**CSCI 213 – Software Development using Frameworks (Spring 2024)**

**Learning outcomes:**

At the end of the Final project, students should be able to demonstrate:

1. an ability to identify, formulate, and solve complex engineering problems by applying engineering, science, and mathematics principles.
2. an ability to communicate effectively with a range of audiences.
3. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
4. an ability to function effectively on a team whose members provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.
5. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to conclude.
6. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

**Team Project Instructions**

***Overview:*** A vital task in this course is to collaboratively work on a team project to plan requirements, develop design, and implement a web-based server-side application using:

1. [Jira Software](https://www.atlassian.com/software/jira/templates/bug-report) - a software development tool for project management and issue tracking.

1. [Software Requirements Specification (SRS)](https://www.geeksforgeeks.org/software-requirement-specification-srs-format/) - outlines the requirements for a software system. It serves as a contract between the stakeholders and the development team, detailing what the software will do, design constraints, and how it will be used. Note: template is provided.
2. [Lucidchart](https://www.lucidchart.com/pages) software - is a web-based diagramming application that enables users to create a variety of diagrams and charts, including flowcharts, organizational charts, wireframes, and mind maps. The platform supports collaboration, allowing multiple users to work on a document simultaneously and share feedback in real-time.
3. [GitHub](https://github.com/) - is the leading platform for software development, collaboration, and security.
4. [Visual Studio 2022 Community](https://visualstudio.microsoft.com/free-developer-offers/) - is a powerful and versatile integrated development environment (IDE) for software developers and teams.
5. [Slack](https://slack.com/) - Slack is a collaboration and communication platform designed to facilitate teamwork among individuals and groups. It allows users to send messages, share files, and integrate with various tools and services. Slack supports channels for organized discussions, direct messaging for private conversations, and features like video calls and workflow automation to enhance productivity.

**The Final project will be divided into *three phases*: Requirements(5%), Design(5%), and Implementation(5%).**

Each phase has specific deliverables and guidelines to ensure a structured approach to project development.

***Project Description:***

The Proposal phase/ Pre-phase.

The proposal is the initial phase, where teams outline the project concept, goals, and high-level plan.

**Deliverable:** The team leader should submit a Word document with the following information:

• Team Information: Include the team leader's name and the names and emails of all members.

• Project Title: Provide a clear, concise title for your project.

• Project Objectives: Define the primary goals your project aims to achieve.

• Project Scope: Outline the project's boundaries, including what is and is omitted.

• Project Description: Briefly describe the project, its purpose, and the problem or opportunity it addresses.

Project Requirements phase.

This phase involves defining, documenting, and managing software requirements. It consists of gathering stakeholder needs, analyzing requirements for clarity and feasibility, specifying them in detail, validating them against user needs, and managing changes to the requirements throughout the software development lifecycle.

**Deliverable:** The team leader should submit the Software Requirements Specification (SRS) document, the hyperlink link to the Jira project account, and screenshots of the Jira product backlog to the Blackboard learning system. Please use the provided SRS document template.

* 1. Scrum Project Management**:** Implement Scrum project using Jira for task management. The “Team leader” should create a Jira Scrum project and assign tasks to team members with deadlines.
  2. The plan schedule should include a task breakdown (user stories, design tasks, implementation tasks, etc.).
  3. Use of Slack for communication.
  4. Submit regular updates to Jira with task progress, including screenshots.

**3. Project Design**

**Objective:** Develop a comprehensive design plan for the web application, including UI/UX design, architecture, and the relational database structure (ERD diagram).

For this web project, you will use a 3-tier web architecture. There are three layers/tiers:

* Presentation (client) layer
* Application (business) layer
* Data access layer

Presentation/Client Layer

When we talk about the presentation layer, we refer to the front end of an app. This layer includes such elements as static content and dynamic interface that are visible to end users. The environment of this layer is any browser. Among the technologies that are used in this case, we can name HTML, CSS, or JavaScript. The potential frameworks to choose from include Angular, React, and Vue.

Business/Application Layer

Speaking of the business layer, which is also referred to as the application layer, is a part of the app back-end. The web app's back-end determines business logic and responses to browser requests that are sent to the presentation layer. It consists of the core application logic and outlines all internal flow for data and requests. In this case, servers, serverless cloud platforms, or PaaS are the most convenient environments. Among the programming languages that are used in this case, we can name C#, JavaScript, Java, Python, and PHP. The potential back-end frameworks to choose from include ASP.NET, express.js, nest.js, Spring, Flask, Django, and Symfony.

Data Access Layer

The final layer in this architecture is the data access layer, which plays a crucial role by closely interfacing with the business layer, retrieving essential information from the servers. Simultaneously, the data service layer acts as an intermediary, effectively segregating the business logic from the client side during request processing.

It is an app back-end part that contains databases and DBMS (database management systems) that collect, manage, and store data. The environment can be the same as the application layer. Among the Database management systems, we can name PostgreSQL, Microsoft SQL Server, MySQL, MongoDB, Cloud Offerings, and so on.

All the layers work independently and communicate with each other through the relevant components.

