



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Summer 23-24

Section: A

Software Quality Assurance and Testing

Education Assistant

A Report submitted
By

| SN | Student Name | Student ID |
|----|--------------------|------------|
| 1 | Nokibul Arfin Siam | 21-44793-1 |
| 2 | Md Sahadul Haque | 21-45417-3 |
| 3 | Mahamoda Akter | 20-44269-3 |
| 4 | Jahir islam | 20-43882-2 |

Software Test Plan

for

<Project: Education Assistant>

Version 1.0 approved

Prepared by <author: Nokibul Arfin Siam>

<organization: AIUB>

<date created: 28/09/2024>

Table of Contents

| | |
|---|----|
| Revision History..... | 3 |
| 1. TEST PLAN IDENTIFIER: RS-MTP01.3 | 4 |
| 2. REFERENCES..... | 4 |
| 3. INTRODUCTION..... | 4 |
| Background to the Problem | 4 |
| Solution to the Problem | 4 |
| 4. REQUEIREMNT SPECIFICATION | 5 |
| 4.1 System Features..... | 5 |
| 4.2 System Quality Attributes..... | 7 |
| 4.3 System Interface | 10 |
| 4.4 Project Requirements | 12 |
| 5. FEATURES NOT TO BE TESTED | 15 |
| 6. TESTING APPROACH..... | 16 |
| 6.1 Testing Levels | 16 |
| 6.2 Test Tools..... | 16 |
| 6.3 Meetings..... | 17 |
| 7. TEST CASES/TEST ITEMS | 19 |
| 8. ITEM PASS/FAIL CRITERIA | 20 |
| 9. TEST DELIVERABLES..... | 21 |
| 10. STAFFING AND TRAINING NEEDS | 22 |
| 11. RESPONSIBILITIES | 23 |
| 12. TESTING SCHEDULE | 25 |
| 13. PLANNING RISKS AND CONTINGENCIES..... | 26 |
| 14. APROVALS..... | 28 |

Revision History

| Revision | Date | Updated by | Update Comments |
|----------|------------|--------------------|-----------------|
| 0.1 | 2024.05.04 | Md Sahadul Haque | First Draft |
| 0.2 | 2024.09.28 | Nokibul Arfin Siam | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

1. TEST PLAN IDENTIFIER: EA-SaaS1.2

2. INTRODUCTION

Background to the Problem

Education in Bangladesh faces significant challenges due to the high costs of private education services, which many families cannot afford. This financial barrier prevents numerous students from accessing quality educational resources and opportunities for personal development. The country's dense population further exacerbates the issue, creating competition for limited educational resources. Moreover, teachers and researchers often lack platforms where they can explore, develop, and share their ideas effectively, limiting innovation and collaboration in education. Additionally, students seeking guidance on thesis writing and academic projects frequently find themselves without proper direction or mentorship, which is critical for their academic success.

The root cause of this problem lies in the inadequacy of accessible educational resources and the limited scope for collaboration between students and teachers. This issue is important to address because quality education is essential for individual growth and national development. Without intervention, many students will continue to fall behind, hindering their potential to contribute positively to society.

Solution to the Problem

To address these challenges, the Education Assistant (EA) project proposes an online platform that facilitates collaboration between students and teachers. This platform will enable teachers to share their knowledge and resources while providing students with the necessary support for academic success. The solution is particularly appropriate as it leverages technology to bridge the gap between students and educational resources, creating an inclusive environment for learning.

The EA platform allows students to consult with teachers, search for thesis papers, download available books, and engage in discussions about common academic challenges. Teachers can provide their availability for consultations, upload academic materials, and post solutions to student queries. By fostering real-time collaboration, the EA project aims to create a more effective learning experience compared to existing platforms.

The EA project is feasible as it aligns with the business objective of providing free educational resources to underprivileged students in Bangladesh, promoting quality education and enhancing the capabilities of teachers and researchers. This initiative is designed to empower students and teachers, allowing them to work together toward shared academic goals.

Existing studies in this problem area highlight the limited availability of dedicated educational platforms that facilitate collaboration between students and teachers. While platforms like Facebook and YouTube offer some resources, they lack the professional structure needed for effective

educational support. Other tools like Google Scholar provide valuable information but do not offer real-time collaboration or mentorship. Although Microsoft Teams exemplifies some collaborative features, it does not cater specifically to the educational context and requirements of students and teachers in Bangladesh. The EA project fills this gap by providing a tailored solution that addresses the unique needs of its users, ultimately contributing to improved educational outcomes.

3. REQUIREMENT SPECIFICATION

3.1 System Features

1 System Login

Functional Requirements:

- 1.1 The software shall allow users to log in using their valid username and password.
- 1.2 The username and password will be verified against the database.
- 1.3 If the login is successful, the Newsfeed page (Teacher/Student) will be displayed.
- 1.4 If the username or password is incorrect, the system will display an option for "Forgot Password" and login via code.
- 1.5 If the username and password are inserted incorrectly more than three times, the user will be blocked from login for 15 minutes.

Priority Level: High

Precondition: User must have a valid username and password.

2 Registration

Functional Requirements:

- 2.1 Student Registration:
 - 2.1.1 Name, Date of Birth (DOB), Gender, Phone number, Email, Address, Field of Study, and academic info (optional).
 - 2.1.2 Username (validated by the system to check availability), create password, and confirm password.
 - 2.1.3 If any required information (Name, DOB, Gender, Phone number, Email, Address, Field of Study) is missing, the student will not be able to register.
 - 2.1.4 The system shall validate if the username is taken, and if so, display a message: "Username taken! Try another username."
 - 2.1.5 The system will suggest password rules (at least one symbol, one character, one numeric input, minimum 6 characters).
- 2.2 Teacher Registration:
 - 2.2.1 Name, DOB, Gender, Phone number, Email, Address, Domain(s), and Availability.
 - 2.2.2 Experiences (optional), Username (validated by the system), create password, confirm password.
 - 2.2.3 If any required information (Name, DOB, Gender, Phone number, Email, Address, Domain) is missing, the teacher will not be able to register.
 - 2.2.4 The system shall validate the availability of the username and prompt: "Username taken! Try another username."
 - 2.2.5 The system will suggest password rules (at least one symbol, one character, one numeric input, minimum 6 characters).

