

Deliverable 1 Report

1. Team Information:

Project Title: TestVault

Group Name: Fantastic Six

Group Members:

- (1) Md Afikur Rahman Khan
- (2) Arup Datta
- (3) Jarin Tasnim Ishika
- (4) Nhi Le
- (5) Vaishnavi Sabna
- (6) Rachelle Blosser

2. Project description:

Detailed description: TestVault is a Test Case Management System designed to create and organize test cases efficiently for seamless collaboration between testers, developers, and managers. Creating, updating, and deleting test cases is easy with the user-friendly responsive design from any type of device: laptop, phone, or tablet. Test cases may be organized into modules and prioritized for searching. Summarized reports and alerts on upcoming deadlines are sent to users.

Key features:

- (1) Create test cases
- (2) Edit test cases
- (3) Delete test cases
- (4) Organize test cases into modules
- (5) Clone test cases
- (6) Use test cases across different projects
- (7) Attach to test case
 - Screenshots
 - Files
 - Error Logs
- (8) Tag test cases
 - Priority
 - Module
 - Functionality
- (9) Filter and search test cases by
 - Tags
 - Priority
 - Status
- (10) Assign test cases to team members
- (11) Role-based access control
 - Admin
 - Manager
 - Developer
 - Quality Assurance (QA)

- (12) Comment test cases with discussion threads
- (13) Display interactive dashboards with summary statistics
 - Pass/Fail rate
 - Total test cases
- (14) Export reports
 - Portable document format (PDF)
 - Comma separated value (CSV)
- (15) Alert users for overdue test run or pending reviews
- (16) Remind users with notifications for assigned tasks or deadlines

Languages: Java, JavaScript, Hyper Text Markup Language 5 (HTML5), Cascading Style Sheets 3 (CSS3)

Framework: Spring Boot

Others: Thymeleaf, jQuery

Version Control Platform: GitHub

Project Management Platform: Trello

Wire Framing and Design: Figma, Draw.io

GitHub Deliverables 3,4,5

- 3. README of directory (folder) structure of project

<https://github.com/my-unt-projects/TestVault/blob/main/README.md>

- 4. Directory structure for the project source code with planning documents, meeting minutes, and project reports

Note Deliverable 1:

<https://github.com/my-unt-projects/TestVault/blob/main/note-deliverable1.txt>

Source Code:

<https://github.com/my-unt-projects/TestVault/tree/main/src>

Documents such as Planning Documents, meeting minutes, reports are under:

<https://github.com/my-unt-projects/TestVault/tree/main/docs>

- 5. Initial set of meeting minutes (time, attendees, things discussed, etc.) (GitHub).

<https://github.com/my-unt-projects/TestVault/tree/main/docs/meetings>

6. Timeline

Deliverable 3 - Phase 1 (Core Functionalities)

Start Date: 02/25/2025

End Date: 03/17/2025

Duration: 3 weeks

High Level Task Breakdown

Week 1	Week 2	Week 3
<ul style="list-style-type: none"> • Create project structure • Integrate third-party libraries • Design Database (DB) schema • Finalize boilerplate • Testing 	<ul style="list-style-type: none"> • Create user interface (UI) design and HTML template for test case • Create Read Update (CRUD) operation • Code Back-end for create/update/delete of test cases • Integrate UI and backend model for test case • Testing 	<ul style="list-style-type: none"> • Code organization of test cases into different modules and projects • Code clone test cases from existing test cases • Testing • Create necessary docs

Deliverable 4 - Phase 2 (Feature Enhancements)

Start Date: 03/18/2025

End Date: 04/07/2025

Duration: 3 weeks

High Level Task Breakdown

Week 1	Week 2	Week 3
<ul style="list-style-type: none"> • Code to add attachments to test cases • Code to tag test cases and filtering capabilities 	<ul style="list-style-type: none"> • Code role-based access control • Test • Add seed data for test cases and distinct types of users 	<ul style="list-style-type: none"> • Code test case assignment to team members and send notification email to the assignee • Write documentation • Integration Test • Performance Test

Deliverable 5 - Phase 3 (Final Product)

Start Date: 04/08/2025

End Date: 04/28/2025

Duration: 3 weeks

High Level Tasks Breakdown

Week 1	Week 2	Week 3
<ul style="list-style-type: none"> • Code discussion thread for each test case • Implement alerting and notifications 	<ul style="list-style-type: none"> • Build report generation in PDF, Excel, CSV etc. • Test • Document Integration/Performance 	<ul style="list-style-type: none"> • Review Final Product • Test full application • Write documentation • Finalize reports

<ul style="list-style-type: none"> Create Summary Dashboard 	/User Acceptance Test (UAT) <ul style="list-style-type: none"> Release Application Beta Version 	
--	--	--

Gantt Chart

The following diagram shows the number of weeks to complete each deliverable & phase. We have already completed deliverable-1 (Project Proposal and Planning) and progressed to Deliverable-2 (System Specification & Requirements).

