**New Zealand Diploma in Information Systems**

**HTCS5607 IS Application Project**

**TECHNICAL REPORT TEMPLATE**

**Project Name:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Team Name(s)** | **Student ID** | **Email** | **Phone** |
| ***Project Manager***  *[Name]* |  |  |  |
| ***Deputy Project Manager***  *[Name]* |  |  |  |
| ***Team Members***  *[Name (s)]* |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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**

*dd/mm/yyyy*

Table of Contents

[1. Document Control 3](#_Toc88077754)

[1.1 Version History 3](#_Toc88077755)

[2. Executive Summary 4](#_Toc88077756)

[3. Introduction 5](#_Toc88077757)

[4. Technology Review 6](#_Toc88077758)

[5. IT Methodology 10](#_Toc88077759)

[6. Project Management 11](#_Toc88077760)

[6.1 Project Management Narrative 11](#_Toc88077761)

[6.2 Project Plan with Milestones 11](#_Toc88077762)

[6.4 Project Meetings 11](#_Toc88077763)

[6.5 Project Reports 11](#_Toc88077764)

[6.6 Project Risk and Issue Analysis 11](#_Toc88077765)

[7. Requirements Analysis 12](#_Toc88077766)

[7.1 Introduction 12](#_Toc88077767)

[7.2 Use Case Diagram 12](#_Toc88077768)

[7.3 Business Use Case Narratives (Descriptions) 12](#_Toc88077769)

[7.4 Activity Diagrams 12](#_Toc88077770)

[7.5 Overall Class Diagram 12](#_Toc88077771)

[8. Project Design 13](#_Toc88077772)

[8.1 Introduction 13](#_Toc88077773)

[8.2 Software List 13](#_Toc88077774)

[8.3 Version Control Software 13](#_Toc88077775)

[8.4 Design Use Case Narratives (Descriptions) 13](#_Toc88077776)

[8.5 Sequence Diagrams 13](#_Toc88077777)

[8.6 Deployment Diagram 13](#_Toc88077778)

[8.7 Database Design 13](#_Toc88077779)

[8.8 Annotated User Interface Designs 13](#_Toc88077780)

[8.9 Test Plan 13](#_Toc88077781)

[9. Project Training 14](#_Toc88077782)

[9.1 End User Background and Training Objectives 14](#_Toc88077783)

[9.2 Training Materials 14](#_Toc88077784)

[9.3 Training Deliverables 14](#_Toc88077785)

[9.4 Evaluation 14](#_Toc88077786)

[10. Conclusion & Lessons Learned 15](#_Toc88077787)

[References 16](#_Toc88077788)

[Appendices 18](#_Toc88077789)

# 1. Document Control

## 1.1 Version History

This document has had the following revisions:

| **Version** | **Date** | **Author** | **Description of Change** |
| --- | --- | --- | --- |
| 0.1 | 17/11/2021 | Fairuz | Initial draft |

# 2. Executive Summary

An introduction will be laid out in “Introduction” section which will talk about the purpose and background of the report. It continues into the next section where a review of the different programming languages, IDE, data access methods, similar-type hospital management applications and current project application will be discussed. This report will delve into the type of methodology that is used for the project and will go into brief detail of the phases reflected onto the method adopted.

The report will then summarise on the steps undertaken to develop the application with evidence according to the chosen system development lifecycle. Within the section, an overall plan and necessary milestones will be outlined along with meeting schedule, project status reports, and the identification of potential risks and issues. A requirement analysis with the relevant diagrams will be covered in the next section, and design phase artefacts such as annotated user interface diagrams accompanied by design narratives and test plans will be featured in the following section.

In regard to deployment, more information such as training background, objectives and a video for the purpose of training users on the application will be addressed before concluding the report with an overview, perceptive thoughts on the project and recommendations.

