PROJECT TITLE: "MAIDS ON CALL SYSTEM"

Jackson Karieri Maina - HDB312-D1-1074/2022

Jomo Kenyatta University of Agriculture and Technology

HCBT 3204: Business Systems Project

Business Systems Project Report

Dr. Dennis Kaburu (PhD)

May - August, 2025

Contents

CHAPTER 1

INTRODUCTION

1.0 Abstract

The Maids On Call System is an innovative online maid management platform designed to streamline the processes of service booking, maid allocation, and client preference handling.
The primary objective of the system is to enhance efficiency in managing client requests and service delivery within the domestic services sector. By automating these processes, the system reduces administrative workload, minimizes human error, and improves the overall customer experience. Furthermore, the platform provides a user-friendly interface for clients to schedule and manage services in real time, thereby ensuring accuracy, timeliness, and flexibility in service provision.

1.1 Background of the Problem

Preliminary investigations, including user interviews, revealed that the existing manual approach to managing maid services presents several operational inefficiencies. These include the reliance on paper-based records and spreadsheets to track maid availability, service schedules, and client preferences. Such practices are prone to human error, including double-bookings, missed appointments, and incorrect data entry, which frequently result in customer dissatisfaction.
The lack of real-time updates further complicates service provision, as administrators are unable to make immediate adjustments in response to last-minute client requests.

Consequently, clients experience delays and inconsistencies in service delivery. In addition, the manual allocation of maids leads to scheduling imbalances, where some staff are overburdened while others remain underutilized. The absence of an integrated client-facing platform also forces customers to rely on intermediaries, thereby increasing miscommunication and reducing transparency. In summary, the manual system creates a significant bottleneck in service delivery. An automated system such as Maids On Call has the potential to eliminate these inefficiencies by offering a seamless, reliable, and transparent mechanism for both administrative staff and clients.

1.2 Problem Statement

The existing manual booking system is inefficient, prone to errors, and unable to meet the dynamic needs of clients. Clients are currently required to make appointments either by phone or in person, leading to delays and frequent scheduling conflicts. Furthermore, administrators lack access to real-time availability updates, resulting in overbookings and poor service allocation. Additionally, client-specific preferences are not consistently tracked, often resulting in dissatisfaction when services fail to meet individual expectations.

The manual process also imposes a significant administrative burden, as staff must manage maid assignments without automated support. Ineffective communication channels exacerbate the problem, leading to missed appointments and operational confusion. The Maids On Call System seeks to address these challenges by automating booking, allocation, and communication processes. This will ensure that services are delivered more efficiently and reliably, while simultaneously reducing administrative workload.

1.3 Project Objectives

1.3.1 General Objective

To develop an automated, cloud-based maid management system that leverages artificial intelligence (AI), and cloud computing technologies to improve efficiency, accuracy, and client satisfaction in service delivery.

1.3.2 Specific Objectives

I. To investigate existing maid management systems in order to identify gaps and areas for improvement.

II. To analyze the potential of AI, and cloud computing in enhancing booking, scheduling, and personalized client services.

III. To develop an automated booking system that enables real-time service reservations while minimizing errors.

IV. To design and implement a real-time scheduling and tracking module to optimize maid allocation and reduce conflicts.

V. To conduct testing and validation of the system to ensure functionality, usability, and alignment with project requirements.

1.4 Project Scope

1.4.1 Scope

The system is designed to automate the entire maid service management process for both residential and commercial clients. It encompasses service booking, maid allocation, scheduling, service tracking, client preference management, and communication. Clients will have the ability to make online bookings with real-time updates, while administrators will be able to allocate maids efficiently, monitor performance, and communicate updates instantly. The cloud-based design ensures scalability and reliability, enabling the platform to serve individual households as well as larger organizations. Additionally, the system supports integration with communication and notification services, further improving operational transparency.

1.4.2 Target Users

The system will serve three main categories of users:

I. Administrators**:** Includes system administrators, booking coordinators, and client relationship managers. Their responsibilities include managing bookings, updating databases, troubleshooting, and maintaining service quality.

II. Maids/Workers: They will have restricted access to view assigned schedules and client details and mark completed tasks.

III. Clients: Will interact with the system to book services, select preferences, view schedules, receive notifications, and provide service feedback.

1.4.3 Platform Specification

The Maids On Call system is designed as a cloud-based application, ensuring accessibility across desktops, laptops, tablets, and smartphones. It leverages cloud infrastructure such as Google Cloud Platform or Microsoft Azure for scalability. The front-end is developed using JavaScript, React.js, and jQuery, while the back-end incorporates PHP and Node.js with Express.js for asynchronous operations. Data is managed through MySQL or PostgreSQL, with scalable alternatives such as Firebase or Azure SQL. Security measures include SSL/TLS encryption, OAuth 2.0, and JWT for authentication.

1.5 constraints

The "Maids On Call" system faces several constraints, including budget limitations, user cooperation, and hardware requirements. Limited funding may restrict the Scope of the project, potentially excluding advanced features like AI-driven client preference management or extensive real-time tracking. Cloud hosting and modern technologies come with operational costs, and a basic version of the system may be required if the budget is constrained.

User cooperation is also crucial, as resistance to adopting the new system could delay implementation, particularly if administrative staff or clients prefer traditional methods. Additionally, hardware limitations, such as outdated devices or unreliable internet access, may hinder full system utilization. Careful financial planning and user engagement strategies will be necessary to manage these constraints and ensure successful implementation.

1.6 Project Milestones

I. Research and Analysis: Involves the investigation of existing maid management systems and researching relevant technologies (AI, ML, cloud computing).

II. System Design and Architecture: Focuses on defining the technical structure, choosing platforms (e.g., Google Cloud, Azure), and designing the system architecture.

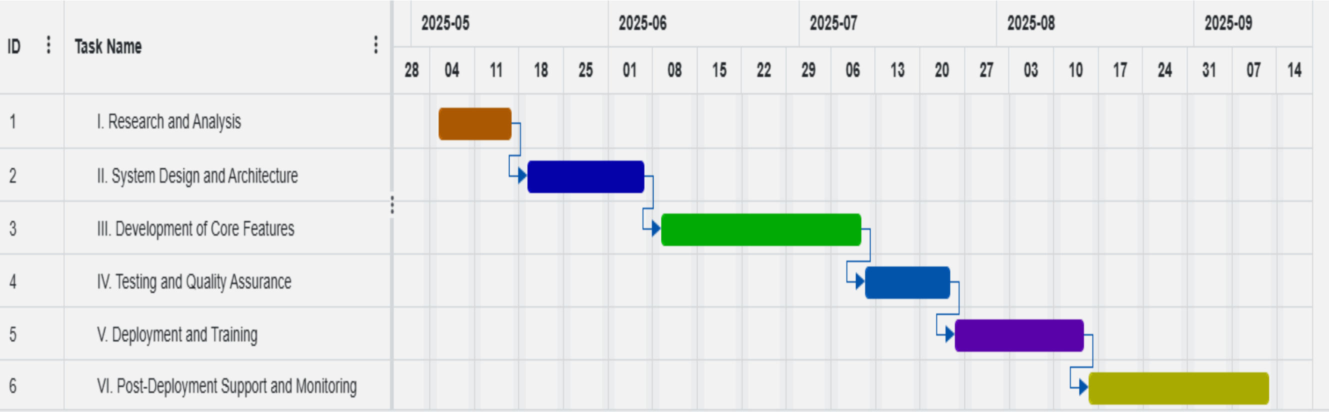
III. Development of Core Features: Involves the development of core features such as automated booking, scheduling, and client preference management.

IV. Testing and Quality Assurance: Involves testing the system for bugs and errors, and ensuring all functionalities are working correctly.

V. Deployment and Training: Includes system deployment and training of administrative staff and clients.

VI. Post-Deployment Support and Monitoring: Focuses on troubleshooting, monitoring system performance, and addressing user feedback.

Gantt Chart Project Milestones for Maids On Call System



ID	Task Name
1	I. Research and Analysis
2	II. System Design and Architecture
3	III. Development of Core Features
4	IV. Testing and Quality Assurance
5	V. Deployment and Training
6	VI. Post-Deployment Support and Monitoring

1.6 Significance of the Project

I. Streamlined Booking Process: Clients can easily book services online, reducing the need for phone calls or in-person visits.

II. Real-Time Scheduling and Tracking: Administrators can track maid availability and manage schedules instantly, minimizing conflicts.

III. Personalized Service Management, the system tracks client preferences, ensuring maid assignments are tailored to specific needs.

IV. Reduced Administrative Workload: Automation of booking, scheduling, and allocation tasks reduces staff's manual workload.

V. Improved Communication: The integrated platform ensures seamless communication between clients, maids, and administrators, improving service organization

1.7 Summary

The "Maids On Call" system is a cloud-based solution designed to automate the booking, scheduling, and allocation processes for maid service providers. By leveraging technologies like AI, ML, and cloud computing, the system aims to address inefficiencies in the current manual processes, such as scheduling conflicts, administrative workload, and poor communication. It will streamline client bookings, offer personalized services based on client preferences, and provide real-time scheduling and tracking for better maid management. This project's benefits include improved operational efficiency, enhanced customer satisfaction, and scalability, making it a crucial tool for both administrative staff and clients in the maid service industry.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

A comprehensive review of existing knowledge, systems, and technologies related to the project under study is very important in better understanding and implementation of the proposed system. This review is critical in identifying the strengths and weaknesses of current maid management platforms, as well as highlighting gaps that the proposed system seeks to address. In addition, the review explores technological advancements such as artificial intelligence (AI), machine learning (ML), and cloud computing, which can be leveraged to improve the functionality, scalability, and efficiency of maid management systems.

2.1 Existing System Review

Several maid service platforms currently operate in the Kenyan market. Each platform provides varying features related to booking, service management, and client support. A comparative analysis of these platforms is presented below.

2.1.1 Balozy Mama Fua

Balozy Mama Fua operates nationwide across all 47 counties in Kenya. It offers a wide range of household services, including cleaning, plumbing, electrical work, and beauty services. Clients can book services through its website or mobile application, with the option for same-day bookings. Payments are processed securely via M-Pesa and credit/debit cards.

One of its notable strengths is the verification of service providers, ensuring professionalism, safety, and trust. Additionally, the platform includes a rating and review system to maintain service quality. However, Balozy is limited in versatility as it focuses predominantly on cleaning services. Furthermore, clients without access to smartphones or internet services may experience difficulties using the platform.

2.1.2 Kalinoi

Kalinoi provides diverse services, including cleaning, caregiving, gardening, event planning, and technical support. It operates across multiple regions in Kenya, including Nairobi, Mombasa, Kisumu, and Eldoret. Its mobile application offers instant booking capabilities, allowing users to request services on demand. Payments are made securely through M-Pesa or bank cards.

The platform's intuitive and user-friendly interface provides a positive user experience. However, Kalinoi faces challenges associated with being a relatively new market entrant, including limited funding, scalability concerns, and the need to build trust among users.

2.1.3 MamaFua App

MamaFua primarily operates in Nairobi, with plans to expand into other regions. It specializes in cleaning services, with additional offerings such as laundry, plumbing, and handyman services. Payments can be made via M-Pesa or bank cards, with cash-on-delivery options available for first-time users.

Comparative Analysis of Maid Service Platforms in Kenya

Feature	Balozy Mama Fua	Kalinoi	MamaFua App
Service Coverage	Nationwide, covering all 47 counties in Kenya.	Operates in multiple regions across Kenya, including Nairobi,	Primarily focused on Nairobi, with plans to expand to other regions.

		Mombasa, Kisumu, Eldoret, and more.	
Service Categories	Offers a wide range of services beyond cleaning, including plumbing, electrical repairs, beauty services, and more.	Provides diverse services such as cleaning, caregiving, gardening, event planning, tech repairs, and more.	Specializes in cleaning services, with additional offerings like laundry, handyman, and plumbing services.
Booking Process	Users can book services via the Balozy app or website, with options for same-day bookings.	Offers instant booking capabilities through its app and website, allowing users to book services on demand.	Provides online booking through its app and website, with options for cash on delivery for first-time users.
Payment Methods	Supports secure payments through the app, with options like M-Pesa and credit/debit cards.	Accepts secure payments via the app, including M-Pesa and card payments.	Offers cash on delivery for first-time users; subsequent payments can be made through the app using M-Pesa or cards.
User Ratings & Reviews	Features a rating and review system where users can rate service providers, helping	Includes a review system where clients can provide feedback on services received,	Allows users to rate and review service providers, promoting accountability and quality assurance.

	maintain quality standards.	aiding in quality control.	
Verification of Service Providers	Ensures that all service providers, including "Mama Fua" (cleaning professionals), are verified, enhancing trust and safety for clients.	Connects clients with self-employed service providers, with varying levels of verification.	Vets and trains cleaning professionals before listing them on the platform, ensuring quality service delivery.
Mobile Application	Offers a user-friendly mobile app available on both Android and iOS platforms, facilitating easy access to services.	Provides a mobile app for both Android and iOS users, enabling seamless service booking and management.	Has a mobile app compatible with Android and iOS devices, allowing users to book services and manage appointments conveniently.
Customer Support	Provides customer support through the app, email, and phone, ensuring assistance is available when needed.	Offers customer support via the app, email, and phone, aiming to resolve issues promptly.	Provides customer support through the app, email, and phone, with dedicated assistance for users and service providers.

Unique Selling Proposition	Empowers domestic workers by providing them with digital visibility and fair compensation, promoting dignity and safety.	Offers a wide range of services beyond cleaning, catering to diverse household needs.	Focuses on providing professional cleaning services with transparent pricing and a user-friendly booking process.

2.1.4 Advantages and Disadvantages of Existing Maid Service Platforms in Kenya

Here's a table comparing the advantages and disadvantages of the existing maid service platforms in Kenya:

Platform	Advantages	Disadvantages
Balozy Mama Fua App https://www.balozy.com/	- **Verified Professionals**: Ensures that all service providers, including "Mama Fua," are verified, enhancing trust and safety.	- **Limited Service Categories**: Focuses mainly on cleaning services, limiting versatility for clients.
	- **Wide Coverage**: Operates across multiple counties, connecting clients with service providers in various regions.	- **Platform Accessibility**: Users without smartphones or internet access may find it challenging to use.
	- **Empowerment of Workers**: Provides digital visibility and fair compensation for domestic workers.	
Kalinoi https://www.kalinoi.com/	- **Diverse Service Offerings**: Offers cleaning, caregiving,	- **New Entrant**: As a newer platform, it may still be

	gardening, and more, providing comprehensive solutions.	building trust and a user base.
	- **User-Friendly Interface**: Designed to be intuitive, making it easy for clients to browse services and book appointments.	- **Resource Constraints**: May face challenges in scaling due to limited funding and marketing efforts.
	- **Wide Geographic Reach**: Operates across various cities and towns, increasing accessibility for users in different regions.	
MamaFua App https://www.mamafua.co.ke/	- **Comprehensive Service Range**: Offers a variety of services, including cleaning, plumbing, electrical repairs, and handyman services.	- **Service Quality Variability**: The quality of service can vary based on the individual provider.
	- **Transparent Pricing**: Provides clear pricing, helping clients make informed decisions without hidden costs.	- **Platform Navigation**: Some users have found the interface not to be user-friendly and need improvements.
	- **Established Network**: Has a large network of verified professionals, offering a diverse pool of service providers.	

Each platform has its own set of advantages and limitations. Balozy Mama Fua focuses on verified professionals and wide coverage, but lacks diverse service offerings and accessibility. Kalinoi provides a broad range of services and an intuitive interface, but faces challenges with scalability and building trust. MamaFua App offers a variety of services and transparency but struggles with inconsistency in service quality and platform navigation. In developing a new system, it's essential to integrate the strengths of these platforms while addressing their weaknesses to create a more efficient and user-friendly service.

