

CS673S16 Software Engineering

Team 1 - FriendZone

Software Design Document

Your project Logo
here if any

<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Ed Orsini	Team Leader & Configuration Leader	<u>Ed Orsini</u>	<u>3/16/2017</u>
Cory Stone	QA Leader	<u>Cory Stone</u>	<u>3/16/2017</u>
Michael Eskowitz	Security Leader	<u>Michael Eskowitz</u>	<u>3/16/2017</u>
Robert Gomez	Environment and Integration Leader	<u>Robert Gomez</u>	<u>3/16/2017</u>
Arpita Vats	Requirement Leader	<u>Arpita Vats</u>	<u>3/16/2017</u>
Ravi K Rajendran	Design Leader	<u>Ravi K Rajendran</u>	<u>3/16/2017</u>
Nick Hattabaugh	Implementation Leader	<u>Nick Hattabaugh</u>	<u>3/16/2017</u>

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
<u>1</u>	Ed Orsini & Michael Eskowitz	<u>3/16/2017</u>	<u>Initial draft</u>

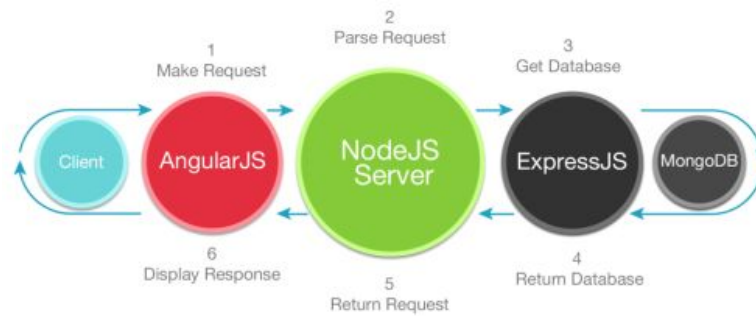
[Introduction](#)
[Software Architecture](#)
[Design Patterns](#)
[Key Algorithms](#)
[Classes and Methods](#)
[References](#)
[Glossary](#)

Introduction

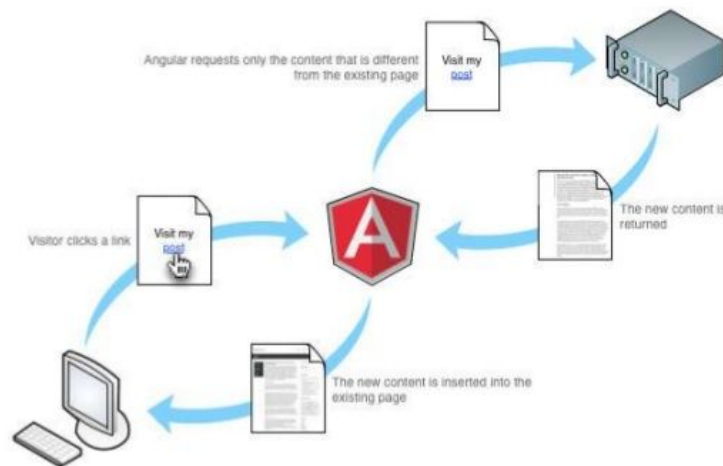
This purpose of this document is to detail the software architecture for our web application. This document will discuss the design patterns used, key algorithms, classes, methods, and other things deemed important. We have chosen to use the MEAN stack to implement our application. The MEAN stack uses the JavaScript language to develop web application. The name MEAN, itself, is an acronym for the primary components in the stack and stands for **M**ongoDB, **E**xpress, **A**ngular.js, and **N**ode.js. We have selected this particular stack as it will be beneficial and productive for all the team members to learn. The MEAN stack is very popular with an active community full of resources. Further, JavaScript is a very popular language, listed in the top 10 most popular programming languages according to the Tiobe index. Another great benefit of using the MEAN stack is that the code is written with JavaScript for the front-end, back-end, and database tiers providing us with the ability to only concentrate on knowing one language well. We agreed this was an intelligent decision allowing our team to work efficiently within the four-month time frame of the Software Engineering course.

Software Architecture

In this section, we will decompose the MEAN stack, talk about the typical user request and how it is different from the traditional HTTP request, discuss relationships between components, dependencies, and talk about how we will build our system. The diagram below presents an overview of the MEAN stack.



The basic Angular.js request is initiated by the user clicking a link or button. Angular takes the request and routes it to a particular code block for processing. At this point, it may not necessarily make a call to the back-end. Angular will only make a call to the back-end when absolutely necessary. An example request that is routed to the back end would be a database query. Even in this instance, Angular will not create full HTTP request. Instead, Angular will only request and update sections of the page (DOM) that need to be updated, resulting in higher efficiency and faster page loads.



MongoDB

MongoDB is the database component of the MEAN stack. MongoDB is a NoSQL database system which uses an unstructured manner for persisting data and does not have tables as you would find in SQL related systems. This functionality allows us to begin writing code without having completely determined all the fields and table structures within our application.

Some of the advantages of using MongoDB are:

- Schemaless
- No complex joins
- Structure of single query is clear
- Deep query-ability
- Tuning
- Ease of scale-out
- Conversion/mapping of application objects to database objects not needed
- Uses internal memory for storing the working set, enabling faster data access

Express

Express is the framework used with Node.js. The framework was developed in order to provide a common toolset to developers for creating web applications, APIs and other HTTP related features.

