**New Zealand Diploma in Information Systems**

**HTCS5607 IS Application Project**

**TECHNICAL REPORT TEMPLATE**

**Project Name:** saint albert

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Team Name(s)** | **Student ID** | **Email** | **Phone** |
| ***Project Manager***  *Jorge Campero* | 1551601 | jorgecamperogarcia@gmail.com | 0224715962 |
| ***Deputy Project Manager***  *Jorge Campero* | 1551601 | jorgecamperogarcia@gmail.com | 0224715962 |
| ***Team Members***  *Jorge Campero* | 1551601 | jorgecamperogarcia@gmail.com | 0224715962 |
|  |  |  |  |
|  |  |  |  |

**Client Stakeholders**

|  |  |  |
| --- | --- | --- |
| **Client Stakeholders** | **Full name and title** | **Contact details** |
| **Project Sponsor(s)** | *Lei Song, Simon Dacey* | *[phone, email]* |
| ***[add other roles as appropriate]*** | *[full name and title]* | *[phone, email]* |

**DATE OF SUBMISSION**

*24/11/2021*

Table of Contents

[1. Document Control 4](#_Toc80799791)

[1.1 Version History 4](#_Toc80799792)

[1.2 Contribution to Report sections 4](#_Toc80799793)

[1.3 Glossary 4](#_Toc80799794)

[2. Executive Summary 5](#_Toc80799795)

[3. Introduction 6](#_Toc80799796)

[4. Technology Review 7](#_Toc80799797)

[5. IT Methodology 9](#_Toc80799798)

[6. Project Management 10](#_Toc80799799)

[6.1 Project Management Narrative 10](#_Toc80799800)

[6.2 Project Plan with Milestones 10](#_Toc80799801)

[6.3 Project Governance Responsibilities 10](#_Toc80799802)

[6.4 Project Meetings 10](#_Toc80799803)

[6.5 Project Reports 11](#_Toc80799804)

[6.6 Project Risk and Issue Analysis 11](#_Toc80799805)

[7. Requirements Analysis 12](#_Toc80799806)

[7.1 Introduction 12](#_Toc80799807)

[7.2 Use Case Diagram 13](#_Toc80799808)

[7.3 Business Use Case Narratives (Descriptions) 14](#_Toc80799809)

[7.4 Activity Diagrams 22](#_Toc80799810)

[7.5 Overall Class Diagram 30](#_Toc80799811)

[8. Project Design 31](#_Toc80799812)

[8.1 Introduction 31](#_Toc80799813)

[8.2 Software List 31](#_Toc80799814)

[8.3 Version Control Software 31](#_Toc80799815)

[8.4 Design Use Case Narratives (Descriptions) 32](#_Toc80799816)

[8.5 Sequence Diagrams 40](#_Toc80799817)

[8.6 Deployment Diagram 47](#_Toc80799818)

[8.7 Database Design 48](#_Toc80799819)

[8.8 Annotated User Interface Designs 51](#_Toc80799820)

[8.9 Test Plan 56](#_Toc80799821)

[9. Project Training 57](#_Toc80799822)

[9.1 End User Background and Training Objectives 57](#_Toc80799823)

[9.2 Training Materials 57](#_Toc80799824)

[9.3 Training Deliverables 57](#_Toc80799825)

[9.4 Evaluation 57](#_Toc80799826)

[10. Conclusion & Lessons Learned 58](#_Toc80799827)

[References 59](#_Toc80799828)

[Appendices 60](#_Toc80799829)

# 1. Document Control

## 1.1 Version History

This document has had the following revisions:

| **Version** | **Date** | **Author** | **Description of Change** |
| --- | --- | --- | --- |
| 0.1 |  |  | Initial draft |

## 1.2 Contribution to Report sections

| **Project Team Member name** | **Student ID** | **Report Section** |
| --- | --- | --- |
| Jorge Campero | 1551601 | All |
|  |  |  |
|  |  |  |
|  |  |  |

## 1.3 Glossary

To provide clarity, terms and acronyms used in this document are defined as follows:

| **Term / Abbreviation** | **Definition** |
| --- | --- |
| Supervisor | Technical Advisor |
|  |  |

# 2. Executive Summary

The aim is to develop a database application for the Saint Albert Hospital. The current paper-based system is not able to keep the records and it does not prevent errors from happening. In the first section we find the project management, which in-depth detail shows what the workflow would be for the development (including timelines and milestones). The second section shows the necessities and requirements that Saint Albert Hospital is expecting from the final product. The third section displays the actual base from where the development will start and how the product is expected to look like. Finally, we have the training, which consist of a video which in detail, shows non-technical users about how to use the database application.

# 3. Introduction

Saint Albert Hospital is looking to replace their paper-based system. The senior administrator wants to keep track of the costs of the patients’ admissions, but the current system is unable to meet that requirement and there have been problems with the doctor’s fees and medication’s cost. The aim of this report is to provide an in-depth analysis of how a database system would look like and how it could be developed and maintain. As the system is aimed at non-technical users, this should as easy as possible to be used. The report analyses the requirements and put in perspective how would the system flow would look like and how the final product for the end user would be like.

# 4. Technology Review

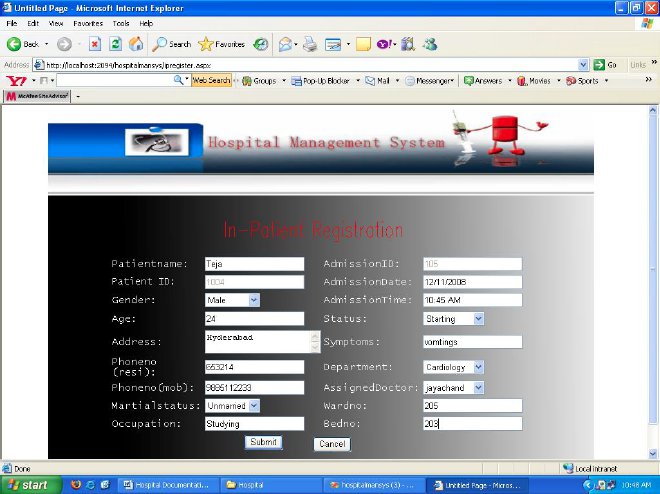
Some of the modern GUI languages are C#, Python, C++, and Java. C# is mainly used for windows applications, while Python, C++ and Java are cross-platform. These languages have different frameworks that helps with the development of the user interface, for instance Python has Tkinter, PyQt5 and Kivy, each of these frameworks have its advantages when developing an UI, being Tkinter the most popular one due to its level of customization. For C++ we can find Qt, wxWidgets, JUCE, CEGUI, and CEF. For Java we can use AWT, Swing and JavaFX, being JavaFX the most popular one nowadays, while AWT and Swing are more of a legacy framework. For this project, I need to use one language that offers a balance between functionality and design, in this case C# is the best one as it offers great possibilities for design.

