US Army Corps of Engineers (USACE) Work Request Application

CMSC495 – Team 3 PROJECT schedule & managment PLAN

Table of Contents

[1. Project Plan 6](#_Toc90290709)

[1. Project Summary 6](#_Toc90290710)

[1.1 Problem, Purpose, and Solution: 6](#_Toc90290711)

[1.2 Assumptions and Constraints 6](#_Toc90290712)

[2. Project Organizational Structure 6](#_Toc90290713)

[**2.1 Roles and Responsibilities** 6](#_Toc90290714)

[**2.1.1. Project Manager — Dave Leake** 6](#_Toc90290715)

[**2.1.2. Requirements Manager/Technical Writer — Will Aurelien** 7](#_Toc90290716)

[**2.1.3. Test Director — Will Aurelien** 7](#_Toc90290717)

[**2.1.4. Software Designer — Ian Oliver** 7](#_Toc90290718)

[**2.1.5. User Experience/Training Manager — Will Aurelien** 7](#_Toc90290719)

[3. Scope Management Plan 8](#_Toc90290720)

[3.1 Scope Statement 8](#_Toc90290721)

[3.2 Requirements Identification and Modification 8](#_Toc90290722)

[3.3 Work Breakdown Structure 9](#_Toc90290723)

[3.4 Sponsor Acceptance 9](#_Toc90290724)

[**3.5 Scope Control** 10](#_Toc90290725)

[4. Schedule Management Plan 10](#_Toc90290726)

[5. Cost Management Plan 10](#_Toc90290727)

[5.1 Introduction 10](#_Toc90290728)

[5.2 Cost Management Approach 11](#_Toc90290729)

[5.3 Measuring Project Costs 11](#_Toc90290730)

[5.4 Reporting Format 11](#_Toc90290731)

[5.5 Cost Variance Response Process 12](#_Toc90290732)

[5.6 Cost Change Control Process 12](#_Toc90290733)

[6. Quality Management Plan 12](#_Toc90290734)

[7. Communication Management Plan 13](#_Toc90290735)

[7.1 Communication Plan 13](#_Toc90290736)

[8. Risk Management Plan 14](#_Toc90290737)

[9. Procurement Management Plan 14](#_Toc90290738)

[9.1 Introduction 14](#_Toc90290739)

[9.2 Procurement Management Approach 15](#_Toc90290740)

[9.3 Procurement Definition 15](#_Toc90290741)

[9.4 Type of Contract Vehicles to be Used 16](#_Toc90290742)

[9.5 Procurement Risks 16](#_Toc90290743)

[9.6 Procurement Risk Management 16](#_Toc90290744)

[9.7 Cost Determination 17](#_Toc90290745)

[9.8 Standardized Procurement Documentation 17](#_Toc90290746)

[9.9 Contract Approval Process 17](#_Toc90290747)

[9.10 Decision Criteria 18](#_Toc90290748)

[9.11 Vendor/Performer Management 18](#_Toc90290749)

[9.12 Performance Metrics for Procurement Activities 19](#_Toc90290750)

[10. Technical Process Plan 19](#_Toc90290751)

[10.1 Process Model 19](#_Toc90290752)

[10.2 Technical Requirements 19](#_Toc90290753)

[2. Requirements Specification 20](#_Toc90290754)

[Introduction 20](#_Toc90290755)

[Purpose 20](#_Toc90290756)

[Scope 20](#_Toc90290757)

[Definitions, acronyms, abbreviations 20](#_Toc90290758)

[References 21](#_Toc90290759)

[Overall Description 22](#_Toc90290760)

[Product Perspective 22](#_Toc90290761)

[A) System interfaces 26](#_Toc90290762)

[B) User interfaces 26](#_Toc90290763)

[C) Hardware interfaces 30](#_Toc90290764)

[D) Software interfaces 30](#_Toc90290765)

[E) Communications interfaces 31](#_Toc90290766)

[F) Memory 31](#_Toc90290767)

[G) Operations 31](#_Toc90290768)

[H) App adaptation requirements 31](#_Toc90290769)

[Product Functions 32](#_Toc90290770)

[User Characteristics 33](#_Toc90290771)

[Use Case 1.0**:** End User Submits a work request 33](#_Toc90290772)

[Use Case 2.0**:** Analysis of work requests 33](#_Toc90290773)

[Constraints 34](#_Toc90290774)

[Assumptions and Dependencies 34](#_Toc90290775)

[Specific Requirements 35](#_Toc90290776)

[Performance Requirements 35](#_Toc90290777)

[Logical Database Requirements 35](#_Toc90290778)

[Work Request Proposed ERD 35](#_Toc90290779)

[Design Constraints 37](#_Toc90290780)

[Standards Compliance 37](#_Toc90290781)

[Software System Attributes 37](#_Toc90290782)

[Reliability 37](#_Toc90290783)

[Availability 37](#_Toc90290784)

[Security 37](#_Toc90290785)

[Maintainability 37](#_Toc90290786)

[Portability 37](#_Toc90290787)

[3. Test Plan 38](#_Toc90290788)

[Introduction 38](#_Toc90290789)

[Application Components Tested 38](#_Toc90290790)

[Application Features Tested 38](#_Toc90290791)

[Types of Tests Needed 39](#_Toc90290792)

[Test Documentation 39](#_Toc90290793)

[Testing Environment and Needs 39](#_Toc90290794)

[Testing Schedule 40](#_Toc90290795)

[Testing Breakdown 40](#_Toc90290796)

[Manual Testing 40](#_Toc90290797)

[User Scripts 40](#_Toc90290798)

[Unit Testing 41](#_Toc90290799)

[Acceptance Testing 42](#_Toc90290800)

[Testing Criteria 42](#_Toc90290801)

[Potential Risks 43](#_Toc90290802)

[Defects and Risks 43](#_Toc90290803)

[4. Usability Test Plan 44](#_Toc90290804)

[Document Overview 44](#_Toc90290805)

[Executive Summary 44](#_Toc90290806)

[Methodology 45](#_Toc90290807)

[Participants 45](#_Toc90290808)

[Training 45](#_Toc90290809)

[Procedure 45](#_Toc90290810)

[Roles 46](#_Toc90290811)

[Trainer 46](#_Toc90290812)

[Facilitator 46](#_Toc90290813)

[Data Logger 46](#_Toc90290814)

[Test Observers 46](#_Toc90290815)

[Ethics 47](#_Toc90290816)

[Usability Tasks 47](#_Toc90290817)

[Usability Metrics 47](#_Toc90290818)

[Scenario Completion 47](#_Toc90290819)

[Critical Errors 48](#_Toc90290820)

[Non-critical Errors 48](#_Toc90290821)

[Subjective Evaluations 48](#_Toc90290822)

[Scenario Completion Time (time on task) 48](#_Toc90290823)

[Usability Goals 49](#_Toc90290824)

[Completion Rate 49](#_Toc90290825)

[Error-free rate 49](#_Toc90290826)

[Time on Task (TOT) 49](#_Toc90290827)

[Subjective Measures 49](#_Toc90290828)

[Problem Severity 49](#_Toc90290829)

[Impact 50](#_Toc90290832)

[Frequency 50](#_Toc90290833)

[Problem Severity Classification 50](#_Toc90290834)

[Reporting Results 51](#_Toc90290835)

[5. Product Design 51](#_Toc90290836)

[Introduction 51](#_Toc90290837)

[Purpose of The Product Design Specification Document 51](#_Toc90290838)

[General Overview and Design Guidelines/Approach 51](#_Toc90290839)

[Assumptions 51](#_Toc90290840)

[Constraints 51](#_Toc90290841)

[Architecture Design 52](#_Toc90290842)

[Logical View 52](#_Toc90290843)

[Software Architecture 54](#_Toc90290844)

[Performance 54](#_Toc90290845)

[Alternate architecture (Depricated) 55](#_Toc90290846)

[System Design 55](#_Toc90290847)

[Use-Cases 55](#_Toc90290848)

[Database Design 55](#_Toc90290849)

[User Interface Design 55](#_Toc90290850)

[WORK REQUEST TAB 55](#_Toc90290851)

[NEW WORK REQUEST TAB 56](#_Toc90290852)

[Product Design Specification Approval 58](#_Toc90290853)

[6. Development History 58](#_Toc90290854)

[Appendix A – Project Charter 59](#_Toc90290855)

[Appendix B – Project Team 60](#_Toc90290856)

[Appendix C – Statement of Work 61](#_Toc90290857)

[Appendix D - Schedule 62](#_Toc90290858)

[Appendix E – Gantt Chart Example 63](#_Toc90290859)

[Appendix F- Scope Baseline 63](#_Toc90290860)

[References 64](#_Toc90290861)

# ­Project Plan

## 1. Project Summary

### 1.1 Problem, Purpose, and Solution:

The current USACE Europe District work requests are tracked on a SharePoint 2013 Custom Application. SharePoint 2013 is past end of life, and the district needs to develop a replacement for the Work Request application. Group 3’s solution will be developed on a lightweight web interface that will allow the district to submit Work Requests, automate cost calculations, and provide reports and analytics on work requests.

### 1.2 Assumptions and Constraints

It is assumed that the employees of the businesses that use our application will be familiar with how to navigate to a webApp and how to operate a basic graphical user interface designed for work requests.

The constraints will be that the end user is accessing the application from a desktop and not a mobile device as we will not be creating a responsive design for mobile device constraints. Outside of that, any desktop or laptop should be able to access and run our application.

## 2. Project Organizational Structure

This project’s organizational structure is split into several different units. The overall structure is led by the Project Manager, and each unit consists of a leader responsible for the management of that section. However, it is important to note that all team members are flexible and responsible for multiple duties throughout all units.

## **2.1 Roles and Responsibilities**

### **2.1.1. Project Manager — Dave Leake**

The Project Manager is responsible for acting as a liaison between the team and upper management (professor), including communication between teammates and unit leads. Additionally, the Project Manager is responsible for scheduling and ensuring that all deliverables are completed and submitted on time, as well as organizing and recording all necessary forms of contact and conversation among the team.

### **2.1.2. Requirements Manager/Technical Writer — Will Aurelien**

The Requirements Manager is responsible for the organization of all documentation and works with the Project Manager, the client, and all team members to ensure that all deliverables meet the requirement in the statement of work. The Requirements Manager for this team also takes the role of the Technical Writer and is responsible for creating and maintaining documentation required in the process of developing the software solution.

### **2.1.3. Test Director — Will Aurelien**

The Test Director is responsible for creating testing scenarios to ensure that all requirements for the software are met. Additionally, the Test Director is responsible for carrying out these scenarios and identifying any defaults in the program, as well as maintaining documentation of these tests and directing developers towards the issue that requires attention.

### **2.1.4. Software Designer — Ian Oliver**

The Software Designer is responsible for creating the design of the program in accordance with business, mathematical, and scientific principles that correlate with the client’s request. Additionally, the Software Designer is responsible for working with the Project Manager, and all developers to create a final, functional product as requested by the client.

### **2.1.5. User Experience/Training Manager — Will Aurelien**

The User Experience and Training Manager is responsible for managing the project from a front-end perspective as requested by the client. This involves working with the Software Designer and Project Manager to ensure that the front-end development meets the project’s requirements by closely assessing the development process to ensure that the program is being built to fulfill the requirements.

## 3. Scope Management Plan

### 3.1 Scope Statement

The goal of the Work Request Application is to provide a functional, user-friendly, low-maintenance management solution for the USACE Europe District. The intended product is a Work Request form generator and scheduler. Request Forms are filled with a custom-built wizard for one of three different categories:

* + - 1. Categories
         * Engineering
         * Construction
         * Environmental
      2. Work Request Information
         * General Information
         * Project Information
         * Assignment Type
         * Cost Distribution
         * Personnel Assignment
         * History

Schedules, general resources, timelines, and statuses of individual or all workorders can also be viewed from the Analytics section of the application. Functionally the same as reports, display only.

### 3.2 Requirements Identification and Modification

All application requirements were provided and documented by the project sponsor prior to the initiation of planning and design. Should the sponsor desire modification of these requirements, the Project Manager’s approval will be required after consultation with designers and/or developers. Due to timeline restrictions, significant modifications will not be possible.

### 3.3 Work Breakdown Structure



### 3.4 Sponsor Acceptance

Upon completion of the application, the sponsor will be provided a demonstration of all functionalities. The sponsor will be given a trial period, not to exceed 10 days, before acceptance or rejection is expected. The sponsor will ensure all requirements have been met and the system is functioning properly in the local environment. The sponsor will sign a formal acceptance letter before project closeout.

### **3.5 Scope Control**

All actions and effort will focus on the accomplishment of the stated goals, objectives, and requirements. This project is restricted by tight timeline constraints and therefore additional services, features, and functions will not be pursued.

## 4. Schedule Management Plan

The Work Request Application will take 8 weeks from development to deployment. Group 3 will follow a strict schedule to meet that deadline. To assure the team doesn’t fall behind, an Assignment Schedule was created (see Appendix C). The Assignment Schedule is an easy-to-follow weekly planner that identifies the team member leading the topic and its due date.

## 5. Cost Management Plan

### 5.1 Introduction

Group 3 Inc, Cost Management Plan for the costs on the Work Request Application will be managed throughout the project’s software development lifecycle (SDLC). This document sets the format and standards by which the project costs are measured, reported, and controlled. The Cost Management Plan:

* Identifies who is responsible for managing costs.
* Identifies who has the authority to approve changes to the project or its budget.
* How cost performance is quantitatively measured and reported upon.
* Report formats, frequency and to whom they are presented.

