

Project 3

User documentation and system documentation

Ready RMS

A RMS (Restaurant-management system) application

Course	CPUT, DIPICT, Application Development
Subject name	Project
Subject code	Project362S
Supervisor	Mr. Temuso Makhurane
Group ID	TM1
Due date	21.10.25

Team members

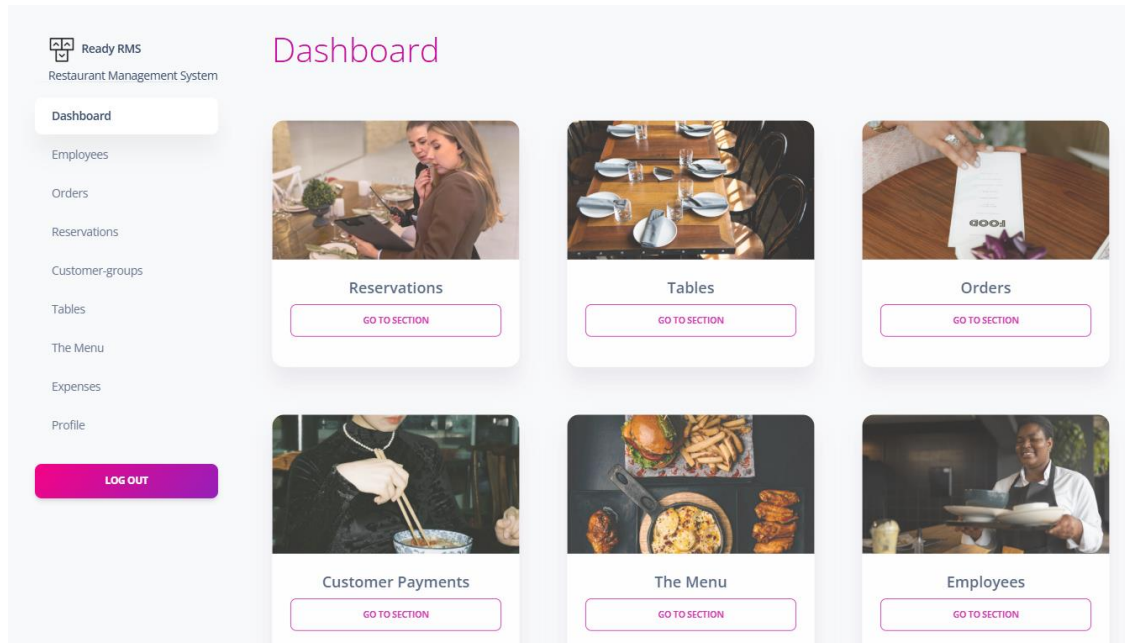
Name and surname	Student number
Ammaar Swartland	230159478
Kyran Gajoo	230971229
Salaamah Peck	230207170
Leslie Van Niel	230453260
Esethu Gift Mehlo	216125723

Table of Contents

User documentation.....	3
Home/Dashboard section.....	3
Add-record section.....	4
Edit-record sections	5
Reservation section.....	6
Payment section	9
Table section	12
.....	12
Order section	13
Menu section.....	14
Employee section	15
Employee-shift section	16
Employee-salary section	18
Expense section.....	20
System documentation.....	22
System overview.....	22
Problem definition.....	22
Functional Requirements.....	23
Non-functional Requirements	23
Solution proposal.....	24
Database-design.....	25
ERD.....	25
Entities (tables)	26
Tools and software used.....	27
Front-end	27
Backend	27
Version control systems and collaboration	28
Challenges.....	29
Achievements.....	31
Potential improvements	33

User documentation

Home/Dashboard section

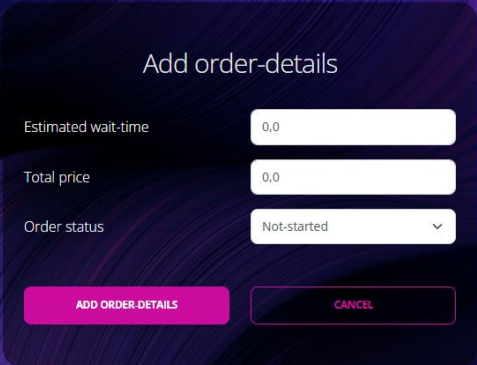


The 'Home section' contains links to the various pages of the Restaurant-management system. There are links in the form of cards. There is also a side-navbar with links to the rest of the app.