We have also to consider the IDE (integrated development environment) of each language, for C# we have Visual Studio which is without a doubt the best IDE for this language, its grab and drop system for the GUI part makes Visual Studio easy to use. For Python we have PyCharm which is perfect for testing and for uploading the code to GitHub. And for the other languages Visual Studio Code can be used as it is relatively easy to use and support several programming languages. As I said before, both Python and C# have great IDEs but for the purpose of this project, Visual Studio is the best because for the development of the user interface it offers the option of grabbing the objects and paste them on the form.

For storing the data, we have several options such as MySQL, Microsoft Access, and OpenOffice Base, which are all local databases. For server like databases, we have Azure MySQL, Amazon Aurora, and MongoDB Cloud. For this project, Microsoft Access is the best as it meets the requirements, and it is compatible with C#.

As for similar projects done in the past, we have this project which have similar use cases as the ones describe in the Saint Albert project. Here are some screenshots from that project. (Code with C, 2015)





As we can see in that project, the developer had created an application which allows the hospital to register a patient and to create an Admission at the same moment. It generates a unique identifier automatically (PatientID and AdmissionID)

The project is essential as the Saint Albert Hospital is lagging and the patient’s satisfaction might decrease if Saint Albert Hospital do not upgrade their system.

# 5. IT Methodology

There are several types of development methodologies such as Agile and Waterfall, each of those have their advantages and disadvantages. Some of the advantages of the Agile methodology is that it is quicker and allows for more flexibility in the development of the project, while the Waterfall methodology is that structure is clear from the start, and it allows the team to plan for a specific timeline. It is better suit for small projects. For this project I have chosen the waterfall methodology as the project is small and the goals are clear.

The phases of the waterfall methodology are Information Gathering/Requirements, Analysis, Design, Development, Deployment and Maintenance.

# 6. Project Management

## 6.1 Project Management Narrative

The project followed the waterfall methodology for development, the first step was to gather information about the project itself and similar projects, and that is shown in the technology review. The second step was to analyse the information to create the use cases which later would be used to for the design and development stage. Moreover, to keep record of the changes, a repository in GitHub was created in which every time a document was modified, the repository would be updated. The link to the repository is [JorgeCampero/5607 (github.com)](https://github.com/JorgeCampero/5607) .

## 6.2 Project Plan with Milestones

The project started on the 20th of September 2021 and was expected to be finished on the 21st of November 2021. The first milestone is the information gathering, which is expected to start on the 20th of September and finish on the 24th of September. The second milestone is the analysis which is expected to start on the 25th of September and finish on 7th of October. The third milestone is the design which is expected to start on the 18th of October and finish on the 27th of October. The fourth milestone is development which is expected to start on the 1st of November and finish on the 13th of November. The fifth milestone is deployment which is expected to start on the 15th of November and finish on 18th of November. And the last milestone is Maintenance which is expected to on th2 20th of November and finish on the 21st of November.

## 6.3 Project Governance Responsibilities

Due to Covid-19, all the project was carried out by Jorge Campero. He followed the initial plan and used software such as excel and visual paradigm to be able to test the product.

## 6.4 Project Meetings

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Duration** | **Participants** | **Type** |
| 29/09/2021 | 15 minutes | Lei Song, Jorge Campero | Online |
| 21/10/2021 | 15 minutes | Simon Dacey, Jorge Campero | Online |
| 27/10/2021 | 15 minutes | Lei Song, Jorge Campero | Online |
| 31/10/2021 | 15 minutes | Simon Dacey, Jorge Campero | Online |
| 10/11/2021 | 15 minutes | Lei Song, Jorge Campero | Online |
| 18/11/2021 | 15 minutes | Simon Dacey, Jorge Campero | Online |

## 6.5 Project Reports

The projects status reports were used to update how the project was tracking and if there were any issues.

## 6.6 Project Risk and Issue Analysis

There were several risks during the project, such as the possibility of an extended lockdown and the possibility of catching covid which would have affected the timelines.

# 7. Requirements Analysis

## 7.1 Introduction

The St Albert Hospital needs a database application to keep track of costs of the admissions, to do this the goal is to develop a database application which would automate the process of keeping the records while also allowing to add new records, updating them, and deleting them. Moreover, the application should be capable of generating reports. In this section, the requirements of the system will be explained.