During the monthly synchronization meeting with upper management and the Project Sponsor and the Project Manager shall review the project’s cost performance, fund obligations, accruals, and expenditures to ensure execution is meeting defined benchmarks for the preceding month. The Project Manager is responsible for managing and reporting on the project’s cost throughout its software development lifecycle, cost deviations, de-obligations, and presenting the Project Sponsor with options for getting the project back on budget. The Project Sponsor has the authority to make changes to the project to bring it back within defined benchmarks and budget.

### 5.2 Cost Management Approach

Costs for the Work Request Application shall be managed at the fourth level of the Work Breakdown Structure (WBS). Control Accounts (CA) will be created at this level to track costs incurred by individual developers. Earned Value calculations for the CA’s will measure and manage the financial performance of the project. Although activity cost estimates are detailed in the work packages, the level of accuracy for cost management is at the fourth level of the WBS. Credit for work will be assigned at the work package level. Work started on work packages will grant that work package with 50% credit; whereas the remaining 50% is credited upon completion of all work defined in that work package. These work packages will have cost variance thresholds of ± 10% and of ± 20%. Once a variance reaches ± 20% it will require immediate corrective action by the Project Manager.

### 5.3 Measuring Project Costs

Performance of the Work Request Application will be measured using the Earned Value Management system. The following four Earned Value metrics will be used to measure projects cost performance.

* Schedule Variance (SV)
* Cost Variance (CV)
* Schedule Performance Index (SPI)
* Cost Performance Index (CPI)

|  |  |  |
| --- | --- | --- |
| Performance Measure | Yellow | Red |
| Schedule Performance Index (SPI) | between 90% - 80% or between 110% - 120% | less than 80% or greater than 120% |
| Cost Performance Index (CPI) | between 90% - 80% or between 110% - 120% | less than 80% or greater than 120% |

### 5.4 Reporting Format

Reporting for cost management will be included in the monthly project status reports. The Monthly Project Status Report shall include a section labeled, “Cost Management”. In this section, the Earned Value Metrics from the preceding month will be displayed. All cost variances outside of benchmarks in the Cost Management Plan will be addressed; to include all mitigations taken. Finally, Change Requests which are triggered based on deviations from defined benchmarks will be identified and tracked in the Project Monthly Status Report.

### 5.5 Cost Variance Response Process

The Control Thresholds for this project is a CPI or SPI of less than 20% or greater than 120%. If the project reaches one or both Control Thresholds, a Cost Variance Corrective Plan of Action & Milestones is required. The Project Manager will present the Project Sponsor with multiple Courses of Action (COAs) to mitigate the variances within five business days. Within three business days from when the Project Sponsor selects a COA, the Project Manager shall present the Project Sponsor with a formal Cost Variance Corrective Plan of Action & Milestones. The Cost Variance Corrective Plan of Action & Milestones shall detail the necessary corrective measures to bring the project back in defined benchmarks and the milestones by which the effectiveness of the actions in the plan will be measured. Upon acceptance of the Cost Variance Corrective Plan of Action & Milestones, it will then become part of the project plan and the project will be updated to reflect this.

### 5.6 Cost Change Control Process

The cost change control process will follow established policy detailed in the change request process. Approvals for project budget/cost changes must be reviewed by upper management and approved by the project sponsor.

**5.7 Project Budget**

Given that all team members of the Group 3. have agreed to voluntarily dedicate their time towards the creation of the Work Request Application and given that all resources acquired will be open source, the cost of this project will be $0.00, which reduces the need for a project budget.

## 6. Quality Management Plan

Quality Control will be the responsibility of every individual team member, as we each will be held accountable for our sections of the project. As a quality control safety net, Group 3’s Test Director will act as a backup quality control reviewer. On top of everyone performing individual quality control and the Test Director thoroughly testing the software solution, the code will be developed in accordance with the Google JavaScript Style guide.

## 7. Communication Management Plan

The Communications Management Plan will outline the necessary periodic communications between all parties involved in the project. This ensures all parties are up to date on progress and possible changes throughout the lifecycle of the project.

### 7.1 Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Communication | Medium | Frequency | Goal | Owner |
| **Project Team** | | | | |
| Project Status | Slack/Asana | Weekly (Monday) | Review status of project modules and discuss details of deliverables for the week | Project Manager |
| Deliverable Consolidation | Slack/Asana | Weekly (Friday) | Discuss consolidation details for deliverables and any issues or delays | Project Manager |
| Task Review | Slack/Asana | Weekly | Discuss feedback from deliverables submitted the prior week | Project Manager |
| Final Product Demonstration | Zoom | At end of project | Demonstrate and explain all functionality to project sponsor | Project Manager |
| Issue Resolution | Call/text | As needed | Resolve issues that could delay or prevent on-time deliverable submission | All |
| **Project Sponsor** | | | | |
| Deliverable Submission | UMGC Submission Folder | Weekly | Provide project status update and present tentative deliverable | Project Manager |
| Final Product Demonstration | Zoom | At end of project | Demonstrate and explain all functionality to project sponsor | All |
| Feedback | UMGC Discussion Board | Weekly | Receive feedback on weekly deliverables | All |

## 8. Risk Management Plan

The first step the team will take towards risk management is identifying all possible risks and planning the procedure to avoid the risks. This requires tracking potential changes to risk factors along the development process, as well as identifying solutions for any risks that may rise. In the case of any risk rising, the team will immediately report to the Project Manager, who maintains documentation of all risks and solutions through the development process.

To avoid risk in scope, the client will agree to the terms of development, which will be followed by the development team.

To avoid risk in communication and scheduling, all team members have agreed to dedicate eight weeks of their time to the development of the project and agreed upon a means of communication, as well as a flow of communication to direct any conversation towards the right channel.

To avoid quality risk, the development team has established certain requirements, such as frequent and clear communication, observing and following milestones and progress, and constant coordination between all team leads to ensure requirements are met on all ends of the development process. Further information is available in the Risk Register (Appendix G)

## 9. Procurement Management Plan

### 9.1 Introduction

The Procurement Management Plan sets the procurement framework by which the Work Request Application project will follow. This document shall serve as a guide for managing procurements throughout the software development life cycle of the EPS project and will be updated as acquisition needs change. This plan identifies performers and the items and/or services that will be procured from them, the types of contracts to be used in support of this project, i.e., Indefinite delivery/indefinite quantity (IDIQ) contracts and/or Support Service Contracts, the contract approval process, and decision-making criteria. Other items included in the procurement management plan include procurement risks and procurement risk management considerations; how costs will be determined; how standard procurement documentation will be used; and procurement constraints.

### 9.2 Procurement Management Approach

The Project Manager shall provide project oversight and management for all acquisition and procurement activities for the Work Request Application project. The Project Manager shall work with the project team to identify all items to be procured for the successful completion of this project. The Project Management Office (PMO) will review the procurements list prior to submission to the Acquisitions Division and Financing Division. The Acquisitions and Financing divisions shall review the procurement items, determine whether it is advantageous to Group 3 to make or buy the items and Statement of Work (SoW).

### 9.3 Procurement Definition

The following procurement items and/or services have been determined to be essential for the Work Request Application completion and success. The following list of items/services, justification, and timeline are pending PMO review for submission to the Acquisition and Financing divisions:

|  |  |  |
| --- | --- | --- |
| Item/Service | Justification | Needed By |
| Item A; Eclipse | Needed to develop the required Java code | 20211031 |
| Item B; SQL Lite | Needed to host Work Request Database | 20211031 |
| Item C; QT Jambi | Needed to develop dynamic widgets | 20211031 |

In addition to the above list of procurement items, the following individuals are authorized to approve purchases for the Work Request Application project team.

**Name Role**

Dave Leake Project Manager

Ian Oliver Lead Designer

### 9.4 Type of Contract Vehicles to be Used

All items and services to be procured for the Work Request Application, will be solicited under IDIQ and Support Service contracts. The project team will work with the Acquisition and Finance Divisions to define the item types, quantities, services, and required delivery dates. The Acquisition and Finance divisions will then solicit bids from various performers to procure the items within the required time frame and at a reasonable cost under the IDIQ and Support Service vehicles. This contract will be awarded with one base month and 1 option month.

### 9.5 Procurement Risks

While all risks will be managed in accordance with the project’s risk management plan, there are specific risks which pertain specifically to the acquisition process which must be considered:

* Unrealistic schedule and cost expectations for performers
* Manufacturing capacity capabilities of vendors
* Conflicts with current contracts and vendor relationships
* Configuration Management for upgrades and improvements of purchased technology
* Potential delays in shipping and impacts on cost and schedule
* Questionable past performers
* Potential that final product does not meet requirements

### 9.6 Procurement Risk Management

Project procurement efforts involve external organizations and potentially affect current and future business relationships as well as internal supply chain and vendor management operations. Because of the sensitivity of these relationships and operations the project team will include the project sponsor and a designated representative from the Acquisition and Finance divisions in all project meetings and status reviews.

### 9.7 Cost Determination

For the Work Request Application, Group 3 shall issue a Broad Agency Announcement (BAA) to solicit proposals from various vendors and performers, which describe how they will meet the requirements of the Work Request Application and the cost of doing so. All proposals will include vendor/performer support of each of the items listed in the Section 9.3 table.

### 9.8 Standardized Procurement Documentation

In this environment, the goal is to simplify the procurement management by any means necessary to facilitate successful completion of the Work Request Application. To aid in simplifying these tasks, a documentation standard will be instituted for all steps of the acquisition process.

The Project Management Office (PMO) maintains a repository on the Group 3 Google Drive which contains standard project management and procurement documentation that will be used for the Work Request Application project.

### 9.9 Contract Approval Process

The first step in the contract approval process is to determine what items and/or services will require procurement from outside vendors/performers. This will be determined by conducting a cost analysis on products or services which can be provided internally and compared with purchase prices from vendors. Once cost analyses are complete and the list of items and services to be procured externally is finalized, the Finance and Acquisition divisions will send out BAAs. Once solicitations are complete and proposals have been received, the approval process will begin. The Project Manager (Dave Leake), representatives from the Acquisition and Finance divisions, will conduct a review of all vendor proposals to determine which meet the criteria established by the project team and the Acquisition and Finance divisions. Purchases less than $25,000.00 only require the approval of the Project Manager (Dave Leake); whereas purchases greater than $25,000.00 must be approved by the Contract Review Board. For these larger purchases the contract review board will be to determine which contract will be accepted. The Contract Review Board consists of representatives from the project team, Acquisitions and Finance divisions, and the PMO.

### 9.10 Decision Criteria

The criteria for the selection and award of acquisition contracts under the EPS project will be based on the following decision criteria:

* Ability of the vendor to provide all items by the required delivery dates
* Quality
* Cost
* Expected delivery dates
* Comparison of outsourced cost versus in-sourcing
* Past performance

These criteria will be measured by the contracts review board and/or the Project Manager. The ultimate decision will be made based on these criteria as well as available resources.

### 9.11 Vendor/Performer Management

The Project Manager is ultimately responsible for managing vendors/performers. To ensure the timely delivery and high quality of products from the vendors/performers the Project Manager, or their designee will meet weekly with the Acquisition and Finance divisions and each vendor/performer to discuss the progress for each procured item. The meetings can be Zoom or by teleconference means. The purpose of these meetings will be to review all documented specifications for each product as well as to review the quality test findings. This forum will provide an opportunity to review each item’s development, or the service provided to ensure it complies with the requirements established in the project specifications. It also serves as an opportunity to ask questions or modify contracts or requirements ahead of time to prevent delays in delivery and schedule. The Project Manager will be responsible for scheduling this meeting on a weekly basis until all items are delivered and determined to be acceptable.

### 9.12 Performance Metrics for Procurement Activities

While the Acquisition and Finance divisions have their own internal metrics for procurement, the following metrics are established for vendor performance for the EPS project procurement activities. Each metric is rated on a 1-3 scale as indicated below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Vendor | Quality | Delivery | Documentation | Cost | Time | Unit | Efficiency |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1 – Unsatisfactory

2 – Acceptable

3 – Exceptional

In addition to rating each vendor/performer, actual values will be noted to build a past-performance database for selecting vendors for future procurement activities.

## 10. Technical Process Plan

Group 3 will be following the Software Development Life Cycle (SDLC) to help develop and deploy the Reservation/Billing Application. “SDLC is a process that produces software with the highest quality and lowest cost in the shortest time” (2019). To help implement everything the team will be utilizing various tools and techniques

### 10.1 Process Model

Group 3 will use the Agile method, because it allows the team to adjust to the rapid changes and the short project delivery timeline. It’s iterative approach inherently allows to adapt the project for necessary changes.

### 10.2 Technical Requirements

Group 3 will use the following tools and techniques for the development of the Work Request Application:

* Language – Java
* Database – SQL Server / Local Host or Azure (Pending Funds)
* Code repository – GitHub
* Documentation – Microsoft Word
* Communication – Slack/Asana

Coding for the front-end will be in java using the downloaded window add in. Additionally, SQL Server will be used as a database to record necessary information from the application. GitHub will help the team collaborate with the code in a free and easy to use location. All documentation will be completed in Microsoft Word for the sake of compatibility and usability.

# Requirements Specification

## Introduction

## Purpose

The purpose of this Software Requirement Specification (SRS) is to define the requirements of the Work Request Module that will be integrated into the Existing Pulse Application that is used in the US Army Corps of Engineers Europe District. The intended audience is all authorized users of the current Work Request Application in the district.

## Scope

This document is sub section of PULSE\_SRS\_LEVEL\_0. It describes the Work Request Application for the PULSE Application. It will describe the implementation and features of the sub module.

## Definitions, acronyms, abbreviations

DFD – Data Flow Diagram

ERD – Entity Relationship Diagram

PULSE – Name chosen for Access database that was later upsized to an application.

SRS – Software Requirements Specification

Work Request – Information regarding Work Requests to support 3 Business lines of effort: Construction, Engineering &Environmental.