**Format:** Submit a professional design document with clear sections, diagrams, and visual elements.

**4. Project Implementation**

**Objective:** Build the web application according to the design plan, ensuring all functionalities are implemented and integrated properly.

**Deliverables:**

* **Version Control:** Use GitHub for version control. All code must be committed regularly, with meaningful commit messages. The repository should be well-organized with a clear README file.
* **Task Management:** Continue using Jira to track progress, moving tasks through the stages of progress, review, and completion. Document progress with regular screenshots.
* **Development Process:** Implement the frontend using Angular and Bootstrap, connect to the backend built in Django, and integrate with Google Firebase for data storage. REST APIs should be developed to enable communication between frontend and backend.
* **Testing and Debugging:** Conduct thorough testing of the application, document any issues, and fix bugs. Ensure all functionalities work as expected across different browsers and devices.
  1. **Final Submission:** Prepare a final report that includes: The GitHub repository URL.
  2. Screenshots of the Jira board throughout the project.
  3. A reflection on the development process, challenges faced, and lessons learned.
  4. **Project Demo:** Each team will demo their project, showcasing all features, explaining the architecture, and reflecting on the project process.

**Format:** Submit the final report as a PDF, including all required elements.

**Note:** Throughout the project, ensure effective communication within the team using Slack, and maintain a consistent, collaborative effort. The project will be graded on technical implementation, documentation quality, project management, and overall presentation

1. University Attendance Management System –

This system will help the students sign attendance, professors determine when they have lectures, and pull attendance records straight from the system. Professors will know the students in their classes etc.

1. E-Commerce WebApp –

This web app will be able to display items and their respective pictures. These items will

also display prices and can be filtered by name, type, or price. The website will hold several different types/ categories of items The web app will use a database that will hold item information, such as price, name, stock, and ID number. The database will also hold customer information such as name, address, email, and customer ID number. There will also be a table for purchases that will make use of item ID and customer ID. Anyone can browse the website, however, to purchase items one must create an account. Customers will also be able to view their shopping cart.

1. Ice Cream Shop –

Our team intends to bring The Ice Cream Shop to the modern world. In creating a web presence, we aid in solving an advertising problem by allowing visitors a central place to access information about the shop. The application will reduce wait times by presenting a menu to the customer before they place their order. It allows for grab-and-go service by allowing customers to pre-order their dessert and select a pickup time.

1. Budget Tracking Application –

This application would allow users to help budget their finances by tracking their expenses, and forms of income, to help them have a better grasp of their financial situation. The application would allow the user to add different types of expenses, such as one-time fees, or reoccurring bills. The user would also be able to add distinct types of income, depending on the type, or how often, the use.

1. Inventory System

The proposed web application will facilitate the tracking of stocks by users. The user authentication will be handled either through Auth0 or manual authentication, and the application will maintain two database tables to store user information and stock portfolio data. To search for a stock, the user can input a stock name, and the application will retrieve relevant data through polygon.io API. The retrieved data will be graphically represented using chart.js and displayed to the user. The user information table will store data provided by Auth0, and the portfolio table will store the names of stocks selected by the user.

1. Workflow Management –

This professional application provides tools for an enterprise to assign and carry out workflows. Administrators can manage user groups and create a hierarchy of user management among the users in which each user has their login. Managers can assign and

create workflows for their teams. Workflows are created using templates and can be modified at any point of the workflow’s life cycle. Users can add notes and progress updates to the workflows to provide constant updates to the project. Upon completion of a workflow, they are stored as a completed workflow which can be accessed in the workflow history section. Users can easily view their workflow due dates by using the deadline calendar.

1. Fitness Tracker –

This app allows for easier use of fitness programs and allows users to feel more confident in their exercise routines, and their healthy lifestyles. Tracks by day, week, and month, Calories consumed with fat, fiber, and sugar, Calories burned by exercise type and length, and hydration tracking. Generate graphs for weight, calories, hydration, and calories burned.

1. **Weather App –**

The application will display the live temperature to the user in addition to providing the user with weather forecasts, precipitation predictions, and daily high/low temperatures. The application will also display any current weather advisories that include the campus (i.e., Wind Chill Advisory, Tornado Watch, etc.).

1. Organizer App –

The project will be an application that is built specifically for college students to organize their classwork and personal lives. The application will include daily weekly, and monthly calendars and a to-do list with the ability to categorize by class or otherwise. The student could add and delete events and tasks, which have the option of displaying on the calendar, to-do list, or both.

1. Scrum Management Tool –

We are looking to create a simple scrum management tool where users will be able to log.

into their dashboards. The dashboard will have options to see what has happened when the user is gone and options to investigate the different projects, they are a part of. When in a project interface the user can communicate with their team using different chat channels. They can also create tasks that have priority and discretion in what needs to be done. Each task will also have its thread. Team members can check tasks during a sprint and mark tasks as completed.

1. Sports/eSports Statistics Tool –

This web application will allow the user to take live statistics for a sports or esports event. It will include options for various sports, which could include but is not limited to: Hockey, Football, Basketball, Rocket Leauge, Valorant, League of Legends.