## 7.2 Use Case Diagram

Diagram

Description automatically generated

## 7.3 Business Use Case Narratives (Descriptions)

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 10 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to add a selected patient’s admission details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Add Admission” function.  Step 2- The system displays the “Add Admission” form with all fields blank.  Step 3- The assistant administrator enters the admission’s details (admission description and admission date).  Step 4- The system checks that the details are filled in correctly.  Step 5- The system displays a list of patients (patient id, last name, and first name).  Step 6- The assistant administrator selects a patient.  Step 7- The system displays a list of wards (ward id and ward name).  Step 8- The assistant administrator selects a ward.  Step 9- The assistant administrator elects to add the admission.  Step 10- The system checks that the details are filled in correctly.  Step 11- The system saves the admission’s details.  Step 12- The system displays the message “Admission added successfully”.  Step 13- The system displays the “Add another admission?” prompt.  Step 14 - The assistant administrator elects to close the form.  Step 15- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4A1- The system identifies that some details are missing.  Step 4A2- The system prompts for the completion of the details. | |
| Step 9A1- The assistant administrator elects to cancel.  Step 9A2- The system closes the form. | |
|  | Step 14A1- The assistant administrator elects to add another admission.  Step 14A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 11 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to update a selected admission’s details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Update Admission” function.  Step 2- The system displays the “Update Admission” form with a list of all the current admissions (admission id and description).  Step 3- The assistant administrator selects the admission that has details that needs updating.  Step 4- The system displays the admission’s details (admission id, description, admission date, status, patient last name, patient first name, and ward name).  Step 5- The assistant administrator updates the relevant details (description, status (either current or complete only), and admission date only).  Step 6- The system validates the entries in the fields.  Step 7- The system prompts for confirmation to change the admission’s details.  Step 8- The assistant administrator confirms the change of details.  Step 9- The system saves the admission’s details.  Step 10- The system displays the message “Admission updated successfully”.  Step 11- The system displays the “Update another admission?” prompt.  Step 12- The assistant administrator elects to close the form.  Step 13- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator elects to cancel the operation.  Step 3A2- The system closes the form. | |
| Step 7A1- The system identifies missing or incorrect fields.  Step 7A2- The system prompts for completion of the entry. | |
|  | Step 8A1- The assistant administrator elects to cancel the changes.  Step 8A2- The system closes the form. | |
|  | Step 12A1- The assistant administrator elects to update another admission.  Step 12A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |
| **USE CASE NAME:** | Delete Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 12 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to delete a selected admission’s details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Delete Admission” function.  Step 2- The system displays the “Delete Admission” form with a list of all the closed admissions (admission id and description).  Step 3- The assistant administrator selects the admission that requires deleting.  Step 4- The system displays the admission’s details (admission id, description, admission date, and status).  Step 5- The assistant administrator elects to delete the admission.  Step 6- The system deletes all payments associated with the admission.  Step 7- The system deletes the admission.  Step 8- The system displays the message “Admission deleted successfully”.  Step 9- The system displays the “Delete another admission?” prompt.  Step 10- The assistant administrator elects to close the form.  Step 11- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator elects to cancel the operation.  Step 3A2- the system closes the form. | |
| Step 5A1- The assistant administrator elects to cancel the operation.  Step 5A2- the system closes the form. | |
|  | Step 10A1- The assistant administrator elects to delete another admission.  Step 10A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Admissions Report | **USE CASE TYPE** |
| **USE CASE ID:** | 13 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | The assistant administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to produce the admissions report. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Admissions Report” function.  Step 2- The system displays the “Admissions Report” form.  Step 3- The assistant administrator selects the option to generate the report.  Step 4- The system gets the details (admission ID, description, admission date, and status) of each admission.  Step 5- The system gets the patient’s last name and first name for each admission.  Step 6- The system gets the name of each medication prescribed to each admission.  Step 7- The system then displays the admissions report (admission ID, description, admission date, status, patient last name, patient first name, and medication names for each admission).  Step 8- The system closes the form.  Step 9- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator elects to cancel without generating the report.  Step 3A2- The system closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Prescription | **USE CASE TYPE** |
| **USE CASE ID:** | 15 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Pharmacy Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the pharmacy administrator to remove a prescription from a selected admission. | |
| **PRE-CONDITIONS:** | The pharmacy administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The pharmacy administrator selects the “Remove Prescription” function.  Step 2- The system displays the “Remove Prescription” form with a list of all the current admissions (admission id and description) that have prescriptions.  Step 3- The pharmacy administrator selects the admission to remove the prescription from  Step 4- The system displays the admission’s details (admission id, description, and patient’s last name and first name).  Step 5- The system displays a list of the prescriptions (medication name, prescription date, and amount) prescribed to the admission.  Step 6- The pharmacy administrator selects a prescription.  Step 7- The pharmacy administrator elects to remove the prescription from the admission.  Step 8- The system deletes the prescription details.  Step 9- The system displays the message “Prescription removed successfully”.  Step 10- The system displays the “Remove another prescription?” prompt.  Step 11- The pharmacy administrator elects to close the form.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The pharmacy administrator elects to cancel the operation.  Step 3A2- The system closes the form. | |
| Step 7A1- The pharmacy administrator elects to cancel the operation.  Step 7A2- The system closes the form. | |
|  | Step 11A1- The pharmacy administrator elects to remove another prescription  Step 11A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 27 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the research administrator to add research to a selected doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Add Research Project” function.  Step 2- The system displays the “Add Research Project” form with a list of all the doctors (doctor id, last name, and first name).  Step 3- The research administrator selects the doctor to add the research project to.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty).  Step 5- The system displays the research projects’ details (outcome, budget, and research topic description) for each research project already linked to the selected doctor.  Step 6- The system displays a list of the research topic (research topic id, description, and level).  Step 7- The research administrator selects a research topic.  Step 8- The research administrator enters the research project’s outcome, end date and budget.  Step 9- The research administrator elects to add the research project.  Step 10- The system saves the research project details (research project id, doctor, research topic, outcome, end date, and budget).  Step 11- The system displays the “Research project added successfully” message.  Step 12- The system displays the “Add another research project?” prompt.  Step 13- The research administrator elects to close the form.  Step 14- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator elects to cancel the operation.  Step 3A2- The system closes the form. | |
| Step 9A1- The research administrator elects to cancel the operation.  Step 9A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to add another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Project | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to remove research from selected doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Remove Research Project” function.  Step 2- The system displays the “Remove Research Project” form with a list of all the doctors (doctor id, last name, and first name) who have research projects.  Step 3- The research administrator selects the doctor to remove the research project from.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty).  Step 5- The system displays the research projects’ details (outcome, budget, and research topic description) for each research project linked to the selected doctor.  Step 6- The research administrator selects the research project to remove.  Step 7- The research administrator elects to remove the research project.  Step 8- The system deletes the research project’s details.  Step 9- The system displays the message “Research project removed successfully”.  Step 10- The system displays the “Remove another research project?” prompt.  Step 11- The research administrator elects to close the form.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator elects to cancel the operation.  Step 3A2- The system closes the form. | |
| Step 7A1- The research administrator elects to cancel the operation.  Step 7A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to remove another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |
| **USE CASE NAME:** | Produce Research Projects Report | **USE CASE TYPE** |
| **USE CASE ID:** | 32 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to produce the research projects report. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Produce Research Projects Report” function.  Step 2- The system displays the “Research Projects Report” form.  Step 3- The assistant administrator selects the option to generate the report.  Step 4- The system gets the details (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description) of each research project.  Step 5- The system then displays the wards report (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description of each research project.) sorted by ward research project id.  Step 6- The system closes the form.  Step 7- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator elects to cancel, without generating the report.  Step 3A2- The system closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

