### Assignment 3 - Project Estimates, Resource Plan & Schedule

SE 638: Software Project Management

Group #3: Shaima Albugami, Steve Greulich, Michelle Ibarra, Yiyun Zhang

November 1, 2020

Contents

[Assignment 3 - Project Estimates, Resource Plan & Schedule 1](#_Toc55118426)

[Work Summary 3](#_Toc55118427)

[1. Revised low level WBS 4](#_Toc55118428)

[3. Effort Estimate and Resource Plan 23](#_Toc55118429)

[4. Project Schedule 24](#_Toc55118430)

[5. Report Schedule 24](#_Toc55118431)

[References 27](#_Toc55118432)

### Work Summary

Each team member researched each question individually before meeting up to share their findings. After debating and gathering insight for each question, the parts were divided equally between team members. The team met twice to assign sections and perform a group review.

Shaima Albugami

* Section 2 – Activity scenarios
  + Phase 1
* Section 3
  + Phase 1 bullet 1 & 2
* Section 4
  + Phase 1
* Section 5
  + Summary for phase 1

Michelle Ibarra

* Section 1 – Revised Low Level WBS
  + Phase 1
  + Phase 2
* Section 2 – Activity scenarios
  + Pre-Development
* Section 3
  + Section Summary
* Section 5
  + Summary of the Critical Activity Scenario (Pre-Development - DevSecOps pipeline)

Steven Greulich

* Section 1 – Revised Low Level WBS
  + Pre-Development
  + Phase 3
* Section 3
  + Phase 2
  + Resource Plan
* Section 4
  + Phase 2
  + Section Summary
* Section 5

Yiyun Zhang

* Section 3
  + Phase 3
  + Resource Plan
* Section 4
  + Phase 3
* Section 5
  + Summary of the Hiring process
  + Summary of the Schedule risks

### 1. Revised low level WBS

**Note:** For the purposes of this schedule, we are assuming that the Philadelphia Medical Group hired a consulting company to come in during the pre-development phase to perform an initial evaluation of the overall project. At the end of the pre-development phase, they will let the Philadelphia team know of the estimated project schedule and cost to implement. In start the Phase 1 activities, the Philadelphia Group will need to sign off on the Funding Release.

**Pre-Development**

* Project Kickoff
* High – Level Business Requirements
  + Hold requirements elicitation sessions
  + Document High level business requirements
  + Review / approve high-level business requirements
* Project Schedule
  + Meet with appropriate teams to get preliminary estimates
  + Documentation project timeline (including dependencies)
  + Review project schedule
* Project resource requests
  + Put in requests for resources
  + Get resources assigned
* Funding Release
  + Creation of Funding Release
  + Review of Funding Release
  + Approval of Funding Release
* Infrastructure Setup
  + Database Software
    - Request quotation from vendors
    - Purchase licenses
    - License installers / keys delivered
  + Performance Testing
    - Identification of performance testing tool
    - Request quotation from vendors
    - Purchase licenses
    - License installers / keys delivered
  + Servers
    - Request quotation from vendors for hardware
    - Purchase servers
    - Servers allocated
    - Access granted to users
  + Code Repository Tool
    - Identification of code repository tool
    - Request quotation from vendors
    - Purchase licenses
    - License installers / keys delivered
    - Access granted to users
    - Software Installed / verified on developers laptops
  + Security Scanning Tool
    - Identification of scanning tool
    - Request quotation from vendors
    - Purchase licenses
    - License installers / keys delivered
    - Access created to users
    - Software Installed / verified on necessary machines
  + Continuous integration/continuous deployment tool
    - Identification of tool
    - Request quotation from vendors
    - Purchase licenses
    - License installers / keys delivered
    - Software installed / verified

**Phase 1**

* **Develop Appointment module**
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* **Develop employee records module**
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* **Develop Patient records module**
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* **Develop Inventory Management Module**
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* Vulnerability Assessment
  + Execution of vulnerability scan
  + Creation of reports
  + Vulnerability remediation by development team
  + Re-scan for vulnerabilities
  + Final report created
* Performance Testing
  + Creation of Performance Test Cases
  + Execution of test cases
* Regression Testing
  + Creation of regression suite of test cases
  + Execution of regression suite
* User Acceptance Testing
  + Creation of User Acceptance Testing
  + Execution of User Acceptance testing
  + Client sign off for testing
* System Training
  + - Develop training materials and resources
    - Staff training
    - Staff workshops
* Deployment
  + Creation of rollout plan
  + Review of roll out plan
  + Dry-run of roll out plan
  + Execution of rollout