## References

IEEE Recommended Practice for Software Requirements Specifications. (Revision of IEEE Std 830-1993) <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>

O:\@DataTeam\\_Documents\DesktopReference\ PULSE-CodingStandardQuickReference.docx

## Overall Description

### Product Perspective

#### Figure 1 – PULSE DFD Level 0 Context Diagram

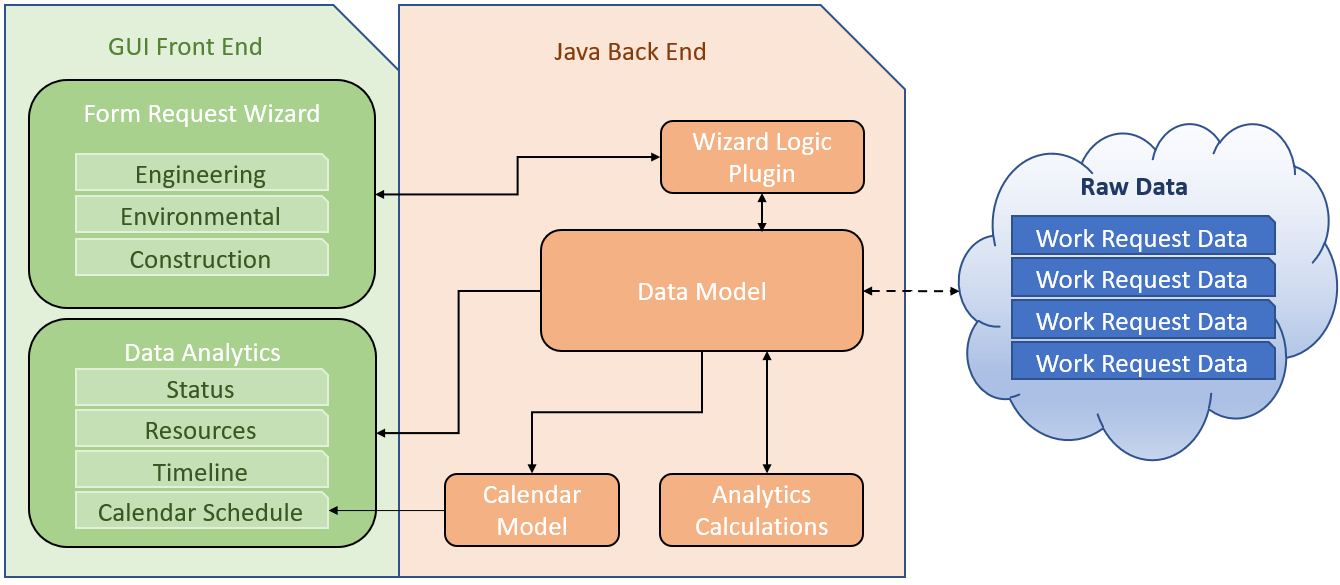


#### Figure 2 – PULSE DFD Level 1



#### Figure 3 – PULSE Work Request State Diagram

Reference from High-Level Design:



Visio Object to be updated as needed:



#### Figure 4 – PULSE Application Auditing State Diagram

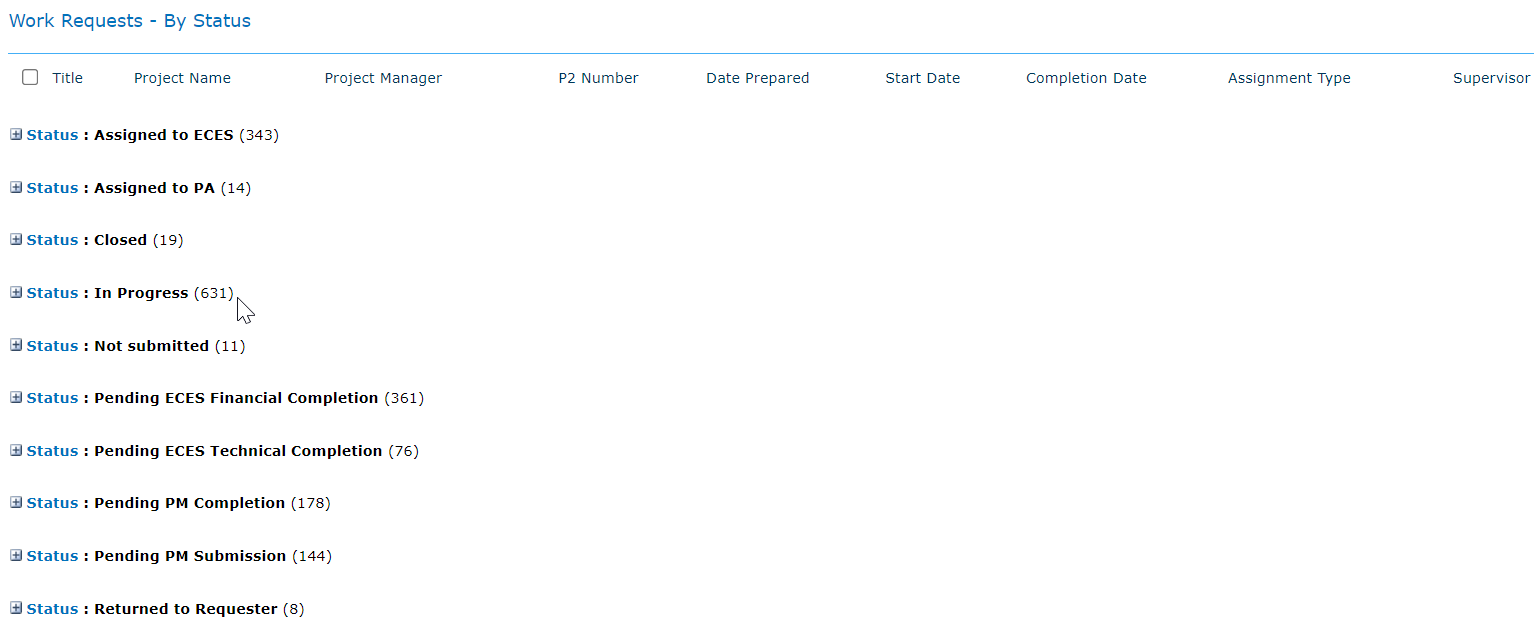


### System interfaces

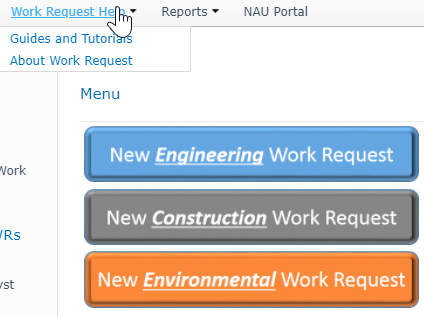
Java Window Applications

### B) User interfaces

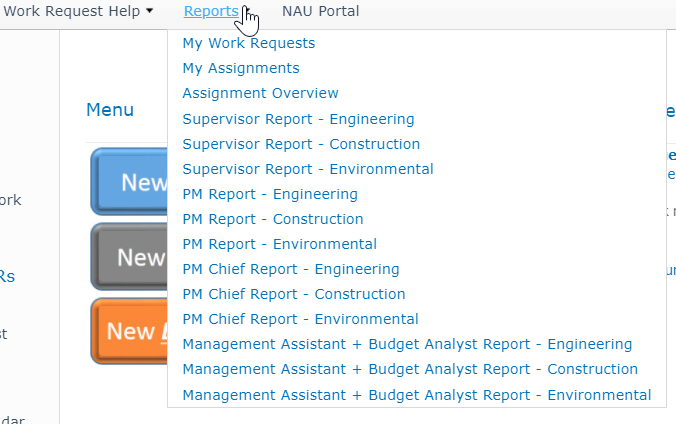
#### Main Page



#### Main Page – Work Request Help Menu

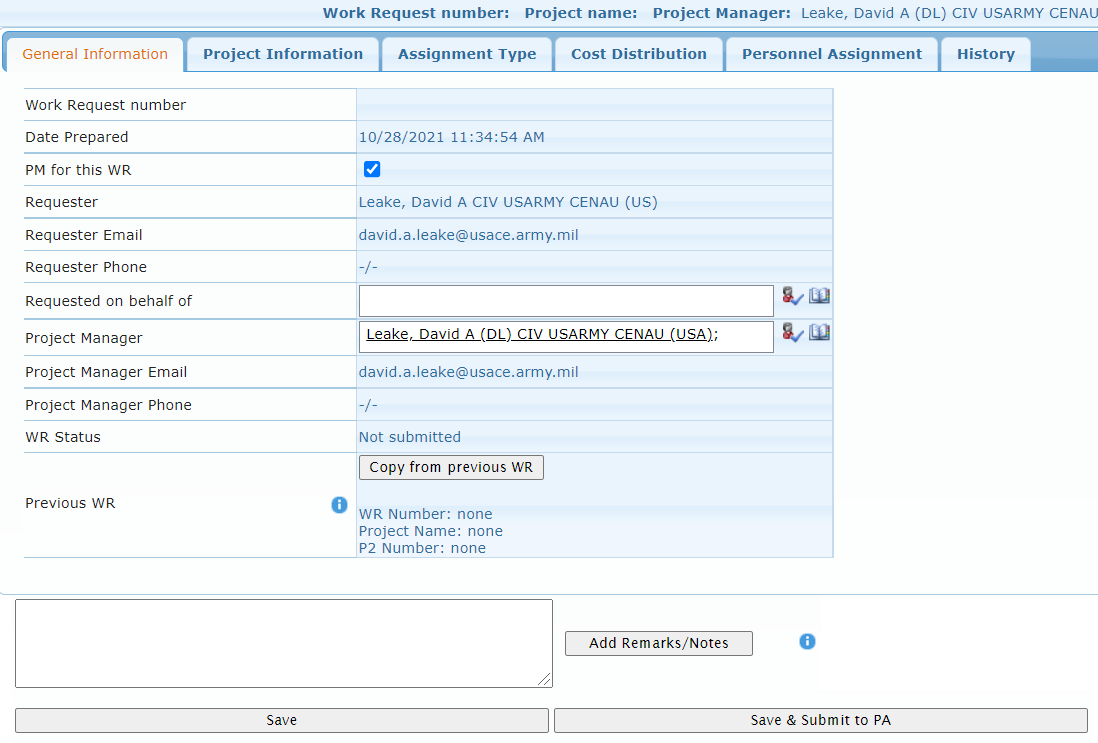


#### Main Page – Work Request Reports Menu

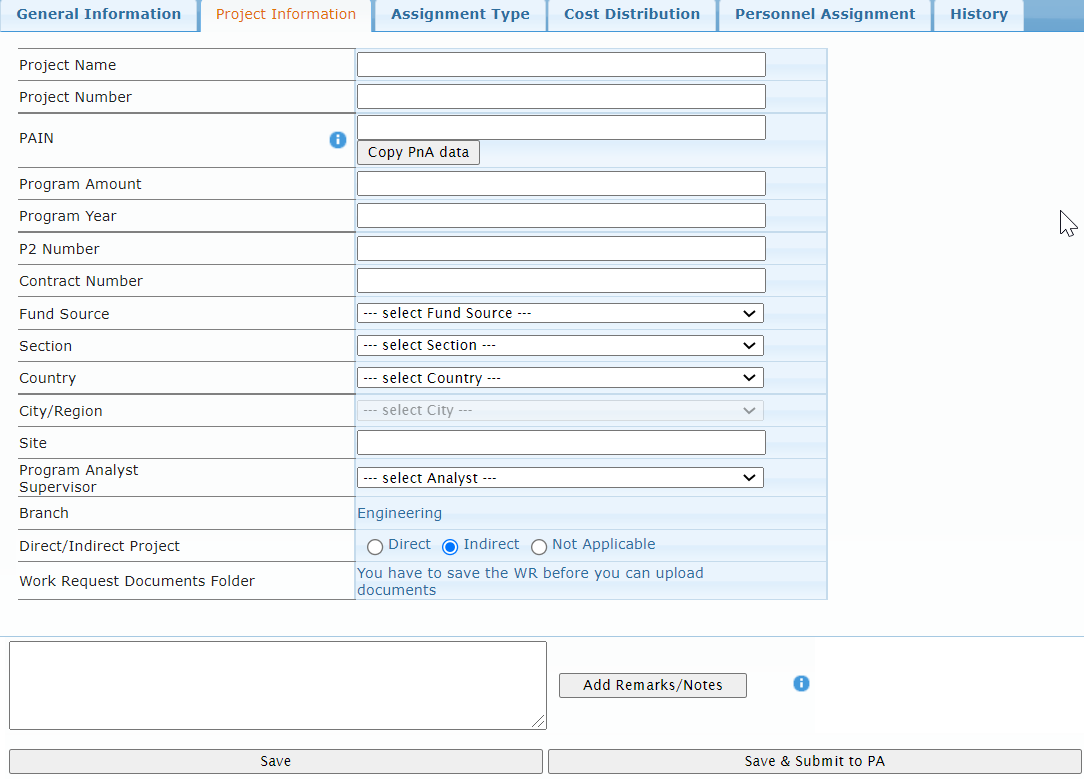


#### New Work Request

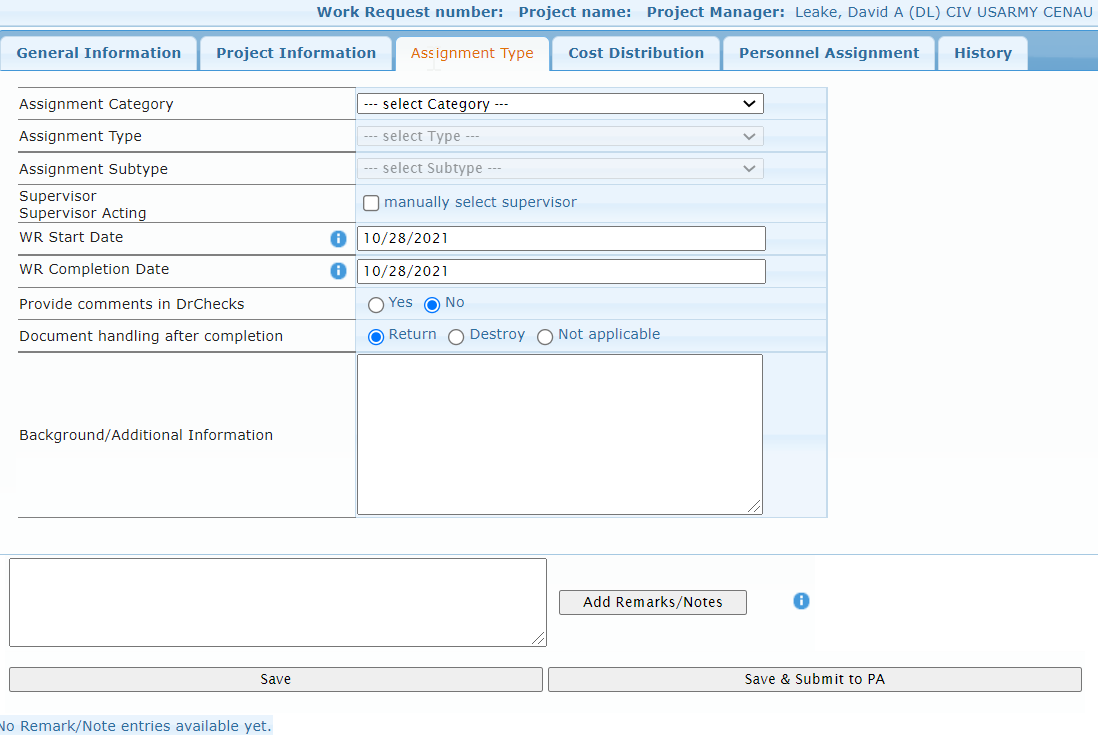
##### General Information



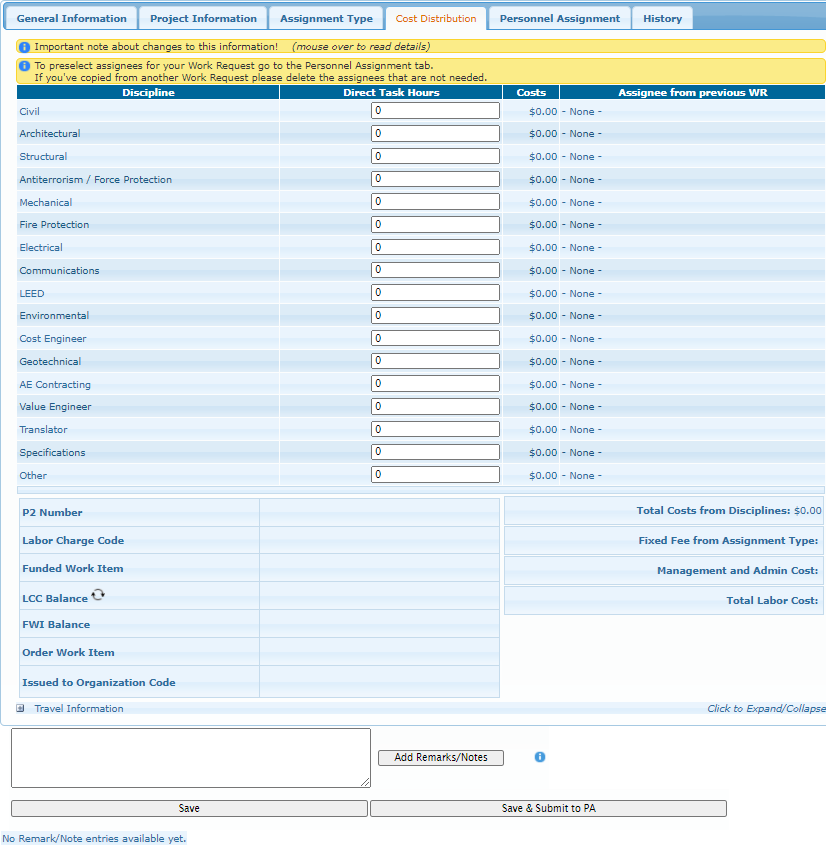
##### Project Information



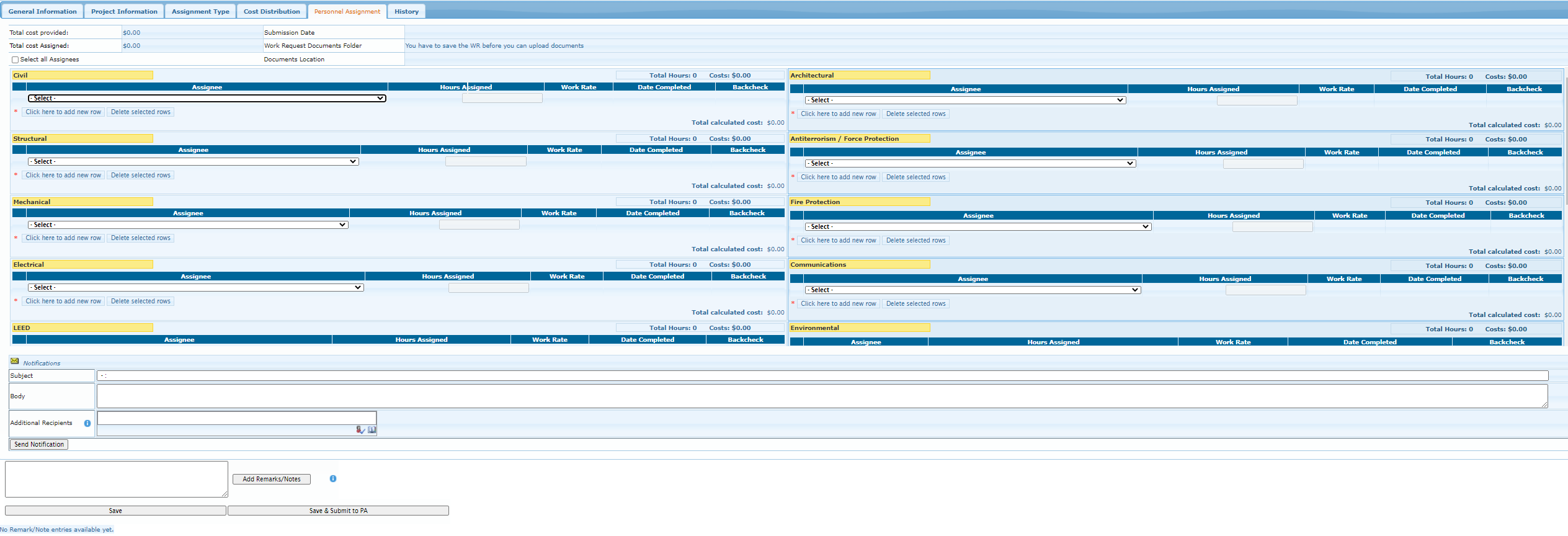
##### Assignment type



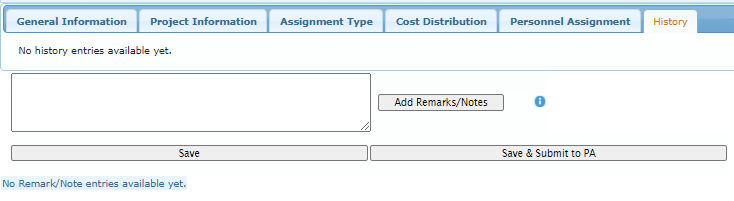
##### Cost Distribution



##### Personnel Assignment



##### History



### Hardware interfaces

Protocols: HTTPS:443, LDAP: 389, RPC: 135, SQL:1433

### Software interfaces

#### Azure connection SQL Server 2019 / Windows Server 2022

* 1. cmsc495team03final.eastus.cloudapp.azure.com
  2. See Test Plan for access

#### SQL Server 2019 (localhost)

Mnemonic: SQL

Specification number: Compatibility Level 130

Version number: 13

Source: Microsoft

#### SQL Server management Studio

Mnemonic: SSMS

Specification number: Compatibility Level 130

Version number: 13

Source: Microsoft

#### 3. Individual Development Environment

Name: Eclipse

### E) Communications interfaces

HTTPS:443

SQL: 1433

### F) Memory

Standard Windows Server and SQL Server requirements.

### G) Operations

This should specify the normal and special operations required by the user such as

a) The various modes of operations in the user organization (e.g., user-initiated operations).

b) Periods of interactive operations and periods of unattended operations: None

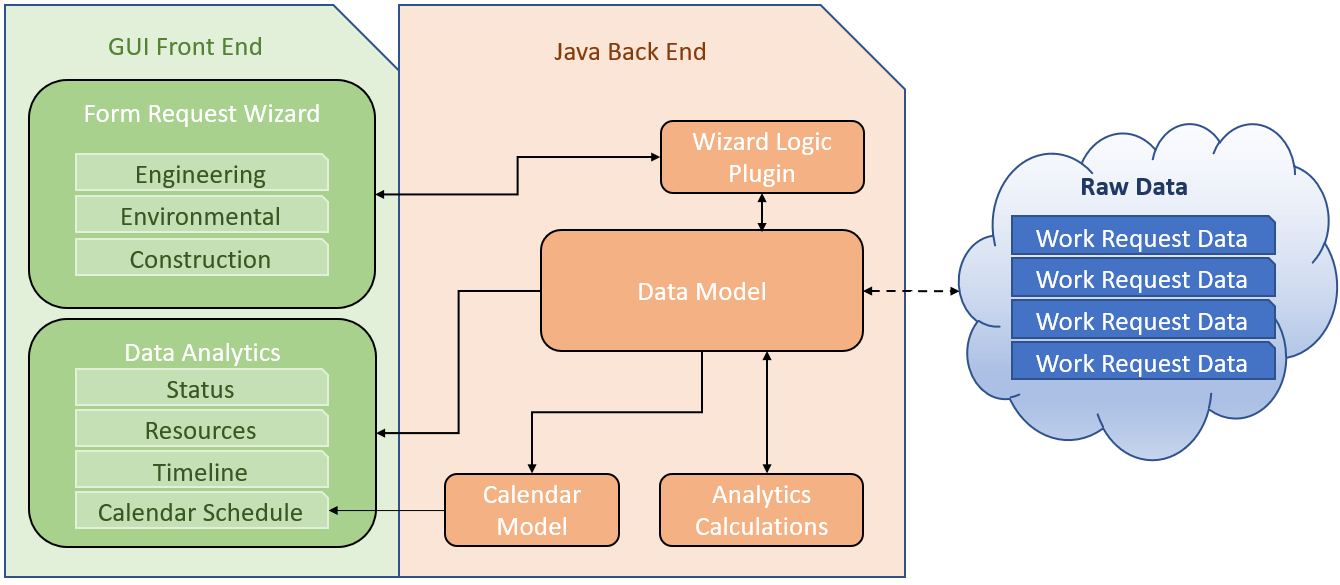
c) Data processing support functions: None

d) Backup and recovery operations are inherited from WARCOM Data Center Policy.

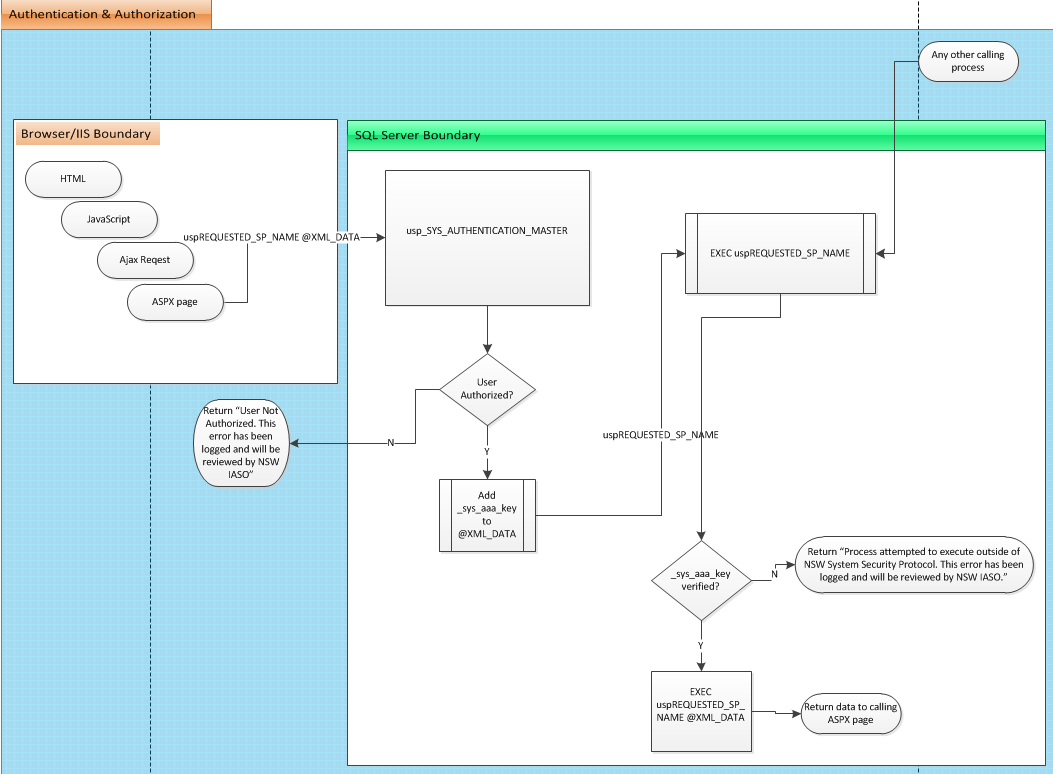
### H) App adaptation requirements

PULSE Application can be adapted to any setting.

