Fundamentals of Software Design Unit 5

David Finger

Colorado Technical University

Abstract

Define the most important concepts of MVC architecture regarding GUI design and development. Show 2 Design patterns to enhance a MVC paradigm with regards to a mobile application. Using a simple web application development tool kit, create a prototype for hello world web display..

The Most important design concept of the Model-View-Controller, here within further referred to as MVC, paradigm for a GUI is the separation of the roles between the model, view, and controller portions into usable layers. This allows for each portion to be applied separately and their resources stored separately as well.

The Model Portion of the MVC paradigm is any data used by the software program itself. This includes all icons, files, databases, or pointer objects that are available within the software (*MVC (Model-View-Controller) Definition*, n.d.).

The View portion of the MVC paradigm is anything used to display objects within the application software. This includes the windows that are shown as well as any button or text that appear within the application window (*MVC (Model-View-Controller) Definition*, n.d.).

The Controller portion of the MVC paradigm is used to update both the Model and View portions. This section takes user input and sets in motion the corresponding updates within the code to make things happen. An example of this would be for the user to resize the screen, which the controller tells both the view and model portions how to change for everything to still operate properly within the software environment (*MVC (Model-View-Controller) Definition*, n.d.).

Design Patterns to enhance the MVC paradigm for the GUI of a Mobile App commonly used are the Decorator design pattern, and the Strategy design pattern. Regarding mobile applications, designers must take special considerations into account during the design process. These considerations are that the application must be able to support dynamic properties, avoid being to complex in its interactions, and maintain a certain level of flexibility within the application.

**Design Pattern 1:** The Decorator Design pattern is used within the MVC controller layer to allow users to add functionality to existing objects without altering the overall structure of the object. In this sense, the decorator class wraps the original object class which adds functions to the original class without interfering with what the original class could do or show (*Design Patterns - Decorator Pattern - Tutorialspoint*, n.d.).

Diagram

Description automatically generated

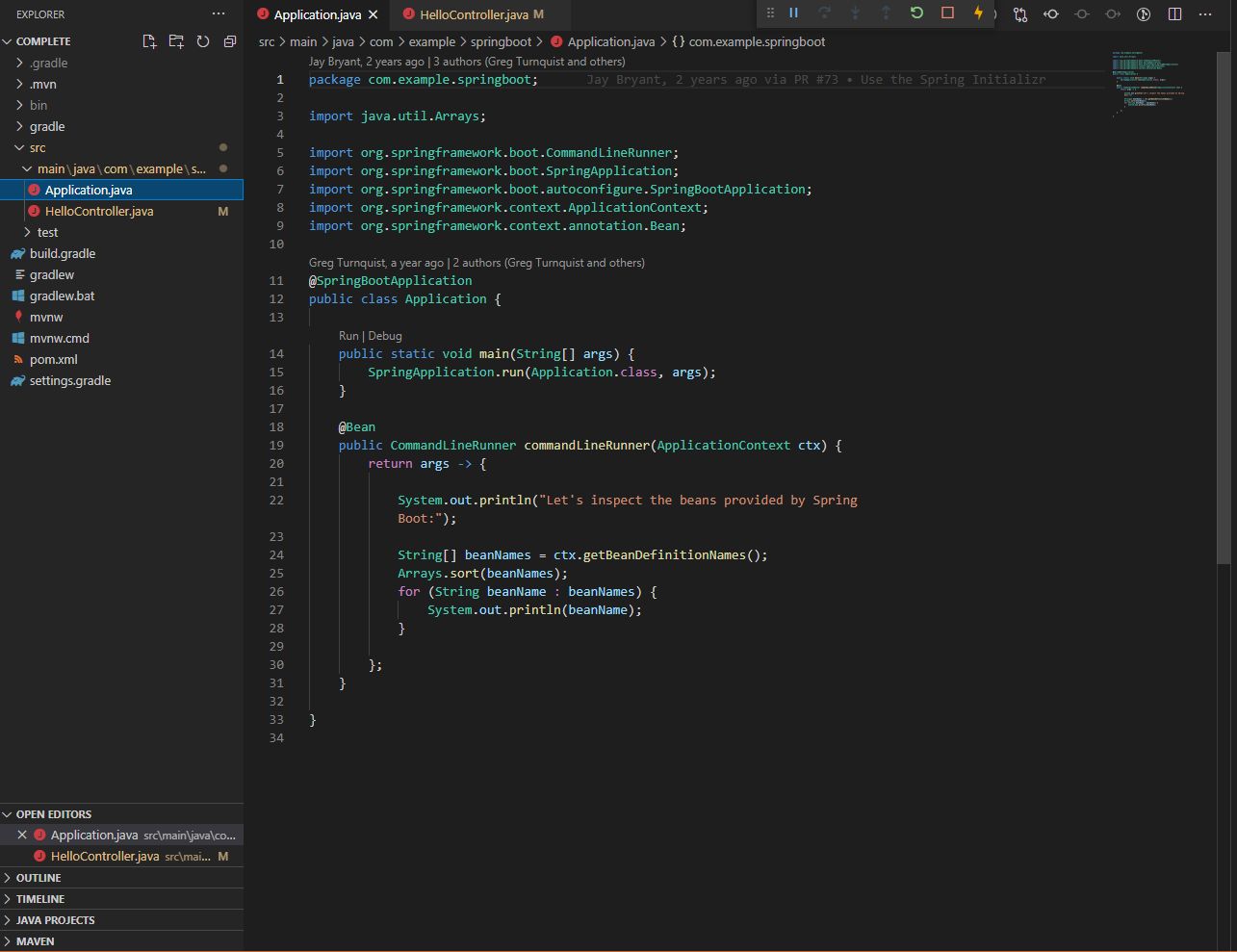
**Design Pattern 2:** The Strategy Design Pattern is used to show the structure behind how the controller interacts with the view layer to apply changes requested by the user. The strategy pattern is used to alter the view of objects under certain selected user specified operations (*Design Patterns - Strategy Pattern - Tutorialspoint*, n.d.).

