Software Design Specification

for

LETU Previewer Website

Version 1.0

Prepared by

|  |  |  |
| --- | --- | --- |
|  |  | Colin Tompkins |
|  |  | Jacob Swehla |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| Instructor: | Dr. Brent Baas |
| Course: | Software Engineering I |
| Date: | 3/18/2016 |

**Table of Contents**

[1 Introduction 1](#_Toc446097208)

[1.1 Document Purpose 1](#_Toc446097209)

[1.2 Intended Audience 1](#_Toc446097210)

[1.3 Definitions, Acronyms and Abbreviations 1](#_Toc446097211)

[1.4 References and Acknowledgments 1](#_Toc446097212)

[2 System Overview 1](#_Toc446097213)

[3 Design Considerations 2](#_Toc446097214)

[3.1 Assumptions and Dependencies 2](#_Toc446097215)

[3.1.1 Related Software 2](#_Toc446097216)

[3.1.2 Operating Systems 2](#_Toc446097217)

[3.1.3 End-user Characteristics 2](#_Toc446097218)

[3.1.4 Possible Changes in Functionality 2](#_Toc446097219)

[3.2 Goals and Guidelines 2](#_Toc446097220)

[4 Architectural Strategies 2](#_Toc446097221)

[4.1 MVC Architecture 2](#_Toc446097222)

[4.1.1 Model 2](#_Toc446097223)

[4.1.2 View 3](#_Toc446097224)

[4.1.3 Controller 3](#_Toc446097225)

[5 System Architecture 3](#_Toc446097226)

[5.1 Use-Case Diagram 3](#_Toc446097227)

[5.2 State Transition Diagram 4](#_Toc446097228)

[6 Detailed System Design 4](#_Toc446097229)

[6.1 Home Page 4](#_Toc446097230)

[6.1.1 Classification 4](#_Toc446097231)

[6.1.2 Definition 4](#_Toc446097232)

[6.1.3 Responsibilities 5](#_Toc446097233)

[6.1.4 Composition 5](#_Toc446097234)

[6.2 Maps Page 5](#_Toc446097235)

[6.2.1 Classification 5](#_Toc446097236)

[6.2.2 Definition 5](#_Toc446097237)

[6.2.3 Responsibilities 5](#_Toc446097238)

[6.2.4 Composition 5](#_Toc446097239)

[6.2.5 Resources 5](#_Toc446097240)

[6.3 Dining Page 5](#_Toc446097241)

[6.3.1 Classification 5](#_Toc446097242)

[6.3.2 Definition 5](#_Toc446097243)

[6.3.3 Responsibilities 6](#_Toc446097244)

[6.3.4 Composition 6](#_Toc446097245)

[6.3.5 Resources 6](#_Toc446097246)

[6.4 Schedules Page 6](#_Toc446097247)

[6.4.1 Classification 6](#_Toc446097248)

[6.4.2 Definition 6](#_Toc446097249)

[6.4.3 Responsibilities 6](#_Toc446097250)

[6.4.4 Composition 6](#_Toc446097251)

[6.4.5 Resources 6](#_Toc446097252)

[6.5 Activities Page 6](#_Toc446097253)

[6.5.1 Classification 6](#_Toc446097254)

[6.5.2 Definition 7](#_Toc446097255)

[6.5.3 Responsibilities 7](#_Toc446097256)

[6.5.4 Composition 7](#_Toc446097257)

[6.5.5 Resources 7](#_Toc446097258)

[6.6 Help Page 7](#_Toc446097259)

[6.6.1 Classification 7](#_Toc446097260)

[6.6.2 Definition 7](#_Toc446097261)

[6.6.3 Responsibilities 7](#_Toc446097262)

[6.6.4 Composition 7](#_Toc446097263)

[6.6.5 Resources 7](#_Toc446097264)

# Introduction

## Document Purpose

This document is a system design specification (SDS) for the LETU Previewer Website, a project for Software Engineering I. It is intended to comprehensively explain the design for the LETU Previewer Website. The LETU Previewer Website System Requirement Specification provides the requirements for this project.