A 'log out' button can be seen at the bottom of the sidebar. This is used to log out of the app and return to the 'sign up' section.

Add-record section

These sections are used to add new records to the database. They all follow the same design.

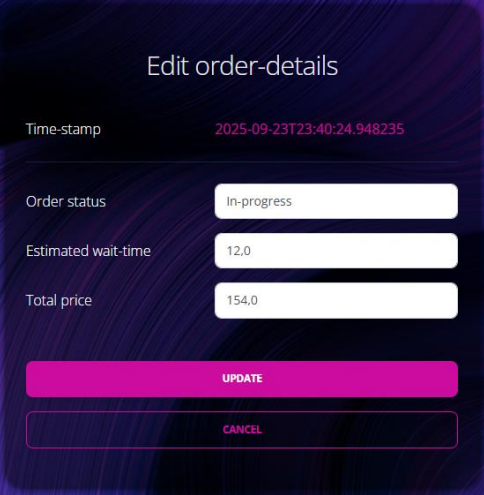


The image shows a modal form titled "Add order-details" centered on a dark background with a vibrant, wavy purple and blue pattern. The form is a dark gray rectangle with rounded corners. It contains three input fields: "Estimated wait-time" and "Total price" are text inputs with "0,0" entered; "Order status" is a dropdown menu with "Not-started" selected. At the bottom are two buttons: a solid magenta "ADD ORDER-DETAILS" button and a white "CANCEL" button with a magenta border.

Add order-details	
Estimated wait-time	<input type="text" value="0,0"/>
Total price	<input type="text" value="0,0"/>
Order status	<input type="text" value="Not-started"/>
<div><div>ADD ORDER-DETAILS</div><div>CANCEL</div></div>	

Edit-record sections

These sections are used to update existing records that are stored in the database. They all follow the same design.



The image shows a modal form titled "Edit order-details" centered on a dark background with a purple and blue wavy pattern. The form is a dark gray rectangle with rounded corners. It contains the following fields and buttons:

- Title:** Edit order-details
- Time-stamp:** 2025-09-23T23:40:24.948235
- Order status:** In-progress
- Estimated wait-time:** 12,0
- Total price:** 154,0
- Buttons:** A solid magenta "UPDATE" button and a magenta-outlined "CANCEL" button.

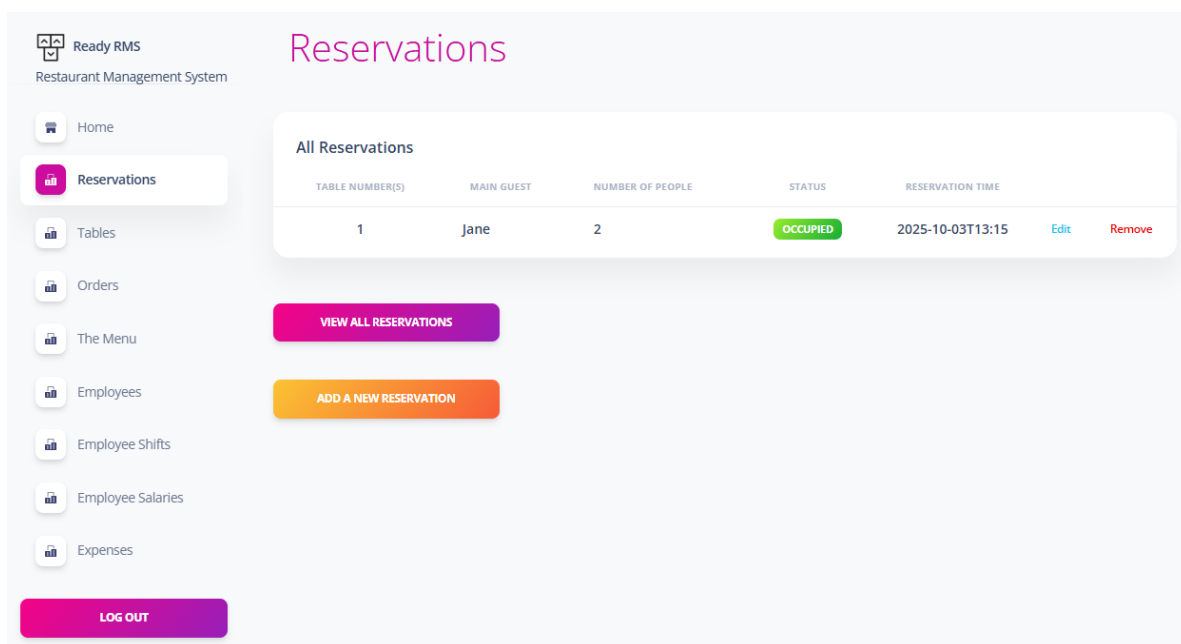
Field	Value
Time-stamp	2025-09-23T23:40:24.948235
Order status	In-progress
Estimated wait-time	12,0
Total price	154,0