2.2 System Adaptation for Maids On Call

The proposed Maids On Call System seeks to integrate the strengths of the reviewed platforms while mitigating their weaknesses. Specifically, the system is designed to incorporate:

- Real-Time Availability and Flexible Scheduling: Unlike some platforms that lack dynamic scheduling, the system enables real-time modifications to bookings and maid assignments.
- Transparent and Affordable Pricing: Clear and consistent pricing eliminates uncertainty for clients, making the service accessible to a wider demographic.
- Quality Control and Service Consistency: A feedback-driven rating system ensures accountability, enabling the platform to monitor and improve service delivery continuously.
- Expanded Geographic Reach: Unlike MamaFua, which is geographically limited, the system is designed with scalability in mind, ensuring expansion both locally and regionally.
- Instant Customer Support: Integrated live chat and communication tools enhance customer support and responsiveness.

2.3 Technological Opportunities

Beyond existing platforms, emerging technologies provide significant opportunities to enhance maid management systems.

- Artificial Intelligence (AI) can be applied to predict client preferences and recommend suitable maids based on booking history and feedback.

- Machine Learning (ML) can enhance allocation algorithms, improving efficiency in scheduling and reducing conflicts.
- Cloud Computing ensures scalability, reliability, and accessibility, supporting both mobile and desktop platforms while reducing infrastructure costs.

By leveraging these technologies, Maids On Call will surpass existing market offerings, providing a comprehensive, user-centered solution.

2.4 Summary

While several maid service platforms exist in Kenya, none offer a complete solution that combines transparency, real-time flexibility, scalability, and technological integration. The Maids On Call System addresses these shortcomings by incorporating the best practices of current platforms, while adopting advanced technologies such as AI, and cloud computing to enhance personalization, efficiency, and service quality.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

System analysis and design constitute critical stages in the development of the Maids On Call System. These phases ensure that the system is structured to meet both user requirements and organizational objectives. Analysis involves examining functional and non-functional requirements, while design translates these requirements into system architecture, workflows, and database structures. This chapter discusses the development methodology, requirement specifications, system modeling, and database design applied in the project.

3.1 Development Approach

The project employed the Agile Software Development Methodology. Agile was selected because of its iterative, flexible, and user-centered approach, which aligns with the dynamic requirements of the system.

3.1.1 Rationale for Agile Selection

I. Flexibility in Change Management: Agile supports continuous modification of features based on user feedback. For example, if clients request additional booking filters, these can be implemented within subsequent sprints without major disruption.

II. User Engagement: The methodology emphasizes collaboration between developers, administrators, and clients, ensuring that the evolving system reflects actual user needs.

III. Incremental Delivery: Features are developed in small, manageable iterations, allowing functional modules such as booking or scheduling to be deployed early for testing.

IV. Risk Management: Agile identifies challenges early in the development process, enabling prompt corrective action.

3.1.2 Agile Implementation Examples on "Maids On Call" Systems

- Sprint 1: Development of a basic booking interface for clients to select services and make reservations.
- Sprint 2: Integration of maid allocation and real-time scheduling features.
- Sprint 3: Development of client preference management, enabling customers to select service preferences such as timing or specific personnel.
- Sprint 4: Incorporation of AI-based personalization features, such as maid recommendations based on prior bookings.

The use of Agile ensures rapid prototyping, early feedback, and continuous system improvement, ultimately resulting in a more responsive and reliable platform.

3.2 Requirement Specification

The system requirements were categorized into functional and non-functional specifications.

3.2.1 Functional Requirements

I. User Management: Authentication and authorization for administrators, maids, and clients.

II. Booking Management: Ability for clients to make, modify, and cancel bookings.

III. Scheduling: Real-time allocation of maids based on availability and workload.

IV. Preference Tracking: Storage and retrieval of client-specific service preferences.

V. Communication Module: Notifications and updates delivered to clients and maids.

VI. Feedback System: Clients can rate services and provide feedback to enhance quality control.

3.2.2 Non-Functional Requirements

I. Performance: The system should support concurrent bookings without delays.

II. Scalability: Capable of expanding to serve regional and national markets.

III. Security: Use of SSL/TLS encryption and secure authentication protocols (OAuth 2.0, JWT).

IV. Usability: Intuitive, responsive interface accessible across multiple devices.

V. Reliability: High system uptime and fault tolerance through cloud infrastructure.

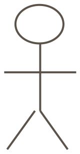
3.3 Logical Design

The Logical Design phase focuses on detailing the structure and flow of the "Maids On Call" system through various diagrams. These diagrams help illustrate the system's functionality, user interactions, and the relationships between components. Below are the diagrams required for this phase, along with an explanation of each.

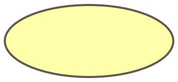
UML Diagrams:

Actor:

A coherent set of roles that users of use cases play when interacting with the use cases.



Use case: A description of a sequence of actions, including variants, that a system performs that yields an observable result of value to an actor.



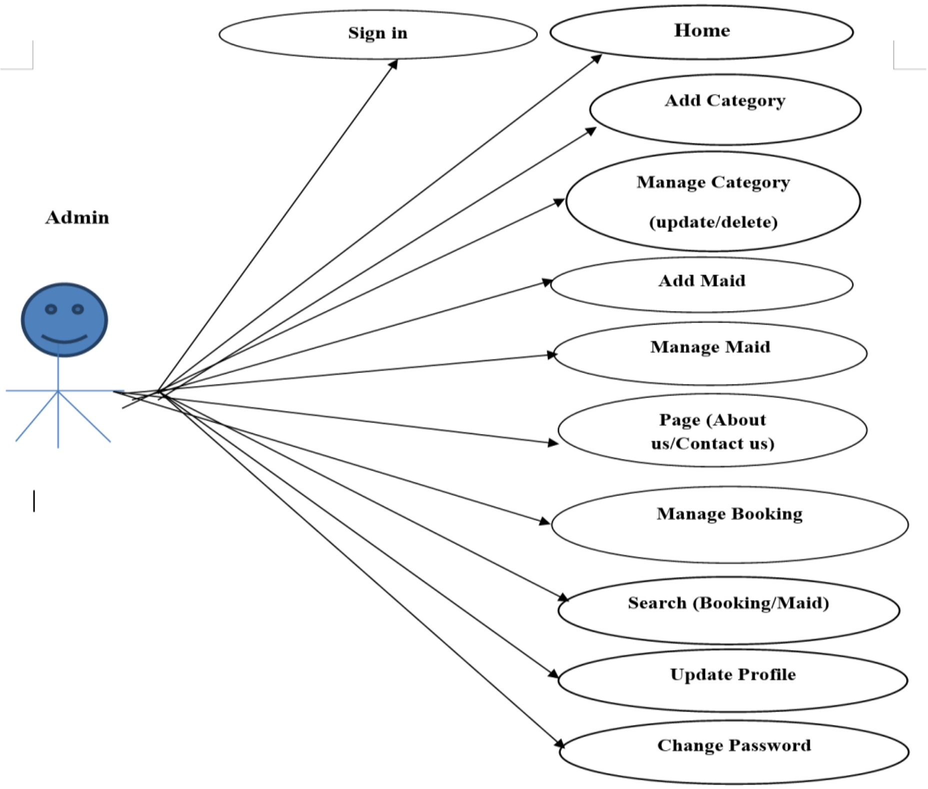
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing, and documenting the system. This is the step that is taken while developing any product after analysis. The goal of this is to produce a model of the entities involved in the project, which later need to be built. The representation of the entities that are to be used in the product being developed needs to be designed.

3.3.1 Use Case Diagram - (Interaction Between System and User / Capture Business Scenario)

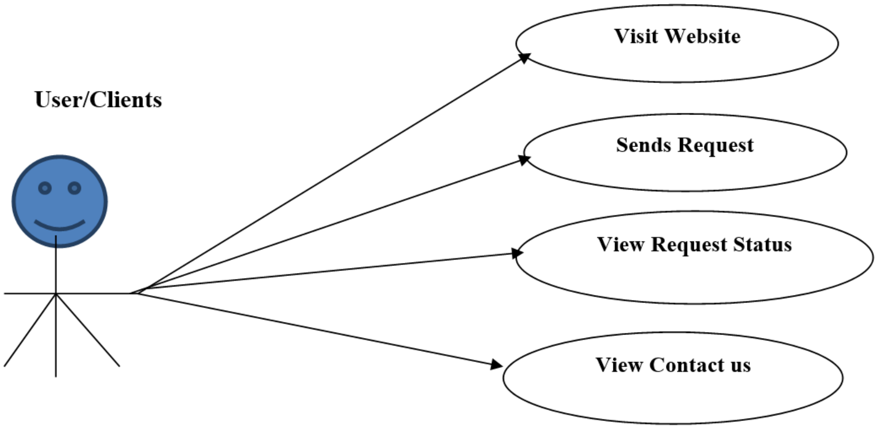
A Use Case Diagram illustrates the interactions between users (clients, administrators, maids) and the system. It defines the system's functionalities based on user roles and how they interact with the system. A Use case is a description of a set of actions.

Graphically, it is rendered as an ellipse with a solid line including only its name. A use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationships. It is an association between the use cases and actors. An actor represents a real-world object—primary Actor – Sender, Secondary Actor Receiver.

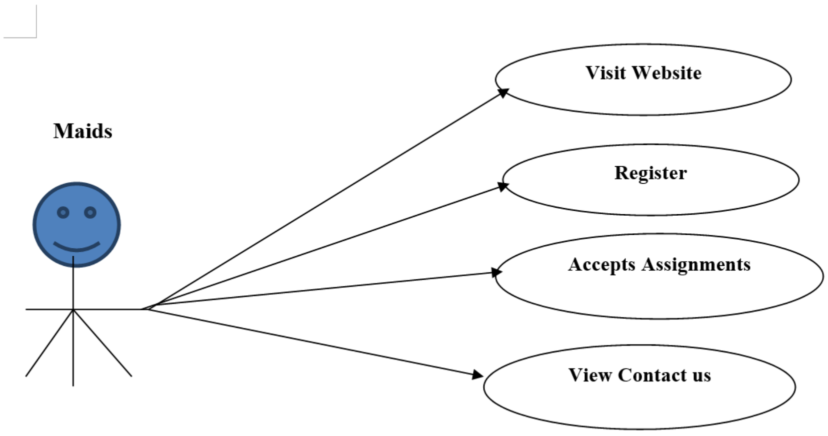
Use Case Diagram



Users Case Diagram

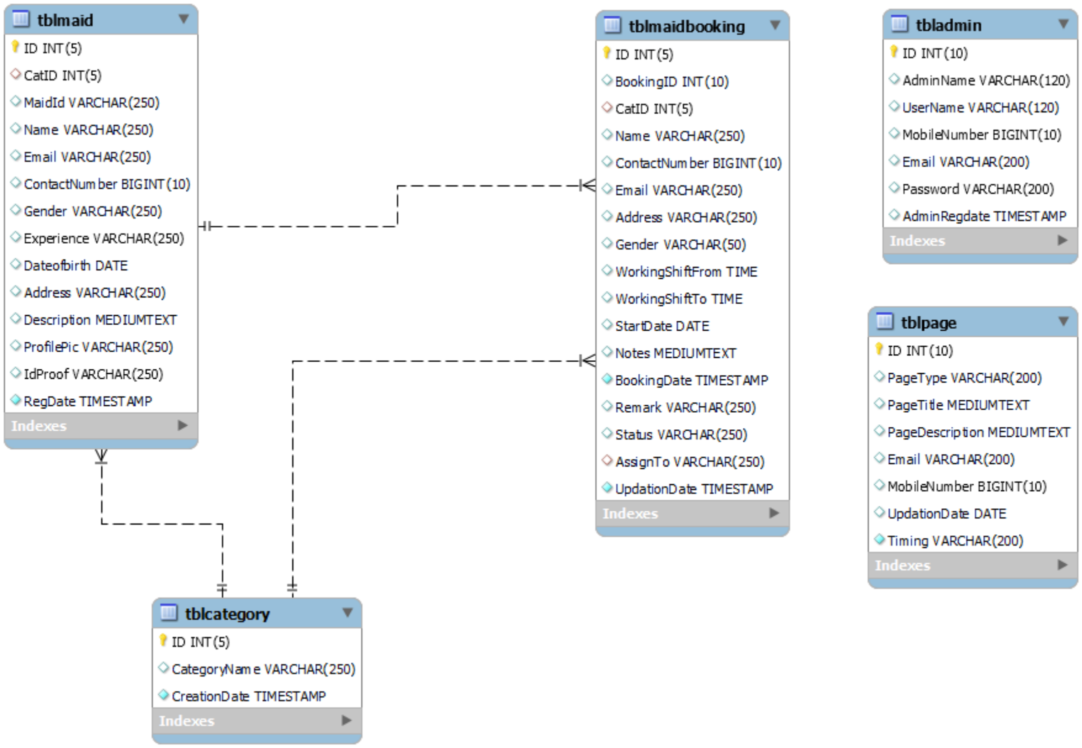


Maids Case Diagram



3.3.2 Class Diagram:

A description of a set of objects that share the same attributes, operations, relationships, and semantics



3.3.3 Activity Diagram

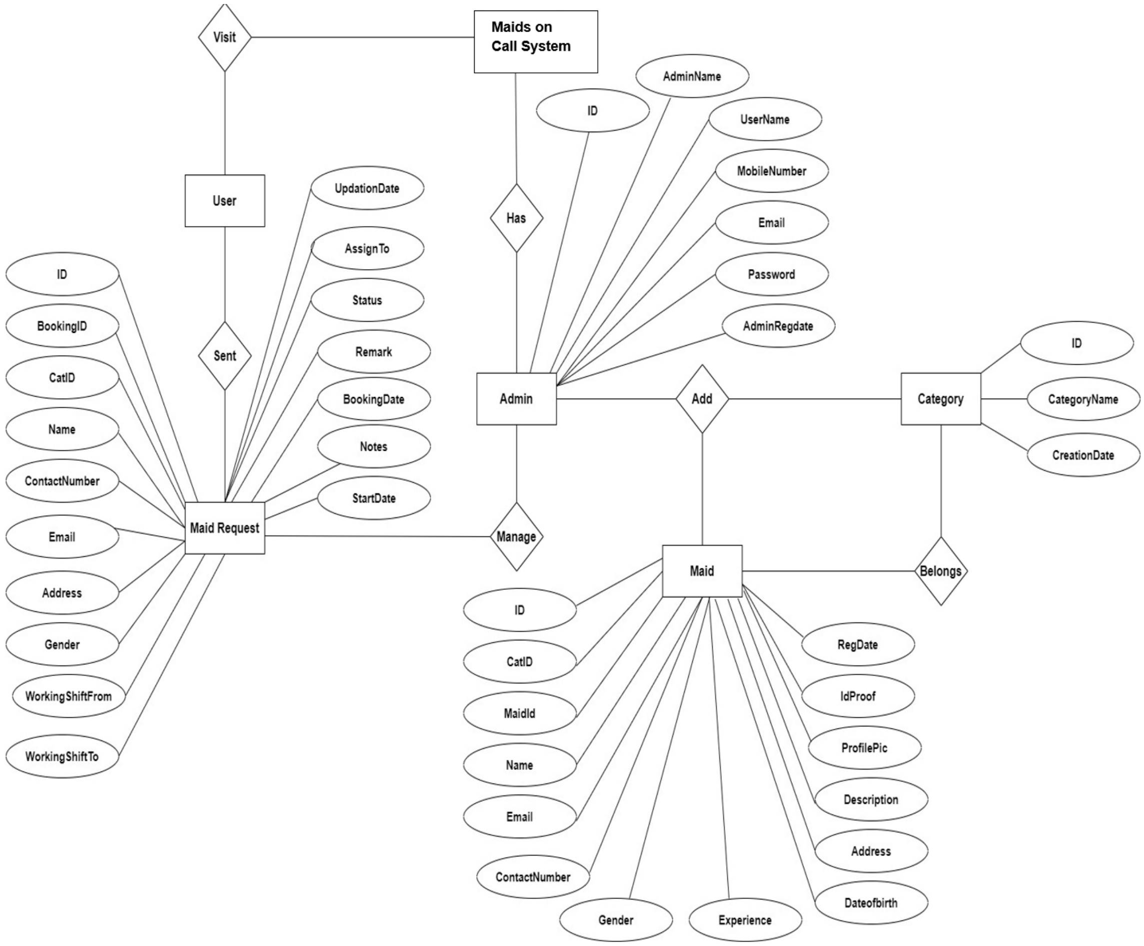
The activity diagram for the "Maids On Call" system outlines the process flow between the Customer and the Admin. It visually illustrates the steps involved in searching for maids, making a booking, admin approval, and final actions like payment processing and service confirmation. The diagram is divided into two swimlanes: one for the Customer and another for the Admin, highlighting their respective actions and decision points. Key decisions include checking maid availability, processing the booking, and finalizing the service based on admin approval.

