**Object Oriented Programing Project Report**

Figure 1



**Hospital Management System**

**Prepared by**

**at**

**Seneca Polytechnic**

**March 2024**

**Table of Contents**

[Contributions 3](#_Toc140089285)

[I Presentation YouTube link 3](#_Toc140089286)

[II Project Description 3](#_Toc140089287)

[1 Project Overview 3](#_Toc140089288)

[2 The Purpose of the Project 3](#_Toc140089289)

[2a The User Business or Background of the Project Effort 3](#_Toc140089290)

[3 The Scope of the Work 4](#_Toc140089291)

[III Requirements 4](#_Toc140089292)

[4 Product Use Cases 4](#_Toc140089293)

[IV Design 4](#_Toc140089294)

[5 UML Class Diagram 4](#_Toc140089295)

[V Testing and Evaluation 4](#_Toc140089296)

[6 Features to be tested. 4](#_Toc140089297)

[7 Pass/Fail Criteria 4](#_Toc140089298)

[VI Project Issues 4](#_Toc140089299)

[8 Lessons Learnt 4](#_Toc140089300)

[VII Conclusion 4](#_Toc140089301)

[VIII References 4](#_Toc140089302)

### ****Contributions****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student ID** | **Name** | **Components** | **Details** | **Development hours** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Presentation YouTube link

# Project Description

## Project Overview

The Hospital Management System aims to facilitate efficient management of patient records, appointment scheduling, and medical staff tracking within a hospital or healthcare facility. It provides a comprehensive solution for organizing patient information, managing appointments, and ensuring seamless communication between medical staff and patients.

## The Purpose of the Project

### The User Business or Background of the Project Effort

Content

The Hospital Management System addresses the need for a centralized platform to manage patient data and appointments effectively. It caters to the requirements of healthcare professionals by streamlining administrative tasks, optimizing resource utilization, and enhancing patient care delivery.

Motivation

The motivation behind the project lies in improving the overall efficiency and effectiveness of hospital operations. By automating various tasks such as appointment scheduling, patient record management, and medical staff tracking, the system aims to reduce administrative burden, minimize errors, and enhance the overall quality of healthcare services.

Considerations

Considerations include the severity of existing challenges in hospital management, the significance of addressing these challenges, and the potential impact of implementing an efficient Hospital Management System on patient care, staff productivity, and resource utilization.

## The Scope of the Work

The scope of the project encompasses the design, development, and implementation of a Hospital Management System that meets the specified requirements outlined in the project guidelines. This includes the creation of patient class hierarchies, implementation of exception handling mechanisms, integration of STL containers for data storage and manipulation, appointment scheduling functionalities, management of medical staff, and user interface development.

