VIGILBOARD

VigilNet

Version 1.0

10-26-2023

William Mahoney, David Abbot, Isaac Bamidele, Chance Currie
williammahoney@my.unt.edu
davidabbot@my.unt.edu
oluwafeyisayobamidele@my.unt.edu
chancecurrie@my.unt.edu

Project Proposal Document

Version 1.0

Contents

A. Project Overview 2
B. Current Problems and Proposed Solutions 2
C. Requirements 2
    1. Functional Requirements 2
    2. Non-Functional Requirements 3
    3. Constraints 4
D. Specifications 4
    1. Functional Requirements Specifications 4
    2. Non-Functional Requirements Specifications 5
E. Proposal 6
    1. Functional Requirements 6
    2. Non-Functional Requirements 7
    3. Timeline 8

© 2023 VigilNet

A. Project Overview

The purpose of this project is to address any potential security vulnerabilities existing in a healthcare database. The VigilBoard will provide insight to any potential issues regarding security to a healthcare management system. This will be accomplished by providing a vulnerability scan to the database and will give a report about an existing security issue. For the purpose of our project we will be creating the healthcare database as a test subject of the VigilBoard. The VigilBoard will integrate with [insert tool] to collect vulnerability information. User authentication will be used to ensure that only authorized members are able to use the dashboard. The dashboard will provide a visualization for the user to see the potential vulnerabilities, the severity of vulnerabilities and the potential assets that are affected. This will lead to a prioritization of vulnerabilities that need to be reconciled. The VigilBoard will provide recommendations to the security issues that are found and give detailed information about the existing vulnerabilities. This information will be an asset to the database owner as it will give a detailed explanation about any current vulnerabilities and how to address them.

B. Current Problems and Proposed Solutions

Healthcare databases and their accompanying management systems are usually outdated due to rules and regulations that prevent the upgrade to better systems. With outdated management systems, there are always security issues, and an attacker could be looking to exploit this, leading to a breach in confidential information, possible data loss, and much more depending on the intentions of the attacker. VigilBoard will run vulnerability scans on these outdated management systems, and it will generate a report of recommended solutions to best counteract any potential vulnerabilities that it finds. This will help the database owner better understand how they can protect their systems despite being outdated, and ensures that the proper steps of protecting an outdated system are followed.

C. Requirements

1. Functional Requirements

Phase Color Code

Blue - Plan Phase	Yellow - Develop Phase	Red - Build Phase
Purple - Test Phase	Pink - Release and Deliver	Green - Deploy Phase
Brown - Operate Phase	Grey - Monitor Phase	

ID	Functional Requirements	Team Member Responsible	Effort (in %)
WM1	Project planning	William Mahoney	5%
IB1	System Design	Isaac Bamidele	5%
DA1	Software Requirement Analysis	David Abbot	5%
CC1	Change Management Planning	Chance Currie	5%
CC2	Application code development	Chance Currie	25%

© 2023 VigilNet

WM2	Application code development	William Mahoney	15%
WM3	Code Review	William Mahoney	5%
IB2	Database Development	Isaac Bamidele	25%
DA2	Documentation	David Abbot	25%
CC3	Build	Chance Currie	5%
WM4	SAST	William Mahoney	10%
DA3	Dependency Vulnerability Checking	David Abbot	5%
IB3	Store Artifacts	Isaac Bamidele	10%
DA4	Release Packaging	David Abbot	5%
CC4	Build Configuration and Audit	Chance Currie	5%
DA5	System Test	David Abbot	10%
CC5	Unit Test	Chance Currie	10%
WM5	Manual Security Test	William Mahoney	20%
DA6	Integration test	David Abbot	10%
IB4	Database security test	Isaac Bamidele	10%
IB5	Test Audit	Isaac Bamidele	10%
CC6	Database functional test	Chance Currie	10%
WM6	Release Go/ No-go decision	William Mahoney	3%
WM7	Deliver Released Artifacts	William Mahoney	2%
DA7	Delivery Results Review	David Abbot	5%
IB6	Ops Team Acceptance	Isaac Bamidele	5%
CC7	Configuration Integration Testing	Chance Currie	5%
CC8	Post-deployment Security Scan	Chance Currie	15%
WM8	Post-deployment Checkout	William Mahoney	15%
DA8	Create linked clone of VM master image	David Abbot	7.5%
IB7	Database Installation	Isaac Bamidele	15%
DA9	Database artifact deployment	David Abbot	7.5%
CC9	Backup	Chance Currie	4%
IB8	Scale	Isaac Bamidele	5%
WM9	Load Balancing	William Mahoney	5%
DA10	Feedback	David Abbot	5%
WM10	Logging	William Mahoney	15%
DA11	Logging Analysis	David Abbot	7.5%
IB9	Database Monitoring and Security Auditing	Isaac Bamidele	7%
CC10	Log Auditing	Chance Currie	15%
IB10	System Performance Monitoring	Isaac Bamidele	7%
DA12	System Security Monitoring	David Abbot	7.5%