Reservation section

The Reservation section handles all customer reservation processes in the Restaurant Management System.

The core functions include:

- Creating, viewing, editing and deleting reservations and their associated customers details.
- Validating for available tables, reservation time and dates and customer group details.



Viewing All Reservations

1. From the Home dashboard or side menu select the Reservations option.
2. The main Reservations page with all reservations from the system will be displayed. Details displayed will include:
 - The name of the customer who made the reservation
 - The table number of the reservation
 - Reservation date and time
 - The number of people in the customer group
3. Buttons to Edit or Remove a particular reservation are located at the end of each reservation record

4. The Add new reservation button allows the user to add a new reservation.
5. The View All reservation button will load a page with all details of the reservations, and includes:
 - The All Payments button: which will display all the payments entered in the system.
 - The Add Payment button at the end of each reservation: allows a user to add a payment to the selected reservation.

Adding a Reservation

1. On the main reservation page click the Add a new Reservation button
2. A form to add the details of the new reservation will be displayed with the following fields:
 - Main Guest (Payer) Information
 - First name: refers to the first name of the customer making the reservation
 - Last name: refers to the last name/surname of the customer making the reservation
 - Phone number: refers to the contact number of the customer making the reservation
 - Reservation Information
 - Date and Time: the date and time of the reservation
 - Maximum Time: refers to the maximum time, in minutes, the customer/customer group may occupy the reserved tables
 - Start Time: the time the reservation period commences
 - End Time: the time the reservation period concludes
 - Table Information
 - Table Number: the number of the restaurant table reserved for the reservations (selected from a dropdown list of all restaurant tables in the system)
 - Number of people: refers to the total number of people included in the reservation.

3. Once all field are correctly completed, click the Add Reservation button to save the add the reservation to the system.
4. If successfully saved, the page will be redirected to the main reservation page. The added reservations will be displayed along all the other reservations.
5. The Cancel button can be clicked at any point to leave the add reservation form and return to the main reservation page without saving.

Editing an existing reservation

1. On the main reservation page, click the Edit button at the end of the reservation which needs to be edited.
2. The page will display the Edit Reservation form with the existing details from the selected reservation.
3. Update the necessary fields.
4. Once the required details have been modified click the Update button to save the changes.
5. If the reservation has been updated and saved successfully, the page will display the Reservation list which will reflect the updated details.
6. The Cancel button can be clicked at any time to discard any changed and return to the main reservations page.