## Intended Audience

This SDS will explain the project implementation of the LETU Previewer Website to the professor of software engineering I, project members, fellow classmates, and developers. This document could also be useful to the client, although the design details are of less interest to the client than the detailed requirements (which can be found in the accompanying SRS).

## Definitions, Acronyms and Abbreviations

The Client Mike VanBrocklin

The Product LETU Previewer Website

GUI Graphical user interface

SDS System design specification

SRS System requirements specification

JS JavaScript

LETU LeTourneau University

## References and Acknowledgments

1. TutorialsPoint (n.d.). “AngularJS MVC Architecture” [Online]. Available: http://www.tutorialspoint.com/angularjs/angularjs\_mvc\_architecture

# System Overview

The LETU Previewer Website is an interactive web-based mobile application originating from Software Engineering I. It will supplement the current system of paper maps and handouts. There is very little back-end implementation to this website, and most of the design went into to the GUI itself. This document details the design of the website’s pages and their individual elements. The website consists of six different pages, each of which has its own function. The product’s “Help” page will be self-contained and independent of any other websites or applications. The “Schedules” and “Activities” features will be dependent on the LeTourneau website, from which information will be collected and parsed. The “Maps” feature will be dependent on Google Maps. The “Dining” feature will be dependent on both Google Maps and the *Bon Appétit* website. This product will interface with the LeTourneau website, Google Maps, and the *Bon Appétit* website via the Internet.

# Design Considerations

## Assumptions and Dependencies

### Related Software

This website is constructed using the Angular JS framework, and is therefore dependent on the Angular JS libraries.

### Operating Systems

Because this software is web-based, it is not tied to any one particular operating system. The website will run on a mobile device in either the Chrome or Firefox web browser. All mobile device operating systems (e.g. Android, iOS, Windows) will support the software.

### End-user Characteristics

Two major assumptions were made about the user. Firstly, the user is assumed to have a smart phone, or another internet-capable mobile device. Secondly, the user is assumed to have access to the Internet.

### Possible Changes in Functionality

The application will be dependent on several websites, including Google Maps, the LeTourneau website, and the *Bon Appétit* website. If any of these websites are inaccessible at a given time, it could result in limited application functionality.

## Goals and Guidelines

The goal of this project is a user-friendly website that was simple to use, intuitive, and informative. Its purpose is to help guests’ visit run smoothly by bringing helpful resources to their mobile devices. It is intended to encourage visitors to return to campus by providing relevant information about the university and campus events.

# Architectural Strategies

## MVC Architecture

Because AngularJS was the language chosen to implement the product, the AngularJS MVC Architecture is the software design pattern chosen to implement the application. A Model View Controller pattern is made up of three parts.

### Model

The model is the lowest level of the pattern responsible for maintaining data. In the case of our website, the data maintained by the model includes the webpage location, any selected building, the Google Maps map, and any markers associated with the map.

