**CSC 394 - Team 5**

**Requirements Analysis Document**

Nick Bashor

Patrick Grzegorczyk

Wesam Jabali

Joey Navarro

Kevin Sass

Luigi Yebra

**Table of Contents**

1. **Introduction**
   1. Purpose of the system
   2. Scope of the system
   3. Objectives and success criteria of the project
   4. Definitions, acronyms, and abbreviations
   5. References
   6. Overview
2. **Current system**
3. **Proposed system**
   1. **Overview**
   2. **Functional requirements**
      1. Student users
      2. Faculty users
      3. All users
   3. **Nonfunctional requirements**
      1. Usability
      2. Reliability
      3. Performance
      4. Supportability
      5. Implementation
      6. Interface
      7. Packaging
      8. Legal
   4. **System models**
      1. Scenarios
      2. Use case model
      3. Analysis object model
      4. Dynamic model
      5. User interface, navigational paths, and screen mockups
4. **Glossary**

**Introduction**

1. **Purpose of the system**

The system under development is meant to provide a communicative and collaborative interface in the form of a web application for usage by students and faculty in the context of an academic setting. The system must facilitate project management, project completion, monitoring of progress, and visualization of project timelines.

1. **Scope of the system**

The system is intended to be made available to all students enrolled in, as well as all faculty employed by, a singular academic institution. The system must manage an arbitrary number of users (both students and faculty members) as well as an arbitrary number of academic courses.

1. **Objectives and success criteria of the project**
   1. **Objectives**
      1. Implement team communication interface(s)
      2. Implement visualizations for relevant data analytics
      3. Implement gamification techniques associated with tasks
      4. Implement essential user management functions
      5. Implement an efficient database schema
   2. **Success criteria**
      1. Visualizations are easy to understand
      2. Application uptime is continuous and reliable
      3. Response time exhibits minimal latency
      4. Application can scale up to handle larger numbers
      5. User interface is intuitive and accessible
2. **Definitions, acronyms, and abbreviations**
   1. **Definitions**
      1. Refer to the **Glossary** for more information
   2. **Acronyms**
      1. **CSS** - Cascading Style Sheets
      2. **DBMS** - Database Management System
      3. **GUI** - Graphical User Interface
      4. **JS** - JavaScript
      5. **NPM** - Node Package Manager
      6. **SQL** - Structured Query Language
   3. **Abbreviations**
      1. *Not applicable*
3. **References**
   1. *Not applicable.* Further research still needs to be done at this point.
4. **Overview**
   1. **Function of system**
      1. Student and faculty users can create accounts
      2. Faculty can create course listings, manage course material, and administer grades
      3. Students can enroll in courses, submit assignments, and track their overall academic progress
      4. All users can communicate with each other via chat rooms and discussion boards
   2. **Reason for development**
      1. Educational institutions worldwide have seen a large increase in the usage of digitally-administered academic management tools
      2. We likewise aim to develop an online web application that implements common features of such digital academic aids, with the hopes of providing a clean user experience coupled with a sufficiently robust data management infrastructure