# 3. Introduction

Many hospitals and other medical institutions in the 21st century are no stranger to technological advancements that are often utilised to assist in the management of patients, medical staff, research and other critical aspects. Their definitive purposes vary from helping doctors decide on the appropriate diagnosis or care for a patient, to tracking medical assets in hospital wards, to providing better patient services (Álvarez López et al., 2018; Ngorsed & Suesaowaluk, 2016; Shahmoradi et al., 2017).

These technologies usually involve using computer and mobile devices to display applications that function as an interface, online interconnectivity and databases to store important data. Patients and medical staff members alike will no doubt reap the benefits from having these management systems put in place such as less workload and pressure on staff, automation, better patient experience and improved network security (Sushko, 2021).

However, there are factors to consider before deciding on the type of application to be implemented including, the functional and non-functional requirements, specific budget for the application and how the development of the application and training will take place (Sushko, 2021).

The purpose of the report aims to provide a solution for Saint Albert Hospital that has been facing issues in managing their finances. The application will be able to satisfy the hospital’s requirement of recording financial, patient and their admission information. The report will provide a clear plan, analyse on the functions required, design the application according to the functions stated and detail the execution of the application.

# 4. Technology Review

There are many programming languages that are in existence in the technological world that have been developed in the last few decades that allow developers to use as instruments of creating and storing organized information. At the time of writing of this review, two languages have already been known to the author, Python and C# - the latter being the language of focus in the project. However, there are many others that are applicable to this project which could be applied in the near future.

One such language that may be relevant is C++ which according to Tiobe’s index, is the top four most popular language (Krill, 2021). This is evident in the fact that C++ is commonly used in programming games, one of the largest sectors in the entertainment industry (Beattie, 2021). Its characteristics include high level language - a language that the common people are able to understand as oppose to binary language - that delivers productivity, utilization of objects in programming, exception handling and the ability of using the same code repeatedly (Gallagher, 2020). The only downfall of it is the sheer magnitude of C++ a beginner will have to learn as well as the complexity of it (Bolton, 2019).

Another language that is also popular and useful for this project is Perl as it is relatable to this project due to its file management properties. It performs word searches which makes it suitable for servers and databases. It is a high-level language and like C++, also uses object oriented programming but flexible in programming whereby it does not need a strict way of writing. The advantages include, easy to learn basics, versatility as it can combine with other languages and wide application in a range of IT fields such as “web development, networking and bioinformatics”. It comes pre-installed in Linux but can also be used in other operating systems with a Perl interpreter. However, the disadvantages include library modules that may be complicated at first glance, the GUI not being the most ideal among all the languages and a requirement of experience in recognizing the Patterns (Brent, 2021).

On a web framework, JavaScript is the language that was quite recently revived by AJAX (Asynchronous Javascript and XML), that is, “exchanging data with a server, without reloading the whole page”(Shekhar, 2015). It is a web development tool that is object oriented and aimed more for end-user applications but then later expanded to other applications such as, server, mobile and desktop. It allows user interaction of a web page, manages data in a database with node.js, and allows for developers to construct mobile and desktop applications (Kidder, 2019).

It is quicker when dealing with client-side applications as it runs in the client’s browser. It is easy to grasp as a beginner and use in applications. It is also a popular language implemented in almost everything that is web-based and it is flexible enough to be used with other languages and many other applications. It does not require the need of a website server and provides visual appeal to interfaces.

However, it can pose as a security risk when the code is run on the client’s computer and there is also the difficult task of interpreting Javascript for different browser adaptability (Gupta, 2019).

JavaScript is a more viable language that is easy to learn and can be implemented for this project as it is not just geared towards web-based applications but also, desktops and servers. This gives programmers more options of how they would want to build their application and explore their creativity on interfaces.

**IDE**

A basic IDE (Integrated Development Environment) that assists a programmer in creating software and applications comprises of functions including but not limited to, “source code editor, debuggers, compiler, interpreter,” and “build automation” (Green, 2021).