## 7.4 Activity Diagrams*Diagram, schematic Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generatedDiagram Description automatically generated*

## 

## 7.5 Overall Class Diagram

Diagram

Description automatically generated

# 8. Project Design

## 8.1 Introduction

The goal of this project is to make a database application that meets the design requirements such as formal template, and that is easy to use for non-technical users. In this section, the actual look and flow of the system will be shown.

## 8.2 Software List

Microsoft Access

Visual Paradigm

Visual Studio 2019

Excel

Word

Git

## 8.3 Version Control Software

*Table

Description automatically generated*

## 8.4 Design Use Case Narratives (Descriptions)

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 10 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to add a selected patient’s admission details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Add Admission” function.  Step 2- The system displays the “Add Admission” form with all fields blank.  Step 3- The assistant administrator enters the admission’s details (admission description and admission date).  Step 4- The system checks that the details are filled in correctly.  Step 5- The system displays a list of patients (patient id, last name, and first name) in a combo box.  Step 6- The assistant administrator selects a patient.  Step 7- The system displays a list of wards (ward id and ward name) in a combo box.  Step 8- The assistant administrator selects a ward.  Step 9- The assistant administrator clicks on the “add admission” button.  Step 10- The system checks that the details are filled in correctly.  Step 11- The system saves the admission’s details.  Step 12- The system displays the message “Admission added successfully”.  Step 13- The system displays the “Add another admission?” prompt.  Step 14 - The assistant administrator clicks on the “Return” button.  Step 15- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5A1- The system identifies that some details are missing or incorrect, displays the “Please fill in all fields correctly” message.  Step 5A2- The system returns to step 3. | |
| Step 9A1- The assistant administrator clicks on the “Return button.  Step 9A2- The system closes the form. | |
|  | Step 14A1- The assistant administrator elects to add another admission.  Step 14A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 11 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to update a selected admission’s details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Update Admission” function.  Step 2- The system displays the “Update Admission” form with a list of all the current admissions (admission id and description) in a combo box.  Step 3- The assistant administrator selects the admission that has details that needs updating.  Step 4- The system displays the admission’s details (admission id, description, admission date, status, patient last name, patient first name, and ward name) in text boxes.  Step 5- The assistant administrator updates the relevant details (description, status (either current or complete only), and admission date only).  Step 6- The assistant administrator clicks the “Update Admission” button.  Step 7- The system validates the entries in the fields.  Step 8- The system displays the message “Are you sure that you want to update this admission?”.  Step 9- The assistant administrator clicks “Yes”.  Step 10- The system saves the admission’s details.  Step 11- The system displays the message “Admission updated successfully”.  Step 12- The system displays the “Update another admission?” prompt.  Step 13- The assistant administrator clicks the “Return” button.  Step 14- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 8A1- The system identifies that some details are missing or incorrect, displays the “Please fill in all fields correctly” message.  Step 8A2- The system returns to step 5. | |
|  | Step 9A1- The assistant administrator clicks “No”.  Step 9A2- The system closes the form. | |
|  | Step 13A1- The assistant administrator elects to update another admission.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 12 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to delete a selected admission’s details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Delete Admission” function.  Step 2- The system displays the “Delete Admission” form with a list of all the closed admissions (admission id and description) in a combo box.  Step 3- The assistant administrator selects the admission that requires deleting.  Step 4- The system displays the admission’s details (admission id, description, admission date, and status) in text boxes.  Step 5- The assistant administrator clicks on the “Delete Admission” button.  Step 6- The system deletes all payments associated with the admission.  Step 7- The system deletes the admission.  Step 8- The system displays the message “Admission deleted successfully”.  Step 9- The system displays the “Delete another admission?” prompt.  Step 10- The assistant administrator clicks on the “Return” button.  Step 11- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator clicks on the “Return” button.  Step 3A2- the system closes the form. | |
| Step 5A1- The assistant administrator clicks on the “Return” button.  Step 5A2- the system closes the form. | |
|  | Step 10A1- The assistant administrator elects to delete another admission.  Step 10A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Admissions Report | **USE CASE TYPE** |
| **USE CASE ID:** | 13 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | The assistant administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to produce the admissions report. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The assistant administrator selects the “Admissions Report” function.  Step 2- The system displays the “Admissions Report” form.  Step 3- The assistant administrator clicks on the “Generate report” button.  Step 4- The system gets the details (admission ID, description, admission date, and status) of each admission.  Step 5- The system gets the patient’s last name and first name for each admission.  Step 6- The system gets the name of each medication prescribed to each admission.  Step 7- The system then displays the admissions report (admission ID, description, admission date, status, patient last name, patient first name, and medication names for each admission).  Step 8- The assistant administrator clicks on the “Return” button.  Step 9- The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Prescription | **USE CASE TYPE** |
| **USE CASE ID:** | 15 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Pharmacy Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the pharmacy administrator to remove a prescription from a selected admission. | |
| **PRE-CONDITIONS:** | The pharmacy administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The pharmacy administrator selects the “Remove Prescription” function.  Step 2- The system displays the “Remove Prescription” form with a list of all the current admissions (admission id and description) that have prescriptions in a combo box.  Step 3- The pharmacy administrator selects the admission to remove the prescription from.  Step 4- The system displays the admission’s details (admission id, description, and patient’s last name and first name) in text boxes.  Step 5- The system displays a list of the prescriptions (prescriptionID, medication Name, prescription date, and amount) prescribed to the admission in a list box.  Step 6- The pharmacy administrator selects a prescription.  Step 7- The pharmacy administrator clicks on the “Remove prescription” button.  Step 8- The system deletes the prescription details.  Step 9- The system displays the message “Prescription removed successfully”.  Step 10- The system displays the “Remove another prescription?” prompt.  Step 11- The pharmacy administrator clicks on the “Return” button.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The pharmacy administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 7A1- The pharmacy administrator clicks on the “Return” button.  Step 7A2- The system closes the form. | |
|  | Step 11A1- The pharmacy administrator elects to remove another prescription  Step 11A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 27 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the research administrator to add research to a selected doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Add Research Project” function.  Step 2- The system displays the “Add Research Project” form with a list of all the doctors (doctor id, last name, and first name) in a combo box.  Step 3- The research administrator selects the doctor to add the research project to.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty) in text boxes.  Step 5- The system displays the research projects’ details (outcome, budget, and research topic description) for each research project already linked to the selected doctor in data grid view.  Step 6- The system displays a list of the research topic (research topic id, description, and level) in a combo box.  Step 7- The research administrator selects a research topic.  Step 8- The research administrator enters the research project’s outcome, end date and budget.  Step 9- The research administrator clicks on the “Add Research” button.  Step 10- The system saves the research project details (research project id, doctor, research topic, outcome, end date, and budget).  Step 11- The system displays the “Research project added successfully” message.  Step 12- The system displays the “Add another research project?” prompt.  Step 13- The research administrator clicks on the “Return” button.  Step 14- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 9A1- The research administrator clicks on the “Return” button.  Step 9A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to add another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to remove research from selected doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Remove Research Project” function.  Step 2- The system displays the “Remove Research Project” form with a list of all the doctors (doctor id, last name, and first name) who have research projects in a combo box.  Step 3- The research administrator selects the doctor to remove the research project from.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty) in a text box.  Step 5- The system displays the research projects’ details (outcome, budget, and research topic description) for each research project linked to the selected doctor in a list box.  Step 6- The research administrator selects the research project to remove.  Step 7- The research administrator clicks on the “remove research” button.  Step 8- The system deletes the research project’s details.  Step 9- The system displays the message “Research project removed successfully”.  Step 10- The system displays the “Remove another research project?” prompt.  Step 11- The research administrator clicks on the “Return” button.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 7A1- The research administrator clicks on the “Return” button.  Step 7A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to remove another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |
| **USE CASE NAME:** | Produce Research Projects Report | **USE CASE TYPE** |
| **USE CASE ID:** | 32 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to produce the research projects report. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Produce Research Projects Report” function.  Step 2- The system displays the “Research Projects Report” form.  Step 3- The research administrator clicks the “Generate report” button.  Step 4- The system gets the details (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description) of each research project.  Step 5- The system then displays the research project report (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description of each research project.) sorted by research project id.  Step 6- The research administrator clicks on the “Return” button.  Step 7- The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