### View

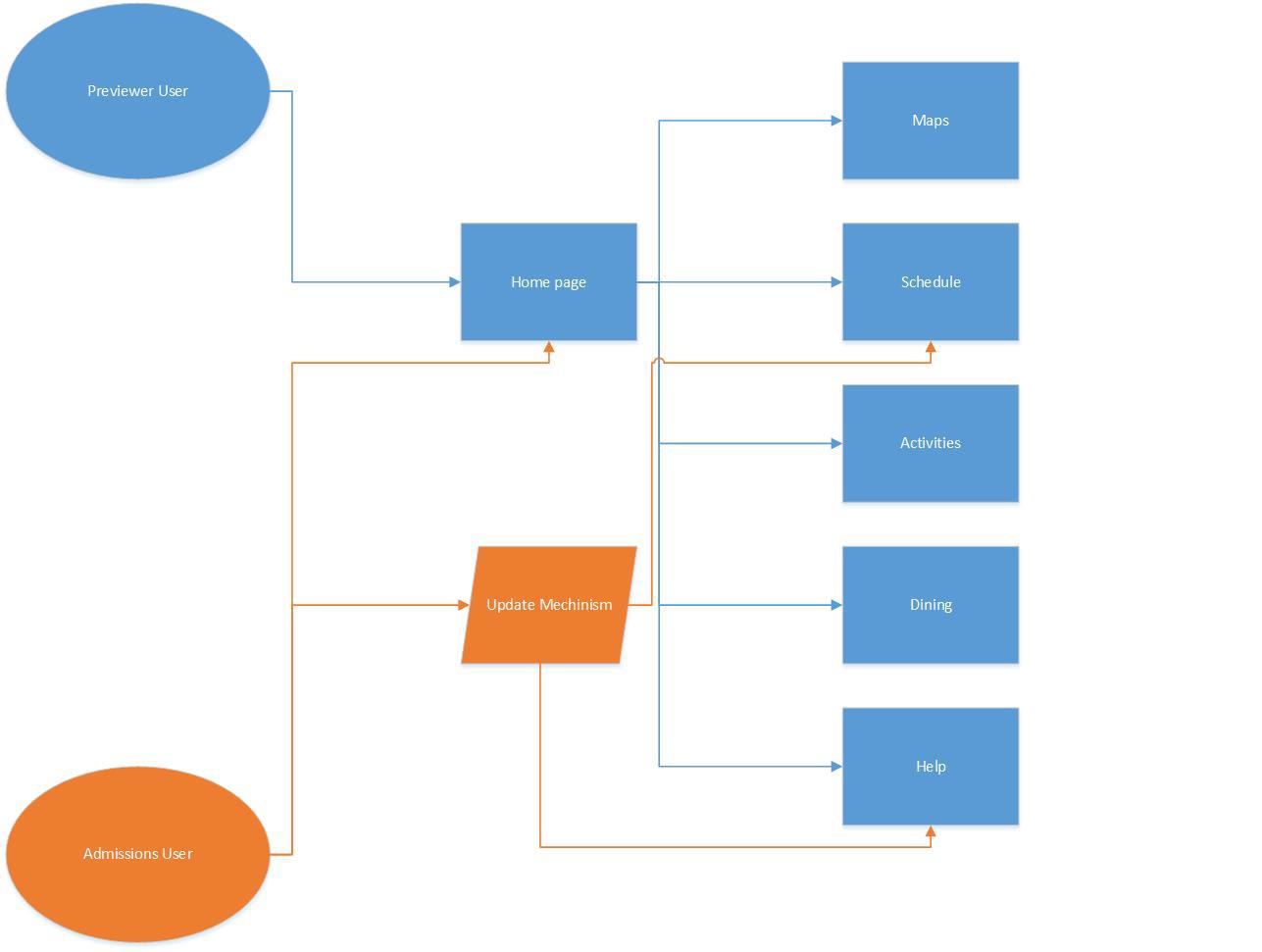
The view is responsible for displaying all or a portion of the data to the user. With our website, there are six pages that the user will view. There is the home page, the maps page, the dining page, the schedules page, the activities page, and the help page.

### Controller

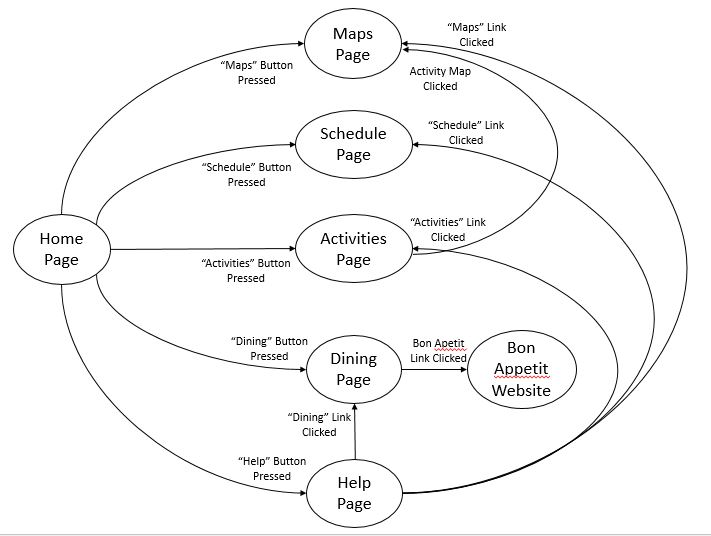
The controller is software code that controls the interactions between the Model and View. The controllers are developed using AngularJS, and there is one main controller per page. The controllers allow the transitions depicted in the State Transition Diagram (see section 5.2).