2. Non-Functional Requirements

ID	Non-Functional Requirements	Team Member Responsible	Effort (in %)
WM11	GUI Design	William	5%
CC11	Documentation (how to install/run)	Chance Currie	1%
IB11	Demo Video	Isaac Bamidele	1%

3. Constraints

- *Legal - Privacy concerns with regards to healthcare data.*
    a. *HIPAA - Mandatory HIPAA compliance is required to safeguard all patient private information, medical and non medical.*
    b. *PHI - Emphasize that the scans will not directly access the information in the databases but only provide information regarding the security of the database structure.*
- *Financial - NO BUDGET*
- *Availability - Must provide a consistent uptime to monitor vulnerabilities in the system. Must be available to admins when they need it for compliance and scheduled scans.*
- *Usability - System should be easy to use. Navigating the dashboard must remain clear.*
- *Expertise - This is the first time anyone in our team has participated in creating a technical project of this size.*
  b.

D. Specifications

1. Functional Requirements Specifications

ID	Functional Requirement Specification
WM1	Team Collaboration System - Discord
WM2	IDE - Visual Studio Code
WM3	Code quality review tool - Github code review tool
WM4	Static Application Security Test tool - Snyk, Github Security
WM5	Network security test tool - Nmap, Metasploit
WM6	CI/CD Orchestrator - Github Actions
WM7	Artifacts repository - Ironbank, Amazon Elastic Container Registry(ECR)
WM8	Test scripts - Smoke Tester
WM9	VM management capability - Amazon Elastic Load Balancing

© 2023 VigilNet

WM10	Logging - Graylog
DA1	Team Collaboration System/Issue Tracking System - Trello/Github
DA2	Document Editor - Word
DA3	Dependency Checking Tool - Coverity
DA4	Release Packaging Tool - Jenkins
DA5	Test Tool Suite - DbFit
DA6	Test Tool Suite - DbFit
DA7	Human activity - N/A
DA8	Virtualization Manager - Veeam ONE
DA9	Artifact Repository, Database Automation Tool - Iron Bank, DbFit/LiquiBase
DA10	Log Aggregator & Log Analysis and Auditing - Graylog
DA11	ISCM, Issue Tracking System - Jit, Github/Trello
CC1	Project Management System - Trello
CC2	IDE - Visual Studio Code
CC3	Build Tool - Visual Studio Code
CC4	SAST Tool - GitHub
CC5	Test Tool Suite / Test Tool Coverage - Embunit (assuming program is written in C or C++)
CC6	Database Security Scan and Test Tool - Dbfit
CC7	Configuration Integration Test Tool - GitHub
CC8	Security Compliance Tool - Metasploit
CC9	Backup Management - GitHub
CC10	Log Aggregator - Graylog
IB1	Software System design tool - Draw.io
IB2	Database Development Software- MySQL
IB3	Artifact Repository - Iron Bank, Amazon Elastic Container Registry(ECR)
IB4	Database Security Test tool - Scuba Database Vulnerability Scanner
IB5	Test Management Tool -Tuskr
IB6	Human Activity - N/A
IB7	Database Automation Tool - LiquiBase
IB8	VM Management Capability - Veeam ONE
IB9	Database Security Monitoring Tool -Tuskr
IB10	Operation Monitoring Tool -ManageEngine Applications Manager

2. Non-Functional Requirements Specifications

ID	Non-Functional Requirement Specification
IB11	Demo Video -Vimeo
CC11	Documentation (how to install/run) - Google Docs
WM11	GUI Design - powebi/tableu/databox

E. Proposal

1. Functional Requirements

[List the outputs for your functional requirements with the team member responsible to meet the functional requirement.]