Priority Level: High

Precondition: None

3 Profile

Functional Requirements:

- 3.1 After successful login, users can view their profile.
- 3.2 Teacher's Profile:
 - 3.2.1 Teachers will have different actions available (Consult, Post, Comment, Update Schedule, Update Info, Search).
 - 3.2.2 Confirm Consultations:
 - 3.2.2.1 Teachers can view consultation requests from students.
 - 3.2.2.2 If a teacher accepts a consultation, the student will be notified.
 - 3.2.2.3 Teachers can reschedule consultations or cancel them, and students will be notified accordingly.
 - 3.2.3 Post:
 - 3.2.3.1 Teachers can upload Reports, Theses, Research Papers, and Questions.
 - 3.2.3.2 They can ask for research collaboration or request help.
 - 3.2.3.3 Teachers can publish posts, which can be seen by others within the same domain.
 - 3.2.3.4 Non-published posts will remain invisible to others.
 - 3.2.4 Update:
 - 3.2.4.1 Teachers can update their consultation schedules.
 - 3.2.4.2 General information can also be updated by both teachers and students.
 - 3.2.5 Search:
 - 3.2.5.1 Teachers can search for user profiles, publications (posts, research papers), and resources by entering keywords.
 - 3.2.5.2 If no results are found, the system will display "no results."
- 3.3 Student's Profile:
 - 3.3.1 Students will have different actions available (Consult, Post, Comment, Update, Search, Make Donations).
 - 3.3.2 Request Consultations:
 - 3.3.2.1 Students can request consultations with teachers based on the available schedule.
 - 3.3.2.2 The request status will remain pending until confirmed by the teacher.
 - 3.3.3 Request for Solutions:
 - 3.3.3.1 Students can post problem statements to request help.
 - 3.3.3.2 Others within the same domain can comment and provide solutions.
 - 3.3.4 Update:
 - 3.3.4.1 Students can update their general information and click confirm to save changes.
 - 3.3.5 Search:
 - 3.3.5.1 Students can search for user profiles, publications, and resources by entering keywords.
 - 3.3.5.2 If no results are found, the system will display "no results."

Priority Level: Medium

Precondition: Successful login

4 Resources

Functional Requirements:

- 4.1 The system shall provide a searchable database of field-related resources (research works, publications).
- 4.2 Search Operation:
 - 4.2.1 Users (students and teachers) can search for resources by entering keywords.
 - 4.2.2 If the search matches, the result will include research works from the database.
 - 4.2.3 If no match is found, no results will be displayed.

Priority Level: Medium

Precondition: None

5 Donations

Functional Requirements:

5.1 Non-User Donation:

- 5.1.1 The system will prompt non-users to fill in Name, Address, and Contact information.
- 5.1.2 After selecting the donation amount and payment method, the user will click the confirm button.
- 5.1.3 If the payment is successful, the system will display "Thank you for donating to us."
- 5.1.4 If the payment fails, a failure message will be shown.

5.2 User Donation:

- 5.2.1 Registered users can donate by selecting the donation amount and payment method, then clicking the confirm button.
- 5.2.2 If the payment is successful, the system will display "Thank you for donating to us."
- 5.2.3 If the payment fails, a failure message will be shown.

Priority Level: Low

Precondition: Payment method integration

4.2 System Quality Attributes

1. System Login

1.1. Security:

- 1.1.1. Password Handling: Passwords should be securely stored using encryption techniques (e.g., hashing with salt). Authentication mechanisms should ensure that passwords are never stored in plain text or transmitted insecurely.
- 1.1.2. Account Lockout Mechanism: After multiple failed login attempts, the system should temporarily block access to mitigate brute-force attacks. A time-based lockout (e.g., 15 minutes) ensures that attackers can't repeatedly guess credentials.
- 1.1.3. Verification Code: After three failed login attempts, the system generates a random verification code, adding an extra layer of security. This mitigates the risk of bots attempting to force their way into user accounts.

1.2. Usability:

- 1.2.1. User Feedback: Clear error messages should guide users through failed attempts, explaining when a password or username is wrong. After three incorrect attempts, prompt users with options like "Forgot Password" and explain how to unblock their account.
- 1.2.2. User-Friendly Recovery: For forgotten passwords, the system should allow recovery via email or phone, making it easy for users to regain access without contacting support.

1.3. Reliability:

- 1.3.1. Consistent Lockout Behavior: The system should ensure that after the defined number of incorrect attempts, the lockout mechanism activates every time. It should consistently prevent access for the set time limit (15 minutes) and reset the counter afterward.
- 1.3.2. Redundancy for Code Generation: If the system uses email or phone to send verification codes, redundancy measures (e.g., retry mechanisms) should ensure delivery even if the first attempt fails.

2. Registration (Student/Teacher)

2.1. Security:

- 2.1.1. Password Validation: Passwords should meet specific criteria (e.g., a mix of characters, numbers, symbols) to ensure they are strong enough to prevent easy guessing. The system must ensure that users cannot reuse common passwords or past compromised passwords.
- 2.1.2. Data Validation & Sanitization: All registration inputs (Name, DOB, email, etc.) should be validated for correctness (e.g., format checks) and sanitized to avoid SQL injection or other data-related attacks.
- 2.1.3. Unique Username Enforcement: The system must check if a username is already taken to prevent conflicts. This check should be performed securely, ensuring that usernames are not exposed in a vulnerable way.

2.2. Usability:

- 2.2.1. Real-time Feedback: As users fill out the form, the system should provide real-time feedback on username availability, password strength, and missing required fields. For instance, if a username is already taken, users should immediately receive a message, "Username taken! Try another."
- 2.2.2. Clear Validation Messages: If the user enters invalid data, clear, actionable messages should guide them on how to correct it. For example, if a password is too weak, the system should suggest making it stronger by adding more characters, symbols, etc.

2.3. Maintainability:

- 2.3.1. Easy Field Updates: The registration system should be flexible, allowing developers to add, remove, or modify fields (e.g., new attributes like hobbies or interests) without extensive rework.
- 2.3.2. Separation of Logic: The system should use clean, modular code, separating validation, database interaction, and feedback logic, making it easier to maintain and expand in the future.

3. Profile (Teacher/Student)

3.1. Scalability:

- 3.1.1. Handling Growth: As the platform scales to thousands or millions of users, the system should handle large volumes of profile updates, post submissions, and consultation requests without performance degradation. Techniques like database indexing, load balancing, and caching can be used to maintain responsiveness.
- 3.1.2. Dynamic Loading: Profiles should load data dynamically, ensuring that even when large amounts of information (e.g., consultation schedules, research papers) are displayed, the user interface remains responsive.

3.2. Usability:

- 3.2.1. Intuitive Profile Updates: The system should allow users (students and teachers) to easily update personal information, schedules, and post content. Clear buttons, such as "Edit Profile" and "Save Changes," guide users through these processes.
- 3.2.2. Post Validation & Suggestions: The platform should validate research papers, thesis uploads, and collaborations to ensure domain relevance. An AI-powered suggestion tool could be integrated to recommend relevant collaborators or resources based on a user's domain or research interest.

3.3. Modifiability:

- 3.3.1. Modular Profile Components: Each section of a user's profile (consultations, posts, etc.) should be developed as independent components so that changes in one area do not

affect others. For example, adding new fields to the consultation schedule should not interfere with research collaboration features.