3.3.4 The ER (Entity-Relationship) Diagram

The ER diagram for the "Maids On Call" system represents the various entities involved in the system and their relationships. It includes the following main entities:

- tblmaid: Contains information about the maids, including their ID, name, contact details, experience, gender, and proof of identity. It also has a relationship with tblcategory to categorize maids based on their skills or specialties.
- tblcategory: Represents the different categories of maids (such as cleaners, specialists, etc.), with a relationship to the tblmaid entity to assign maids to specific categories.
- tblmaidbooking: Tracks the booking details, such as the booking ID, the maid being booked, contact information, working shift, and service start and end times.
- tbladmin: Stores administrator information, including the Admin's credentials and the date of registration, managing bookings and approvals.
- tblpage: Contains details about the system's pages, such as the page type, description, and related contact information for administrative purposes.

These entities are interconnected, where a maid can belong to one or more categories, bookings link maids to customers, and the Admin manages and approves these bookings. The ER diagram efficiently shows how each entity is related through primary and foreign keys, ensuring a well-structured database for the system.



3.3.5 Data Flow Diagram

A Data Flow Diagram (DFD) for the "Maids On Call" system outlines how data moves within the system, illustrating the processes, data stores, data inputs, and outputs. Here's an overview of the DFD for the system, divided into different levels:

Level 0 DFD

Outlines the key components interacting with the central system, with each process representing a different functional area of the system. The **Maids On Call System** receives input from all these processes and external entities, performing necessary actions, and sending output to each of these systems based on their roles in managing maid services. This diagram serves as an overview of how the system interacts with different components for efficient functioning.

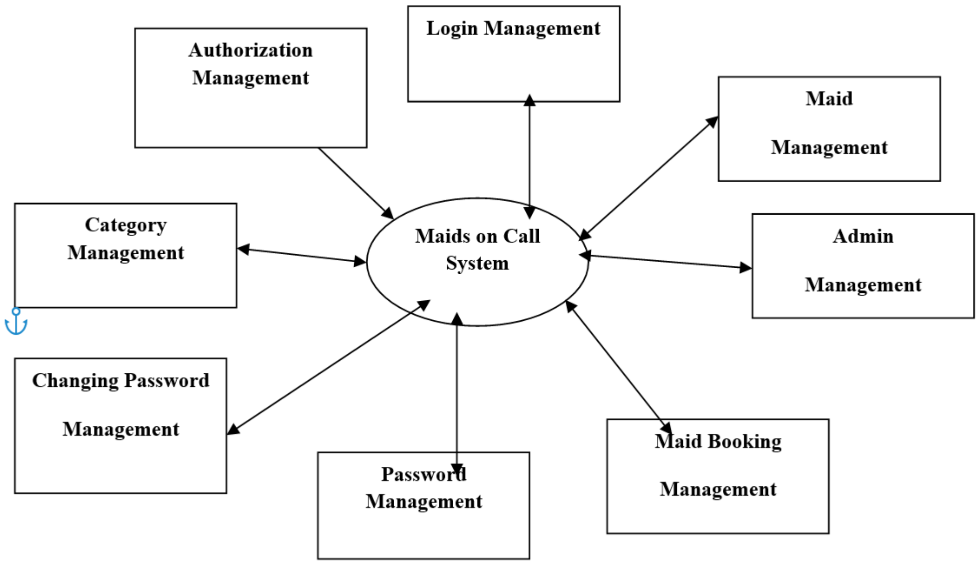
Key Components:

Maids on Call System: The central process that interacts with various management functions.

External Entities and Processes:

- Authorization Management: Handles user authentication and authorization for accessing the system.
- Login Management: Manages user login functionality.
- Maid Management: Handles the details and management of maids within the system.
- Admin Management: Manages administrative functions such as approving bookings, updating system settings, etc.
- Category Management: Manages the categorization of maids based on their skills or services.
- Changing Password Management: Facilitates users in changing their passwords securely.
- Password Management: Deals with storing and managing user passwords securely.
- Maid Booking Management: Manages the booking process for maid services.

Zero Level DFD



Level 1 DFD

The First Level DFD (Data Flow Diagram) for the "Maids On Call" system provides a more detailed view of the system's processes and the flow of data between them. It illustrates how the core system interacts with various management modules, which are each responsible for a specific function within the system. Here is a breakdown of the components.

Key Components:

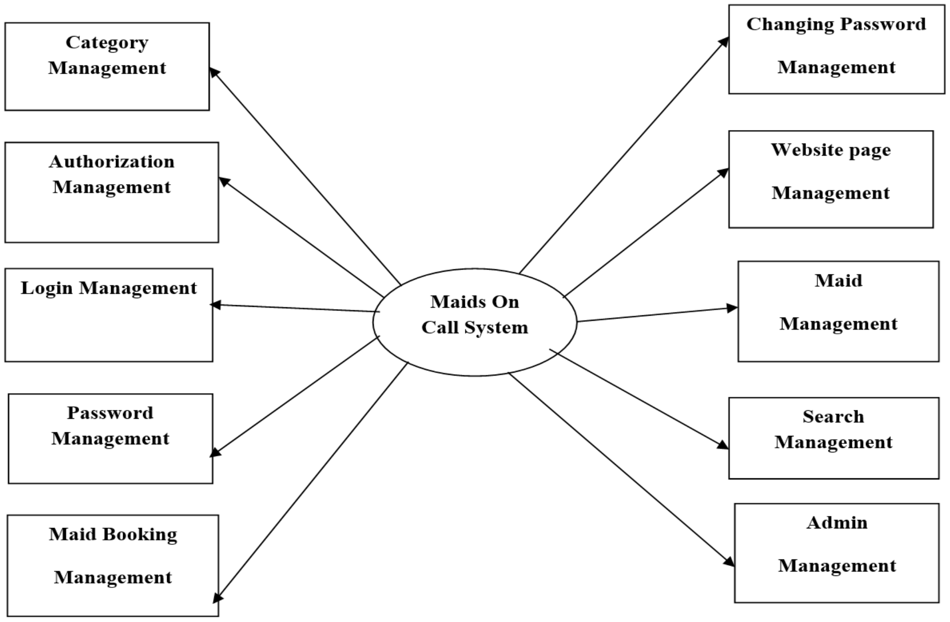
- Maids On Call System: This is the central process that coordinates interactions with all the management modules.

External Entities and Processes:

- Category Management: Handles the categorization of maids into different service types or categories.
- Authorization Management: Manages user permissions, ensuring users have the correct access levels for various system functions.

- Login Management: Manages user logins, including authentication and session management.
- Password Management: Ensures secure storage and retrieval of user passwords.
- Maid Management: Manages all the details of the maids, including their profiles, assignments, and availability.
- Maid Booking Management: Manages the entire booking process for customers, including selecting maids, setting schedules, and confirming bookings.
- Search Management: Facilitates search functionalities for customers looking for maids based on specific criteria (e.g., location, skills).
- Admin Management: Allows administrators to manage the system, including approval of bookings, managing maids, and monitoring system performance.
- Website Page Management: Manages the content and updates related to the system's website, including user interface elements.

First level DFD



Level 2 DFD

The Second Level DFD for the "Maids On Call" system provides a deeper look into the interaction between the Admin and the system, detailing the processes and data flows involved in managing the system's modules.

Key Components:

- Admin: The central external entity interacting with the system, managing and overseeing various functions within the system.

Login Process:

- Login to System: Admin logs into the system using credentials.
- Check Credentials: The system verifies whether the Admin's login credentials are valid.
- Forgot Password: If credentials are forgotten, the Admin can reset the password through the system.

Role-Based Access Control:

- Check Roles of Access: The system checks the Admin's role to determine which modules and functionalities the Admin is authorized to access.
- Manage Modules: The core management processes that the Admin can interact with, including:
- Add Category: The Admin can add new maid categories.
- Manage Category (Update/Delete): The Admin can update or delete existing categories.
- Add Maid: The Admin can add new maids to the system.
- Manage Maid (Update/Delete): The Admin can update or delete maid details.
- Manage Booking: Admin manages booking information, including approval or modification of bookings.

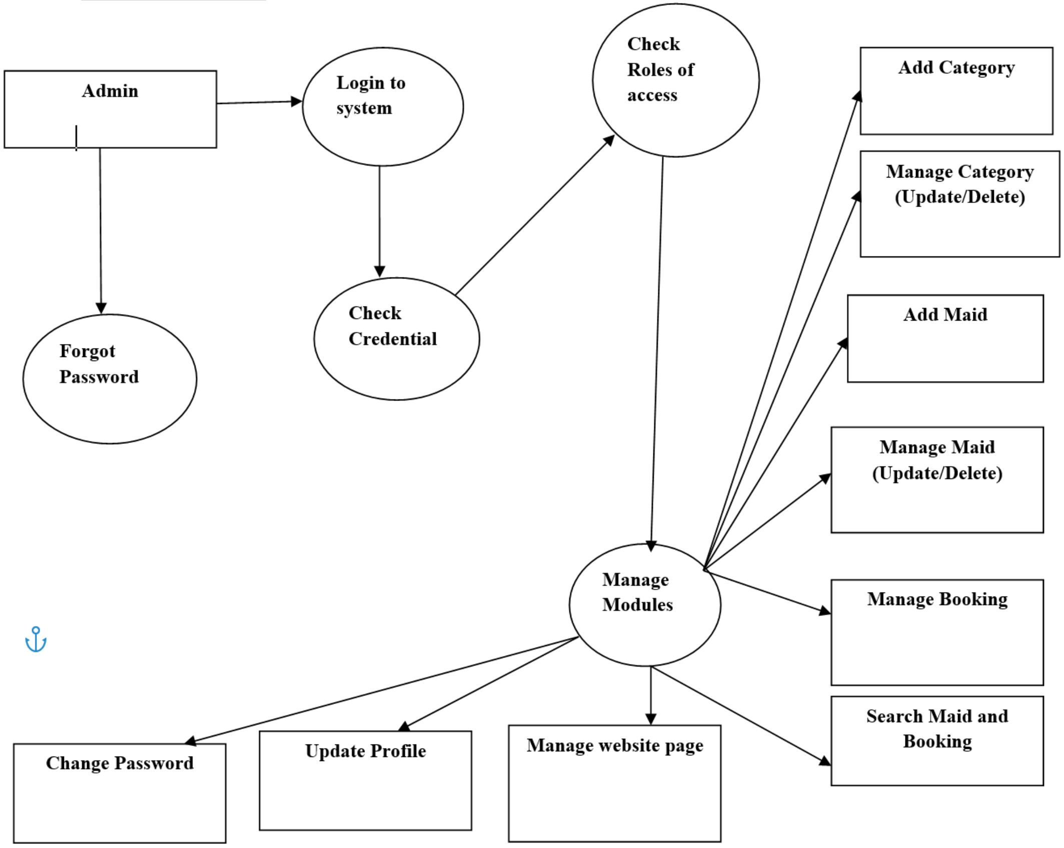
- Search Maid and Booking: The Admin can search for specific maids and review related bookings.
- Manage Website Page: The Admin can update and manage the website's pages, such as content updates or changes.

Flow of Data:

Admin interacts with the system by logging in, checking credentials, and managing various system modules based on their role.

The Manage Modules process interacts with multiple management functions such as adding, updating, or deleting maids, categories, and bookings.

Second Level DFD



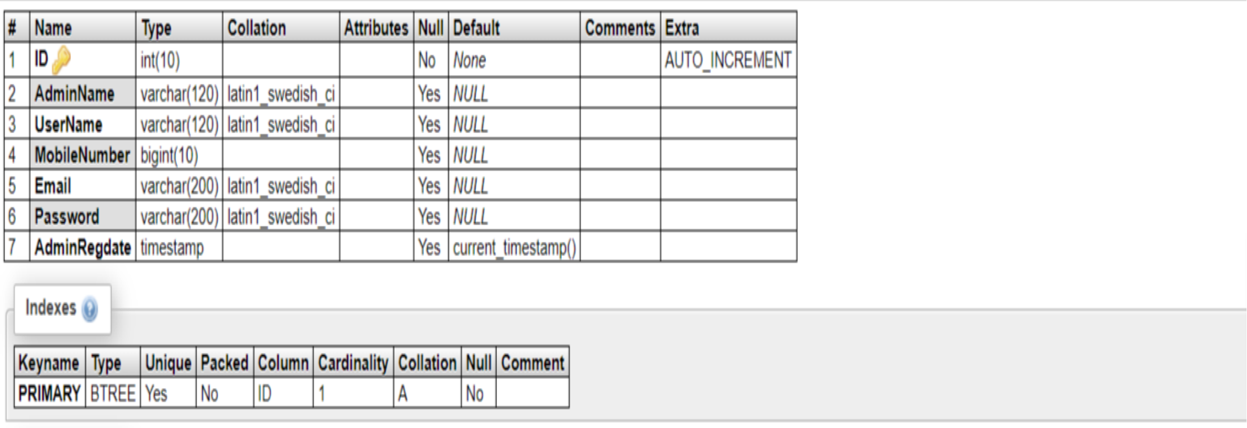
3.4 Database Design

The Database Design for the "Maids On Call" system leverages MySQL to structure and manage the data efficiently using tables, relationships, and keys. The database design aims to capture and store all the necessary information related to maids, categories, bookings, user roles, and system interactions. Here's a description of how the system's database design is structured using MySQL data tables:

Key Tables in the Database Design:

Admin Table :(Table name is tbladmin)

This store admin's personal and login details.



