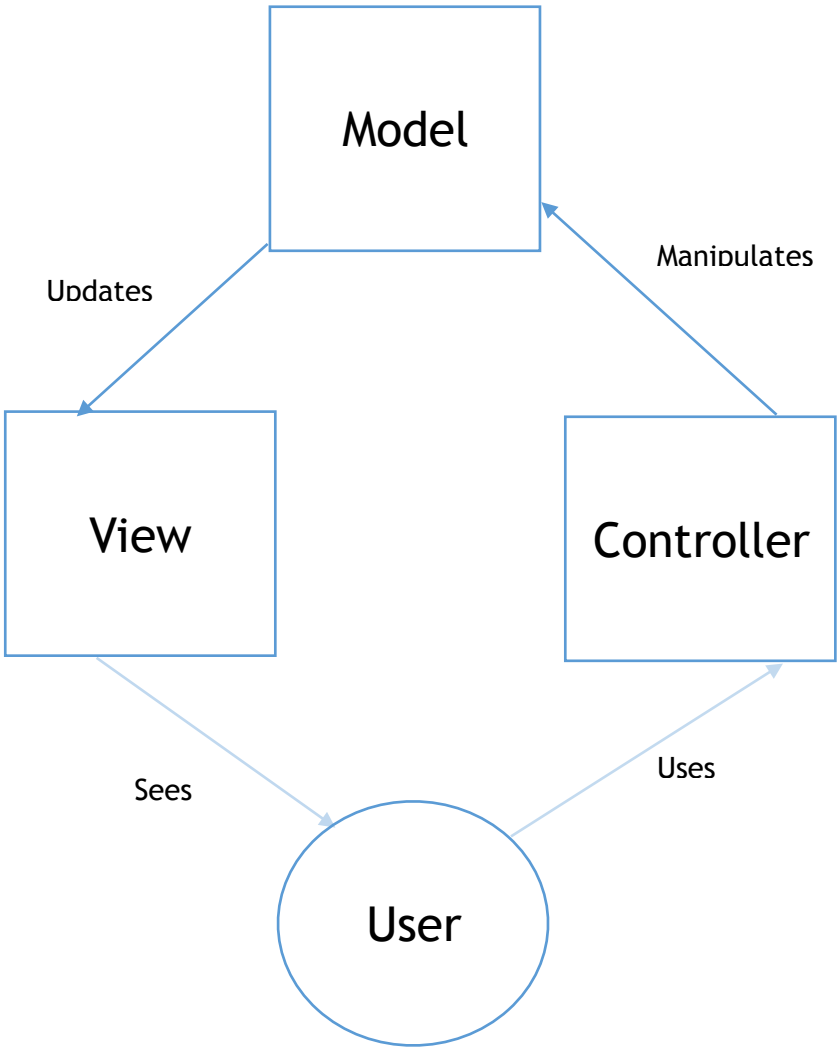
  https://developer.chrome.com/apps/app_frameworks

1. “TaxiMobileSolutions" can be thought of as a MVC (Model View Controller) Architecture. This divides the complete Application into 3 parts, namely Model - The Data part, View - The display part, Controller - The business Logic part.

- The *model* is the central component of the pattern. It expresses the application's behavior in terms of the problem domain, independent of the user interface.It directly manages the data, logic and rules of the application.

- A *view* can be any output representation of information. Multiple views of the same information are possible.

- The *controller*, accepts input and converts it to commands for the model or view.



2. **Model-view-controller** (**MVC**) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. So when it comes to the “TaxiMobileSolutions” App, we need the architecture pattern that can divide the application in such a way we have the information separated yet strongly connected to each other. In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them:

- The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

- The view means presentation of data in a particular format, triggered by a controller's decision to present the data.

- The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

Thus in the app, the user can update the information via controller, the updated info can be viewed on the views, while internally the model manages the user data based on the requests from the view or controller.

3. Sequence Diagram for calling a taxi service company: