Prince Sattam Bin Abdulaziz University
College of Computer Eng. & Sciences
Software Engineering Department

توكلنا

Tawakkalna

Step1:write functional and non-functional requirements for selected app:

❖ Functional Requirements

1. The user shall be able to login into the system using their national /residency id and password
2. The user shall be able to register a new account by entering a national /residency id and password.
3. The user shall be able to change his password.
4. The user should be able to hide his profile picture.
5. The user should be able to switch the application's language between either Arabic or English languages.
6. The user should be able to view news posted by the government.
7. The user shall be able to report a suspected covid-19 case.
8. The user shall be able to request permission to go out when under quarantine.
9. The user shall be able to view the electronic permits he received.
10. The user shall be able to provide his location on the map manually or using GPS.
11. The user shall be able to determine his residency location.
12. The user shall be able to update his residency information.
13. The user shall be able to request supplies permit inside the neighborhood.
14. The user shall be able to receive and view replies to his requests.
15. The user shall be able to view his violations of lockdown times.
16. The user shall be able to report violations of lockdown.
17. The user shall be able to request covid-19 vaccine injection for himself.
18. The user shall be able to request covid-19 vaccine injection for one of his dependents or sponsored individuals.
19. The user shall be able to book an appointment by entering the healthcare center, date, and time of taking the covid-19 vaccine injection.
20. The user shall be able to modify the healthcare center, date, and time of the appointment he booked.
21. The user shall be able to cancel an appointment after providing justification.
22. The user shall be able to cancel requests after providing justification.
23. The user shall be able to view his upcoming appointments.
24. The user shall be able to preview the health condition of the user's sponsored individuals and dependents using a QR code.
25. The user shall be able to view the times when roaming is allowed.
26. The user shall be able to view the reports published by the Ministry of Health regarding the number of covid-19 cases.
27. The user shall be able to preview his PoA in which he is either the principal or Attorney-In-Fact.
28. The system shall be able to notify the user of a suspected/infected/exposed case nearby the user.
29. The user shall be able to request an emergency medical permit.
30. The user shall be able to preview the details of Hajj permits.

❖ Non-Functional Requirements

1. Usability: The application user interface should be clear and self-explanatory
2. Size: The application shall take less than 150MB of storage on the user device
3. Security: The application must encrypt and store the information it manages safely.
4. Maintainability: The application must track and log the errors and mistakes for future analysis.
5. Reliability: The application system should be available 98% of the time.
6. Performance: The application should be able to authenticate users in less than 5 seconds.
7. Robustness: The application shall not fail upon wrong input data from the user.

Prince Sattam Bin Abdulaziz University
College of Computer Eng. & Sciences
Software Engineering Department

جامعة الأمير سطام بن عبد العزيز
Prince Sattam Bin Abdulaziz University

HUNGER STATION

Step 2: Review requirements written by another team

The name of the selected App:

Hunger Station

The functional requirements:

1- The system shall allow searching in the list of Restaurants

The system shall allow the user to search for specific restaurant in the list of available restaurants. (The requirement is not specific on what to search)

2- The system should allow the customer to see the Bill

The system shall allow the customer to see the Bill. (Customer must be able to view their bills upon purchase)

3- The customer shall register if he/she want execution of the order

The customer shall be able to register account by providing his phone number and password in order to purchase from the application . (form is weak)

4- The customer shall pay with different ways Cash / Visa / PayPal / mada etc.

The customer shall be able to pay using either Cash / Visa / PayPal / mada . (Functional requirement must not contain unspecified details like “etc”)

5- The system should allow the customer to track the delegate after payment

6- The customer shall enter information when registering (Incorrect same as requirement three)

7- The system should allow the customer to modify his data

The system should allow the customer to modify her location . (unspecific data)

8- The system should allow the customer to add more than one location

9- The system shall display the delivery fee to the customer

10- The system should display a list of restaurants that have offers

11- The system shall provide multiple food stores for a customer’s selection.

The system shall display all the available food stores for the customer. (Incorrect not accurate)

12- The system shall display to the customer the prices of different food items.

13- The customer shall be able to add multiple food items to the cart in the same order.

14- The customer shall be able to add the same food item more than once.

15- The customer shall be able to view his previous orders.

16- The system should show to the customer the expected delivery time for each order.

17- The system shall notify the customer if there is an update to the application .

18- The customer should be able to specify the side additions to the meals.

The customer shall be able to specify the side additions to the meals . (This is a must requirement)

19- The system shall allow the user to view the terms and conditions before confirming the order.