One of the best free IDE software to use is Visual Studio Code with the added advantage of being open source or code that is accessible and editable for everyone who knows coding languages such as C ++, Java, Javascript, PHP, and Python. It can be downloaded for operating systems including Windows, Linux and macOS. Some of its best features provided are a terminal that can be opened from within, the use of .Net, customization of additional features and easy deployment of applications. Its implementation however, may not be suitable for huge coding work and its simplified features may not be easy for some to work with (Green, 2021).

Another good IDE for programmers to use would be SlickEdit that consists of many great features including SmartPaste, which allows copying and pasting of code with indentations still intact. On top of that, hardcore programmers will be delighted to learn that it allows cross-platform coding in 70+ languages along with the utilisation of a variety of debuggers. It provides version control systems that enable users to back up and compare symbol changes in the respective files (Kumar, 2021). Unlike the Visual Studio Code, it is expensive just for a single license which makes it more suitable for a company of professional programmers (Powers, 2012).

Finally, the Eclipse is a free IDE not just for professionals but also for beginners and takes on languages including “C, C ++, Fortran, Javascript, PHP, Python, Ruby, and others” (Green, 2021). It also features the ability to install Junit, a regression testing tool for Java (Chacon, 2018). It can also remotely debug code with Java virtual machine, allows completion of a .net application on the IDE itself, and contains a flexible environment. It can be quite overwhelming for anyone starting out in coding with many types of projects any programmer is able to create from the IDE. The IDE is mainly plugin or extension based so there is no telling that it will not encounter any issues (Green, 2021).

**Database**

There are a variety of methods of accessing data from databases that can be implemented for this project. Relational databases store data using tables which consist of rows and columns that reflect the attribute and record respectively. These are used in systems such as Oracle, SQL server and MySQL and are easy to use without prior training and require no hassle in changing the entries (Panwar, 2021). It is also quicker to access data, it enhances security and avoids duplicity. It is costly in maintaining these databases as it needs additional software, it is slower in performance, requires huge memory in storing data, the systems may lose data unintentionally during a transfer and it has limitations in data storage so that data may be lost if the limits have been exceeded (Roomi, 2021).

There is also the object-oriented database whereby it makes use of reusable objects stored in the database. The objects require two elements, data in the form of sound, text, graphic and video and method or instructions that tells what to do with object. The database works well with languages such as Delphi, Ruby, C++, Java and Python. There are many variations of the database like TORNADO, Gemstone, ObjectStore, Gbase and Vbase (Panwar, 2021). The advantages of these databases are that data can be stored easily and their ability to interact with other objects for visual creations. The disadvantages are that they are pricey to utilize and are not widely used (Alzahrani, 2016; Panwar, 2021).

A document database uses documents to store data and is now widely utilised due to its quick data search. Some document databases include Hadoop/Hbase, Cassandra, Hypertable, MapR, Hortonworks and many others. The pros include the flexibility to store huge data and the simple storing of newly added data and additional fields if required. However, it does not work with relational relationships in data as it becomes difficult to handle (Panwar, 2021; Williams, 2021).

Many existing medical facilities and institutions have successfully adopted database applications that assist with the day-to-day management of patients, medical staff and even equipment to better serve the community and improve efficiency in the medical setting.

One such application that has proven to be promising is the tagging of hospital assets through RFID or radio frequency identification. It uses a computer to manage the RFID reader and database. It also uses a mobile application to track the assets and manage them by either adding, removing and updating item information (Álvarez López et al., 2018).

Another useful example of database application in a hospital system is the development of knowledge management systems. This enables support for a doctor’s decision in providing the appropriate care and treatment for a patient who suffers from co-morbidity or diagnosed with a primary disease as focal point of the treatment while adapting to or observing one or more diseases that are treated as secondary (Abidi & Abidi, 2009; Riaño & Ortega, 2017). Knowledge management system tools have been identified as a web repository of medical information, online tool that allows interaction and the creation of content, and database management of medical language (Shahmoradi et al., 2017).

