THOMPSON RIVERS UNIVERSITY

SENG 1210 - Programming for Engineers II

Information Systems in Health Care

Ruth Befikadu (T00696672)

Ryan Martin (T00549875)

Shaylee Broadfoot (T00551934)

March 31st, 2023

Table of Contents

List of Figures

List of Acronyms

IS - Information System

OOP - Object Oriented Programming

List of Tables

1 Introduction

An information system is a software program used for keeping, accessing and updating any data that is inputted to it. As technology is advancing, software systems are being integrated into many institutions to make it easier for them to function and keep track of all the information flowing within the institution.

Nowadays, research indicates Integrated Healthcare Information Systems are becoming an important part of the healthcare system [1]. The emergence of IS in healthcare has been fueled by the rapid development of information technology including information technology in biomedicine [1]. This can be particularly effective in the emergency department where quick access to some information is crucial for a patient to receive care and live [2].

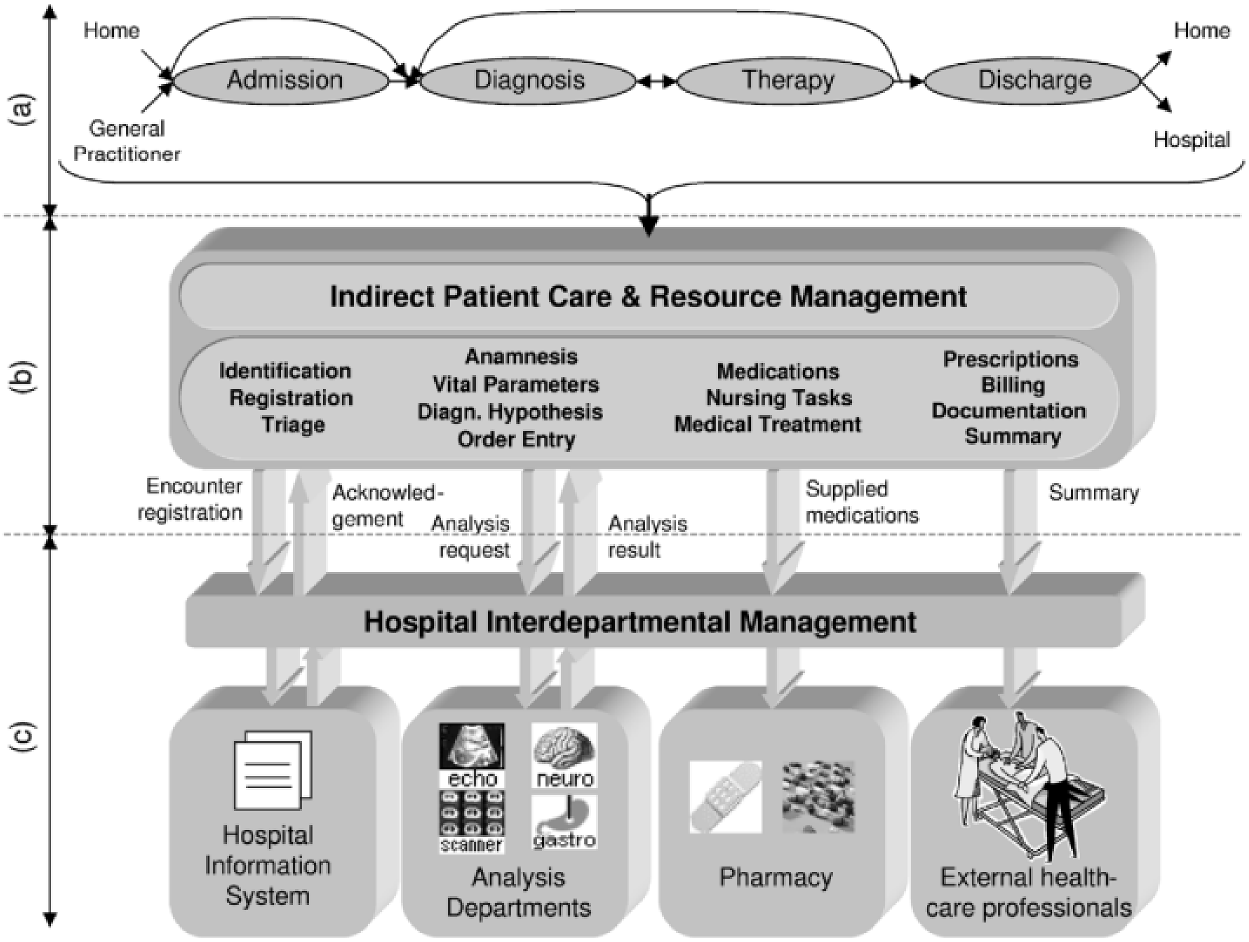

(a)
Home
General Practitioner
Admission
Diagnosis
Therapy
Discharge
Home
Hospital
(b)
Indirect Patient Care & Resource Management
Identification Registration Triage
Anamnesis Vital Parameters Diagn. Hypothesis Order Entry
Medications Nursing Tasks Medical Treatment
Prescriptions Billing Documentation Summary
Encounter registration
Acknowled-gement
Analysis request
Analysis result
Supplied medications
Summary
(c)
Hospital Interdepartmental Management
Hospital Information System
echo
neuro
scanner
gastro
Analysis Departments
Pharmacy
External health-care professionals

Figure 1: Emergency service environment a) patient flow b) system functions c) other entities [2]

This IS will help health care workers to register patients, categorize patients, take in patients information or history, register employees and add employee's information. This kind of system will help keep track of all the information in the specific healthcare center it is being used.

Making this system will help patients register and get service faster. This will save the amount of time nurses, even patients, have to spend inputting information into different sections. The IS design proposed in this report will have many characteristics which make it easy to gather and input information and user friendly. Some characteristics are:

- Clear and aesthetic visuals
- list of functions section
- menu for different service
- a section for the brief description of the software

Following this, the report will discuss the detailed problem statement and the criterias needed to solve this. After the formulation of different solutions, the best will be selected according to a decision matrix and be designed and coded. Finally a full conclusion on all the parts of the software will be discussed.

2 Design Problem

The following section includes defining the problem and discuss the functions, objectives and constraints of the program

2.1 Problem Definition

Table 1: Problem definition.

Drafted by:	Ryan M., Ruth B., Shaylee B.