**Phase 2**

* **Develop Patient Facing Record History**
  + DBMS:
    - Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* **Develop Patient Facing Appointment Module**
  + DBMS:
    - Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
* **Develop Patient Self Service Check-in**
  + DBMS:
    - Install & configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* Vulnerability Assessment
  + Execution of vulnerability scan
  + Creation of reports
  + Vulnerability remediation by development team
  + Re-scan for vulnerabilities
  + Final report created
* Performance Testing
  + Creation of Performance Test Cases
  + Execution of test cases
* Regression Testing
  + Creation of regression suite of test cases
  + Execution of regression suite
* User Acceptance Testing
  + Creation of User Acceptance Testing
  + Execution of User Acceptance testing
  + Client sign off for testing
* System Training
  + - Develop training materials and resources
    - Staff training
    - Staff workshops
* Deployment
  + Creation of rollout plan
  + Review of roll out plan
  + Dry-run of roll out plan
  + Execution of rollout

**Phase 3**

* Develop report generation module
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* Develop external inventory management module
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
    - System Integration Testing
      * Creation of System Integration Test Cases
      * Execution of System Integration Test Cases
* Develop payment and accounting module
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* Develop outgoing module
  + DBMS:
    - Install & Configure DBMS Engine & SQL Interface
    - Test SQL interface software
  + REST API Service Server:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + Client App:
    - Requirements
      * Requirements Elicitation
      * Documentation of requirements
      * Review requirements
    - User Interface design
      * Initial Wireframe design
      * Review of wireframe
      * Create design based off feedback
      * Review design
    - Software Design
      * Create design document
      * Review of design
    - Software Development
      * Development of software
      * Code Review
    - Unit Testing
      * Creation of test cases
      * Execution of test cases
  + System Integration Testing
    - Creation of System Integration Test Cases
    - Execution of System Integration Test Cases
* Vulnerability Assessment
  + Execution of vulnerability scan
  + Creation of reports
  + Vulnerability remediation by development team
  + Re-scan for vulnerabilities
  + Final report created
* Performance Testing
  + Creation of Performance Test Cases
  + Execution of test cases
* Regression Testing
  + Creation of regression suite of test cases
  + Execution of regression suite
* User Acceptance Testing
  + Creation of User Acceptance Testing
  + Execution of User Acceptance testing
  + Client sign off for testing
* System Training
  + - Develop training materials and resources
    - Staff training
    - Staff workshops
* Deployment
  + Creation of rollout plan
  + Review of roll out plan
  + Dry-run of roll out plan
  + Execution of rollout
* Warranty / Maintenance Period
  + Pay vendor invoices
  + Close out any open purchase orders
  + Conduct lessons learned
  + Creation of support documentation
  + Knowledge transfer of solution to support personnel

Assumptions:

* During executions of testing, defect fixing and re-testing will occur

**Rationale**

When developing the low-level Work Breakdown Structure (WBS), our team considered what activities would be necessary to develop each component. The factors driving the approach for developing each component include definition of the component’s purpose and user; software testing, system training, and release management. For instance, there are common activities among all of the components, such as requirements gathering or software design. While the activities are the same, the component’s purpose and user that will distinguish how the component is development. For instance, the Appointment Module for Doctors, Nurses and Office Admins would function and look differently than from the Patient Facing Appointment Module. This is to say, just because the Appointment Module and Patient Facing Appointment Module appear similar, their user type drives the needs and differences.

Moreover, other component activities include events which all components must address. For instance, before software is actively developed, the Business Analyst, Developer, and Test need to work together to define the functionality being developed as well as the test that will be used to validate the functionality. Similarly, all software must be documented and tested. This documentation activity benefits the Client and Operations and Maintenance (O&M) team by establishing how Development team implemented the setup, build, test, and deployment of the functional software.

Aside from the delivery of functional software, the system training will be required for the Philadelphia Medical Group staff. The Doctors, Nurses and Office Admins are the primary users interfacing with inputting Patient information, setting up appointments, ordering supplies, etc. The successful adoption of the system largely depends on this group of users. It is essential to provide them training and workshops to support the learning and coaching of the new system. These trainings would be organized by user (Doctor, Nurses and Office Admins) to walk through primary business processes and how the user would interface with the system to manage that business process.