ID	Functional Requirements Output	Team Member Responsible
WM1	Organized teamwork	William Mahoney
WM2	source code	William Mahoney
WM3	code review comments	William Mahoney
WM4	static code scan report	William Mahoney
WM5	vulnerability report	William Mahoney
WM6	go/no go decision	William Mahoney
WM7	new release in artifact repository	William Mahoney
WM8	test results	William Mahoney
WM9	balanced resource utilization	William Mahoney
WM10	event logs	William Mahoney
DA1	Feature requirements, Performance requirements, Privacy requirements, Security requirements	David Abbot
DA2	Application Programming Interface (API) documentation	David Abbot
DA3	Vulnerability report	David Abbot
DA4	Released package w/ checksum and digital signature	David Abbot
DA5	System performance test results	David Abbot
DA6	Unit integration test results	David Abbot
DA7	Push Go/No-Go Decision	David Abbot
DA8	VM master linked clone VM instance	David Abbot
DA9	Running Database	David Abbot

© 2023 VigilNet

DA10	Backlog or updated requirements	David Abbot
DA11	Alerts and remediation report	David Abbot
DA12	Vulnerabilities, Incompliance findings, assessment and recommendations, warnings and alerts	
CC1	Screenshots of Trello board	Chance Currie
CC2	Source Code	Chance Currie
CC3	Build Report	Chance Currie
CC4	Version Controlled Build Report	Chance Currie
CC5	Test Report	Chance Currie
CC6	Database Test Results	Chance Currie
CC7	Configuration Results	Chance Currie
CC8	Security Vulnerability Findings	Chance Currie
CC9	Backup Data and/or Image	Chance Currie
CC10	Audit Logs	Chance Currie
IB1	Functional design diagram	Isaac Bamidele
IB2	Test Data	Isaac Bamidele
IB3	Version Controlled Database Artifact	Isaac Bamidele
IB4	Database Test Results	Isaac Bamidele
IB5	Test Audit log	Isaac Bamidele
IB6	Accepted release package	Isaac Bamidele
IB7	Pictures of running Database system	Isaac Bamidele
IB8	Optimized resource allocation	Isaac Bamidele
IB9	Warnings and Alerts logs	Isaac Bamidele
IB10	Reports of recommended actions for warnings and alerts.	Isaac Bamidele

2. Non-Functional Requirements

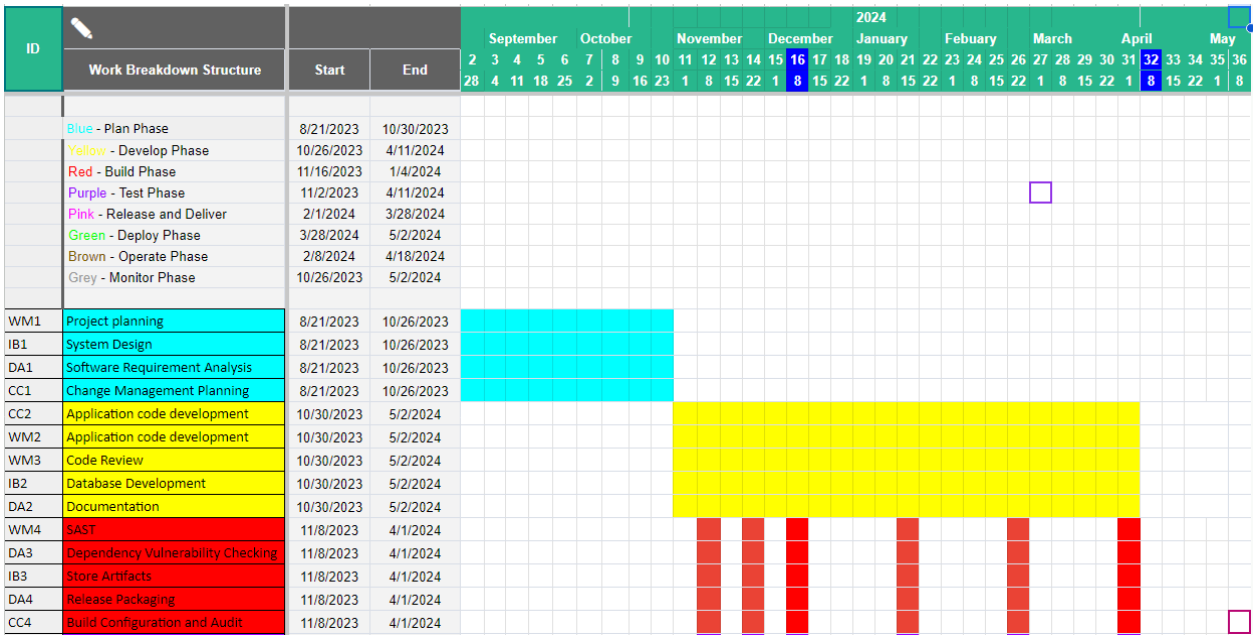
[List your non-functional requirements with the team member responsible to meet the non-functional requirement.]