20- The system shall allow the customer to add a discount coupon code.

21 - The system shall display for the customers a list of nearest restaurant to their location

22- The system should provide English \ Arabic language

23- The system must show a site map that shows the distance of the delivery representative to deliver the order

The system shall be able to show a site map that shows the distance of the delivery representative to the delivery location. (Incorrect representation form)

24- The system shall allow a chat page to communicate with the restaurants delivery

The system shall provide a chat page where customers can communicate with the restaurants delivery.(Didn’t specify the participant of the communication process)

25- The customer shall get ordering tips which include receiving information about locations of restaurant areas

26- The customer should be able to send a feedback after the order arrival

27- The customer shall receive a warning sign if anything of the ordering process went wrong

28- The system shall provide customers with a service number for emergency cases

29- The customer should be able to cancel the order

The customer shall be able to cancel the order after a specified period of time . (This is a must requirement and it lacks detail)

30- The system shall save the canceled order money on the App wallet

The non-functional requirements:

Performance Requirements:

1- The system shall be available 24 hours a day

2- The system shall be able to load any page with no more than 3 seconds

Reliability Requirement:

3- The system should be reliable, that is it can carry its specific operations under all conditions.

Robustness Requirement:

4- The system must be robust in face of hardware power failures that could lead to data loss.

5- Security requirements:

customer's personal information shall be encrypted and stored safely .

6- Portability requirements

System shall be accessible from different platforms (mobile, tabs, computers).

Step 3:

In order to avoid selecting the same architectural pattern twice, each team should post here their selected architectural pattern once selected.

Each team must select a different possible pattern.

We choose pattern MVC.

Course Project: Step 4

Each team has to prepare a short presentation of five slides (not including title or conclusion) in English about the selected pattern.

MVC

Software design and architecture

What is MVC pattern?

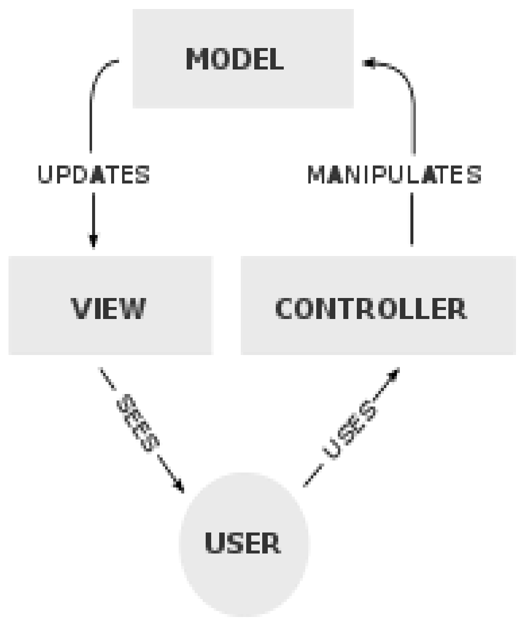
Model–view–controller (MVC) is a software architectural pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

MVC components:

Model : The central component of the pattern. It is the application's dynamic data structure independent of the user interface.

View : Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

Controller : Accepts input and converts it to commands for the model or view.



MVC advantages:

Easily Modifiable: MVC makes it easy to implement modifications to the entire app.

Easy planning and maintenance : The MVC paradigm helps a developer in the initial planning phase of the application because it gives an outline of how to arrange ideas into actual code.

Returns data without formatting : With the help of MVC developers can create their own view engine by receiving raw data.

MVC disadvantage:

Cost of frequent updates Developers cannot completely ignore the view of the model even if they are decoupled if the model undergoes frequent changes, the views could be flooded with update requests . Views like graphical displays may take some time to render

The isolated development process by UI authors, business logic authors and controller authors may lead to delay in their respective modules development

MVC must have strict rules over methods (appropriate reactions from Controller).

How MVC architecture works?

The controller receives the request for the application and passes it to the model to send and receive data.

The view then uses the data from the controller to generate presentable information to the end-user. The view is only concerned with how the information is presented, not with the end presentation. Thus, it is the Html file that renders the information from the controller.

The view then sends its final presentation to the controller, and the controller then sends the response to the browser, which then displays the response to the end-user.

Conclusion:

MVC makes the development process smoother compared to the traditional approaches. MVC is a great tool to translate your ideas into the code and you can easily come back to your code since you are aware of which code does what. The principle of partitioning lies in its core, so you have the input logic, business logic, and UI as separate components. It makes it possible to reuse the code and work simultaneously on several features, which accelerates development speed up to three times. Also, the way code is organized within MVC makes it easy for other developers to understand your code.