- 3.3.2. Customizable Actions: The system should allow easy customization of teacher and student actions. For instance, teachers could add new consultation methods, and students could request new types of help with minimal changes to the overall system.

3.4. Interoperability:

- 3.4.1. Integration with Other Modules: Profiles must seamlessly interact with consultation, research collaboration, and posting modules. For instance, if a teacher schedules a consultation, that information should reflect in both the consultation module and the teacher's profile.

- 3.4.2. Cross-Platform Access: Ensure that profile functionalities work consistently across devices (mobile, desktop, tablet)

4. Resources

4.1. Performance:

- 4.1.1. Efficient Search Mechanism: Since the database will store many research papers, reports, and academic resources, the search feature should be optimized. Implement indexing and caching mechanisms to ensure that search queries return results quickly, even with large datasets.

- 4.1.2. Load Management: If multiple users are searching for resources simultaneously, the system must manage the load efficiently, ensuring that the performance does not degrade for any individual user.

4.2. Usability:

- 4.2.1. Clear Search Results: When searching, users should receive clear and accurate results based on their keyword. The system should provide filtering options (e.g., by date, relevance, or domain) to help users refine their searches.

- 4.2.2. No Results Feedback: If no resources match the search query, the system should provide feedback like, "No resources found. Try adjusting your search criteria," instead of leaving the user confused.

4.3. Scalability:

- 4.3.1. Handling Growth in Resources: As more resources are added to the database, the system must maintain efficient retrieval times. The use of scalable databases and load distribution across multiple servers will ensure that search performance remains consistent.

- 4.3.2. Expanding Resource Types: The system should be flexible enough to accommodate new resource types or categories (e.g., audio, video, and articles), requiring minimal changes to the core infrastructure.

5. Donations

5.1. Security:

- 5.1.1. Secure Payment Processing: All payment transactions must be encrypted using secure protocols (e.g., HTTPS, SSL) to protect sensitive data such as credit card information and personal details. Additionally, third-party payment processors (like Stripe or PayPal) should be integrated to handle transactions securely.

- 5.1.2. Fraud Prevention: Implement mechanisms to detect and prevent fraudulent donation attempts. This could include captcha verification, IP address checks, and transaction monitoring.

5.2. Reliability:

- 5.2.1. Consistent Payment Handling: The donation process must reliably handle both successful and failed transactions. Users should always be informed if a transaction was

processed successfully or if it failed (with reasons such as insufficient funds or transaction timeouts).

5.2.2. Backup Systems: Ensure that donation functionality has redundancy in case of system failures. For example, if a payment service is temporarily down, users should be able to retry the transaction later without data loss.

5.3. Availability:

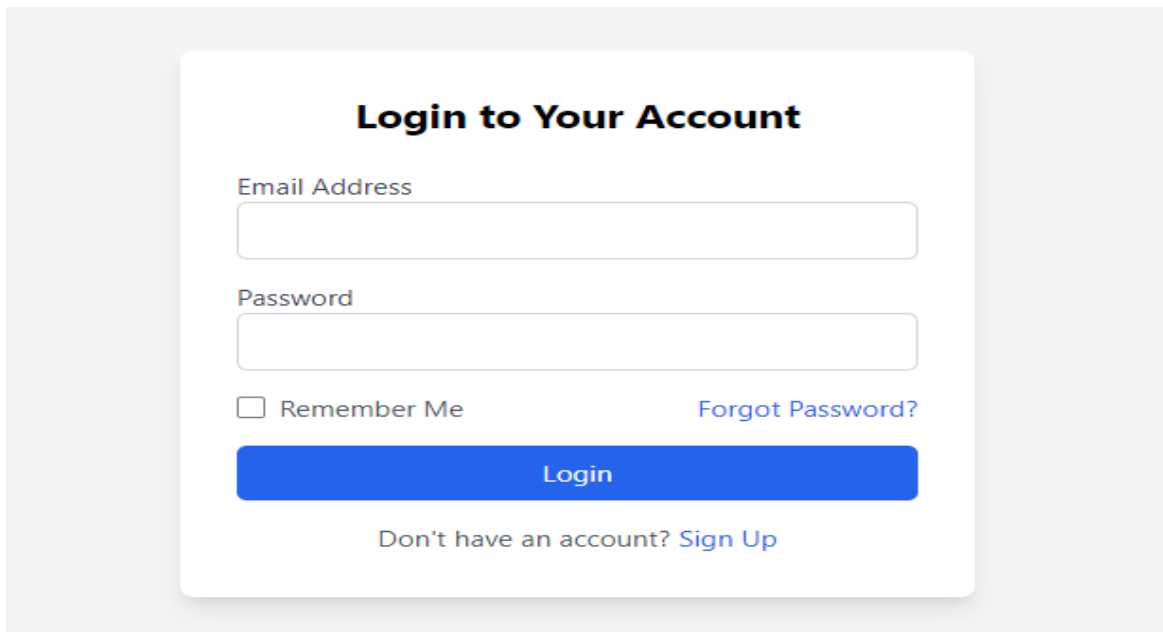
5.3.1.24/7 Donation Access: Since donations can be made by both registered and non-registered users, the donation system should have high availability with minimal downtime. Regular backups and maintenance processes should be implemented to ensure this.

5.3.1.1. Scalability for Large Donations: The system should scale to handle large volumes of donations during peak periods, such as fundraising campaigns or disaster relief efforts.

4.3 System Interface



Figure1: Home page



Login to Your Account

Email Address

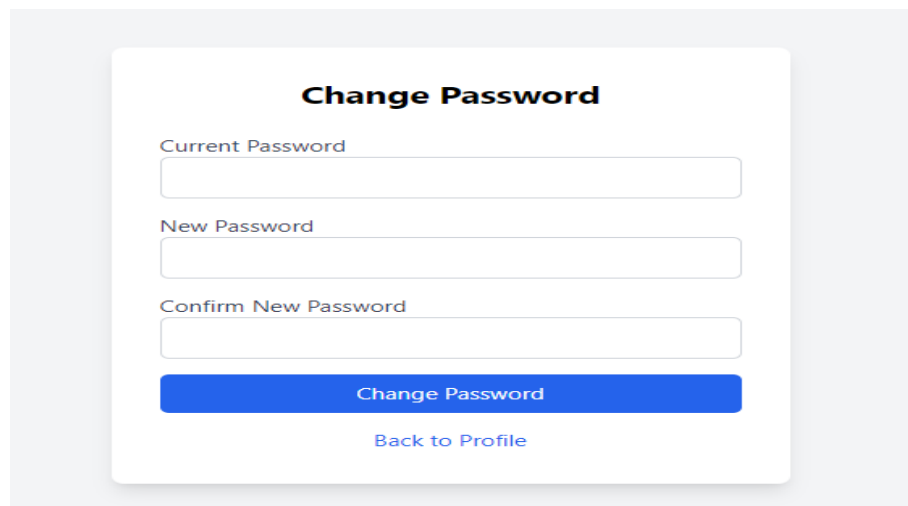
Password

☐ Remember Me [Forgot Password?](#)