Deleting a reservation

1. On the main reservation page, click the Remove button at the end of the reservation which needs to be deleted.
2. A prompt will appear to confirm the action of deleting the selected reservation
3. Click Cancel to void the action of deleting the selected reservation.
4. Click Ok to confirm that the reservation must be permanently deleted.
5. Once confirmed, the reservation will be deleted.
6. The page will automatically load all the remaining reservations

Payment section

Adding a Payment

1. On the main reservation page click the View All Reservations button, the detailed Reservation list page will display.
2. Locate the reservation which the payment must be added to.
3. Click Add Payment at the end of the reservations.
4. A form to add the details of the new payment will be displayed with the following fields:
 - Date and time of payment
 - Amount: refers to the total amount for the customer of the selected reservation
 - Transaction Type: refers to the method of payment, Cash/Card etc. (Selected from a dropdown)
5. Once all field are correctly completed, click the Add Payment Details button to save the add the payment to the system.
6. If successfully saved, the page will be return to the Customer Payments page with all payments in the system. The added payment will be displayed along all the other payments.
7. The Cancel button can be clicked at any point to leave the add payment form and return to the detailed reservation page without saving.

Editing a payment

1. On the main reservation page click the View All Reservations button, the detailed Reservation list page load and display
2. Click the All Payments button, the page displaying all payments in the system will load.
3. Locate the payment which requires modification and click the Edit button at the end of the payment.
4. The page will display the Edit Payment details form with the existing details from the selected payment.
5. Update the necessary fields.
6. Once the required details have been modified click the Update button to save the changes.

7. If the payment has been updated and saved successfully, the page will display the Customer Payments page which will reflect the updated details .
8. The Cancel button can be clicked at any time to discard any changed and return to the Customer Payments page.

Deleting a payment

1. On the main reservation page click the View All Reservations button, the detailed Reservation list page load and display.
2. Click the All Payments button, the page displaying all payments in the system will load.
3. Locate the payment which requires deletion and click the Remove button at the end of the payment.
4. A prompt will appear to confirm the action of deleting the selected payment
5. Click Cancel to void the action of deleting the selected payment.
6. Click Ok to confirm that the payment must be permanently deleted.
7. Once confirmed, the payment will be deleted.
8. The page will automatically load all the remaining payments

Error Handling

1. Missing Fields

If any required fields are not completed an error message will display, alerting that all required fields must be filled in add the reservation

2. Phone number:

If the phone number entered is not a valid contact number, the following error message will appear:

“Phone number must be valid”

A valid phone number must be entered to add the reservation

3. Reservation Date is in the past

Should the selected date/time be before the current date/time, the following error message will appear:

“Reservation date/time is before he current date.”

To successfully add the reservation, select a date and time which falls after the current date and time.

4. Time Order Error

If the End Time of the reservation is before the Start Time of the reservation, the following error message will appear:

“End time must be after start time”.

To successfully add the reservation, select an end time which is after the start time.

5. Maximum Duration Error

The time between the Start Time and the End Time should not exceed or be greater than the set Maximum Time, if it does, the following error message will appear:

“Reservation time exceed maximum time limit”

To successfully add the reservation, ensure the time between the start and end of the reservation is within set maximum time.