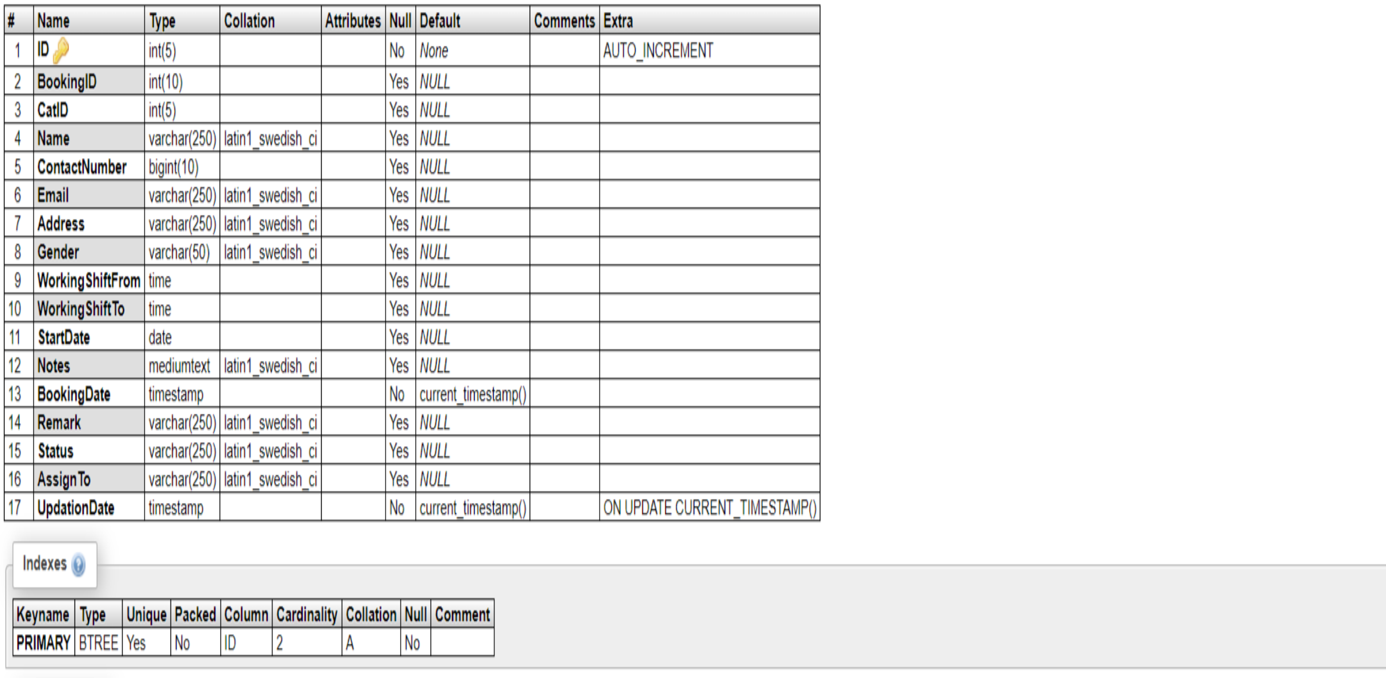
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	None		AUTO_INCREMENT
2	AdminName	varchar(120)	latin1_swedish_ci		Yes	NULL		
3	UserName	varchar(120)	latin1_swedish_ci		Yes	NULL		
4	MobileNumber	bigint(10)			Yes	NULL		
5	Email	varchar(200)	latin1_swedish_ci		Yes	NULL		
6	Password	varchar(200)	latin1_swedish_ci		Yes	NULL		
7	AdminRegdate	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	1	A	No	

Booking Table (Table name is tblmaidbooking)

This stores the details of making a maid request



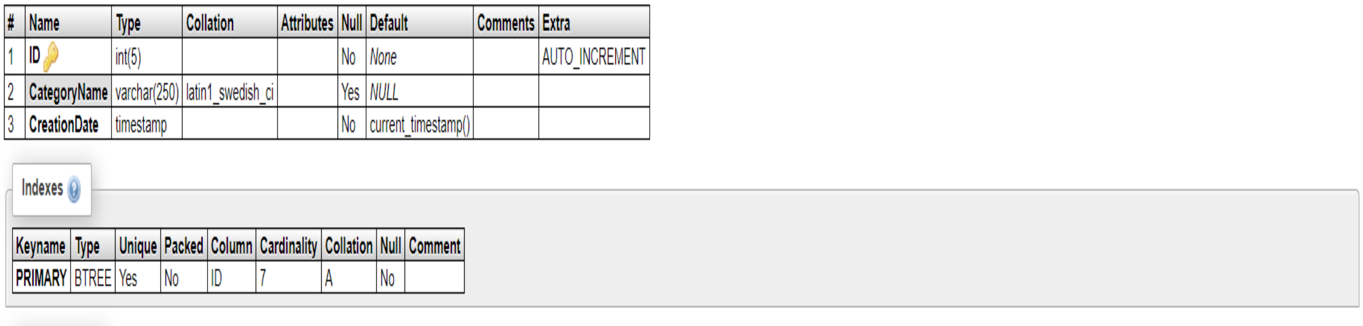
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	BookingID	int(10)			Yes	*NULL*		
3	CatID	int(5)			Yes	*NULL*		
4	Name	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
5	ContactNumber	bigint(10)			Yes	*NULL*		
6	Email	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
7	Address	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
8	Gender	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
9	WorkingShiftFrom	time			Yes	*NULL*		
10	WorkingShiftTo	time			Yes	*NULL*		
11	StartDate	date			Yes	*NULL*		
12	Notes	mediumtext	latin1_swedish_ci		Yes	*NULL*		
13	BookingDate	timestamp			No	current_timestamp()		
14	Remark	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
15	Status	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
16	AssignTo	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
17	UpdationDate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	2	A	No	

Maid Category Table: (Table name is tblcategory)

This stores the maid category detail



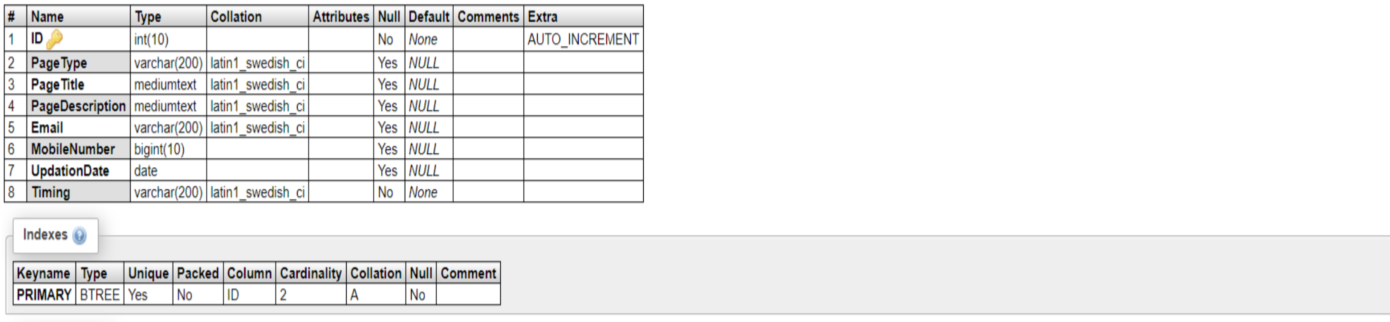
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	CategoryName	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
3	CreationDate	timestamp			No	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	7	A	No	

Page Table: (Table name is tblpage)

This table stores the about us and contact us.



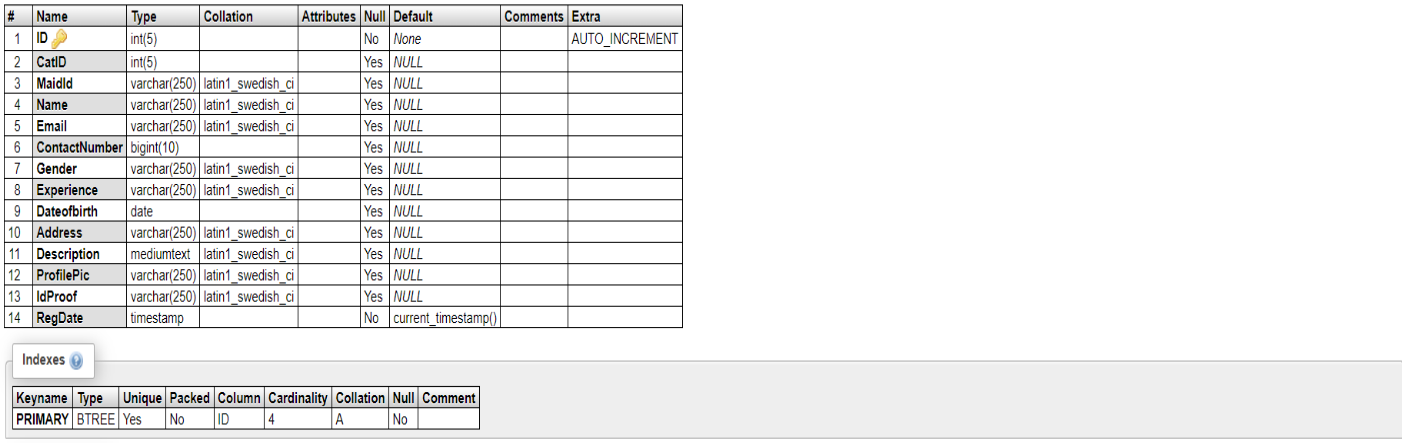
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	PageType	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	PageTitle	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	PageDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
5	Email	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
6	MobileNumber	bigint(10)			Yes	*NULL*		
7	UpdationDate	date			Yes	*NULL*		
8	Timing	varchar(200)	latin1_swedish_ci		No	*None*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	2	A	No	

Maid Table: (Table name is tblmaid)

This table stores the details of maids available.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 🔑	int(5)			No	*None*		AUTO_INCREMENT
2	CatID	int(5)			Yes	*NULL*		
3	MaidId	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
4	Name	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
5	Email	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
6	ContactNumber	bigint(10)			Yes	*NULL*		
7	Gender	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
8	Experience	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
9	Dateofbirth	date			Yes	*NULL*		
10	Address	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
11	Description	mediumtext	latin1_swedish_ci		Yes	*NULL*		
12	ProfilePic	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
13	IdProof	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
14	RegDate	timestamp			No	current_timestamp()		

Indexes

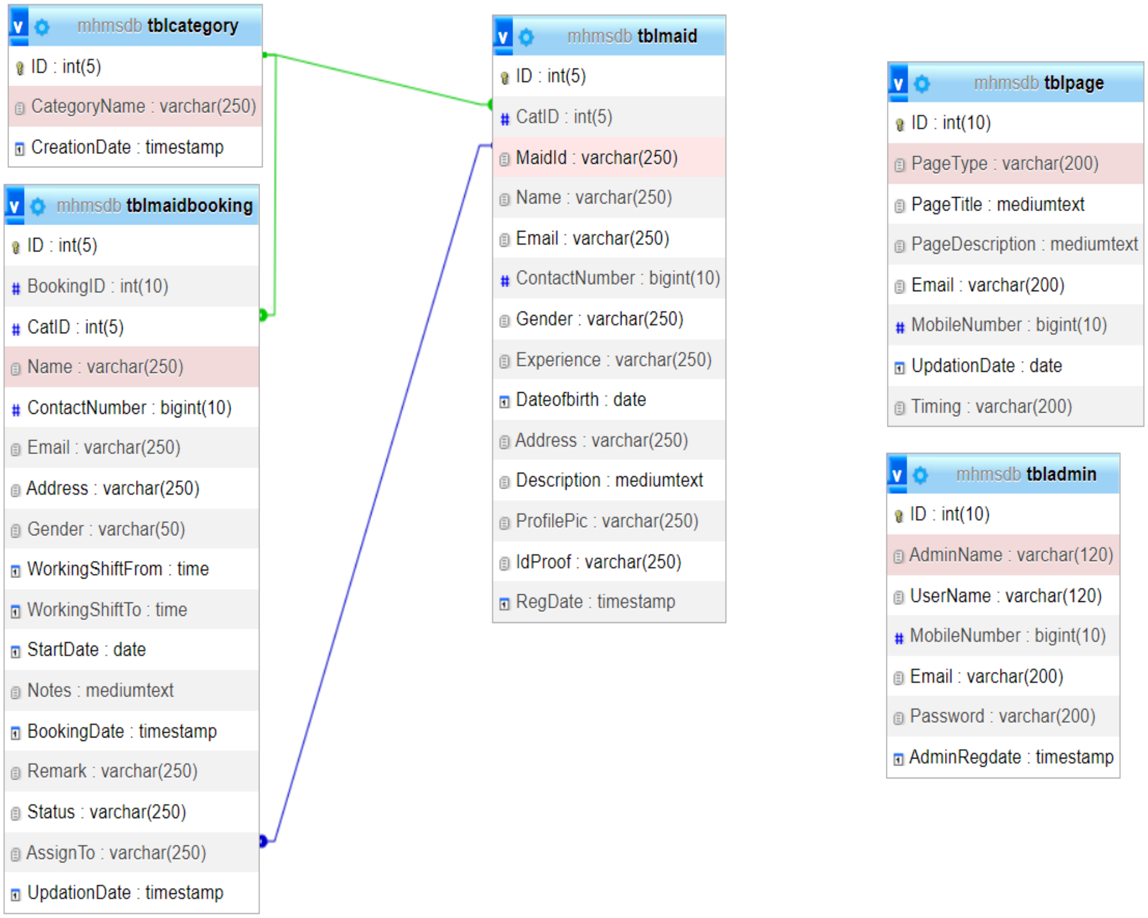
Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ID	4	A	No	

3.5 Database Schema

The Database Schema for the "Maids On Call" system is structured with several key tables that handle different aspects of the system's operations. The schema captures essential relationships between entities such as maids, bookings, categories, admins, and website pages, as detailed below.

Using a MySQL Database, the DB Schema is well-structured for the "Maids On Call" system, covering the management of maids, categories, bookings, admins, and website content. The use of foreign keys ensures the integrity of relationships, while each table serves a distinct role in managing the data associated with the maids, bookings, and system administration. This database design ensures efficient data management and smooth system operation.

Key Tables and Relationships:



mhmsdb **tblcategory**

ID : int(5)
CategoryName : varchar(250)
CreationDate : timestamp

mhmsdb **tblmaidbooking**

ID : int(5)
BookingID : int(10)
CatID : int(5)
Name : varchar(250)
ContactNumber : bigint(10)
Email : varchar(250)
Address : varchar(250)
Gender : varchar(50)
WorkingShiftFrom : time
WorkingShiftTo : time
StartDate : date
Notes : mediumtext
BookingDate : timestamp
Remark : varchar(250)
Status : varchar(250)
AssignTo : varchar(250)
UpdationDate : timestamp

mhmsdb **tblmaid**

ID : int(5)
CatID : int(5)
MaidId : varchar(250)
Name : varchar(250)
Email : varchar(250)
ContactNumber : bigint(10)
Gender : varchar(250)
Experience : varchar(250)
Dateofbirth : date
Address : varchar(250)
Description : mediumtext
ProfilePic : varchar(250)
IdProof : varchar(250)
RegDate : timestamp

mhmsdb **tblpage**

ID : int(10)
PageType : varchar(200)
PageTitle : mediumtext
PageDescription : mediumtext
Email : varchar(200)
MobileNumber : bigint(10)
UpdationDate : date
Timing : varchar(200)

mhmsdb **tbladmin**

ID : int(10)
AdminName : varchar(120)
UserName : varchar(120)
MobileNumber : bigint(10)
Email : varchar(200)
Password : varchar(200)
AdminRegdate : timestamp

CHAPTER 4

SYSTEM IMPLEMENTATION AND TESTING

4.0 Introduction

The "Maids On Call" system is implemented as a web-based platform, designed to be accessible across various devices, ensuring convenience for both customers and administrators. The web platform is optimized for standard web browsers, including Google Chrome, Mozilla Firefox, and Safari, making it compatible with the majority of user devices, whether desktop computers, laptops, tablets, or smartphones.

4.1 Front-End Implementation:

The front-end interface is designed with a user-centric approach, ensuring ease of use and a smooth user experience. The front end is built using HTML, CSS, and JavaScript, along with React.js (or a similar front-end framework) to provide a dynamic and interactive experience. The design is responsive, meaning the layout and content automatically adjust to fit the screen size of different devices, offering a seamless experience for customers booking services from smartphones or computers. Key features of the front-end include:

- Search functionality: Customers can easily search for maids by service category, availability, and location.
- Booking interface: The booking process is streamlined, allowing customers to input their details and preferences, select maid services, and make payments smoothly.
- Admin Dashboard: A clean and intuitive dashboard for admins to manage bookings, maids, categories, and view reports.

