WASHINGTON STATE UNIVERSITY
VANCOUVER

CS 320 Course Project Final Report

for

MEMEit

Prepared by

Group Name: MEMEit

Ian Kieswether	11594545	ian.kieswether@wsu.edu
Logan Nelson	11577923	logan.nelson@wsu.edu
Yekaalo Habtemichael	11585389	y.habtemichael@wsu.edu

Date: December 13, 2019

Contents

1 Introduction

1.1 Project Overview

Our project is to design and develop a multi-page web application. For our team we focused on developing a cloud based storage system for memes and other images. The project utilizes a landing page which links to our web application page which uses the Meteor web app. framework with several modules/libraries added to increase its functunality like Materialize. The system stores data both client-side and server-side using MongoDB as our database system and stores the data in collections, which behave similarly to dictionaries.

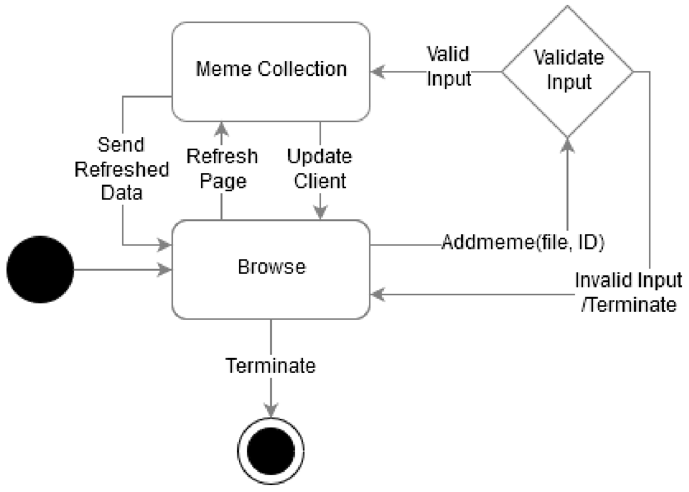
1.2 Definitions, Acronyms and Abbreviations

App – Abbreviation for Application
Database - An organized collection of data, accessible through functions by the website
Landing Page - A webpage where the user first arrives when accessing our website.
Meme - For our purposes, a meme is a picture with text that conveys humor to the viewer if they have the appropriate knowledge and context for the humor trying to be conveyed.