References:

- https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
- https://www.section.io/engineering-education/what-is-mvc-and-how-does-it-work/
- https://www.careerride.com/MVC-disadvantages.aspx

Thank You!

Do you have any questions for me before we go?

Step 5 (final step):
Each team has to prepare a tutorial explaining how to apply the selected architecture pattern for the selected app.

MVC in Tawakkalna Application

Table of Contents

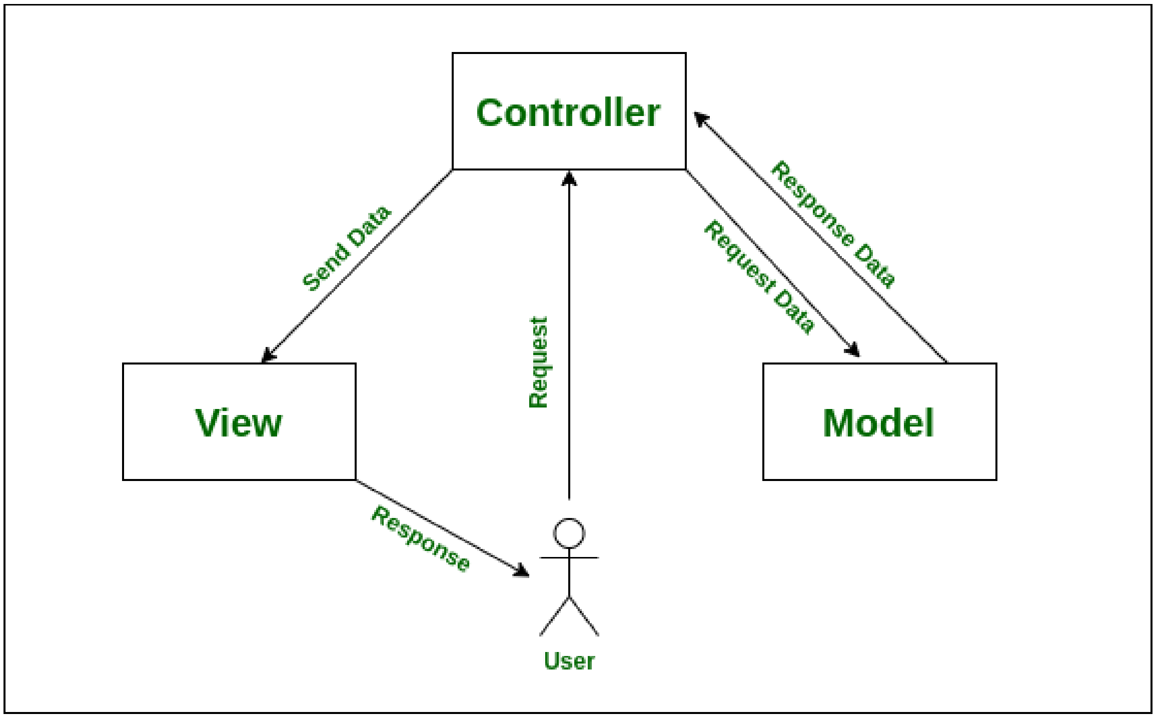
1. Introduction:

The MVC model is a structure that is relied upon to analyze a visualization of the proposed application. The system is divided into three main sections, and each of these sections passes data to other sections. In this article, we will review the structure through which we can rely on the MVC model. In simulating how the proposed system (our application) works.

We focus initially on that our application, according to the MVC model, should be divided into three sections, the section for showing results and the section for receiving requests, and the section for communicating with the database and passing it to the second section, and thus the second section returns the data it received from the section The third to the first section of the show, and thus we note the sequence followed in this process, and here we will simulate the processes through which we can use the MVC model in our proposed application.

As we know in the beginning that the application consists of a number of services it provides, and therefore each of those services, the MVC model must have its own structure or service that can be processed and obtain the corresponding results, and thus to clarify the structure of the use of the MVC model in our system an app called “توكلنا”.

view: In the case of the proposed application, this section is the interfaces of the Android or iPhone application, which are the interfaces with which the user interacts directly, which includes many interfaces, including creating the account, logging in, reviewing articles and reporting various cases, and thus in the case of our application, we can represent the view part of the MVC model as follows:



1.1 View Part:

- An interface for creating an account within the application.
- A special interface for logging into the account within the application.
- The application's settings interfaces, which include a number of different options.
- An interface for reviewing articles.
- Other interfaces that are represented in reporting a case of Corona, or requesting a vaccine, and other different interfaces included in the application.

