APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 673-691
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Pregnant Checkup Management System for Private Clinic

Pang Shu Wen[1], Nurul Aswa Omar[2]*

[1,2] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: nurulaswa@**uthm**.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.036

Article Info

Received: 13 June 2024
Accepted: 28 September 2024
Available online: 15 Disember 2024

Keywords

Checkup, Management, Appointment, Clinic

Abstract

The Pregnant Checkup Management System faces issues such as long queues, inefficiency from duplicating checkup records for staff and patients, wasted time and resources, excessive paper use, and a lack of digitalization. The Pregnant Checkup Management System for Private Clinics in Kluang, Johor is a web-based solution designed to address inefficiencies in the manual recording of pregnant patient checkup information. Developed using the waterfall methodology, the system transitions from paper-based record-keeping to a computerized approach, enhancing security and effectiveness. Acting as an administrator, the head nurse has comprehensive permissions to edit, create, delete, and view information in the system. The findings includes saving time and paper, streamlining the checkup recording process, and providing efficient database storage, eliminating the need for manual searching. Ultimately, the proposed system aims to optimize the management of pregnant checkups in private clinics, offering a smoother and more effective process.

1. Introduction

Pregnancy is the state of having implanted products in the uterus, culminating in abortion, delivery, or elective abortion [1]. It involves fertilization, nidation, and typically lasts 40 weeks [2]. Nowadays, with the high improvement and advance of technology, many web-based systems have taken the place of manual paper-based systems. But many private clinics in Malaysia are still facing difficulties with long queues, especially for pregnant checkup patients [3]. The proposed pregnant checkup management system aims to replace paper-based systems in Malaysian private clinics, enhancing efficiency, reducing queues, and providing systematic database management. The system serves three separate users which are patients, nurses, and doctors. System allowing patients to view recorded information and generate reports, while healthcare professionals manage patient information, generate reports, and administer the system.

Despite the widespread use of computerized systems globally, private clinics, particularly in Malaysia, have yet to adopt them for managing pregnant patients' information and reports. The current reliance on manual, paper-based methods create three key issues. Firstly, extended wait times and inefficiencies arise due to the sequential manual recording of checkup information for patients in two separate books, causing delays and hindering the overall progress of daily checkups. Secondly, the lack of a computerized management system leads to time-consuming searches for patient information during appointments, with the added risk of document loss in unforeseen events such as fires. A computerized system would enhance efficiency and data security for private clinics. Lastly, patients face the inconvenience of bringing their checkup books to appointments, risking loss or forgetfulness. Manual recording also introduces the potential for inaccuracies in tracking the baby's

This is an open access article under the CC BY-NC-SA 4.0 license.

movements. Implementing a computerized pregnant checkup management system can mitigate these challenges, improving efficiency, reducing queues, and enhancing the overall progress of checkups in private clinics.

The objective of this project is to design a comprehensive pregnant checkup management system of private clinics for the clinics administration and patient at private clinics in Kluang, Johor, Malaysia, to develop a web-based management system for storing data and retrieve data from the database and to test the pregnant checkup management system for private clinics has successfully run with the functionality and usability. There are three users with each their own modules. Doctor acts as administrator to handle staff information module, patient module, checkup information module, appointment module and report module. For the nurse to handle module is almost same with doctor except for the doctor submodule in the staff information module. For the patient is to handle patient module, checkup information module, appointment module and report module. The expected result of this proposed system is to transform the traditional method, which is a manual system, into a computerized system to manage the pregnant checkup management system in private clinics. This will increase the efficiency of checkup progress, shorten the queue, reduce wasting time, and reduce paper waste. Increase the management and storage of the recorded information of patients in a secure manner and be able to retrieve it easily. Furthermore, the pregnant checkup management system can fulfil the objectives and solve the problems as stated in the problem statement.

2. Related Work

This part discussed the related work about checkup management system, web-based system and comparison between existing system and proposed system.

2.1 Clinic Management System

The checkup management system is a system that records the patient's checkup information in a systematic and computerized way to store all the related data in a database. The associated workers, such as nurses and doctors in a private clinic, will easily manage it. However, due to the growth of society, the paper format needs to be revised and made more manageable to record and serve a massive number of people efficiently and conveniently. Furthermore, it solves several problems of the traditional record method, such as inorganized storage, loss of the records due to increased patients, paper waste, and limited time and space for exchanging information [4]. Medical data can easily be stored and retrieved from the database, when necessary, anywhere, and at any time. At the same time, the checkup system will function widely for patients registering as users and will not be limited to storing medical records only. The system also included doctors and nurses licensed to practice in private clinics. This system enables the spread of communication between doctors, nurses, and patients to improve the security of confidential data and increase the effectiveness of electronic data storage [5].

2.2 Web-Based System

The proposed system is a web-based application designed for accessibility through web browsers, offering convenience to nurses, doctors, and pregnancy patients without requiring installation on devices, like a mobile application. Users can access the system through a website link with login credentials, providing administrative capabilities for healthcare professionals and user accounts for pregnant patients. The shift towards web-based systems is crucial in modern healthcare, addressing the complexities and potential loss associated with paper-based data storage [6]. The centralized nature of web-based systems ensures uniform software usage, reducing costs and efforts associated with individual device installations and upgrades [7]. This system's accessibility from any location empowers patients to schedule appointments, update pregnancy-related information, and allows nurses to receive real-time updates. Utilizing a web-based system aligns with the project's objectives, promoting paperless practices, efficiency, and time-saving measures in private clinics.

2.3 Comparison between Existing System and Proposed System

Three existing systems will be discussed and investigated through each characteristic of the system, including advantages, disadvantages, and features, to compare with the proposed system. The three existing systems are Doctor2U, Pomona Valley Health Centers, and MedPark Hospital that are shown in Table 1 below. Legend for √ is have that particular feature in the system, while for × is do not have that particular features in the system.

Table 1 *Comparison between three existing systems and proposed systems*

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 673-691

Features/Systems	Doctor2U	Pomona Valley Health Centers	MedPark Hospital	Proposed System
Registration and login	√	×	√	√
Change system language	×	√	√	×
Checkup service	√	√	√	√
Appointment	×	√	√	√
Notice information board	√	√	√	√
Package and promotion	√	×	√	×
Make payment through website	√	√	×	×
Patient module	√	√	√	√
Staff module	×	√	×	√
Print report	×	×	×	√

Table 1 above shows the comparison between three existing systems and proposed systems. There are 3 features such as "Checkup service", "Notice information board", and "Patient module" have the same features between Doctor2U system, Pomona Valley Health Centers system, MedPark Hospital system and pregnant checkup management system which is proposed system. The proposed system lack of function change system language that Pomona Valley Health Centers system and MedPard Hospital possess these features. While proposed system lack another feature which is "Package and promotion" features. Proposed system is mainly for manage checkup information, hence there is no product or service that can be sell to the patient as a external services. Same goes for features "Make payment through website", it is unnecessary in proposed system because it is checkup service provided to pregnancy patient which is not involving any patient as it is manage information.

3. Methodology

This section shows the waterfall model that used to develop the proposed system. Waterfall model consists of a list activity that are required implement in each phase.

3.1 Waterfall Model

The waterfall model was introduced in the year 1970 by Winston Royce. The waterfall model is a sequential project management methodology that begins with another phase before the previous phase has finished. There are six stages included in the waterfall model such as requirements, analysis, design, implementation, testing, and deployment phase [8]. Furthermore, in creating this system, the waterfall methodology has been opted for as it is well-suited for generic systems or software. This means the system can identify all its requirements from the beginning, along with broad specifications. It is particularly suitable for software projects aiming to build a system from the ground up. The process involves gathering system requirements aligned with the chosen research topic and progressing through development until the product is thoroughly tested [9]. Waterfall methodology is used in this proposed system. Figure 1 below shows the waterfall model for each phase.

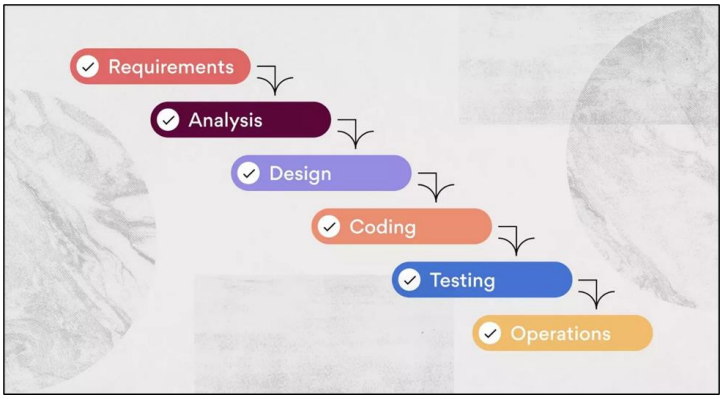


Fig. 1 *Waterfall model for each phase. Adapted from [10]*

3.2 System Development Workflow

Table 2 below shows the system development workflow. There are 6 phases in the water model and implementation during the development of the proposed system. The first phase was the requirement phase, which was to gather the required information and identify the problem that occurred to decide to propose a system to solve that problem. Interview sessions with targeted clinics must be done. At the same time, the second phase is the analysis phase. They are analyzing the modules that should be in the system, including limitations, the system workflow, and other related considerations. The third phase was the system design phase. This phase required a wireframe as a draft and designing the output as we think of it. DFD, ERD, and flowchart for each user were vital to have. The fourth phase is implementation, which is the development of the system's coding and database. After completing the development system, one of the staff clinics will test the proposed system in the fifth phase, the naming testing phase. At the same time, the last phase is the operations phase, which releases the complete developed system.

