**NJERU OLIVER NJIRU 663565**

**INTRODUCTION TO SOFTWARE ENGINEERING INDIVIDUAL PROJECT**

**Discuss model view architecture and show how you could have designed the architecture of group project using these architectural style**

In this case, I would use the Model View Architecture to show how I could have designed the architecture of the Video Rental and Sales System that was my group project if it were online, but before I do that, I need to explain what the Model View Architecture is. The Model View Architecture actually is called the Model View Controller architecture according to (Point, n.d.) and its definition is that it is an architectural pattern that separates three main logical components: the model, the view and the controller all of which are vital. Since our project was the Video Rentals and Sales which I’m going to implement via the web, it made sense for me to use the MVC simply because it supports a great deal of scalability which is vital in a Video Rental and Sales system to reach a bigger audience of subscribers.

I will now explain the roles of the three logical components of the MVC model. According to (Point, n.d.), the Model logic corresponds to all the data-related logic that the user works with and can represent either the data being transferred between the view and the controller logical components or any other business logic-related data. In this case, the customer will retrieve their information regarding their account details, after they have signed up, from the database, manipulate it in terms of say changing their profile picture, name or payment details, and send this new data back to the database that will perform an update across the system and use the new data to render to the customer.

The second logical component of the MVC model is the View component. The view component, according to (Point, n.d.) is used for all the User Interface logic of the application and in our case, it will be the Landing page and the different dynamic webpages that the customer will be interacting with as they navigate through the website. Since it is a Video Rental and Sales website, we will have to make sure that out implementation is not only visually appealing and eye catching to the customer, but also has a seamless and interactive User Experience to make sure that we convince the user to either rent a video or make a purchase and show them the maximum visual quality they can get from the particular video, be it for children or adults. We would add interactive components such as dropdowns, check lists or radio buttons and use frameworks such as React to make it look good.

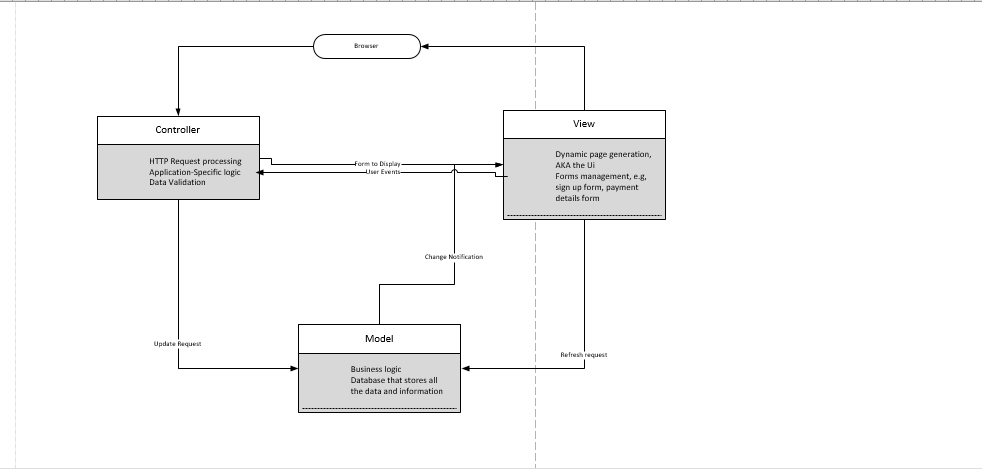
The third logical component of the MVC model is the Controller component, which according to (Point, n.d.), acts as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the model component and interact with the views to render the final output to the user. In out case, say the user want to sign up for a new account, they successfully fill in the sign up form and click the sign up button. Once the button is clicked, the controller is now responsible for first making sure that the connection between the User interface and the database is established, so in this case, the controller can perform a try and catch statement so that try and see whether the connection is established or not. The try part will check for the connection and if successful proceed to execute the rest of the code but if not successful then the catch part will be print out the error that is causing the connection to not be established.

Now, with all this explained, why should we implement this architecture to our system? According to (Hernandez, 2021), we should implement the MVC so that we can separate the frontend component from the backend component. By doing this, it will be much easier to manage and make changes to either side without them interfering with each other or bringing up a system conflict. According to (Svirca, 2020), if we use the MVC model, we get to tests components independently and have a high and efficient test result, we also get to deploy and maintain the different components of the MVC. From that, we also get to reuse components, say in our case, as components in other features within the system. Using the MVC also means that we get to have a system that is Search Engine Optimization friendly to search engines like Googe, Bing, Brave, DuckDuckGo and the rest, according to (Martin, 2022), which can help the system be visible to a huge audience and bring in more traffic which increases our chances of rent a video or selling one.

Finally, according to (Martin, 2022), by using the MVC, we get to use popular frameworks in the industry such as:

* Laravel
* Symphony
* Django
* Spring MVC
* Ruby on Rails
* Catalyst
* CherryPy
* Yii
* Rails
* Fuel PHP
* CodeIgniter
* Zend Framework
* CakePHP

The figure below shows the MVC:



The MVC above shows how the system would be built on the web all the way from the View, UI, the Controller, to handle and process requests from the view and the model and the model which has the business logic and the database.

# References

Hernandez, R. D. (2021, April 19). *The Model View Controller Pattern – MVC Architecture and Frameworks Explained*. Retrieved from FreeCodeCamp: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/

Martin, M. (2022, July 2). *MVC Framework Tutorial for Beginners: What is, Architecture & Example*. Retrieved from Guru99: https://www.guru99.com/mvc-tutorial.html

Point, T. (n.d.). *MVC Framework - Introduction*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_introduction.htm#:~:text=The%20Model%2DView%2DController%20(,development%20aspects%20of%20an%20application.

Svirca, Z. (2020, May 29). *Everything you need to know about MVC architecture*. Retrieved from TowardDataScience: https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1