1.2 Controller Part:

As we mentioned in the previous paragraph that the view part is the graphical interfaces included in the application, but in the case in which the user interacts with the graphical interfaces, this interaction must be sent to the server, and therefore the moment the request is received at the server, it is part of the controller and therefore we can summarize This part is in the case in which the account is created, and we will simulate that process as follows:

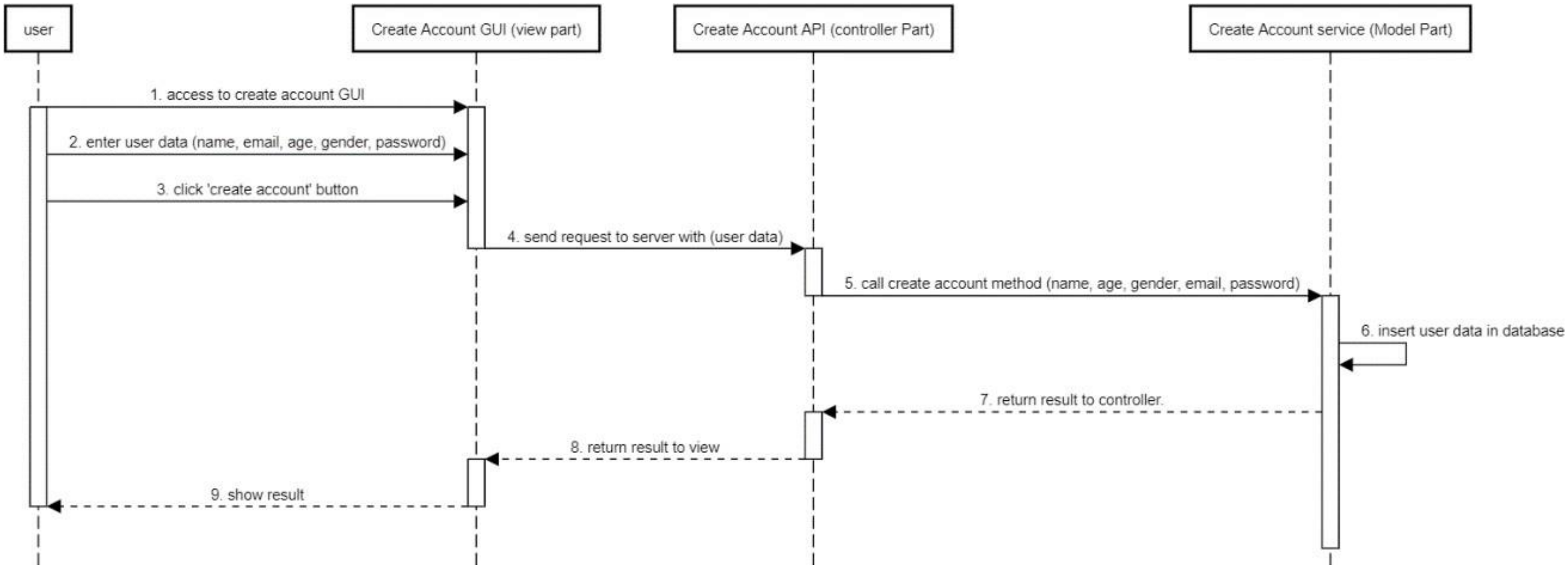
- The user enters the account creation interface (view part).
- The user enters the data within the required fields (name, age, gender, email, password), and presses the Create Account button.
- The moment the user presses the Create account button, the data is sent directly to the controller service located on the server that receives that request and determines the

model responsible for its implementation, and thus receives requests from the graphical interfaces resulting from the interaction of the user from that interface called the controller and he is responsible for determining model to be implemented.

1.3 Model Part:

As we mentioned in the previous stage, when the request that was sent from the user is received by the Controller, the controller specifies the model service that should be executed, and therefore the model part is concerned with receiving requests from the Controller, communicating with the database, retrieving the results and returning those results back to the controller So in our case, the result of creating the account for the user after connecting to the database will return the model to Controller and then Controller will return the result to view and here the result will be shown to the user.

The following figure shows the mechanism of the account creation process within the MVC model.


user
Create Account GUI (view part)
Create Account API (controller Part)
Create Account service (Model Part)
1. access to create account GUI
2. enter user data (name, email, age, gender, password)
3. click 'create account' button
4. send request to server with (user data)
5. call create account method (name, age, gender, email, password)
6. insert user data in database
7. return result to controller.
8. return result to view
9. show result