TECHNICAL MANUAL

Student Number: **20750181**

Year: **2025-2026**

Purpose: to enable future developers to understand the QMC system architecture, database design, core logic, and installation procedures necessary for future maintenance and feature expansion.

COMP4039 - Coursework

1. Installation Guide

This section describes the required procedure to install and run the system. The system requires Docker and Docker Compose to be installed. No additional configuration of PHP, Apache, or MySQL is required for the installation.

Installation should be performed using the provided Docker environment within the directory. The docker configuration mirrors the environment used during development and so ensures consistency with the intended design and functionality of the system. Ensure the directory structure remains unchanged, as the Docker configuration mounts the html/cw/ directory and the mariadb-data/ directory directly into the container.

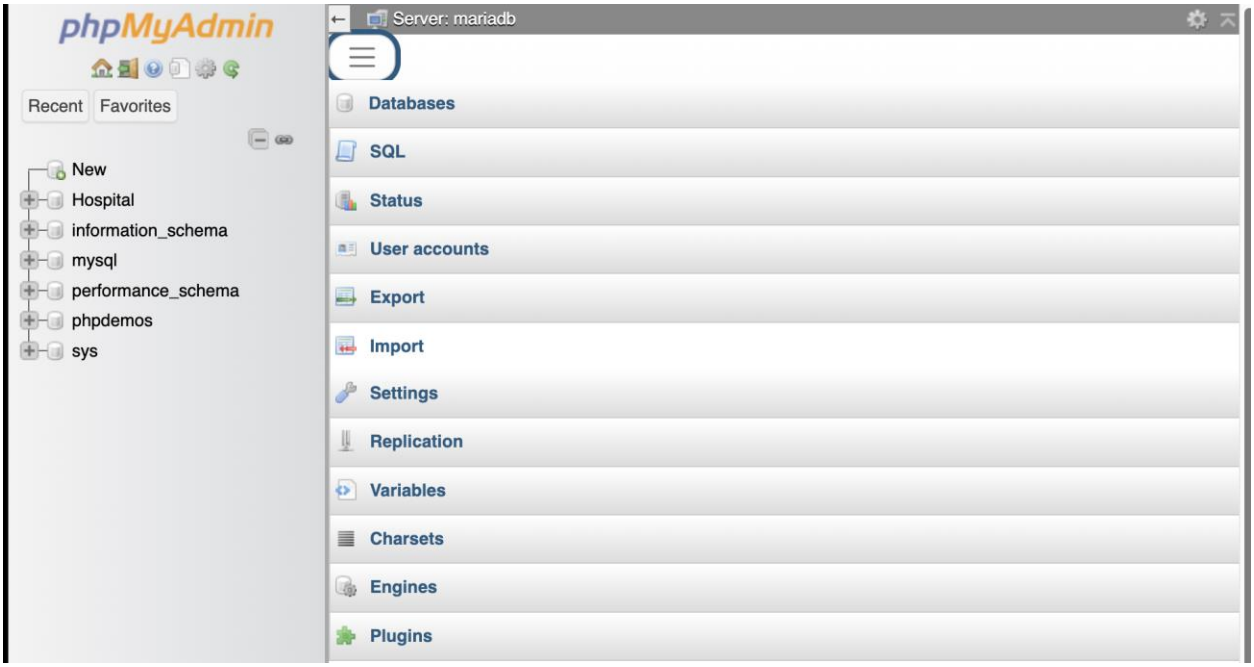
1.1 Installation Procedure

1. Open a terminal window and navigate to the project directory by entering the command *'cd ../COMP4039-CW-psxmj4-20750191'.* From here you then want to type the command *'cd psxmj4-20750181-Docker'.*

2. From inside this Docker directory, build and initialise the Docker environment by executing the command *'docker-compose up --build'* within the terminal.