Program description	C++ program that will be used in the healthcare system.
Program concept	Use of object oriented programming to make an easy, effective and user friendly IS.
Benefits	- Speed in information access - Categorization of a person's personal information - Separation of information into patient and employees

Target market	Health care providing institutions
Stakeholders	Design team, professors, health care companies
Attribute and possible features	Main title, Employee and patient Menu, Information menus
Possible area of innovation	Make it an app, website or a downloadable software.
Problem statement	This project uses the concept of OOP and the logical skill of programming to make a program for solving the problem of organization and access to big data.

2.2 Design Requirements

2.2.1 Functions

- Register and categorize patients
- Register and manage healthcare employees and their information
- Provide quick access to patients data in the emergency department

2.2.2 Objectives

- User-friendly interface for easy data input and retrieval
- Fast and efficient data processing to reduce waiting times
- Integration with both Patients and Staff systems for seamless data exchange
- Scalability to accommodate future growth

2.2.3 Constraints

- Compliance with data privacy regulations and standards. - Constraint not met
- Integration with existing healthcare infrastructure and systems. - Constraint met
- Compatibility with different operating systems and devices. - Constraint not met
- Cost-effective development and maintenance of the system. - Constraint met
- High availability and reliability of the system to minimize downtime. - Constraint met

3 Solution

3.1 Solution 1

- Two classes. Patients and Employees
- One main
- Main calls functions described in Class.h files
- Each function pulls data from the database

3.2 Solution 2

- Two classes. Patients and Employees
- One main
- Main calls functions described in Class.h files
- Class.h files read from database at runtime
- Database stored in matrix that can be manipulated until user wants to terminate
- Matrix saved to database
- string manipulation

3.3 Final Selected Solution

After looking at the two solutions above, more improvement needed to be made to make the final solution. This solution is an improved version of the above 2.

- Patient and Employee database
- One main
- Login password for security
- Calls excel database only once
- header and implementation files for the class
- Vector and string manipulation
- Class to import the file content

Table 2 Decision matrix chart for the considered alternatives

		Solutions					
		Solution 1		Solution 2		Final Solution	
Criteria	Weight	Score	Partial Score	Score	Partial Score	Score	Partial Score
Criterion 1	0.40	6/10	0.240	6/10	0.240	7/10	0.280
Criterion 2	0.25	3/5	0.150	4/5	0.200	5/5	0.250
Criterion 3	0.20	9/15	0.120	12/15	0.160	9/15	0.120
Criterion 4	0.15	9/10	0.135	6/10	0.090	8/10	0.120
Sum	1.00		0.645		0.690		0.77

The decision matrix produces Solution 2. This is the correct choice as users will alter the database an unpredictable amount of times in a session.

