

VNU HCM - University of Science
Faculty of Information Technology

Subject: Java Programming

Report #2

Java IO/Database

NOVEMBER 27

Class: 19CLC-KTPM2

Authored by: Group 4 - Cookies

Phạm Trọng Vinh Khuê - 19127038

Trần Đại Hoàng Trung - 191270081

Trần Thanh Tùng - 19127311

Table of Content

I - INDIVIDUAL CONTRIBUTIONS BREAKDOWN	3
II - INTRODUCTION	3
III - ANALYSIS AND DESIGN	4
IV - IMPLEMENTATION	5
V - SAMPLE DATA	7
VI - RESULT	9
VII - PLAN	9
1. Task decomposition	9
2. Project management	9
3. Risk management	14
4. Workspace management	14
VIII - REFERENCES	15

I. Individual Contributions Breakdown

All team members contributed equally.

II. Introduction

This report is created in order to show the way my team stores and loads application's data, how we design the data structure after analyzing application's requirements and show sample data structure that can be used to test the application later. This report also provides the result my team has achieved, the contribution of each member and the detailed plan for the rest of the project.

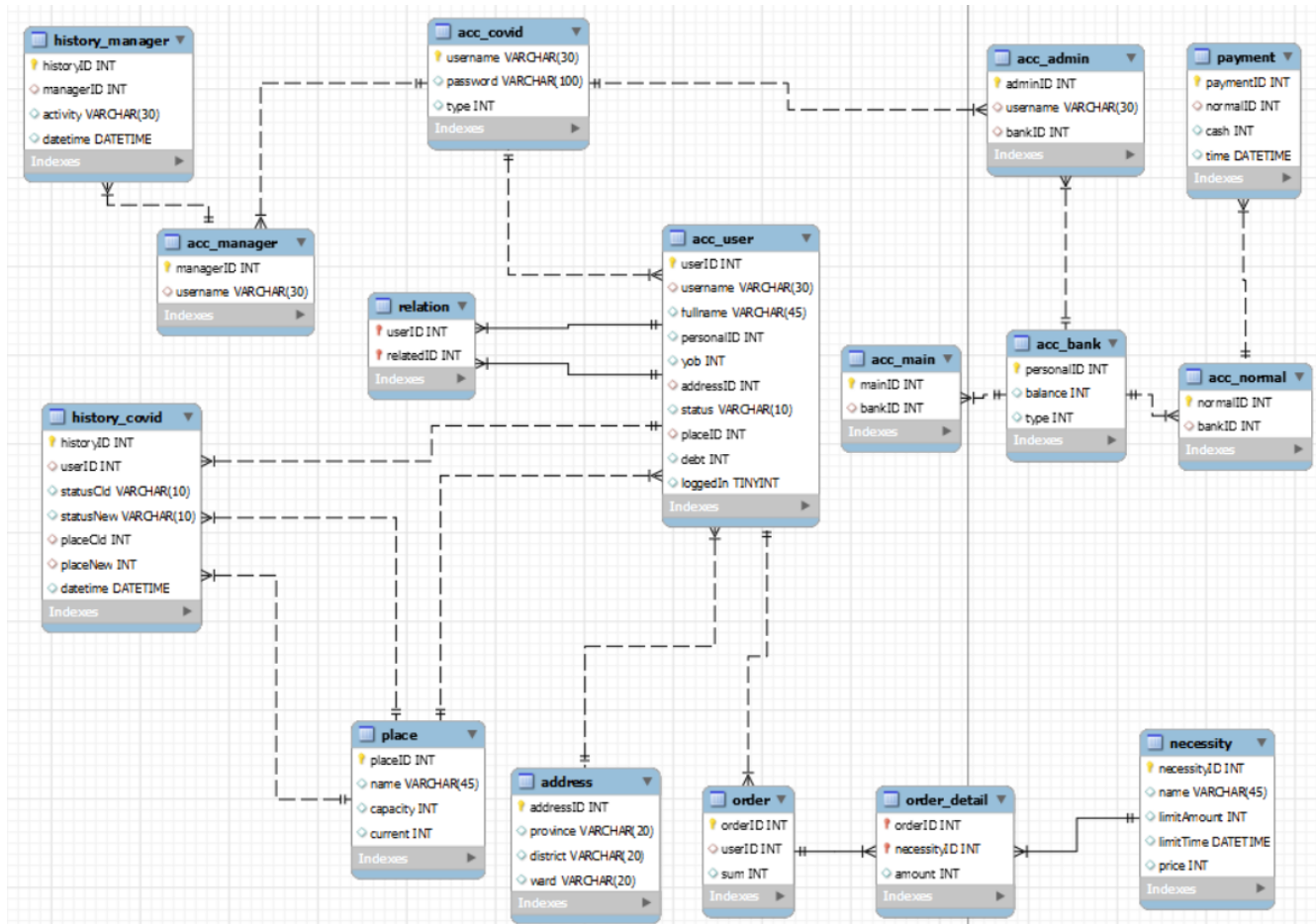
We store application's data in a database management system. With this method, data is stored and used efficiently while ensuring security. Database management system connect to Java through JDBC (Java Database Connectivity). JDBC library includes APIs for the following tasks:

- Connect to databases
- Create SQL or MySQL statements
- Execute SQL or MySQL queries in the database.
- View and modify the resulting records.

These tasks allows us to import data from database management system and export data to database management system. Moreover, my team has the basics of database management thanks to the Database course. So, storing data using a database management system is our choice for this project. VNEID can be used to declare the locations that users will go in the near future.

III. Analysis and Design

For this project, we'll be using MySQL to store data and information. Through meetings, we've come to a conclusion of how our database should be designed, and how the components should interact with each other. All of these are visibly explained with the diagram below.



- **acc_covid**: This table stores basic attributes of an account of COVID System such as type, username, password, checkLogin.
- **acc_user**: This table stores personal information of patients.
- **relation**: This table indicates which pair of patients have related to one another.
- **acc_admin**: This table stores additional information of administrators.
- **acc_manager**: This table is used to help view the history of managers.
- **history_manager**: This table stores history of actions by managers.
- **acc_bank**: This table contains information about payment accounts for users in general.
- **acc_normal**: This table contains accounts which deal with payment-related work.

-
- **acc_main:** This table represents the main account of Payment System to receive payment from normal account.
 - **payment:** This table stores information of payment procedures made by normal accounts.
 - **address:** This table stores the address of users.
 - **place:** This table contains information about the places where patients are resided.
 - **history_covid:** This table keeps track of the patients' status, as well as place changes.
 - **necessity:** This table contains information of necessity packages.
 - **order:** This table contains specific orders made by users.
 - **order_detail:** This table stores detailed information of necessity packages in an order made by the users.

IV. Implementation

There are a few steps to make a connection to the database using Java:

- Setup JDBC environment
 - Install Database: MySQL
 - Install Database Drivers: Add mysql-connector-java-8.0.27.jar to project
 - Set Database Credential: Username and password
- Implement the JDBC programming procedure
 - Load the database driver: `Class.forName("com.mysql.jdbc.Driver");`
 - Obtain a connection: `DriverManager.getConnection(String url);` or `DriverManager.getConnection(String url, Properties prop)` or `DriverManager.getConnection(String url, String user, String password);`
 - Create and execute statements (SQL queries): `Statement` or `PreparedStatement` and its methods
 - Use result sets (tables) to navigate through the results: `ResultSet` and its methods
 - Close the connection: `close();`

Example:

```
public class ExampleDatabase {
    // Database URL
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        try {
            // STEP 2: Register JDBC driver
            Class.forName(JDBC_DRIVER);
            // STEP 3: Open a connection
            System.out.println("Connecting to database...");
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // finally block used to close resources
            try {
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException se) {
                se.printStackTrace();
            } // end finally try
        } // end try
        System.out.println("Goodbye!");
    } // end main
}
```

run:

Connecting to database...