Hospitals may face long queues of patients waiting for their doctor’s appointment or to make payment especially if the facility is the only one that serves a region. An online-based queue management system might be the solution for these patients instead of the conventional method of standing in line or even congregating at a central area to wait for their queue number to be displayed on a board.

This solution may prevent patients from having to miss their queue number or risk a longer wait if they do fail to show up. The application can be accessed wirelessly on their mobile devices and utilises MySQL on PHPMyAdmin to store data, Java for the creation of the application and Glassfish for the webserver. The management of the system itself is done on a Windows-enabled machine by the queue administration staff (Ngorsed & Suesaowaluk, 2016).

All three web applications that have been described use databases to store data and interfaces for user interaction with the system. In terms of the mode of communication used, these applications either use wireless internet/intranet or ethernet to connect to the network. Both the RFID system and queue management system use mobile devices with applications as user interfaces, providing convenience for users.

The use of RFID tags on assets is unique in that they are automatically picked up by the antennas on RFID readers and the system does not require constant effort in managing these items on the asset list. In the case study of the knowledge management system, a patient with co-morbidity may benefit from healthcare that is customized according to an individual’s medical profile (Álvarez López et al., 2018). The online queue system can potentially boost service for patients, provide a low-risk transmission environment during a pandemic, and may also avoid patients from missing any important appointments (Bazant & Bush, 2021; Ngorsed & Suesaowaluk, 2016). The application in this project not only serves to keep track of patient and admission records, but also provide researchers a database to add, edit and update records of their research topic by simply using a “NoSQL” database software.

# 5. IT Methodology

There are 2 types of approaches the project can apply to develop an application. The waterfall method progresses a project by completing an entire phase before starting another. For instance, a team of a web application finishes on gathering information and analysing the requirements before starting on the design of user interfaces and drawing up sequence and activity diagrams and test plans.

This ensures the team to follow the milestones that are clearly charted according to plan and this approach is usually best suited for small and easy projects. It allows all team members to be on the same page and hence, provide everyone in the team with greater understanding of what is going on. However, the possibility of making changes to a project after a phase has been concluded is difficult to carry out which could cause delays and increase the costs (Gaille, 2020).

Agile is an alternative method to developing an application and provides developers with the flexibility to make modifications in a completed phase at any time in the lifecycle. This is especially useful for customer-centric projects and are unclear about the requirements. Developer teams can work their way through the lifecycle as many times as they are able to, also called “iterations” and narrow down towards a specific goal. The advantages are that teams can resolve issues faster, rapid deployment of applications and are effective in reducing costs. The disadvantages include difficulty in tracking tasks, not knowing whether the projects have gone over-budget, projects that could stretch for long periods of time, and risk of scope creep, or being side-tracked by sudden requests for changes or by adding unnecessary features.

The project will adopt the waterfall method which consists of information gathering - the requirement analysis phase, project design – the design phase, implementation and testing – development and testing phase, deployment, and finally, launch – user training phase.

# 6. Project Management

## 6.1 Project Management Narrative

In the requirement analysis phase, there was a need for clarification about the use case requirements and all business use case descriptions were drafted. The use case descriptions were already completed by the end of week 1. In week 2, the use case diagrams, activity diagrams and overall class diagram were drawn up to understand the functional requirements in the application, boundary and roles of the users and completed by the end of the week.

In the design phase, a list of software and version control tool were decided upon based on experience of the project manager. Design-level use case descriptions were completed by middle of week 3 which will provide a list of actions by user, the triggers and alternate actions if any. By the end of the week, sequence diagrams were drawn and completed by the middle of week 4 to provide more clarity on the sequential order of the use case descriptions.

However, there were diagrams and use case descriptions that needed more work on, which delayed the database designing till the middle of week 5. The database design was completed with a data dictionary, logical ERD and standard notation produced by the end of the week 5. The database design assists with the creation of a database that stores the data for application.

