**Healthpost Database Management System**

Rajesh Sunar

Shobha Adhikari

Fundamentals of Programming with C

Jan 5, 2025

**Healthpost Database Management System**

**Executive Summary**

The Healthpost Database Management System (HDMS) is a project aimed at developing a C-program-based system to manage and streamline healthcare services at a local health post. The system will include functionalities for managing patient records, appointments, health worker data, prescriptions, medications, and medical reports. By implementing this system, we aim to reduce manual processes, enhance data accessibility, and improve the efficiency of healthcare services.

**Project Background**

**Problem Statement:**

Local health posts often face challenges in manually managing large volumes of data in rural places. Issues such as misplaced patient records, inefficient appointment scheduling, and difficulty tracking medication stock can lead to delays and reduced quality of care.

**Motivation:**

During community work, our friend discovered that most of Nepal's healthcare posts need better data filing systems in the village. They need to record all the required information about the medicines provided to patients or the diseases being treated. Patient records may be stored in various places, making it difficult to retrieve information efficiently. This can lead to errors in patient care, especially in emergencies.

Given the manual nature of record-keeping, there is a significant risk of records being lost or destroyed due to natural disasters, accidents, or poor storage conditions. This is particularly a problem in areas prone to floods, landslides, and other natural disasters. To resolve these issues, we would like to create a useful database management system for the Bhotang health post. **Project Objectives**

1. Specific: Develop a C-programming-based HealthPost Database Management System to handle key entities such as patients and health workers.

2. Measurable: Ensure the system can store and retrieve data for at least 100 patients and 10 health workers.

3. Achievable: Implement core features such as adding, updating, and querying data using basic C programming techniques.

4. Relevant: Address existing inefficiencies in healthpost data management processes.

5. Timebound: Complete the project within the dateline.

**Project Scope**

Included:

* Management of patient information (e.g., name, DOB, gender, contact).
* Health worker data management (e.g., name, specialization, contact).

**Excluded:**

* Appointment scheduling and tracking.
* Medication inventory management.
* Integration with external systems (e.g., hospital databases).
* Advanced analytics or reporting features.

**Methodology**

**Approach:**

* Analyze and design the database structure based on the ER diagram provided.
* Write modular C-programming code to handle CRUD (Create, Read, Update, Delete) operations for each entity.

**Tools and Technologies:**

Compiler: GCC, G++

Code Editor: VSCode

Version Control: GitHub for collaboration and version tracking.

**Techniques:**

* Arrays for data storage.
* Functions for modular coding.
* Loops and conditionals for logical operations.
* File handling for data persistence.

**Deliverables**

1. Functional C-programming code for the Healthpost Database Management System.

2. User documentation explaining how to use the system.

3. A presentation summarizing the project outcomes.

4. Source code hosted on GitHub for review and evaluation.

**Timeline**

**Week1**

Project Planning and Setup

* Finalize project requirements and scope
* Set up the development environment
* Plan and outline the modular code structure for managing entities
* Create the base structure of the program.

**Week 2**

Core Functionalities Development

* Develop functions for:
* Patients and Health worker data management.
* Implement CRD (Create, Read, Delete) operations for the above entities.
* Test individual modules for functionality and correctness.
* Start implementing file handling for persistent data storage.

**Week 3**

Extended Features and Testing

* Refine and optimize the code for better performance.
* Debug and resolve any issues in file handling and data relationships.
* Conduct module integration testing to ensure smooth interaction between all system components.

**Week 4**

Documentation, Presentation, and Finalization

* Finalize and polish the code.
* Prepare the project presentation:
* Summarize project outcomes.
* Highlight features and functionality.
* Host the source code on GitHub.
* Conduct final testing to ensure all functionalities work seamlessly.
* Submit/Present the completed project and documentation.

**Expected Challenges**

1. Data Storage Constraints: Limited capacity of arrays may require optimized storage solutions.

2. Debugging Complexity: ensuring all relationships between entities work as intended.

3. File Handling: Implementing efficient file I/O operations for data persistence.

4. Time Management: Balancing the project timeline with academic obligations.

**Conclusion**

The Healthpost Database Management System project aims to provide a reliable and efficient solution for managing healthcare data in local health posts. By leveraging C programming, the system will streamline operations, reduce manual errors, and enhance overall service delivery. This project is a practical application of programming concepts and demonstrates the potential of technology in improving healthcare services.