# **CSC154 Project Sprint 3 Overview**

**Objectives**

* Implement the project High-level Design based on the project requirements gathered in Software Requirements Specification Version 1 in project Sprint 2.
* Complete and submit the Software Requirements Specification Version 2.

**Schedule**

* 10/21/2024 – 11/03/2024

**Readings and References**

* Chapter 1, 2, 3, 4, 5, 6, 19 in the textbook and supplemental lesson.
* Documents and videos posted in References.
* Templates posted in Project Sprint 3:
  + Software\_Requirements\_Specification\_Template\_v2.docx
  + Weekly\_Report\_Template.docx

**Requirements and Tasks**

1. **10/21/2024 – 10/27/2024 ------- Specify and** **define the contents in High-level Design and create the draft of Software Requirements Specification version 2**

**Project manager** willorganize the team activities to discuss the contents in project High-level Design and complete the draft of Software Requirements Specification (SRS) version 2:

* Review the Software Requirements Specification (SRS) version 1 created in Sprint 2.
* Review all documents posted in “Group Project Sprint 3 Week 1” at Black Board.
* Discuss and define the following components in project High-level Design:
  + **Fatima  
    Security** (Required) - key components that ensure the system is secure from potential threats.
    - Specify the security design for **at** **least two** of the following aspects:
      * Security Architecture Overview
      * Authentication and Authorization
      * Data Protection
      * Logging and Monitoring
      * Secure Software Development Lifecycle (SDLC) Integration
      * Compliance and Regulatory Considerations
      * Disaster Recovery and Incident Response
      * User and Role Management
      * Security Testing Strategy
  + **James  
    Hardware** (Required) – the physical components that ensure the software system can function as expected.
    - Specify the hardware design for **at least three** of the following aspects:
      * Processor (CPU) Requirements
      * Memory (RAM) Requirements
      * Storage (Disk Space) Requirements
      * Graphics Processing Unit (GPU) Requirements
      * Network Requirements
      * Peripheral Devices
      * Operating System and Platform Requirements
  + **Katelyn**
  + **User Experience** (Required) – it’s essential to ensure the product is user-friendly and meets end-user needs.
    - Describe how the user interacts with the application, such as:
      * Usability Requirements
      * User Flow and Navigation
      * Responsiveness and Performance
      * Feedback and Error Handling
      * User Profiles and Personalization
    - Use a free ***User Experience Design Software*** to create **wireframe** for each user interface.
  + **Fatima**
  + **Architecture** (Required) – the components that help to define the structure, interaction, and constraints of the system.
    - Describe how an application’s pieces fit together at a high level in one or more of the following architectures:
      * Layered (N-Tier) Architecture
      * Client-Server Architecture
      * Microservices Architecture
      * Monolithic Architecture
      * Event-Driven Architecture
      * Service-Oriented Architecture (SOA)
      * Serverless Architecture
      * Component-Based Architecture
  + **James**
  + **Database** (Required) – it’s essential to ensure the application’s data handling is well-structured, efficient, and scalable.
    - Specify what Database Management System could be used to store and manage the data for the project.
    - Use a free ***Database Diagram Design Tool*** to design the **ERD** (Entity Relationship Diagram) of the database schema.
  + **Raegan**
  + **Top-level Classes** (Required) – they represent the main building blocks of the software system. These classes encapsulate the core functionality and form the basis of the overall system architecture. They serve as a bridge between the requirements and the detailed design phase.
    - Design the major top-level classes and their components.
    - Use a free ***Class Diagram Tool*** to draw the **Class Diagram** showing classes, their attributes, operations, and relationships between them.
  + **Sha  
    Data Flow and States** (Required) – they are crucial for understanding how the system will operate and interact with users or other systems.
    - Design and show the “flow” through different processes or states in the application.
    - Use a free ***Data Flow Diagram Tool*** to create the **Data Flow Diagram**.
  + **Katelyn + James  
    Reports** (Required) - outline the reporting capabilities of the system and define what data should be available and how it will be presented.
    - Select **at least one** type of the following reports the project will create:
      * Customer reports
      * Sales Reports
      * User Activity Reports
      * Financial Reports
      * Inventory Reports
  + Sha  
    Internal Interfaces (Optional)
    - Specify how pieces of the application will interact.
  + Raegan  
    External Interfaces (Optional)
    - Specify how the application will interact with other applications and resources.
  + James  
    Other Outputs (Optional)
    - Specify the possible outputs generated by the project:
      * Printouts
      * Web pages
      * Data files
      * Image files
      * Audio files
      * Video files
      * E-mail/text
  + Katelyn  
    Configuration Data (Optional)
    - Specify what data and parameters will be configurable in the configuration page or file.
  + Fatima  
    Training (Optional)
    - Describe the training plan for the user:
      * In-person or online
      * Printed materials
      * Instructional videos
      * Online documentation
* Create the **draft** of Software Requirements Specification (SRS) version 2 using the posted **template**.
* Create the Weekly Project Progress Status Report and submit it to Blackboard for grading.

1. **10/28/2024 – 11/03/2024 ------ Finalize the Software Requirements Specification (SRS) version 2**

* **Project Manager** organizes team activities to review the draft of Software Requirements Specification (SRS) version 2.
* **Project Manager** organizes team activities to finalize the Software Requirements Specification (SRS) version 2.
* Submit the final version of Software Requirements Specification (SRS) version 2 to Blackboard for grading.
* Create and submit the Weekly Project Progress Status Report to Blackboard for credits.

**Deliverables**

* Weekly Project Progress Status Report for the week 10/21/2024 – 10/27/2024.
  + Due date at Black board: 11:59 pm on 10/27/2024.
* Weekly Project Progress Status Report for the week 10/28/2024 – 11/03/2024.
  + Due date at Black board: 11:59 pm on 11/03/2024.
* Final version of Software Requirements Specification (SRS) version 2
  + Due date at Black board: 11:59 pm on 11/03/2024.

**Submissions**

|  |  |  |
| --- | --- | --- |
| **Submitter** | **Document** | **Due Date** |
| Project manager | Week 1 Progress Status Report | 10/27/2024 |
| Project manager | Week 2 Progress Status Report | 11/03/2024 |
| Project manager | Final version of Software Requirements Specification (SRS) version 2 | 11/03/2024 |

**Grading Rubric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Unacceptable** | **Needs Improvement** | **Good** | **Excellent** |
| **Weekly Progress Report (100 pts)** | The report is not submitted. (0 pts) | The report is not completed or does not meet all requirements. (1-60 pts) | The report is mostly completed and generally meets the requirements. (61 – 99 pts) | The report is completed and meets all requirements. (100 pts) |
| **Software Requirements Specifications version 2 (100 pts)** | The document is not submitted. (0pts) | The document is not completed or does not meet all requirements. (1 – 60 pts) | The document is mostly completed and generally meets the requirements. (61 – 99 pts) | The document is completed and meets all requirements. (100 pts) |