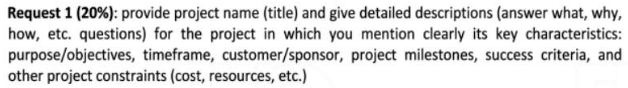
****

Request 1:   
**Project Name**: Student Management System for FU

**What**: Develop a web based Student Management System to automate student related tasks such as registration, attendance tracking, grade management, and communication.

**Why**: To streamline administration tasks, reduce manual errors, and enhance access to data for student, lecture, and staff.

**How**: Using agile development methodology. Technologies inclue Angular (front-end), .NET Core (back-end), and SQL Server (database). The process involves gathering requirements, system design, development, testing, and deployment.

**Purpose/objective**:

Automate student record management.

Improve communication among student, staff, and lecture.

Enhance accuracy and accessibility of student data.

**Timeframe**: 6 months

**Customer/Sponsor**: FU IT Department.

**Project millstone**:

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 lecture.

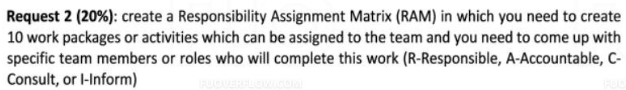
On time and on budget deployment.

30% reduction in processing time for student related administrative tasks.

**Project Constraints:**

**Cost**: $50,000

**Resources**: 1 Project Manager, 4 Developers, 2 Tester, 1 Database Administrator, 1 UX/UI Designer.



Request 2:

Work Package/Activitty Project Manager Developers Tester UX/UI Designer Database Admin

Requirement Gathering A R I C I

System Design A R C I C

Database Design I C I C R

Front-End Development I R I R I

Back-End Development I R I I C

Intergration Testing I C R I I

User Acceptance Testing C I R C I

Deployment A R I I R

User Training R I I I I

Post Deployment Support A I I I C

****

Request 2:

**Project Manager:**

**Responsibility:**

Manage the overall project timeline, budget, and resources.

Coordinate and facilitate communication between the project team and stakeholder.

**Software Developer:**

**Responsibility:**

Design and implement the system architecture and application logic

Write high quality, maintainable code for the front end and back end components

**UX/UI Designer:**

**Responsibility:**

Conduct user research and gathering requirements

Design intuitive and accessible user interface

**QA Tester:**

**Responsibility:**

Develop and execute test plans to ensure the system meet requirements

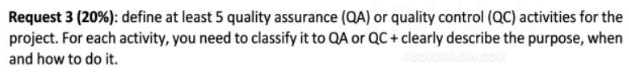
Identify and document defects, and work with the development team to resolve them.

**Database Administrator**

**Responsibility:**

Design and implement the database schema

Ensure the database is secure, scalable, and performant



Request 3:

**Code Review (QA)**

Purpose: Ensure adherences to coding standards and identify issues early

When: During development, before merging code

How: Peer review of code using checklist to detect issues

**Unit Testing (QC)**

Purpose: Verify that individual component work as intended

When: After each moudle is developed

How: Developers write unit test for each component

**Intergration Testing (QC)**

Purpose: Ensure difference module work together correctly

When: After individual modules are developed and integrated

How: Test cases executed to verify module interaction

**User Acceptance (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 an controlled environment

**Process Reviews (QA)**

Purpose: Make sure the project follows the right standards and steps

When: Regularly during the project

How: Project manager reviews project process and documentation

****

Request 3:

**In-scope:**

**Student Registration**: Functionally for students to register for course, update personal information, and view academic records

**Attendance Tracking**: Automated tracking and reporting of student attendance

**Grade Management**: Tools for instructor to input, update, and view student grade

**Communication Features**: Enabling communication between students, instructor, and staff via messaging, announcements, and calenders

**Reporting and Analytics**: Generating reports and dashboard to provide insights into student data and trends

**Out-of-scope:**

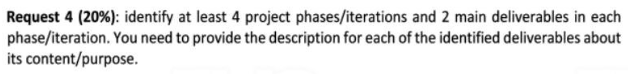
**External System Integration**: Connecting the Student Management System to other university system

**Moible App Development**: Creating a dedicated mobile application for accessing the system

**Advanced Analytics and Predictive Modeling**: Applying machine learning for predicting student outcomes

**Course Scheduling**: Functionality to manage class timetable and student scheduling

**Admission Management**: Processing student application and managing the admissions process



Request 4:

**Phase 1: Initiation**

**Deliverables 1:** Project Charter

**Content**: Define project scope, objective, stackholders, and constraints

**Deliverables 2:** Initial Plan

**Content**: High level timeline, resource allocation, and project milestones

**Phase 2: Planning**

**Deliverables 1:** Detailed Requirement Specification

**Content**: In depth documentation of functional and non functional requirements

**Deliverables 2:** System Design Document

**Content**: Blueprint of system architecture, database design, and UX/UI design

**Phase 3: Execution**

**Deliverables 1:** Developed System Modules

**Content**: Functional system components including front-end and back-end, and database

**Deliverables 2:** Intergrated System

**Content**: Fully integrated system with all modules working together

**Phase 4: Closure**

**Deliverables 1:** User Training

**Content**: Documentation and training resources for users

**Deliverables 2:** Project Closures Report

**Content**: Final project summary, lessons learned, and post deployment plan

****

**FS (Finish-to-Start)**, **SS (Start-to-Start)**, **SF (Start-to-Finish)**, **FF (Finish-to-Finish)**

Request 4:

**Gather Requirement:**

**Relationships**: FS with Develop System Design

**Develop System Design:**

**Relationships**: SS with Gather Requirement and FS with Implement Front-end

**Implement Front-end:**

**Relationships**: SS with Implement Back-end and FS with Integrate Components

**Integrate Components:**

**Relationships**: SS with Implement Back-end and FS with Conduct Testing

**Conduct Testing:**

**Relationships**: SS with Conduct Testing and FS with Deploy to Production

**Deploy to Production:**

**Relationships**: SS with Conduct Testing

**Provide Training:**

**Relationships**: FF with Obtain User Feedback

**Obtain User Feedback**

**Relationships**: SF with Provide Training and FS with Iterate on Solution

**Iterate on Solution:**

**Relationships**: SS with Obtain User Feedback



Request 5:

**Project Management:**

**Description**: Oversight of the project timeline, budget, and resources by the Project Manager

**Estimation**: 120 man-days

**System Design and Architecture:**

**Description**: Designing the overall system architecture, including front-end, back-end, and database components

**Estimation**: 80 man-days

**Front-end Development:**

**Description**: Implementing the user interface and interactive features of the application

**Estimation**: 160 man-hours

**Back-end Development:**

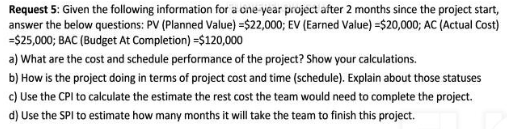
**Description**: Deploying the server side logic, api, and data processing functionalities

**Estimation**: 240 man-hours

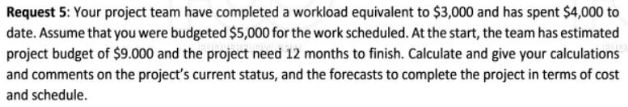
**Testing and Quality Assurance:**

**Description**: Executing test plans, identifying and resolving defects

**Estimation**: 80 man-hours



Request 5:  
We have  
+ PV = 22000$  
+ EV = 20000$  
+ AC = 25000$  
+ BAC = 120000$  
  
CPI = EV / AC = 20000 / 25000 = 0.8  
SPI = EV / PV = 20000 / 22000 = 0.909  
  
CPI = 0.8 < 1 => The project is over budget  
SPI = 0.909 < 1 => The project is over scheduled  
  
EAC = BAC / CPI = 120000 / 0.8 = 150000=> With the current rate, the final budget will cost 150000 more than the original by 150000 - 120000 = 30000$  
  
Months take to finish = 12 / SPI = 12 / 0.909 = 13.2 months  
=> It's take 13.2 months to complete at the current SPI



We have:  
+ EV (Earned Value) = 3000$  
+ AC (Actual Cost) = 4000$  
+ PV (Planned Value) = 5000$  
+ BAC = 9000$  
  
CPI = EV / AC = 3 / 4 = 0.75  
SPI = EV / PV = 3 / 5 = 0.6  
  
CPI = 0.75 < 1 => The project is over budget  
SPI = 0.6 < 1 => The project is under scheduled  
  
EAC = BAC / CPI = 9000 / 0.75 = 12000 => With the current rate of CPI the final budget will be 12000 more than the original BAC 12000 - 9000 = 3000[/imath]  
  
months to finish = 12 months / SPI = 12 / 0.6 = 20 months => The actual time will be 20 months with the current SPI