**Current system**

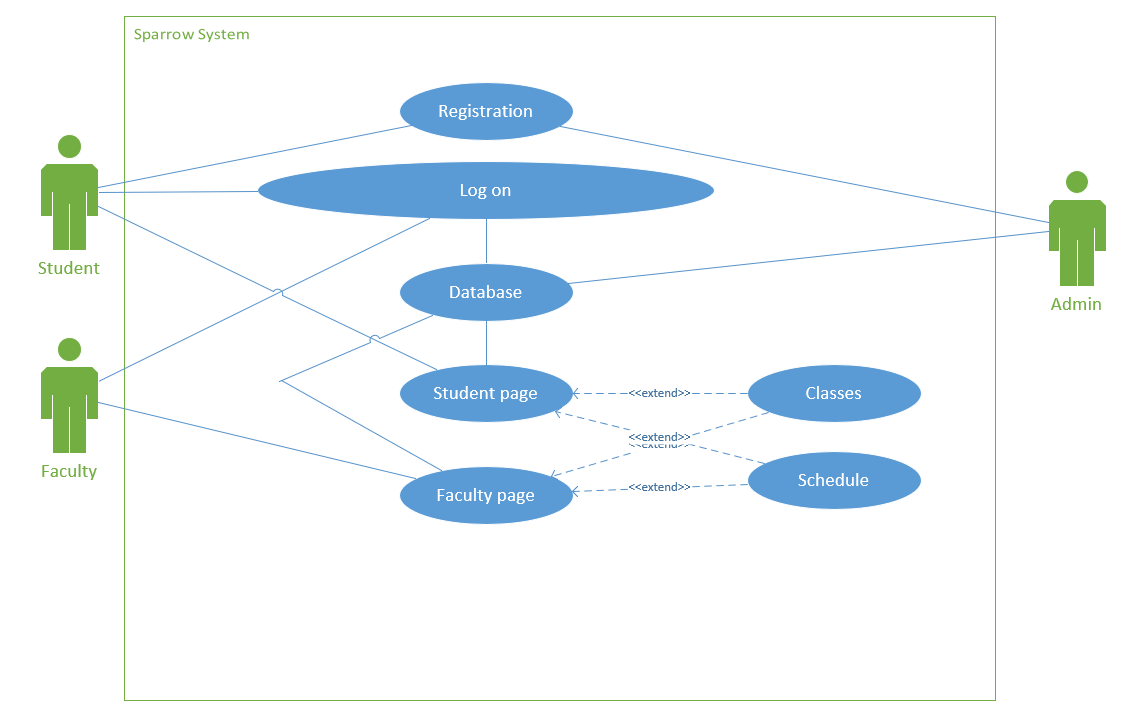
Frontend is currently operational. (Implemented with beta Vuetify 3 using Vue 3). Specifically, the login and registration page are both currently in progress. We are presently experimenting with the Bootstrap framework.

**Proposed system**

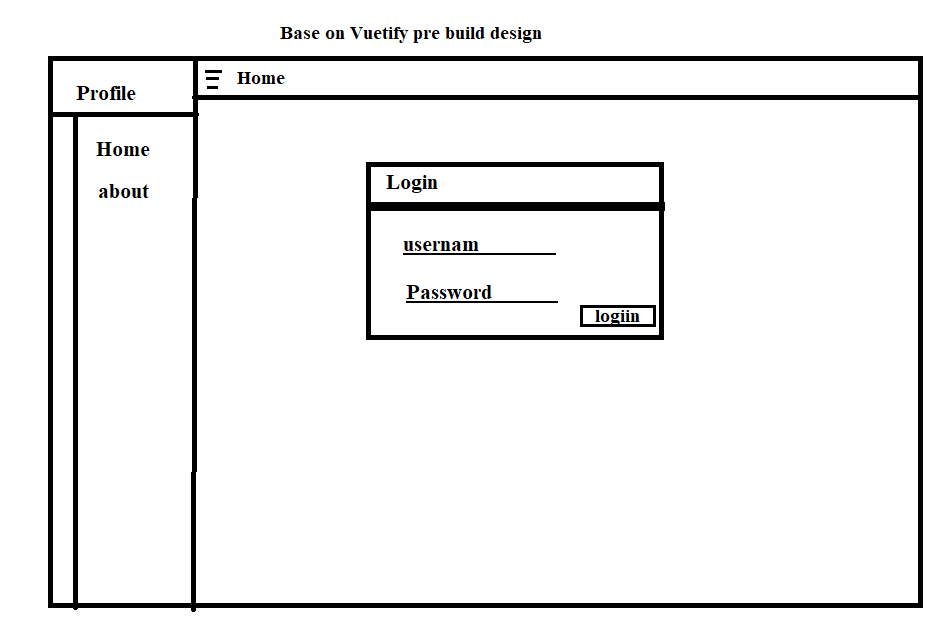
1. **Overview**

The system under development is intended to be a tool for usage by educational institutions to facilitate project management for and collaboration between students and faculty members.