Diagram

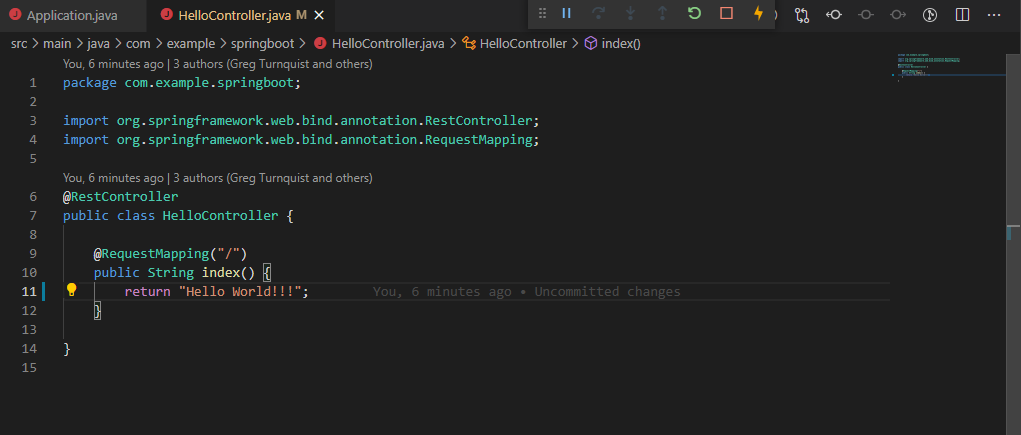
Description automatically generated

**Prototype of the GUI of a Simple Web Application:**

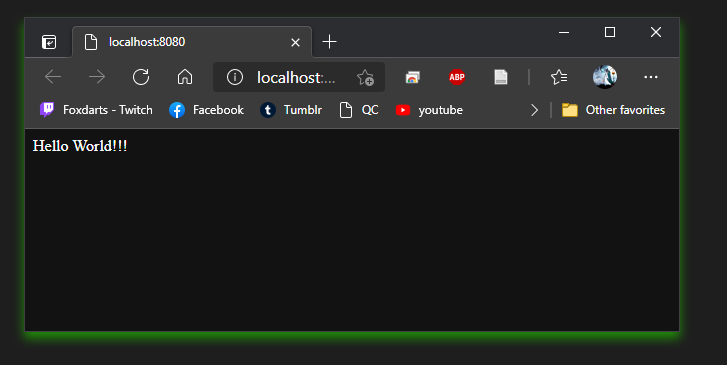
Code for Model screenshot:



Code for Controller Screenshot:



Output screenshot, aka view:



This code didn’t really change from its original form. It is a straightforward MVC design that clearly shows the model being created by code, then being used by the controller to display to the view screen.

Resources

*MVC (Model-View-Controller) Definition*. (n.d.). Retrieved May 3, 2021, from <https://techterms.com/definition/mvc>

*1. MVC GUI Design*. (n.d.). Retrieved April 29, 2021, from <https://www.cs.wcupa.edu/rkline/java/mvc-design.html>

*Best Practice Software Engineering—Model View Controller*. (n.d.). Retrieved April 29, 2021, from <http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html>

*Design Patterns for Mobile Applications*. (n.d.). Retrieved April 29, 2021, from <https://www.infoq.com/presentations/mobile-design-patterns/>

Shahbudin, F. E. (2013). Design Patterns for Developing High Efficiency Mobile Application. *Journal of Information Technology & Software Engineering*, *03*(03). <https://doi.org/10.4172/2165-7866.1000122>

*Design Patterns—Decorator Pattern—Tutorialspoint*. (n.d.). Retrieved May 3, 2021, from <https://www.tutorialspoint.com/design_pattern/decorator_pattern.htm>

*Design Patterns—Strategy Pattern—Tutorialspoint*. (n.d.). Retrieved May 3, 2021, from <https://www.tutorialspoint.com/design_pattern/strategy_pattern.htm>

*Build and Deploy Java Web Apps to the cloud with Visual Studio Code*. (n.d.). Retrieved May 4, 2021, from <https://code.visualstudio.com/docs/java/java-webapp>