4.2 Back-End Implementation:

The back-end is responsible for handling data management, processing, and communication between the front-end interface and the database. It is built using the PHP Laravel Framework with an Express.js server to handle HTTP requests and API routes. The back-end logic includes:

- Database Connectivity: The system uses MySQL to manage the relational database, where tables for maids, customers, bookings, categories, and admin roles are stored. SQL queries are optimized for quick retrieval and storage of data.
- Authentication and Authorization: Multi-user roles (Admin and Customer) are managed through an authentication system with JWT (JSON Web Tokens) for secure login and session management. Admins have extended privileges to manage and update maids, bookings, and website content.
- Payment Integration (To be done later): The back-end will be integrated with payment gateways such as Stripe or PayPal to handle secure financial transactions, ensuring safe and efficient payments for bookings.
- Data Validation and Processing: All customer inputs, bookings, and admin changes are validated to ensure correctness before being stored in the database, minimizing errors and ensuring data integrity.
- The back-end is optimized for performance, ensuring that queries to the database are efficient, even as the number of users or data grows. It uses techniques like caching for frequently accessed data, load balancing for scalability, and database indexing to speed up search operations.

4.3 Testing:

Unit Testing:

Unit testing involves testing each module in isolation to ensure that every component works as intended. This includes testing core features such as:

Maid Search: Ensuring that search results are accurate and returned based on the criteria set by the user.

Booking Functionality: Verifying that customers can book services, input personal details, and receive confirmation.

Integration Testing:

Once individual modules are tested, integration testing ensures that they work together seamlessly. This phase verifies that the various system components, such as the front-end interface, back-end server, and database, interact correctly. Key integrations tested include:

Booking to ensure that a booking is successfully created

Admin and Customer Interaction: Testing the flow of data between the Admin's dashboard and customer bookings, including the ability for admins to view and manage bookings.

System Testing:

System testing verifies the overall performance, stability, and functionality of the complete system. It involves checking the system's response under various conditions:

Load Testing: Ensuring the system can handle multiple users and bookings simultaneously without crashing.

Security Testing: Verifying that user data, including passwords and payment details, is stored securely using encryption and hashing techniques.

UI/UX Testing: Ensuring the user interface works as expected and that users can interact with the system without difficulty.

User Acceptance Testing (UAT):

UAT is the final stage of testing, where real users test the system to confirm it meets their needs and expectations. During UAT, a group of representative users (both customers and admins) interact with the system to validate the following:

Customer Experience: Whether customers can easily search for and book maids, complete payments, and manage their bookings.

Admin Experience: Whether admins can efficiently manage maids, bookings, categories, and system content.

Functionality Validation: To ensure that all system features work correctly in real-world scenarios, including the ease of navigation and clarity of instructions, functionality validation was conducted.

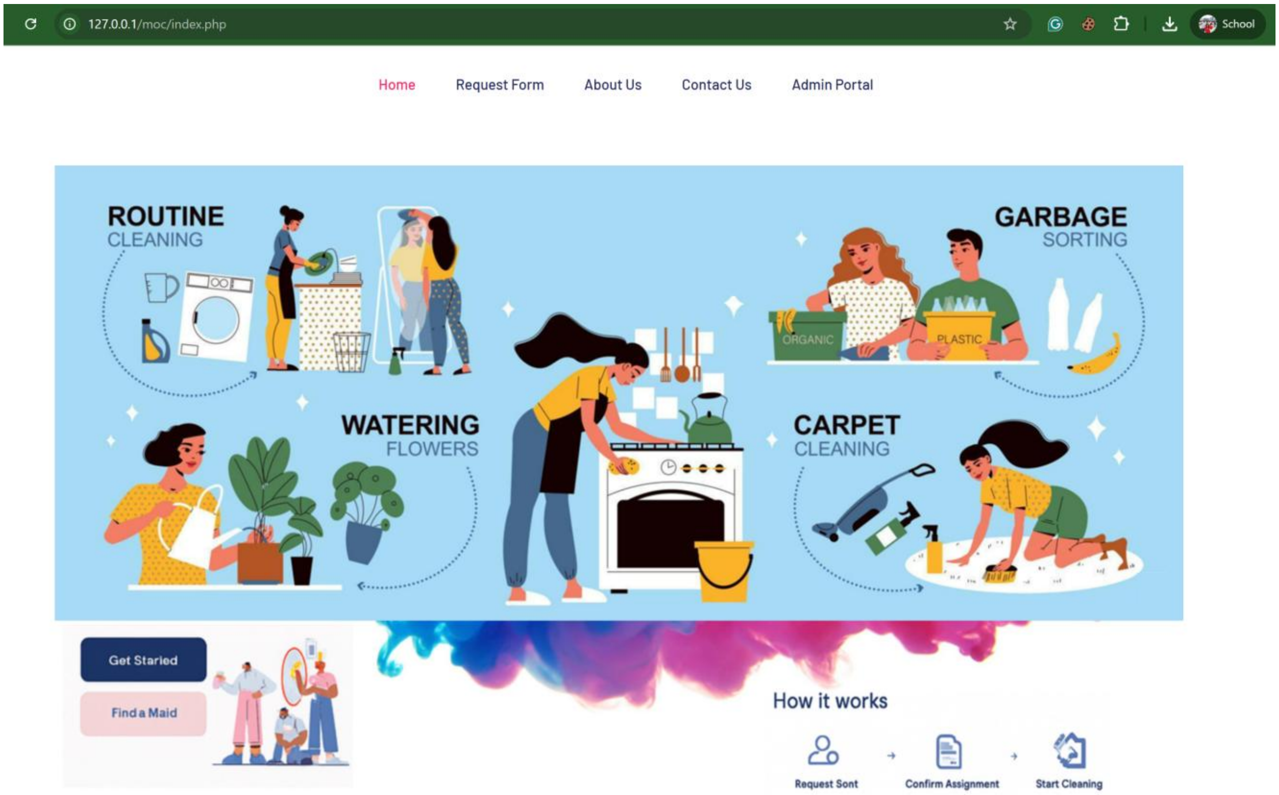
The implementation and testing phases of the "Maids On Call" system ensured that the platform is functional, secure, and user-friendly. The system has been tested at each stage to identify and fix issues before deployment, ensuring that users have a seamless experience, whether they are customers booking services or admins managing the system.

4.4 Evaluation

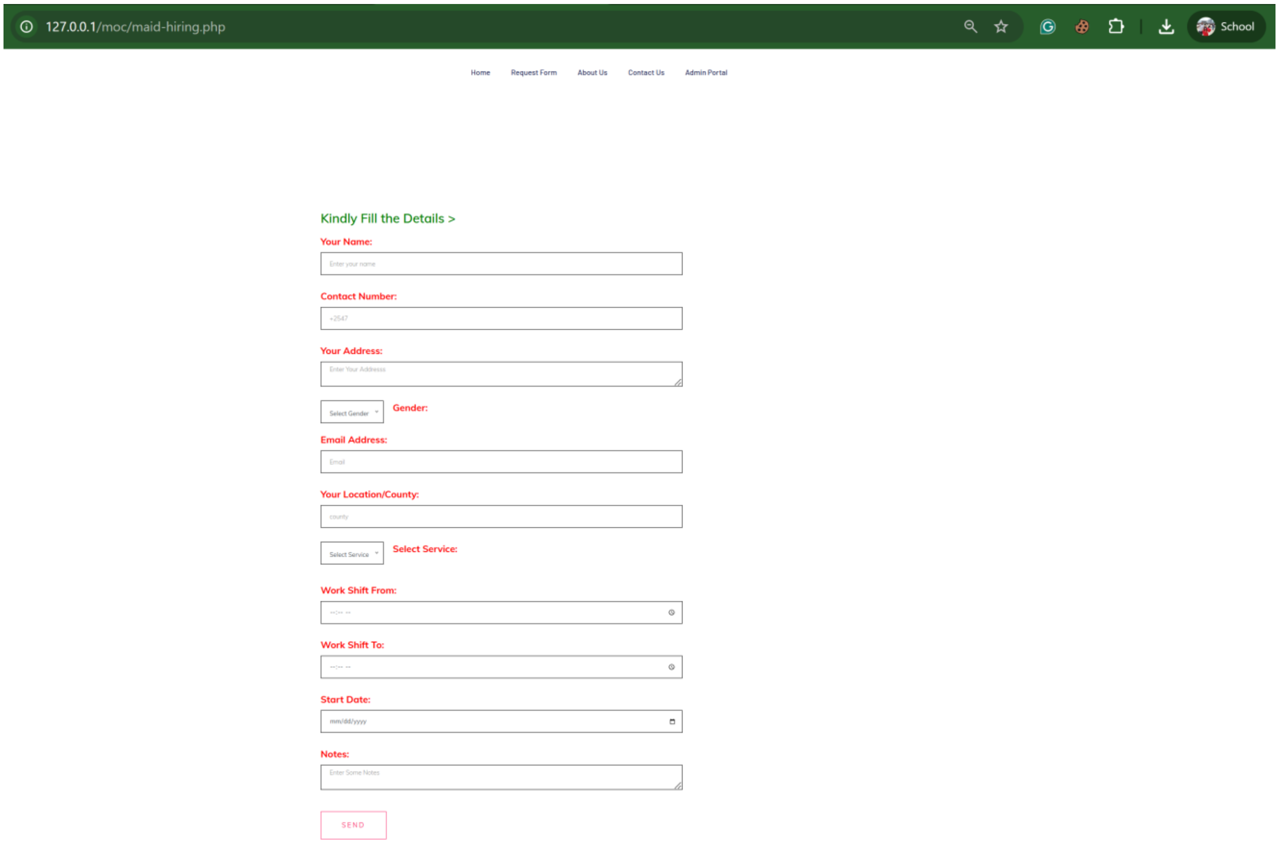
Project is locally hosted on URL: http://127.0.0.1/moc/index.php Xampp Server,

Below are key web pages of the "Maids On Call" system user interfaces

Home Page



Request Form Page


127.0.0.1/moc/maid-hiring.php

Kindly Fill the Details >

Your Name: Enter your name

Contact Number: +2547

Your Address: Enter Your Addresss

Select Gender Gender:

Email Address: Email

Your Location/County: county

Select Service Select Service:

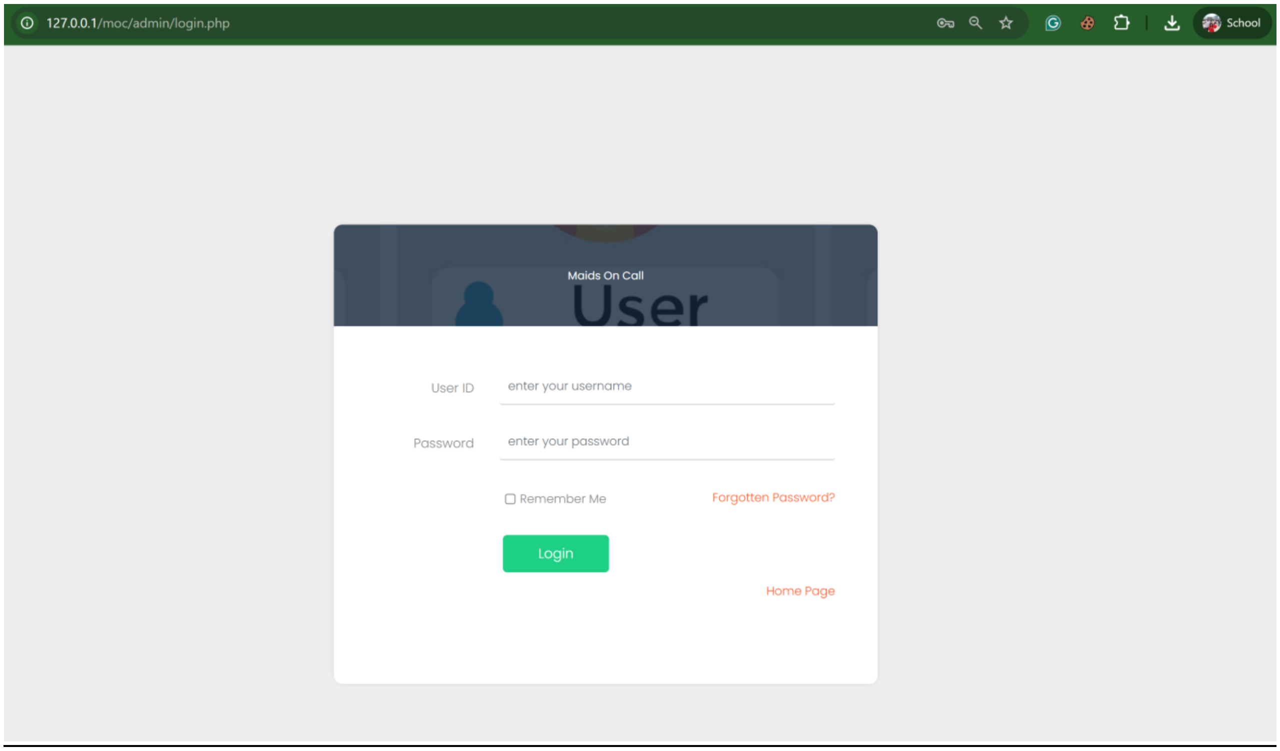
Work Shift From: --:-- --

Work Shift To: --:-- --

Start Date: mm/dd/yyyy

Notes: Enter Some Notes

Admin Page

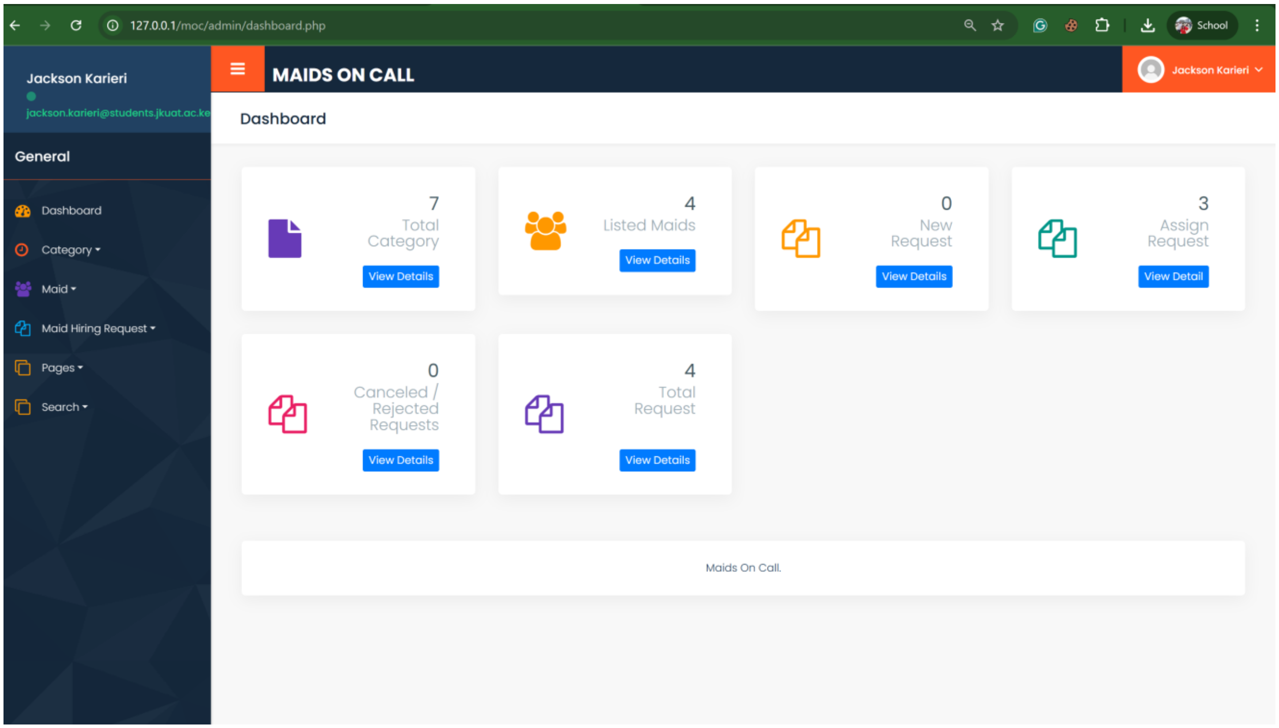

127.0.0.1/moc/admin/login.php

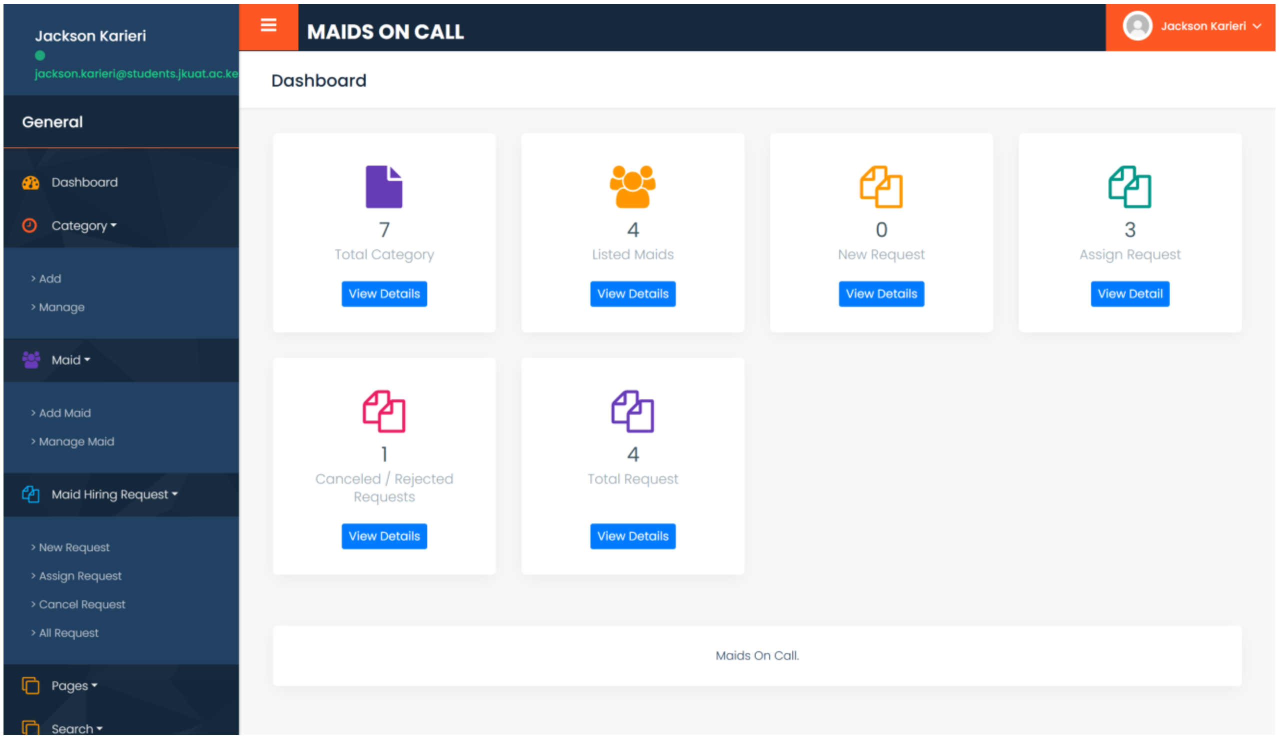
User ID enter your username

Password enter your password

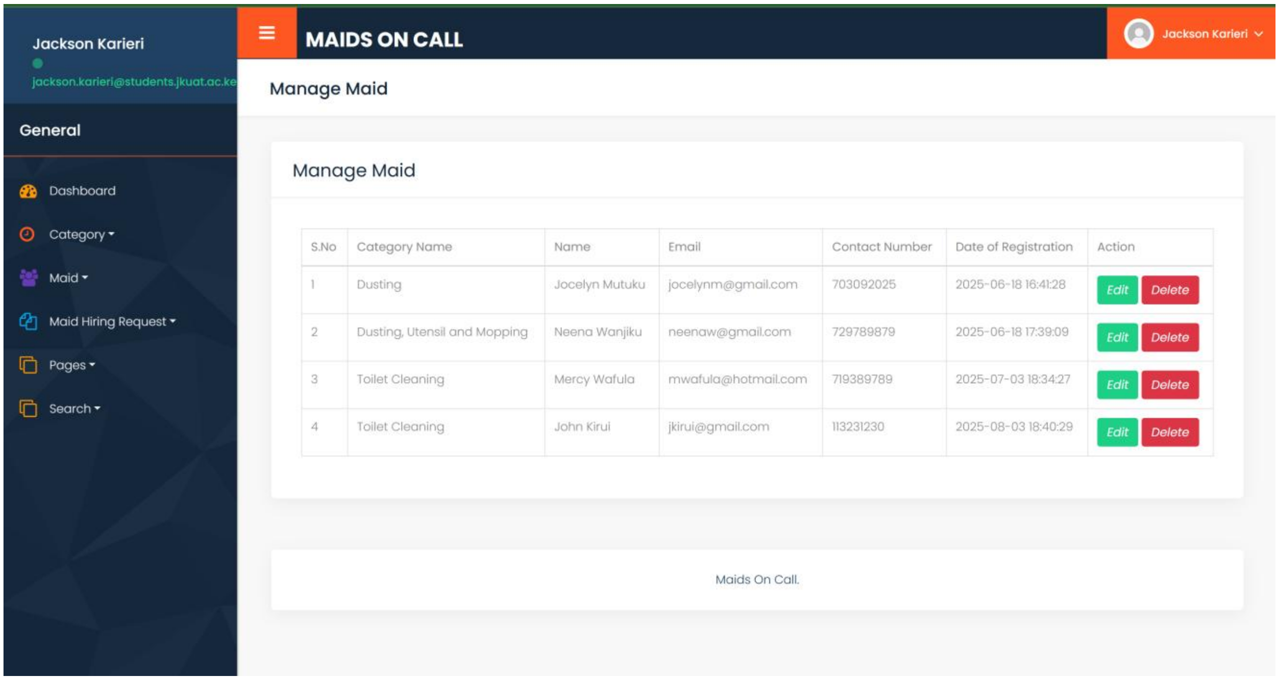
□ Remember Me

Admin Dashboard Page





Maids Page



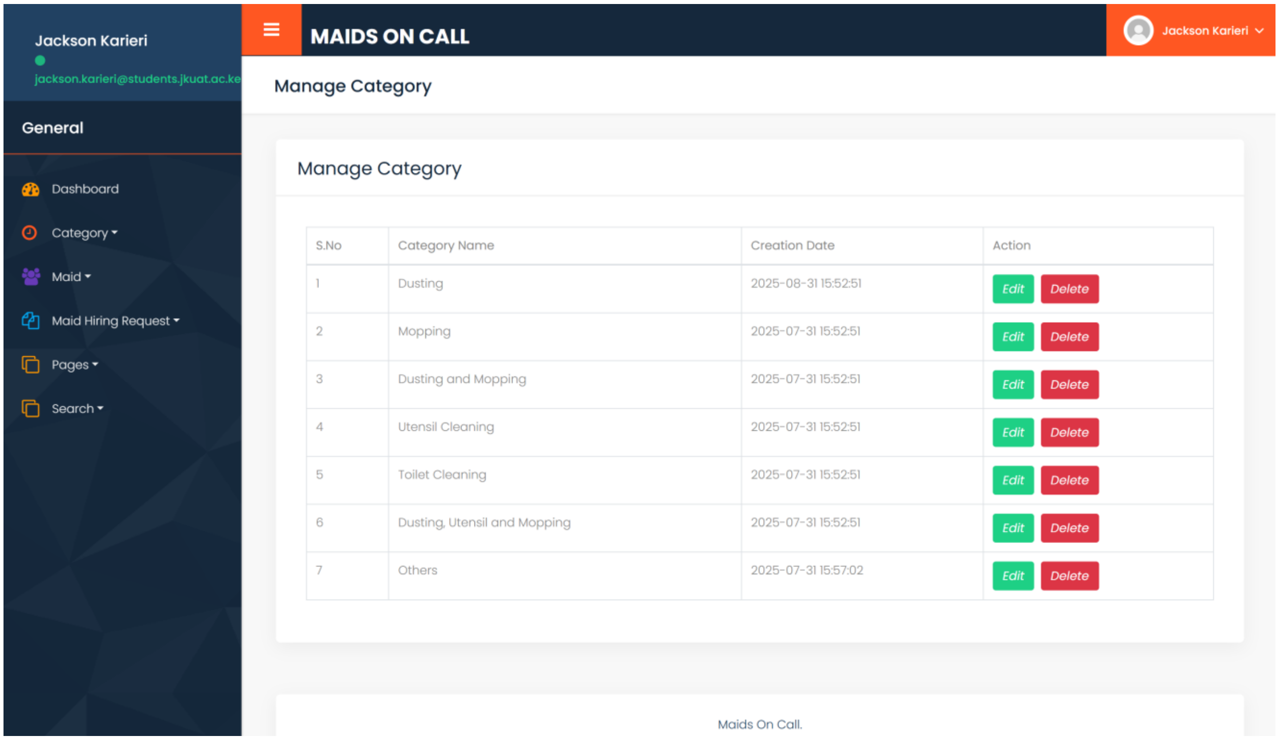
Manage Maid

Manage Maid

S.No	Category Name	Name	Email	Contact Number	Date of Registration	Action
1	Dusting	Jocelyn Mutuku	jocelynm@gmail.com	703092025	2025-06-18 16:41:28	Edit Delete
2	Dusting, Utensil and Mopping	Neena Wanjiku	neenaw@gmail.com	729789879	2025-06-18 17:39:09	Edit Delete
3	Toilet Cleaning	Mercy Wafula	mwafula@hotmail.com	719389789	2025-07-03 18:34:27	Edit Delete
4	Toilet Cleaning	John Kirui	jkirui@gmail.com	113231230	2025-08-03 18:40:29	Edit Delete

Maids On Call.

Category Page



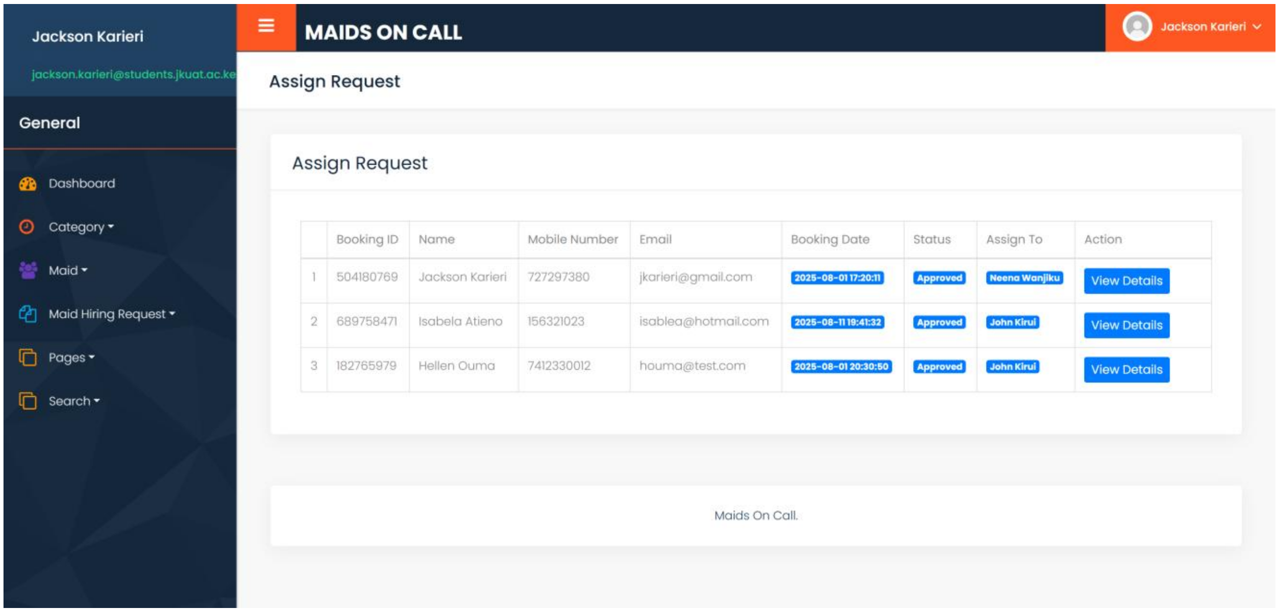
Manage Category

Manage Category

S.No	Category Name	Creation Date	Action
1	Dusting	2025-08-31 15:52:51	Edit Delete
2	Mopping	2025-07-31 15:52:51	Edit Delete
3	Dusting and Mopping	2025-07-31 15:52:51	Edit Delete
4	Utensil Cleaning	2025-07-31 15:52:51	Edit Delete
5	Toilet Cleaning	2025-07-31 15:52:51	Edit Delete
6	Dusting, Utensil and Mopping	2025-07-31 15:52:51	Edit Delete
7	Others	2025-07-31 15:57:02	Edit Delete

Maids On Call.

Assigned Request Page



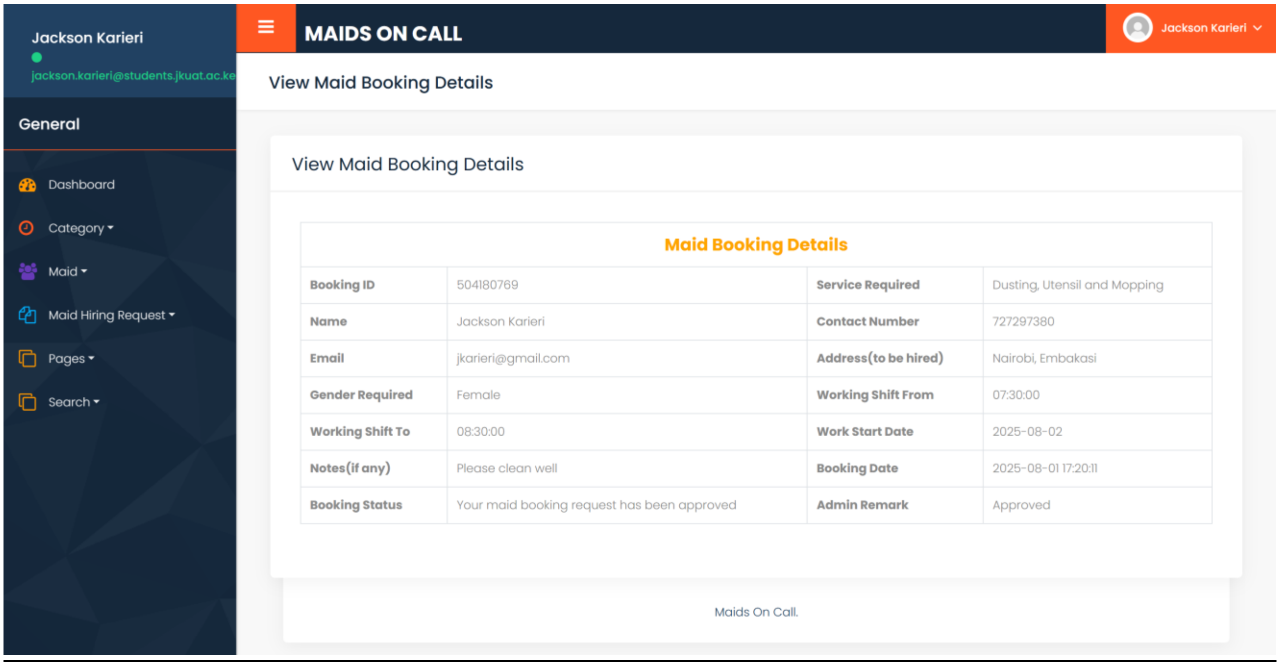
Assign Request

Assign Request

	Booking ID	Name	Mobile Number	Email	Booking Date	Status	Assign To	Action
1	504180769	Jackson Karieri	727297380	jkarieri@gmail.com	2025-08-01 17:20:11	Approved	Neena Wanjiku	View Details
2	689758471	Isabela Atieno	156321023	isablea@hotmail.com	2025-08-11 19:41:32	Approved	John Kirui	View Details
3	182765979	Hellen Ouma	7412330012	houma@test.com	2025-08-01 20:30:50	Approved	John Kirui	View Details

Maids On Call.