Lastly, there will be defined release management approach. This encompasses how system will be supported through operations and maintenance. This may include online resources that provides Frequently Asked Questions, Tutorials, or Help Desk ticket support. Also, it is imperative that the System Owner signs off on the system to document the acceptance and approval to deploy the software system for production use. After the deploy, it is helpful for the Client to participate in a satisfaction survey to analyze how the system is meeting their needs of their business. This helps course correct the team to address and remedy issues for the next phase of design and development. Lastly, our team would perform a lesson learned exercise to reflect on what went well, what didn’t go as planned and what should be changed in order to improve team performance moving forward.

2. Activity scenario analysis for Pre-Development activities and for Phase 1

1. **Pre-Development Activity Scenario**

**a. Descriptive Walkthrough to Complete the activity**

The following is a description of establishing the DevSecOps pipeline done during the Pre-Development phase. Gartner defines DevSecOps as, the integration of security into emerging agile IT and DevOps development as seamlessly and as transparently as possible. Ideally, this is done without reducing the agility or speed of developers or requiring them to leave their development toolchain environment.” (Gartner, 2020) The tools analysis establishes what tools will be utilized for the development, build, and delivery of software. There will be code repository to manage the version control of the code among the development team. Further, our team has required the following tests will be done to ensure and increase confidence of the delivery of functional code: unit testing, integration testing, regression testing, security scan test, and performance testing. Various tools will be required to support the software from the moment it is committed to the moment it is deployed.

b. Human (skillsets) other resources required

This section identifies the human and resources required for supporting the DevSecOps pipeline activity.

1. Human Resources
   1. Software engineer (front-end and back-end)
      1. Develop, test, debug, and deliver functional software.
      2. Developing standard procedures for developing and deployment software.
      3. Expertise in JavaScript, Node.js, Java, C++, Go, Python, etc.
      4. Expertise in unit, integration, regression, and performance testing.
      5. Problem solver and solution-oriented leader.
   2. Architect
      1. Design future state system architecture based on business and client needs.
      2. Provide technical system design recommendations.
      3. Evaluate third-party tools to support the development of software.
      4. Problem solver and solution-oriented leader.
   3. Infrastructure Engineer
      1. Design and define automated processes for supporting commit through deployment of software.
      2. Implement standardized automated deployment processes (DevSecOps pipeline).
      3. Experience in automation tools such as GitHub Actions, Jenkins, TravisCi, Ansible, Bamboo, etc.
   4. Automated Tester
      1. Experience using automation tools such as GitHub Actions, Jenkins, TravisCi, Ansible, Bamboo, etc.
      2. Design and define automation test conditions, scripts, and data.
      3. Define and update test cases for software peer reviews.
   5. Other Resources
      1. Git will be the version control tool for managing commits to source code.
      2. GitHub will be used as the development platform to support the development team’s commits to the system source code base to the deployment.
         1. Source Code Management (version control)
         2. Actions (automate deployment)
         3. CodeQL (security scanning)

**c**. **A brief description of how the activity out comes will be evaluated, with evaluation criteria:**

This activity will be evaluated based off the software being able to be pass the various tests incorporated in the DevSecOps pipeline and deploy into its hosting environment.

* **Automation:** This activity outcome will be evaluated from when the Developer to check-in code into a central code repository to when the code is deployed for hosting. Once code is checked-in, there must be a kickoff of automated tests for testing the functionality of the code.
* **Unit Testing:** The unit tests validates the software (i.e. unit) has met the defined task in the user story. If the does not software pass the unit test, then the developer must investigate and remediate the issues.
* **Integration Testing:** After there is a group of units that have passed their unit tests, the software can then undergo integration testing. Integration testing orchestrates the testing of a group of units as a broader component or system.
* **Regression Testing:** Subsequently, our team will also perform regression testing on the software. The benefit to regression testing is to observe and validate whether the new software has modified, disrupted, or damaged existing functional software.
* **Security Scanning:** There will be security scanning to follow to identify vulnerabilities in the software. Also, performance testing applies a particular load to the software to evaluate the speed, how fast, and stability of the software before its “breaks” i.e. the software is unable to function.
* **Deployment:** Once the software has been tested without error, the software is ready to be deployed into the defined hosting environment.

**d. Potential Risks**

The following are the potential risks identified with establishing this DevSecOps pipeline to support the build, test, and deployment of software is critical to the delivery of functional software to the Client.