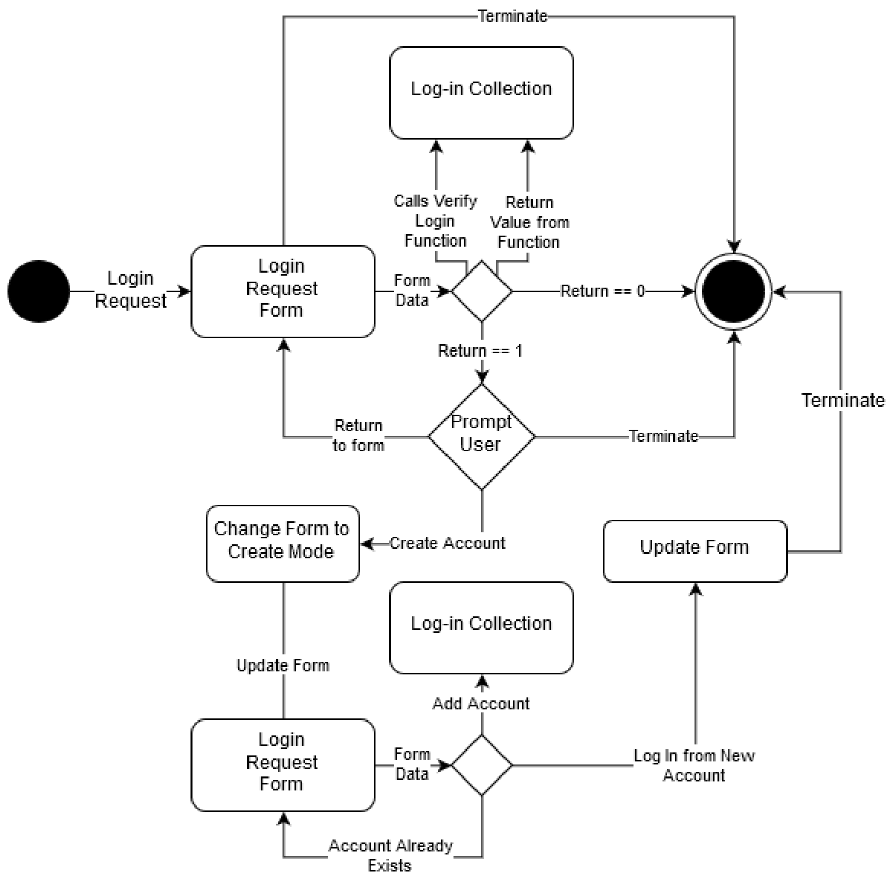
2 Design

2.1 System Modeling

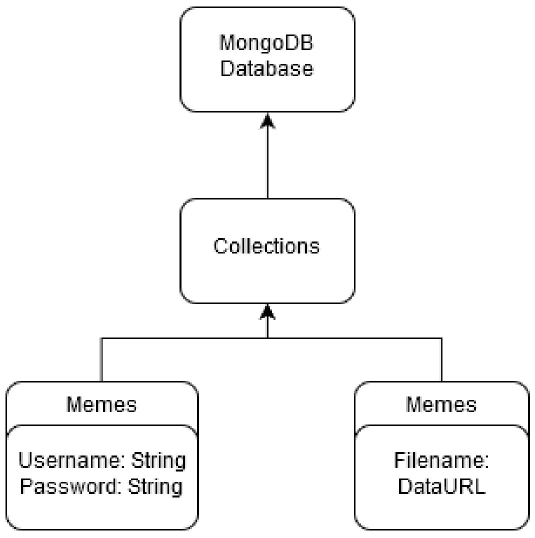
Behavior Diagrams



Adding Meme to Database (Collection)
Fig. 1



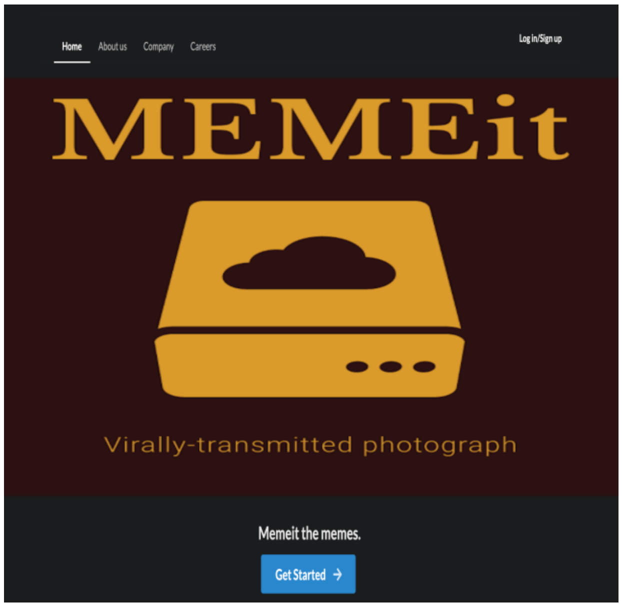
Log-in and/or adding Account to Database (Collection)
Fig. 2