Table 2 *System development workflow*

Phase	Activity	Deliverable	Tools
Requirements	-Gather important information -Determine the objective of developed system -Determine the project schedule and methodology used -Determine budget and time used	-Conduct interview session with the target clinic -Identify project scope -Illustrate Gantt Chart	-Microsoft Project
Analysis	- Analyze the limitation of manual system - Determine current system - Analyze the flow of system Analyze the system modules Analyze functional and non-functional requirements	- List out the details information of functional requirements and non-functional requirements	-Draw .io
System Design	-Illustrate diagram to design the system process -Design database management system -Design user interface	- Illustrate flowchart -Illustrate context diagram -Illustrate data flow diagram (DFD) -Illustrate entity-relational diagram (ERD) -Illustrate user interface design	-Draw .io
Implementation	- Write the coding of the developed system -Create the database to connect with the developed system	- Develop the source code and connect by creating database of the system	- Visual Studio Code -XAMPP

Table 2 *System development workflow (cont.)*

Testing	- Module testing -Integrate testing -Fix debug	-Integrate and refine the coding	-Visual Studio Code
Operations	-Release the developed system		

4. System Analysis and Design

Based on user requirements, defining the proposed system involves designing the system architecture, flowchart, context diagram, data flow diagram (DFD), and entity-relationship diagram (ERD) to ensure a thorough understanding and effective planning. These designs must meet the requirements and be usable.

4.1 System Requirement Analysis

Requirement analysis is a phase in which information is collected and gathered to produce a complete and successful system by collecting characteristics and features of the system. In this project, an interview session with the "Pusat Pakar Wanita Dan Perbidanan Johor," located in Kluang, Johor, was carried out with the nurse. The nurse has provided information related to the process of the checkup patient, the way of making appointments, and the way they store the patient's data. There are two types of requirement analysis which are functional requirement analysis and non-functional requirement analysis. Table 3 and Table 4 below show the functional requirement analysis and non-functional requirement analysis respectively.

Table 3 *Functional requirement analysis*

Modules	Functionalities	User
Login	• Patients are required to login to the system by entering email and password. • All type nurse and doctor are required to login to the system by entering ID and password. • System allows all users to change password if they forgotten their password.	Doctor, head nurse, general nurse and patient
Register	• Head nurse will register an account for doctor and nurse. • Patient able to register an account on their own.	Head nurse and patient
Manage nurse information	• Head nurse has permission to create, read, update, and delete nurse information and quantity of the nurse.	Head nurse
Manage doctor information	• Head nurse able to create, read, update, and delete doctor information.	Head nurse
Manage patient information	• Doctor and all type nurse able to add, view, update and delete patient information according to the quantity of the patient.	Doctor, head nurse, and general nurse
Record checkup information	• Doctor and all type nurse will manage and record the checkup information of the patient during process of pregnant checkup.	Doctor, head nurse, and general nurse
Manage appointment	• Doctor and all type nurse able to control appointment date and time by add, view, update and	Doctor, head nurse, general nurse, and

Table 3 *Functional requirement analysis (cont.)*

	delete after discussing with the patient.	patient
	• Patient able to add, view, update, and delete appointment according the convenient and free time of the patient.	
Search information of patient	• All type nurse able to search information of patient in the list of the patient that have been registered and recorded.	Doctor, head nurse, and general nurse
Search information of nurse	• Doctor and head nurse able to search information of nurse in the list of the nurse that have been registered and recorded.	Doctor and head nurse
Generate and print report	• System can generate report for doctor, nurse, and patient to print out when they are required.	Doctor, head nurse, general nurse, and patient
Logout	• All the users can logout the system.	Doctor, head nurse, general nurse, and patient

Table 4 *Non-Functional requirement analysis*

Non-functional requirement	Functionality
Performance	• The response time of the system is short. • The functionality of the system is available for the doctor, nurse, and patient to perform create, read, update, and delete.
Security	• Required email and password for the patient. • Required ID and password for the nurse and doctor. • Password entered by all the users is encrypted.
Usability	• The system is user-friendly to all the users which is ease of understand and ease of use.
Availability	• The system is accessible in 24 hours every day for each user to use.
Reliability	• The system will reach the accuracy for pregnant checkup management.
Operational	• The system is able to be used in any web browsers such as Chrome, Microsoft Edge and so on. • All the data and information will be stored in the database.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 673-691

4.2 Context Diagram

The context diagram, a level 0 data flow diagram, includes three components: external entities, process, and data flow lines. The external entities are the doctor, nurse, and patient. The process represents the pregnant checkup management system for a private clinic. Input and output data flow between the process and external entities through input and output flow lines, enabling the system to process information. Figure 2 below shows the context diagram for Pregnant Checkup Management System for Private Clinic.

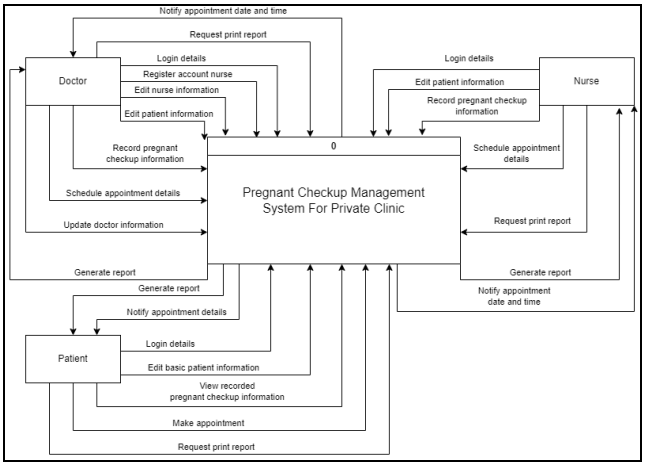


Fig.2 *Context Diagram for Pregnant Checkup Management System for Private Clinic*

4.3 Data Flow Diagram (DFD)

Data flow diagram (DFD) displays the way information moves through a system in a graphical. Normally DFD will serve as a first stage to create system overview. DFD will display the flow of the data into and out of the system. The input of the DFD for the process 1.0 is register and login, process 2.0 is manage staff information, process 3.0 is manage patient information, process 4.0 is manage checkup information, process 5.0 is manage appointment and process 6.0 is generate report. The Pregnant Checkup Management System will be listed in Appendix A.

4.4 Entity Relationship Diagram (ERD)

Entity relationship diagram is a structural diagram that is used to illustrate the database design. There are seven database tables in the system. The ERD will be attached in Appendix B.

4.5 Flowchart Diagram

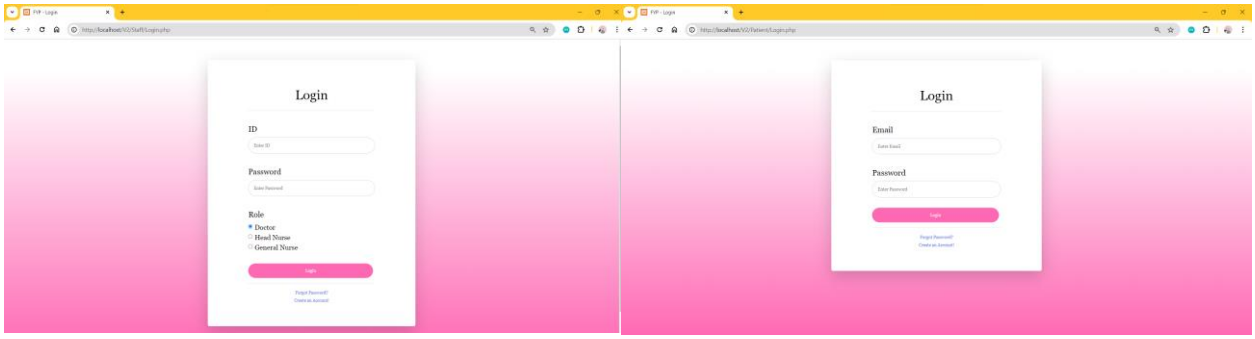
The system flowchart is essential for illustrating the process flow before developing a comprehensive system. It sequentially defines each system process step by step. The flowchart consists of various shape boxes, each representing different functionalities. These boxes are interconnected with data flow lines that logically represent the connections throughout the flowchart. The flowchart for doctor, nurse, and patient side will be attached to Appendix C.

5. Implementation

This section displays the output of the project outcome by displaying interface and code segment of the web system for each module in detail. The modules in this system are staff module, patient module, checkup management module, appointment module and report module.

5.1 Login Module

For the login module, the staff need to enter correct ID, password, and choose role while patients need to enter correct email and password to login to the system. Figure 3 below shows interface and code segment of login module.