At the same time, annotated user interface designs were drawn and had been completed by the start of week 6 which would help with the arrangement and appropriate length of buttons, textboxes, list boxes, combo boxes and labels when creating the application. The test plans were designed and completed by middle of week 6 and were ready for system testing by the time coding commenced in week 7.

Weeks 7 and 8 had focused mainly on the implementation and testing phase. The code for all use cases were written and tested simultaneously using the V model method, or a testing phase that is parallel to the coding phase (Hamilton, 2021).

A deployment diagram was already drawn for the deployment phase in week 6 to ensure that the systems, hardware and software are accurately mapped out and interconnected together.

Finally the user training phase….

## 6.2 Project Plan with Milestones

The overall plan of the project is written in table 1 below and a detailed gantt chart with milestones and tasks is attached to the appendices.

Table 1: Overall project Plan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Week 1 | Week 2 | Week 3 | Week 3, 4, 5 | Week 5, 6 | Week 7, 8 | Week 9 | Week 9, 10 |
| **Information Gathering** | | **Design** | **Design** | **Design** | **Implementation and Testing** | **Deployment** | **User Training** |
| -Clarification of use case requirements  -Completion of business use case descriptions | - Drawing up use case diagrams, activity diagrams, class diagram. | - Writing the design-level use case descriptions. | - Drawing up sequence diagrams  - Designing the database | - Drawing annotated user interface designs  -Test plans | - Coding  -Testing | -Deployment diagram with descriptions | -User Training materials and deliverables |

## 6.3 Project Meetings

*Include a schedule of your meetings (date, duration, participants, and type) and attach the minutes of each meeting to the appendices*

## 6.4 Project Reports

*Discuss the project status reports and attach your project status reports to the appendices*

## 6.5 Project Risk and Issue Analysis

*Discuss project risks and issues and attach your project risk and issue register to the appendices*

# 7. Requirements Analysis

## 7.1 Introduction

## 7.2 Use Case Diagram

## 7.3 Business Use Case Narratives (Descriptions)

## 7.4 Activity Diagrams

## 7.5 Overall Class Diagram

# 8. Project Design

## 8.1 Introduction

## 8.2 Software List

## 8.3 Version Control Software

## 8.4 Design Use Case Narratives (Descriptions)

## 8.5 Sequence Diagrams

## 8.6 Deployment Diagram

## 8.7 Database Design

*Include ERD and data dictionary*

## 8.8 Annotated User Interface Designs

## 8.9 Test Plan

# 9. Project Training

## 9.1 End User Background and Training Objectives

## 9.2 Training Materials

## 9.3 Training Deliverables

## 9.4 Evaluation

# 10. Conclusion & Lessons Learned

# References

Abidi, S. R., & Abidi, S. S. R. (2009). Towards the Merging of Multiple Clinical Protocols and Guidelines via Ontology-Driven Modeling. In C. Combi, Y. Shahar, & A. Abu-Hanna (Eds.), *Artificial Intelligence in Medicine* (pp. 81–85). Springer Berlin Heidelberg.

Álvarez López, Y., Franssen, J., Álvarez Narciandi, G., Pagnozzi, J., González-Pinto Arrillaga, I., & Las-Heras Andrés, F. (2018). RFID technology for management and tracking: E-health applications. *Sensors (Switzerland)*, *18*(8), 1–17. https://doi.org/10.3390/s18082663

Alzahrani, H. (2016). Evolution of Object-Oriented Database Systems. *Global Journal of Computer Science and Technology: C Software and Data Engineering*, *16*(3), 1–5.

