

**THE UNIVERSITY OF ZAMBIA**

**SCHOOL OF NATURAL SCIENCES (NS)**

**DEPARTMENT OF COMPUTER SCIENCE (CS)**

**COMPUTER ID:** 2020990849

**NAME:** COSSAM SOKO

**COURSE:** Software Quality Assurance (CSC4642)

**TAKE HOME TEST:** Project Development and Quality Plans

**LECTURER:** Mrs. Monica Kabemba

**INTRODUCTION**

Effective planning is important for the success of projects. Moreover, ensuring quality assurance is a key factor in meeting project goals. Therefore, to achieve these quality assurance objectives, comprehensive development and quality plans will be formulated for my final year project—a secondary school placement system. These plans will focus on the following aims:

1. Organizing project development activities: Establishing a timeline for each project phase to guarantee timely delivery.
2. Assessing the number of personnel required for project completion: Identifying the team size and roles necessary for successful development and implementation.
3. Calculating the estimated budget: Determining the total costs involved, including resources, tools, and personnel.
4. Recruiting additional team members as needed: Finding and onboarding extra team members to support the development process, if necessary.
5. Allocating the necessary development resources: Ensuring that all required tools and environments are accessible for efficient project execution.
6. Addressing potential development risks: Identifying possible risks and creating strategies to mitigate them throughout the project lifecycle.
7. Carrying out essential Software Quality Assurance activities: Implementing quality assurance measures to confirm that the system meets specified standards and user expectations.
8. Providing the supervisor with essential data for project oversight: Regularly updating stakeholders with relevant information to facilitate effective management and control

The development of this secondary school placement system incorporates various elements related to both development and quality assurance. These components include project deliverables, project interfaces, development methodology and tools and the map of the development process showing planned development activities, planned software quality assurance activities, estimates of activity duration and logical sequence of activities.

**PEOJECT PRODUCTS**

**Software Application**:

Secondary School Placement System: This is the main product of the project. It will be a web-based platform designed to facilitate the application and placement process for secondary schools. Features may include application submission, real-time status tracking, and communication tools for students and schools.

**Documentation:**

User Manual: This document will provide instructions for users on how to navigate and utilize the system. It will cover functionalities of the system.

Technical Documentation: This will outline the system architecture, database schema, API specifications, and any third-party integrations. It will help future developers understand the system’s design and facilitate maintenance and updates.

Test Plans: This document will detail the testing approach, including test cases, methodologies, and expected outcomes. It will help ensure that the system meets the defined requirements and functions correctly.

Deployment Package: This will include all the necessary files for deploying the application on a server, along with clear instructions on how to set up the environment and migrate data if needed.

**PROJECT INTERFACES**

**Admin Dashboard:** This interface will allow administrators to manage student applications, view statistics, and oversee the placement process. It will include features for adding and editing school information and managing user accounts.

**Student Dashboard:** A user-friendly interface where students can submit their applications, check their application status.

**School Dashboard:** This will provide schools with access to view incoming applications, evaluate candidates, and communicate decisions to students through messaging features.

System Interfaces:

**Database Interface**: The application will interact with a database (e.g., SQL LITE) for data storage and retrieval. This interface will handle CRUD operations (Create, Read, Update, Delete) for student and school data.

**PROJECT METHODOLOGY AND DEVELOPMENT TOOLS**

**Methodology:**

Businesses now operate in a global environment that is constantly evolving. As a result, it's nearly impossible to establish a complete and unchanging set of software requirements. Requirements are often in flux, and customers frequently struggle to anticipate how a system will influence their workflows, how it will integrate with other systems, or which user tasks should be automated. Often, it is only after the system is deployed and users have experience with it that the actual needs become evident.

Due to this, the agile development methodology (specifically SCRUM) which allows for iterative development and continuous feedback has been chosen for the project’s development. This method not only enables the rapid delivery of core software but also allows for ongoing user input, enabling the software to be developed iteratively through successive stages. Regular sprint planning and reviews which will ensure that the project aligns with user needs and stakeholder expectations.

By design, the system is being developed in incremental stages, with the supervisor—who acts as the end-user—reviewing each iteration and identifying any changes to be incorporated in the next cycle.

**Development Tools:**

**Programming Language:** Python programming language

**Framework:** Django provides a robust framework for building secure and scalable web applications quickly, complete with built-in features like authentication and an admin panel.