(a) **(b)**

```php
// Execute doctor query
mysqli_stmt_execute($stmtDoctor);

// Get result for doctor query
$resultDoctor = mysqli_stmt_get_result($stmtDoctor);

// Check if result is valid for doctor query
if ($resultDoctor && mysqli_num_rows($resultDoctor) > 0) {
    $rowDoctor = mysqli_fetch_assoc($resultDoctor);
    // Verify the entered password against the hashed password in the database
    if (password_verify($enteredPassword, $rowDoctor['password'])) {
        // Password is correct for a doctor
        $_SESSION['dr_id'] = $id;
        $_SESSION['role'] = 'doctor'; // Set role

        // Verify role
        if ($role !== 'doctor') {
            echo "<script>alert('Incorrect role for doctor');</script>";
            exit();
        }

        // Redirect user to the appropriate homepage based on role
        header("Location: /V2/Staff/HomePageStaff.php");
        exit();
    } else {
        // Password verification failed for doctor
        echo "<script>alert('Incorrect password');</script>";
        exit();
    }
}
```

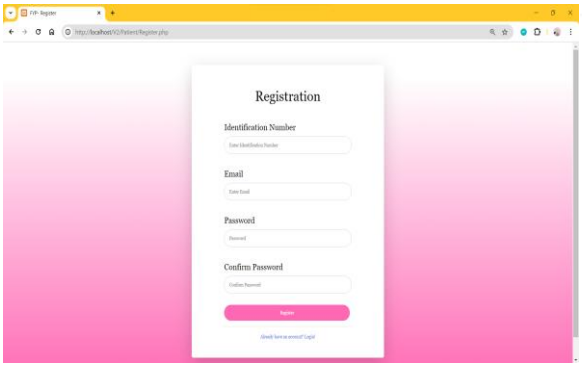
```php
<?php
session_start();

if (isset($_POST['login'])) {
    // Include the database configuration file
    include("../Config.php");

    // Check if the database connection is successful
    if (!$mysqli) {
        die("Database connection failed: " . mysqli_connect_error());
    }

    // Get the form data
    $email = $_POST['email'];
    $password = $_POST['password'];

    // Prepare the SQL statement
    $sql = "SELECT patient_ic, email, password FROM patient_basic_information WHERE email = ?";
    $stmt = $mysqli->prepare($sql);

    // Bind parameters
    $stmt->bind_param("s", $email);

    // Execute the SQL statement
    $stmt->execute();

    // Store the result
    $stmt->store_result();

    // Check if the user exists
    if ($stmt->num_rows == 1) {
        // Bind the result to variables
        $stmt->bind_result($db_patient_ic, $db_email, $db_password);

        // Fetch the result
        $stmt->fetch();
```

(c) **(d)**

Fig.3 *Login module (a) Interface staff side; (b) Interface patient side; (c) Code segment staff side; (d)Code segment patient side*

5.2 Register Module

For the registration module, registration for staff is done by the head nurse, which is role as admin. Patients are required to register an account for the first time when their identification number is recorded by staff in the system. Registration for a patient for the first time is to validate whether have stored the identification number previously in the database. Figure 4 below shows the interface and code segment of register module.



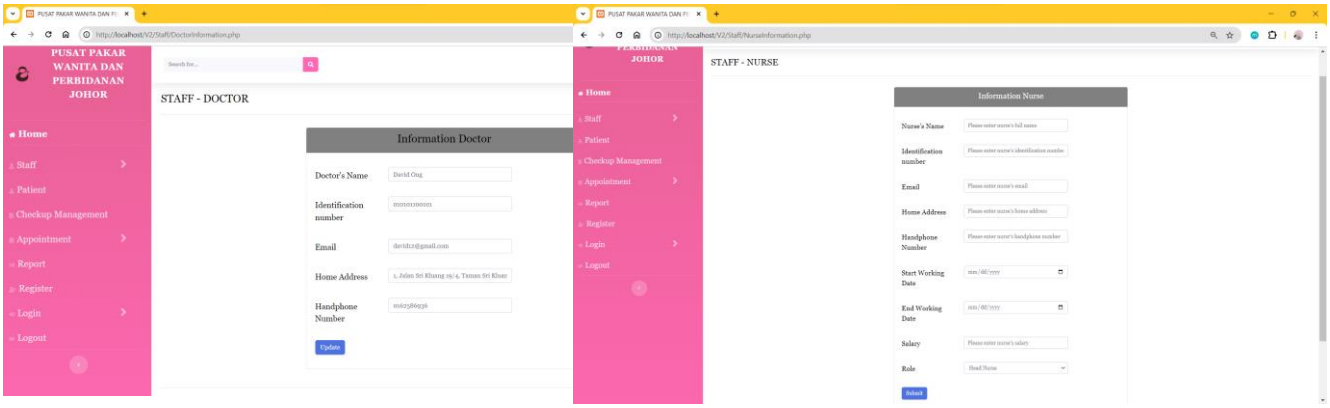

```php
// Check if form is submitted
if (isset($_POST['register'])) {
    // Retrieve and sanitize input
    $patient_ic = isset($_POST['patient_ic']) ? mysqli_real_escape_string($mysqli, $_POST['patient_ic']) : '';
    $email = isset($_POST['email']) ? mysqli_real_escape_string($mysqli, $_POST['email']) : '';
    $password = isset($_POST['password']) ? mysqli_real_escape_string($mysqli, $_POST['password']) : '';
    $confirm_password = isset($_POST['confirm_password']) ? mysqli_real_escape_string($mysqli, $_POST['confirm_password']) : '';

    // Validate input
    if (empty($patient_ic) || empty($email) || empty($password) || empty($confirm_password)) {
        // Handle empty fields
        echo '<script>alert("Please fill in all the fields.");';
        echo 'window.location.href = "/V2/Patient/Register.php";</script>';
        exit;
    }

    // Check if the patient is registered by staff
    if (!isPatientRegistered($mysqli, $patient_ic, $email)) {
        // Handle patient not registered by staff
        echo '<script>alert("This identification number or email is not registered by staff.");';
        echo 'window.location.href = "/V2/Patient/Register.php";</script>';
        exit;
    }

    // Validate password
if ($password !== $confirm_password) {
    // Handle password mismatch
    echo '<script>alert("Passwords do not match.");window.location.href = "/V2/Patient/Register.php";</script>';
    exit;
}

if (strlen($password) < 8 || !preg_match("/^(?=.*\d)(?=.*[A-Z])(?=.*[a-z])(?=.*\W).{8,}$/", $password)) {
    // Handle insufficient password complexity
    echo '<script>alert("Password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one number, an
    echo 'window.location.href = "/V2/Patient/Register.php";</script>';
```

(a) **(b)**

Fig.4 *Registration module for patient (a) Interface; (b) Code segment*

5.3 Staff Module

For the staff module, there will be doctor submodules and nurse submodules. Doctor submodule is updating doctor personal information by doctor. Nurse submodules have to insert, update, view, and delete own and other general nurse by the head nurse. While general nurses are able to update their own information and view. Figure 5 below shows the interface and code segment of staff module.



(a) **(b)**

```php
$message = ""; //Initialize an empty variable to hold the message

// Retrieve doctor ID from session
$dr_id = $_SESSION['dr_id'];

// Retrieve doctor information from the database based on dr_id
$sql_select_doctor = "SELECT * FROM doctor WHERE dr_id = ?";
$stmt_select_doctor = mysqli_prepare($conn, $sql_select_doctor);
if (!$stmt_select_doctor) {
    die("Error in preparing statement: " . mysqli_error($conn));
}
mysqli_stmt_bind_param($stmt_select_doctor, "i", $dr_id);
if (!mysqli_stmt_execute($stmt_select_doctor)) {
    die("Error in executing statement: " . mysqli_error($conn));
}
$result_select_doctor = mysqli_stmt_get_result($stmt_select_doctor);

if (!$result_select_doctor) {
    die("Error in SQL query: " . mysqli_error($conn));
}

// Check if the doctor exists
if (mysqli_num_rows($result_select_doctor) > 0) {
    // Doctor exists, fetch the data
    $row_doctor = mysqli_fetch_assoc($result_select_doctor);

    // Populate the form fields with the retrieved data
    $full_name = $row_doctor['full_name'];
    $ic_no = $row_doctor['ic_no'];
    $email = $row_doctor['email'];
    $address_home = $row_doctor['address_home'];
    $handphone_no = $row_doctor['handphone_no'];
} else {
    $message = "Doctor not found.";
}
```

(c)