3.3.1 Components

Some of the components in the final solution’s code and used to make this code are:

- Windows
- Compilers
- Excel files

- Libraries (cmath, vector, algorithms, string, iostream, iomanip)
- Login statement
- Statement to introduce the system and the menus
- Space for user to enter choice
- Output of results
- Option for return
- Option for multiple operations

3.3.2 Features

The features in the program include:

- Title introducing what the program does and for what purpose it is used
- Patient and Employee selections
- Variety of menu options
- 6 menu options for both. Menu for employee include:
    - Create new employee
    - Remove Employee
    - Employee Profile
    - Display all employee profiles
    - Display employee names
    - Update existing employee

Menu for patient include:

- Display patient names
- Display patient information
- Edit patient information
- Add new patient
- Delete patient

- User friendly display
- Straight forward instructions and directions
- Good class manipulation
- Use of functions
-

3.3.3 Design Considerations

For a design to be sustainable, it must be able to be used many times without negative impact to the environment. Our program surely contributes to that.

Sustainability and environmentally friendly

- Use of class
- Use of functions
- Eliminating multiple file reading
- Having a comment

These ways make the code short and easy to read for future improvement. The fact that the code is condensed means that it will consume less energy in terms of processing power, memory power and even raw electricity. Having some comments in the code also will help the code be understood by a new person working on it. In addition, it will also help the person who wrote it to recall it when viewing it after some time.

Ethical aspect and security

The program is made to be only used by IT employees. This project doesn't include programming for security with java specifically. It is ethically sound because employers always have the information of the employee. As long as the company uses it wisely and properly, all will be ethical and privacy will be kept. Hacking is unlikely since C++ is a strong program so it is very secure.

Aesthetics and ease of navigation

The IS is aesthetically sound and easy to navigate through. It is user friendly in guiding the user each step of the way. The code allows for users to do multiple commands after opening it. It doesn't require the program to be run again and again to use different menu options.

Economic consideration

This program will help hospitals and other health care providing institutions in saving money in the Microsoft 365 package. This program will help them economically and is a more effective option compared to manually entering information in each cell of an excel file. Economic decisions were made in the pretext of saving energy since this system will be up and running possibly, 24/7.

3.3.3.1 Environmental Considerations

Table 3: Environmental considerations

Environmental Consideration	Regulatory Aspect	Description
Physical Security	HIPAA	The system must have physical safeguards in place to protect the hardware, software, and data from unauthorized access, theft, or damage.
Data Security	HIPAA/GDPR	The system must have technical safeguards to ensure the confidentiality, integrity,

		and availability of patient and employee data.
User Access Control	HIPAA	The system must have controls in place to ensure that only authorized personnel can access sensitive data. This can include password policies, multi-factor authentication, and role-based access control.

3.3.3.2 Societal Considerations

Table 4: Societal considerations

Societal Consideration	Description
Cultural Diversity	The system must be designed to accommodate cultural differences among patients and employees. This can include providing language support, respecting cultural norms and practices, and ensuring that the system does not perpetuate bias or discrimination.
Ethical Considerations	The system must adhere to ethical principles such as respect for patient autonomy, beneficence, non-maleficence, and justice. This can include ensuring that patients have informed consent before their data is collected or used, protecting patient confidentiality, and avoiding conflicts of interest.
Security	The system must ensure the security of patient and employee data, and protect against cyber-attacks or breaches.

Accessibility	The system must be accessible to all employees, including those with disabilities.

3.3.3.3 Health and Safety Considerations

Table 5: Health and safety considerations

Health and Safety Consideration	Regulatory Aspect	Description
Data Privacy	HIPPA/GDPR	The system must ensure the privacy of patient and employee data. This can include measures such as data encryption, access controls, and data anonymization. The system must comply with regulations such as HIPAA and GDPR regarding the collection, storage, and sharing of personal information.
Infection Control	OSHA/CDC	The system should support infection control measures by minimizing the need for physical contact with patients and reducing the risk of transmission of infectious diseases.
Employee Confidentiality	OSHA	The system should prioritize employee confidentiality by ensuring that it is designed and implemented in a manner that does not compromise employee confidentiality.