Login

Don't have an account? [Sign Up](#)

Figure2: Login Page



Change Password

Current Password

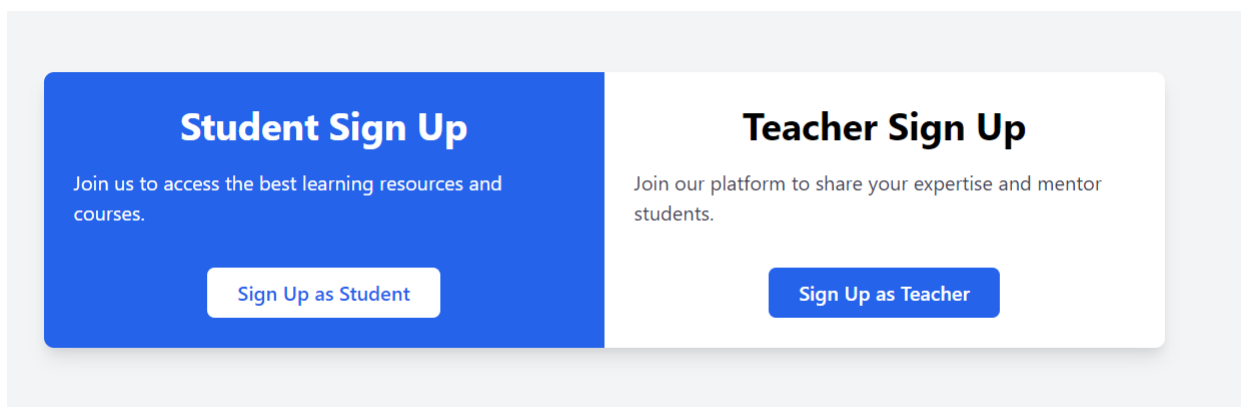
New Password

Confirm New Password

Change Password

[Back to Profile](#)

Figure2: Change password



Student Sign Up

Join us to access the best learning resources and courses.

Sign Up as Student

Teacher Sign Up

Join our platform to share your expertise and mentor students.

Sign Up as Teacher

Create Your Account

First Name

Last Name

Date of Birth

Institution

Department

Email

Phone Number

Address

Username

Password

Confirm Password

Sign Up


Figure2: Student Registration page

Create Your Account

First Name

Last Name


Date of Birth




Institution

Timeslot

Start Time



End Time



Email

Phone Number

Address

Sign Up

Figure3: Teacher register

4.4 Project Requirement

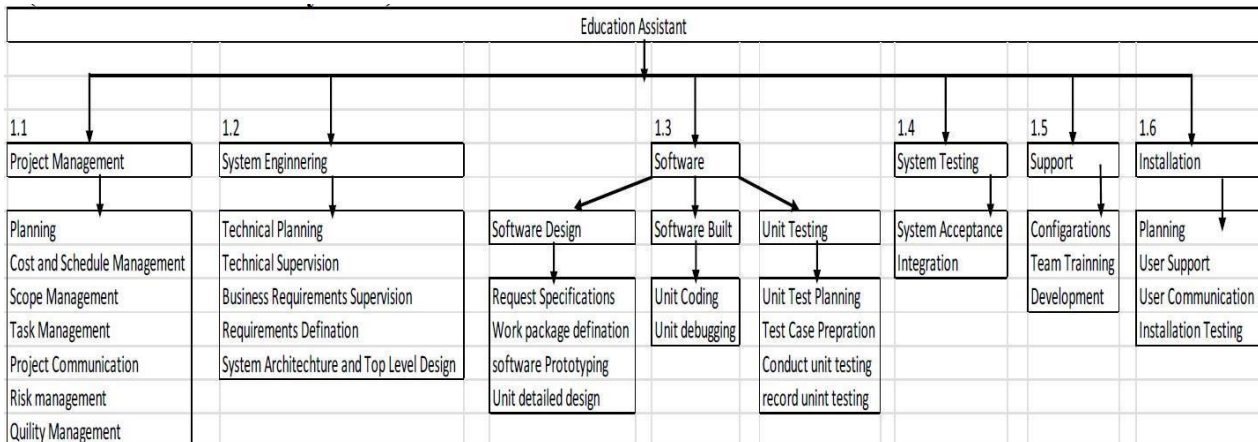


Figure4: WBS (Work Breakdown System)

Constructive Cost Model

As project is Organic,

Coefficient<Effort Factor>=2.4

project complexity, P=1.05

SLOC-dependent coefficient=0.38

SLOC=5,000

Effort = PM = Coefficient<Effort Factor>*(SLOC/1000)^P=2.4*(5,000/1,000)^{1.05}= 13.005 Person-Months

Development time = DM = 2.50*(PM)^T=2.50*(13.005)^{0.38}=6.63≈7

Months Required number of people = ST = PM/DM=13.005/7=1.85≈ 2

| Task Planned | Effort (hours) | Actual Effort (hours) |
|--------------|----------------|-----------------------|
| 1 | 6.5 | 7.2 |
| 2 | 3.2 | 3.7 |
| 3 | 12.8 | 13.4 |
| 4 | 8.3 | 9.1 |
| 5 | 4.7 | 4.9 |
| 6 | 9.1 | 10.5 |
| 7 | 5.6 | 6.2 |
| 8 | 15.9 | 16.8 |
| 9 | 11.2 | 11.9 |
| 10 | 7.4 | 8.1 |
| 11 | 14.5 | 14.2 |
| 12 | 6.8 | 7.5 |
| 13 | 2.9 | |
| 14 | 10.1 | |
| 15 | 8.7 | |
| Total | | |

EVA Analysis,

BCWP=106

BCWS=127.7

ACWP=113.5

BAC = 13.005*30=390.15≈ 391 Days

SPI = BCWP/ BCWS = 106/ 127.7 = 0.890070477

SV = BCWP - BCWS = 106 – 127.7 = -21.7 person-day ≈ -22 person-days

CPI = BCWP/ ACWP = 106/113.5=0.933920

CV = BCWP – ACWP = 106-113.5= -7.5 person-day

Project Constraints:

1. Time: Development time is estimated to be 7 months based on the Constructive Cost Model (COCOMO). However, with a negative schedule variance (SV) of -22 person-days, this indicates a potential delay. Therefore, the actual development time might extend to around 8 months, considering the delay of approximately 1 month.

2. Budget: The total estimated effort is 13.005 person-months (PM). Assuming a monthly developer salary of 30,000 BDT, the total budget estimate is calculated as 13.005 PM * 30,000 BDT = 390,150 BDT, approximately 391,000 BDT.