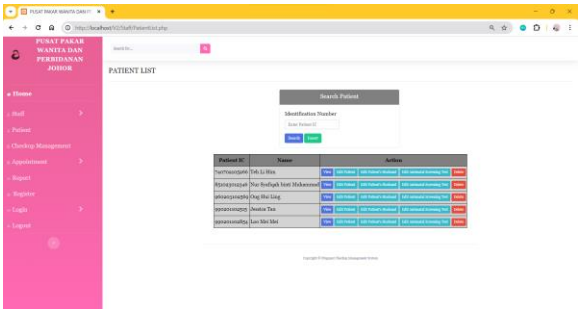
```php
    // Prepare and execute the SQL query for insertion
    $sql_insert = "INSERT INTO nurse (nurse_id, full_name, ic_no, email, address_home, handphone_no, start_date, end_date, salary,
    $stmt_insert = mysqli_prepare($conn, $sql_insert);
    if (!$stmt_insert) {
        die("Prepare failed: " . mysqli_error($conn));
    }

    // Generate temporary password
    $temporary_password = generateTemporaryPassword();

    // Hash the temporary password for storage
    $hashed_password = password_hash($temporary_password, PASSWORD_DEFAULT);

    mysqli_stmt_bind_param($stmt_insert, "sssssssssss", $nurse_id, $name, $ic_no, $email, $address_home, $handphone_no, $start_date

    if (mysqli_stmt_execute($stmt_insert)) {
        $message = "Data stored in the database successfully.";
        $message = "Generated Nurse ID: $nurse_id";

        // Send the temporary password email
        if (sendTemporaryPasswordEmail($email, $temporary_password)) {
            $message = "Temporary password sent to the nurse's email.";
        } else {
            $message = "Failed to send the temporary password email.";
        }
    } else {
        echo "ERROR: Unable to execute $sql_insert. " . mysqli_error($conn);
    }
    mysqli_stmt_close($stmt_insert);
}
mysqli_close($conn);
}

// Function to generate a 5-digit nurse ID
1 reference
function generateNurseID() {
```

(d)

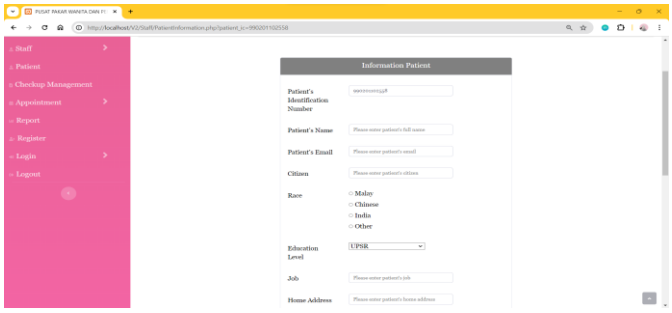
Fig.5 *Staff module (a) Interface update doctor information; (b) Interface update nurse information; (c) Code segment for update doctor information; (d) Code segment for update nurse information*

5.4 Patient Module

For the patient module, all staff are accessible to insert new patients, update, view, and delete old patients for the staff side. While for the patient side, patients are able to view, update their own basic information, and update their husband information only. Figure 6 below shows the interface and code segment of patient module.



(a)



(b)

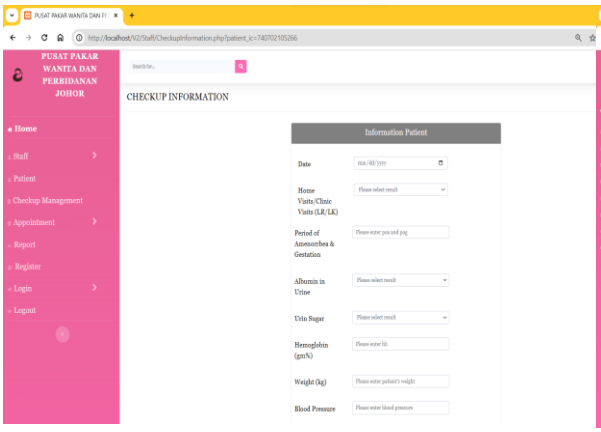
```php
if ($_SERVER['REQUEST_METHOD'] === 'GET' && isset($_GET['patient_ic'])) {
    $patient_ic = $_GET['patient_ic'];

    // Check if the patient exists
    $query = "SELECT * FROM patient_basic_information WHERE patient_ic='$patient_ic'";
    $result = mysqli_query($conn, $query);

    if (mysqli_num_rows($result) > 0) {
        // Patient exists, show the form with patient details
        // You need to add your HTML form here to display patient details and allow updating if needed
        $message = "Patient IC: $patient_ic already exists in the database.";
    } else {
        // Patient does not exist, insert the patient IC into the database
        $insert_query = "INSERT INTO patient_basic_information (patient_ic) VALUES ('$patient_ic')";

        if (mysqli_query($conn, $insert_query)) {
            $message = "Patient IC: $patient_ic inserted successfully.";
            // Redirect to the same page with patient_ic parameter to display the form
            header("Location: /V2/Staff/PatientInformation.php?patient_ic=$patient_ic");
            exit();
        } else {
            $message = "Error inserting patient IC: " . mysqli_error($conn);
        }
    }
} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get form data and perform basic validation
    $patient_ic = $_POST['patient_ic'];
    $name = $_POST['name'];
    $email = $_POST['email'];
    $citizen = $_POST['citizen'];
    // Correctly capture the race field based on the selected option
    if (isset($_POST['race'])) {
        $race = $_POST['race'];
        if ($race === 'OTHER') {
            $race = isset($_POST['other_race']) ? $_POST['other_race'] : ""; // If "Other" is selected, use the value from the text input field
```

(c)

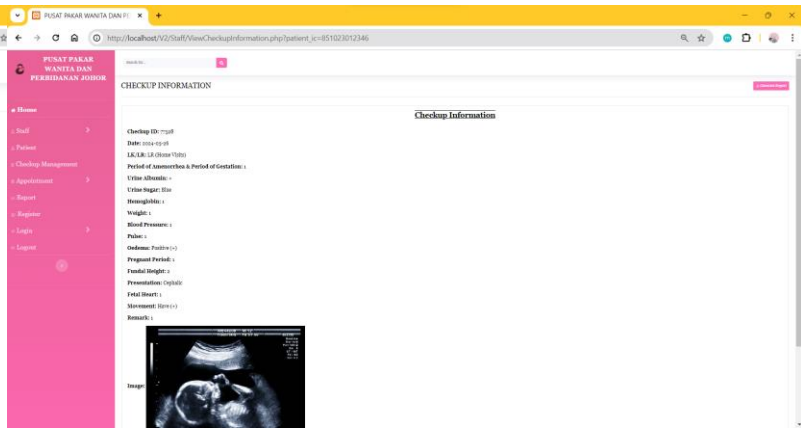
Fig. 6 *Patient module (a) Interface search patient with patient list; (b) Interface insert new patient; (c) Code segment insert new patient*

5.5 Checkup Management Module

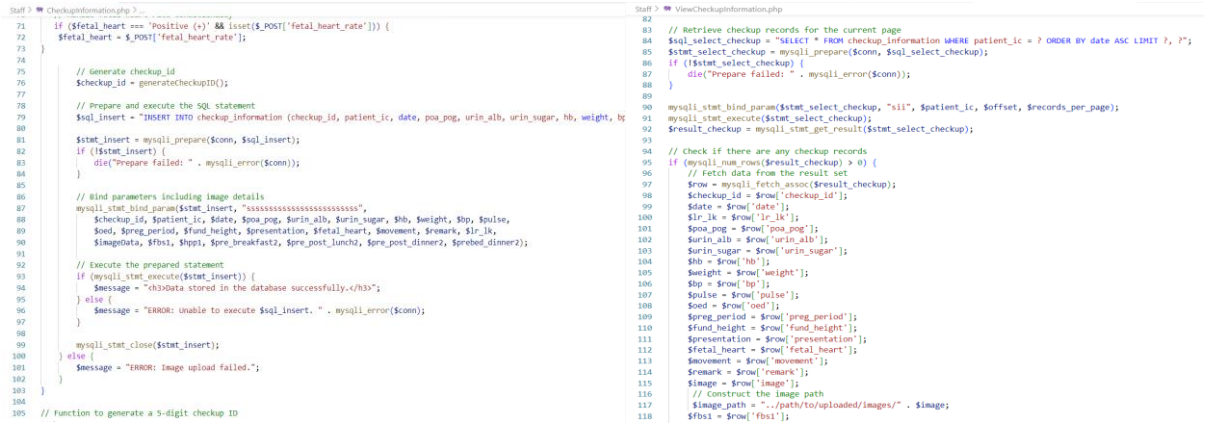
For the checkup management module, the staff side is able to search the patient wanted to record new page of checkup information for each time checkup duration. Staff are also able to update the checkup information and delete. While for the patient side is only able to view their own checkup information each time recorded. Figure 7 below shows the interface and code segment of checkup management module.