Goodbye!

BUILD SUCCESSFUL (total time: 0 seconds)

V. Sample Data

acc_bank

	personalID	balance	type
▶	261547286	1000000	2
	261547311	1500000	2
	261548596	500000	2
	261548924	2000000	2

acc_covid

	username	password	type
▶	261547286	NULL	3
	261547311	NULL	3
	261548596	NULL	3
	261548924	NULL	3
	ptvkhue	fd3283d2b3b14e5f0944bac9c6165c796bee86a...	2
	tdhtrung	5db9e5f14a52696e69a7b3577c475dbac623255...	2
	tttung	f4dd2271ece91a96fca6c3d835af614a5f8cf631f...	2

acc_manager

	managerID	username
▶	1	ptvkhue
	2	tdhtrung
	3	tttung

acc_normal

	normalID	bankID
▶	3	261547286
	4	261547311
	2	261548596
	1	261548924

acc_user

	userID	username	fullname	personalID	yob	addressID	status	placeID	debt	loggedIn
▶	1	261548924	Phạm Trọng Vinh Khuê	261548924	2001	1	F4	2	0	0
	2	261548596	Phạm Hoàng Anh	261548596	1996	3	F2	1	0	0
	3	261547286	Trần Đại Hoàng Trung	261547286	2001	2	F3	2	0	0
	4	261547311	Trần Thanh Tùng	261547311	2001	8	F1	4	0	0

address

	addressID	province	district	ward
▶	1	Bình Thuận	Phan Thiết	Xuân An
	2	Bình Thuận	Phan Thiết	Phú Trinh
	3	Bình Thuận	Tuy Phong	Bình Thạnh
	4	Bình Thuận	Tuy Phong	Phan Rí Cửa
	5	Hồ Chí Minh	Quận 8	Phường 4
	6	Hồ Chí Minh	Quận 7	Phường 5
	7	Hồ Chí Minh	Bình Chánh	Bình Hưng
	8	Hồ Chí Minh	Quận 1	Phường 2

relation

	userID	relatedID
▶	3	1
	4	1
	1	3
	1	4

place

	placeID	name	capacity	current
▶	1	Bệnh viện đa khoa tỉnh Bình Thuận	1000	500
	2	Trường THPT chuyên Trần Hưng Đạo	100	30
	3	Bệnh viện 7A	1500	700
	4	Nhà thi đấu Phú Thọ	300	75
	5	Kí túc xá ĐHQG TP. Hồ Chí Minh	5000	3000

necessity

	necessityID	name	limitAmount	limitTime	price
▶	1	Gạo (1kg)	50	NULL	10000
	2	Muối (1kg)	10	2022-06-24 00:00:00	10000
	3	Dầu ăn (1l)	20	NULL	50000
	4	Nước rửa tay (1 chai)	50	NULL	30000