6. Duplication Errors:

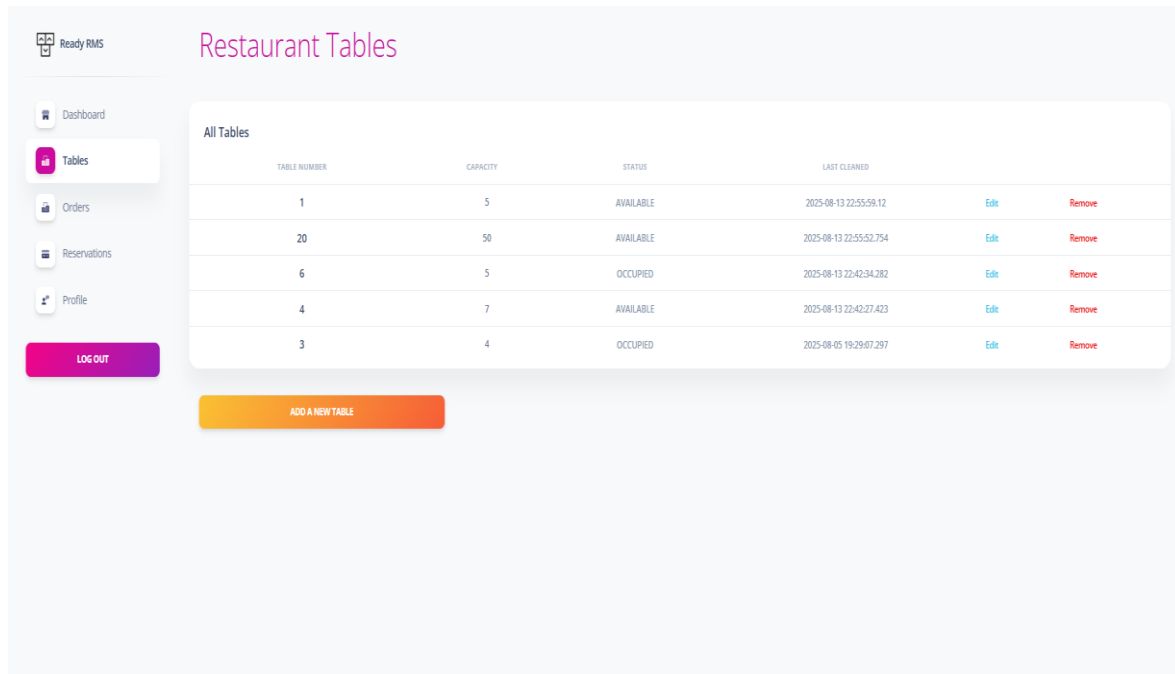
If a table has already been reserved for the selected date/time, the following error message will appear:

“Table already reserved for selected date/time”

To successfully add the reservation, select an alternative table.

Table section

The table section shows you all the available tables with relevant information for easier seating and record keeping



Viewing all records from a table

- All the table records will be visible upon access of this section

Inserting a record

1. Click the orange 'add a new table' button underneath the list of records.
2. Add all details in the form
3. Click the 'save' button at the bottom of the form to save the record to the database