(a)



(b)

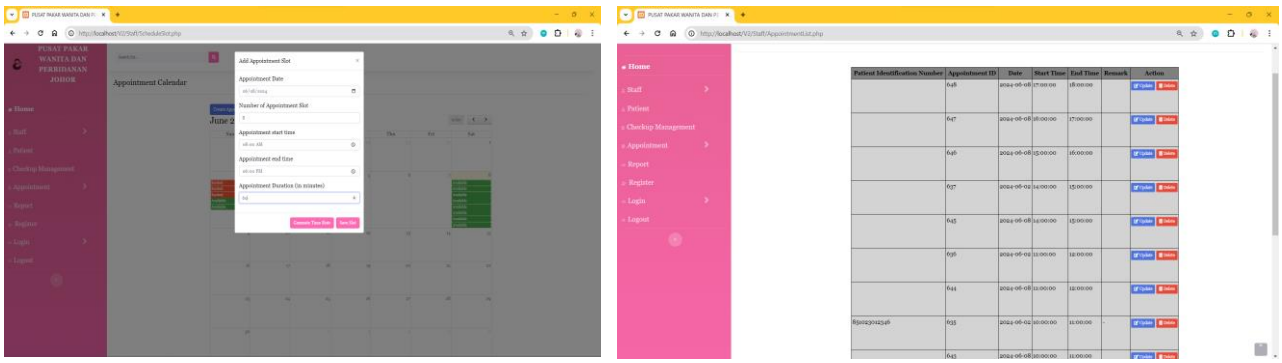


(c) **(d)**

Fig.7 *Checkup management module (a) Interface insert new checkup information; (b) Interface view new checkup information; (c) Code segment insert new checkup information; (d) Code segment view checkup information*

5.6 Appointment Module

For the appointment module, staff will create appointment slots for patient to make appointments. Patients will be able to select the wanted appointment slot or select by the staff. There is a appointment list on the staff side, they can perform update and delete appointment slot. For the patient side only able to view the appointment they made on the list. Figure 8 below shows the interface and code segment of appointment module.



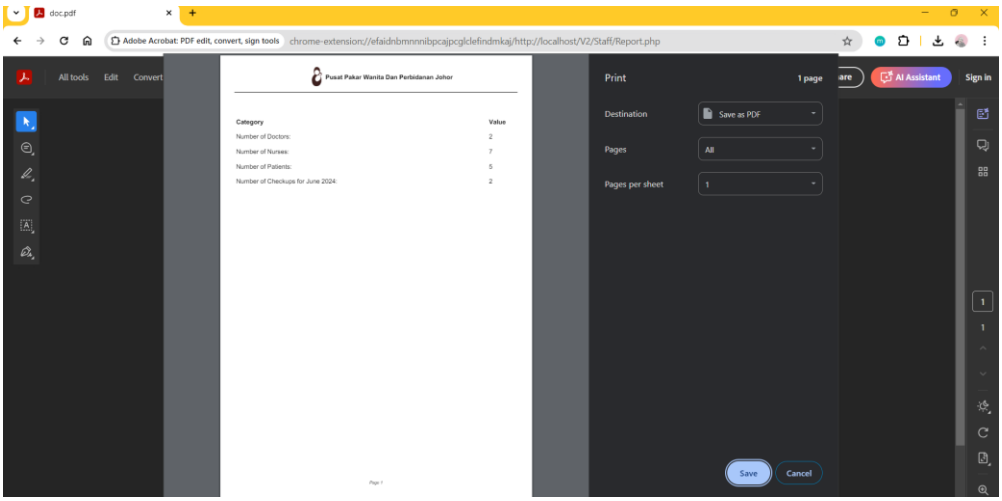
(a) **(b)**

(c) **(d)**

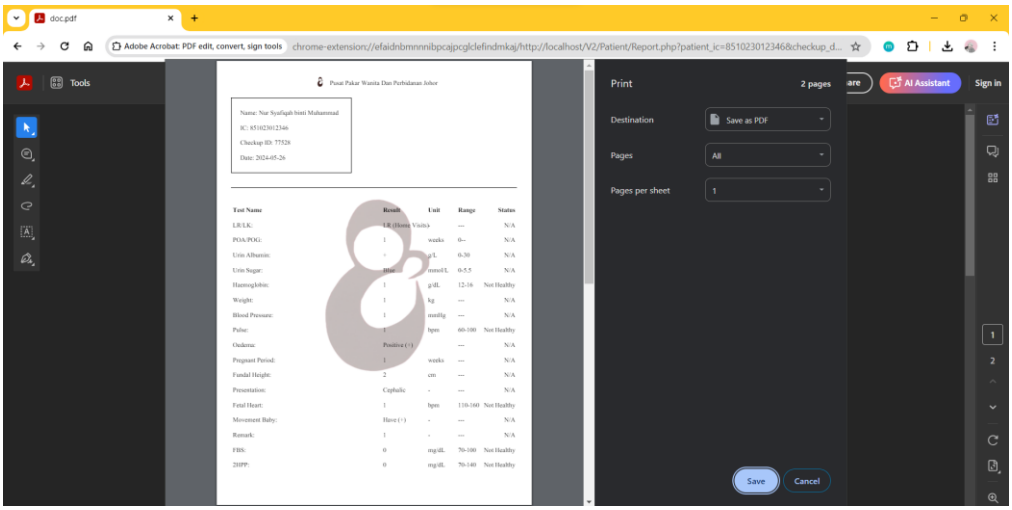
Fig.8 *Appointment module (a) Interface of create appointment slot; (b) Interface of appointment list; (c) Code segment for create appointment slot; (d) Code segment for appointment list*

5.7 Report Module

For the report module, the staff report will print out the summary number of staff, patients and checkups done for current month which is view by staff. While patient report will print out their own checkup information based on the date of checkup they search, and it is view by patient. Figure 9 below shows the interface of report module for staff side and patient side.



(a)



(b)

Fig.9 *Interface of report module (a) Staff side; (b) Patient side*

6. Conclusion

In conclusion, the development of the Pregnant Checkup Management System marks a significant achievement in streamlining the process of pregnancy checkups for both clinic staff and patients. While the system successfully addresses various challenges such as long queues and data duplication, it does exhibit some limitations, particularly in the reporting and notification functionalities. However, with planned future enhancements like implementing a "Download All" button for reports and integrating automated notifications for appointment scheduling, these shortcomings can be effectively addressed, further optimizing the system's

performance and usability. Overall, the system stands as a testament to the collaborative efforts aimed at improving healthcare delivery and patient experience in prenatal care settings. Although all modules and functions of the Pregnant Checkup Management System have been successfully developed and achieved the project objectives, this system still has some limitations. For the report module, there is no button to download all the checkup information reports with different dates at once. Patients are required to download the checkup reports one by one. Besides that, when staff creates an appointment slot, there is no notification to inform patients when an appointment slot is available. When patients have made an appointment by booking, there is no notification to notify staff immediately about which patients have made an appointment on which date. The system requires staff to view the appointment made through the appointment calendar or appointment list. Since there are existing limitations on the web-based system, several improvements or solutions can assist in enhancing the system to increase performance and functionalities. There should be a button, "Download All," for patients to download all the checkup reports without finding the date to download them one by one to save time and make it convenient for patients. Next, the system can be enhanced by adding notifications. When staff create new appointment slots, the system should send notifications through email to each registered patient automatically. The same goes for patients. When a patient makes an appointment, the system should send an email notification to the staff's email to inform them of the date and time to have an appointment with the patient.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Pang Shu Wen, Nurul Aswa Omar;* ***data collection:*** *Nurul Aswa Omar;* ***analysis and interpretation of results:*** *Pang Shu Wen, Nurul Aswa Omar;* ***draft manuscript preparation:*** *Nurul Aswa Omar. All authors reviewed the results and approved the final version of the manuscript.*

An author name can appear multiple times, and each author name must appear at least once. For single authors, use the following wording:

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Z. N. Pascual and M. D. Langaker, "Physiology, Pregnancy," StatPearls, May 16, 2023. https://www.ncbi.nlm.nih.gov/books/NBK559304/.

[2] R. J. Swanson and B. Liu, "Conception and pregnancy," in Fertility, Pregnancy, and Wellness, pp. 53-71, Elsevier, 2022. DOI: 10.1016/B978-0-12-818309-0.00011-3.

[3] K. B. Koay, A. Sukumar, L. Letchmunan, and A. W. Marashdih, "Improving clinic queues in Malaysia using time-series extrapolation forecast and web-based appointment," Compusoft: An International Journal of Advanced Computer Technology, vol. 8, no. 9, 2019.