## Product Functions



Viso Object to Be Updated:



#### Pulse -Work Request module

1. The Work Request Module will be used to enter work requests for the following USACE Europe District Business Functions:
   1. Construction
   2. Engineering
   3. Environmental

## User Characteristics

The PULSE – Work Request will be used by all authorized end users of the PULSE Application to create and manage Work Requests. The module will be used mainly by Construction, Engineering and Environmental teams, normally with advanced degrees of Technical Knowledge.

### Use Case 1.0**:** End User Submits a work request

An end user requests an account and corresponding rights to needed areas of the PULSE Application

**Precondition:** The PULSE Application needs to be configured properly on the local area network and with access to SQL Server where the database is stored. The system needs to be ready to accept user Work Request submissions.

**Post-condition:** The system provide analytics based on Work Request categories.

**Actor Profile:** End user desires to use one or more features of the PULSE Work Request Application. User is assigned to USACE Europe District and has a bona-fide need for access.

**Sequence of events:**

1. Customer opens the Work Request App.
2. Customer selects the Work Request Category.
3. Customer is provided a form, prepopulated with data domain according to Category.
4. Customer submits the Work Request
5. System can analyze and report on Work Request Categories.

### Use Case 2.0**:** Analysis of work requests

Organizational leadership requests information on Work Requests processed by the PULSE Application.

**Precondition:** PULSE Application was used to create a work request in one of the 3 categories: Construction, Engineering or Environmental.

**Post-condition:** Desired information is provided to leadership in accordance with time requirements specified in organizational policy.

**Actor Profile:** Work Request report requests usually come from leadership due to organizational requirements concerning a Work Request in support of Europe District Projects.

**Sequence of events:**

1. Administrator navigates to https://PULSE.navsoc.socom.mil/app/user\_manager.html
2. Administrator selects user account request.
3. Administrator decides to approve or disapprove request.
4. If approved, administrator configures the account with appropriate access.
5. System notifies End user of all actions taken.

## Constraints

Hardware limitations (e.g., signal timing requirements): None

Interfaces to other applications: PULSE Application

Parallel operation: None

Audit functions: Inherited with Auditing functionality

Control functions: None

Higher-order language requirements: C#, HTML, AJAX, jQuery, SQL, Java

Signal handshake protocols (e.g., XON-XOFF, ACK-NACK): TCP/IP

Reliability requirements: 24x7

Criticality of the application: Critical to daily operations and Work Request actions.

## Assumptions and Dependencies

PULSE Application is developed on Windows Server 2012 Enterprise with Internet Information Services (IIS) 8.5 installed.

PULSE Database is developed on SQL Server 2012 Enterprise edition or SQL Lite or My SQL

## Specific Requirements

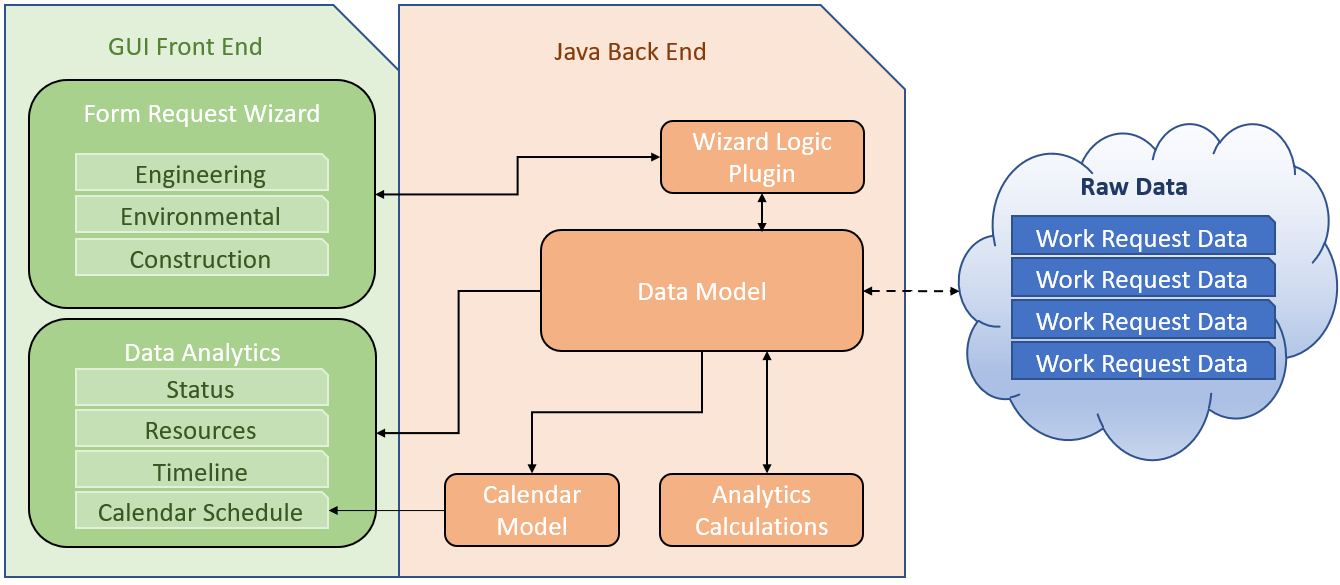
## Performance Requirements

The system shall support a minimum of 1000 simultaneous users to be supported.

They system shall process 95% of all transactions in less than 2 seconds.

## Logical Database Requirements

### Work Request Proposed ERD





## Design Constraints

### Standards Compliance

All Coding will be completed using the PULSE Application Coding Standard Quick Reference Guide, located:

@DataTeam\\_Documents\Desktop\Reference\PULSE-CodingStandardQuickReference.docx

## Software System Attributes

### Reliability

PULSE Application will be available 24 hours a day, 7 days a week.

### Availability

Daily backups will be performed. Disaster Recover from latest backup is acceptable.

### Security

Microsoft SQL Server Transparent Data Encryption will protect data at rest (DAR).

SSL will protect data in transit (DIT).

All Data Manipulation Language will be captured in SYS\_AUDITING\_TABLE\_LOGS

All User activity will be captured in SYS\_ AUDITING\_ACTIVITY\_LOGS

All SQL errors will be captured in SYS\_ AUDITING\_SQL\_ERROR\_LOGS

All Application errors will be capture in SYS\_ AUDITING\_WEB\_ERROR\_LOGS

All outgoing email will be captured in SYS\_ AUDITING\_EMAIL\_LOGS

### Maintainability

The maintainability of the Work Request module will be inherited from the design of the PULSE Application. Due to being written in industry-standard programming languages and a sound database foundation, the entire PULSE Application will be scalable to organizational needs.

### Portability

The PULSE WORK REQUEST is a sub-module of The PULSE Application which should be able to be ported to any other Microsoft network with minimal code changes.

0%of components are host-dependent code

0% of code is host dependent.

Use of a proven portable language: /JAVA

Use of a particular operating system: Microsoft Windows Server 2019/SQL Server 2019 on Azure

# Test Plan

## Introduction

The Project Test Plan (PTP) defines the scope, methods, resources, and schedule of all testing activities to ensure that the Work Request application meets all the project requirements. This plan outlines all the application components and features that need testing, the specific type of testing required, the team members responsible for testing, the testing schedule, and any potential risks involved with executing the test plan. The test team will consist of the participating members of Team 3 Dave Leake, Ian Oliver and, Aurelien (Will)Tchouente-Tsebo.

## Application Components Tested

The Work Request Application consists of three primary components which can be categorized into frontend and backend functional areas. The frontend component will consist of a user interface (UI) written with QT and Java. The backend components consist of Microsoft SQL Server. Each of these components are tested to ensure they function as intended when interacting with one another.

For the UI, all user accessible functions are tested manually. The backend SQL Server is tested via various scripts and reports. This guarantees that the Work Request Application operates as expected and ensures the user experience is consistent. Manual code review also takes places for all components.

## Application Features Tested

* UI
  + Front Page/ Splash Page
    - Login Page
    - Username/Email field
    - Password
  + Work Requests
    - General Information
    - Cost Distribution
    - Analytics
  + New Work Request
    - General Information
* Backend
  + Unit Tests
  + Manually verifying any errors found

## Types of Tests Needed

Multiple forms of testing will be implemented including, manual, unit, and acceptance testing. The Test Director developed unit tests for each function in the SQL backend. Acceptance testing is performed by the entire team to determine that the Work Request application fulfills all the project plan requirements. As outlined above, each feature is tested to confirm expected functionality and error handling.

## Test Documentation

Each test is documented using a Pass/Fail table which includes, the test case, all inputs, expected output, actual output, and the result (Pass/Fail). Screenshots accompany each test case and are included with testing documentation along with the Pass/Fail table. All release versions of the Work Request application have undergone testing. A testing report has been generated to document testing results throughout each phase of development and version release. The test reports include a heading with test outline, a pass/fail table, and a summary of the test results. A sample of the testing table is shown below in Figure 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID # | Input | Expected Output | Actual Output | Pass/Fail |
| 1. Login | Username and password | Main UI window | Main UI window | Pass |
| 2-User can interact with all work request functionalities. | N/A app pulling data form server | UI window | UI Window | Pass |
| 3-major functionalities like searching, filtering, etc. work as expected | Button clicks | Interaction with database | Info from database populate UI | Pass |
| 4-Pulling data from database into UI | N/A populates on login | Work request filled with info from server | Work request filled with info from server | Pass |
| 5-importing data from UI into database | Click submit button | SQL server profiler shows data was imported | SQL server profiler shows data was imported | Pass |
| 6-Analytics | N/A populates on login | Charts | Charts | Pass |
| 7- Wrong password detection | Wrong username and or password | Access denied | Access Denied | Pass |

*Table 1: Pass/Fail table for test documentation.*

Graphical user interface, website

Description automatically generated

*Figure 2: Test case 1 login*

A screenshot of a computer

Description automatically generated

*Figure 3: Test case 2 output from correct login*

Graphical user interface, application

Description automatically generated

*Figure 4: Test case 3 wrong login input detection*

## Testing Environment and Needs

Testing has been performed on the *Windows* environment. The Work Request application has multiple dependencies that must be resolved to successfully launch and test the application. These dependencies have been formally documented and provided to the testers and are included in the testing report.

At a minimum, the testing environment will require an up-to-date version of:

* Java Development Kit (ex. JDK-17.0)
  + <https://www.oracle.com/java/technologies/downloads/#jdk17-windows>
* Eclipse IDE
  + <https://www.eclipse.org/ide/>
  + Help > Install New Software
  + In the "Work with:" section paste in <https://download.eclipse.org/windowbuilder/latest/>
  + Select all the checkboxes that show up and install
* SQL Server 2019
  + Latest Pulse backup (Restored to Localhost)
  + Azure VM Turned on and accessible via the internet
    - Server name: Cmsc495team03.eastus.cloudapp.azure.com
    - SQL Authentication 1 of following authorized UserName & PassWord combos
      * Sa
        + 1234567890
      * Terry
        + 1234567890
      * Ian
        + 1234567890
      * Will
        + 1234567890
* Word processing software (ex. Microsoft Word) for documentation.

Once the application successfully passes all test cases in the development environment, it will transition to a production staging environment and undergoes additional testing before deploying to production. All tests are conducted by following appropriate testing guidelines and a detailed user guide.

## Testing Schedule

Testing will be completed by Monday of each week.

## Testing Breakdown

### Manual Testing

The Work Request UI portion is primarily tested using manual methods as outlined above in section 3. The SQL database is also manually inspected for accurate structure as defined in the Project Design documentation. Manual testing is repeated on all elements that were modified between development phases.

### User Scripts

* Launch Phase4\_v1.4\_defaultValues.jar file
* Enter username and password to login to the Azure Server.
* User can see all tabs (Work requests/ new work request etc.…)
* User can interact with all tabs
* Content of pages is properly aligned, well managed and without spelling mistakes.
* UI should be filled with data pulled from server.
* Go to new user request to fill a new request and push to the server
* Close application then opens again to be able to view most recent pushed request on work requests tab.

### Unit Testing

Each of the functions of the Java backend has been unit tested with the Junit framework, with each test case identified in table 1 below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID #** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Work Requests tab | Login into Azure server | Work requests tab | Work request tab | Pass |
| General information tab | Drop down button on work request status is Clicked | List of statuses from database | List of statuses from database | Pass |
| Cost distribution tab | Cost Distro by Discipline | Data pulled from database | Data pulled from database | Fail (Still Not fully Wired) |
| Work request report | Close app and open again | Report displayed at bottom of work requests tab | Report displayed at bottom of work requests tab | Pass |
| Save Button | Save button on relevant field pressed | All fields on the same tab sends current field data | All fields on the same tab sends current field data | Pass |

*Table 2: Work Request Unit Tests.*

A screenshot of a computer

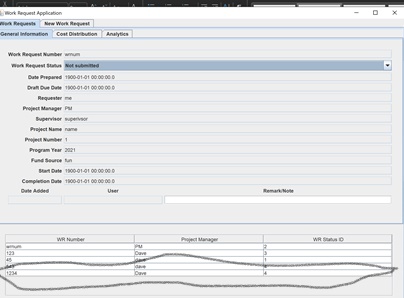
Description automatically generated

*Figure 5: General information tab with data from server*

A screenshot of a computer

Description automatically generated

*Figure 6: New work request tab to input data for new work request.*



*Figure 7: submitted new work request visible after closing and opening the app again.*

A screenshot of a computer

Description automatically generated*Figure 8: Cost distribution tab with data from server*

### Acceptance Testing

Acceptance testing will be performed throughout the week and feedback provided to developers to facilitate agile changes to be made based on team feedback. Final acceptance testing will be performed after the final phase of development to verify the application completely fulfills all requirements set forth by the project plan. Each user-facing function will be tested to ensure correct implementation. The table below breaks down all the user facing functions into individual acceptance tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID #** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Work Requests tab | Login into Azure server | Work requests tab | Work request tab | Pass |
| General information tab | Drop down button on work request status is Clicked | List of statuses from database | List of statuses from database | Pass |
| Cost distribution tab | N/A | Data pulled from database | Data pulled from database | Pass |
| New Work Request  Submit | Submit Button Clicked | Completed Form Displayed; Data exists in DB | Completed Form Displayed; Data exists in DB | Pass |
| Save Button | Save button on relevant field pressed | All fields on the same tab sends current field data |  |  |

*Table 1: Work Request Acceptance Tests.*

### Testing Criteria

The result of a test, either pass or fail, is determined by the test output matching the expected/desired output. For example, if a user requests a password-reset link by entering a valid e-mail address and the user never receives the password reset email, then the test is considered failed. Contrarily, if the password reset email is received and the user can successfully reset his/her password then the test passes. This testing methodology has been applied to each developed test. If a test fails, the development team will be alerted of the defect and a fix will be implemented before the completion of the current development phase.