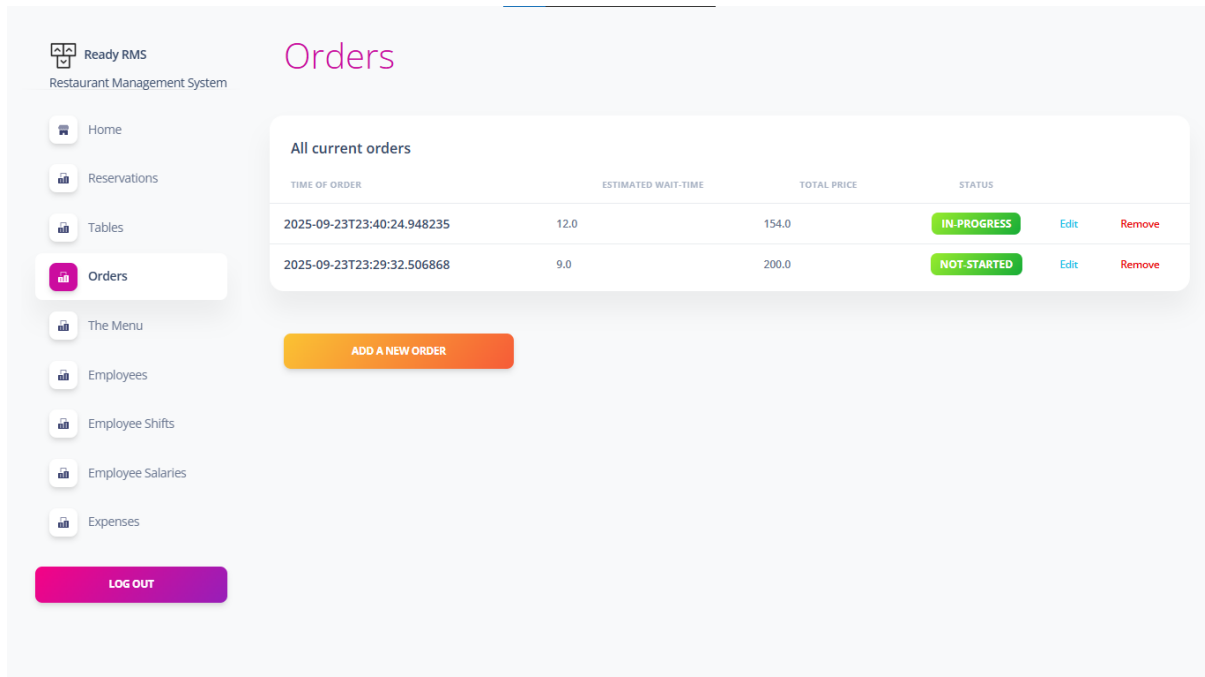
Deleting a record

1. Click the word 'remove' (red colour) on the far-right side of the record
2. Confirm for the deletion of the record

Updating a record

1. Click the word 'edit' (blue) on the far-right side of the record, to the left of the record 'remove' option
2. Update the details
3. Click the 'update' button at the bottom of the form to update the record

Order section



Viewing all records from a table

- All the records (order records) will be visible when accessing this section

Inserting a record

1. Click the 'add a new order' button
2. Add all the information in the form
3. Click the button at the bottom of the form to save the record to the database

Deleting a record

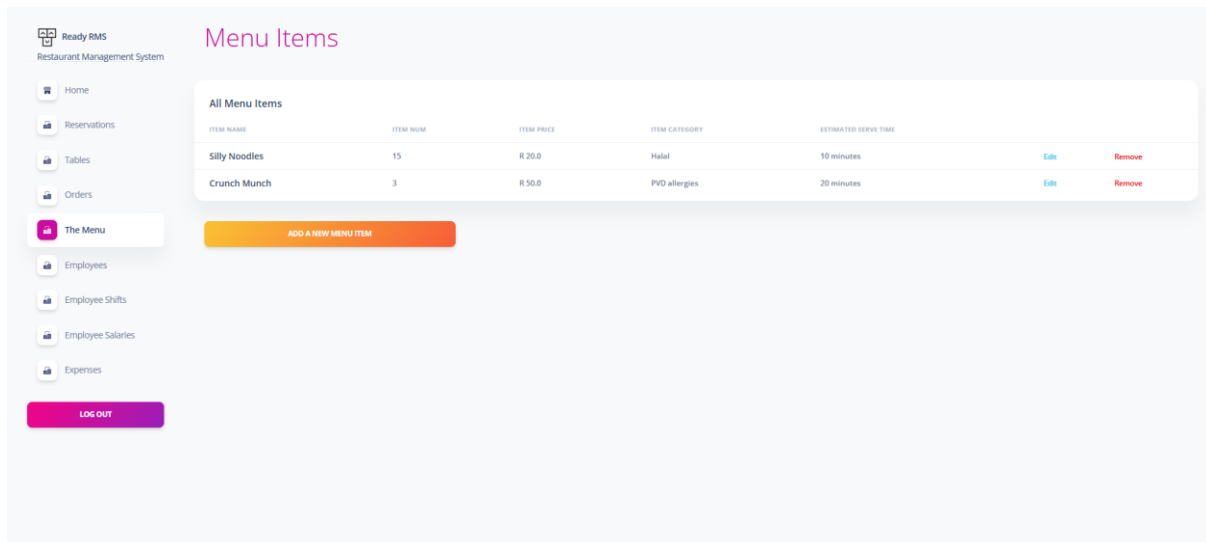
1. Click the word 'remove' (red) on the far-right side of the record
2. Confirm the deletion of the record

Updating a record

1. Click the word 'edit' (blue) on the far-right side of the record, to the left of the record 'remove' option
2. Update the details
3. Click the button at the bottom of the form to update the record

Menu section

The menu section displays menu items and additives and the time it takes to prepare them as well as other relevant information.