ID	Non-Functional Requirements	Team Member Responsible
WM11	Beautiful and functional GUI	William Mahoney
CC11	Word document or PDF	Chance Currie
IB11	Beautiful and articulate demo video	Isaac Bamidele

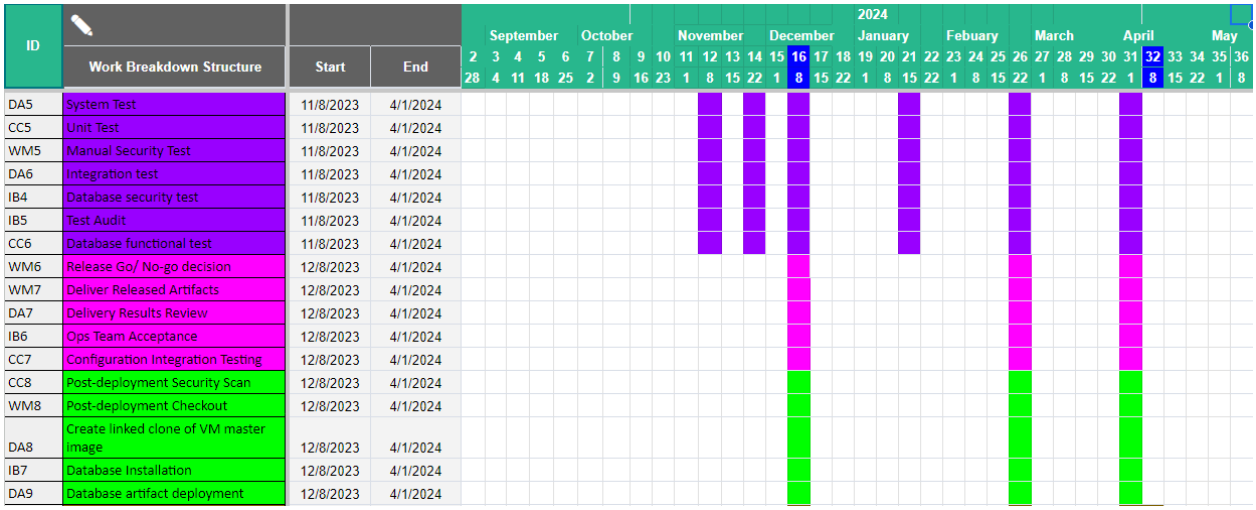
3. Timeline

[Provide a Gantt chart that shows the start and end date of each of the requirements for each team member. Use start of the Fall semester as the start date and third week of April (approximately) for the end date of the project .]