3. Resources: The required number of people based on the estimation is around 2 developers. This team size assumes of no significant scaling, and that the team can work efficiently to meet the deadlines.

4. Environment: The development team should have access to all necessary tools, such as software, hardware, and testing environments. Any potential external factors, such as remote work challenges or political issues, should be considered and planned to avoid further delays.

5. Complexity: The project complexity coefficient ($P = 1.05$) indicates that it is an organic project, suggesting a lower complexity with well-defined requirements, a small team, and standard technology. This means there are fewer risks and challenges, making the project relatively easier to manage.

Budget Estimation:

The estimated budget is based on the person-months required and the developer's monthly salary. With 13.005 person-months and an assumed salary of 30,000 BDT per month, the budget is calculated as $13.005 * 30,000 = 390,150$ BDT, which is approximately 391,000 BDT.

Time Estimation:

The COCOMO model estimates the development time as 7 months. However, with a negative schedule variance of -22 person-days, the project could face a delay of about 1 month, extending the total development time to around 8 months.

Effort Estimation:

The estimated total effort is 13.005 person-months. Assuming 22 working days per month, the effort in person-days is calculated as $13.005 * 22 = 286.11$ person-days.

Summary of Constraints:

Total budget is approximately **391,000 BDT**.

Total development time is estimated to be around **8 months**.

Required team size is **2 developers**.

Total effort is estimated to be 13.005 person-months or 286.11 person-days.

5. FEATURES NOT TO BE TESTED

Features related to the collaboration between students and teachers in real-time problem-solving may not be tested in isolation. Instead, the effectiveness of this collaboration will be assessed through overall user interactions and satisfaction metrics, as these are inherently linked to various functional aspects of the platform.

The user registration process for both students and teachers, while crucial, will not be tested for every individual scenario. Instead, testing will focus on the overall functionality of the registration forms and the validation mechanisms for usernames and passwords, as the system's ability to handle these correctly will be evaluated indirectly through the user login process.

The functionality for searching thesis papers and educational resources will not be specifically tested in every context. Instead, it will be assessed as part of the broader usability testing, where users' ability to navigate and find relevant materials will be observed and analyzed.

The ability of teachers to upload educational resources, such as papers and solution posts, will not be individually tested for every document type. Testing will focus on the general upload functionality, ensuring that the system accepts various document types and that related validation mechanisms work as intended.

Features addressing the consultation scheduling between students and teachers will be tested indirectly. The focus will be on ensuring that the overall appointment system functions correctly, with specific scenarios being verified through user feedback and interaction logs.

The user interface design and its impact on user experience will not be exhaustively tested for every possible interaction. Instead, general usability testing will be conducted to ensure that the interface supports easy navigation and enhances user engagement, particularly in the context of collaborative learning.

The data security measures implemented to protect user information will be tested indirectly through overall system security assessments rather than exhaustive testing of every individual security feature. The focus will be on ensuring compliance with general data protection standards and best practices.

The integration of various resources and external links will not be tested in isolation. Instead, the effectiveness of these integrations will be assessed as part of the overall user experience, focusing on how well users can access and utilize the external resources available through the platform.

6. TESTING APPROACH

6.1 Testing Levels

The testing for the Education Assistant (EA) project will consist of **Unit**, **System/Integration**, and **Acceptance** test levels. Given the project's scope, most testing will be carried out by the test manager, with participation from the development team. Since the EA project is non-profitable and has resource constraints, additional test personnel may not be available.

- **Unit Testing** will be conducted by individual developers, and all unit tests must be approved by the development team leader. Each developer will provide proof of testing, including test case lists, sample output, and defect logs. The unit test documentation will be shared with the testing team to ensure proper validation before proceeding to system-level testing. Tools like **Mocha** will be used to verify the functionality of individual components, such as registration forms, resource uploads, and consultations.
- **System/Integration Testing** will be carried out by the test manager and the development team leader, with help from individual developers as required. This phase will ensure that various modules of the EA system—such as user authentication, file uploads, and consultations—work seamlessly together. Any critical defects must be resolved before progressing, but the system may enter integration testing with no more than two major defects that do not obstruct core functionalities. **Selenium WebDriver** will be used for testing the integration of front-end components with the back-end services.
- **Acceptance Testing** will be performed by selected end users (both teachers and students) with assistance from the test manager and the development team leader. The testing will focus on ensuring that the platform meets the user's needs in real-world scenarios, such as searching for thesis papers, consulting with teachers, and collaborating on problem-solving. Acceptance testing will be conducted in parallel with the existing online tools and resources, such as Google Scholar or other research platforms, to validate the EA system's effectiveness over a period of one month after integration testing concludes.

This structured testing approach will ensure that the Education Assistant platform meets its objectives of providing a seamless, collaborative environment for students and teachers.

6.2 Test Tools

The test tools to be used for the Education Assistant (EA) project include Selenium WebDriver for automated UI testing and Mocha for unit and integration testing. These tools will help ensure that the software functions correctly, meets requirements, and maintains quality throughout the development process.

- **Selenium WebDriver** will be used for automating the testing of web interfaces. It will allow testers to simulate user actions on the Education Assistant platform, such as logging in, accessing resources, and performing consultations. This ensures the platform's user interface behaves as expected across different browsers and environments. Selenium will also be integrated into a continuous testing framework to catch issues early during the development cycle.

- **Mocha** will be used for unit and integration testing of the platform's backend. It provides a framework for running tests on server-side logic, ensuring that individual components of the Education Assistant system, such as the user registration and resource upload features, work correctly in isolation and when integrated together. This ensures backend functionality is robust and error-free.
- **Google Sheets** will be used for tracking test cases, bugs, and overall test progress. It will allow the team to maintain a clear overview of test results and follow up on defects efficiently.

The initial prototypes for the user registration and consultation screens will be developed using standard web design tools such as Figma and will be reviewed by the project stakeholders before final development and testing begin. These prototypes will help ensure that the user interface is intuitive and meets user expectations before being subjected to functional tests with Selenium.

7. TEST CASES/TEST ITEMS

*****View Excel file for test cases*****

8. ITEM PASS/FAIL CRITERIA

1. User Login

- Pass: Users can successfully log in with valid credentials and access their respective interfaces (Student/Teacher).
- Fail: Users cannot log in with valid credentials or experience errors during the login process.

2. User Registration

- Pass: Users can register by providing all mandatory information (Name, DOB, Gender, Phone number, Email, Address, Field of Study) without errors, and the system accurately checks for username availability and password strength.
- Fail: Users are unable to register due to missing mandatory fields, username conflicts, or password requirements not being met.

3. Profile Access

- Pass: After successful login, users can access their profiles and perform all related actions (view, update information, etc.).
- Fail: Users cannot access their profiles or perform required actions, such as updating their information.

4. Consultation Requests

- Pass: Students can successfully request consultations with teachers, and teachers can confirm or reschedule these requests.
- Fail: Consultation requests are not processed, or notifications are not sent to users about confirmations or changes.

