Architecture Design Document:

Version <1.1>

Table of Contents

1. Introduction 3
   - 1.1 Purpose 3
   - 1.2 Scope 4
   - 1.3 Definitions, Acronyms, and Abbreviations **Error! Bookmark not defined.**
   - 1.4 References **Error! Bookmark not defined.**
   - 1.5 Overview **Error! Bookmark not defined.**

2. Architectural Goals and Constraints 4

3. Architectural Representation **Error! Bookmark not defined.**
   - 3.1 Architectural Views **Error! Bookmark not defined.**
   - 3.2 Architectural Design Patterns 5
   - 3.3 Architectural Style 5
   - 3.4 Architectural Process 5

4. Deployment View 6

5. Quality 7

6. Functions 7

Revision History

Date	Version	Description
<5/04/2018>	<1.0>	First Draft of the Software Architecture Doc.
<12/04/2018>	<1.1>	This revised version contains a more complete portion for each of the design states.

Introduction

This introduction provides an overview of the entire *Software Architecture Document* for the Mr E Online System. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the system.

Purpose

This document provides an architectural overview of the Mr E Online System (MEO).

The primary purpose of the Mr E Online System (MEO) is to satisfy the needs of online video rental solution.

This document is intended to capture and convey the significant decisions which have been made in designing and building the system. It is a way by which the systems' architect and others involved in the project can better understand the problems to be solved and how it will be represented with this system.

Scope

Definitions, Acronyms and Abbreviations

MEO: Mr E Online

DAC: Data Access Control.

MVC: Model View Controller

Architectural Goals and Constraints

The MEO architecture has been designed with the following objectives in mind:

1. To facilitate in making sure that the Bricks and Mortar store is not the only way to get your Video rented out, and profitability can continue.
2. To help manage all physical and digital videos to be maintained.
3. To allow customer retention to be maintained.
4. To allow the store to become more profitable from their bricks and mortar store but also adding the online store also to generate income.
5. To keep in touch with our current customers, and get a better customer reach without being limited to the physical store.

The major design and implementation constraints for the system are:

1. Simplicity
2. Flexibility

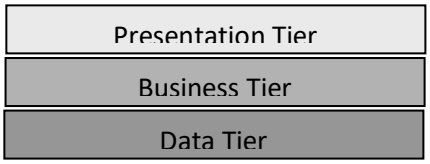
Architectural Design Patterns

The design pattern used to create the MEO is the MVC (Model View Controller) design pattern. The MVC design patter clearly separated the web application's behavior, presentation and control. The modularity of this design pattern allows for easier code reuse, more centralized control, bugs easier to track down and code

easier to modify. The presentation, or view, of the MEO has been implemented keeping in mind the Model usage pattern, which makes use of Views as front controller. Some of the sections of the MEO where architecture is used are the DAC for the DB.

Architectural Style

As with any other things, a style may be used to satisfy any functional, non-functional or aesthetic needs in a software system. The MEO, in particular, follows the three-tier architectural style: presentation tier, business tier, and data tier. The following is a simple description of what will be included in each of the tiers:



- Presentation Tier: used to format and present the information to the user.
- Business Tier: used to implement the logic that will drive the system and the reason why the system exists.
- Data Tier: in charge of storing the data (databases) and other external services that the system may use.

Deployment View

The MEO deployment view of a system shows the physical nodes on which it executes. The CPSS is comprised of three physical nodes: the browser, the application server, and the database server. The simplicity of the CPSS physical view can be seen in the diagram below:

A link from Github will be provided together with the database for deployment.

You will need to do a database restore on the database on Ms-sql.

Once the project is open, you have to change the **Config.xml** to the connections the database, which you will find in main project called MrEOnline

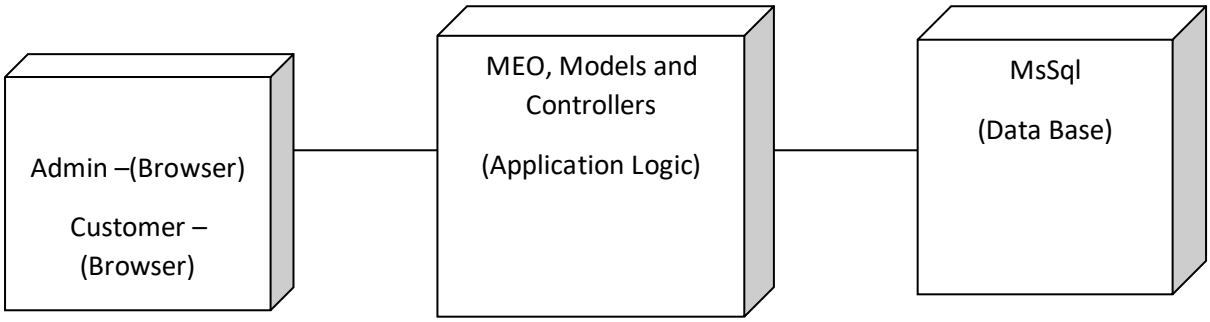
Than have a connection string to the DAC file in the project.

Than **build** the project to see if there are any errors.

Buid as follows: DAC, MrEOnline, ViewModel , then CORE , then MrEOnline again

Than run the project.

- **Credentials:** admin@mreonline.co.za and password: admin



Quality

System quality issues or concerns will go here should there be a need to change or add more functions.

Functions:

1. Create a xml file in the project for MEO
    - The xml file is created because this is where the connection information to the database is housed.
    - Where you will find the following connection string
2. A new project is created within MEO solution. It will be created as a class library
    - This new project is call DAC. DAC will be used to connect and communicate with the config.xml. In this class library called DAC, a class called Constrarts where a connection string is created.
    - And a folder called Stored Procedures will be sitting , that will be used to call the store procedures will be sitting, that will be used to call all the database stored procedures that are sitting on MsSql DB.
3. Added a landing page called “Home” index. This is where the customer and employees will see when they get to the system upon loading. They will be prompt to Login or signup if they’re customer and employees as admins.
4. The index view page consists of the login and registration function. The reason why this is done is to minimise response time, however, in the HomeController it will call methods that are designated for each function(login and registration).
5. The stored procedure called registration class is now added.
6. A new class library project is there because this is where all the functionality between DAC and MVC will be.
7. The class library called ViewModel is where all the results that come from the Database go through before you can see your View.
8. A reference called Dapper is added to help with the writing of the sql queries.
9. Admin dashboard: This is where all admin will manage the customers and the videos.
    - A video upload file is implemented so that the admin can upload a csv of the video’s they have at the MEO store.
    - A csv is created of the videos that must be uploaded.
10. An admin can be created with this system as well to help manage everything as well.
11. Once the file(csv) is uploaded the videos can be viewed. The admin can update, add, archive and restore videos.