Hospital Management System

Project Kickoff and Planning Report

Tanya Mistry Krishang Shah

October 25, 2025

1. Project Kickoff

Goals and Expected Outcomes

Our goal is to build a small but functional Hospital Management System (HMS) using Python, Tkinter, and MySQL. The system supports three roles - Patient, Doctor, and Admin each with clear permissions: patients book appointments, doctors manage prescriptions, and admins handle billing. Expected outcomes include a clean desktop interface, a normalized database, and a working demo showing smooth interaction between all roles.

Defining the Project Scope

In Scope:

- Role based login and authentication.
- Appointment creation, approval, and completion.
- Prescription records and simple billing.

Out of Scope:

- Multi-hospital setup, inventory, or insurance modules.
- Notifications and analytics features.

The focus is to keep it practical, local, and easy to maintain.

Deliverables and Milestones

- **Phase 1 - Database:** schema and seed data.
- **Phase 2 - UI:** login, dashboards, and navigation.
- **Phase 3 - Core Logic:** CRUD for appointments, prescriptions, bills.
- **Phase 4 - Testing:** validation and documentation.

Tentative Timeline:

- **Nov 1:** Database schema finalized and tested.
- **Nov 10:** Login and basic dashboards complete.
- **Nov 18:** CRUD operations integrated and functional.
- **Nov 24:** Final testing, polish, and documentation submission.

Capabilities and Gaps

Tanya focuses on SQL design and Krishang handles Python and UI. Both know Python + SQL, but will learn Tkinter layouts and password hashing early. No external dataset is needed, dummy entries will be seeded manually.

2. Team Discussions

Core Skills and Roles

Tanya Mistry - Database Lead: Creates schema, writes SQL scripts, defines relationships, and maintains consistency.

Krishang Shah - Application Lead: Builds Tkinter screens, connects UI to MySQL, validates inputs, and manages role based logic.

Collaboration Plan

Tanya's database will serve as the foundation for Krishang's interface. Integration happens incrementally once each module is stable, it's linked and tested together. Weekly check-ins keep progress aligned and bugs tracked.

Challenges and Learning Goals

Neither member has built a full GUI-database app before, so key learning areas are: form validation, SQL parameterization, and Tkinter table displays. Small prototypes will be tested before merging into the final build.

Tools and Technologies

Languages: Python 3.11, SQL (MySQL 8). **Libraries:** `mysql-connector-python`. **Tools:** VS Code, MySQL Workbench, Git. This stack is lightweight, offline friendly, and easy for both members to use.

3. Skills and Tools Assessment

Resources and Frameworks

We'll rely on documentation and tutorials, no external mentorship required. Key components:

- `mysql-connector-python` - database connectivity.
- Tkinter's `Treeview` - data tables in the UI.

Proficiency and Workflow

Both members will mirror the same environment and test on identical data. A shared checklist and brief weekly review keep progress consistent. Version control will ensure that database and application changes stay in sync.

Task Assignment

- Tanya - schema, seed data, SQL views.
- Krishang - UI logic, CRUD operations, authentication.
- Joint - integration testing and report writing.

4. Initial Setup

Environment

- **Software:** Python 3.11, MySQL 8, MySQL Workbench.
- **Libraries:** `mysql-connector-python`.

The database runs locally, and connection details are stored in a single config file for both systems.

Version Control

A GitHub repository (private) will include:

- `/db` - schema & seed scripts.
- `/app` - Python UI & controllers.
- `README.md` - setup guide.

Feature branches are merged only after review to avoid conflicts.

Current Progress

Database tables and relationships are complete; connection tests from Python succeed. The login screen is functional. Next steps: appointment booking, doctor prescriptions, and admin billing.

Conclusion

This project combines database design and Python programming into a practical hospital workflow system. With a clear scope and complementary skills, we can deliver a stable, well-documented prototype. Future extensions may include automated notifications or a web interface, but this version focuses on a reliable desktop HMS demonstrating sound engineering and teamwork.
GitHub Repository: https://github.com/tanyamistry/DS5110-Project-HMS