5. Posting and Collaboration

- Pass: Teachers can post resources, research papers, and collaboration requests, while students can post questions or problems and receive feedback.
- Fail: Posts are not saved, comments cannot be made, or notifications for collaborations are not sent.

6. Search Functionality

- Pass: Users can successfully search for resources, publications, or user profiles and receive accurate results based on their queries.

- Fail: Searches yield no results when they should or display incorrect information.
- 7. **Resource Access**
 - Pass: Users can access and download available resources and materials without issues.
 - Fail: Users encounter errors or restrictions when trying to access resources.
- 8. **Donation Process**
 - Pass: Users can successfully make donations, and confirmation messages are displayed for both registered and non-registered users.
 - Fail: Donation attempts fail due to errors in form submission, payment processing, or incorrect information.

The test process will conclude once each functional requirement passes its criteria, ensuring the system operates as intended. The project will proceed to the next phase when the test manager confirms the functionality and performance of the system meets the established standards.

9. TEST DELIVERABLES

The following deliverables will be created during the testing process for the **Education Assistant (EA)** project:

- **Acceptance Test Plan:** This plan will outline the approach for validating the system against end-user expectations. It will define the scenarios under which teachers and students will test the system, including registration, searching for resources, downloading content, and collaborating on solutions. The success criteria will include user satisfaction, system usability, and the platform's ability to meet educational needs.
- **System/Integration Test Plan:** The integration of various modules (like user authentication, consultation scheduling, resource sharing, and notifications) will be tested to ensure seamless interactions between components. This test plan will include use cases where students and teachers perform end-to-end tasks, such as logging in, accessing resources, and consulting with one another.
- **Unit Test Plans/Turnover Documentation:** Unit testing will focus on validating the functionality of individual modules, such as the registration form, search functionality, and user roles (student vs. teacher). The developers will create test cases for each module and provide turnover documentation that includes test cases, defect logs, sample outputs, and other relevant details to the test manager.
- **Screen Prototypes:** Screen designs will be created and shared before development for user interface (UI) components such as the student dashboard, teacher consultation interface, and the resource repository. These prototypes will serve as a reference for both development and testing teams to ensure consistency.
- **Report Mock-ups:** EA will generate reports for teachers on student progress, consultations completed, and resource usage. Mock-ups of these reports will be shared with stakeholders before development to ensure they meet user needs. These mock-ups will be used in acceptance and system testing.
- **Defect/Incident Reports and Summaries:** During testing, any identified issues or defects will be documented in detail, including the steps to reproduce, severity, and status. Summaries of defects will be generated periodically to help track progress in fixing these issues.
- **Test Logs and Turnover Reports:** The test logs will capture every test case executed, including the outcome, date, time, and the tester responsible. These logs will be critical for reviewing testing progress and verifying that all functionalities have been thoroughly tested. Turnover reports will accompany the logs, documenting the completion of each testing phase.

10. STAFFING AND TRAINING NEEDS

For the **EA project**, adequate staffing and proper training will be essential to ensure smooth and thorough testing processes.

- **Staffing:**

Ideally, there will be one full-time tester dedicated to system/integration and acceptance testing phases. Initially, the tester may be assigned part-time duties for activities like reviewing system designs and participating in early reviews. However, as the project progresses (around four months in), this individual will need to transition into a full-time role to manage the workload associated with thorough system and acceptance testing. If a dedicated tester is unavailable, the project manager/test manager will assume the testing role to ensure coverage. In that case, the responsibility of conducting tests will be shared between the manager and the development team.

- **Training:**

Proper training will be crucial for various roles in the project:

- **Developers and Testers:** They will need training on the operational aspects of the EA platform, including the core features such as user roles, resource sharing, and consultation scheduling. Additionally, they will require training on the testing tools (e.g., Selenium WebDriver and Mocha) and how to create and execute automated test scripts.
- **Operations Staff:** Once the EA system is ready for deployment, the operations staff (e.g., system administrators) will need training on maintaining the system, managing user data, and troubleshooting basic issues related to performance or functionality.
- **End Users (Students and Teachers):** The training for students and teachers will focus on guiding them through the EA platform's interface, from logging in and registering to accessing resources, consulting with others, and collaborating on research.

Training materials may include user manuals, video tutorials, and live demonstrations.

11. RESPONSIBILITIES

A clear definition of responsibilities is crucial for the successful execution of testing processes in the **EA project**:

- **Project Manager/Test Manager:** The project manager will assume overall responsibility for the testing phase. They will ensure that testing activities remain on schedule, manage resource allocation, and address any obstacles encountered by the testing team. Additionally, they will coordinate between the testing and development teams and ensure that all test cases are covered adequately.
- **Developers:** Developers will be responsible for performing unit testing on the components they develop. This includes writing test cases for each function, module, or feature they implement. After unit testing, developers will hand over the module to the test manager along with the unit test results and defect logs for system and integration testing.

- **Testers:** Testers will execute system, integration, and acceptance tests. They will ensure that the application behaves as expected in real-world scenarios. They will document defects found during the testing process and will maintain detailed logs of test cases and results. Testers will also verify that all defects have been resolved before moving to the next phase of testing.
- **End Users (Students and Teachers):** End users will participate in the acceptance testing phase, where they will interact with the EA platform under real-world conditions. They will provide feedback on usability, performance, and any issues they encounter. Their input will be critical in validating that the system meets its intended goals.
- **Training Staff:** Trainers will be responsible for developing and delivering training materials for developers, testers, operations staff, and end users. They will conduct workshops, create user guides, and facilitate any onboarding sessions needed to ensure proper usage of the system.

| | TM | PM | Dev Team | Test Team | Client |
|--|-------------------------|-----------------------|--------------------------|--------------------------|-------------------------|
| Acceptance Test Documentation & Execution | Primary | Coordination, Support | Support | Execution, Support | Feedback, Participation |
| System/Integration Testing Documentation & Execution | Lead, Documentation | Coordination | Support | Execution, Documentation | Feedback |
| Unit Test Documentation & Execution | Review | Review | Primary, Execution | Review | |
| System Design Reviews | Participation | Lead, Coordination | Primary, Feedback | Feedback | Feedback |
| Detail Design Review | Participation | Lead, Coordination | Primary, Feedback | Feedback | Feedback |
| Test Procedures & Rules | Primary, Development | Review, Approval | Feedback | Execution, Compliance | |
| Screen & Report Prototype Reviews | Feedback | Coordination | Primary, Development | Feedback | Feedback, Approvals |
| Change Control & Regression Testing | Coordination, Execution | Coordination | Feedback, Implementation | Primary, Execution | Feedback |
| | | | | | |

- **TM (Test Manager)** is primarily responsible for all test-related activities, including writing and executing test plans, and managing change control and regression testing.
- **PM (Project Manager)** oversees the project coordination and participates in reviews to ensure that all activities align with the project goals.
- **Dev Team** is primarily responsible for the actual creation of design elements, executing unit tests, and implementing changes based on feedback from testing.