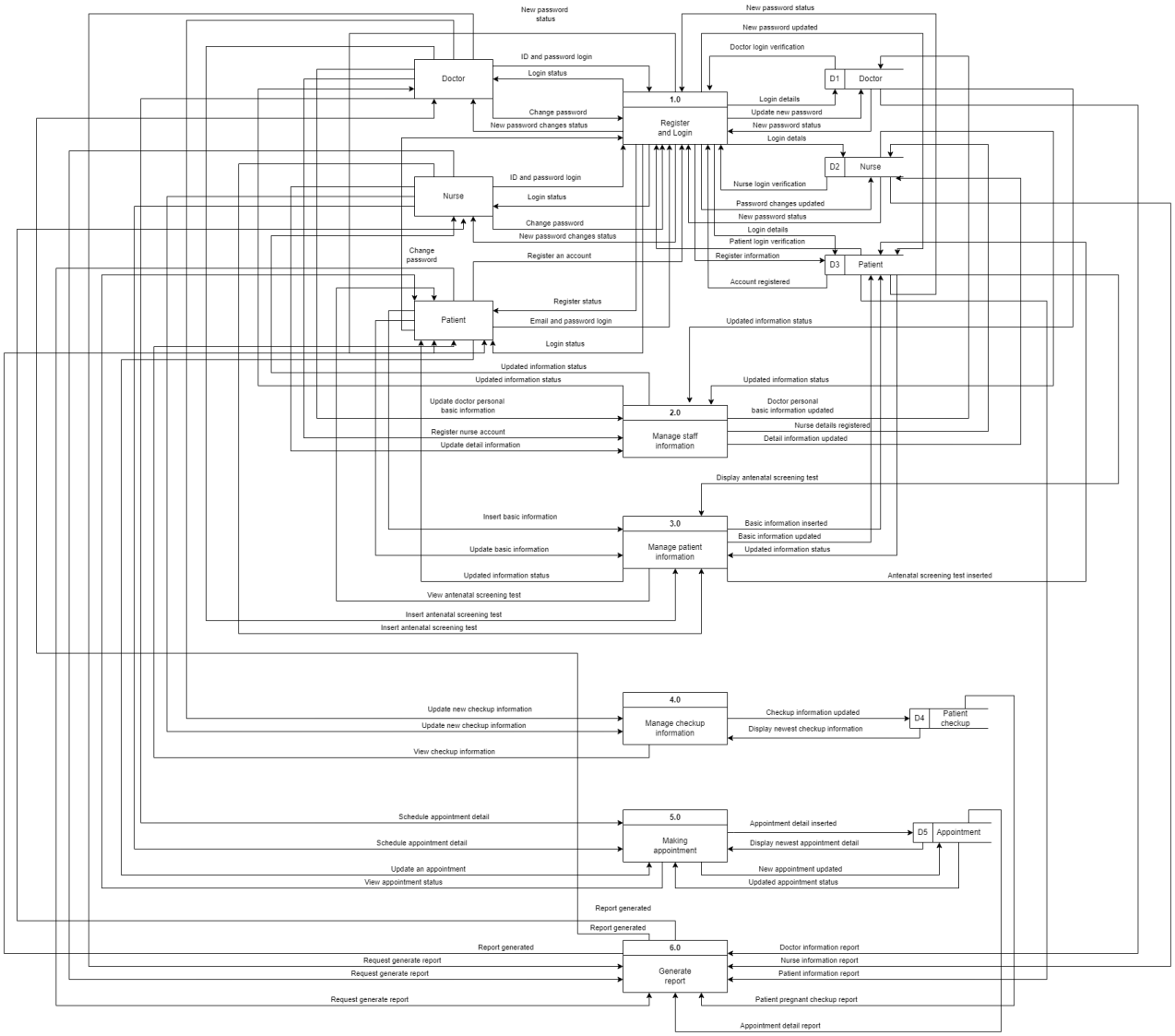
[4] Y. Zhao, L. Liu, Y. Qi, F. Lou, J. Zhang, and W. Ma, "Evaluation and design of public health information management system for primary health care units based on medical and health information," Journal of infection and public health, vol. 13, Art. no. 4, 2020.

[5] M. Mounica, B. Madhu, Rao, S. Sai, and Raju, "Automated data management for HealthCare using machine learning," in IEEE, 2022, pp. 1080–1083.

[6] A. C. Babu, V. N. C. S. Teja, A. D. Reddy, E. N. Kumar, and V. Srinivas, "Web Based Hospital Management System," in 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 1109-1113, IEEE, March 2023.

[7] N. R. Dissanayake and G. K. A. Dias, “Rich Web-based Applications: An Umbrella Term with a Definition and Taxonomies for Development Techniques and Technologies,” 2018. https://doi.org/10.18178/ijfcc.2018.7.1.513.
[8] A. Sinha and P. Das, “Agile methodology vs. traditional waterfall SDLC: A case study on quality assurance process in software industry,” in 2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), pp. 1-4, IEEE, September 2021.
[9] S. Herawati, Y. D. P. Negara, H. F. Febriansyah, and D. A. Fatah, "Application of the waterfall method on a web-based job training management information system at Trunojoyo University Madura," in E3S Web of Conferences, vol. 328, p. 04026, EDP Sciences, 2021.
[10] A. Lavergne, “Top 10 Project Management Methods,” *Salesdorado*, Oct. 20, 2022. https://salesdorado.com/en/sales-strategy/methodes-de-gestion-de-projet/.