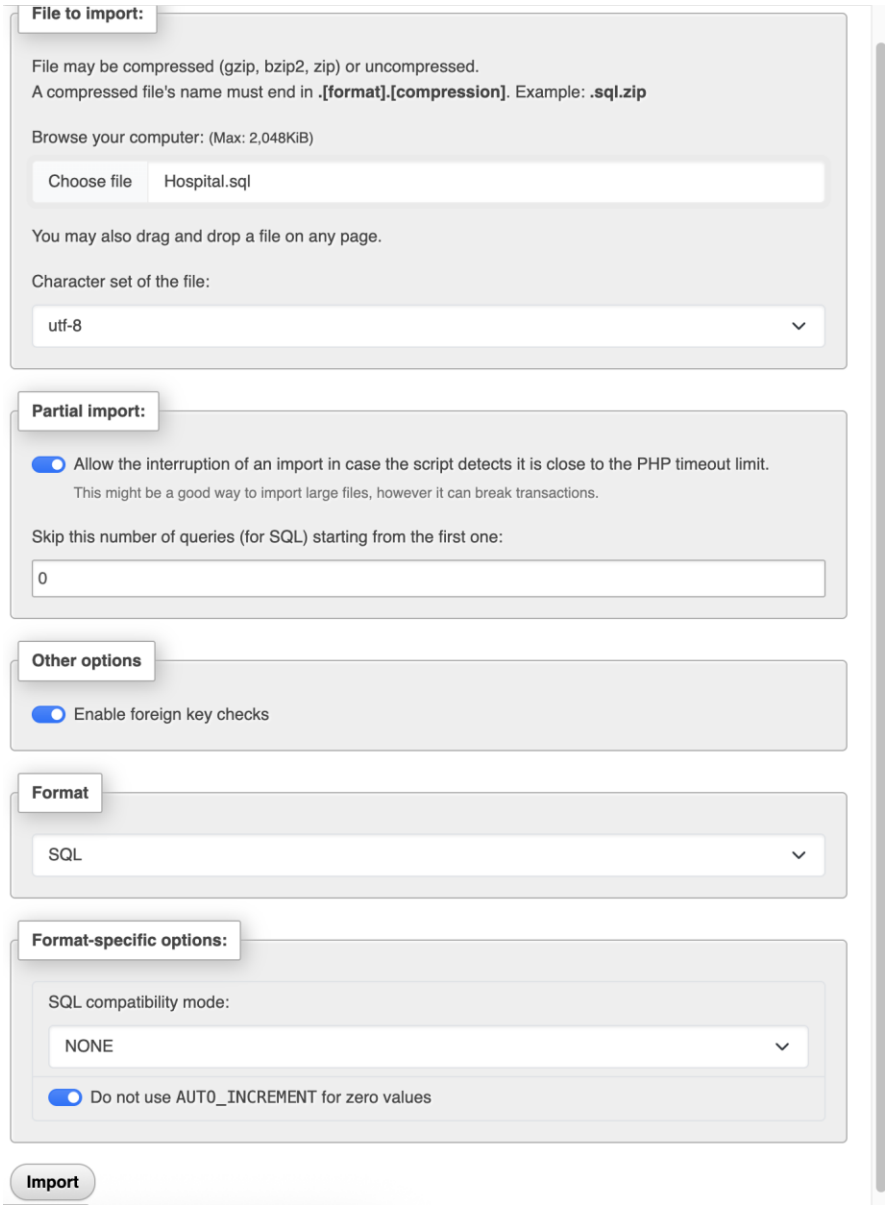
3. Once running, open a browser and go to *http://localhost:8081* which will open phpMyAdmin.

4. Withinp phpMyAdmin open the Database dropdown menu. Select ‘Import’ from the navigation bar.



(Step 4 – navigate to Import in phpMyAdmin)

5. Click ‘choose file’, navigate to the project folder, open the mariadb directory and select ‘Hospital.sql’. Click import at the bottom of the page. This loads the full database required by the system.



(Step 5 – Import 'Hospital.sql')

6. From here navigate to http:.//localhost/cw/index.php entry point, a developer can log in using the provided administrative credentials (**username**: jelina, **password:** iron99) to access all administrator-level functionality implemented within the system. This allows the developer to verify core features such as user management, parking permit approvals, and auditing tools.

7. To stop the system, run '*docker-compose down'* in the terminal from the same directory. This will shut down the containers without deleting database data.

2. System Architecture

The QMC system is structured using a Three-Tier Architecture, separating the presentation, application, and data layers. This separation is enforced through deliberate directory organisation. Frontend (presentation) files live in the root directory, while the essential system files (like db_connection.php and session_security.php) are grouped within the /includes folder. The core application logic is strictly contained within the /data_access layer, which houses all Data Access Object (DAO) files (e.g., DoctorDAO.php, PatientDAO.php, AuditDAO.php). Within each DAO file is a series of global functions which can be used across all files, by calling the file at the top of the new script using *require_once 'FILENAME.php';*. Global functions enable for cleaner, and more interpretable code, and so should continue to made use of and added to.

The system follows a Three-Tier Architecture because it enables faster and more modular development, supports scalability, and strengthens security by preventing direct communication between the presentation and data layers. This architectural pattern must be preserved in any future maintenance or feature expansion.

Inline comments are included throughout the codebase to support developers in understanding the purpose and behaviour of each component.

When creating a new page, it is essential to include *'includes/db_connection.php'* at the top of the file if the page will interact with the Hospital database. You must also import *'includes/session.php'* and call '*require_login();'* to ensure that only authenticated users can access the page. Because the system handles confidential patient and staff information, this access control is a critical security requirement. The session logic also enforces automatic logout after 15 minutes of inactivity; this timeout can be adjusted within session.php.

The includes directory also contains '*header.php*' and '*footer.php*', which should be included on every new page. These files provide consistent navigation and maintain the system's user-friendly interface.

GitHub served as the system's version-control backbone, providing structured commit tracking, branch management, and a safeguarded audit trail of all code changes. Maintaining this workflow is strongly recommended, as it preserves system integrity, reduces the risk of regressions, and ensures re coverability in the event of data corruption or development errors.

