|  |
| --- |
|  |
| CS3213 HW-4 |
| Team Moai |
|  |
|  |
|  |

**Members:**

Darren-Gavin Ho A0087858E

Jason Teo A0084895J

Lin Weizhi A0087084X

Luo Guo Jing A0086765M

Ong Shi Yun A0087666L

Contents

[1 Work Description 2](#_Toc371971309)

[1.1 HW-2 2](#_Toc371971310)

[Models and Views 2](#_Toc371971311)

[Collections 2](#_Toc371971312)

[Events 2](#_Toc371971313)

[OAUTH2 2](#_Toc371971314)

[1.2 HW-3 3](#_Toc371971315)

[Model 3](#_Toc371971316)

[View 3](#_Toc371971317)

[Event 3](#_Toc371971318)

[Sync 3](#_Toc371971319)

[HTML Templating 3](#_Toc371971320)

[1.3 HW-4 4](#_Toc371971321)

[2 Design Considerations 5](#_Toc371971322)

[2.1 HW-2 5](#_Toc371971323)

[User Authorisation 5](#_Toc371971324)

[2.2 HW-3 5](#_Toc371971325)

[OVERALL Structure 5](#_Toc371971326)

[3 Team Roles 5](#_Toc371971327)

[HW-3 & HW-4 5](#_Toc371971328)

[4 Learning Experience 6](#_Toc371971329)

[5 References 7](#_Toc371971330)

[6 Suggestions and Future Improvements 7](#_Toc371971331)

[HW-2 7](#_Toc371971332)

[HW-3 7](#_Toc371971333)

[7 GIT Repository URLs 7](#_Toc371971334)

# 1 Work Description

## 1.1 HW-2

Task: To develop a single-page JavaScript application, using **Backbone.js** and **Ruby on Rails** for the front-end application and with **movies@cs3213** as the API server.

Our application for HW-2 was named **Super Movie App**.

### Models and Views

Using **Backbone.js**, we created numerous Models and Views to form our application data and interface. Models created included those for movies and reviews. Views included those for viewing a list of movies, viewing movie reviews, adding a new movie entry, adding a new movie review, editing a movie entry and editing a movie review.

### Collections

We also used the Collections class of **Backbone.js** to store the movies and reviews received from the **movies@cs3213** server.

### Events

To allow a user to navigate around the page, we implemented Events in the form of buttons for clicking. This allowed the user to shift between Views easily. For example, to go create a new movie entry, the user would just have to click on the corresponding button. The Event will be registered by the application and the View for creating the new movie entry will be rendered in response.

### OAUTH2

For certain tasks, user authorisation was required before they could be performed. To achieve this, we implemented our application as a client for **OAuth2** authorisation. When the user first opens the page, he will be navigated to the log-in page if he is not already logged. The user will then log in to **movies@cs3213** and authorise our application to carry out those tasks.

## 1.2 HW-3

Task: To create a custom JavaScript framework that can replace **Backbone.js** in Homework 2.

The framework created was named **Barebone.js**. It implements a Model class, a View class, an Event class and a Sync class.

### Model

Similar to **Backbone.js**, our Model class contains the interactive data and the logic to manipulate it: conversion (to JSON), retrieval (resetting the state to that of the server), saving (for persisting the state to a server), deletion (destroying the state of the server).

To use the Model class, you extend **Barebone.Model** with domain-specific methods. It only provides a basic set of functionality for managing changes to the data.

### View

The View class allows for the logical organisation of the interface. Each view can be tied to a Model and will update independently when its corresponding Model changes.

To use the View class, you extend **Barebone.View**. It comes with a render function that the user overrides with his own domain-specific instructions.

### Event

The Event class is used by an object when it has custom events that are to be bound and triggered.

Classes that need to handle events have an Event object in their prototype.

### Sync

The Sync class is responsible for performing Ajax requests. It uses **jQuery** to do so. It allows the user to perform Create, Read, Update and Delete operations with a server.

### HTML Templating

The framework also allows for HTML templating. This was done using **Underscore.js**.

## 1.3 HW-4

Task: To replace **Backbone.js** in Homework 2 with the custom framework.

For HW-4, our goal was to be able to replace **Backbone.js** with **Barebone.js** in its every occurrence without major alteration to the code of HW-2. This required us to conform to the structure of **Backbone.js** in our work for HW-3.

# 2 Design Considerations

## 2.1 HW-2

### User Authorisation

In the implementation of our application, we forced the user to be logged in at all times when he uses it. This was done in this manner due to time constraints. It would have taken us more time to implement the application in such a way that allowed the user to have a different set of access rights to tasks depending on whether or not he was logged in. By simplifying it in this way, we ensured that all the features we implemented would only have to cater to users who were logged in.

## 2.2 HW-3

### OVERALL Structure

The overall structure of **Barebone.js** follows that of **Backbone.js** closely. We chose to do it this way as we found it difficult to find a starting point if we were to create our own design. By mimicking how **Backbone.js** structured its code, we were able to start on implementing similar functionalities first, but using our original implementation. The objective was to be able to achieve minimum functionality before worrying about extra features.

# 3 Team Roles

## HW-3 & HW-4

Darren DOM Events HW-4

Jason Model HW-4

Weizhi Model View Event Sync HW-4

Guo Jing Report HW-4

Shi Yun HW-4

# 4 Learning Experience

In summary, HW-2, HW-3 and HW-4 taught us the significance of frameworks in today’s software development scene. Frameworks capture the essence of good patterns and enable us as programmers to quickly develop programs without having to re-implement commonly used features.

HW-2 familiarised us with **Backbone.js** by requiring us to use it in our implementation of an application. This allowed us to see the use of frameworks first-hand, as well as the convenience that that brings about.

HW-3 further established our understanding of the software design principles behind frameworks by asking us to create our very own framework. This process led us to analysing what was done to **Backbone.js** to make it work, and then to applying what we learnt to our own product.

HW-4 required us to utilise our custom framework in implementing an application, one as specified by HW-2. This process allowed to ensure that what was done in HW-3 could work in a real setting and uncovered the loopholes of our framework. This brought us to a higher level of understanding of the requirements of a robust framework, and how frameworks are used in software development.

From working on these assignments, we have come to realise that frameworks are the result of developers discovering software design principles through repetition and their desire to allow others to utilise their knowledge conveniently. By learning the design principles that go into developing a framework, every software developer can stand to improve himself.

# 5 References

**Backbone.js** Official Page:

[http://backbonejs.org](http://backbonejs.org/)

**Ruby on Rails** by Michael Hartl:

<http://ruby.railstutorial.org/>

# 6 Suggestions and Future Improvements

## HW-2

**Super Movie App** can be improved by implementing features that differentiate whether a user is logged in or not. This would also require the application to be implemented such that it does not force the user to be logged in at all times when using the application.

This would enhance the overall user-friendliness of the application.

## HW-3

**Barebone.js** can be improved by the implementation of more features commonly used by applications of the MVC flavour. These features include additional classes for Collections and Routers. On top of this, HW-3 can be improved by implementing more functions in existing classes.

# 7 GIT Repository URLs

HW-2:

GIT: <https://github.com/ecc-weizhi/cs3213MovieApp.git>

Heroku: <https://super-movie-app.herokuapp.com>

HW-3:

GIT: <https://github.com/dvin89/cs3213hw3.git>

HW-4:

GIT: <https://github.com/dvin89/cs3213hw4.git>

Heroku: https://cs3213hw4.herokuapp.com