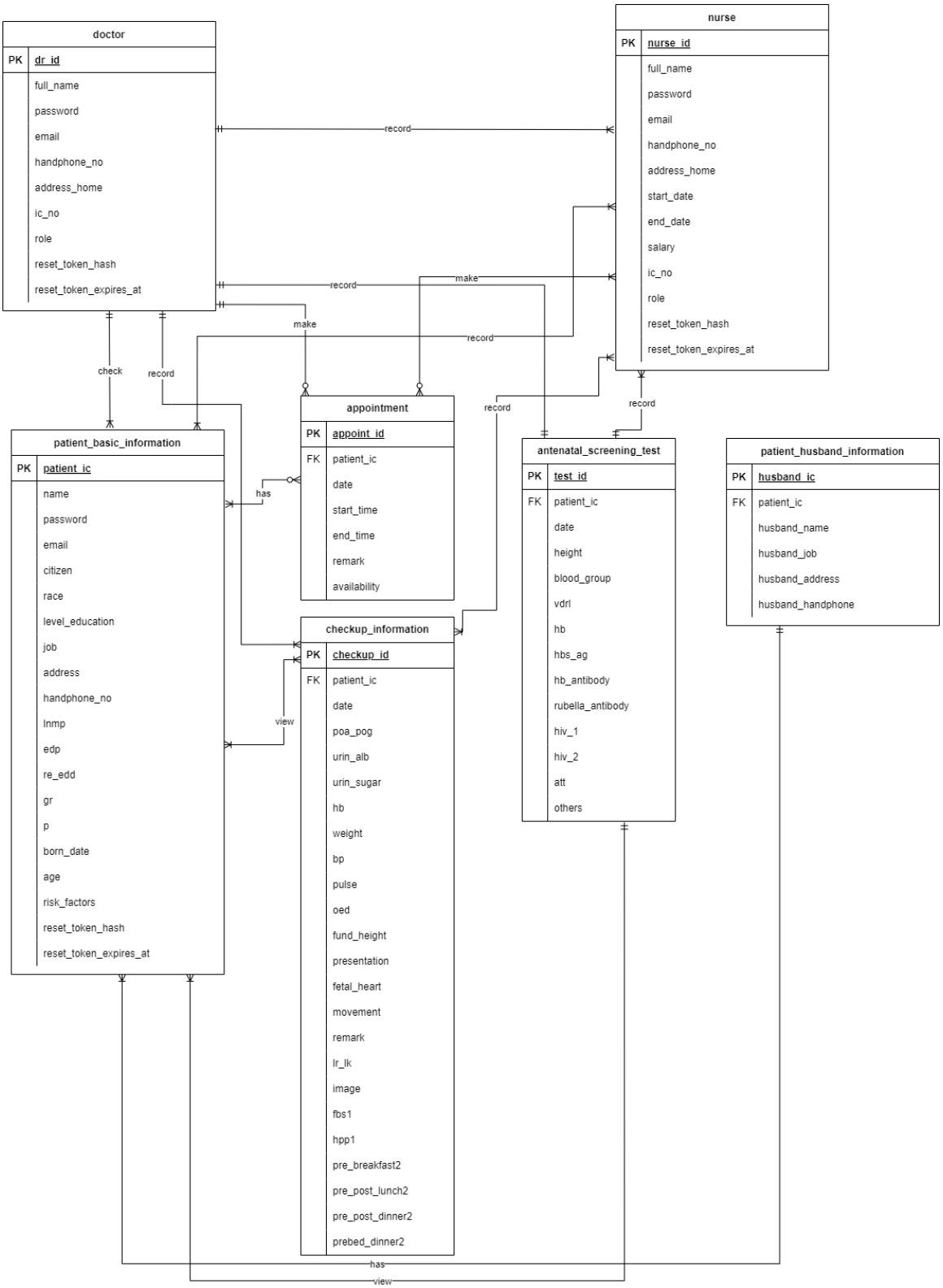
Appendix A: Data Flow Diagram



Data Flow Diagram

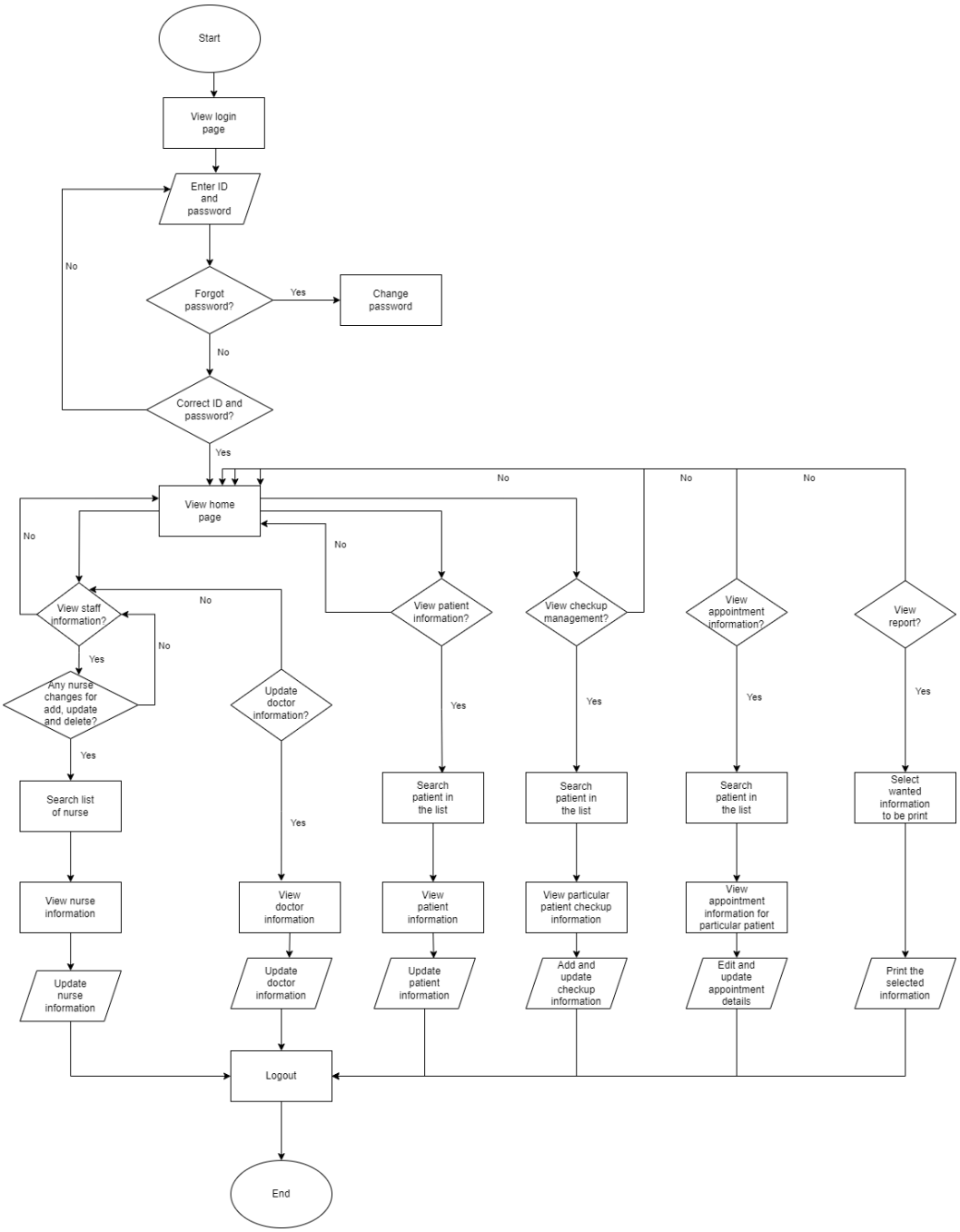
Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 673-691