Maid Booking Details/Assigned



View Maid Booking Details

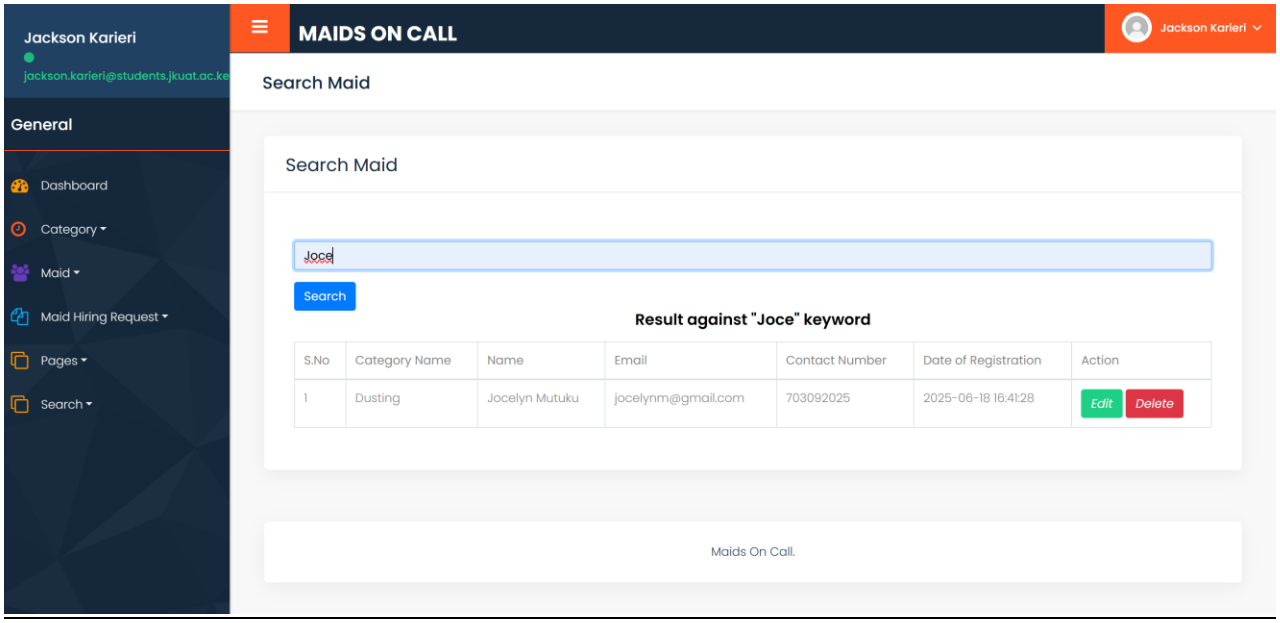
View Maid Booking Details

Maid Booking Details

Booking ID	504180769	**Service Required**	Dusting, Utensil and Mopping
Name	Jackson Karieri	**Contact Number**	727297380
Email	jkarieri@gmail.com	**Address(to be hired)**	Nairobi, Embakasi
Gender Required	Female	**Working Shift From**	07:30:00
Working Shift To	08:30:00	**Work Start Date**	2025-08-02
Notes(if any)	Please clean well	**Booking Date**	2025-08-01 17:20:11
Booking Status	Your maid booking request has been approved	**Admin Remark**	Approved

Maids On Call.

Search Functionality



Search Maid

Search Maid

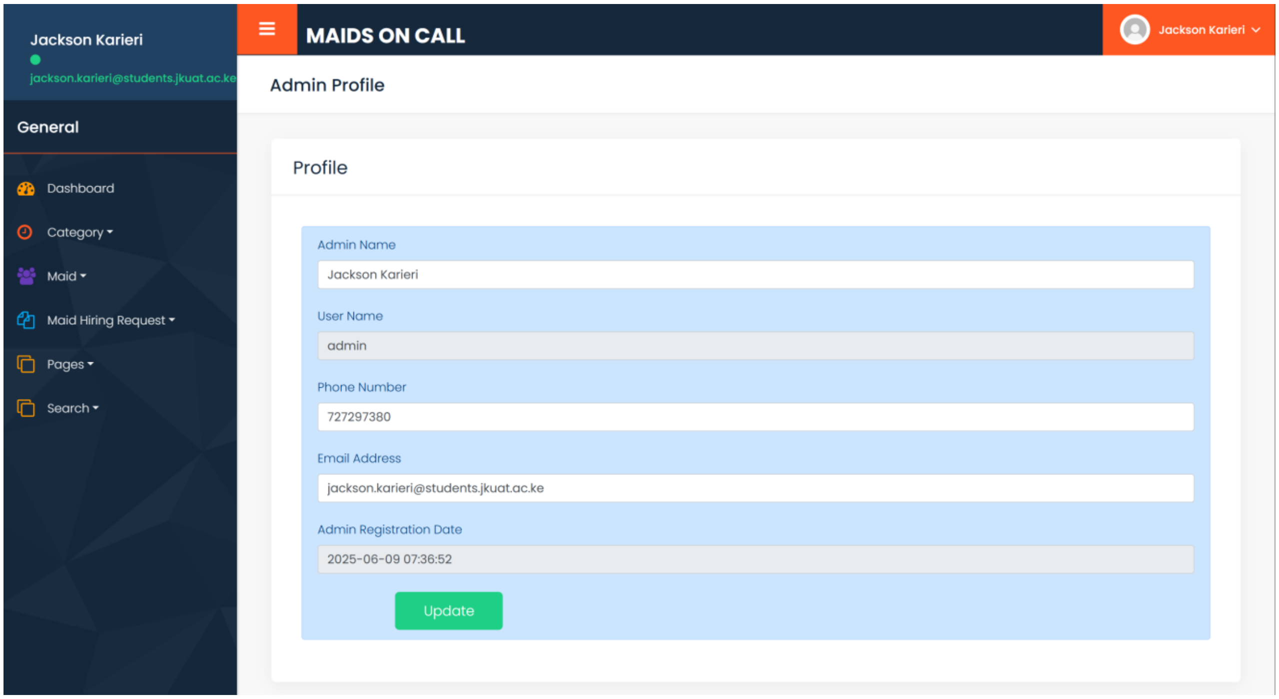
Joce

Result against "Joce" keyword

S.No	Category Name	Name	Email	Contact Number	Date of Registration	Action
1	Dusting	Jocelyn Mutuku	jocelynm@gmail.com	703092025	2025-06-18 16:41:28	Edit Delete

Maids On Call.

Admin Profile Page



Admin Profile

Profile

Admin Name
Jackson Karieri

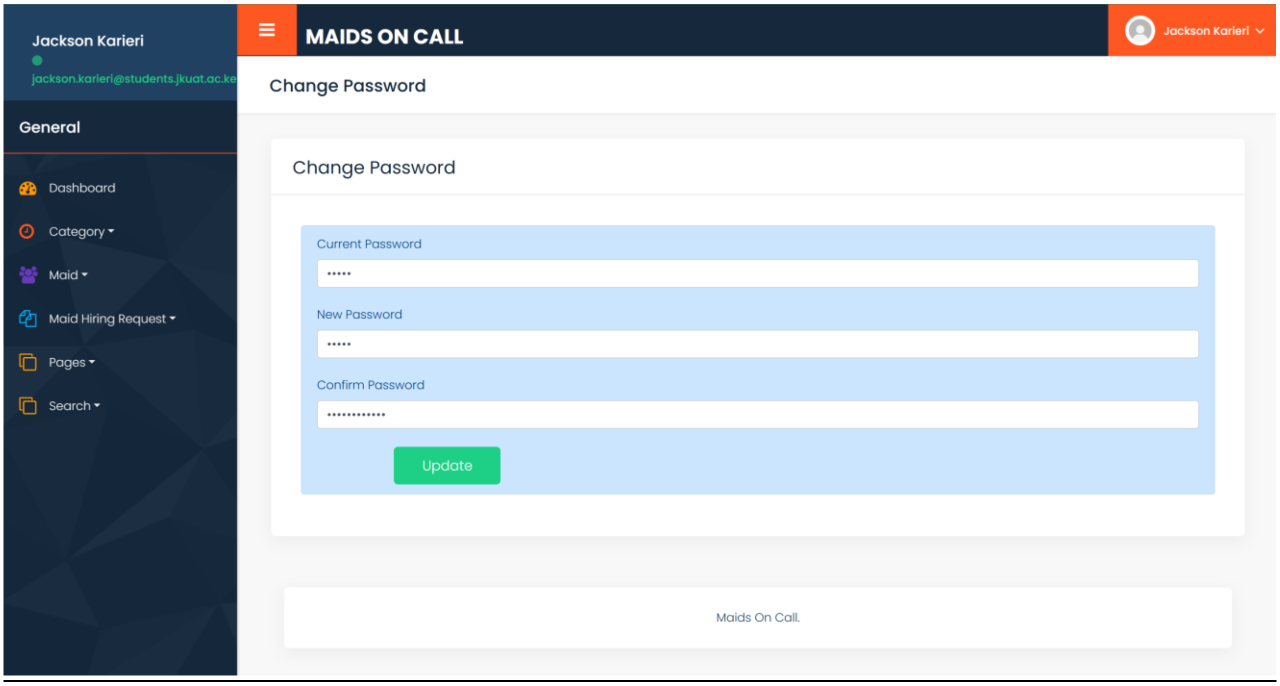
User Name
admin

Phone Number
727297380

Email Address
jackson.karieri@students.jkuat.ac.ke

Admin Registration Date
2025-06-09 07:36:52

Change Password Page



Current Password: •••••

New Password: •••••

Confirm Password: ••••••••••••

4.5 Summary

The implementation and testing of the Maids On Call System confirmed that the platform is functional, secure, and user-friendly. Unit, integration, and system testing validated the accuracy of modules and data flow. User Acceptance Testing demonstrated that the system met stakeholder needs, thereby ensuring its readiness for deployment. Although certain features, such as online payment integration, remain future enhancements, the current version delivers reliable maid booking, allocation, and communication functionalities.

CHAPTER 5

6.1 LIMITATIONS

The development and evaluation of the Maids On Call System encountered several challenges that influenced the Scope and outcomes of the project. A primary limitation was the time factor, as the project had to be completed within strict academic deadlines. This constraint restricted opportunities for extensive testing, user trials, and refinement of advanced features.

Another significant challenge was financial limitations. Limited resources affected the ability to fully integrate advanced modules such as artificial intelligence-driven personalization and large-scale real-time tracking. Additionally, the costs associated with cloud hosting and system deployment necessitated the adoption of a more basic system configuration during the initial stages.

6.2 CONCLUSION

The Maids On Call System successfully addresses the inefficiencies associated with the traditional manual management of maid bookings, allocations, and scheduling. By leveraging automation, cloud computing, and user-centered design, the system enhances efficiency, reduces administrative workload, and improves customer satisfaction.

From a theoretical perspective, the project contributes to the body of knowledge on business systems automation, demonstrating the value of integrating cloud-based and AI-supported solutions in service-oriented industries. From a practical perspective, it provides a working model that eliminates scheduling conflicts, improves communication, and ensures transparency in service delivery. Finally, at a policy level, the system offers insights that can guide organizations and policymakers toward formalizing and digitizing domestic service operations, ensuring greater accountability, efficiency, and client satisfaction.

In summary, the system bridges research, theory, and practice, offering a scalable and adaptable solution with the potential to transform maid service management in both residential and commercial contexts.

6.3 RECOMMENDATIONS

To strengthen the Maids On Call System and ensure its sustainability and scalability, several recommendations are proposed:

I. Extended Testing and Piloting – Future work should involve broader user trials across different environments and demographic groups to refine system usability, performance, and scalability.

II. Integration of Payment Gateways – Secure digital payment options such as M-Pesa, PayPal, or Stripe should be incorporated to simplify transactions and enhance user convenience.

III. Development of Mobile Applications – Dedicated Android and iOS applications should be created to increase accessibility, particularly for clients who primarily rely on mobile devices.

IV. Artificial Intelligence Enhancement – AI-powered features, such as intelligent maid recommendations based on booking history, feedback, and client preferences, should be implemented to increase personalization.

V. Strengthened Customer Support – The inclusion of live chat systems or AI-driven chatbots would provide real-time responses to client queries and improve overall customer experience.

VI. Data Security Improvements – As the system handles sensitive client information, enhanced security protocols such as advanced encryption and compliance with data protection policies should be adopted.

VII. Scalability and Market Expansion – The system should be designed for expansion beyond local use, with the ability to handle national or even international markets while maintaining reliability during peak demand.