3.3.3.4 Economic Considerations

Since this specific program is made for health care institutions, the program is made with the aim of making it extremely affordable. There are some economic considerations that were taken which include:

- Making the code user friendly regardless of user
- Make the code compact to avoid andy wastage of energy which is economic
- Easy to make
- Existence of well defined comments make it easy to update and manipulate

3.3.4 Limitations

Some limitations that we see with the program are:

- Since different computer systems have different libraries, this code will not work with old computers
- The code's limitation is the computer's storage limitation. The number of data the code can handle is limited by the computer
- The data is accessible only by the company usable, so sharing the same data across many companies' platforms is not included in this code.
- Can't do collaborative data manipulation since this code is integrated with excel.
- Require people to manually enter peoples' names and informations (not automated)

4 Team Work

This section includes all meetings where we were all present and completed an activity together. This includes coding and/or writing a report.

4.1 Meeting 1

Time: February 25, 2023, 2:00 pm to 6:00 pm

Agenda: Start on the basic menu window and class structures

Team Member	Previous Task	Completion State	Next Task
Ryan Martin	N/A	N/A	Basic Window
Ruth Befikadu	N/A	N/A	Employee Class
Shaylee Broadfoot	N/A	N/A	Patient Class

4.2 Meeting 2

Time: March 4, 2020, 2:00 pm to 5:00 pm

Agenda: Detain sub-menu and window development

Team Member	Previous Task	Completion State	Next Task

Ryan Martin	Basic Window	60%	Main Menu
Ruth Befikadu	Employee Class	70%	Employee Menu
Shaylee Broadfoot	Patient Class	76%	Patient Menu

4.3 Meeting 3

Time: March 7, 2020, 2:00 pm to 5:00 pm

Agenda: Input and output options of the code

Team Member	Previous Task	Completion State	Next Task
Ryan Martin	Main Menu	75%	Employee-Patient Integration
Ruth Befikadu	Employee Menu	80%	Employee Information I/O
Shaylee Broadfoot	Patient Menu	70%	Patient Information I/O

4.4 Meeting 4

Time: March 28, 2023, 2:00 pm to 5:00 pm

Agenda: Itegrations

Team Member	Previous Task	Completion State	Next Task
Ryan Martin	Employee-Patient Integration	50%	Report Completion
Ruth Befikadu	Employee Information I/O	60%	Report Completion
Shaylee Broadfoot	Patient Information I/O	60%	Report Completion

4.5 Meeting 5

Time: March 31, 2023, 3:00 pm to 6:00 pm

Agenda: report writing

Team Member	Previous Task	Completion State	Next Task
Ryan Martin	Report Completion	90%	Testing
Ruth Befikadu	Report Completion	90%	Testing
Shaylee Broadfoot	Report Completion	90%	Testing

4.6 Meeting 6

Time: April 1, 2023, 6:00 pm to 9:00 pm

Agenda: software testing

Team Member	Previous Task	Completion State	Next Task
Ryan Martin	Testing	100%	-
Ruth Befikadu	Testing	100%	-
Shaylee Broadfoot	Testing	100%	-

5 Project Management

This section details the management of the team's tasks and responsibilities over the nine week period of the project. The project was broken into four phases:

1. Project Conception and Initiation,
2. Project Definition and Planning,
3. Project Launch and Execution,
4. Project Report and Testing.

Phase 1 included choosing a topic for an information system and then researching the typical uses and required data to implement it.

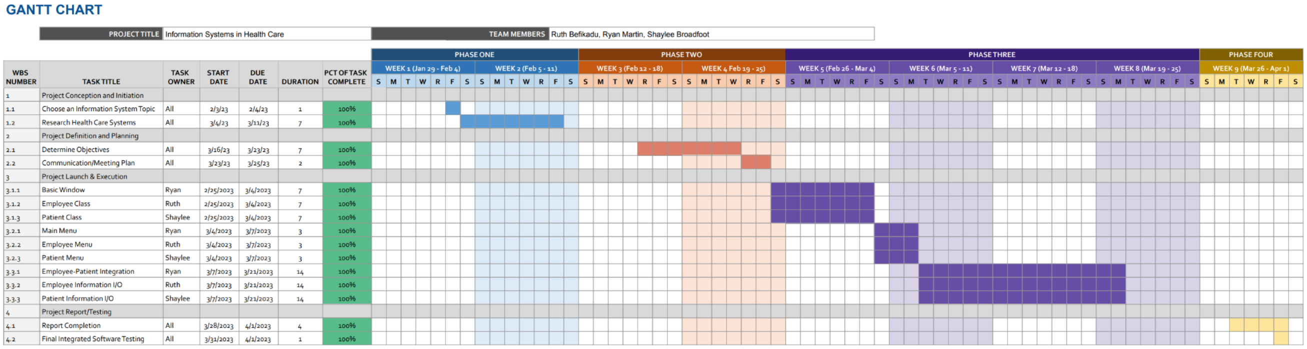
Phase 2 consisted of discussions regarding the team's goals and objectives for the information system, and how to communicate and schedule the predicted tasks to complete the project on time.

