A database for managing a fictional small public hospital somewhere in Louisiana

The Serpents

Project Portfolio

11/13/2023

[Introduction 2](#_Toc125988975)

[The Serpents Team 3](#_Toc125988976)

[System Requirements 4](#_Toc125988977)

[Requirements 4](#_Toc125988978)

[Epics 4](#_Toc125988979)

[Epic #1 4](#_Toc125988980)

[User Stories 4](#_Toc125988981)

[User Story #1 4](#_Toc125988982)

[User Story #2 4](#_Toc125988982)

[Project Management 5](#_Toc125988983)

[Continuity of Operations Plan (COOP) 5](#_Toc125988984)

[Project Plan 0](#_Toc125988985)

[System Architecture Design and Development < Milestone 1: Proposal & Milestone 2: Architecture> 0](#_Toc125988986)

[System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation> 0](#_Toc125988987)

[Project Postmortem <Postmortem> 0](#_Toc125988988)

[Project Wins 0](#_Toc125988989)

[Root Cause Analysis 0](#_Toc125988990)

[Lessons Learned 0](#_Toc125988991)

[System Design <Milestone 2: System Architecture> 1](#_Toc125988992)

[System Architecture <Milestone 2: System Architecture> 1](#_Toc125988993)

[Component Design 1](#_Toc125988994)

[Data Flow 1](#_Toc125988995)

[System Components <Milestone 3: System Implementation> 1](#_Toc125988996)

[Component [Component Name 1] 1](#_Toc125988997)

[Component [Component Name 2] 1](#_Toc125988998)

[Component [Component Name n] 1](#_Toc125988999)

[Design Pattern <Milestone 3: System Implementation> 1](#_Toc125989000)

[Design Pattern <Milestone 3: System Implementation> 2](#_Toc125989001)

[System Implementation <Milestone 3: System Implementation> 3](#_Toc125989002)

[Project portfolio template directives and placeholders (delineated by “[ ]” or “< >” and/or highlighted or optional sections not included) should be removed from the document prior to submission. Empty sections for inclusion in later submissions may remain in the document for early submissions.]

[IMPORTANT: All diagrams developed using Enterprise Architectures must include the following acknowledgement: “Thanks to SPARX Systems for LSU student and faculty use of Enterprise Architect for academic purposes”.]

# Introduction

These features will require the use of JavaScript to provide the user interface for entering the information needed in the database. Our database will be PostgreSQL, that way we can record patient information.

Core Features:

* Reads and stores patient information: patients’ appearance, physical characteristics, symptoms (coughs, rash) and medical and family history.
* Provide patient lookup to look recall and display patient history and information.
* Appointment System that accepts patients and stores their status.

Viable Features:

* Records of employees and their information: doctors, residents, years of experience
* Inventory of medicines and/or equipment.

Stretch Features

* Working hours for employees
* Universities that the hospital is doing fellowship with

# The Serpents Team

The team is structured in a loose fashion.

Paul will handle scheduling, Java, and C++ coding and will try to take the initiative when opening new software and running it. He should be the first to share his screen when demonstrating things on Discord, and he manages the Google DOCX that the team uses to work in parts of the project template collaboratively. He has experience in Java, Python, Linux Terminal, and Overleaf, as well as experience using most of the apps in Google Drive and Microsoft Office, including Collaboratory. He’ll try to coordinate what he can in the first milestone and pick up with C++ coding after.

Cory has a friend who is a resident radiologist and can ask him questions regarding his work. He handles JavaScript/TypeScript, React, PostgreSQL, Node.js. He will take a role in the first milestone trying to get functionality between JavaScript and C++.

Ramon has a mother who is a registered nurse and can be asked about what type of information is asked about a patient and any other information needed. Has experience with Java and beginner level of C++.

Colby Blank: Front-end developer involving React, typescript, tailwind, JavaScript, and C++. Will help with formatting and functionality of the main page. He will take a role in the first milestone trying to get functionality between JavaScript and C++.

