CST-451 Capstone Project Requirements Document

Patrick Garcia

Grand Canyon University

Instructor: Professor Brandon Bass

3/14/2022

**ABSTRACT**

As we move forward, we must discuss the specific requirements of this project. Sticking with the conceptual for a moment, user stories are provided that describe problems the average user may have. These stories are used to mold the application to solve problems that potential users face. If these problems are sufficiently handled, the user base may increase and remain retained. These are our functional requirements and outline what the application should accomplish. This is followed by non-functional requirements. User stories that describe what the application should be, as opposed to what its functions should accomplish. Dictating what the application should be, will allow developers to optimize every aspect of the application. Technical requirements follow, outlining the specific technical aspects of the project. The chosen programming language and IDE are provided in this section, as well as more user stories, specifically tailored towards technical requirements.

Design diagrams are provided to provide the reader with a clear visual representation of the flow of data within our application. As well as mock diagrams providing a rough estimate of what the average user will experience when using the app. These diagrams allow us to slowly move from the conceptual, towards the concrete. Outlining the layout of particular views allows developers to be able to hone in to their artistic vision, and begin to shape a flow for the application functions.

| History and Signoff Sheet |
| --- |

**Change Record**

| **Date** | **Author** | **Revision Notes** |
| --- | --- | --- |
| 3/10/2022 | Patrick Garcia | Initial draft for review/discussion |
| 6/28/2022 | Patrick Garcia | Updated Sections: *User Interface Design, Logical System Design* |
|  |  |  |

| **Overall Instructor Feedback/Comments** |
| --- |

| **Overall Instructor Feedback/Comments** |
| --- |

**Integrated Instructor Feedback into Project Documentation**

☐ Yes ☐ No

**TABLE OF CONTENTS**

**Functional Requirements 4**

**Non-Functional Requirements 5**

**Technical Requirements 6**

**Logical System Design 7**

**User Interface Design 8**

**Reports Design 9**

**Functional Requirements**

**User Stories**

Some text narrative based user stories are provided below. These are included to help the reader gain a greater understanding of the application. They are written from the point of view of a potential user for the app. Outlining a user's potential problem, and how our application is able to solve said problem.

**School is Back in Session**

In this step, the school season has begun, and students have found themselves with a range of courses and assignments. Many students will opt for their normal organizational methods, others may not.

“As a student, I need a solution that helps me organize my classes and their associated assignments.”

“As a student who travels, I need an organizational solution that I can take with me wherever I go.”

“As a student who lives with ADHD, I need an organizational tool that is easy and streamlined, and that will provide automatic updates to keep me organized, even when I forget to check the application itself.”

“As a disorganized student, my grades, and mental health, have suffered.”

“As a student, my current class is referencing information learned in a previous class, but I cannot find my old assignment files and notes.”

**Registering an Account**

In this step, students have discovered our application and have decided to create an account. Expectations have already been set in their minds, which must be met.

“As a student, I want to easily be able to upload my course information into the application during the sign up process, so I won't have to go back and do it later.”

“As a student who worries about my digital security, I want two-factor authentication to keep my class schedule and course files accessible only by myself.”

“As a student with many online accounts, I want the sign-up process to be quick and easy.”

**Getting started with the Application**

In this step, students have downloaded the app and registered an account. They must now get familiar with our app, and begin to explore its functionalities. This step is essential for user retention. If the user does not like what they see in this step, they will not return.

“As a student, I want a quick application tour to get me familiar with the app, its functions, and its organizational methods.”

“As a student who is not tech savvy, I want a clear and simple dashboard and UI.”

“As a student, I want to see my assignment schedule in a calendar view for planning ahead, and in a list view for my immediate assignments.”

**Non-Functional Requirements**

**Non-Functional User Stories**

Some text narrative based Non-Functional user stories are provided below. These are included to help the reader gain a greater understanding of the application. They are written from the point of view of the system itself, simply stating what the application must be.

