### 

### 

### Request 1: Detailed Project Description (20%)

#### **Project Name**: Student Management System (SMS) for FU

#### Project Overview:

* **What**: Develop a web-based Student Management System (SMS) to automate student-related tasks such as registration, attendance tracking, grade management, and communication.
* **Why**: To streamline administrative tasks, reduce manual errors, and enhance access to data for students, faculty, and staff.
* **How**: Using agile development methodology. Technologies include Angular (front-end), .NET Core (back-end), and SQL Server (database). The process involves gathering requirements, system design, development, testing, and deployment.

#### Key Characteristics:

* **Purpose/Objectives**:
  + Automate student record management.
  + Improve communication among students, staff, and faculty.
  + Enhance accuracy and accessibility of student data.
* **Timeframe**: 6 months
* **Customer/Sponsor**: FU IT Department and Registrar’s Office.
* **Project Milestones**:
  + **Requirements Gathering** (4 weeks)
  + **System Design** (4 weeks)
  + **Development Completion** (8 weeks)
  + **Testing and Quality Assurance** (4 weeks)
  + **System Deployment** (4 weeks)
* **Success Criteria**:
  + Fully functional system meeting all requirements.
  + Successful user acceptance from staff, students, and faculty.
  + On-time and on-budget deployment.
  + 30% reduction in processing time for student-related administrative tasks.
* **Project Constraints**:
  + **Cost**: $50,000
  + **Resources**: 1 Project Manager, 4 Developers, 2 Testers, 1 Database Administrator, 1 UI/UX Designer.
  + **Technology Constraints**: Must integrate with existing university IT infrastructure.

### 

### Request 2: Responsibility Assignment Matrix (RAM) (20%)

| **Work Package/Activity** | **Project Manager** | **Developers** | **Testers** | **UI/UX Designer** | **Database Admin** |
| --- | --- | --- | --- | --- | --- |
| 1. Requirement Gathering | A | R | I | C | I |
| 2. System Design | A | R | C | R | C |
| 3. Database Design | I | C | I | C | R |
| 4. Front-End Development | I | R | I | R | I |
| 5. Back-End Development | I | R | I | I | C |
| 6. Integration Testing | I | C | R | I | I |
| 7. User Acceptance Testing | C | I | R | C | I |
| 8. Deployment | A | R | I | I | R |
| 9. User Training | R | I | I | I | I |
| 10. Post-Deployment Support | A | I | I | I | C |

**Legend**:

* **R** = Responsible
* **A** = Accountable
* **C** = Consult
* **I** = Inform

### 

### Request 3: Quality Assurance and Control Activities (20%)

1. **Code Review (QA)**
   * **Purpose**: Ensure adherence to coding standards and identify issues early.
   * **When**: During development, before merging code.
   * **How**: Peer review of code using a checklist to detect issues.
2. **Unit Testing (QC)**
   * **Purpose**: Verify that individual components work as intended.
   * **When**: After each module is developed.
   * **How**: Developers write unit tests for each component.
3. **Integration Testing (QC)**
   * **Purpose**: Ensure different modules work together correctly.
   * **When**: After individual modules are developed and integrated.
   * **How**: Test cases executed to verify module interaction.
4. **User Acceptance Testing (QC)**
   * **Purpose**: Confirm the system meets the needs of the end-users.
   * **When**: After system testing, before deployment.
   * **How**: End-users test the system in a controlled environment.
5. **Process Reviews (QA)**
   * **Purpose**: Ensure the project follows established standards and processes.
   * **When**: Periodically throughout the project lifecycle.
   * **How**: Project manager reviews project processes and documentation.

### 

### Request 4: Project Phases and Deliverables (20%)

**Phase 1: Initiation**

* **Deliverable 1**: Project Charter
  + **Content**: Defines project scope, objectives, stakeholders, and constraints.
* **Deliverable 2**: Initial Project Plan
  + **Content**: High-level timeline, resource allocation, and project milestones.

**Phase 2: Planning**

* **Deliverable 1**: Detailed Requirements Specification
  + **Content**: In-depth documentation of functional and non-functional requirements.
* **Deliverable 2**: System Design Document
  + **Content**: Blueprint of system architecture, database design, and UI/UX design.

**Phase 3: Execution**

* **Deliverable 1**: Developed System Modules
  + **Content**: Functional system components including front-end, back-end, and database.
* **Deliverable 2**: Integrated System
  + **Content**: Fully integrated system with all modules working together.

**Phase 4: Closure**

* **Deliverable 1**: User Training Materials
  + **Content**: Documentation and training resources for users.
* **Deliverable 2**: Project Closure Report
  + **Content**: Final project summary, lessons learned, and post-deployment plan.

### 

### Request 5: Main Cost Items and Estimations (20%)

1. **Development Effort**
   * **Description**: Time spent by developers to write and test code.
   * **Estimation**: 400 man-days.
2. **Testing Effort**
   * **Description**: Time spent by testers to perform QA/QC activities.
   * **Estimation**: 150 man-days.
3. **UI/UX Design**
   * **Description**: Time spent designing and refining the user interface.
   * **Estimation**: 80 man-days.
4. **Database Design and Management**
   * **Description**: Time spent designing, implementing, and managing the database.
   * **Estimation**: 70 man-days.
5. **Project Management**
   * **Description**: Time spent by the project manager overseeing and coordinating the project.
   * **Estimation**: 50 man-days.