Phase 3 was the longest as this is when project execution took place. Each member had three main tasks throughout this period which involved all of the coding and bulk of the debugging for the system.

Phase 4 included four days to complete the report for the system, and a few hours of final testing in the lab period to smooth out any parts of the program if necessary.

The detailed schedule and progress of these phases and their tasks is outline in the Gantt chart below in *figure 2*. There are two periods seen where no tasks are assigned intentionally planned to provide slack time around midterms.

Additionally, table 6 below aims to more clearly outline the tasks found in the Gantt chart. It also indicates the task number of any predecessors for each task, the slack time in days at any points in the schedule, and indicates the critical path with the use of a check mark.



GANTT CHART

PROJECT TITLE Information Systems in Health Care

TEAM MEMBERS Ruth Befikadu, Ryan Martin, Shaylee Broadfoot

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	Project Conception and Initiation					
1.1	Choose an Information System Topic	All	2/3/23	2/4/23	1	100%
1.2	Research Health Care Systems	All	3/4/23	3/11/23	7	100%
2	Project Definition and Planning					
2.1	Determine Objectives	All	3/16/23	3/23/23	7	100%
2.2	Communication/Meeting Plan	All	3/23/23	3/25/23	2	100%
3	Project Launch & Execution					
3.1.1	Basic Window	Ryan	2/25/2023	3/4/2023	7	100%
3.1.2	Employee Class	Ruth	2/25/2023	3/4/2023	7	100%
3.1.3	Patient Class	Shaylee	2/25/2023	3/4/2023	7	100%
3.2.1	Main Menu	Ryan	3/4/2023	3/7/2023	3	100%
3.2.2	Employee Menu	Ruth	3/4/2023	3/7/2023	3	100%
3.2.3	Patient Menu	Shaylee	3/4/2023	3/7/2023	3	100%
3.3.1	Employee-Patient Integration	Ryan	3/7/2023	3/21/2023	14	100%
3.3.2	Employee Information I/O	Ruth	3/7/2023	3/21/2023	14	100%
3.3.3	Patient Information I/O	Shaylee	3/7/2023	3/21/2023	14	100%
4	Project Report/Testing					
4.1	Report Completion	All	3/28/2023	4/1/2023	4	100%
4.2	Final Integrated Software Testing	All	3/31/2023	4/1/2023	1	100%

Figure 2: A Gantt Chart outlining the tasks and responsibilities of the project, with a start date of February 3rd and an end date of March 31st.

Table 6: Gantt chart tasks and their predecessors, any slack time, and indication of the critical path.

	Tasks	Predecessors	Slack Time	Critical Path
1.1	Choose an Information System Topic		0	✓
1.2	Research Health Care Systems	1.1	4	✓
2.1	Determine Objectives	1.2	0	✓
2.2	Communication/Meeting Plan		0	
3.1.*1*	Basic Window	2.1	0	✓
3.1.*2*	Employee Class	2.1	0	✓
3.1.*3*	Patient Class	2.1	0	✓
3.2.*1*	Main Menu	3.1.1	0	✓
3.2.*2*	Employee Menu	3.1.2	0	✓
3.2.*3*	Patient Menu	3.1.3	0	✓
3.3.*1*	Employee-Patient Integration	3.2.1, 3.2.2, 3.2.3	7	✓
3.3.*2*	Employee Information I/O	3.2.2	7	✓
3.3.*3*	Patient Information I/O	3.2.3	7	✓
4.1	Report Completion	3.3.1, 3.3.2, 3.3.3	0	✓
4.2	Final Integrated Software Testing	3.3.1, 3.3.2, 3.3.3	0	

6 Life-Long Learning

This project demonstrated the importance of teamwork and communication while collaborating on a coding project. The work was divided so that members were working on different classes at the same time, and others may not directly see that code for a period of time. Therefore, to prepare for later integration, it was vital that the team held frequent discussions and updates about details like what menu options would look like, plans for accessing data, or required functions. Additionally, writing organized code with descriptive class, function, and variable names, and appropriate comments aids in integrating these codes, as it is easier for each person to understand and follow.