## 

## 8.5 Sequence Diagrams

### Add Admission

Diagram, schematic

Description automatically generated

### Add Research Project

### Diagram Description automatically generated

### Delete Admission

Diagram

Description automatically generated with medium confidence

### Produce Admission Report

Diagram

Description automatically generated

### Produce Research Projects Report

Graphical user interface, text, application, letter

Description automatically generated

### Remove Prescription

Diagram

Description automatically generated

### Remove Research Project

### Diagram, schematic Description automatically generated

### Update AdmissionDiagram Description automatically generated

## 

## 8.6 Deployment Diagram

Diagram

Description automatically generated

As the database is expected to be local, everything is part of the PC. One of the artefacts is the Database itself which will run the schemas and the other is Visual Studio which will run the C# app which is where the end product is located.

## 8.7 Database Design

Diagram

Description automatically generated

**Data Dictionary**

**Patients**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| PatientID | Yes | Auto-number | 8 | Primary Key  1-99999999 inclusive |
| LastName | Yes | Text | 25 | - |
| FirstName | Yes | Text | 25 | - |
| StreetAddress | Yes | Text | 50 | - |
| Suburb | Yes | Text | 20 | - |
| City | Yes | Text | 20 | - |
| EmailAddress | Yes | Text | 30 | - |
| PhoneNumber | No | Number | 15 | - |
| InsuranceCode | Yes | Text | 7 | - |

**Admissions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| AdmissionID | Yes | Auto-number | 8 | Primary Key |
| Description | Yes | Text | 30 | - |
| AdmissionDate | Yes | Date | 10 | Format: DD/MM/YYYY |
| Status | Yes | Text | 20 | Current, Complete |
| WardID | Yes | Foreign Key (WARD) | 2 | - |
| PatientID | Yes | Foreign Key (PATIENT) | 3 |  |

**Wards**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| WardID | Yes | Auto-number | 2 | Primary Key |
| Ward Name | Yes | Text | 20 | - |
| Location | Yes | Text | 10 | - |
| Capacity | Yes | Number | 2 | 1 to 20 inclusive |

**Payments**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| PaymentCode | Yes | Auto-number | 8 | Primary Key |
| AdmissionID | Yes | Foreign Key (ADMISSION) | 8 | - |
| Amount | Yes | Currency | 6 | - |
| Date | Yes | Date | 10 | Format: DD/MM/YYYY |

**Doctors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| DoctorID | Yes | Auto-number | 4 | Primary Key |
| LastName | Yes | Text | 25 | - |
| FirstName | Yes | Text | 25 | - |
| StreetAddress | Yes | Text | 50 | - |
| Suburb | Yes | Text | 20 | - |
| City | Yes | Text | 20 | - |
| Specialty | Yes | Text | 30 | - |
| PhoneNumber | No | Text | 20 | - |
| Salary | Yes | Number | 6 | 20000.00-200000.00 inclusive |