# Requirements

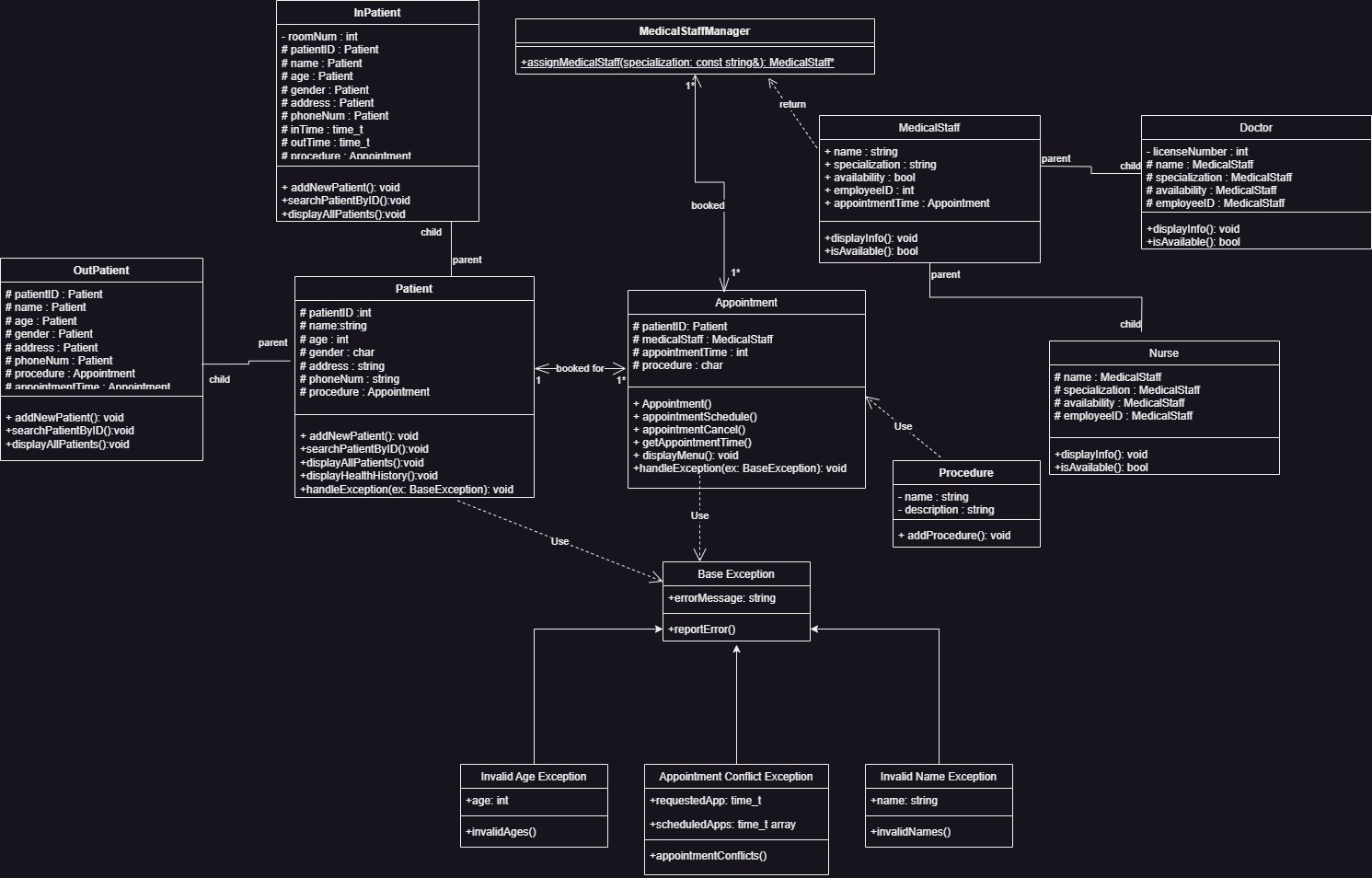
## Product Use Cases

* Use Case 1: Add New Patient  
  Input: user input  
  Output: a pointer to an object of Patient, InPatient or OutPatient   
  Summary:   
  This use case prompts the user to input demographic information and then generate a pointer to the object that stores all the information. The pointer should be compatible with all classes before being specified.
* Use Case 2: Search for Patient by ID  
  Input: user input  
  Output: a pointer to an object of Patient, InPatient or OutPatient  
  Summary:  
  This use case prompts the user to input a patient ID and then tries to match the ID with existing records. If there is a match, a pointer to the object that stores all the information will be returned. If there is no match, a message will show up, and user will be guided to add new patient. The pointer should be compatible with all classes before being specified.
* Use Case 3: Schedule Appointment  
  Input: user input and a list of Appointment objects  
  Output: the list of Appointment objects being updated  
  This use case prompts the user to input details of an appointment. Inputs include patient ID, specialization of the doctor/nurse. The system will then check for the availability of the medical staff and prompt the user to input the desired appointment time in text. If everything is checked, a new Appointment class object will be created and pushed to the back of the list. At the same time, the medical staff’s availability is also updated.
* Use Case 4: Assign Procedure   
  Input: user input, patient ID  
  Output: a Procedure class object recording the steps a patient will go through  
  This use case prompts the user to input details of an appointment as customized input. The Procedure object is linked to an Appointment object to record the details of the appointment.
* Use Case 5: Manage Medical Staff

Input: user input  
Output: a vector of MedicalStaff objects  
This use case prompts the user to add medical staff (doctors and nurses) and records their availability. The system will ask the user to put in details including the medical staff’s ID, name,

# Design

## UML Class Diagram



# Testing and Evaluation

## Features to be tested.

* Patient record management  
  10 patient records were used to test the program. The testing set consisted of a mixture of inpatients and outpatients for the specific purpose of testing whether the smart pointer was functioning.   
  Then, we prompted the system to search with ID to return a pointer to a specific patient object, then display the detailed information of that patient.
* Appointment scheduling
* Medical staff management
* Procedure assignment

## Pass/Fail Criteria

**Pass:** All implemented functionalities perform as expected without errors.

**Fail:** Any functionality that does not meet the specified requirements or produces unexpected results.

* Patient record management: **Pass**Addnew patient:文本

  描述已自动生成文本

  描述已自动生成  
  文本

  描述已自动生成  
  Search by ID: 文本

  描述已自动生成

# Project Issues

## Lessons Learned

* Importance of comprehensive planning and requirements analysis.
* Effective utilization of object-oriented principles and design patterns.
* Significance of thorough testing and validation to ensure system reliability.
* Challenges encountered during implementation and strategies for overcoming them.
* Container of pointer:  
  The combination of usage of STL container and pointers can be confusing. In the definition of patient management system, we used a smart pointer (unique\_ptr) to point to Patient class objects and its derived class objects, and then created a list to store the pointer using push\_back function. According to Anonymous et al. (2015), upon being pushed back, the ownership of the underlying linkage is actually transferred into the container, and the pointer becomes a null pointer. This is similar to a deep copy in that the activity of the pointer after this will not change the value of the one being stored into the container. To access the stored value afterwards, an iterator or a search function must be created.  
  图表

  中度可信度描述已自动生成

# Conclusion

Write the conclusion of the project by discussing the issues, solutions and the outcome.

# References

Image link: https://pixabay.com/illustrations/health-care-medicine-healthy-2082630/

SAMPLE REFS:

[1] Deitel, P. J., & Deitel, H. M. (2014). C++ How to Program. How to program series. *Prentice Hall*, *3*, 42..

[2] Anonymous et al (2015). Vector of pointer points to NULL when using push\_back(). *Stack Overflow*. <https://stackoverflow.com/questions/56637297/vector-of-pointer-points-to-null-when-using-push-back> (Retrieved Mar. 23, 2024)