* **Inconsistent testing:** Not having a DevSecOps pipeline could increase risk to the project by not having a common method for mitigating risks upfront, early and often.
* **Infrequent remediation:** This standardized DevSecOps pipeline will help Developers identify problems with the software, identify problems when software is integrating into a component and/or the larger system, and provides common method for performing security scan.
* **Lack of confidence in releases:** Moreover, without a DevSecOps pipeline, there would be less confidence throughout the development cycle on how well the singular software functionality is performing and how well the software is functioning together.

**Phase 1**

**a. Descriptive Walkthrough to Complete the activity**

The following is description of the activities required for the components that are going to be developed in Phase 1 which are appointment module, employee records module, patient records module, and inventory management module. The activities include the development of DBMS, API, clinic app, unit testing, and system integration testing. A detailed descriptive walkthrough to complete the development of each module:

* **Develop DBMS:** 
  + Install & Configure DBMS Engine & SQL Interface
  + Test SQL interface software
* **REST API Service Server:** 
  + **Requirements** 
    - Requirements Elicitation
    - Documentation of requirements
    - Review requirements
  + **Software Design** 
    - Create design document
    - Review of design
  + **Software Development** 
    - Development of software
    - Code Review
  + **Unit Testing** 
    - Creation of test cases
    - Execution of test cases
* **Client App:** 
  + **Requirements** 
    - Requirements Elicitation
    - Documentation of requirements
    - Review requirements
  + **User Interface design** 
    - Initial Wireframe design
    - Review of wireframe
    - Create design based off feedback
    - Review design
  + **Software Design** 
    - Create design document
    - Review of design
  + **Software Development** 
    - Development of software
    - Code Review
  + **Unit Testing**
* Creation of test cases
* Execution of test cases
* **System Integration Testing**
* Creation of System Integration Test Cases
* Execution of System Integration Test Cases

**b. A list of human and other resources required, with mention of their skillsets**

* **Human Resource:**
  + Project Manager - The project manager will be responsible for adhering that the project is within time, budget and scope. If there are issues that are raised that cannot be resolved, it is the duty of the project manager to escalate as necessary to get a timely resolution.
  + Business Analyst - The Business Analyst will be responsible for eliciting the business requirements from the clients at the Philadelphia Medical Group. The business analyst will also be the liaison between the development team and the Philadelphia Medical group to ensure all requirements are clearly understood.
  + Infrastructure Engineer – The Server Engineer will be responsible for setting up of the necessary infrastructure in place (I.e. Microsoft Windows Server or Linux setup)
  + Database Administrator - The database administrator will be performing the initial database installation and configuration.
  + Developer (Front End) - The front-end developer will be mainly responsible for building the User interfaces of the application. In addition, they will be developing any front-end logic such as calling the appropriate APIs to either send or receive data from the backend system. This resource will also be creating and executing unit test cases as it pertains to the front end.
  + Developer (Back-end) - The backend developer will be responsible for developing the APIs, and any server-side applications that are necessary to be used. This resource will also be creating and executing unit test cases as it pertains to the back end.
  + Quality Assurance Resource - The Quality Assurance resource will be creating and executing System Integration Testing. In addition, they will be ensuring execution of the various other testing phases (I.e. User Acceptance Testing,
  + Performance Tester – The Performance Tester will be creating and executing performance test cases.
  + Security Vulnerability Assessor – The vulnerability assessor will be brought on to look for any known vulnerabilities within the application. After remediations have been performed by the development team, they will be performing another round of testing.
  + Philadelphia Medical Staff (Nurse) - One of the nurses employed by the Philadelphia medical group. This resource will be responsible for providing requirements as it pertains to the nurses roles and responsibilities. In addition, this resource will be performing User acceptance testing for the Nurse related functionality.
  + Philadelphia Medical Staff (Doctor) - One of the Doctors employed by the Philadelphia medical group. This resource will be responsible for providing requirements as it pertains to the Doctors roles and responsibilities. In addition, this resource will be performing User acceptance testing for the Nurse related functionality.
  + Philadelphia Medical Staff (Admin) - One of the Admins employed by the Philadelphia medical group. This resource will be responsible for providing requirements as it pertains to the Admins roles and responsibilities. In addition, this resource will be performing User acceptance testing for the Nurse related functionality.
* **Infrastructure:**
  + Database Management Systems
  + API’S

**c**. **A brief description of how the activity out comes will be evaluated, with evaluation criteria:**

Along with the various testing that will be executed to ensure the modules are performing as expected, there are certain evaluation criteria that will be followed to evaluate the activities outcome:

* **Usability and user interaction**: is it easy to learn all the functionalities.
* **Technical Aspects of the component:** the component is reliable, secure, confidential and bug free
* **Information:** The information is clear, accurate, and beneficial to the users

**d. Potential Risks**

The following are the potential risks identified with the deliverable, the client business context, or specific activities.