order

	orderID	userID	sum
▶	1	1	90000
	2	2	50000

order_detail

	orderID	necessityID	amount
▶	1	1	1
	1	3	1
	1	4	1
	2	1	1
	2	2	1
	2	4	1

payment

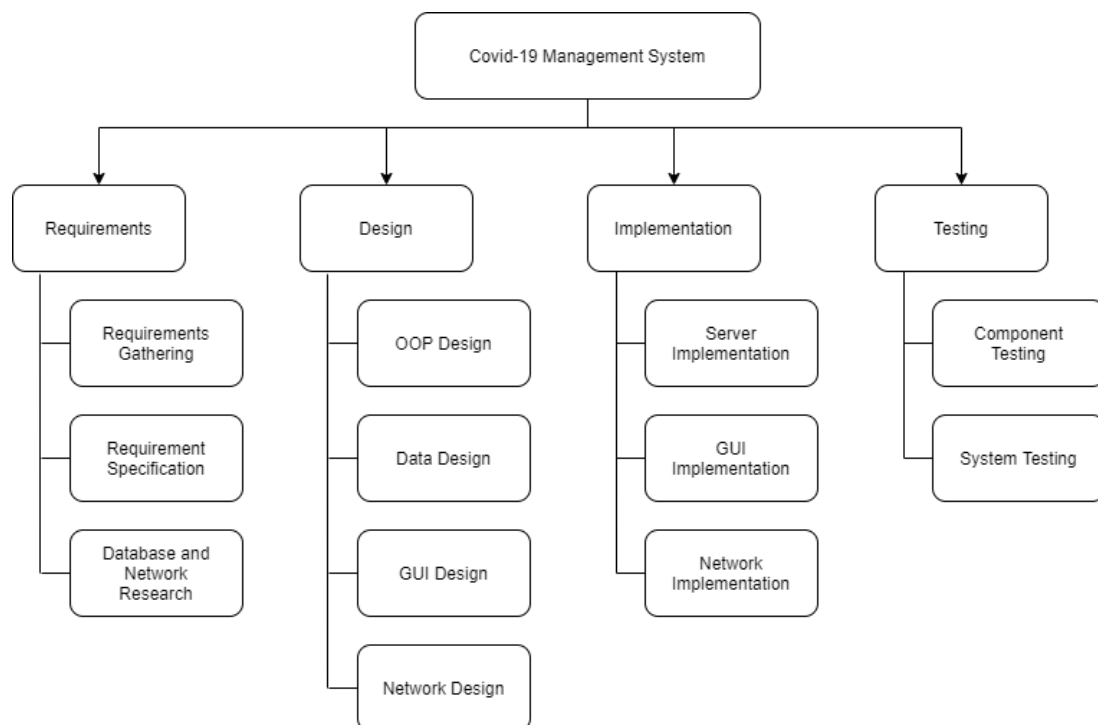
	paymentID	normalID	cash	time
▶	1	1	90000	2021-11-26 00:00:00
	2	2	50000	2021-11-25 00:00:00

VI. Result

During the two weeks of programming code & writing this report, we have analyzed the project requirements in detail and come up with the suitable data structure for our application. We also select a database management system called MySQL. With this system, data is secure and conveniently accessible through queries. Thanks to the database diagram and MySQL, we have solved many problems about storing, retrieving data and completed some parts of the project. Although we had to spend time for the deadlines of other courses, all members tried their best to to complete this report. With our project properly planned out, we will develop the application in the best way.

VII. Plan

1. Task Decomposition



2. Project Management

a) Project Estimates

The estimated time for the whole project is 8 weeks, following the detailed plan in part 2.2 below. This estimate is based on the fortnightly tasks for each phase of the project. As this is a voluntary program, the user satisfaction would be the compensation for our development team. Re-estimation would only occur in the event of feedback received once the program published.