Similarly, the discrepancies between different compilers was noticed; and thus, ensuring the whole team is using the same compiler writing and debugging their code. In this situation, two members used Visual Studio on windows PCs, while one used a Xcode on a Mac, and while the would would run smoothly on one, once the sharing and integration of codes began, numerous errors appeared that were not an issue previously.

In previous course material, the only examples regarding reading and writing data were to text files. While this would work, it was decided that a spreadsheet program would be better suited to organize and store the patient and employee data, and did some research on using excel. It was found that exporting to different file types is quite straightforward as the suffix of the file name just had to be changed from ".txt" to ".csv" to input or output. Initially, it was expected that reading and writing to specific cells would also be straightforward; however, the code was found to be quite lengthy and more complicated than desired for this project.

Since individual cells were not being referenced in this program, data was read and written line by line as comma separated values. Upon testing, it was found that very small errors and inconsistencies in code could result in a large mess in the database; thus, the values of good data management were learned here. For instance, when an emergency situation is considered, one wrong input or output in data regarding allergies or medications could cost someone their life.

Finally, the initial plan for the project was to implement a GUI for the information system and had actually created a window in the beginning. At this stage, the menu options and full functioning of the system was completely planned out, which made it difficult to create any further code for this. The code for a simple window and button was very lengthy and again more complex than projected for this project, so the concept was never revisited; thus, the importance of a thorough design plan before starting a project was highlighted, specifically as the programs designed become larger and more complex.

7 Conclusion and Future Work

The goal of this project was to design an information system tailored to the healthcare field, specifically, the emergency department. It was planned to perform three main functions:

- register and categorize patients,
- register and manage healthcare employees and their information and,
- provide quick access to patients data in emergency situations.

The implemented program provides options to register, edit and delete patients and employees along with their personal information and certainly provides quick access to this data as opposed to manually searching for it in a computer file, or searching physical records. However, categorization of patients was not implemented due to time constraints, and lower prioritization.

Additionally, four objectives were set for this project – all of which were achieved adequately. This includes user-friendliness; the system accepts simple one step inputs and clear menu options for all functions. Next was efficient data processing, and while there are certainly ways to improve the efficiency of the code, the time to access data is instantaneous from the user's perspective and is acceptable for this setting. Thirdly, integration between staff and patient systems was completed, with both being accessible from the main menu, and further functions being available within each. And finally, the program was to be designed to consider scalability to accommodate future growth. As the system is very modifiable with no limit to how many patients and staff can be added or removed, it is expected that future growth of this information system is easily viable.

This result of this design only satisfied two of the constraints set out which included cost-effective development and maintenance of the system and ensuring high availability and reliability of the system to minimize downtime. Since the current program runs on Visual Studio and can be used with any CSV file, there are no costs associated with the software used, and with no need to shut down the system for updates using these platforms, there should be little to no downtime involved. It should be noted that the remaining constraints were set for the information system to be implemented professionally; thus, the reason they were not met at this stage in the design.

Overall, the design product was considered to be successful by the requirements laid out previously with lots of room for expansion. As mentioned previously, the design of this system exhibits scalability options regarding the database, and for extra efficiency and ease of use, there are some future improvements to be explored.

Future improvements include:

- Make the system autonomous so that people scan their employee or patient cards for them to update or enter the database
- Make this code into a full fledged software that can be integrated to the internet and shared across all the company's platforms for statistical purposes.
- Usage of another file type besides excel for easy collaboration will also make the system better
- Linking this to cloud storage will help hospitals avoid spending money on physical storage devices and store all their patient's data on the cloud.
- Making a different version of the same code with different language (Previous version c++) will make it easy for hospitals with outdated computers to benefit from this information system

8 References

[1] L. Li, R. -L. Ge, S. -M. Zhou and R. Valerdi, "Guest EditorialIntegrated Healthcare Information Systems," in IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 4, pp. 515-517, July 2012, doi: 10.1109/TITB.2012.2198317.

[2] T. Amouh et al., "Versatile clinical information system design for emergency departments," in IEEE Transactions on Information Technology in Biomedicine, vol. 9, no. 2, pp. 174-183, June 2005, doi: 10.1109/TITB.2005.847159.