**Prescriptions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Max Length** | **Range/List** |
| PrescriptionID | Yes | Auto-Number | 8 | Primary Key |
| MedicationName | Yes | Foreign Key (ADMISSION) | 30 | - |
| PrescriptionDate | Yes | Date | 10 | Format: DD/MM/YYYY |
| Amount | Yes | Number | 3 | 1 to 150 inclusive |
| AdmissionID | Yes | Foreign Key (ADMISSION) | 8 | - |

**Medication**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Max Length** | **Range/List** |
| MedicationID | Yes | Auto-number | 6 | Primary Key |
| MedicationName | Yes | Text | 30 | - |
| Cost | Yes | Currency | 6 | 0.50 to 9999.99 inclusive |

**Research Project**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| ResearchProjectID | Yes | Auto-number | 6 | Primary Key |
| Outcome | Yes | Text | 30 | Journal article, conference paper, conference poster, book chapter, and book |
| EndDate | Yes | Date | 10 | Format: DD/MM/YYYY |
| Budget | Yes | Currency | 7 | - |
| ResearchTopicID | Yes | Foreign Key (RESEARCHTOPICS) | 3 | - |
| DoctorID | Yes | Foreign Key (DOCTORS) | 4 | - |

**Research Topic**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| ResearchTopicID | Yes | Auto-Number | 3 | Primary Key |
| Description | Yes | Text | 30 | - |
| Level | Yes | Number | 5 | - |

**Allocation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| AllocationID | Yes | Auto-number | 6 | Primary Key |
| JobFee | Yes | Currency | 5 | Between 1 to 99999 |
| Role | Yes | Text | 30 | - |
| AdmissionID | Yes | Foreign Key (ADMISSION) | 3 | - |
| DoctorID | Yes | Foreign Key (DOCTORS) | 4 | - |

## 8.8 Annotated User Interface Designs

*Diagram

Description automatically generatedChart

Description automatically generated with low confidenceDiagram

Description automatically generatedDiagram

Description automatically generatedGraphical user interface, diagram

Description automatically generatedDiagram

Description automatically generatedDiagram, schematic

Description automatically generatedDiagram

Description automatically generated*

## 8.9 Test Plan

In Appendices.

# 9. Project Training

## 9.1 End User Background and Training Objectives

The goal of this training video is to show how to use the functionalities of the forms to the end users. At the end of the training, the users will know how to add, update, and delete an admission and generate a report. Also, how to add and remove a research project and how to generate a report about the research projects.

## 9.2 Training Materials

The video attached to the document.

## 9.3 Training Deliverables



## 9.4 Evaluation

The video contains an explanation of how to use the forms. A future improvement could show the different alternative courses instead of just the main one.

# 10. Conclusion & Lessons Learned

The project has passed through different phases, each of those which have left me with different learnings. The requirement analysis phase gave me the perspective that was needed to success and to meet the expectations of this project. The design face was important as it shaped the way the final product would look like. The development was the combination of both two first faces and the training is the result. Throughout these 8 weeks, an important number of hours so that this project could be a success, and it was, as all the goals were met and some of the results were better than expected. The lesson that this experience has left me is that it is important to manage the time well and to plan for the unexpected. For future projects, I would recommend allowing more time to the development stage as it is the heart of the project and it should not be rushed.

# References

Top 10 programming languages for desktop apps in 2021. (n.d.). decipherzone.com. https://www.decipherzone.com/blog-detail/top-programming-languages-for-desktop-apps-in-2021

Costa, C. D. (2020, December 7). Top 10 Python GUI frameworks for developers. Medium. https://towardsdatascience.com/top-10-python-gui-frameworks-for-developers-adca32fbe6fc

5 cross-platform GUIs for C++. (2017, October 26). Dice Insights. https://insights.dice.com/2016/11/18/5-cross-platform-guis-for-c/

Java GUI frameworks guide. (2021, July 6). CodersLegacy. https://coderslegacy.com/java-gui-frameworks-guide/

MySQL :: MySQL 8.0 reference manual :: 1.2.1 what is MySQL? (n.d.). MySQL :: Developer Zone. https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Hospital management system ASP.NET project. (2015, May 17). Code with C. https://www.codewithc.com/hospital-management-system-asp-net-project/

Five advantages of Agile software development methodologies. (2020, May 8). EagleDream Technologies. https://eagledream.com/news/application-modernization-news/5-advantages-agile-software-development-methodologies/

The pros and cons of waterfall methodology. (2018, August 10). Intelligent Diagramming | Lucidchart. https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology

Sharma, L. (2021, August 15). WaterFall model. TOOLSQA. https://www.toolsqa.com/software-testing/waterfall-model/

The traditional waterfall approach. (n.d.). University of Missouri–St. Louis. https://www.umsl.edu/~hugheyd/is6840/waterfall.html

# Appendices

## Gantt Chart

A picture containing chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Background pattern

Description automatically generated with medium confidence

## Meeting minutes

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Graphical user interface, table