Some of the advantage of using Express includes:

- Routing features
- Sub routes
- Static file serving
- Compatible with many popular template engines
- View caching
- Route case sensitivity
- Compatible with third party NPM modules
- eTag support
- Useful properties and methods on the request object
- Useful methods for constructing the response object

Angular.js

Angular.js is an MVC framework developed by Google. Angular.js really is more a Model, View, and Whatever instead of a Model View Controller as the framework intends to be a solid base for Single Page Applications (SPAs). Angular.js is optimized for creating mobile friendly sites. Additionally, we are also using Bootstrap in conjunction with Angular.js. Bootstrap is a Javascript/CSS framework that allows us to create responsive sites consistently and with less effort than by doing so from scratch. Just for clarity, we are also using SASS for CSS creation which helps with optimizing CSS output and maintainability among other things. Some of the main advantages of Angular.js include two-way data binding, dependency injection, separation of concerns, testability, and abstraction.

The separation of concerns is reached by dividing the application among:

- Partials
- Controllers
- Directives
- Services
- Filters

Node.js

Node.js is the core of the back-end. Node runs Chrome's V8 engine and has non-blocking, event-driven I/O. The application is capable of handling multiple requests on a single service without blocking each other.

Some of the benefits of Node.js include:

- Asynchronous I/O
- JavaScript for the front-end and back-end
- Large community and large set of resources
- Fast

In addition to Node.js, we are using several modules to facilitate our development and tasks. These modules are considered dependencies which are managed via Node Package Manager. We manage these dependencies with a package.json file.

Bootstrap

Bootstrap is a front-end (JavaScript/CSS) framework for building responsive web applications.

Some of the benefits of using Bootstrap include:

- Ease of use
- Responsiveness
- Speed of development
- Customizable
- Consistency
- Support
- Packaged JavaScript components
- Simple Integration
- Pre-styled Components

Oauth

Oauth is an open standard for token-based authentication and authorization on the internet. We will be using Oauth to authenticate our users via Facebook. Our users will be able to login using the same username and password that they have on Facebook. The benefit to leveraging what Oauth provides is that our users can avoid creating yet another username and password. Also by using Oauth, our application will not need to save session information on the server, and will be able to easily set the expiry date in the token itself.

Design Patterns

The main design patterns that we will use are the patterns associated with the MVC architecture provided by Angular.js. According to the Gang of Four, MVC is nothing more than a combination of Strategy, Composite, and Observer patterns. Implicitly, we are using these patterns since they are associated with MVC.

Some of the patterns that Angular.js uses within our software system include the following:

For Services, Angular.js uses *Singleton, Factory, Decorator, Facade, Proxy, Active Record, and Intercepting Filters*.

For Directives, Angular.js uses *Composite, Strategy, Observer, Interpreter, and Template View*.

For Scope, Angular.js uses *Observer, Chain of Responsibilities, and Command*.

For Controllers, Angular.js uses *Page Controller, Module, and Data Mapper*. According to the Gang of Four, MVC is nothing more than a combination of Strategy, Composite, and Observer patterns.

Key Algorithms

checkAuthenticated()

This function will check if the user is logged in throughout the application. This check will be made in order to ensure that the user has the proper privileges before performing an action. Some examples of the actions include posting a message, uploading an image, posting a comment, chatting with others, and more. This function will check if a valid token exists. If the token does exist, the actions will be performed. Otherwise, the user will not be able to perform the actions.

Classes and Methods

Every main feature of the application will have its own model(s), view, and controller.

PictureModel

(Picture)
title: String,
originalname: String,
filename: String,
path: String,
destination: String,
size: String,
mimetype: String,
User: mongoose.Schema.ObjectId

PictureController (front-end)

getAllPictures()
removePicture()
setPicture()
uploadPicture()

PictureController (back-end)

get()
post()

PictureView

Angular Template (HTML)

ChatModel

ChatController (front-end)

ChatController (back-end)

ChatView

OptionsModel

(Profile)

user: mongoose.Schema.ObjectId,
username: String,
firstname: String,
lastname: String,
gender: String,
birthday: String,
country: String

OptionsController (front-end)

getProfile()
saveProfile()

OptionsController (back-end)

get()
post()

OptionsView

Angular Template (HTML)

WallModel

(Wall)

postId: String,
postMsg: String,
user: mongoose.Schema.ObjectId,
comments: mongoose.Schema.ObjectId,
dateAndTime: String

(PostComment)

commentMsg: String,
user: mongoose.Schema.ObjectId
dateAndTime: String,

WallController (front-end)

getPosts()
makePost()

postComment()

WallController (back-end)

get()
post()

WallView

Angular Template (HTML)

AuthModel

(User)
email: String,
pwd: String,
facebook: String

AuthController (front-end)

register()
login()
authenticate()

AuthController (back-end)

register()
login()
facebook()
createToken()

AuthView

Angular Template (HTML)

References

<https://www.sitepoint.com/introduction-to-mean-stack/>

<http://www.tiobe.com/tiobe-index/>

<https://github.com/mgechev/angularjs-in-patterns>

<http://searchmicroservices.techtarget.com/definition/OAuth>

https://www.tutorialspoint.com/mongodb/mongodb_advantages.htm

<http://stackoverflow.com/questions/28823253/what-are-the-benefits-of-express-over-plain-node-js>

<https://dzone.com/articles/what-are-benefits-nodejs>

<https://www.devsaran.com/blog/10-best-reasons-use-bootstrap-amazing-web-designs>