“The application shall be Accessible”

“The application shall be Portable”

“The application shall be Secure”

“The application shall be Effective”

“The application shall be Real-Time”

“The application shall be Open Source”

**Technical Requirements**

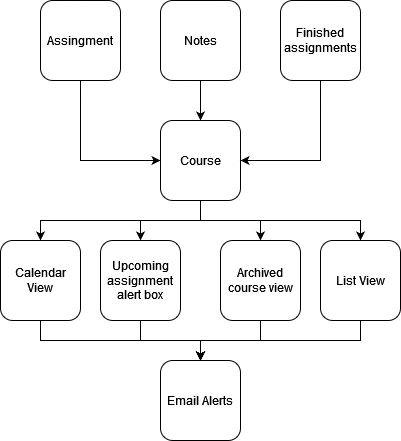
**Expected Technologies**

* C#
* SQL Database
* Visual Studio
* MySQL Workbench
* ASP.NET

**Use Cases**

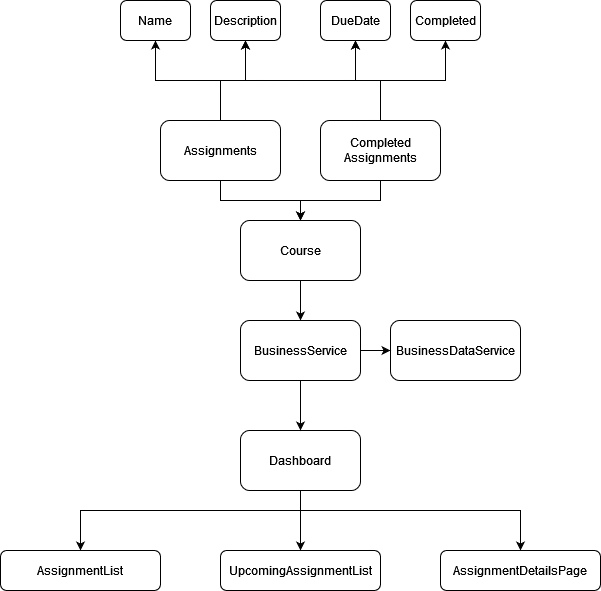
* **Privacy**
  + Two-factor authentication will be used to ensure students' accounts are kept secure.
  + Inputted data will be cleansed to prevent any injection style attacks. Validation rules and C#’s string builders will be useful for this purpose.
  + C# authentication controllers and two factor authentications features will be leveraged for this purpose.
  + Application routes should be protected, to ensure users only have access to pages that pertain to them.
* **Performance**
  + The application must boast quick page load times, and must populate student data using a database quickly. SQL database will be used for this purpose,
* **Accessibility**
  + Using SASS, HTML and CSS, text contrast will be increased, colors will be used effectively to indicate actions, text will be properly spaced, and controls will be large and easily identifiable, all in compliance with Web Content Accessibility Guidelines (WCAG).
  + App should be accessible without an internet connection.
* **Productivity**
  + Users will be able gain all needed course information from a single screen.
  + Application will automatically create directories and subdirectories needed to hold Course and Assignment files.
* **Human Error**
  + Application will automatically determine if course or assignment start/end dates make sense. For example, if a user enters a start date of January 1st 2022, and an end date of January 2nd 2021, the system will alert the user of the discrepancy.

**Logical System Design**



**EDITS**

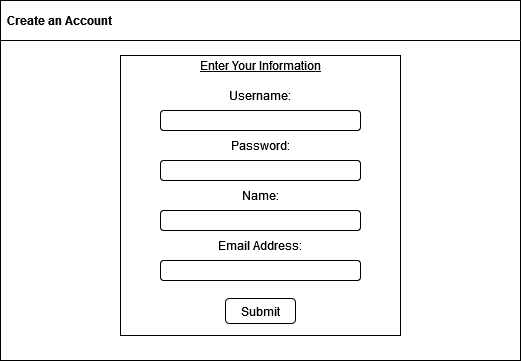
Finalized Logical System Design

****

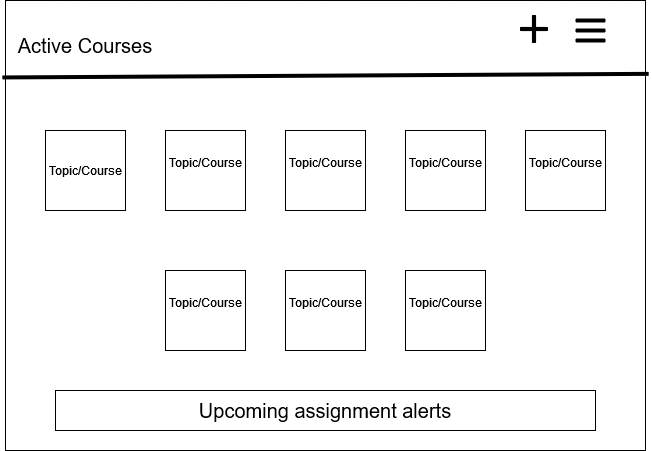
**User Interface Design**

Templates for user views are shown below. These will be refined in later iterations of the project, and are subject to change.