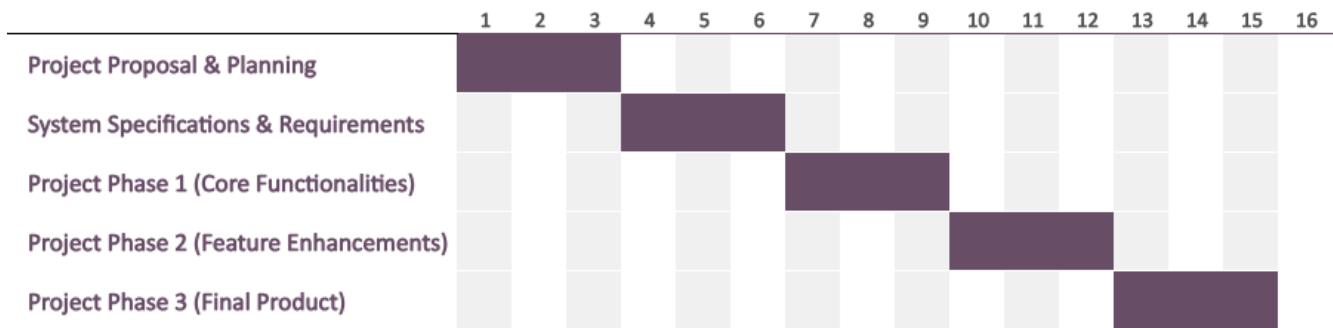


Figure: High Level Deliverable Gantt Chart

7. Risk Management

(1) Not Achieving on Time Delivery

- Monitoring risk: Visualize tasks within a timeline with tools, the Kanban board on Trello and the Gantt chart. Prompt and consistent updating of task status by team members is necessary for visualization tools to be useful.
- Reevaluating risk: Benchmark intermediate goals and individual tasks. Evaluate from the deadline backwards to the present.
- Planning contingency: Focus on priority features. Minimize complexity of the project. Devote more man hours to completion of the project.

(2) Performance Issues

- Monitoring risk: Track database query executing time regularly. Check database performance for complicated database queries such as joining multiple tables and getting large data downloads from database.
- Reevaluating risk: Perform testing on new features and analysis before and after database performance on identified page loads. Perform load testing and use caching mechanisms where possible.
- Planning contingency: Handle errors efficiently for complex queries. Deploy fallback mechanisms such as automated data backup on a weekly or daily frequency.

(3) Code Quality and Maintainability

- Monitoring risk: Define code static strategy before starting the project.
- Reevaluating risk: Check codebase to ensure code quality regularly.
- Planning contingency: Schedule sessions for code reviews with team members or with Teaching Assistant as needed.

8. Team Member Roles Descriptions

Project Management Lead

Maintain the project timeline and deliverable and ensure the project quality. Ensure the team meets deadlines and uses GitHub and Trello effectively. Coordinate meetings and communicate among the team members. Discuss with each lead in the team and monitor their progress. Track any risks and mitigate risks effectively so there is no bottleneck in the project.

Requirements Lead

Support project idea creator in the writing of software requirement specifications. Describe features in the most unitary, non-divisible way which satisfy user and developer goals for the utility of the software. Prioritize features according to importance and time.

Design Lead

Define the user experience and user interface structure of the platform. Ensure the platform meets the requirements of the course project. Discuss with team members team's ability and the design. Create technical design documents.

Implementation Lead for Front End

Ensure the UI is simple and consistent among modules. Develop own user interface tasks and review other team members' work and integrate to the application effectively. Ensure the application is responsive across desktop, phone, and tablet. Ensure the UI is integrated effectively with the backend application interfaces (APIs). Finally make sure the UI follows the design and uses effective rendering techniques.

Implementation Lead for Back End

Ensure the backend (BE) functions properly with core features correctly, meaning the feature works as per the requirements. Configure libraries and optimize database queries for better performance. Maintain code quality, ensuring efficiency and optimal performance.

Configuration Management Lead

Ensure team follows GitHub branching strategies and commits code properly. Make sure that the configuration for the application is externalized and version controlled so that it can be easily changed across different deployment environments.

Testing Lead

Create a test plan including test cases that focuses on the quality and performance of the platform using Trello. Assign tasks to team members and track bugs. Communicate with team members to ensure all the bugs are fixed. Create a test document including test cases, reports, and fixed bugs.

Documentation Lead

Collect documentation written by team members of respective features. Present documentation in written word using a uniform style in accordance with technical writing conventions. Format documentation for readability suitable to both user and developer.

Demo and presentation Lead

Create a structured and engaging demonstration of TestVault. Highlight key functionalities and user experience. Develop a well-organized demo script and a high-quality recorded video walkthrough showcasing core features like test case management, role-based access, reporting, and integrations with tracking tools. Prepare a polished presentation with system workflows, UI interactions, and performance insights. Ensure seamless execution with rehearsals and contingency planning for the live demonstration.

System Administrator Lead

Manage server related configuration and cloud configuration. Integrate monitoring tools such as health check and uptime monitoring. Ensure the system is scalable and performance is optimized.

9. Member contribution table

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Md Afikur Rahman Khan	Initiated the project description. High level task break-down into phases. Updated Gantt chart. Added Project Development Lead & Implementation Lead Frontend Roles. Added description for Risk Management No. 2. Reviewed the full document.	17%	
Arup Datta	Added the backend lead responsibility Reviewed the backend requirements Analyzed risks	17%	
Jarin Tasnim Ishika	Initiated to create a structured and engaging demonstration of TestVault, highlighting its key functionalities and user experience	17%	
Nhi Le	Researched and designed the UI/user experience (UX) for the platform. Designed and structured the test case plan.	17%	
Vaishnavi Sabna	Added the configuration management lead part, Reviewed the document, Reviewed high level task-breakdown.	15%	
Rachelle Blosser	Edited Project Description & Feature List;	17%	

	Req & Documentation Lead Descriptions; Risk 1; Gantt Chart pre-update; Edited roles & risks		
--	--	--	--