

BAHRIA UNIVERSITY KARACHI CAMPUS
DEPARTMENT OF COMPUTER SCIENCE



Bahria University
Discovering Knowledge

Database Administration & Management
(ITC-424)

TOPIC NAME

Elderly CARE HOME MANAGEMENT

Submitted by:

NAME: Madeeha Talib (02-235221-010)

NAME: Arjan Kumar (02-235221-024)

NAME: Hussnain Khalid (02-235221-041)

Submitted To:

Engr. Gul Saba

Submitted On:

Date: 03 June 2025

ACKNOWLEDGE

I am deeply thankful to my respected instructor and mentor for their constant support and guidance throughout the development of this project. Their valuable feedback and encouragement inspired me to explore deeper into database administration concepts and apply them effectively. I would also like to thank my peers for their continuous encouragement and the collaborative environment they fostered during this project.

ABSTRACT

The Elder Care Home Management System is a comprehensive platform designed to automate and streamline the essential functions of elder care facilities. With the rising elderly population and the increasing need for quality care, this system ensures efficient management of residents, rooms, medical records, and visitor scheduling. Developed using PHP and MySQL, it incorporates advanced Database Administration (DBA) concepts like ETL, horizontal fragmentation, stored procedures, and materialized views to deliver optimized performance and data accuracy. The system not only simplifies administrative tasks but also provides scalable and reliable infrastructure suitable for institutional adoption.

Table of Contents REPORT:

INTRODUCTION	5
PROBLEM STATEMENT.....	5
EXISTING SYSTEMS	6
OBJECTIVES AND GOALS	6
PROJECT SCOPE	6
WORKFLOW (UML REPRESENTATION)	7
OVERVIEW OF PROJECT.....	7
TOOLS AND TECHNOLOGIES.....	8
PROJECT FEATURES	8
FUNCTIONAL REQUIREMENTS:.....	8
NON-FUNCATIONAL REQUIREMENTS:	8
MAD IMPLEMENTED CONCEPTS	9
OUTPUT	9
FUTURE WORK.....	12
CONCLUSION	12
REFERENCES	12

INTRODUCTION

The Elder Care Home Management System is a comprehensive, database-driven application tailored to manage the daily operations of a residential care facility for the elderly. These operations include resident management, room assignment, medical scheduling, visitor tracking, and administrative oversight. With the rise in the global elderly population, the need for efficient, secure, and scalable care home systems is more urgent than ever. This project addresses core management functions while integrating advanced Database Administration (DBA) concepts to ensure optimal performance, data integrity, and future scalability[1].

This system eliminates manual and inefficient processes by offering a centralized platform with a user-friendly web interface. It enables staff to interact easily with resident data and empowers administrators with advanced backend tools, enhancing both operational efficiency and data reliability.

PROBLEM STATEMENT

Managing an elder care home involves complex tasks that require precision and sensitivity. Without a centralized digital system, care homes commonly encounter the following problems:

- Data inconsistency due to manual record-keeping
- Inefficient room allocation and status tracking
- Absence of resident summaries and insights
- Limited data import/export capabilities
- No support for data fragmentation and performance optimization

These issues lead to operational delays, miscommunication, and reduced care quality. A robust, integrated system with both management and administrative features is essential for smooth operation and data-driven decision-making.

EXISTING SYSTEMS

Many care homes still rely on outdated tools such as spreadsheets or manual logs. Some use basic digital solutions, but these are often limited to admissions and billing. Common shortcomings of these systems include:

- No real-time updates or data synchronization
- Lack of automated summaries and analytics
- Difficulty handling large datasets during import/export
- Poor user interface design
- No implementation of DBA techniques like fragmentation or materialized views

This project addresses these limitations by introducing modern database features, intuitive design, and modular architecture[2].

OBJECTIVES AND GOALS

- **Develop a centralized system** to manage residents, rooms, and medical information efficiently.
- **Incorporate DBA concepts** such as ETL, stored procedures, and partitioning to enhance backend operations.
- **Enable CSV-based import/export** for better data handling and portability.
- **Provide data summaries** using materialized views and aggregate queries.
- **Enhance usability** through intuitive design and structured data visibility.
- **Ensure scalability and modularity** for future extensions and improvements.