Ethan has prior experience in C++ via personal projects and can handle some of the C++ programming. He also has a loose understanding of Windows API and can help with that should the project go in the Windows direction. In the first milestone he can offer functionality ideas and help plan the overall structure of the program.

Joshua Pugh has experience in Java and some basic knowledge of C and C++

[Project Repository](https://github.com/hkaiserteaching/csc3380-fall-2023-project-group-13)

# System Requirements

## Requirements

The system requires the ability to run C++ code and PostgreSQL as well as some storage.

## Epics

### Epic #1

As a Hospital Administrator, I was to keep track of the hospital’s things and have the information available in a concise format, so I can distribute needed information in an effective manner.

## User Stories

### User Story #1

As a medical supervisor I want to have a database of inventories, so I can manage the hospital’s patients and make sure resources are accounted for.

### User Story #2

As an employee, I want to know what I need to handle today, so I can be informed about my work flow and have direction through the entire shift.

# Project Management

## Continuity of Operations Plan (COOP)

Communication for the group will be done through multiple discord servers/GroupMe. Coordination involves set meetings either in person or over a voice call in which the group will conduct work on the project. This would not be a set time but would incorporate a set of goals to be accomplished in that time. In the event of anyone becoming unavailable the group would pick up the slack and redistribute the main workload. In the case of extended absence or outright removal of the group, a new evaluation will be done of the current roles and redistributed to meet the team’s needs.

The focus of these procedures would allow for weekly meetings in which set goals can be discussed and strategized. If work can be done independently it would be handed to any participant deemed appropriate to take care of said work. This, along with the absence procedure, can allow for every group member to make adequate contributions no matter the level of their programming/organization.

## Project Plan

### System Architecture Design and Development < Milestone 1: Proposal & Milestone 2: Architecture>

[Milestone 1 (Proposal): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 2 (Architecture). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 2 (Architecture): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| 1 | Creating a simple input page for entering patient information. Critical concern is how to go about connecting this formation to a C++ service/backend. |  |  |  |  |  |  |  |

### System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>

[Milestone 2 (Architecture): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 3 (System Implementation). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 3 (System Implementation): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
|  |  |  |  |  |  |  |  |  |

## Project Postmortem <Postmortem>

### Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

### Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions. ]

### Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

# System Design <Milestone 2: System Architecture>

[*Include a short (1-2 sentences) statement about the system design*.]

## System Architecture <Milestone 2: System Architecture>

[*A short description of the system architecture.*]

### Component Design

[*Insert image of system architecture component diagram. Include the name of the team member that created the diagram in EA.*]

[*Architecture overview, to include user I/O, external data sources, and major system components.* ]

### Data Flow

[*Insert image of system architecture data flow diagram. Include the name of the team member that created the diagram in EA.*]

[*Architecture data flow discussion: a high-level description of the data between both internal major components and external data sources.*]

## System Components <Milestone 3: System Implementation>

[*Include a component sub-section for each component in the architecture diagram. Each component subsection will include a class diagram*]

### Component [Component Name 1]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

### Component [Component Name 2]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

### Component [Component Name n]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least design patterns covered in class. Include the name of the team member that created the diagram in EA.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. Include the name of the team member that created the diagram in EA. A second design pattern may be included for bonus points.*]

# System Implementation <Milestone 3: System Implementation>

[*In the table below, include a row for each component in your System Architecture diagram. In the second column, list the programming language(s) used to implement the component and the what % of that programming language is used in the implementation. In the third column, list the team member(s) that implement the component and what % of that implementation was completed by that team member. IMPORTANT NOTE: All architectural components must be implemented by an object-oriented programming language: Java, C++, or C#.*]

|  |  |  |
| --- | --- | --- |
| **Architectural Component** | **Programming Language(s) %** | **Team Member(s) %** |
| *[Data Manager]* | *[C++ (45%)*  *Java (55%)]* | *[Mickey Mouse (15%)*  *Donald Duck (20%)*  *Daisy Duck (40%*  *Pluto (25%)]* |