- **The Test Team** is mainly involved in the execution of tests, compliance with test procedures, and providing feedback on prototypes and system integration tests.
- **Client** participates primarily in acceptance tests, provides feedback on design reviews, and approves prototypes.

This distribution of responsibilities ensures that all aspects of the project are covered by the appropriate roles, promoting a thorough and effective testing process.

12. TESTING SCHEDULE

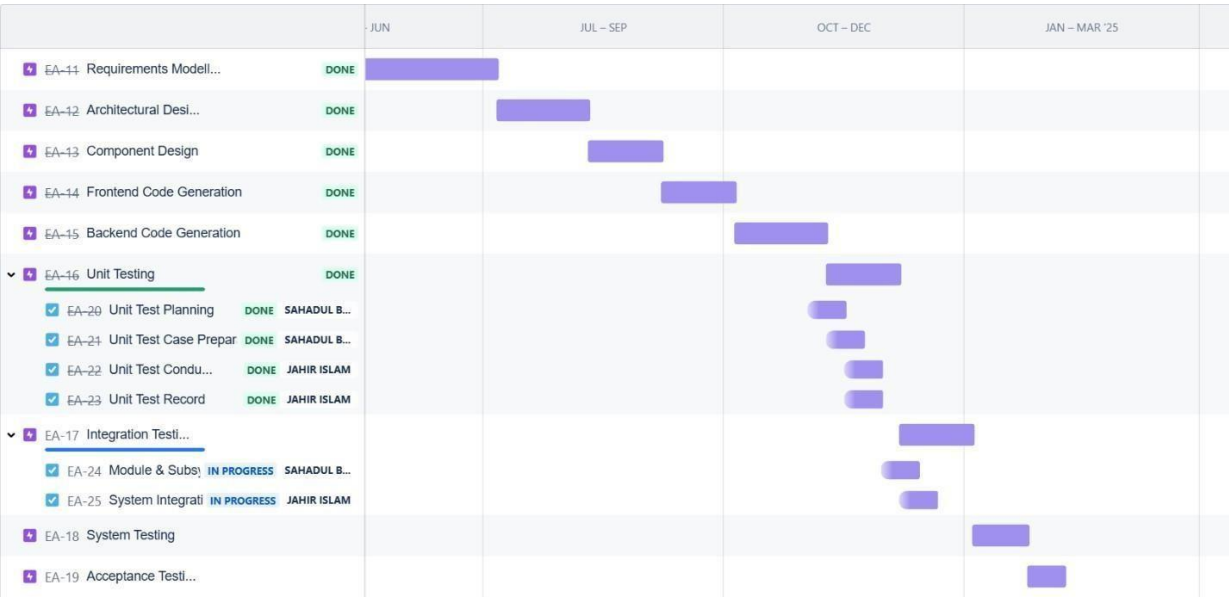


Figure7: total work distribution timeline (jira)

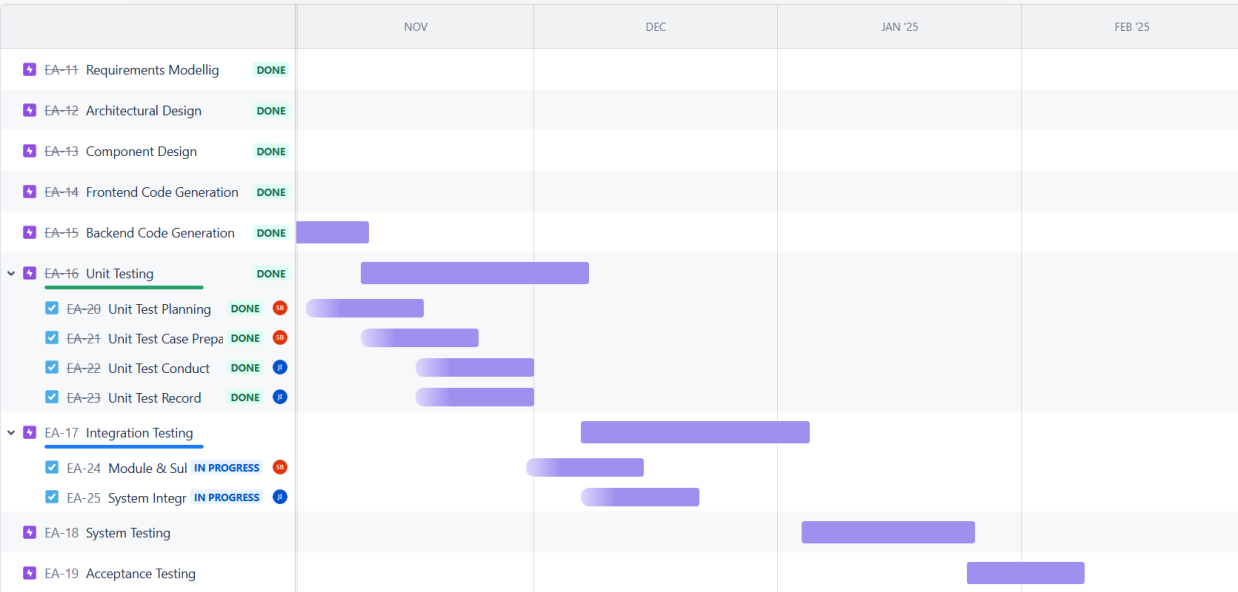


Figure8: test schedules (jira)



















| <input type="checkbox"/> | Type | # Key | Summary | Status | Assignee | Due date |
|--------------------------|---|----------------|--------------------------|-------------|---|--------------|
| <input type="checkbox"/> |  | Jahir islam | | | | |
| <input type="checkbox"/> |  | EA-2 | Architectural Design | DONE |  | Aug 10, 2024 |
| <input type="checkbox"/> |  | EA-4 | Backend Code Generation | DONE |  | Oct 5, 2024 |
| <input type="checkbox"/> |  | EA-6 | Unit Testing | IN PROGRESS |  | Dec 7, 2024 |
| <input type="checkbox"/> |  | EA-8 | System Testing | SCHEDULED |  | Jan 25, 2025 |
| <input type="checkbox"/> |  | EA-12 | Architectural Design | DONE |  | Aug 10, 2024 |
| <input type="checkbox"/> |  | EA-14 | Frontend Code Generation | DONE |  | Oct 5, 2024 |
| <input type="checkbox"/> |  | EA-16 | Unit Testing | DONE |  | Dec 7, 2024 |
| <input type="checkbox"/> |  | EA-18 | System Testing | SCHEDULED |  | Jan 25, 2025 |
| <input type="checkbox"/> |  | sahadul bengin | | | | |

Figure9: work distribution of a worker (jira)

13. PLANNING RISKS AND CONTINGENCIES

Risks are inherent in any project, and planning for these risks in the **EA project** will help mitigate potential delays or issues.