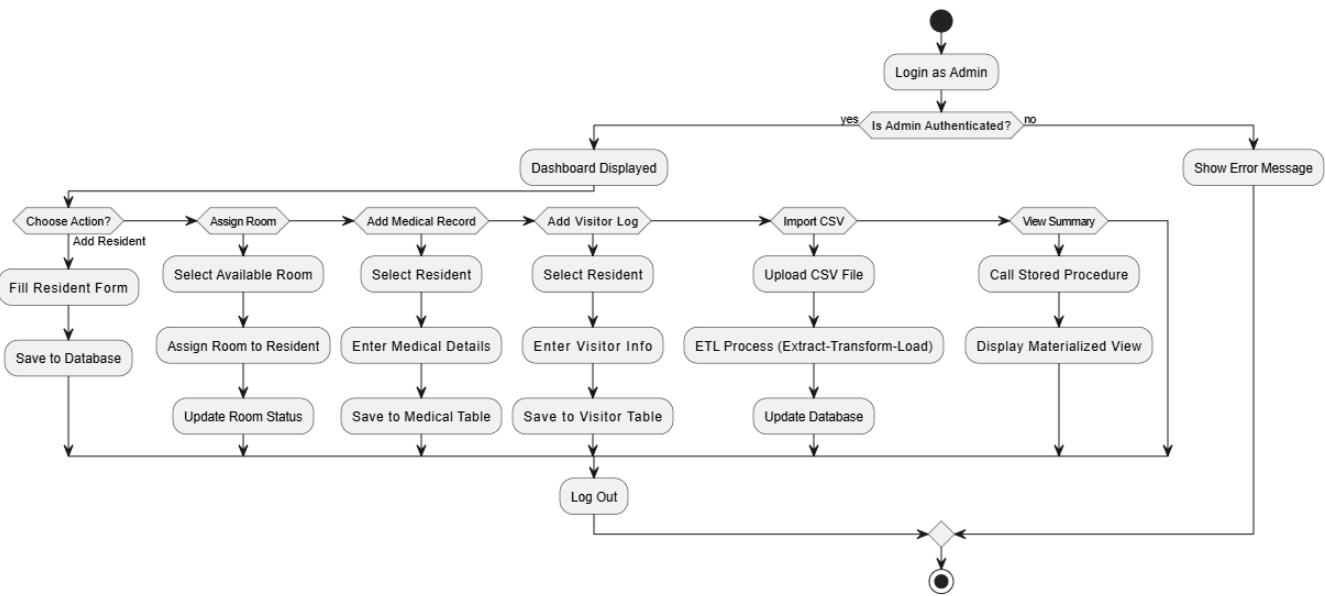
PROJECT SCOPE

This project focuses on automating and streamlining the operations of an elder care facility with a strong MySQL-powered backend. Core functionalities include:

- Resident management (admissions, edits, deletions)
- Room assignment with real-time occupancy status
- Medical records and medication history tracking
- Visitor logs and scheduling
- Admin interface with advanced DBA utilities

Implemented DBA techniques include ETL (CSV handling), horizontal fragmentation, stored procedures, materialized views, and summarized queries to create a balance between usability and backend efficiency.

WORKFLOW (UML REPRESENTATION)



OVERVIEW OF PROJECT

The system is modular, consisting of the following components:

1. **Resident Management Module:** CRUD operations for residents.
2. **Room Management Module:** View and manage room assignments using floor-wise (partitioned) data.
3. **Medical Records Module:** Maintain logs of medical visits and medications.
4. **Visitor Management Module:** Schedule and log family or friend visits.
5. **Admin Panel:** Dashboard for data tools including CSV import/export, partition views, and stored procedure execution.

All modules are connected via a centralized MySQL database.

TOOLS AND TECHNOLOGIES

- **Frontend:** HTML5, CSS3, Bootstrap 5[3]
- **Backend:** PHP 8+
- **Database:** MySQL 5.7/8.0
- **Web Server:** XAMPP (Apache + MySQL)
- **IDE:** Visual Studio Code
- **DBA Concepts Used:** ETL, Fragmentation, Stored Procedures, Materialized Views, UNION/Partition Queries

PROJECT FEATURES

- Resident form with dynamic inputs
- Floor-wise room assignment via partitioned data
- Medical record entry and updates
- Visitor log creation and scheduling
- Summary reports via materialized views
- CSV-based data import/export
- Manual refresh of summaries using stored procedures
- Admin dashboard with DBA utilities

FUNCTIONAL REQUIREMENTS:

- FR1: Admin can manage (add/edit/delete) residents
- FR2: Room availability shown by floor (partitioned view)
- FR3: CSV import to update resident records
- FR4: Export data to CSV for backups
- FR5: Summary generation using materialized view
- FR6: Record and update visitor information

NON-FUNCTIONAL REQUIREMENTS:

- NFR1: Fast response times (within 2 seconds)
- NFR2: Ensured data integrity via validations and transactions
- NFR3: Responsive design for multiple devices
- NFR4: Secure actions with confirmations and role-based tools
- NFR5: Modular coding structure for easy scalability