### Potential Risks

The following Risk Register outlines some potential risks involved with executing this test plan.

|  |  |
| --- | --- |
| **Risk** | **Potential Mitigation** |
| Unable to complete tests in specified timeframe | Test only the components that changed during the last development phase. |
| Tests do not reflect the quality of the application | Adapt testing methods to better suite application requirements. |
| Have to close and open to view work new request |  |
| Date has to entered in the right format else it crashes |  |

### Defects and Risks

The following Risk Register documents the Defects and Risk experienced during the project.

Metrics will be obtained from the source code, from the project management data, or from the execution traces of the source code. From software metrics, we can deduce higher-level metrics, such as the quality of source code or the distribution of defects, but they can be used to build a cost estimation model, apply performance optimization, or improve activities supporting software quality. In our case, the static source code metrics and the metrics obtained from GitHub are taken into consideration. These can be used to characterize the defective code sections on file level or even on source code element level.

Mix-ups can be made despite the help given by the development environments, and blunders might happen because of regular changes with no appropriate specifications, accordingly, tools are needed to help the us with the recognition of mistakes, in this case, defects will be collected and managed via GitHub.

Metrics will be collected from the source code, from the project board information, or from the execution traces of the source code. From programming metrics, we can conclude more significant level measurements, like the nature of source code or the allocation of defects, yet they can be utilized to construct an expense assessment model, apply performance optimization, or further develop exercises supporting programming quality. For our project, the static source code metrics and the measurements acquired from GitHub are used. These can be utilized to describe the deficient code segments on record level or even on source code component level.

|  |  |
| --- | --- |
| **Risk** | **Mitigation** |
| Unable to connect to Pulse Database due to default DB | Create \_AMistakeDoNotDelete, maps all users to this db. |
| Tests do not reflect the quality of the application | Adapt testing methods to better suite application requirements. |
| Unable to connect to Azure database |  |
| Unable to pull or push data from database |  |

# Usability Test Plan

## Document Overview

This document describes a test plan for conducting a usability test during the development of the Work Request Application. The goals of usability testing include establishing a baseline of user performance, establishing, and validating user performance measures, and identifying potential design concerns to be addressed to improve the efficiency, productivity, and end-user satisfaction.

The usability test objectives are:

* To determine design inconsistencies and usability problem areas within the user interface and content areas. Potential sources of error may include:
  + Navigation errors – failure to locate functions, excessive keystrokes to complete a function, failure to follow recommended screen flow.
  + Presentation errors – failure to locate and properly act upon desired information in screens, selection errors due to labeling ambiguities.
  + Control usage problems – improper toolbar or entry field usage.
* Exercise the application or web App under controlled test conditions with representative users. Data will be used to access whether usability goals regarding an effective, efficient, and well-received user interface have been achieved.
* Establish baseline user performance and user-satisfaction levels of the user interface for future usability evaluations.

The user groups that the application will be deployed to will be authorized individuals within an organization that will use the application to access work request for Construction, engineering, and Environmental work groups. Testing will occur remotely, and 3 participants will be doing the testing. The expected date range for testing is to be effectuated every Monday.

## Executive Summary

The goals of this test will be to have the users navigate the application and see if the application’s features (login functions, work requests functions and data analysis functions) are working properly, and hopefully to detect any errors/bug so they can be addressed before deployment.

Upon review of this usability test plan, including the draft task scenarios and usability goals for the Work Request Application, documented acceptance of the plan is expected.

## Methodology

There will be 3 participants doing tests remotely via the use of their laptops or computers. Web browsers will be used to facilitate the participants interaction with the application, and surveys among the participants will be conducted to assess satisfaction and encourage suggestions for improvement.

### Participants

3-4 participants are expected to participate as they make up for the individuals involved in the implementation of the project, thus the eligibility for participation requires group 3 membership.

The participants' responsibilities will be to attempt to complete a set of representative task scenarios presented to them in as efficient and timely a manner as possible, and to provide feedback regarding the usability and acceptability of the user interface. The participants will be directed to provide honest opinions regarding the usability of the application, and to participate in post-session subjective questionnaires and debriefing.

The team will select test participants from within the team as each member has the necessary, skills, background and understanding to perform the tests efficiently.

### Training

The participants will receive and overview of the usability test procedure, equipment, and software

### Procedure

Participants will take part in the usability test for the Work Request Application Test Plan via remote screen-sharing technology. The participant will be seated at their workstation in their work environment. Verbal communication will be supported via telephone.

The facilitator will brief the participant and instruct that he or she is evaluating the Web App/Web application, rather than the facilitator evaluating the participant. Participants will complete a pretest demographic and background information questionnaire. Sessions will begin when all participant questions are answered by the facilitator. The facilitator will inform the participant that time-on-task will be measured and that exploratory behavior outside the task flow should not occur until after task completion.

The facilitator will instruct the participant to read aloud the task description from the printed copy and begin the task. Time-on-task measure will begin. The facilitator will encourage the participants to ‘think aloud’ and that a verbal record will exist of the task-system interaction. The facilitator will observe and enter user behavior and comments, and system interaction in a data logging application.

After each task, the participant will complete the post-task questionnaire and elaborate on the task session. After all tasks have been attempted, the participant will complete a post-test satisfaction questionnaire.

## Roles

The roles involved in a usability test are as follows. An individual may play multiple roles and tests may not require all roles.

### Trainer

* Provide training overview prior to usability testing

### Facilitator

* Provides overview of study to participants
* Defines usability and purpose of usability testing to participants
* Assists in conduct of participant and observer debriefing sessions
* Responds to participant's requests for assistance

### Data Logger

* Records participant’s actions and comments

### Test Observers

* Silent observer
* Assists the data logger in identifying problems, concerns, coding bugs, and procedural errors
* Serve as note takers.

**Test Participants**

* Provides overview of study to participants
* Defines usability and purpose of usability testing to participants
* Assists in conduct of participant and observer debriefing sessions
* Responds to participant's requests for assistance

### Ethics

All persons involved with the usability test are required to adhere to the following ethical guidelines:

* The performance of any test participant must not be individually attributable. Individual participant's name should not be used in reference outside the testing session.
* A description of the participant's performance should not be reported to his or her manager.

## Usability Tasks

Testing will be performed on Windows environment. The backend SQL Server will be tested via various scripts and reports. This guarantees that the Work Request Application operates as expected and ensures the user experience is consistent.

The task descriptions below are required to be reviewed by the application owner, business-process owner, development owner, and/or deployment manager to ensure that the content, format, and presentation are representative of real use and substantially evaluate the total application. Their **acceptance is to be documented** prior to usability test.

## Usability Metrics

Usability metrics refers to user performance measured against specific performance goals necessary to satisfy usability requirements. Scenario completion success rates, adherence to dialog scripts, error rates, and subjective evaluations will be used.

### Scenario Completion

Each scenario will require, or request, that the participant obtains or inputs specific data that would be used in course of a typical task. The scenario is completed when the participant indicates the scenario's goal has been obtained (whether successfully or unsuccessfully) or the participant requests and receives sufficient guidance as to warrant scoring the scenario as a critical error.

### Critical Errors

Critical errors are deviations at completion from the targets of the scenario. Obtaining or otherwise reporting of the wrong data value due to participant workflow is a critical error. Participants may or may not be aware that the task goal is incorrect or incomplete.

Independent completion of the scenario is a universal goal; help obtained from the other usability test roles is caused to score the scenario a critical error. Critical errors can also be assigned when the participant initiates (or attempts to initiate) and action that will result in the goal state becoming unobtainable. In general, critical errors are unresolved errors during the process of completing the task or errors that produce an incorrect outcome.

### Non-critical Errors

Non-critical errors are errors that are recovered from by the participant or, if not detected, do not result in processing problems or unexpected results. Although non-critical errors can be undetected by the participant, when they are detected, they are generally frustrating to the participant.

These errors may be procedural, in which the participant does not complete a scenario in the most optimal means (e.g., excessive steps and keystrokes). These errors may also be errors of confusion (ex., initially selecting the wrong function, using a user-interface control incorrectly such as attempting to edit an un-editable field).

Noncritical errors can always be recovered from during the process of completing the scenario. Exploratory behavior, such as opening the wrong menu while searching for a function, will be coded as a non-critical error.

Testing would suspend when Critical defect is encountered, or when System Crash. And testing will stop when 100% Requirements coverage is achieved, meaning, defined / Desired Defect count is reached, all High Priority defects are identified and fixed, all critical Test cases are passed, test Coverage achieves 95%, and Complete Functional Coverage is achieved.

### Subjective Evaluations

Subjective evaluations regarding ease of use and satisfaction will be collected via questionnaires, and during debriefing at the conclusion of the session. The questionnaires will utilize free-form responses and rating scales.

### Scenario Completion Time (time on task)

The time to complete each scenario, not including subjective evaluation durations, will be recorded.

## Usability Goals

The next section describes the usability goals for Work Request Application.

### Completion Rate

Completion rate is the percentage of test participants who successfully complete the task without critical errors. A critical error is defined as an error that results in an incorrect or incomplete outcome. In other words, the completion rate represents the percentage of participants who, when they are finished with the specified task, have an "output" that is correct. Note: If a participant requires assistance to achieve a correct output, then the task will be scored as a critical error and the overall completion rate for the task will be affected.

**A completion rate of 99% is the goal for each task in this usability test.**

### Error-free rate

Error-free rate is the percentage of test participants who complete the task without any errors (critical **or** non-critical errors). A non-critical error is an error that would not have an impact on the final output of the task but would result in the task being completed less efficiently.

**An error-free rate of 90% is the goal for each task in this usability test.**

### Time on Task (TOT)

The time to complete a scenario is referred to as "time on task". It is measured from the time the person begins the scenario to the time he/she signals completion.

### Subjective Measures

Subjective opinions about specific tasks, time to perform each task, features, and functionality will be surveyed. At the end of the test, participants will rate their satisfaction with the overall system. Combined with the interview/debriefing session, these data are used to assess attitudes of the participants.

## Problem Severity

To prioritize recommendations, a method of problem severity classification will be used in the analysis of the data collected during evaluation activities. The approach treats problem severity as a combination of two factors - the impact of the problem and the frequency of users experiencing the problem during the evaluation.



### Impact

Impact is the ranking of the consequences of the problem by defining the level of impact that the problem has on successful task completion. There are three levels of impact:

* High - prevents the user from completing the task (critical error)
* Moderate - causes user difficulty but the task can be completed (non-critical error)
* Low - minor problems that do not significantly affect the task completion (non-critical error)

### Frequency

Frequency is the percentage of participants who experience the problem when working on a task.

* High: 30% or more of the participants experience the problem
* Moderate: 11% - 29% of participants experience the problem
* Low: 10% or fewer of the participants experience the problem

### Problem Severity Classification

The identified severity for each problem implies a general reward for resolving it, and a general risk for not addressing it, in the current release.

**Severity 1** - High impact problems that often prevent a user from correctly completing a task. They occur in varying frequency and are characteristic of calls to the Help Desk. Reward for resolution is typically exhibited in fewer Help Desk calls and reduced redevelopment costs.

**Severity 2** - Moderate to high frequency problems with moderate to low impact are typical of erroneous actions that the participant recognizes needs to be undone. Reward for resolution is typically exhibited in reduced time on task and decreased training costs.

**Severity 3** - Either moderate problems with low frequency or low problems with moderate frequency; these are minor annoyance problems faced by several participants. Reward for resolution is typically exhibited in reduced time on task and increased data integrity.

**Severity 4** - Low impact problems faced by few participants; there is low risk to not resolving these problems. Reward for resolution is typically exhibited in increased user satisfaction.

## Reporting Results

The Usability Test Report will be provided at the conclusion of the usability test. It will consist of a report and/or a presentation of the results; evaluate the usability metrics against the pre-approved goals, subjective evaluations, and specific usability problems and recommendations for resolution. The recommendations will be categorically sized by development to aid in implementation strategy.

# Product Design

## Introduction

## Purpose of the Product Design Specification Document

The Product Design Specification Document documents and tracks the necessary information required to effectively define architecture and system design to give the development team guidance on architecture of the system to be developed. The Product Design Specification Document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and development team. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

## General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

## Assumptions

There is a general assumption that this product is a proof of concept and not a polish/published work. The team working on this product are preforming an exploratory design.