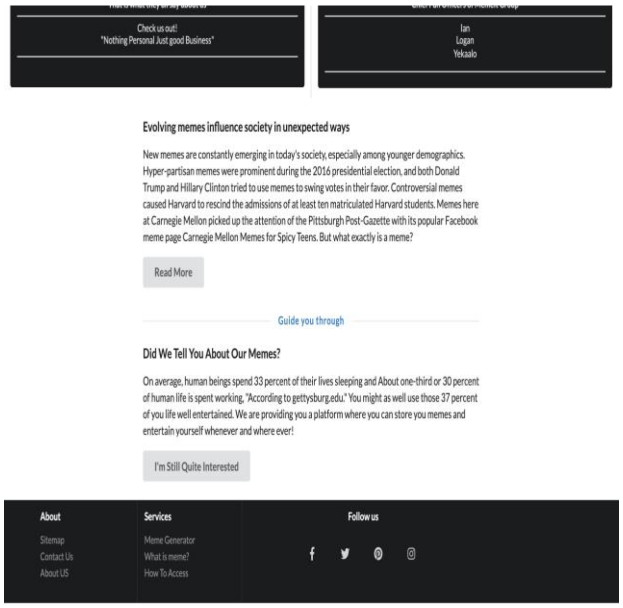


Class Diagram
Fig. 3

2.2 Interface Design

Landing Page Screenshots




Check us out!
"Nothing Personal Just good Business"

Ian
Logan
Yekaalo

Evolving memes influence society in unexpected ways

New memes are constantly emerging in today's society, especially among younger demographics. Hyper-partisan memes were prominent during the 2016 presidential election, and both Donald Trump and Hillary Clinton tried to use memes to swing votes in their favor. Controversial memes caused Harvard to rescind the admissions of at least ten matriculated Harvard students. Memes here at Carnegie Mellon picked up the attention of the Pittsburgh Post-Gazette with its popular Facebook meme page Carnegie Mellon Memes for Spicy Teens. But what exactly is a meme?

Read More

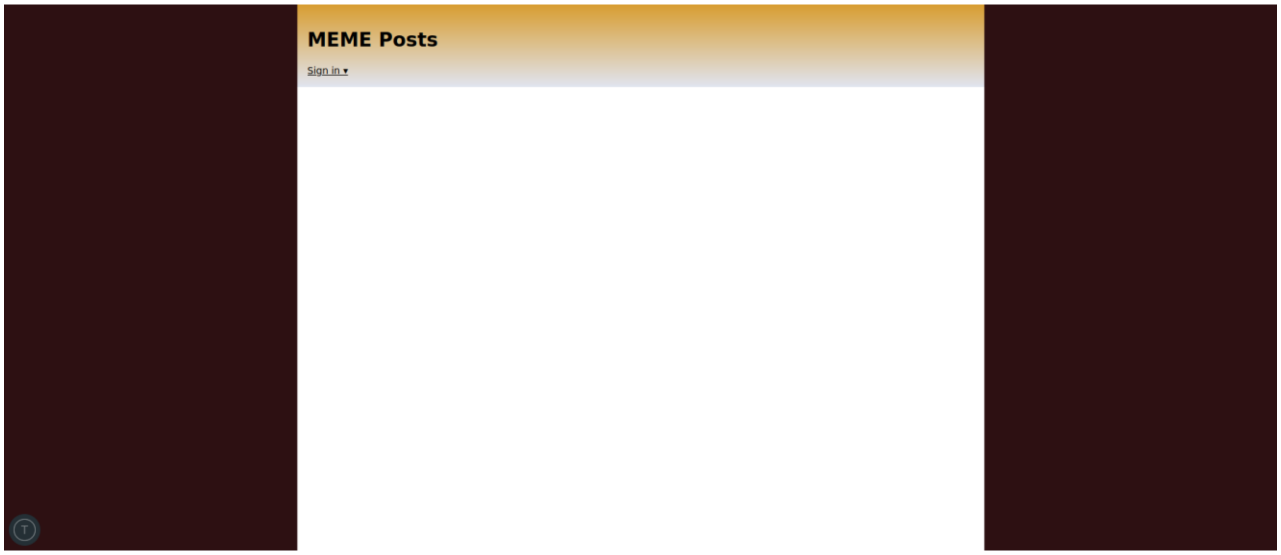
Guide you through

Did We Tell You About Our Memes?

