CS4610
Project Management Plan

Project Report for Mumba
November 18, 2018

Sam Miller

Contents

1 Introduction

Mumba (https://mumba.azurewebsites.com) provides a simple no frills product management solution for small solo projects like those seen in university classes. My website is open to all and I hope that my project can be of use to myself and others in managing their projects and tasks. Anyone can register an account and use my application. Once the user has logged in they can boards which are collections of lists. They can then open the board and create tasks/cards in different lists to keep track of steps to complete a project. There currently is no user to user interactions, a board made by one user cannot be shared with any other user.

My Project is implemented as a full stack C# ASP.Net Core application, I used the MVC pattern to dynamically manage the views returned to the users of my application. The buttons and interactions a user has with the website are controlled by one of the Four controller classes which will be detailed in the Description of Components section. The SQL server behind my web app is dynamically called to bring forth only the correct data that belongs only to the user who requested them. Each http request is routed to the proper controller's method for handling the request.

2 Related Works

The core functionality of Mumba is inspired by project boards (commonly known as a kanban board) is commonly seen in Atlassian's Trello.com. Mumba is much simpler, there are not as many features as included in Trello, for example in Mumba's current state each board has 3 lists and only 3 lists, Trello dynamically allows creation and deletion of lists.

3 System Architectural Design

The system architecture follows the Model View Controller model of web design. The application was deployed across two Azure instances one is the platform as a service Web App Container that hosts my ASP.Net Core MVC web application, the second instance is a Windows Server Running an SQL Server. With a Azure web application a domain name is automatically registered, in my case my website can be reached at https://mumba.azurewebsites.net.

The views of my websize use Razor (HTML + C# functionality) pages as a base and materialize css: https://materializecss.com as the basis for the look and feel of my website.

3.1 Relational table: AppUsers

3.2 Relational table: Identity Users

3.3 Relational table: Boards

3.4 Relational table: Tasks

4 Detailed Description of Components

The following is a detailed Description of each functional componet of the Mumba application, I have organized them by the Controller class that handles the requesst and response for each call. Each handled http request is directed to a controller and then a specific method of that controller. In most cases a new view is called but a small handful of cases do not return a view. Invalid calls will result in a error page being displayed, and in order to view information beyond the register and login page one must be logged in.

4.1 Account

The Account controller handles session management and login/log out functionality.

4.1.1 Login

Takes a LoginRequest model that is generated by taking the username and password from the login views form and it will also grab a boolean value based off of the checkbox for session remembrance. It will then attempt to log the user in using the loginrequest model, if the request fails the user is redirected to the login page. If the login request succeeds then the user is sent to the BoardsController's All method.

4.1.2 Logout

4.2 User

4.3 Boards

4.4 Tasks

5 Conclusion

The final implementation of Mumba delivers a fully functional web application hosted at https://mumba.azurewebsites.net that meets the goals of providing a simple and easy to use project management solution. The service is very secure, with security as a choice from the very beginning. Using the Microsoft Identity package the website handles session management and data security natively. I used the CSS grid to create the many different elements of my pages, and in doing so made my website responsive, it will function and look good on any screen size. The main areas that Mumba could be improved are the following:

- One: Allow the dynamic creation/deletion/naming of lists within boards.
- Two: Create a method of sharing boards between two or more users to allow collaborative projects to be managed in Mumba

Once both of these features are implemented Mumba would be in a state that I would consider feature complete, anything else that would be added would just be extra.