There is a general understanding of a major time constraint and not all the of the desired features will be finished to the polished level that the team may want. As this product is proof of concept this should not prove to be an issue.

## Constraints

The product is written and maintained in the Java programming language and is only intended to run in a Java Runtime Environment (JRE). The development team designed, implemented, and tested only in the Windows Operating System (OS).

The SQL Database in use is based on a government owned system. Much of which is not accessible from outside a secured network. Limiting the usability of much of the database. Two of the tables have been scrubbed and cleansed of protected/sensitive information and are the focus of this proof of concept. As such much of the full capabilities of the project are not available during the proof of concept, such a user lookup, external emailing’s services, or a full catalogue of previous requests.

## Architecture Design

This section outlines the system and hardware architecture design of the system that is being built. For the proof of concept to be considered successful the Work Request Application is required to be able to:

* Add a new record
* View all retrievable records from database
* Access localhost **and** ability to access internet hosted server
* Have some visual breakdowns of the information present in the record

The Work Request Application is designed with a Model-View architecture or Model – View – Controller (MVC) as its more commonly known. Allowing for intermittent connectivity to the database and eliminating the need to continuous connection to function correctly. For this use case the Graphical User Interface will function as both the controller and view for internal model. See figure 1. found below, for visual reference.

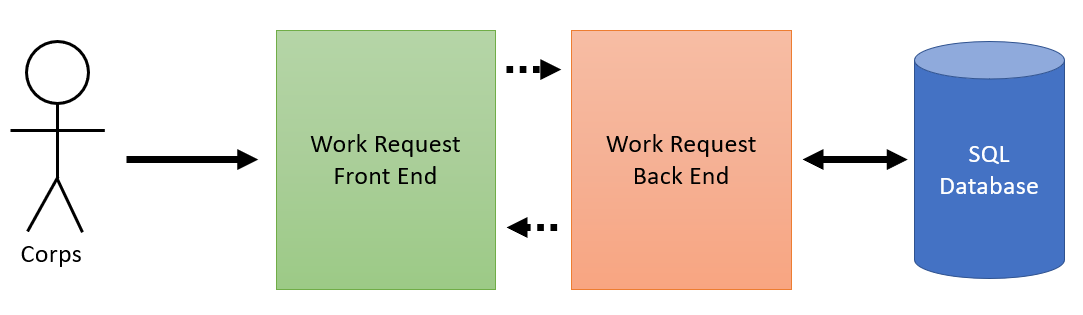


Figure 1. Work Request Application High Level Concept

### Logical View

The Work Request Application will be able to navigate between multiple tabs. One section to display the records stored in the SQL Database, using an internal model as intermediator. And to send requests to SQL Database to store a new record in specified tables.

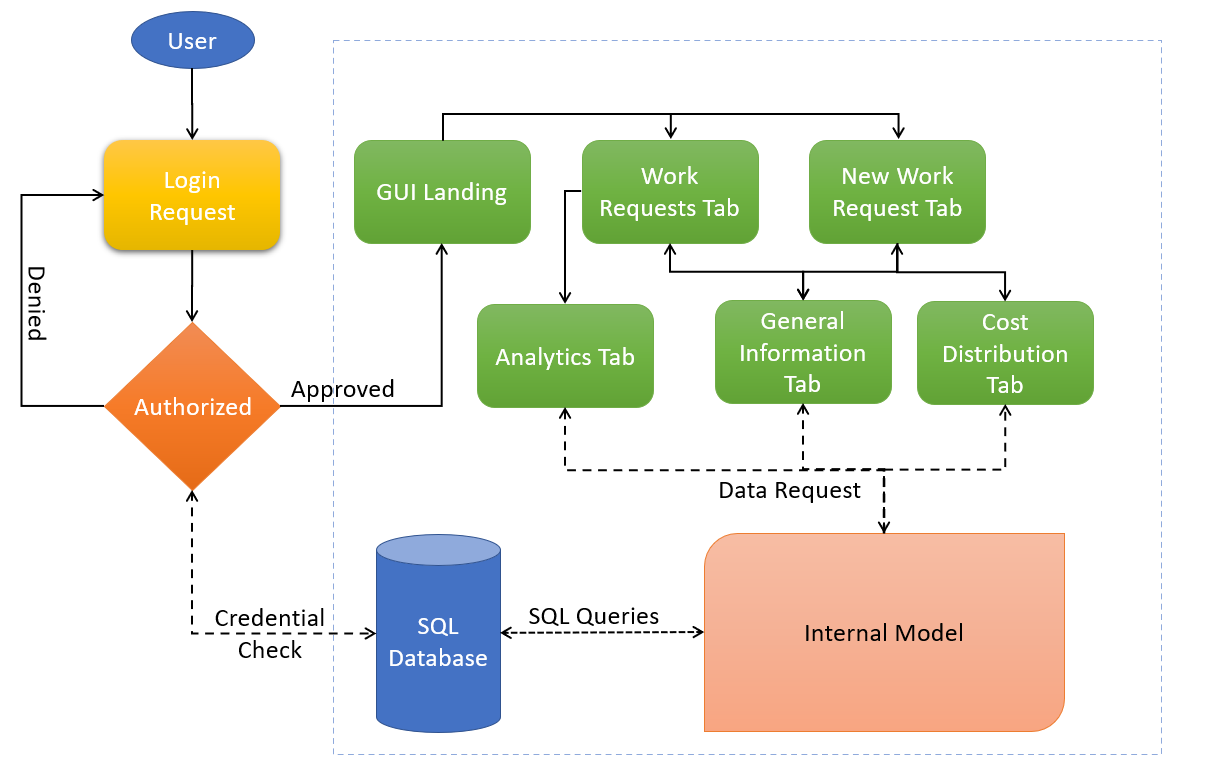


Figure 2. Work Request Application High Level Logic Diagram

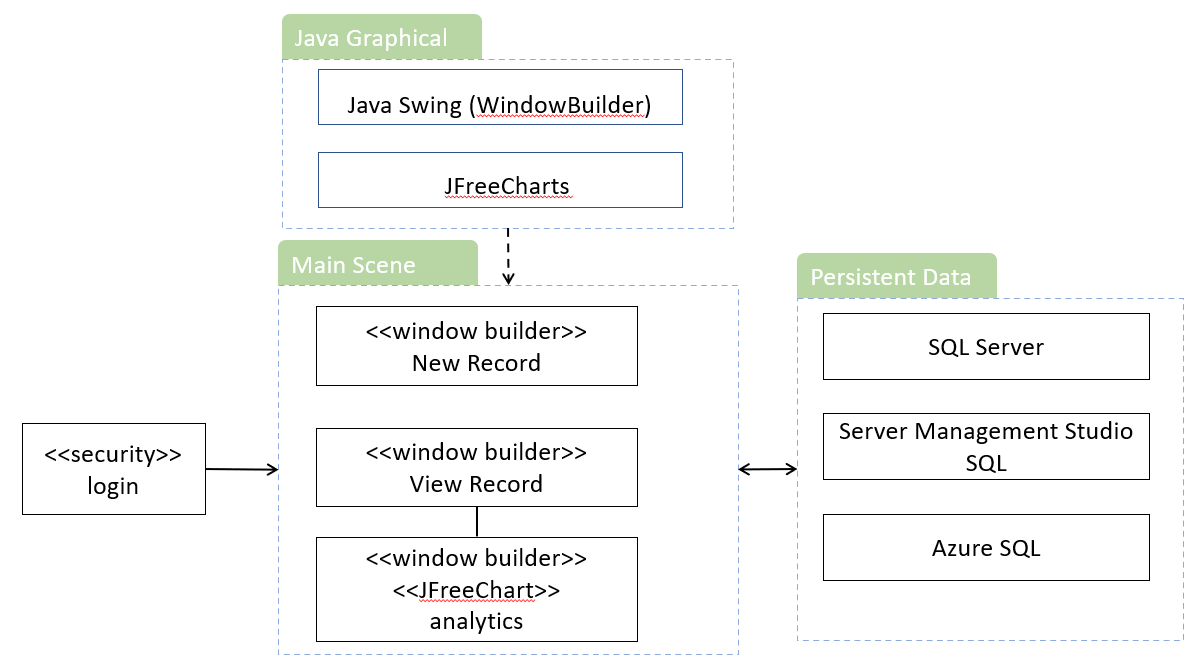


Figure 3. Work Request Application Component Diagram

### Software Architecture

The Work Request Applications main function is for proof of concept for portability of existing software and database to a Java/SQL system. With no official funding for this project all software requirements are free and/or open-source frameworks. Including Java Development Kit 17, Eclipse Community Edition 4.21.0, JFreeChart 1.5.3, MSSQL JDBC 9.4.0, WindowBuilder, Microsoft SQL Server Management Studio v18.0, and Github.

* Language: Java
* Development Environment: Eclipse IDE Version 4.21.0
* Development GUI: WindowBuilder Eclipse Plugin
* Development GUI: JFreeChart 1.5.3
* Database: SQL Server
* Operating System: Windows with JRE 16+
* Source Control: Github

## Performance

There are no performance requirements for the Work Request Application as it is mainly a proof-of-concept project. Functionally each screen, data request, and analytics request would take less than a 3 second lag time. But no hard requirements have been set for performance for this project.

## Alternate architecture (Depricated)

The previous design of this project included functionality that is either unobtainable or adds no benefit to the project as an exploratory design. As such it has been deprecated and replaced with a new design that meets all expected requirements.

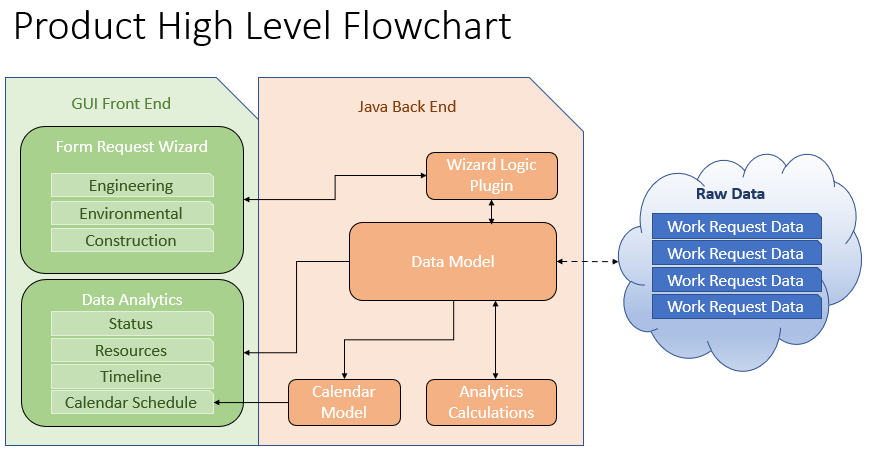


Figure 4. Deprecated High Level Flowchart and Design

## System Design

### Use-Cases

Use cases are described in the Team3WorkRequestHighLevelRequirements.docx.

### Database Design

The Database design and decencies can be found in the files USACE\_ECWR\_Dependencies.pdf and USACE\_ECWR\_DependencyDiagram.pdf

### User Interface Design

The user interface design is focused on two components. A record viewer and a record insertion. The record viewer also has a visual analytics component. For the purposes of this exploratory project, the analytics tab has been limited to a few fields and only two charts. These can be expanded later if the finding is positive.

### WORK REQUEST TAB

The work request tab uses an internal model of ArrayList<Map<String, Object>> to store each data point in the records from the SQL Database. This allows for the connection to be made and verified once per update and only for short burst communications. Allowing for more limited or unstable internet connection to be usable for the application.

The viewing tab for work requests only needs a connection to the Database to request information. This allows for a shorter connection time than inserting a new request into the database.

There is also an analytics tab that take the stored information in the internal model and creates two different visual charts. A pie chart of cost distribution breakdown and a Gantt chart for overall workflow progress. The estimation dates are all that is required for this exploratory phase.

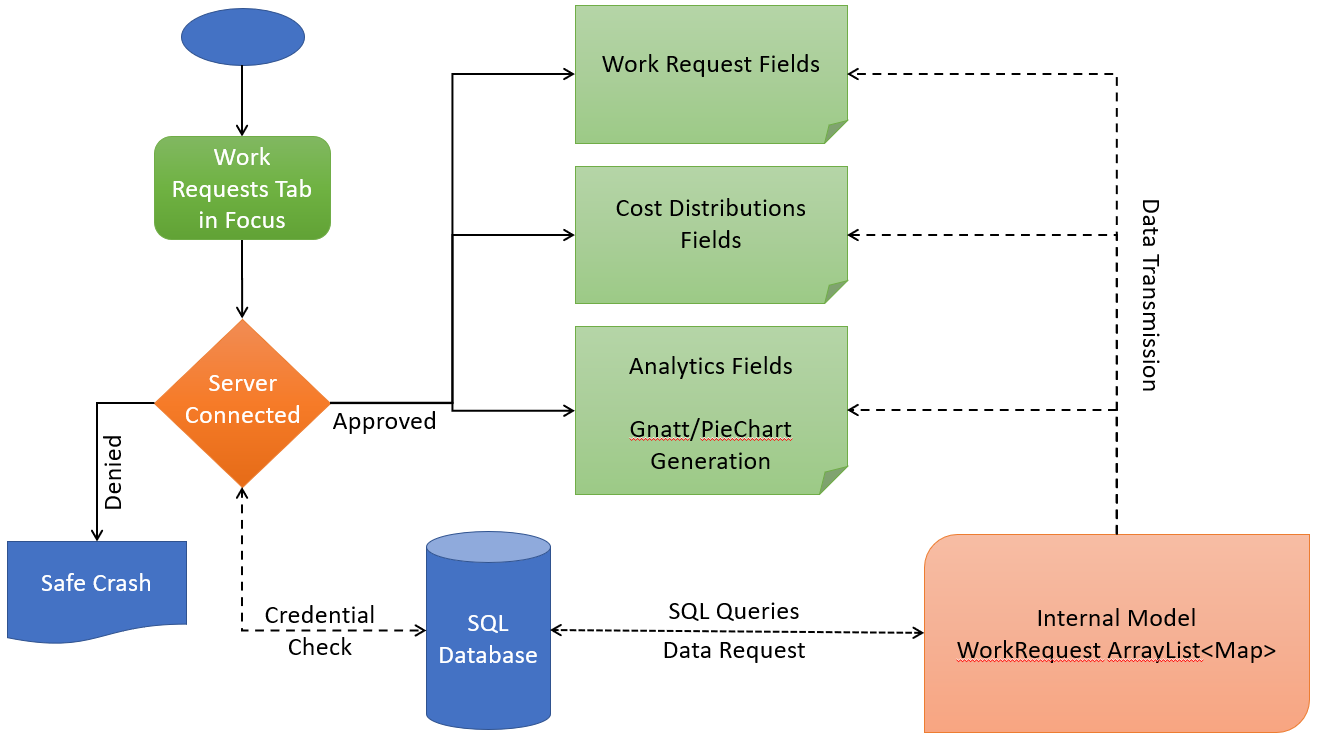


Figure 5. View Work Requests Tab Diagram

### NEW WORK REQUEST TAB

The New Work Request tab is for inserting a new record into the database. Utilizing a sendmap internal model to store the data from SQL server, an insertion request is made from this data to the database. This requires a longer connection on average.