Description automatically generatedTable

Description automatically generated

## Project status reports

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

## Risks and Issues

Timeline

Description automatically generated

## Test Plans

### Database

#### Admission Table

Table

Description automatically generated

#### Patients Table

Table

Description automatically generated

Table

Description automatically generated

#### Doctors Table

Table

Description automatically generated

Table

Description automatically generated

#### Wards Table

Table

Description automatically generated with medium confidence

#### Payment Table

Table

Description automatically generated

#### Prescriptions Table

Table

Description automatically generated

#### Medication Table

Graphical user interface, table

Description automatically generated

#### Research Topic Table

Table

Description automatically generated

#### Research Project Table

Table

Description automatically generated

#### Allocation Table

Table

Description automatically generated

### Test cases

#### Add Admission

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 1 | **Test Case Description** | | This use case enables the assistant administrator to add a selected patient’s admission details. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 2, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Clicks the add admission button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | Enters the admission’s details (admission description and admission date). | | There should be text boxes to write the details in. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a patient (Andrew Fane) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator selects a ward (Apollo) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks on the “add the admission” button. | | The system should save the details and display a success message | | As expected | | | Pass | | |  |
| 6 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: " " | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Leave admission description empty to see if the system detects the errors | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the add admission button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator leaves the admission’s details empty and date (2/11/2021) | | Should allow you to type in. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a patient (Andrew Fane) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator selects a ward (Apollo) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks on the “add the admission” button. | | Displays the “Please fill in all fields correctly” message. Empty form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Administrator clicks return and form closes. | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the add admission button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | Enters the admission’s details (admission description and admission date). | | There should be text boxes to write the details in. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a patient (Andrew Fane) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator selects a ward (Apollo) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Allow the assistant administrator to add another admission. | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the add admission button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | Enters the admission’s details (admission description and admission date). | | There should be text boxes to write the details in. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a patient (Andrew Fane) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator selects a ward (Apollo) | | The assistant administrator should be able to select. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks on the “add the admission” button. | | The system should save the details and display a success message | | As expected | | | Pass | | |  |
| 6 | The assistant administrator elects to add another admission. | | Empty form. | | As expected | | | Pass | | |  |

#### Update Admission

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 2 | **Test Case Description** | | This use case enables the assistant administrator to update a selected admission’s details. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 4, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Description: Chest Pain | | | | |
| 2 |  | | |  | 2 | Status: Complete | | | | |
| 3 |  | | |  | 3 | Admission Date: 4/11/2021 | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Clicks the update admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that has details that needs updating. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator updates the details (Description, status and date) | | There should be text boxes to write the details in. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator clicks on the update admission button. | | Should display confirmation prompt. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks "yes". | | Should save the details and display success message. | | As expected | | | Pass | | |  |
| 6 | The assistant administrator clicks the "Return" button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Description: Chest Pain | | | | |  |
| 2 |  | | |  | 2 | Status: Complete | | | | |  |
| 3 |  | | |  | 3 | Admission Date: 4/11/2021 | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The assistant administrator clicks on the return button and the form closes | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the update admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator clicks on the return button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Description: Chest Pain | | | | |  |
| 2 |  | | |  | 2 | Status: Complete | | | | |  |
| 3 |  | | |  | 3 | Admission Date: 4/11/2021 | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Leave the admission description empty to see if the system identifies the error | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the update admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that has details that needs updating. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator leaves the details empty (Description, status and date) | | You should be allowed to type in | | As expected | | | Pass | | |  |
| 4 | The assistant administrator clicks on the update admission button. | | Displays the "Please fill in all fields correctly" message. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Description: Chest Pain | | | | |  |
| 2 |  | | |  | 2 | Status: Complete | | | | |  |
| 3 |  | | |  | 3 | Admission Date: 4/11/2021 | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The assistant administrator decides to do not update the admission | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the update admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that has details that needs updating. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator leaves the details empty (Description, status and date) | | You allow you to type in | | As expected | | | Pass | | |  |
| 4 | The assistant administrator clicks on the update admission button. | | Should display confirmation prompt. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks "No". | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Description: Chest Pain | | | | |  |
| 2 |  | | |  | 2 | Status: Complete | | | | |  |
| 3 |  | | |  | 3 | Admission Date: 4/11/2021 | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Allow the assistant administrator to update another admission | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the update admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that has details that needs updating. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator updates the details (Description, status, and date) | | There should be text boxes to write the details in. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator clicks on the update admission button. | | Should display confirmation prompt. | | As expected | | | Pass | | |  |
| 5 | The assistant administrator clicks "yes". | | Should save the details and display success message. | | As expected | | | Pass | | |  |
| 6 | The assistant administrator selects to update another admission. | | Empty form | | As expected | | | Pass | | |  |

#### Delete Admission

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 3 | **Test Case Description** | | This use case enables the assistant administrator to delete a selected admission’s details. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 7, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Clicks the delete admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that needs to be deleted. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator clicks on the delete admission button. | | The system deletes the admission and displays success message. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator clicks the "Return" button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the assistant administrator can close the form. | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the delete admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator clicks the "Return" button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the assistant administrator can close the form after selecting an admission | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the delete admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that needs to be deleted. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator clicks the "Return" button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | Admission Description: Broken nose | | | | |  |
| 2 |  | | |  | 2 | Admission Date: 2/11/2021 | | | | |  |
| 3 |  | | |  | 3 | Patient: Andrew Fane | | | | |  |
| 4 |  | | |  | 4 | Ward: Apollo | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the assistant administrator can delete another admission | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Clicks the delete admission button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator selects the admission that needs to be deleted. | | There should be a combo box to choose from. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator clicks on the delete admission button. | | The system deletes the admission and displays success message. | | As expected | | | Pass | | |  |
| 4 | The assistant administrator selects to delete another admission. | | Empty form. | | As expected | | | Pass | | |  |

#### Produce Admissions Report

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 4 | **Test Case Description** | | This use case enables the assistant administrator to produce the admissions report. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 12, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | All the admissions. | | | | |
| 2 |  | | |  | 2 |  | | | | |
| 3 |  | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | The assistant administrator clicks on the produce admissions report button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator clicks on the “Generate report” button. | | The system generates the report. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The assistant administrator has logged onto the system | | |  | 1 | All the admissions. | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the assistant administrator can close the form before generating the report | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The assistant administrator clicks on the produce admissions report button. | | Form should open. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |

#### Remove Prescription

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 5 | **Test Case Description** | | This use case enables the pharmacy administrator to remove a prescription from a selected admission. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 13, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The pharmacy administrator has logged onto the system. | | |  | 1 | Medication Name: Codeine | | | | |
| 2 |  | | |  | 2 | Amount: 2 | | | | |
| 3 |  | | |  | 3 | Admission: Pain | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | The pharmacy administrator clicks on the remove prescription button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The pharmacy administrator selects the admission to remove the prescription from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The pharmacy administrator selects the prescription | | Select from a combo box. | | As expected | | | Pass | | |  |
| 4 | The pharmacy administrator clicks on the “Remove prescription” button. | | The system removes the admission and display a success message | | As expected | | | Pass | | |  |
| 5 | The pharmacy administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The pharmacy administrator has logged onto the system. | | |  | 1 | Medication Name: Codeine | | | | |  |
| 2 |  | | |  | 2 | Amount: 2 | | | | |  |
| 3 |  | | |  | 3 | Admission: Pain | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the pharmacy administrator can close the form. | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The pharmacy administrator clicks on the remove prescription button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The pharmacy administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The pharmacy administrator has logged onto the system. | | |  | 1 | Medication Name: Codeine | | | | |  |
| 2 |  | | |  | 2 | Amount: 2 | | | | |  |
| 3 |  | | |  | 3 | Admission: Pain | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the pharmacy administrator can close the form after selecting a prescription | | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The pharmacy administrator clicks on the remove prescription button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The pharmacy administrator selects the admission to remove the prescription from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The pharmacy administrator selects the prescription | | Select from a combo box. | | As expected | | | Pass | | |  |
| 5 | The pharmacy administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The pharmacy administrator has logged onto the system. | | |  | 1 | Medication Name: Codeine | | | | |  |
| 2 |  | | |  | 2 | Amount: 2 | | | | |  |
| 3 |  | | |  | 3 | Admission: Pain | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the pharmacy administrator can remove another prescription | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The pharmacy administrator clicks on the remove prescription button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The pharmacy administrator selects the admission to remove the prescription from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The pharmacy administrator selects the prescription | | Select from a combo box. | | As expected | | | Pass | | |  |
| 4 | The pharmacy administrator clicks on the “Remove prescription” button. | | The system removes the admission and display a success message | | As expected | | | Pass | | |  |
| 5 | The pharmacy administrator selects to remove another prescription | | Empty form. | | As expected | | | Pass | | |  |

#### Add Research Project

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 6 | **Test Case Description** | | This use case enables the research administrator to add research to a selected doctor. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 14, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The research administrator has logged onto the system | | |  | 1 | Outcome: Journal Article | | | | |
| 2 |  | | |  | 2 | Budget: $5000 | | | | |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |
|  |  | | |  |  |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | | |  |  |  |  |
|  |  | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | The research administrator clicks the add research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to add the research project to. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a topic | | There should be a combo box to choose from | | As expected | | | Pass | | |  |
| 4 | The research administrator enters the research project’s outcome, end date and budget. | | There should be text boxes to type in. | | As expected | | | Pass | | |  |
| 5 | The research administrator clicks on the “Add Research” button. | | The system saves the details and displays success message | | As expected | | | Pass | | |  |
| 6 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can close the form. | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks the add research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can close the form after entering the details. | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks the add research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to add the research project to. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a topic | | There should be a combo box to choose from | | As expected | | | Pass | | |  |
| 4 | The research administrator enters the research project’s outcome, end date and budget. | | There should be text boxes to type in. | | As expected | | | Pass | | |  |
| 5 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can add another research project | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks the add research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to add the research project to. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator selects a topic | | There should be a combo box to choose from | | As expected | | | Pass | | |  |
| 4 | The research administrator enters the research project’s outcome, end date and budget. | | There should be text boxes to type in. | | As expected | | | Pass | | |  |
| 5 | The research administrator clicks on the “Add Research” button. | | The system saves the details and displays success message | | As expected | | | Pass | | |  |
| 6 | The research administrator selects to add another research project. | | Empty form | | As expected | | | Pass | | |  |

#### Remove Research Project

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 7 | **Test Case Description** | | This use case enables the research administrator to remove research from selected doctor. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 8, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The research administrator has logged onto the system. | | |  | 1 | Outcome: Journal Article | | | | |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | The research administrator clicks on the remove research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to remove the research project from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The research administrator selects the research project to remove. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 4 | The research administrator clicks on the “remove research” button. | | The system removes the details and display a success message | | As expected | | | Pass | | |  |
| 5 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system. | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can close the form | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks on the remove research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system. | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can close the form after selecting a research project | | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks on the remove research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to remove the research project from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The research administrator selects the research project to remove. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 4 | The research administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system. | | |  | 1 | Outcome: Journal Article | | | | |  |
| 2 |  | | |  | 2 | Budget : $5000 | | | | |  |
| 3 |  | | |  | 3 | End Date: 30/01/2022 | | | | |  |
| 4 |  | | |  | 4 | Research Topic: Pandemics | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can remove another research project | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks on the remove research project button. | | Form should open and show a combo box. | | As expected | | | Pass | | |  |
| 2 | The research administrator selects the doctor to remove the research project from. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 3 | The research administrator selects the research project to remove. | | Select from a combo box. | | As expected | | | Pass | | |  |
| 4 | The research administrator clicks on the “remove research” button. | | The system removes the details and display a success message | | As expected | | | Pass | | |  |
| 5 | The research administrator selects to remove another research project | | Empty form | | As expected | | | Pass | | |  |

#### Produce Research Projects Report

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 8 | **Test Case Description** | | This use case enables the research administrator to produce the research projects report. | | | | | |
| **Created By** | | Jorge Campero | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Jorge Campero | **Date Tested** | | November 11, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The research administrator has logged onto the system. | | |  | 1 | All research projects | | | | |
| 2 |  | | |  | 2 |  | | | | |
| 3 |  | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the functionalities described in the use case work as expected | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | The research administrator clicks on the produce research projects report button | | Form should open. | | As expected | | | Pass | | |  |
| 2 | The research administrator clicks on the “Generate report” button. | | The system generates the report. | | As expected | | | Pass | | |  |
| 3 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The research administrator has logged onto the system. | | |  | 1 | All research projects | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Verify that the research administrator can close the form before generating the report | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | The research administrator clicks on the produce research projects report button | | Form should open. | | As expected | | | Pass | | |  |
| 2 | The assistant administrator clicks on the “Return” button. | | The system closes the form. | | As expected | | | Pass | | |  |