- **Limited Staff Availability:** The project may experience a shortage of qualified personnel to assist with testing, especially during the acceptance phase. If Reassigned Sales staff are unavailable or unable to participate in reviews, there may be delays in conducting user acceptance testing. In such cases, the testing schedule will be adjusted to accommodate staffing constraints, ensuring that all critical reviews and tests are completed without skipping important steps.
- **Budget Constraints:** Given the non-profit nature of the EA project, budgetary constraints may limit the availability of resources for testing. For example, there may not be enough funds to hire dedicated testers, resulting in additional responsibilities for the project manager. In such scenarios, test scope might need to be adjusted, focusing on the most critical functionalities while ensuring that end-users still receive a quality product.
- **Technical Issues:** Unforeseen technical issues such as performance bottlenecks, integration challenges, or platform stability could cause delays in the testing process. Contingencies like scheduling additional sprints for bug fixing, extending the testing timeline, or allocating extra development resources will be part of the risk mitigation strategy.

End User Availability for Acceptance Testing: Acceptance testing relies on the availability of teachers and students to test the system. Coordinating with end-users for testing sessions might be challenging, especially during exam periods or holiday seasons. If there are delays in securing end-user participation, acceptance testing will be postponed, and alternative methods (e.g., remote testing) may be explored.

| Risk | Category | Probability | Impact |
|---|----------|-------------|--------|
| 1. Size estimate may be significantly low | 1. PS | 1. 60% | 3 |
| 2. Larger number of users than planned | 2. PS | 2. 30% | 1 |
| 3. Less reuse than planned | 3. PS | 3. 70% | 4 |

| | | | | |
|-----|--|--------|---------|---|
| 4. | End-user-s resist system | 4. BU | 4. 40% | 2 |
| 5. | Delivery deadline will be tightened | 5. BU | 5. 50% | 3 |
| 6. | Funding will be lost | 6. CU | 6. 40% | 1 |
| 7. | Customer will change requirements | 7. PS | 7. 20% | 1 |
| 8. | Technology will not meet expectations | 8. TE | 8. 10% | 2 |
| 9. | Lack of training on tools | 9. DE | 9. 0% | 4 |
| 10. | Staff inexperienced | 10. ST | 10. 30% | 2 |
| 11. | Staff turnover will be high | 11. ST | 11. 60% | 3 |
| 12. | Not following planned principles/poor productivity | 12. ST | 12. 20% | 1 |
| 13. | Lack of ownership | 13. ST | 13. 30% | 3 |
| 14. | Unable to manage risk | 14. ST | 14. 10% | 1 |
| 15. | Scope-creep | 15. PR | 15. 0% | 4 |

PS: Project Scope - Risks related to the scope and requirements of the project.

BU: Business and User Impact - Risks that affect the users and business processes.

CU: Customer and Stakeholder Impact - Risks involving stakeholders or funding issues.

TE: Technical and Technology - Risks due to technical challenges or technology limitations.

DE: Development Environment - Risks related to the development tools and environment.

ST: Staffing and Team - Risks concerning the project team's experience, turnover, and management.

PR: Project Management - Risks associated with managing the project's scope and changes.

Impact value:

- 1- Catastrophic
- 2- Critical
- 3- Marginal
- 4- Negligible

- **Size estimate may be significantly low** (Project Scope)
 - **Probability:** 60%
 - **Impact:** High
 - **Contingency:** Regular scope reviews and adjustments to the project plan based on real-time feedback and metrics.
- **Larger number of users than planned** (Project Scope)
 - **Probability:** 30%
 - **Impact:** Low
 - **Contingency:** Scale infrastructure in phases to accommodate growth; monitor system performance continuously.

- **Less reuse than planned** (Project Scope)
 - **Probability:** 70%
 - **Impact:** High
 - **Contingency:** Increase modular design practices and allocate time for refactoring and integration tests.
- **End-user resistance to system** (Business and User Impact)
 - **Probability:** 40%
 - **Impact:** Medium
 - **Contingency:** Implement change management strategies and user training sessions to increase acceptance.
- **Delivery deadline will be tightened** (Business and User Impact)
 - **Probability:** 50%
 - **Impact:** High
 - **Contingency:** Use agile methodologies to prioritize deliverables and maintain flexibility in project timelines.
- **Funding will be lost** (Customer and Stakeholder Impact)
 - **Probability:** 40%
 - **Impact:** Low
 - **Contingency:** Develop a phased funding approach and maintain continuous stakeholder engagement to secure financial commitment.
- **Customer will change requirements** (Project Scope)
 - **Probability:** 20%
 - **Impact:** Low
 - **Contingency:** Maintain a flexible project scope and use iterative feedback loops to accommodate changes.
- **Technology will not meet expectations** (Technical and Technology)
 - **Probability:** 10%
 - **Impact:** Medium
 - **Contingency:** Conduct early prototype testing and keep abreast of new technologies that could replace underperforming tools.
- **Lack of training on tools** (Development Environment)

- **Probability:** 0%
- **Impact:** High
- **Contingency:** Schedule regular training sessions and ensure documentation is up-to-date and accessible.
- **Staff inexperienced** (Staffing and Team)
 - **Probability:** 30%
 - **Impact:** Medium
 - **Contingency:** Pair programming, mentoring programs, and professional development opportunities.
- **Staff turnover will be high** (Staffing and Team)
 - **Probability:** 60%
 - **Impact:** High
 - **Contingency:** Improve workplace environment and benefits to retain staff; have a robust onboarding process for new hires.
- **Not following planned principles/poor productivity** (Staffing and Team)
 - **Probability:** 20%
 - **Impact:** Low
 - **Contingency:** Regular team meetings and reviews to ensure adherence to principles and address productivity issues.
- **Lack of ownership** (Staffing and Team)
 - **Probability:** 30%
 - **Impact:** High
 - **Contingency:** Assign clear roles and responsibilities and recognize contributions to foster a sense of ownership among team members.
- **Unable to manage risk** (Staffing and Team)
 - **Probability:** 10%
 - **Impact:** Low
 - **Contingency:** Establish a dedicated risk management team and conduct regular training on risk assessment techniques.
- **Scope-creep** (Project Management)
 - **Probability:** 0%

- **Impact:** High
- **Contingency:** Implement strict change control processes and regular project scope reviews.

These contingencies help in preparing the project management team to handle uncertainties effectively and ensure that the project remains on track despite potential setbacks.

14. APPROVALS

| | |
|-------------------------------|--|
| Project Sponsor | |
| Development Management | |
| RS Test manager | |
| RS Development Team Manager | |
| Reassigned Sales | |
| Order Entity EDI Team Manager | |