Evolution Four

ECE 458

Parker Hegstrom (eph4)
Peter Yom (pky3)
Wayne You (wxy)
Brandon Chao (bc105)

Abstract

Contents

1 Overall Design **2**

2 Back End Design and Analysis **3**

2.1 New Features and Developments . . . 3

2.2 Benefits of Our Previous Design . . . 3

2.3 Drawbacks of Our Previous Design . . . 3

3 Front End Design and Analysis **3**

3.1 New Features and Developments . . . 3

3.2 Benefits of Our Previous Design . . . 3

3.3 Drawbacks of Our Previous Design . . . 3

4 Individual Portion **3**

1 Overall Design

The overarching design principle we wanted to achieve was modularity. In doing so, we believed we would be able to work separately (with occasional meetings to work through minor problems with the API) and refactor without the worry of breaking another member's code. Figure 1 below shows a high level diagram of how we decided to design our calendar web application.

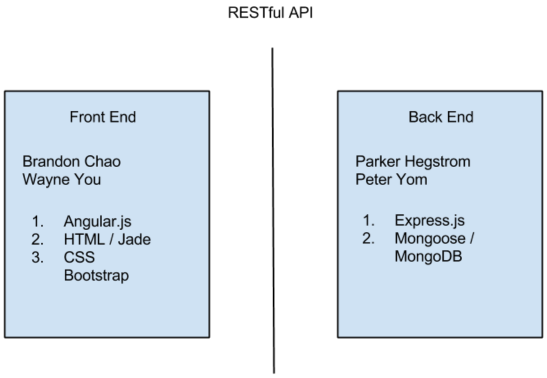

Figure 1: Diagram of our Large Scale Design

Essentially, our back end team provides an exhaustive RESTful API service to our front end. As we received the new requirements for evolution two, the benefits of our modular design came to light as we met to discuss both the refactorings from evolution one that needed to be done and the edits to each modules system design in order to account for the added calendar functionality–event requests and persistent until done events.

The following sections further discuss design choices and implications of those design choices for both our front end and back end teams.

2 Back End Design and Analysis

2.1 New Features and Developments

2.2 Benefits of Our Previous Design

2.3 Drawbacks of Our Previous Design

3 Front End Design and Analysis

3.1 New Features and Developments

3.1.1 PUD Events

3.1.2 Preference-based SignUp

3.1.3 En Masse Event Creation

We implemented en masse event creation using simple text parsing. Since we have the user input the event with a Name, Description, Start Time, and End Time on separate lines, we simply split the input string based on the new line tokens and create separate events for each series of 4 lines of text.

3.1.4 Schedule by Email

Our front-end implementation of this feature was fairly easy. For text schedule, we simply parse through all of the events within the currently display frame and send them to the back-end. For image schedule, we take a screenshot of the calendar in the current display and send it to the back-end to email.

3.2 Benefits of Our Previous Design

3.3 Drawbacks of Our Previous Design

4 Individual Portion

Parker

a) **Designing and Conducting Experiments**

.

b) **Analyzing and Interpreting Data**

.

c) **Designing System Components**

.

d) **Dealing with Realistic Constraints**

.

e) **Teamwork and Team Member Interaction**

.

Peter

a) **Designing and Conducting Experiments**

.

b) **Analyzing and Interpreting Data**

.

c) **Designing System Components**

.

d) **Dealing with Realistic Constraints**

.

e) **Teamwork and Team Member Interaction**

.

Brandon

a) **Designing and Conducting Experiments**

.

b) **Analyzing and Interpreting Data**

.

c) **Designing System Components**

.

d) **Dealing with Realistic Contraints**

.

e) **Teamwork and Team Member Interaction**

.

Wayne

a) **Designing and Conducting Experiments**

.

b) **Analyzing and Interpreting Data**

.

c) **Designing System Components**

.

d) **Dealing with Realistic Constraints**

.

e) **Teamwork and Team Member Interaction**

.