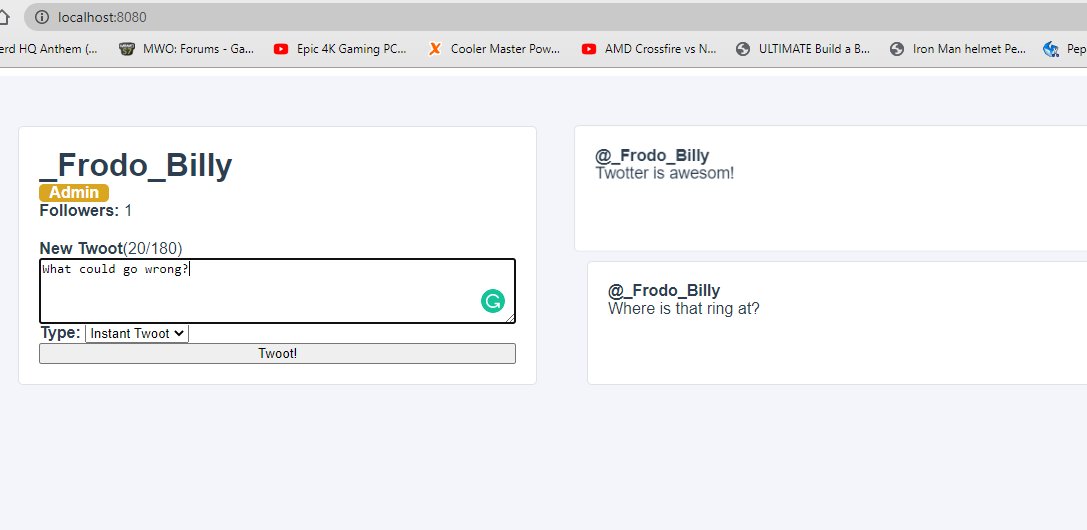
1. **Functional requirements**
   1. **Student users**
      1. Easy addition to class roster(s)
      2. Responsibility/agreement opt-in functionality
      3. Password modification and authentication
      4. Restricted view according to privileges
   2. **Faculty users**
      1. Class section creation
      2. Class content creation
      3. Team creation and modification
      4. Student activity monitoring
      5. Grade administration and access privileges
   3. **All users**
      1. Account creation
      2. Account deletion
      3. Login and logout functionality
      4. Responsibility/agreement opt-in functionality
      5. Password modification and authentication
      6. Security
      7. Communication
2. **Nonfunctional requirements**
   1. **Usability**
      1. Effective visualizations will be implemented to facilitate users’ understanding of relevant information
      2. Essential functionality will be made as simple as possible to use for both student and faculty users
   2. **Reliability**
      1. System will be available at least 95% of the time
      2. System will be able to recover in a timely and safe manner from potential crashes and/or unexpected downtime
   3. **Performance**
      1. System can scale up to accommodate a large quantity of users and/or courses if need be
      2. System will respond to user interactions in a timely manner
   4. **Supportability**
      1. Internet Explorer v7 and above
      2. Microsoft Edge
      3. Mozilla Firefox
      4. Google Chrome
   5. **Implementation**
      1. Vue.JS (Vue 3)
      2. Vuetify
      3. Express.JS or NestJS
      4. PostgreSQL
      5. Knex.JS
   6. **Interface**
      1. User interface is intuitive and straightforward to navigate
      2. Basic features must not be concealed from users
      3. Advanced settings must be partitioned away and well-formatted in order to prevent cognitive overload
      4. Relevant information must not be concealed from users
      5. Important prompts/reminders should be made explicitly visible
      6. Accessibility must be taken into consideration
   7. **Packaging**
      1. *Not applicable.*
   8. **Legal**
      1. *Not applicable.*
3. **System models**
   1. **Scenarios**
4. Future students have to register via home page. An option for registration will be provided in the home screen.
5. In case of forgotten password we will try to implement a return password via email and Captcha.
6. Faculty needs to register with admin(not sure)
7. Class and Schedule interface will be extended to the pages such as student page and faculty page via components of vue. More interfaces might come.
8. Admin can register Faculty and Students.
9. Admin can change the tags of each user.
10. Still research, mitigate sql injection and Cross-Site Scripting.
    1. **Use case model(In progress)**