IMPLEMENTED DBA CONCEPTS

1. **Relational Set Operators:** UNION ALL used to combine room data from multiple floors
2. **Fragmentation:** Horizontal partitioning of room data into separate tables by floor
3. **ETL:** Data loading/unloading via CSV files using PHP scripts
4. **Stored Procedures:** Function RefreshResidentSummary() to update summary data
5. **Materialized Views:** Precomputed summary table for resident statistics
6. **Summarized Queries:** Aggregate SQL used for displaying dashboard data
7. **Simulated Partitioned Tables:** rooms_floor1, rooms_floor2 simulate actual partitioning

OUTPUT

Main Dashboard:

The dashboard displays the following information:

- Recent Admissions: 4 Currently in care
- Medication Alerts: 2 Medications Due
- Rooms Occupied: 1/50 Rooms Occupied
- Scheduled Visits: 1 Scheduled Visits

Recent Admissions table:

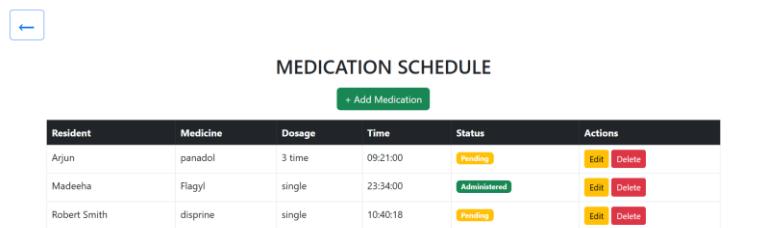
Name	Age	Gender	Admission Date	Status
Madeeha	22	Female	2025-06-03	Active
Arjun	22	Male	2025-06-03	Active
Margaret Johnson	26	Female	2023-05-15	Active
Robert Smith	24	Male	2023-05-10	Active

Resident records:

The Resident records page shows the following table:

ID	Name	Age	Gender	Admission	Status	Actions
5	Arjun	22	Male	2025-06-03	Active	<button>Edit</button> <button>Delete</button>
4	Madeeha	22	Female	2025-06-03	Active	<button>Edit</button> <button>Delete</button>
2	Robert Smith	24	Male	2023-05-10	Active	<button>Edit</button> <button>Delete</button>
1	Margaret Johnson	26	Female	2023-05-15	Active	<button>Edit</button> <button>Delete</button>

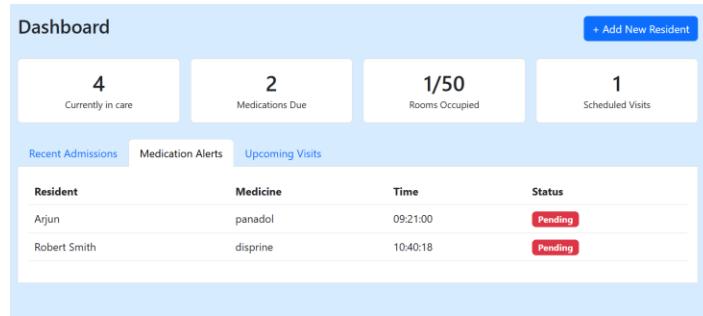
Each resident taking which medication:



A screenshot of a web-based medication scheduling application. At the top, there's a header "MEDICATION SCHEDULE" with a back arrow icon and a green button "+ Add Medication". Below the header is a table with columns: Resident, Medicine, Dosage, Time, Status, and Actions. The table contains three rows of data:

Resident	Medicine	Dosage	Time	Status	Actions
Arjun	panadol	3 time	09:21:00	Pending	<button>Edit</button> <button>Delete</button>
Madeeha	Flagyl	single	23:34:00	Administered	<button>Edit</button> <button>Delete</button>
Robert Smith	disprine	single	10:40:18	Pending	<button>Edit</button> <button>Delete</button>

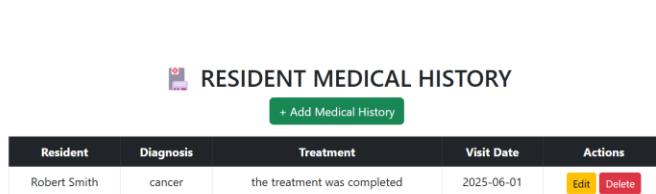
Medication alters:



A screenshot of a dashboard interface. At the top, it says "Dashboard" and has a blue button "+ Add New Resident". Below are four summary boxes: "4 Currently in care", "2 Medications Due", "1/50 Rooms Occupied", and "1 Scheduled Visits". Underneath these are three tabs: "Recent Admissions" (selected), "Medication Alerts", and "Upcoming Visits". A table below shows resident information:

Resident	Medicine	Time	Status
Arjun	panadol	09:21:00	Pending
Robert Smith	disprine	10:40:18	Pending

Medical history:



A screenshot of a medical history interface. At the top, there's a header "RESIDENT MEDICAL HISTORY" with a back arrow icon and a green button "+ Add Medical History". Below the header is a table with columns: Resident, Diagnosis, Treatment, Visit Date, and Actions. The table contains one row of data:

Resident	Diagnosis	Treatment	Visit Date	Actions
Robert Smith	cancer	the treatment was completed	2025-06-01	<button>Edit</button> <button>Delete</button>

Dietary plans:



A screenshot of a food preference interface. At the top, there's a header "RESIDENT FOOD PREFERENCES" with a back arrow icon and a green button "+ Add Food Preference". Below the header is a table with columns: Resident, Preference, Notes, and Actions. The table contains one row of data:

Resident	Preference	Notes	Actions
Margaret Johnson	daal chawal	take daal chawal right after medicine	<button>Edit</button> <button>Delete</button>

Room management:



A screenshot of a room management interface. At the top, there's a header "ROOM MANAGEMENT" with a back arrow icon and a green button "+ Add Room". Below the header is a table with columns: Room No, Floor, Capacity, Occupied, Resident Name, and Actions. The table contains two rows of data:

Room No	Floor	Capacity	Occupied	Resident Name	Actions
1	1	2	3	Robert Smith	<button>Assign</button> <button>Delete</button>
2	1	3	1	Madeeha	<button>Assign</button> <button>Delete</button>

Family visits:

FAMILY VISIT SCHEDULE

+ Add Visit

Resident	Visitor	Relationship	Visit Time	Status	Actions
Margaret Johnson	hills	father	2025-06-03 06:30:26	Confirmed	<button>Edit</button> <button>Delete</button>
Madeeha	raazia	sister	2025-06-03 11:30:00	Pending	<button>Edit</button> <button>Delete</button>

Dashboard

+ Add New Resident

4 Currently in care 2 Medications Due 1/50 Rooms Occupied 2 Scheduled Visits

Recent Admissions Medication Alerts Upcoming Visits

Resident	Visitor	Relationship	Scheduled	Status
Madeeha	raazia	sister	2025-06-03 11:30:00	Pending

Reports:

Resident Summary Report

Summary refreshed successfully!

Resident	Total Visits	Total Medications
Arjun	0	1
Madeeha	1	1
Margaret Johnson	1	0
Robert Smith	0	1

Database admin:

Database Admin Tools

Export Residents: Download all residents as a CSV file. Export Now

Import Residents: Upload a CSV file to import resident records. Import Now

Refresh Resident Summary: Recompute materialized view with latest data. Refresh Summary

View Partitioned Room Data: Shows horizontal fragmentation (floor-wise).

Room	Floor	Capacity	Occupied
1	1	2	1
2	1	3	0
1	1	2	1
2	1	3	0

Resident Summary Report: View precomputed stats from materialized view. Open Report

Summary refreshed successfully!

Audits logs:

Audit Log Viewer

ID	Action	Table	Record ID	User	Time
10	INSERT	residents	5	Admin	2025-06-03 09:17:12
9	UPDATE	residents	1	Admin	2025-06-03 09:16:17
8	UPDATE	residents	4	Admin	2025-06-03 00:10:22
7	UPDATE	residents	4	Admin	2025-06-03 00:10:14
6	INSERT	residents	4	Admin	2025-06-02 22:59:34
5	DELETE	residents	3	Admin	2025-06-02 21:49:13
4	INSERT	residents	3	Admin	2025-06-02 21:49:08
3	UPDATE	residents	2	Admin	2025-06-02 21:42:43
2	UPDATE	residents	2	Admin	2025-06-02 18:28:48
1	UPDATE	residents	1	Admin	2025-06-02 18:28:31

FUTURE WORK

- Add authentication and user role-based access
- Implement MySQL native partitioning techniques
- Develop real-time room vacancy monitoring
- Integrate data visualization (charts and graphs)
- Add audit logs using database triggers
- Enable cloud backup and remote database restore

CONCLUSION

The Elder Care Home Management System provides a full-stack solution to streamline elderly care facility management. By integrating modern DBA practices such as ETL, fragmentation, stored procedures, and materialized views, it ensures both operational excellence and backend efficiency. The project demonstrates a real-world application of advanced database principles and presents a solid foundation for further development or institutional deployment.

REFERENCES

- [1] Oracle Corporation, “MySQL 8.0 Reference Manual,” [Online]. Available: <https://dev.mysql.com/doc>. [Accessed: Jun. 3, 2025].
- [2] The PHP Group, “PHP Manual,” [Online]. Available: <https://www.php.net/manual/en/>. [Accessed: Jun. 3, 2025].
- [3] W3Schools, “PHP and MySQL Tutorial,” [Online]. Available: <https://www.w3schools.com>. [Accessed: Jun. 3, 2025].
- [4] The Bootstrap Team, “Bootstrap Documentation,” [Online]. Available: <https://getbootstrap.com>. [Accessed: Jun. 3, 2025].