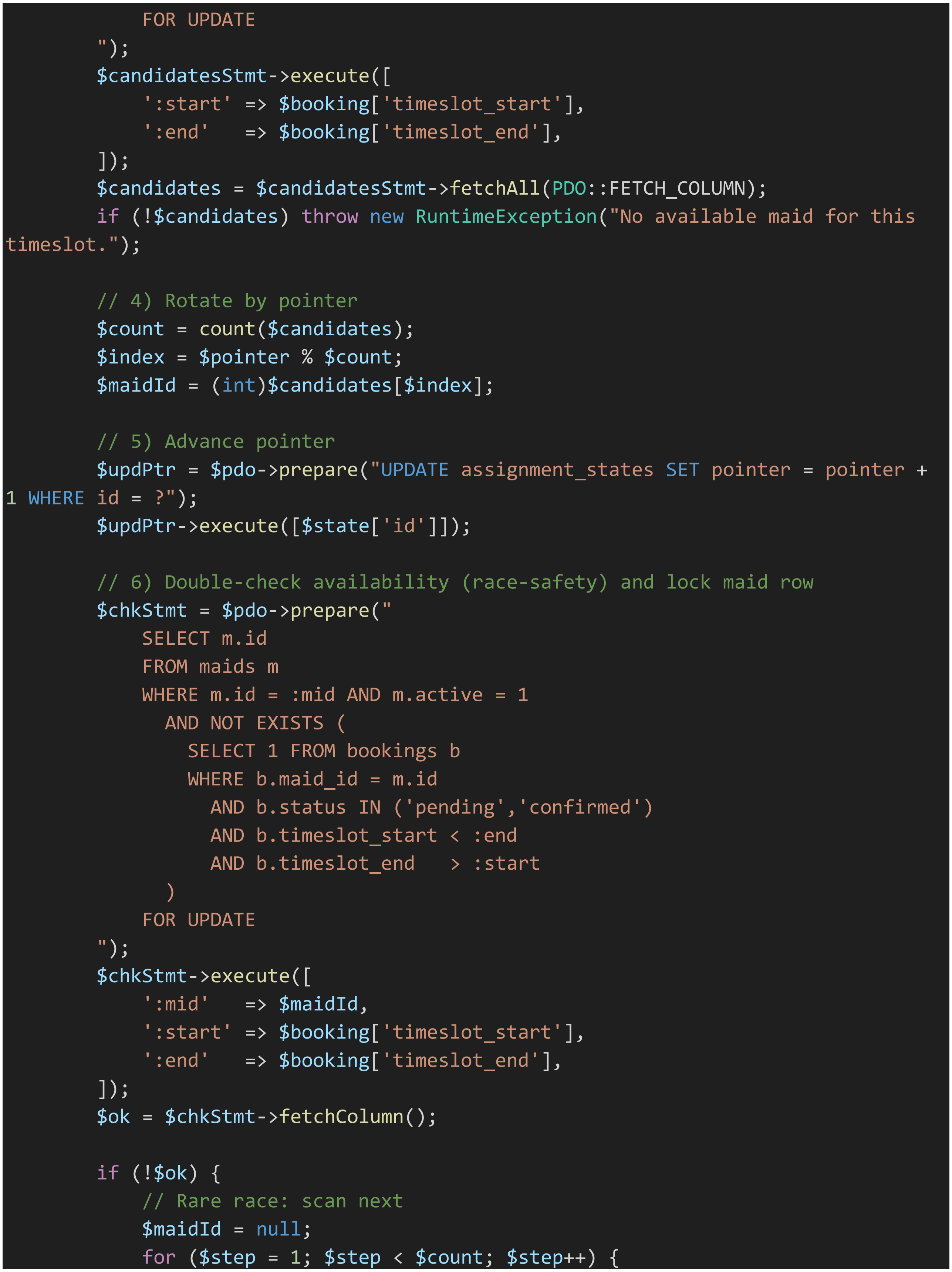
6.4 APPENDICES

Source Code on GitHub

Plain PHP (PDO) + MySQL — Round-robin auto-assignment for maid allocation to clients' requests

Minimal tables

```
-- maids
CREATE TABLE maids (
  id INT PRIMARY KEY AUTO_INCREMENT,
  active TINYINT(1) NOT NULL DEFAULT 1,
  rr_order INT NOT NULL,
  last_assigned_at DATETIME NULL,
  created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
);

-- bookings
CREATE TABLE bookings (
  id INT PRIMARY KEY AUTO_INCREMENT,
  service_id INT NULL,
  timeslot_start DATETIME NOT NULL,
  timeslot_end DATETIME NOT NULL,
  status ENUM('pending',' confirmed','canceled') NOT NULL DEFAULT 'pending',
  maid_id INT NULL
);

-- pointer (one row per pool; NULL = global pool)
CREATE TABLE assignment_states (
  id INT PRIMARY KEY AUTO_INCREMENT,
  service_id INT NULL UNIQUE,
  pointer INT NOT NULL DEFAULT 0
);
One-time init (order + pointer)
-- Give each maid a stable order
SET @rn := 0;
UPDATE maids SET rr_order = (@rn := @rn + 1) ORDER BY created_at, id;

-- Global pool pointer row
INSERT INTO assignment_states (service_id, pointer) VALUES (NULL, 0)
  ON DUPLICATE KEY UPDATE pointer = pointer;
PHP function
<?php
function assignRoundRobin(PDO $pdo, int $bookingId): array {
    $pdo->beginTransaction();
    try {
        // 1) Load & lock booking
        $stmt = $pdo->prepare("SELECT * FROM bookings WHERE id = ? FOR UPDATE");
        $stmt->execute([$bookingId]);
        $booking = $stmt->fetch(PDO::FETCH_ASSOC);
        if (!$booking) throw new RuntimeException("Booking not found.");

        // 2) Lock the pointer row (global pool here; use service_id if you want
per-service pools)
        $stateStmt = $pdo->query("SELECT * FROM assignment_states WHERE service_id
IS NULL FOR UPDATE");
        $state = $stateStmt->fetch(PDO::FETCH_ASSOC);
        if (!$state) throw new RuntimeException("Pointer state missing.");
        $pointer = (int)$state['pointer'];

        // 3) Build candidates: active + available in timeslot; ordered by
rr_order
        $candidatesStmt = $pdo->prepare("
            SELECT m.id
            FROM maids m
            WHERE m.active = 1
              AND NOT EXISTS (
                SELECT 1 FROM bookings b
                WHERE b.maid_id = m.id
                  AND b.status IN ('pending','confirmed')
                  AND b.timeslot_start < :end
                  AND b.timeslot_end   > :start
              )
            ORDER BY m.rr_order ASC
```



```
            FOR UPDATE
        ");
        $candidatesStmt->execute([
            ':start' => $booking['timeslot_start'],
            ':end'    => $booking['timeslot_end'],
        ]);
        $candidates = $candidatesStmt->fetchAll(PDO::FETCH_COLUMN);
        if (!$candidates) throw new RuntimeException("No available maid for this
timeslot.");

        // 4) Rotate by pointer
        $count = count($candidates);
        $index = $pointer % $count;
        $maidId = (int)$candidates[$index];

        // 5) Advance pointer
        $updPtr = $pdo->prepare("UPDATE assignment_states SET pointer = pointer +
1 WHERE id = ?");
        $updPtr->execute([$state['id']]);

        // 6) Double-check availability (race-safety) and lock maid row
        $chkStmt = $pdo->prepare("
            SELECT m.id
            FROM maids m
            WHERE m.id = :mid AND m.active = 1
              AND NOT EXISTS (
                SELECT 1 FROM bookings b
                WHERE b.maid_id = m.id
                  AND b.status IN ('pending','confirmed')
                  AND b.timeslot_start < :end
                  AND b.timeslot_end   > :start
              )
            FOR UPDATE
        ");
        $chkStmt->execute([
            ':mid'    => $maidId,
            ':start' => $booking['timeslot_start'],
            ':end'    => $booking['timeslot_end'],
        ]);
        $ok = $chkStmt->fetchColumn();

        if (!$ok) {
            // Rare race: scan next
            $maidId = null;
            for ($step = 1; $step < $count; $step++) {
                $tryId = (int)$candidates[($index + $step) % $count];
                $chkStmt->execute([
                    ':mid'   => $tryId,
                    ':start' => $booking['timeslot_start'],
                    ':end'   => $booking['timeslot_end'],
                ]);
                if ($chkStmt->fetchColumn()) { $maidId = $tryId; break; }
            }
            if (!$maidId) throw new RuntimeException("No maid available after re-
check.");
        }

        // 7) Persist assignment
        $updBook = $pdo->prepare("UPDATE bookings SET maid_id = ?, status =
'pending' WHERE id = ?");
        $updBook->execute([$maidId, $bookingId]);

        $updMaid = $pdo->prepare("UPDATE maids SET last_assigned_at = NOW() WHERE
id = ?");
        $updMaid->execute([$maidId]);

        $pdo->commit();
        return ['booking_id' => $bookingId, 'maid_id' => $maidId];
    } catch (Throwable $e) {
        $pdo->rollBack();
        throw $e;
    }
}
```

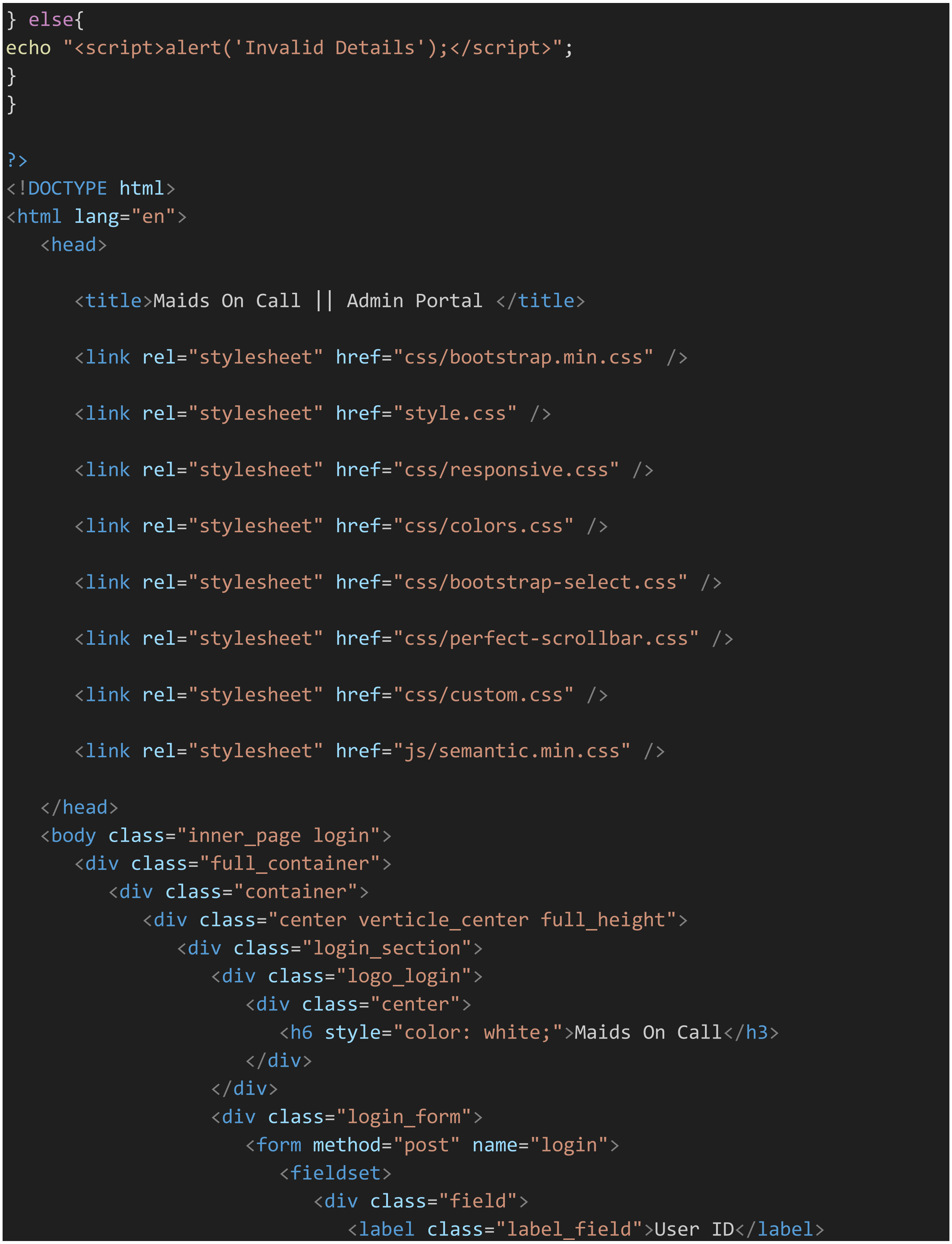
Database Connections Codes

```
<?php
// DB credentials.
define('DB_HOST','localhost');
define('DB_USER','root');
define('DB_PASS','');
define('DB_NAME','mhmsdb');
// Establish database connection.
try
{
$dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME,DB_USER,
DB_PASS,array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES 'utf8'"));
}
catch (PDOException $e)
```

```
{
exit("Error: " . $e->getMessage());
}
?>
```

Admin Login Codes

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');

if(isset($_POST['login']))
  {
    $username=$_POST['username'];
    $password=md5($_POST['password']);
    $sql ="SELECT ID FROM tbladmin WHERE UserName=:username and 
Password=:password";
    $query=$dbh->prepare($sql);
    $query-> bindParam(':username', $username, PDO::PARAM_STR);
$query-> bindParam(':password', $password, PDO::PARAM_STR);
    $query-> execute();
    $results=$query->fetchAll(PDO::FETCH_OBJ);
    if($query->rowCount() > 0)
{
foreach ($results as $result) {
$_SESSION['mhmsaid']=$result->ID;
}

  if(!empty($_POST["remember"])) {
//COOKIES for username
setcookie ("user_login",$_POST["username"],time()+ (10 * 365 * 24 * 60 * 60));
//COOKIES for password
setcookie ("userpassword",$_POST["password"],time()+ (10 * 365 * 24 * 60 * 60));
} else {
if(isset($_COOKIE["user_login"])) {
setcookie ("user_login","");
if(isset($_COOKIE["userpassword"])) {
setcookie ("userpassword","");
        }
      }
}
$_SESSION['login']=$_POST['username'];
echo "<script type='text/javascript'> document.location ='dashboard.php';
</script>";
```



```
} else{
echo "<script>alert('Invalid Details');</script>";
}
}

?>
<!DOCTYPE html>
<html lang="en">
   <head>

      <title>Maids On Call || Admin Portal </title>

      <link rel="stylesheet" href="css/bootstrap.min.css" />

      <link rel="stylesheet" href="style.css" />

      <link rel="stylesheet" href="css/responsive.css" />

      <link rel="stylesheet" href="css/colors.css" />

      <link rel="stylesheet" href="css/bootstrap-select.css" />

      <link rel="stylesheet" href="css/perfect-scrollbar.css" />

      <link rel="stylesheet" href="css/custom.css" />

      <link rel="stylesheet" href="js/semantic.min.css" />

   </head>
   <body class="inner_page login">
      <div class="full_container">
         <div class="container">
            <div class="center verticle_center full_height">
               <div class="login_section">
                  <div class="logo_login">
                     <div class="center">
                        <h6 style="color: white;">Maids On Call</h3>
                     </div>
                  </div>
                  <div class="login_form">
                     <form method="post" name="login">
                        <fieldset>
                           <div class="field">
                              <label class="label_field">User ID</label>
                              <input type="text" class="form-control" placeholder="enter your username" required="true" name="username" value="<?php if(isset($_COOKIE["user_login"])) { echo $_COOKIE["user_login"]; } ?>" >
                           </div>
                           <div class="field">
                              <label class="label_field">Password</label>
                              <input type="password" class="form-control" placeholder="enter your password" name="password" required="true" value="<?php if(isset($_COOKIE["userpassword"])) { echo $_COOKIE["userpassword"]; } ?>">
                           </div>
                           <div class="field">
                              <label class="label_field hidden">hidden label</label>
                              <label class="form-check-label"><input class="form-check-input" id="remember" name="remember" <?php if(isset($_COOKIE["user_login"])) { ?> checked <?php } ?> type="checkbox"/> Remember Me</label>
                              <a class="forgot" href="forgot-password.php">Forgotten Password?</a>
                           </div>
                           <div class="field margin_0">
                              <label class="label_field hidden">hidden label</label>
                              <button class="main_bt" name="login" type="submit">Login</button>
                           </div>
                        </fieldset>
                        <a class="forgot" href="../index.php">Home Page</a>
                     </form>
                  </div>
               </div>
            </div>
         </div>
      </div>

      <script src="js/jquery.min.js"></script>
      <script src="js/popper.min.js"></script>
      <script src="js/bootstrap.min.js"></script>

      <script src="js/animate.js"></script>

      <script src="js/bootstrap-select.js"></script>

      <script src="js/perfect-scrollbar.min.js"></script>
      <script>
         var ps = new PerfectScrollbar('#sidebar');
      </script>

      <script src="js/custom.js"></script>
   </body>
</html>
```

Logout Session Codes

```
<?php
session_start();
session_unset();
session_destroy();
header('location:login.php');

?>
```

CHAPTER 7

REFERENCES

Apache Friends. (n.d.). XAMPP download. Retrieved from https://www.apachefriends.org/download.html

MySQL. (n.d.). MySQL official website. Retrieved from https://www.mysql.com/

MySQL Tutorial. (n.d.). MySQL tutorial. Retrieved from http://www.mysqltutorial.org

PHP Group. (n.d.). PHP official documentation. Retrieved from https://www.php.net/

SitePoint. (n.d.). PHP tutorials. Retrieved from https://www.sitepoint.com/php/

W3Schools. (n.d.). PHP tutorial. Retrieved from https://www.w3schools.com/php/default.asp