On average, human beings spend 33 percent of their lives sleeping and About one-third or 30 percent of human life is spent working, "According to gettysburg.edu." You might as well use those 37 percent of you life well entertained. We are providing you a platform where you can store you memes and entertain yourself whenever and where ever!

I'm Still Quite Interested

Application Screenshots



3 Implementation

3.1 Development Environment

Framework: Meteor Web Framework

Framework Modules: Blaze, Semantic UI

Programming Languages: Javascript, CSS, HTML 5

IDEs: IntelliJ IDEA Ultimate Edition

3.2 Task Distribution

Yekaalo Habtemichael

- HTML Page Designer
- HTML Stylist

Ian Kieswether

- Front-End Implementation of Systems (Login)
- Connecting HTML to Front-End

Logan Nelson

- Back-End Implementation of Systems (Storage)
- Connecting HTML to Front-End and Front-End to Back-End

3.3 Challenges

Managing Multiple CSS Files in the Meteor Framework

- Meteor merges all CSS files into one big one.
    - To fix we had to change CSS classes to be unique to each HTML File.

Proper Utilization of Git

- Difficulty Utilizing multiple branches

4 Testing

4.1 Testing Plan

Testing for Login

Date: Decemeber 6th

- Wrong username and wrong password
- Correct username with wrong password / Wrong username with correct password
- Using both correct username and password
- Creating an account with same username as one exists
- Creating an account with same password as another account
- Logging out

Testing for Memes

Date: Decemeber 9th

- Adding jpgs / pngs / gifs
- Adding incorrect files
- Removing Memes

4.2 Tests for Functional Requirements

We tested the Login system following our plan in Section 4.1 and our results are as shown:

- Did not allow for login using incorrect credentials (Passed)
- Did allow for login with correct credentials (Passed)
- Did not allow for account creation using existing username(Passed)
- Did allow for account creation if user didn’t already exist (Passed)
- Logged out successfully (Passed)

We tested the Meme storage system following our plan in Section 4.1 and our results are as shown:

- Accepted URLs (Passed)
- Did reject incorrecty inputs (passed)
- Did remove memes from database if user had proper authorization (Passed)

4.3 Tests for Non-functional Requirements

- We found that we got the memory to persist in the database for logins and for the meme/image uploads.
- We tested for instability by letting the app run repeated trials without turning off or shutting down. There was some instability issues due to using conflicting libraries. If we were able to do this again we would shrink our current dependancy lists to provide better reliability.
- The app works fast and was responsive as long as their was no instability from the about issue.

4.4 Hardware and Software Requirements

- Using different hardware for the system, like testing it on Logan's Linux OS system and on Ian or Yekaalo's Mac OS systems.
- We also test on using the App on Chrome and Firefox. The result of that test was that both worked while chrome was better suited for the application.

5 Analysis

On each milestone, each person spent probably on the mid end of 10+ hours dealing with issues from meteor, html/css merges, getting the html documents to work with each other without changing styles, and learning new meteor functionality to advance the project. In total there were about 70+ hours put into this project throughout the entire semester. The most difficult milestone was definitely base line implementation. This is due to the various libraries we had to use and learn as well as learning the syntax and some minute details of the web app framework Meteor. Meteor operates differently than anything we had utilized before, as it would merge files into one large html file at startup which became a nightmare for the front-end implementation.

6 Conclusion

Overall, we learned a lot about the Meteor framework, the way Meteor operated and how html and javascript interact to produce a web app. We also learned how to create a system that accept user input in the form of URL's from a text field and using that URL to display an image in a news feed like setup as well as how to accommodate multiple users within a single web app.

Appendix A - Group Log

We met in person at least once a week and usually more often than that. We worked over discord calls and maintained communcations over group text chat. The team communication was somewhat effective because we always left knowing what our next step was in developing our product and a good view of our target finished product.