Appendix B: Entity Relationship Diagram



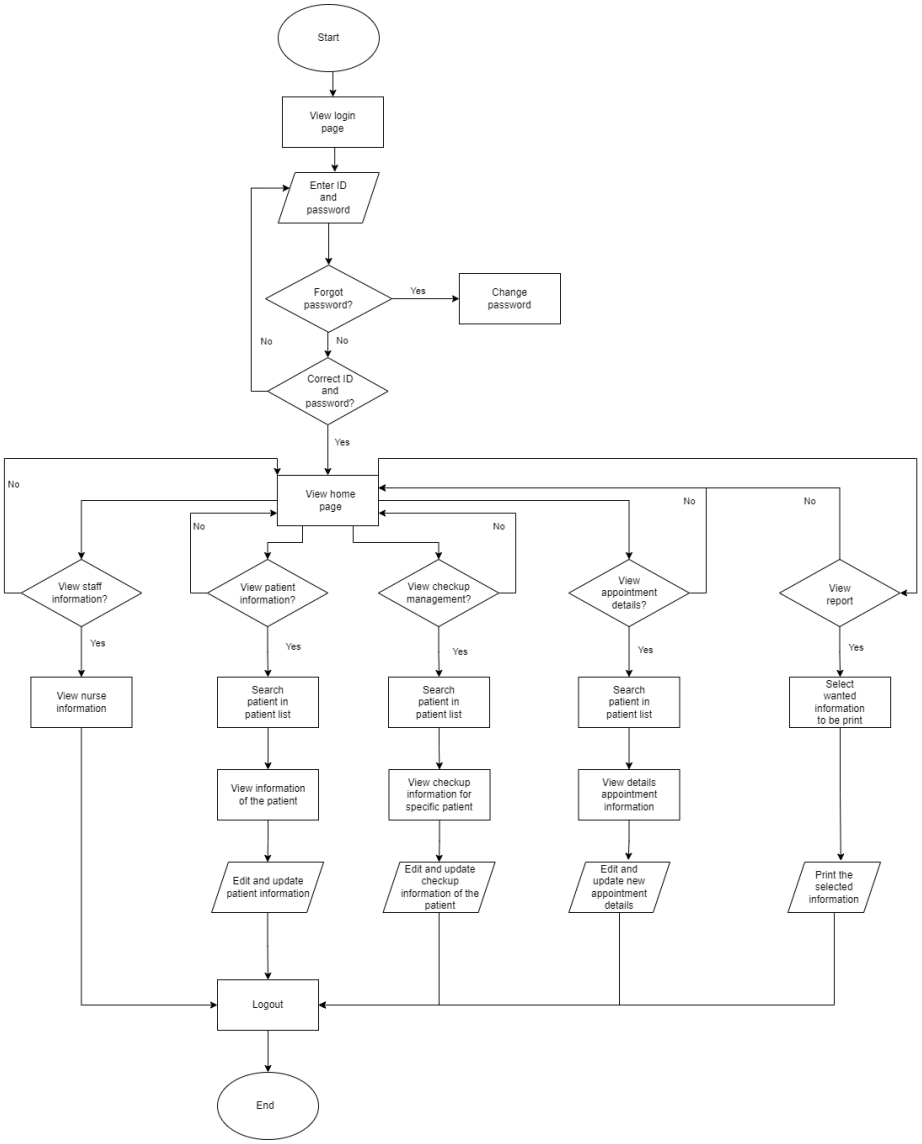
Entity Relationship Diagram

Appendix C: Flowchart Diagram for Doctor, Nurse and Patient Side

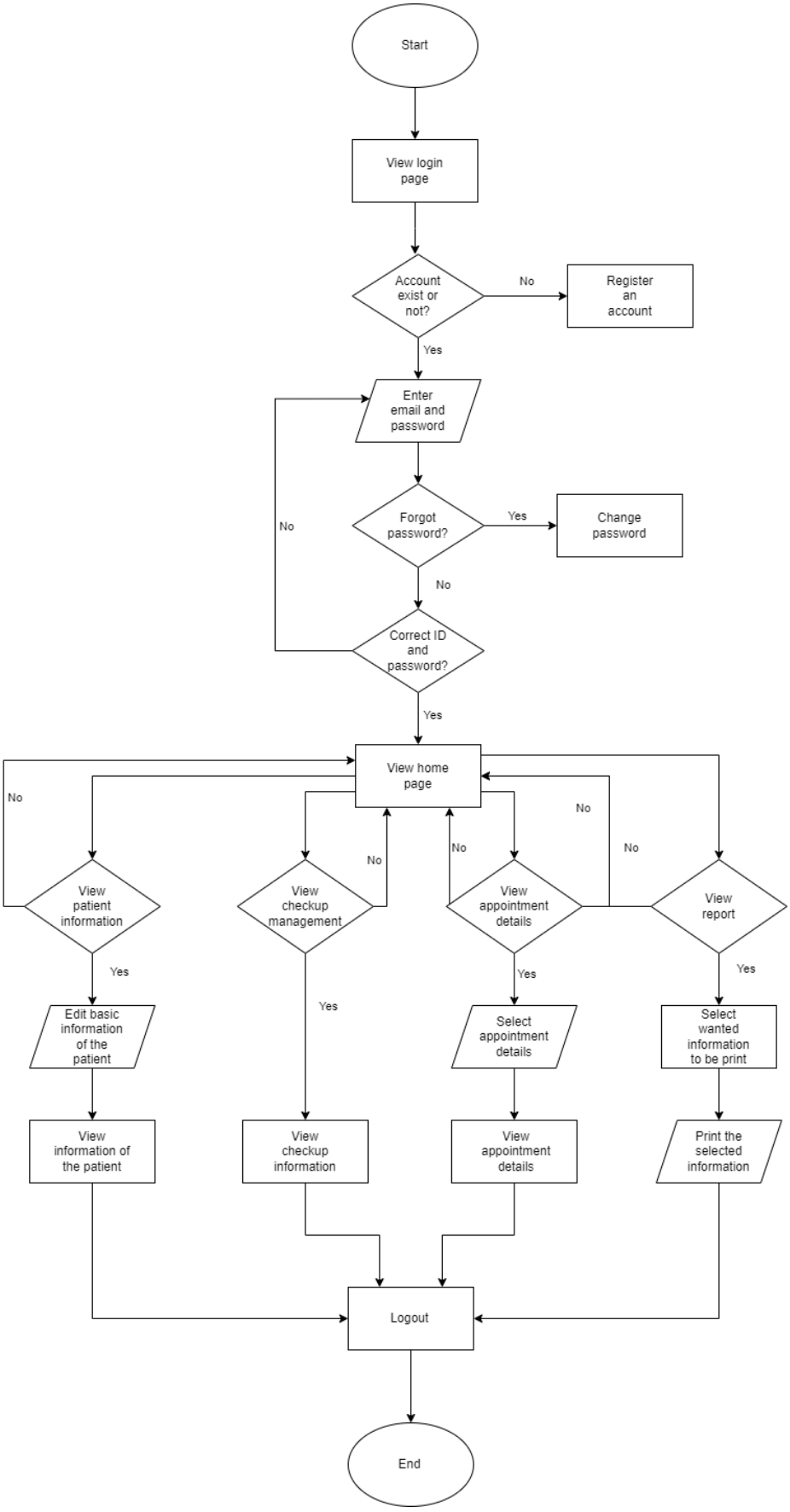


Flowchart Diagram for Doctor Side

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 673-691



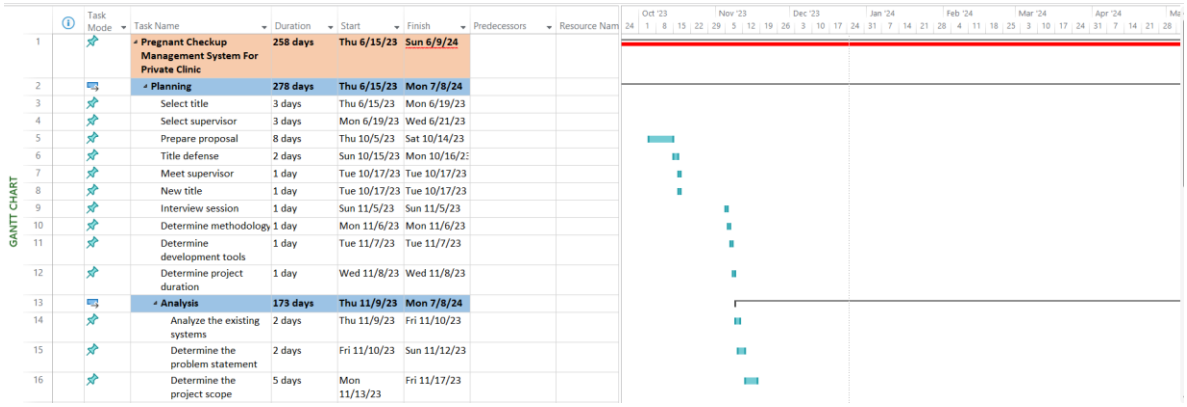
Flowchart Diagram for Nurse Diagram



Flowchart Diagram for Patient Diagram

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 673-691

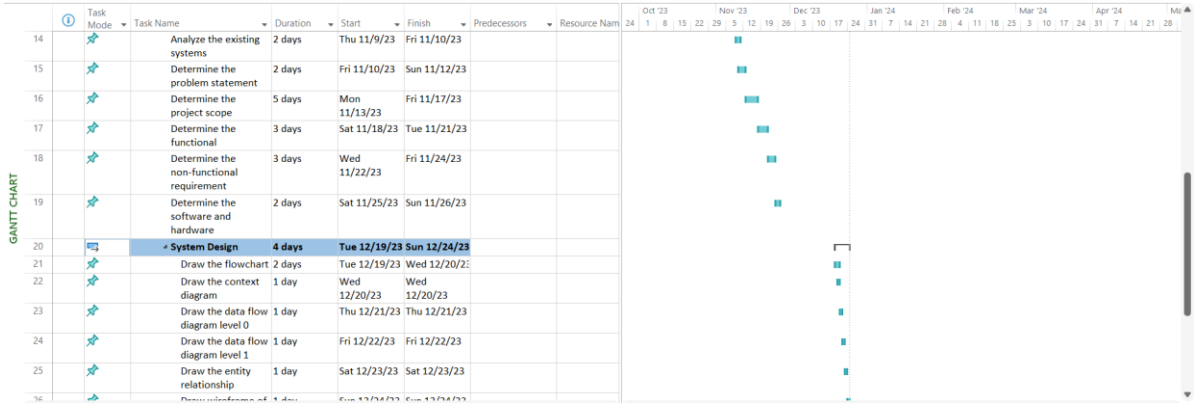
Appendix D: Gantt Chart



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Nam
1		**Pregnant Checkup Management System For Private Clinic**	**258 days**	**Thu 6/15/23**	**Sun 6/9/24**		
2		**Planning**	**278 days**	**Thu 6/15/23**	**Mon 7/8/24**		
3		Select title	3 days	Thu 6/15/23	Mon 6/19/23		
4		Select supervisor	3 days	Mon 6/19/23	Wed 6/21/23		
5		Prepare proposal	8 days	Thu 10/5/23	Sat 10/14/23		
6		Title defense	2 days	Sun 10/15/23	Mon 10/16/2		
7		Meet supervisor	1 day	Tue 10/17/23	Tue 10/17/23		
8		New title	1 day	Tue 10/17/23	Tue 10/17/23		
9		Interview session	1 day	Sun 11/5/23	Sun 11/5/23		
10		Determine methodology	1 day	Mon 11/6/23	Mon 11/6/23		
11		Determine development tools	1 day	Tue 11/7/23	Tue 11/7/23		
12		Determine project duration	1 day	Wed 11/8/23	Wed 11/8/23		
13		**Analysis**	**173 days**	**Thu 11/9/23**	**Mon 7/8/24**		
14		Analyze the existing systems	2 days	Thu 11/9/23	Fri 11/10/23		
15		Determine the problem statement	2 days	Fri 11/10/23	Sun 11/12/23		
16		Determine the project scope	5 days	Mon 11/13/23	Fri 11/17/23		

GANTT CHART

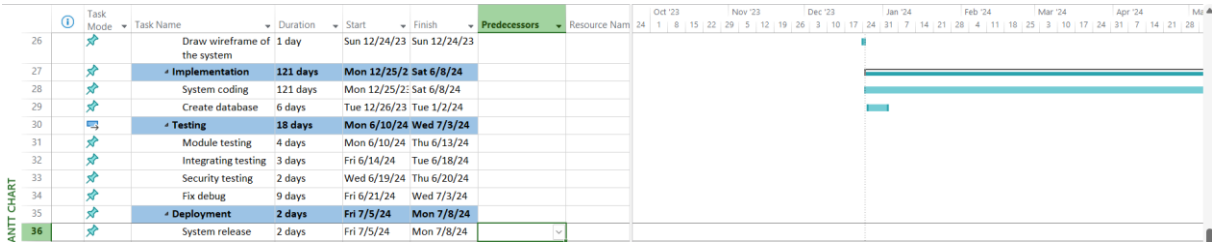
Gantt Chart



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Nam
14		Analyze the existing systems	2 days	Thu 11/9/23	Fri 11/10/23		
15		Determine the problem statement	2 days	Fri 11/10/23	Sun 11/12/23		
16		Determine the project scope	5 days	Mon 11/13/23	Fri 11/17/23		
17		Determine the functional	3 days	Sat 11/18/23	Tue 11/21/23		
18		Determine the non-functional requirement	3 days	Wed 11/22/23	Fri 11/24/23		
19		Determine the software and hardware	2 days	Sat 11/25/23	Sun 11/26/23		
20		**System Design**	**4 days**	**Tue 12/19/23**	**Sun 12/24/23**		
21		Draw the flowchart	2 days	Tue 12/19/23	Wed 12/20/2		
22		Draw the context diagram	1 day	Wed 12/20/23	Wed 12/20/23		
23		Draw the data flow diagram level 0	1 day	Thu 12/21/23	Thu 12/21/23		
24		Draw the data flow diagram level 1	1 day	Fri 12/22/23	Fri 12/22/23		
25		Draw the entity relationship	1 day	Sat 12/23/23	Sat 12/23/23		

GANTT CHART

Gantt Chart (cont.)



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Nam
26		Draw wireframe of the system	1 day	Sun 12/24/23	Sun 12/24/23		
27		**Implementation**	**121 days**	**Mon 12/25/2**	**Sat 6/8/24**		
28		System coding	121 days	Mon 12/25/2	Sat 6/8/24		
29		Create database	6 days	Tue 12/26/23	Tue 1/2/24		
30		**Testing**	**18 days**	**Mon 6/10/24**	**Wed 7/3/24**		
31		Module testing	4 days	Mon 6/10/24	Thu 6/13/24		
32		Integrating testing	3 days	Fri 6/14/24	Tue 6/18/24		
33		Security testing	2 days	Wed 6/19/24	Thu 6/20/24		
34		Fix debug	9 days	Fri 6/21/24	Wed 7/3/24		
35		**Deployment**	**2 days**	**Fri 7/5/24**	**Mon 7/8/24**		
36		System release	2 days	Fri 7/5/24	Mon 7/8/24		

ANTT CHART

Gantt Chart (cont.)