* **Planning related risks:** delayed deliverables
* **Requirement related risks:** not meeting the stakeholder’s requirements
* **Security related risks:** who is authorized to do what and malicious activities

**Section Summary**

This section described the activities that need to be done in the pre-development and phase 1. For phase 1, we discussed these activities for the modules that will be developed in that phase. These activities go from the development of the DBMS to Server development to client app development along with the user interface and the various testing that need to happen to make sure the system works as expected.

We also listed the human resource required to perform these activities, such as project managers, business analysts, infrastructure engineers consisting of network engineers and server engineers, etc. We also mentioned the resources that we need for the system infrastructure. We identified the evaluation criteria that we will use to evaluate the outcome of these activities, such as the usability and user interaction and the technical aspects of the components criteria. Finally, we listed the potential risks that might hinder the activities outcome's achievements, such as planning related risks and security-related risks.

### 3. Effort Estimate and Resource Plan

3.1 Effort Estimate

Please see document titled **“SE638-Grp3-A3-Phase Estimation.xlsx”** for effort estimate. The method used to estimate the effort is included in the referenced document.

3.2 Resource Plan

Please see document titled **“SE638-Grp3-A3-Human Resources.xlsx”** for the human resource estimate. The hourly rate for each human resources is included in the reference document.

**Section Summary**

This section included estimates and a resource plan for the entire project: Pre-Development, Phase 1, Phase 2, and Phase 3. The activities in this section align with the Low-Level Work Breakdown Structure identified in Section 1. Our Phase Estimation reflects the low-level activities broken down by task. Each task is reflects the likelihood of the task being implemented or facing risk of being implemented. Further, our Human Resource Plan reflects the Labor Category and the number of daysexpended on the project by month per year. Per our estimates, 12 different labor categories: Project Manager, Business Analyst, Infrastructure Engineer, Database Administrator, Developer (Front End), Developer (Back-end), Performance Tester, Quality Assurance Resource, Security Vulnerability Assessor, Philadelphia Medical Staff, Automation Tester, and Technical Architect. The average salary rates for each of the roles were taken from Glassdoor. To account for overhead (i.e. benefits, vacation, etc.), a 30% overhead was added to the overall days charged to the project. Moreover, the labor categories that will expend the greatest number of days supporting the implementation of the Philadelphia Medical Group system are the two Developers (Front-End and Back-End), Technical Architect, Business Analyst and Quality Assurance Resource. With respect to our experience in estimating efforts and resources, this identified labor categories supporting the project adequately align provide a reasonable estimate such that a completed Philadelphia Medical Group system will be delivered by June 2023.

### 4. Project Schedule

Please see the file titled “SE638-Grp3-A3.proj” for the Microsoft Project schedule.

Section Summary

Given a project kick off date of January 4, 2020, the overall project timeline will span roughly 42 months. Each phase will be run synchronously to each other, the completion of one phase will start the next phase. Some optimization techniques were taken into consideration to run tasks in parallel to minimize the timeline. For instance, the database, back-end, and front-end development all run in parallel with each other. Once all three sub areas are complete, then system integration testing can occur. In addition, tasks such as User Interface Designs can occur while the front-end developers are working out some technical designs of the software. However, front-end development cannot occur until the User Interface designs have been completed. Another effort for parallelization of tasks, is during user training. While user training is occurring, the development team can get to work putting together plans on implementing the systems into production. Tasks such as Performance Testing, Vulnerability Scanning, and Regression testing however, will be performed sequentially to not alter with the other during each stream's execution (I.e., performance fixes that might negatively impact vulnerability testing).