Viewing all records from a table

- All the menu records will be visible upon access of this section

Inserting a record

- 1 Click the orange 'add a new menu' button underneath the list of records.
- 2 Add all details in the form
- 3 Click the 'save' button at the bottom of the form to save the record to the database

Deleting a record

- 1 Click the word 'remove' (red colour) on the far-right side of the record
- 2 Confirm for the deletion of the record

Updating a record

- 1 Click the word 'edit' (blue) on the far-right side of the record, to the left of the record 'remove' option
- 2 Update the details

Employee section

All Employees						
EMPLOYEE	ROLE	PHONE-NUMBER	EMAIL-ADDRESS	CREATION-DATE		
fred barnes	Manager	7285252	barnes@gmail.com	2025-09-24 14:26:53.619	Edit	Remove
first_name last_name	manager			2025-09-24 14:21:43.851	Edit	Remove
first_name last_name	manager			2025-09-24 14:17:36.801	Edit	Remove
first_name last_name	manager			2025-09-24 14:01:47.985	Edit	Remove
first_name last_name	manager			2025-09-24 13:58:40.495	Edit	Remove
first_name last_name	manager			2025-09-24 13:58:25.933	Edit	Remove
first_name last_name	manager			2025-09-24 13:57:30.315	Edit	Remove
first_name last_name	manager			2025-09-24 13:51:12.88	Edit	Remove
oob, loob,	Kitchen-staff member		boo@gmail.com	2025-09-24 12:59:54.572	Edit	Remove

Viewing all records from a table

- All the records (order records) will be visible when accessing this section

Inserting a record

- 1 Click the 'add a new employee' button underneath the list of records.
- 2 Add all the information in the form
- 3 Click the button at the bottom of the form to save the record to the database

Deleting a record

- 1 Click the word 'remove' (red) on the far-right side of the record
- 2 Confirm the deletion of the record

Updating a record

- 1 Click the word 'edit' (blue) on the far-right side of the record, to the left of the record 'remove' option
- 2 Update the details
- 3 Click the button at the bottom of the form to update the record

Employee-shift section

Ready RMS
Restaurant Management System

Employee Shifts

All Employee Shifts

DAY AND DATE OF CREATION	START-TIME	END-TIME		
2025-10-10T19:04:06.994364	2025-10-10T19:00	2025-10-10T23:00	Edit	Remove
2025-10-06T22:53:43.192308	2025-10-06T22:53	2025-10-06T01:56	Edit	Remove
2025-10-05T00:00	2025-10-05T08:00	2025-10-05T16:00	Edit	Remove
2025-10-05T00:00	2025-10-05T12:00	2025-10-05T20:00	Edit	Remove
2025-10-06T00:00	2025-10-06T10:00	2025-10-06T18:00	Edit	Remove
2025-10-06T00:00	2025-10-06T14:00	2025-10-06T22:00	Edit	Remove
2025-10-06T16:39:45.059697	2025-10-06T16:39	2025-10-06T19:39	Edit	Remove

ADD A NEW SHIFT

LOG OUT

Viewing All Shifts

- From the Home dashboard or side menu select the Employee Shifts icon.

Adding a shift

- 1 On the main Shifts page click the Add a new Shift button
- 2 A form to add the details of the new shift will be displayed in the following fields:
 - Start Time(Employee) Information
 - End Time(Employee) Information(NB. - Maximum time of a shift should be 8 hours.)
- 3 Once all fields are correctly completed, click the Add Shift button to save the add the Shift to the system.
- 4 If successfully saved, the page will be redirected to the main shift page. The added shift will be displayed throughout all the other shifts.
- 5 The Cancel button can be clicked at any point to leave the add shift form and return to the main Shift page without saving.

Editing an existing Shift

- 1 On the main shift page, click the Edit button at the end of the shift which needs to be edited.

- 2 The page will display the Edit Shift form with the existing details from the selected shift.
- 3 Update the necessary fields.
- 4 Once the required details have been modified click the Update button to save the changes.
- 5 If the shift has been updated and saved successfully, the page will display the Shift list which will reflect the updated details.
- 6 The Cancel button can be clicked at any time to discard any change and return to the main shifts page.

