

|  |  |
| --- | --- |
| **Project title** | **Project Proposal** |
| **Course Code** | **DSE211/03** |
| **Course Name** | **Application Development Capstone** |
| **Project Start date** |  |
| **Project Submission Date** |  |

As part of your capstone project, you and your team (three students) are required to develop a web or mobile application. The first step in this process is to create a comprehensive project proposal based on the principles learned in IU1: Project Planning and Requirements Gathering. This proposal will serve as a roadmap for your project, outlining the scope, objectives, user requirements, timeline, and collaborative tools that will be used throughout the development process.

**Proposal Components**

**1. Project Scope and Objectives**

* **1.1 Understanding Project Scope:**
  + **Definition:** Clearly define the boundaries of your project. What are the specific goals, deliverables, tasks, and deadlines? Include the functionalities and features that your application will offer, as well as the technologies you plan to use.
  + **Activity:** Write a detailed scope statement for your project. This should include:
    - A description of the application you will develop (web or mobile).
    - The primary purpose and target audience of the application.
    - The key features and functionalities that will be included in the application.
    - Any constraints or limitations that may impact the development process.
* **1.2 Defining Project Objectives:**
  + **Definition:** Your project objectives should be specific, measurable, achievable, relevant, and time-bound (SMART). These objectives will guide the development process and help you evaluate the success of your project.
  + **Activity:** Define at least three SMART objectives for your project. For example:
    - "Develop a mobile application that allows users to track their fitness activities with 95% accuracy within six months."
    - "Create a web application that supports 500 concurrent users with a response time of less than 2 seconds."
    - "Implement a user-friendly interface that scores at least 80% in usability testing."

**2. User Requirements and Expectations**

* **2.1 Gathering User Requirements:**
  + **Methods:** Choose appropriate methods for gathering user requirements. This could include conducting interviews with potential users, distributing surveys, organizing focus groups, or observing users in their natural environment.
  + **Activity:** Use at least two of the methods mentioned above to gather user requirements. Document the findings and summarize the key user needs and expectations for your application.
* **2.2 Documenting User Requirements:**
  + **Use Cases:** Create use cases that describe how users will interact with your application. Each use case should outline a specific scenario in which a user interacts with your application to achieve a goal.
  + **User Stories:** Write user stories that capture the functionality from the user's perspective. Format: "As a [type of user], I want [an action] so that [a benefit]."
  + **Activity:** Develop at least three use cases and three user stories for your application.
* **2.3 Managing Expectations:**
  + **Communication:** Plan how you will keep stakeholders informed about the project’s progress and how you will manage their expectations. This includes regular updates, meetings, and transparent communication.
  + **Activity:** Outline your communication plan, including the frequency of updates, the medium (e.g., email, meetings), and the key stakeholders involved.

**3. Project Timeline and Milestones**

* **3.1 Developing a Project Timeline:**
  + **Tools:** Use tools like Gantt charts or project management software (e.g., Trello, Asana) to create a project timeline that includes all the key tasks and milestones.
  + **Activity:** Create a Gantt chart for your project. Identify and plot all major tasks, including research, design, development, testing, and deployment.
* **3.2 Setting Milestones:**
  + **Definition:** Milestones are significant checkpoints in the project timeline that mark the completion of major phases or tasks.
  + **Activity:** Identify at least five key milestones for your project and explain their significance. For example:
    - "Complete the requirements gathering phase by Week 3."
    - "Finalize the application design by Week 6."
    - "Deploy the beta version of the application by Week 12."

**4. Collaborative Development Tools and Environments**

* **4.1 Introduction to Collaborative Tools:**
  + **Tools:** Identify the collaborative tools your team will use for version control, project management, and communication (e.g., Git for version control, Trello for task management, Slack for communication).
  + **Activity:** Choose and set up the collaborative tools your team will use. Explain why these tools were chosen and how they will facilitate collaboration.
* **4.2 Setting Up the Environment:**
  + **Development Environment:** Describe the development environment you will set up for your team. This includes the IDEs, coding standards, and version control practices that will be used.
  + **Activity:** Outline your team’s development environment setup. Include details on the tools, languages, and frameworks that will be used, as well as any coding standards that need to be followed.

**Proposal Submission**

* **Deadline:** [Insert Deadline Date]
* **Format:** Submit the project proposal in a PDF format. The proposal should be well-organized, clearly written, and free of any errors.
* **Evaluation Criteria:** Your proposal will be evaluated based on the clarity of the project scope, the relevance of the objectives, the thoroughness of the user requirements, the feasibility of the project timeline, and the appropriateness of the collaborative tools chosen.

**Project Proposal Rubric**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **5 - Excellent** | **4 - Good** | **3 - Satisfactory** | **2 - Needs Improvement** | **1 - Unsatisfactory** |
| **Project Scope and Objectives** | Scope and objectives are exceptionally clear, well-defined, and align perfectly with the project goals. SMART objectives are specific, measurable, achievable, relevant, and time-bound. | Scope and objectives are clear and well-defined. SMART objectives are mostly specific and relevant. | Scope and objectives are somewhat clear, but may lack full specificity or relevance. SMART objectives are adequate but may need refinement. | Scope and objectives are vague or lack detail. SMART objectives are partially defined but require significant improvement. | Scope and objectives are unclear or missing. SMART objectives are poorly defined or absent. |
| **User Requirements and Expectations** | Comprehensive and detailed user requirements are gathered using multiple appropriate methods. User stories and use cases are clearly documented and thoroughly address user needs. | Detailed user requirements are gathered using appropriate methods. User stories and use cases are clear and mostly address user needs. | User requirements are gathered, but may lack depth or use limited methods. User stories and use cases are somewhat clear, but may not fully address user needs. | User requirements are incomplete or gathered using inadequate methods. User stories and use cases are unclear or only partially address user needs. | User requirements are missing or poorly gathered. User stories and use cases are unclear, incomplete, or missing. |
| **Project Timeline and Milestones** | Project timeline is detailed, realistic, and includes well-defined milestones. All key tasks are appropriately scheduled, with dependencies clearly outlined. | Project timeline is clear and realistic, with most key tasks appropriately scheduled. Milestones are well-defined but may need minor adjustments. | Project timeline is included but may lack detail or realism. Milestones are present but may not be fully defined or realistic. | Project timeline is vague, incomplete, or unrealistic. Milestones are poorly defined or missing key elements. | Project timeline is missing or unrealistic. Milestones are absent or irrelevant to the project. |
| **Collaborative Tools and Environments** | Collaborative tools are carefully chosen, fully set up, and well-integrated into the project workflow. The development environment is clearly defined and conducive to efficient teamwork. | Collaborative tools are well-chosen and mostly set up. The development environment is defined and supports teamwork, with minor gaps. | Collaborative tools are chosen, but setup may be incomplete or not fully integrated. The development environment is somewhat defined but may lack clarity. | Collaborative tools are inadequately chosen or poorly set up. The development environment is vague or not fully supportive of teamwork. | Collaborative tools and development environment are poorly chosen, not set up, or missing altogether. |
| **Clarity and Professionalism** | The proposal is exceptionally well-organized, clearly written, and free of errors. It presents the project in a professional and convincing manner. | The proposal is well-organized, clearly written, with minor errors. It presents the project in a professional manner. | The proposal is adequately organized and written, but may contain some errors or lack of professionalism. | The proposal is poorly organized, with significant errors or lack of clarity. Professionalism is lacking. | The proposal is disorganized, difficult to read, with numerous errors and a lack of professionalism. |