### 5. Report Schedule

Having a work breakdown structure helped us determine the activities needed to complete each module. The critical activities go from the development of the DBMS to Server development to client app development along with the user interface and the various testing that need to happen to make sure the system works as expected. After we determined these activities, we discussed the skill set we need to execute these activities. We listed the human resources required and described the role of each one of them in the project. Considering each task will be performed by one person, we estimated the duration of that tasks to be done in no more than 10 days. We broke down our estimation into optimistic, most likely and pessimistic. We also evaluated each task in terms of its risks. The high-risk task took the pessimistic estimation, medium risk took most likely, and low risk took low estimation. After analyzing the tasks and the duration, we used Microsoft Project to produce our project schedule. We took into consideration the slack time that we might need for the tasks that have high risks

One of the critical activities identified for this project was to plan and invest in a DevSecOps pipeline. With a DevSecOps pipeline, the Development team for Philadelphia Medical Group will be able to more frequently test software and remediate bugs. Without an established DevSecOps pipeline, Developers would not have a common method for testing software. Further, there would not be a consistent method for integrating software or establishing a cadence to performing security scans. Establishing the DevSecOps pipeline will give the Development team and Clients increased confidence that the software being delivers functions as needed. This iterative testing approach from unit testing, integration, regression, security scanning test helps Developers isolate bugs as they are identified by the established tests.

While the schedule was being developed, careful consideration was taken into account for what tasks could be run in parallel. For example, performing front-end, back-end and database work in parallel. When the work for each of these three areas is complete, integration testing will occur to ensure that the various aspects of the system are functioning as expected according to the requirements. Due to the size of the team, each module is being developed sequentially rather than parallel. The idea of having a smaller team size allowed for the same developers to work on each module of the system. By following this methodology, the development team would then be aware of the insides of each of the modules on how to integrate it, minimizing defects. The downside of following this approach, is that the project would take a considerable amount of time longer than if there was a larger development team working on each module within a phase in parallel.

There are 3 steps in the hiring process: first round interview, technical interview and third round interview. The first-round interview filters out candidates who doesn’t meet the requirements or a good background. The technical interview focuses on evaluating the technical and operational capabilities of the candidates. The third-round interview contains behavior questions that evaluates the character of the candidate, in order to determine if the candidate is a good match for the corresponding team. Once the candidate passes all three rounds, the candidate can join the corresponding department immediately. The candidate resources are from both social recruitment and internal recommendation. The core team contains 1 human resources director, 1 project manager, 1 business analyst director, 1 infrastructure engineer director, 1 database administrator director, 1 performance tester director, 1 security vulnerability assessor, 1 developer director, 1 medical staff director, 1 automation tester director and 1 technical architect director.

In all 4 phases, the software development of each module has high risk in project schedule. It is based on the anticipation of various unexpected situations such as developer issue, technology limitation, communication problem, system failure, etc. Therefore, the final estimation time of software development is always the pessimistic estimation time in the project schedule. In pre-development phase, the requirements generation and funding release will be major risks. Since in pro-development phase everything is a fresh start, it is expected to take longer time and greater risks in requirements gathering. As for funding release, banks need time to accept, verify, approve and transfer funding among many companies, thus the risk of funding release estimation is high. In phase 1 to phase 3, since all modules are fresh starts except patient record module in phase 2 and external inventory module in phase 3, the software design and software design process of each module has high risk.

### References

DevSecOps. (2020). In *Gartner IT Glossary*. Retreived from <https://www.gartner.com/en/information-technology/glossary/devsecops> .

Square. (n.d.). How to Determine Consulting Fees. Retrieved from <https://squareup.com/us/en/townsquare/consulting-fees>

Salary: Performance Tester. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/performance-tester-salary-SRCH_KO0,18.htm>

Salary: Medical Receptionist. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/medical-receptionist-salary-SRCH_KO0,20.htm>

Salary: Technical Architect. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/technical-architect-salary-SRCH_KO0,19.htm>

Salary: IT Project Manager. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/it-project-manager-salary-SRCH_KO0,18.htm>

Salary: IT Business Analyst. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/it-business-analyst-salary-SRCH_KO0,19.htm>

Salary: Infrastructure Engineer. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/infrastructure-engineer-salary-SRCH_KO0,23.htm>

Salary: Database Administrator. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/database-administrator-salary-SRCH_KO0,22.htm>

Salary: Front End Developer. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/front-end-developer-salary-SRCH_KO0,19.htm>

Salary: Backend Developer. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/backend-developer-salary-SRCH_KO0,17.htm>

Salary: Information Security Analyst. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/information-security-analyst-salary-SRCH_KO0,28.htm>

Salary: Quality Assurance Analyst. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/quality-assurance-analyst-salary-SRCH_KO0,25.htm>

Salary: Automation Engineer. (n.d.). Retrieved from <https://www.glassdoor.com/Salaries/automation-engineer-salary-SRCH_KO0,19.htm>