ECE 458: Engineering Software For Maintainability

Senior Design Course

Spring 2015

Evolution 1 Analysis

Brian Bolze, Jeff Day, Henrique Rusca, Wes Koorbusch

Contents

1 Introduction

Good software design is often seen not only as a science but as an art. It is a craft and, like any other type of engineering, is only mastered over time. The fundamentals of good software design, however, remain sound. In this senior design course, we plan on synthesizing our four years of knowledge through the development of a robust and long-lasting software application implementing a web-based calendar. We plan on applying the core principles of good design to our code and to our design process, while continuously evaluating, refining, and improving on our skills of the software engineering trade.

For our project, we built an application using the Ruby on Rails framework. This framework aided us in developing highly modular and reusable code due to the MVC architecture and Rails' powerful web application stack. With the additional help of great documentation and a strong community, we were able to develop a functional product within weeks, despite minimal domain expertise. While our current application has noticeable areas for improvement, our robust model, well thought out API, and extensive front-end templates promise a maintainable foundation for future development.

2 Project Plan

2.1 Design Goals

2.2 Language Choice

stuf stuf stuf

2.3 Timeline

stuf stuf stuf

2.4 High-Level API

stuf stuf stuf

3 Design Review

3.1 Status

stuf stuf stuf

3.2 Design

As a result of utilizing the Ruby on Rails framework, we separated our program into four distinct large scale sections: the Model, View, Controller, and Database. A browser would be used to

3.3 Alternate Designs

stuf stuf stuf

4 Next Steps

stuf stuf stuf