The flowchart illustrates the high-level system architecture and primary navigation pathways between PHP components. It models the application's control flow from the entry point (index.php) through the authenticated dashboard (home.php) and into the major functional modules

(Figure 1. System architecture flow diagram)

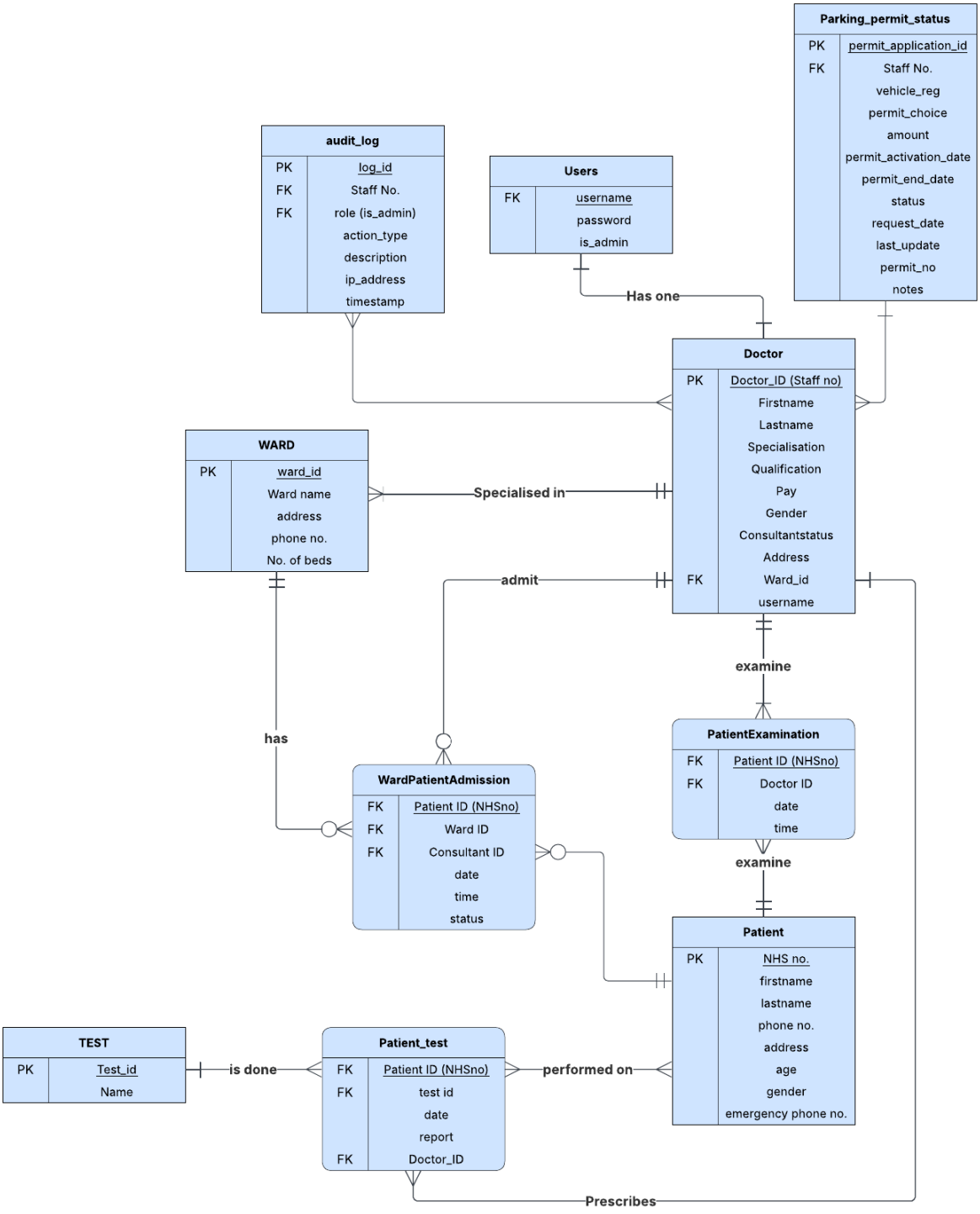
3. Database Architecture

The Entity–Relationship Diagram (ERD) presented in this report illustrates the complete database structure underpinning the QMC system. It captures all core entities including **Patient**, **Doctor**, **Users**, **Ward**, **PatientExamination**, **Patient_test**, **WardPatientAdmission**, **Parking_permit_status**, **audit_log**, and **TEST,** and shows how these components interact to support clinical, administrative, and auditing functionality.

When implementing new functionality, developers must consult the ERD to ensure that any additional data aligns with the existing schema, preserves normalisation standards, and avoids introducing redundant or inefficient data structures. This also ensures that referential integrity is

maintained across all relationships, and that new features integrate cleanly with the system's established constraints and data flows.

Diagram below.



(Figure 2. Entity Relationship Diagram)