# System Architecture

## Use-Case Diagram



## State Transition Diagram



# Detailed System Design

Because this website has very little back-end implementation and because the majority of the MVC architecture focused on the View, this section will only be detailing the six different pages of the website in depth: the home page, the maps page, the dining page, the schedules page, the activities page, and the help page.

## Home Page

### Classification

The home page is a HTML file.

### Definition

The home page is the main page of the website; it will be the first thing that the user sees. It links together all the pages of website. The purpose of the home page is to provide the user with a list of the features available to them.

### Responsibilities

The home page will list the five functions available to the user. The home page will give the user the ability to navigate to any one of the other five pages.

### Composition

The home page is composed of five buttons, aligned vertically. There is one button for each of the other five pages. When one of the buttons is pressed, it will advance the user to the respective page.

## Maps Page

### Classification

The Maps Page is a HTML file.

### Definition

This page will function as a display of the location of buildings on campus and the user’s current location to the best of the location service’s knowledge. Additionally, this page will have a non-interactive map of each building. This file will be served to a browser by the server.

### Responsibilities

This page will be used to provide navigational data to the user. It will be aware of all major buildings on campus and the current location of the user.

### Composition

This page’s main feature will be to display a google maps interface. A drop down box will allow selection buildings on campus, which will snap the google map interface to the selected building. A “You Are Here” pin will also be displayed. Additionally, this page will have a non-interactive map of each building. These should be selectable via a drop down list of buildings.

### Resources

This page will use the Google Maps API to generate the map and display the location of buildings and the “You Are Here” marker.

## Dining Page

### Classification

The Dining Page is a HTML file.

### Definition

This page will provide data on dining services on campus. It will have three expandable sections. They are “Menu”, “Hours”, and “Map”.

### Responsibilities

This section will provide relevant information to dining on campus. This extends to the hours both Saga and the Hive are open, a link to the official menu page, and a Google Maps map showing the location of the relevant facilities, as well as a “You Are Here” marker.

### Composition

The expandable “Menu” component will contain a simple HTTP link to the dining services menu page. The expandable “Hours” component will have an official list of the hours of both the Hive and Saga.

### Resources

This page will use the Google Maps API to generate the map and display the location of buildings and the “You Are Here” marker. This page should also draw its contents from an editable file allowing the Admissions staff to make changes to the otherwise static information (i.e. the “Hours” information).

## Schedules Page

### Classification

The Schedules Page is a HTML file.

### Definition

The Schedules Page will list the events officially scheduled by Admissions for the preview event.

### Responsibilities

This page will serve the as an agenda to the user.

### Composition

This page will contain expandable entries for each event on the list. The “closed” state will display the event title. The “open” state will display specific information about the selected event.

### Resources

This page should draw its contents from an editable file allowing the Admissions staff to make changes to the otherwise static information.

## Activities Page

### Classification

The Activities Page is a HTML file.

### Definition

This component will list upcoming events as expandable entries. These should be filtered to one week after the current time. These entries will be drawn from the LETU website.

### Responsibilities

This page will be responsible for listing the official upcoming special events that are open to everyone on campus, including previewers.

### Composition

Each entry will be expandable to show location, time, and summery of information.

### Resources

This page will pull data directly from a source on LETU.edu.

## Help Page

### Classification

The Activities Page is a HTML file.

### Definition

The Help Page will serve as a FAQ section for previewers with questions.

### Responsibilities

This page will display relevant information decided on by Admissions staff.

### Composition

The entries will consist of expandable components. The “closed” version will show the question, while the “open” version will show the answer.

### Resources

This page should also draw its contents from an editable file allowing the Admissions staff to make changes to the otherwise static information.