Database: SQLITE

**Version Control:** Using Git will help manage changes to the code base and maintaining a history of revisions.

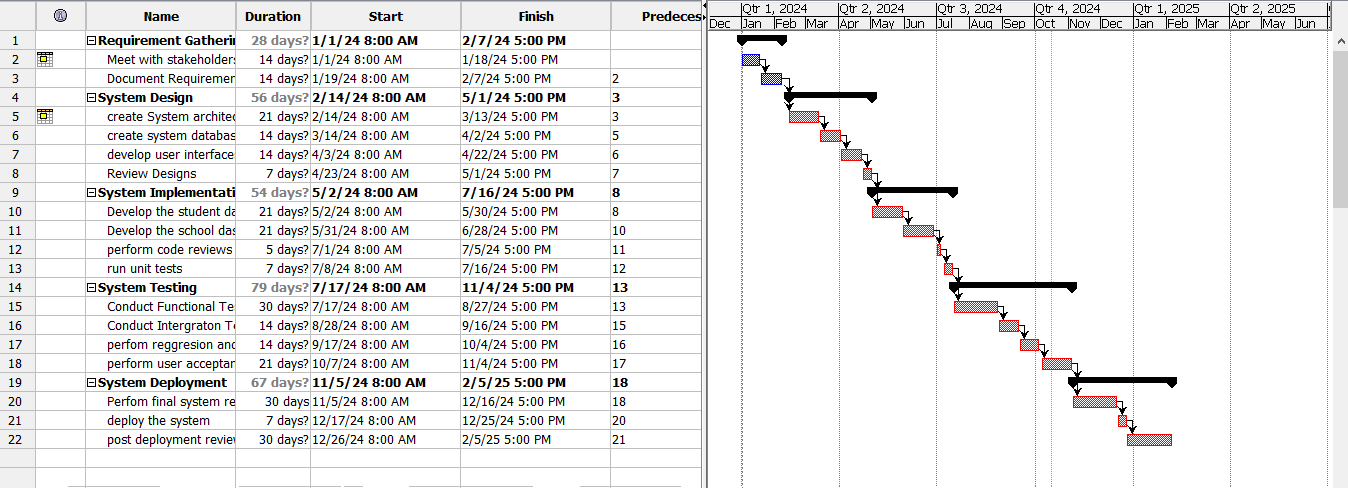
**Project Management:** Tools like projectlibre, Trello will assist in tracking tasks, assigning responsibilities, and managing timelines.

**Testing Framework:** Django’s built-in testing tools will facilitate unit and integration testing to ensure application reliability.

**MAP OF THE DEVELOPMENT PROCESS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PHASE** | **PLANNED DEVELOPMENT**  **ACTIVITIES** | **PLANNED SQA ACTIVITIES** | **ESTIMATED DURATION** | **LOGICAL SEQUENCE** |
| **1.**  **Requirement Gathering** | Meet with stakeholders | Review requirements | 14 Days | **1** |
|  | Document Requirements | Review requirements | 14 Days | **2** |
|  |  |  |  |  |
| **2.**  **Design phase** | create System architecture | Review Designs | 21 Days | **3** |
|  | create system database | Review Designs | 14 Days | **4** |
|  | develop interfaces | Review Designs | 14 Days | **5** |
|  |  |  |  |  |
| **3.**  **Development phase** | Develop the student dashboard | Perform Code Reviews | 21 Days | **6** |
|  | Develop the school dashboard | Perform Code Reviews | 21 days | **7** |
|  |  | run unit tests | 7 Days | **8** |
|  |  |  |  |  |
| **4.**  **Testing Phase** | Conduct Functional Testing | Prepare Test Cases | 30 Days | **9** |
|  | Conduct integration Testing | Execute Integration Tests | 14 Days | **10** |
|  | perform regression and system testing | Execute Regression Tests | 14 Days | **11** |
|  | perform user acceptance testing | Facilitate UAT Sessions | 21 Days | **12** |
|  |  |  |  |  |
| **5.**  **Deployment phase** | Perform final system review | Conduct Final System Validation | 30 Days | **13** |
|  | deploy the system | Monitor Deployment | 7 Days | **14** |
|  | Post Deployment review | Performance monitoring | 30 days | **15** |

**Gant chart**

****

**Development Risks**

**Scope creep is one of the significant Risks that can arise while developing the secondary school placement system**, where the project's requirements expand beyond the original plan. This will occur if there are unclear boundaries or pressure from stakeholders requesting additional features or changes. Without a strict change management process, scope creep will lead to increased costs and extended timelines, ultimately jeopardizing the project’s success.

**User acceptance** is yet another concern. If teachers, students, and parents are resistant to adopting the new system, it can hinder its effectiveness. This resistance may stem from a lack of understanding of the system’s benefits or inadequate training. Engaging users early in the process and providing thorough training can help alleviate this risk.

**Data security and privacy** pose substantial risks as well. Handling sensitive student information requires strict adherence to regulations. Any data breaches or mismanagement can lead to severe legal repercussions, reputational damage, and loss of trust among users. Ensuring robust security measures are in place is essential to mitigate this risk.

**Inadequate** **testing** can lead to undetected bugs and issues that compromise the system's functionality. If testing phases are rushed or not thoroughly executed, users may encounter problems when the system is deployed. Establishing a comprehensive testing strategy, including unit, integration, and user acceptance testing, is vital for mitigating this risk.

**Timeline delays** often emerge from unforeseen circumstances or bottlenecks in development. If critical milestones are missed, the entire project can be pushed back, affecting the school’s operational readiness. Regular monitoring of progress and adjusting plans as needed can help manage this risk effectively.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Risk Management Actions** |
| **Scope Creep** | Expansion of project requirements beyond the original plan, leading to increased costs and extended timelines. | - Implement a strict change management process.  - Define clear project boundaries and requirements upfront.  - Regularly communicate with stakeholders about scope limits. |
| **User Acceptance** | Resistance from teachers, students, and parents due to lack of understanding or inadequate training on the new system. | - Engage users early in the development process.  - Provide comprehensive training and support.  - Gather feedback from users to improve system design and usability. |
| **Data Security & Privacy** | Risks related to handling sensitive student information, including potential data breaches and legal repercussions. | - Implement robust security measures and encryption.  - Conduct regular security audits.  - Train staff on data privacy regulations and best practices. |
| **Inadequate Testing** | Rushed or insufficient testing phases that lead to undetected bugs and system functionality issues. | - Establish a comprehensive testing strategy (unit, integration, user acceptance).  - Allocate adequate time for testing phases.  - Involve users in the testing process. |
| **Timeline Delays** | Unforeseen circumstances or bottlenecks that cause critical milestones to be missed, impacting project readiness. | - Regularly monitor project progress and timelines.  - Identify potential bottlenecks early and address them proactively.  - Adjust plans as needed to stay on track. |