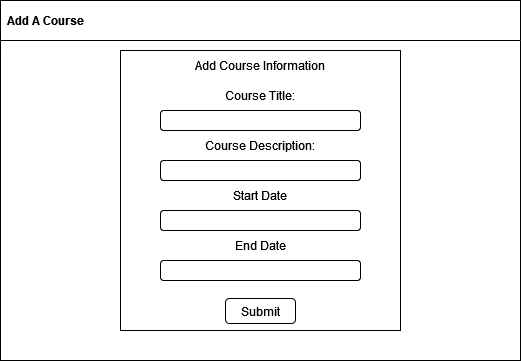
**Registration Page**



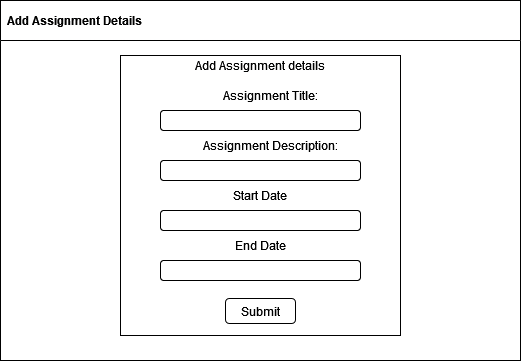
**Dashboard**



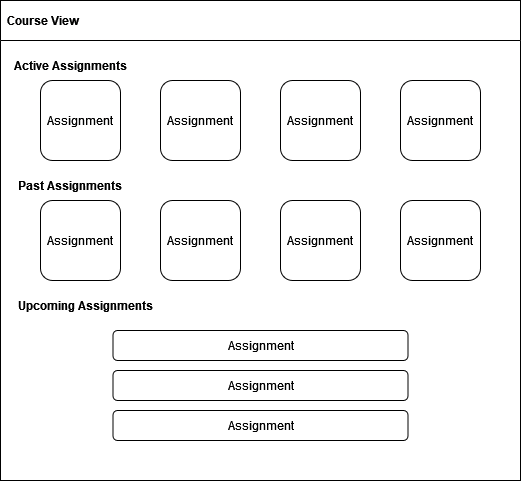
**Add Course**



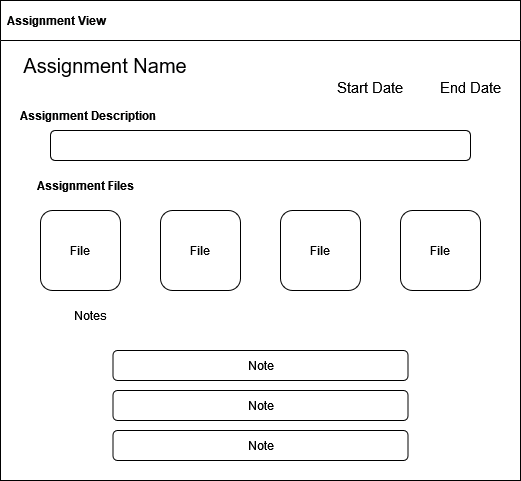
**Add Assignment**



**Course View**

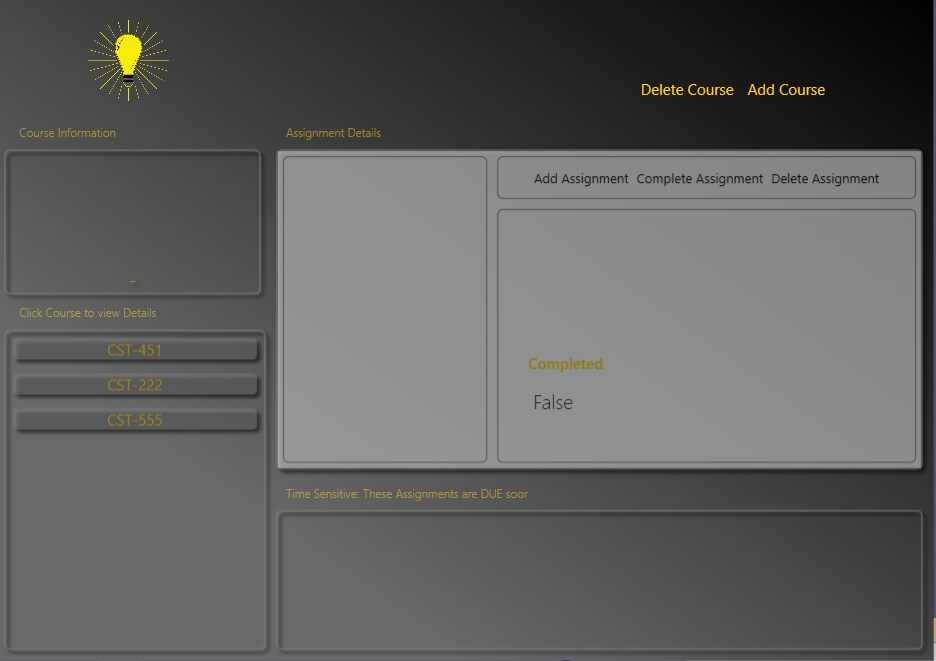


**Assignment View**

****

**EDITS**

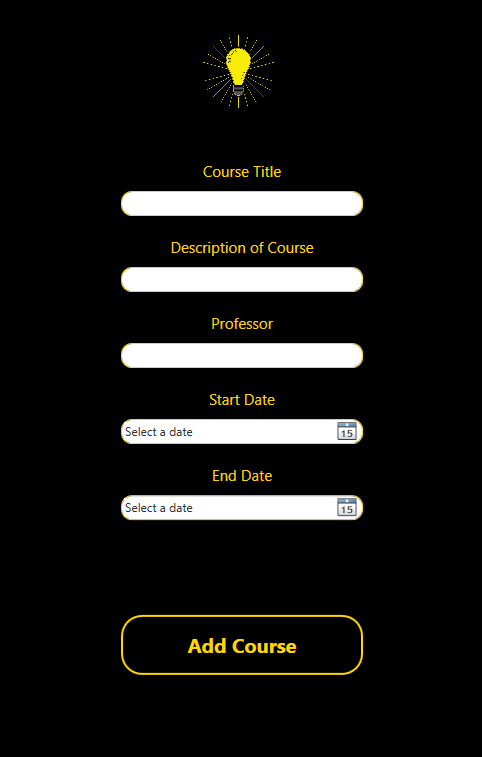
**Finalized Dashboard UI Design**

****

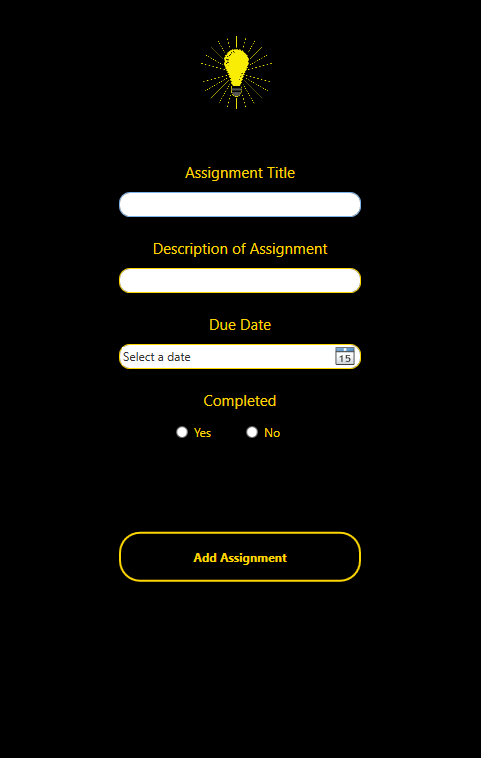
Dashboard populated with course information



**Finalized Add Course Dialog UI Design**

****

**Finalized Add Assignment Dialog UI Design**

****

**Reports Design**

**EDITS**

* JSON Files
  + This app utilizes JSON files to store course and assignment information for later use. The app uses the JSON encoding functions provided with C# to encode Course/Assignment object data into a JSON String. The app then utilizes the *File* C# class to read/write the object data from the JSON file. An example screenshot of this file is included below.