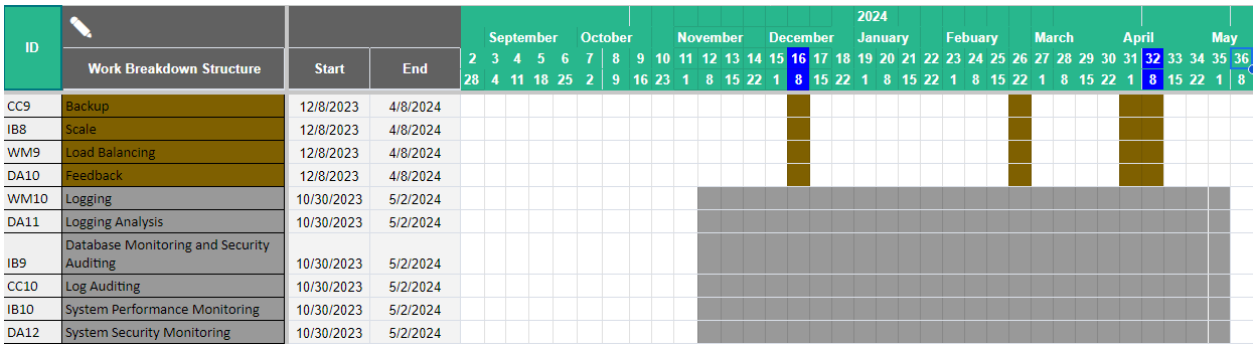
Excel Spreadsheet also attached in Canvas in order to have a better view of the Gantt Chart overall.



ID	Work Breakdown Structure	Start	End
	Blue - Plan Phase	8/21/2023	10/30/2023
	Yellow - Develop Phase	10/26/2023	4/11/2024
	Red - Build Phase	11/16/2023	1/4/2024
	Purple - Test Phase	11/2/2023	4/11/2024
	Pink - Release and Deliver	2/1/2024	3/28/2024
	Green - Deploy Phase	3/28/2024	5/2/2024
	Brown - Operate Phase	2/8/2024	4/18/2024
	Grey - Monitor Phase	10/26/2023	5/2/2024
WM1	Project planning	8/21/2023	10/26/2023
IB1	System Design	8/21/2023	10/26/2023
DA1	Software Requirement Analysis	8/21/2023	10/26/2023
CC1	Change Management Planning	8/21/2023	10/26/2023
CC2	Application code development	10/30/2023	5/2/2024
WM2	Application code development	10/30/2023	5/2/2024
WM3	Code Review	10/30/2023	5/2/2024
IB2	Database Development	10/30/2023	5/2/2024
DA2	Documentation	10/30/2023	5/2/2024
WM4	SAST	11/8/2023	4/1/2024
DA3	Dependency Vulnerability Checking	11/8/2023	4/1/2024
IB3	Store Artifacts	11/8/2023	4/1/2024
DA4	Release Packaging	11/8/2023	4/1/2024
CC4	Build Configuration and Audit	11/8/2023	4/1/2024



ID	Work Breakdown Structure	Start	End
DA5	System Test	11/8/2023	4/1/2024
CC5	Unit Test	11/8/2023	4/1/2024
WM5	Manual Security Test	11/8/2023	4/1/2024
DA6	Integration test	11/8/2023	4/1/2024
IB4	Database security test	11/8/2023	4/1/2024
IB5	Test Audit	11/8/2023	4/1/2024
CC6	Database functional test	11/8/2023	4/1/2024
WM6	Release Go/ No-go decision	12/8/2023	4/1/2024
WM7	Deliver Released Artifacts	12/8/2023	4/1/2024
DA7	Delivery Results Review	12/8/2023	4/1/2024
IB6	Ops Team Acceptance	12/8/2023	4/1/2024
CC7	Configuration Integration Testing	12/8/2023	4/1/2024
CC8	Post-deployment Security Scan	12/8/2023	4/1/2024
WM8	Post-deployment Checkout	12/8/2023	4/1/2024
DA8	Create linked clone of VM master image	12/8/2023	4/1/2024
IB7	Database Installation	12/8/2023	4/1/2024
DA9	Database artifact deployment	12/8/2023	4/1/2024



ID	Work Breakdown Structure	Start	End
CC9	Backup	12/8/2023	4/8/2024
IB8	Scale	12/8/2023	4/8/2024
WM9	Load Balancing	12/8/2023	4/8/2024
DA10	Feedback	12/8/2023	4/8/2024
WM10	Logging	10/30/2023	5/2/2024
DA11	Logging Analysis	10/30/2023	5/2/2024
IB9	Database Monitoring and Security Auditing	10/30/2023	5/2/2024
CC10	Log Auditing	10/30/2023	5/2/2024
IB10	System Performance Monitoring	10/30/2023	5/2/2024
DA12	System Security Monitoring	10/30/2023	5/2/2024