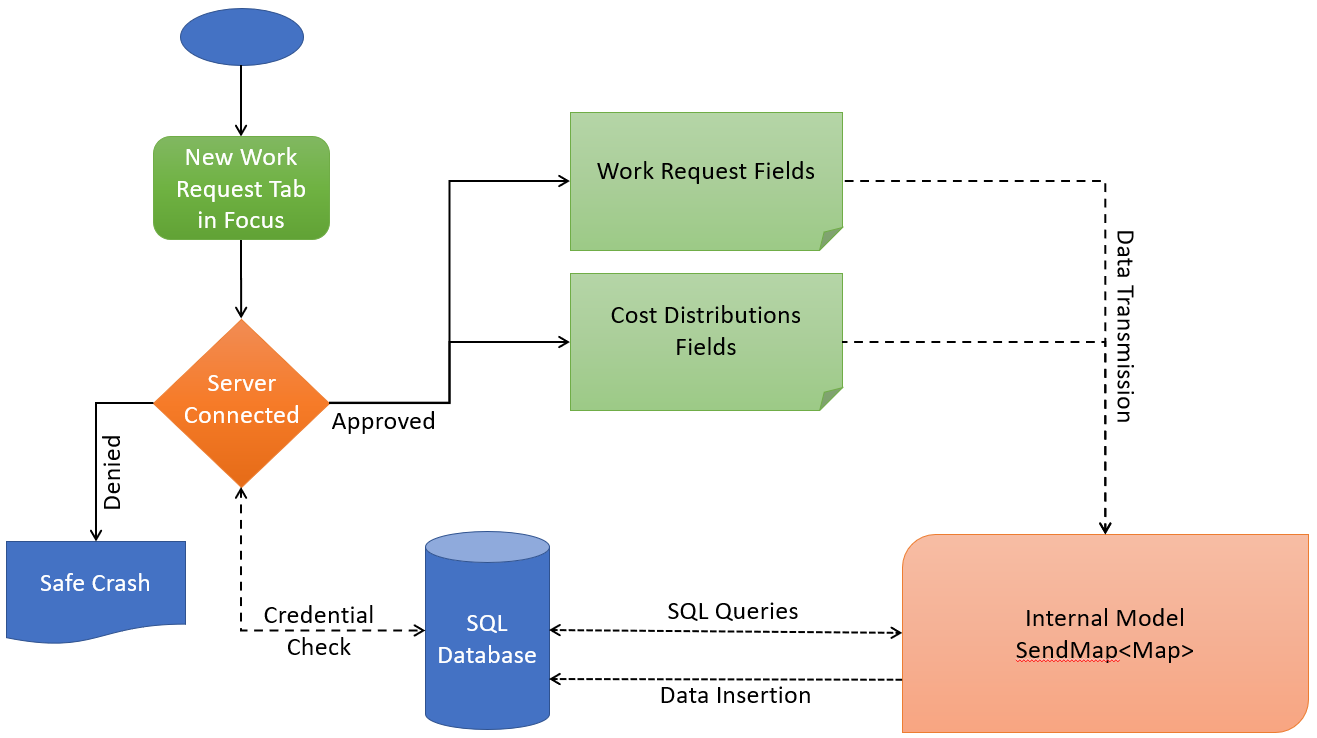


Figure 6. Insert New Work Request Tab Diagram

## Product Design Specification Approval

The undersigned acknowledge they have reviewed the *Work Request Application* **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | Ian Oliver |  |  |
| Title: | Software Designer |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | Dave Leake |  |  |
| Title: | Project Manager |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | Will Tchouente |  |  |
| Title: | Requirements Manager |  |  |
| Role: |  |  |  |

# Development History

Group 3 uses semantic versioning to identify version changes. Semantic versioning uses a three-part version number, in which the first number represents any major changes, the second number represents any minor changes, and the third number represents any patches.

# Appendix A – Project Charter

**Project Charter**

|  |
| --- |
| **Project Name:** Work Request Application |
| **Date: 20211028** |
| **Project Manager: Dave Leake** |
| **Project Sponsor: Terry Mentzos** |
| **Requested Completion Date: 20211214** |
| **Project Justification:** This technological solution will provide a hosted Graphical User Interface (GUI) that will allow USACE employee to access it from any desktop computer. Once logged into, this application will allow the end user to submit work requests in one 1 three main categories: Construction, Engineering or Environmental. For tracking purposes, the software will provide analytics. |

|  |
| --- |
| **Project Overview** |
| 1. Project Plan 2. Test Plan 3. Project Design 4. Phase 1 Source: classes 5. Phase 2 Source: methods 6. Phase 3 Source: user experience and modules 7. Final deliverable |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approvals** | | | |
| **Title** | **Name** | **Signature** | **Date** |
| Project Sponsor | Terry Mentzos | (Email confirmation is acceptable in place of signature.) | 20211025 |
| Project Manager | Dave Leake | David A. Leake | 20211025 |

# Appendix B – Project Team

**Project Team**

|  |
| --- |
| **Project Name:** USACE Work Request Application |
| **Date: 20211028** |
| **Project Manager: Dave Leake** |

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Contact** |
| Project Manager (PM) | Dave Leake | [David.leake@gmail.com](mailto:David.leake@gmail.com) |
| Requirements Manager/Technical Writer (RM/TE) | Will Aurelien | [willtchouente@gmail.com](mailto:willtchouente@gmail.com) |
| Software Designer (SD) | Ian Oliver | [ioliver.work@gmail.com](mailto:ioliver.work@gmail.com) |
| Test Director | Will Aurelien | [willtchouente@gmail.com](mailto:willtchouente@gmail.com) |
| User Experience/Training Manager (UX) | Will Aurelien | [willtchouente@gmail.com](mailto:willtchouente@gmail.com) |

# Appendix C – Statement of Work

**Statement of Work**

**Scope**

As agreed, upon with the client, the contractor will provide the client with a single solution to manage the Work Request categories of the Europe District. The application is titled “Work Request Application.” This solution will come in the form of a standalone java app or Application, which the client will access using a jar file or URL, username, and password. The Work Request Application will allow the user to create Work Requests for a given business function and provide analytics on current work requests.

**Location of Work**

All work will be completed virtually, as the solution is software based and does not require in-person meetings.

**Timeline**

The timeline of work is between the dates of 20211028 and 20211214 with the final product to be delivered on 20211214. The schedule of work is included in the delivered Project Plan, Appendix D.

**Acceptance Criteria**

The acceptance criteria as agreed upon with the client is as follows:

1. Given that a customer submits a Work Request, the application user will properly capture all information provided.
2. Given that a customer requests a report on the status of work request, the system will provide the information.

**Software and Hardware Requirements**

For this application, the client will require the following to run the program:

* A personal computer (Windows or Macintosh)
* A web browser
* Internet connection
* A printer

The information described above are the requirements agreed upon between the contractor and client. The contractor is required to fulfill all mandatory requirements by the scheduled finish date, and the client is required to review and accept the completed application based on its ability to successfully perform the mandated requirements.

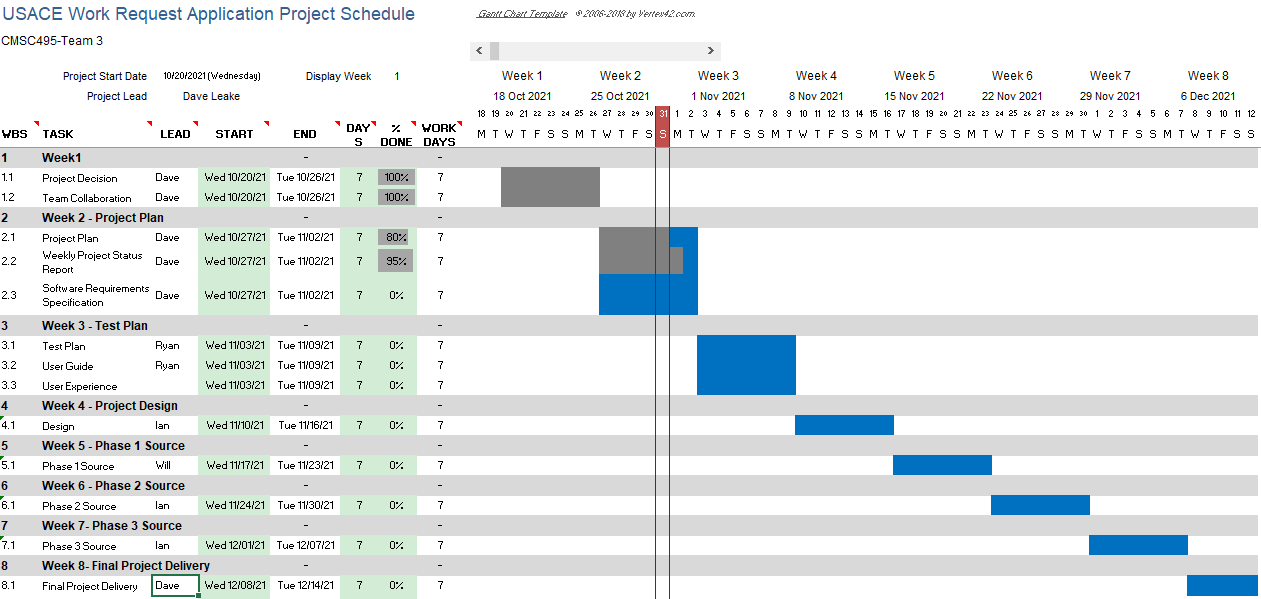
# Appendix D - Schedule

**Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Week | Dates | Lead | Topic | Description | Due Date | Assignments Due |
| 1 | Week 1 | Everyone | Form Teams | * Get to know team members * Pick a topic | - | - |
| 2 | Week 2 | * Dave   + Project Manager * Will * Requirements Manager/ Technical Writer | Project Plan | * Outline Milestones * Delegate responsibility * Describe project’s purpose * Identify the system specifications | 11/02 | Project Plan |
| 3 | Week 3 | * Ryan   + Test Director | Users Guide and Test Plan | * Create test plan with a clear users’ guide * If appropriate create test data files | 11/9 | Test Plan  Peer Review 1 |
| 4 | Week 4 | * Ian   + Software Designer * Will   + User Experience / Training Manager | Design | * Design a user interface and related structures * Finalize test case | 11/16 | Project Design |
| 5 | Week 5 | Everyone | Phase 1 Source | * Software Development | 11/23 | Phase 1 Source  Peer Review 2 |
| 6 | Week 6 | Everyone | Phase 2 Source | * Software Development | 11/30 | Phase 2 Source |
| 7 | Week 7 | Everyone | Phase 3 Source | * Software Development | 12/07 | Phase 3 Source |
| 8 | Week 8 | Everyone | Final Report | * Compile all Topics into a single document | 12/14 | Final  Peer Review 3 |

# Appendix E – Gantt Chart Example

**USACE Work Request Work Request Application Gantt Chart**



# Appendix F- Scope Baseline

**Scope Baseline**

|  |  |  |  |
| --- | --- | --- | --- |
| Scope Baseline: Access | | | |
| System Requirements Checklist | Item | Status | Notes |
| Can the user select login/register by entering the required information? | 1a |  |  |
| Does the application disallow access when incorrect login is entered? | 1b |  |  |
| Can the user view the work request page upon correct login entry? | 1c |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Scope Baseline: Functionality | | | |
| System Requirements Checklist | Item | Status | Notes |
| Does the application generate a report given the right inputs? | 2a |  |  |
| Does the application buttons and tools work as intended? | 2b |  |  |
| Does the application timeout after a certain amount of inactivity? | 2c |  |  |
| Can the user interact with all work requests? | 2d |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Scope Baseline: Management Portal | | | |
| System Requirements Checklist | Item | Status | Notes |
| Can the user generate a work request report? | 3a |  |  |
| Can the user search for any work request current, past, in progress, complete, and cancelled? | 3b |  |  |
| Can the user view desired work request? | 3c |  |  |
| Can the user modify work requests? | 3d |  |  |
| Can the user add or delete work requests? | 3e |  |  |
|  |  |  |  |
|  |  |  |  |

# References

What is Sdlc? (2019, May 06). Retrieved March 26, 2020, from <https://stackify.com/what-is-sdlc/>