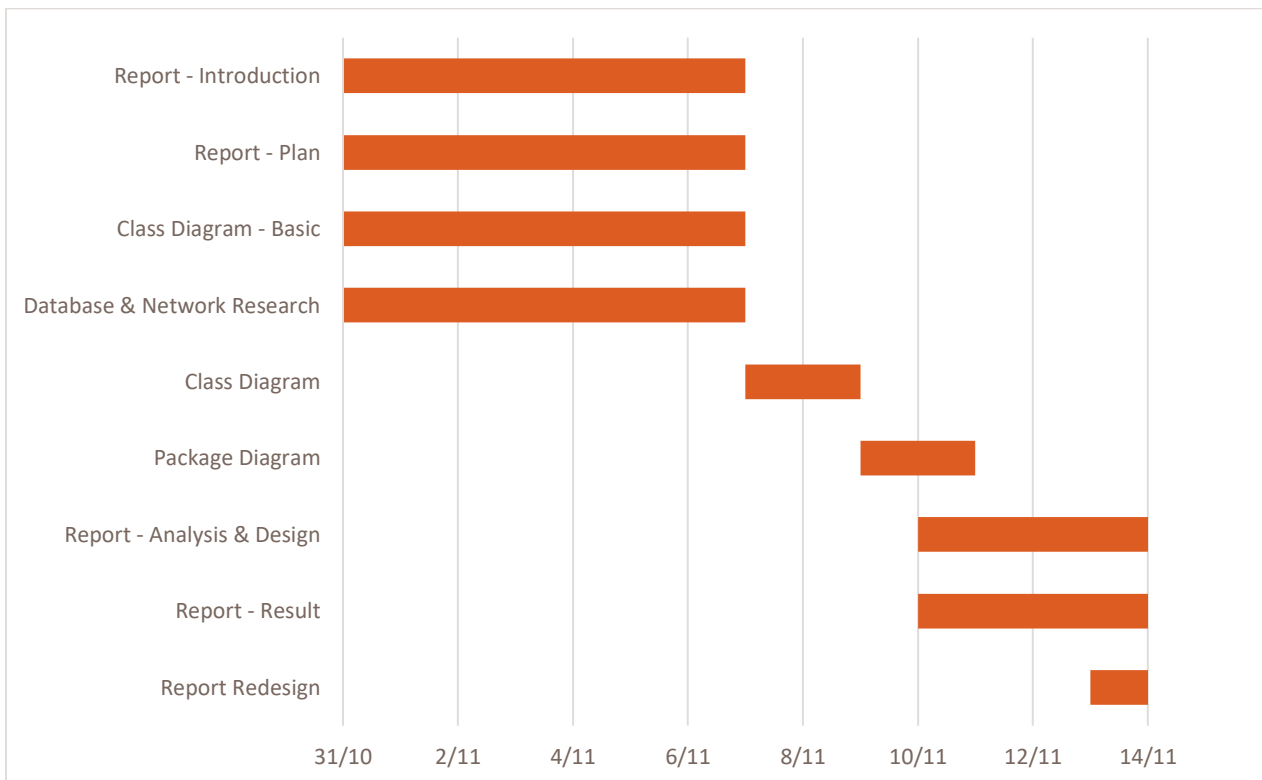
b) Project Plan

❖ *Sprint 1:*

- Duration: 14 days (31/10/2021 - 14/11/2021).
- Outcome: Class diagram, Package diagram, Report #1.
- Task information:

Task	Member	Task type	Duration	Start date	End date
Report - Introduction	Tùng	Document	7	31/10	07/11
Report - Plan	Khuê	Document	7	31/10	07/11
Class diagram - Basic	Trung	Document	7	31/10	07/11
Database and Network Research	All	Training	7	31/10	07/11
Class diagram	All	Document	2	07/11	09/11
Package diagram	All	Document	2	09/11	11/11
Report - Analysis and Design	Khuê, Trung	Document	4	10/11	14/11
Report - Implementation & Result	Tùng	Document	4	10/11	14/11
Report Redesign	Tùng	Document	1	13/11	14/11

- Task information:

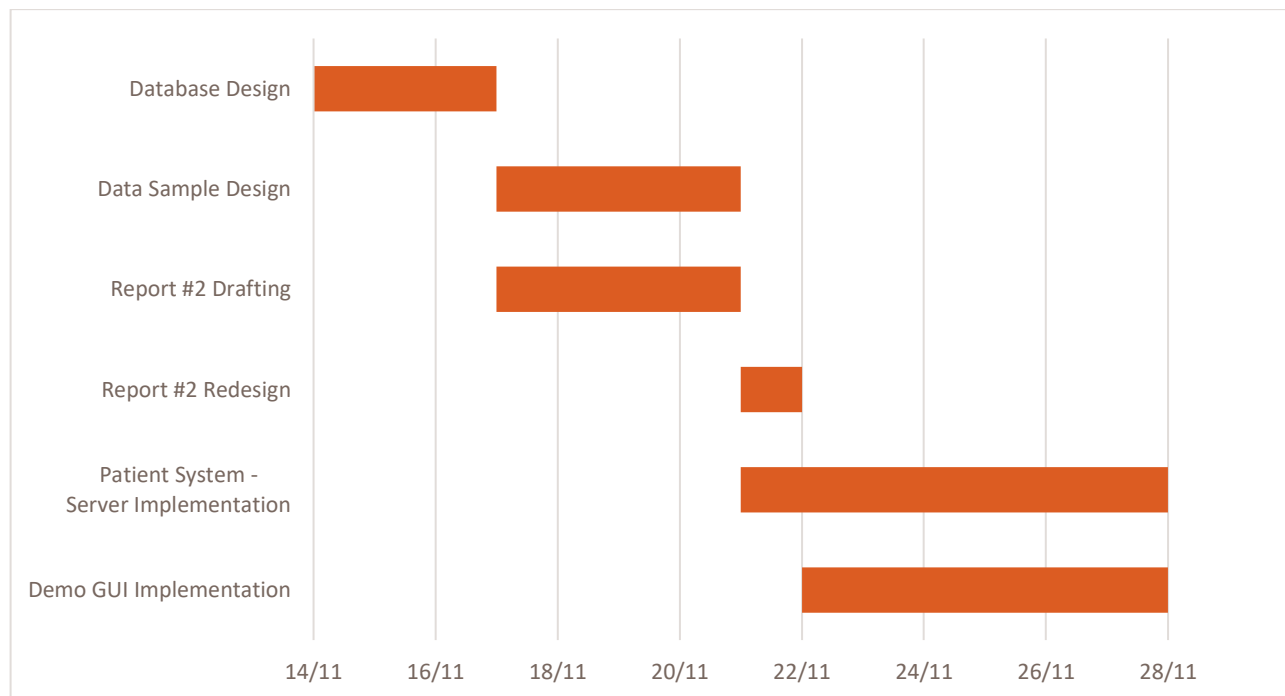


❖ *Sprint 2:*

- Duration: 14 days (15/11/2021 - 28/11/2021).
- Outcome: Database Structure, Sample Data, Report #2, Demo GUI.
- Task information:

Task	Member	Task type	Duration	Start date	End date
Database Design	All	Database	3	14/11	17/11
Data Sample Generation	Khuê Tùng	Database	4	17/11	21/11
Report #2 Drafting	Trung	Document	4	17/11	21/11
Report #2 Redesign	Tùng	Document	1	21/11	22/11
Patient System - Server Implementation	Khuê Trung	Server	7	21/11	28/11
Demo GUI Implementation	Tùng	GUI	6	22/11	28/11

- Task information:

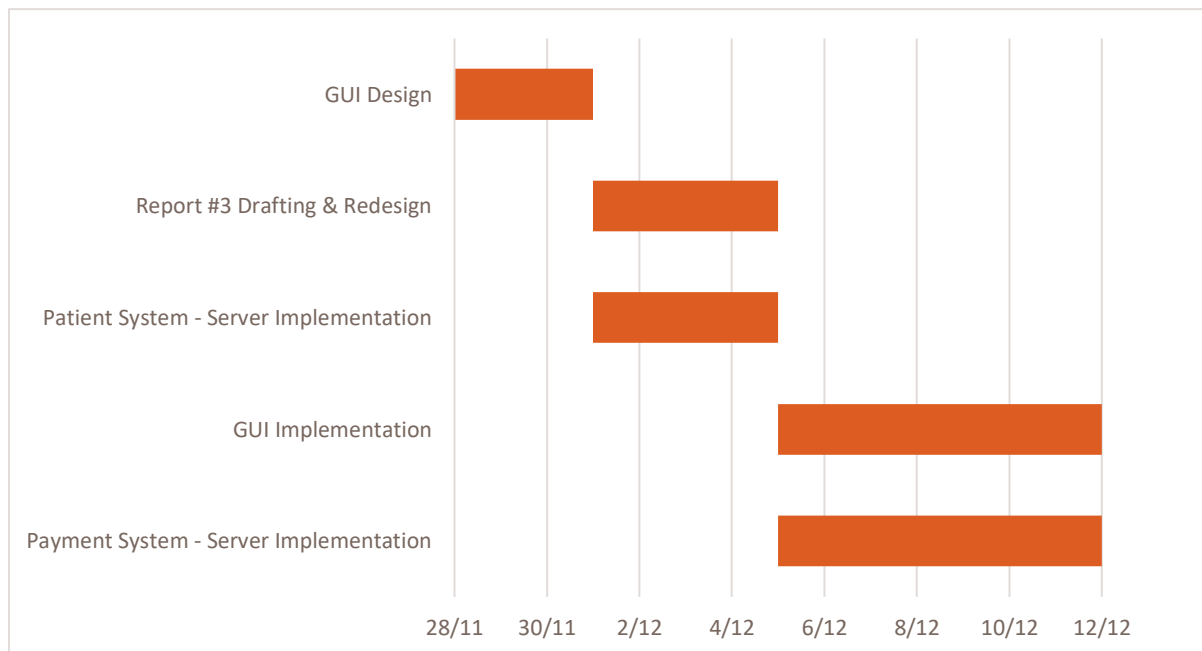


❖ *Sprint 3:*

- Duration: 14 days (28/11/2021 - 12/12/2021).
- Outcome: GUI Structure, Report #3, Patient System.
- Task information:

Task	Member	Task type	Duration	Start date	End date
GUI design	Khuê Trung Tùng	Document	3	28/11	01/12
Report#3 drafting and redesign	Tùng	Document	4	01/12	05/12
Patient System - Server implementation (cont.)	Khuê Trung Tùng	Server	4	01/12	05/12
Patient System - GUI implementation	Tùng	GUI	7	05/12	12/12
Payment System - Server implementation	Khuê Trung Tùng	Server	7	05/12	12/12

- Task information:

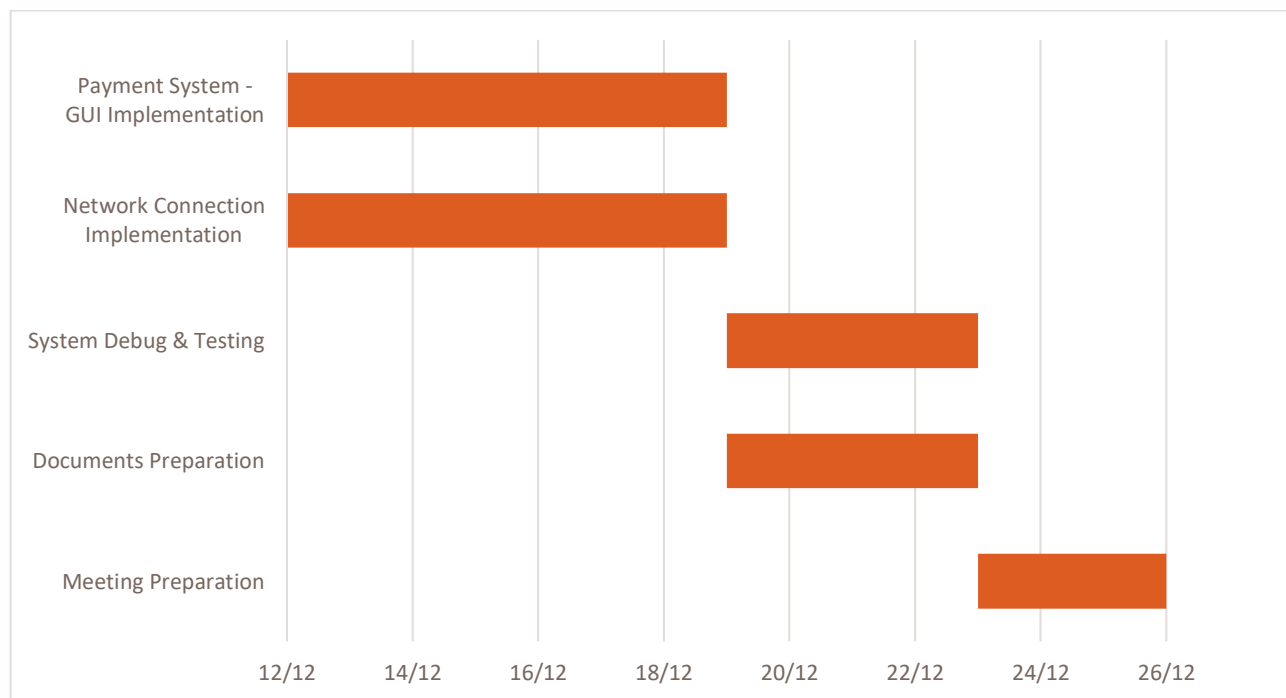


❖ *Sprint 4:*

- Duration: 14 days (12/12/2021 - 26/12/2021).
- Outcome: Payment System, Network Connection, Final Application.
- Task information:

Task	Member	Task type	Duration	Start date	End date
Payment System - GUI Implementation	Tùng	GUI	7	12/12	19/12
Network connection implementation	All	Network	7	12/12	19/12
System Debug and Testing	Khuê Trung	Coding	4	19/12	23/12
Documents preparation	Tùng	Document	4	19/12	23/12
Meeting preparation	All	Other	3	23/12	26/12

- Task information:



3. Risk Management

Risks will be identified in the Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risks are evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Magnitude	Risk Description	Mitigation Strategy
High	To complete the stakeholder's requirements, the team will deal with 3 aspects of programming: Java, Database and Networking. Applying all of those to the app might have trouble and easily lead to poor-quality products.	Each member has to do some study and research in all of those fields.
Medium	The team consists of 3 members instead of 4 as the maximum number. This will be a disadvantage for our team to compete with other teams. Overdue and poor-quality products might be the result of this point.	The whole project must be carefully separated into phases to ensure balanced workloads. Each member is required to read each phase's plan and self-managed time budget to complete his/her tasks on time.
Low	There might be a little gap in skill level among team's members. Misunderstandings may appear and slow down the team's process	Each member needs to keep a cooperative and supportive attitude towards teammates.
Low	A team member suddenly quits the project may lead to an unexpected result to the whole project.	All members must understand the whole project to be able to handle quitted member's work.

4. Workspace Management

a) Collaboration

Our team use GitHub as the official platform for collaboration. Beside folders and files containing code, required documents for developing and publishing our products will also be included in our GitHub.

Each member needs to make full use of GitHub to maximize the quality of products and enhance project's process.

b) Meeting

Meetings for showing project's process, solving unforeseeable problems and planning will be held 1-2 times a week on Zoom. Also, there are alternative platforms in use such as Google Meet or Messenger in case Zoom is not available.

Every member is required to participate in all meetings on time. Missing or more than 15 minutes late will lead that individuality to receive some penalties.

c) Discussion

Daily discussion with small scale will take place on our Messenger group. Members of the team are encouraged to discuss about algorithms and project-related articles on this channel.

VIII. References

- **MySQL Documentation**

<https://dev.mysql.com/doc/>

- **Group's Github**

https://github.com/ptvinhkhue/JavaProgramming_19KTPM2_Covid19ManagementSystem