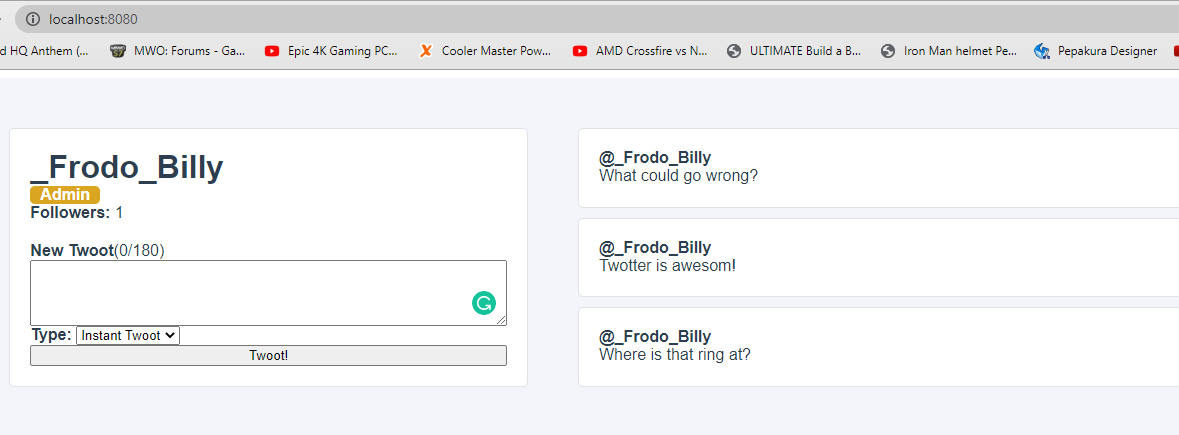
****

* 1. **Analysis object model**
     1. *Not applicable.* Analysis activity has not yet gone underway.
  2. **Dynamic model**
     1. *Not applicable.* Analysis activity has not yet gone underway.
  3. **User interface, navigational paths, and screen mockups**



Some test page that used css(This design base on tutorial of userpage twitter clone)





**Glossary**

**accessibility** - a frontend design concern that addresses usability in the context of differently-abled users of the system

**account** - a data entity that encapsulates all identifying information associated with a single and unique user of the system

**browser** - a software application that allows the user to access the Internet; more specifically, in the context of this system, the means by which a user will interact with this project

**backend** - developmental entities related to the system’s internal logic, such as maintaining, processing, and retrieving records; not meant to be accessible by the user

**faculty** - a user associated with an employee of the university (or similar academic institution) that administers this system; a non-student, usually bearing elevated privileges (see **student**)

**frontend** - developmental entities related to the system’s outward-facing logic, such as chat, account management, and course management interfaces; the portion of the system with which the user interacts

**gamification** - an appealing, enticing implementation of cosmetic incentives to aid users in meeting deadlines, completing submissions, and accomplishing other related menial tasks

**GUI** - acronym for graphical user interface; see **interface**

**interface** - the graphical and textual components of the system with which the user interacts, ultimately bridging external input with internal logic; see **frontend**

**login** - the functionality that enables a user to establish a unique online session via the system, during which they may access their account, view coursework-related materials, etc.

**logout** - the functionality that enables a user to safely and securely terminate an online session that was previously established via the login process

**opt-in** - a method of enabling users to access optional services provided by the system; by default, a user will not be subscribed to this form of service without their explicit and informed consent

**password** - the cryptographic means by which a user may securely verify their identity in order to log into their account

**performance** - a functional metric that measures how efficiently user queries are transmitted and processed by the system, typically in terms of units of time

**reliability** - a functional metric that evaluates how consistently and continuously the system is available to provide its services, typically expressed as a percentage of uptime

**roster** - a list of all student and faculty users who bear access to a single given course

**schema** - the formal organizational structure behind the system’s internal representation of data entities

**student** - a user associated with an individual enrolled in the university (or similar academic institution) who primarily takes courses but does not instruct them; bears limited privileges (see **faculty**)

**supportability** - a functional metric that measures the range of compatibility the system has across modern Internet browsers; see **browser**

**system** - the project currently under development that this document describes; a comprehensive, user-oriented project management tool, collaborative engine, and general-purpose academic aid

**usability** - a functional metric that measures how intuitive or unintuitive the user experience can be evaluated when said user interacts with the system