Bazant, M. Z., & Bush, J. W. M. (2021). A guideline to limit indoor airborne transmission of COVID-19. *Proceedings of the National Academy of Sciences of the United States of America*, *118*(17). https://doi.org/10.1073/pnas.2018995118

Beattie, A. (2021, August 10). *How the Video Game Industry Is Changing*. https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp

Bolton, D. (2019, July 3). *Programming Languages Comparison*. https://www.thoughtco.com/comparing-popular-programming-languages-958275

Brent, M. (2021, July 1). *Perl Tutorial: Variable, Array, Hashes with Programming Example*. https://www.guru99.com/perl-tutorials.html

Chacon, D. (2018, August 2). *JUnit Tutorial With Examples: Setting Up, Writing, and Running Java Unit Tests*. https://www.parasoft.com/blog/junit-tutorial-setting-up-writing-and-running-java-unit-tests/

Gaille, L. (2020, March 19). *15 Advantages and Disadvantages of a Waterfall Model – Vittana.org*. https://vittana.org/15-advantages-and-disadvantages-of-a-waterfall-model

Gallagher, J. (2020, November 3). *How to Learn C++: A Guide to Learning to Code in C++ | Career Karma*. https://careerkarma.com/blog/learn-c-plus-plus/

Green, Y. (2021, January 24). *21 Best Javascript IDE & Code Editors To Use in 2021 [CSS, HTML, JavaScript]*. https://fulcrum.rocks/blog/best-javascript-ide/

Gupta, A. (2019, November 27). *Advantages and Disadvantages of JavaScript*. https://www.tutorialspoint.com/advantages-and-disadvantages-of-javascript

Hamilton, T. (2021, October 8). *V-Model in Software Testing*. https://www.guru99.com/v-model-software-testing.html

Kidder, T. (2019, August 22). *What is JavaScript Used For? | Uses of JavaScript in Applications*. https://www.learnacademy.org/blog/javascript-used-for/

Krill, P. (2021). *Python edges C++ for Tiobe’s programming language of the year | InfoWorld*. https://www.infoworld.com/article/3602554/python-wins-language-popularity-honor.html

Kumar, C. (2021, November 10). *13 Best IDE Every Programmer Should Know About*. https://geekflare.com/ide-for-programmer/

Ngorsed, M., & Suesaowaluk, P. (2016). Hospital Service Queue Management System with Wireless Approach. In J. C. Hung, N. Y. Yen, & K.-C. Li (Eds.), *Frontier Computing* (pp. 627–637). Springer Singapore.

Panwar, A. (2021, June 9). *Types of Database Management Systems*. https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/

Powers, S. (2012, July 23). *SlickEdit | Linux Journal*. https://www.linuxjournal.com/content/slickedit

Riaño, D., & Ortega, W. (2017). Computer technologies to integrate medical treatments to manage multimorbidity. *Journal of Biomedical Informatics*, *75*(November 2016), 1–13. https://doi.org/10.1016/j.jbi.2017.09.009

Roomi, M. (2021, April 14). *6 Advantages and Disadvantages of Relational Database | Limitations & Benefits of Relational Database*. https://www.hitechwhizz.com/2021/04/6-advantages-and-disadvantages-limitations-benefits-of-relational-database.html

Shahmoradi, L., Safadari, R., & Jimma, W. (2017). Knowledge Management Implementation and the Tools Utilized in Healthcare for Evidence-Based Decision Making: A Systematic Review. *Ethiopian Journal of Health Sciences*, *27*(5), 541–558. https://doi.org/10.4314/ejhs.v27i5.13

Shekhar, S. (2015). *ASP.NET MVC Application - Using JQuery, AJAX*. https://www.c-sharpcorner.com/UploadFile/302f8f/Asp-Net-mvc-using-jquery-ajax/

Sushko, I. (2021, November 18). *How to Create a Hospital Management Software | Aimprosoft*. https://www.aimprosoft.com/blog/how-to-develop-a-hospital-management-system/

Williams, A. (2021, March 18). *NoSQL Document-Oriented Database | NoSQL Document Store | Document Databases | RavenDB NoSQL*. https://ravendb.net/articles/nosql-document-oriented-databases-detailed-overview

# Appendices