Deleting a shift

- 1 On the main shift page, click the Remove button at the end of the shift which needs to be deleted.
- 2 A prompt will appear to confirm the action of deleting the selected shift
- 3 Click Cancel to avoid deleting the selected shift.
- 4 Click Ok to confirm that the shift must be permanently deleted.
- 5 Once confirmed, the shift will be deleted.
- 6 The page will automatically load all the remaining shifts

Employee-salary section

Purpose of the Employee Salary Module

The Employee Salary module is responsible for managing salary records of employees within the Restaurant Management System (RMS). It allows administrators to perform CRUD (Create, Read, Update, Delete) operations on salary details. The module ensures that employee salaries, payment methods, and payment history are accurately recorded and easily retrievable.

Salaries				
All Salaries				
PAYMENT-METHOD	AMOUNT	LAST PAYMENT	EMPLOYEE ID	
Cash-staff	3000.0	2025-09-28	7	Edit Remove
EFT	45000.0	2025-09-15	6	Edit Remove
EFT	40000.0	2025-10-04	5	Edit Remove

ADD NEW SALARY DETAILS

LOG OUT

This screenshot shows the main Employee Salaries table in the Restaurant Management System. It lists all existing salary records with columns for Payment Method, Amount, Last Payment Date, and Employee ID. From here, users can Edit or Remove records.

Viewing all records from a table

- 1 Navigate to the 'Employee Salaries' page.
- 2 The list is displayed in a table format on the UI.

Inserting a record

- 1 Click 'Add New Salary Details'.
- 2 Select the employee, enter the payment date, amount, and method.
- 3 Submit the data by clicking the 'add' button.

Updating a record

1. Click 'Edit' next to a salary record.
2. The edit form preloads existing data.

3. Update the fields and submit.

Deleting a record

1. Click 'Remove' next to a salary record.
2. The page refreshes showing updated data without the deleted entry.

Summary

The Employee Salary module integrates multiple layers of the application: the model for persistence, DTO for data validation, repository for database operations, controller for business logic, and Thyme leaf templates for UI. Together, they enable smooth management of employee salary details within the RMS.

Expense section

The screenshot displays the 'Ready RMS Restaurant Management System' interface. On the left is a sidebar with navigation links: Home, Reservations, Tables, Orders, The Menu, Employees, Employee Shifts, Employee Salaries, and Expenses (highlighted in pink). Below these links is a 'LOG OUT' button. The main content area is titled 'Expenses' in pink. It features a table titled 'All Expenses' with columns for DATE, NAME, and AMOUNT. The table lists five expenses: Kitchen Supplies (150.0), Cleaning Services (85.5), Meat Supplier (300.75), Beverage Inventory (210.0), and Electricity Bill (120.25). Each row has an 'Edit' link. Below the table is an orange button labeled 'ADD A NEW EXPENSE'.

DATE	NAME	AMOUNT	
2025-10-05T00:00	Kitchen Supplies	150.0	Edit
2025-10-04T00:00	Cleaning Services	85.5	Edit
2025-10-03T00:00	Meat Supplier	300.75	Edit
2025-10-02T00:00	Beverage Inventory	210.0	Edit
2025-10-01T00:00	Electricity Bill	120.25	Edit

Adding an Expense

- 1 On the main shift page click the View All Shifts button, the detailed Shift list page will display.
- 2 Locate the shift which the Expense must be added to.
- 3 Click Add Expense at the end of the shifts.
- 4 A form to add the details of the new Expense will be displayed with the following fields:
 - a. Date and time of Expense
 - b. Amount: refers to the total amount for the selected Expense
- 5 If successfully saved, the page will be return to the Expenses page with all Expenses in the system. The added Expense will be displayed along all the other Expenses.
- 6 The Cancel button can be clicked at any point to leave the add Expense form and return to the detailed shift page without saving.

Editing a Expense

- 1 On the main shift page click the Expenses button, the page displaying all expenses in the system will load.
- 2 Locate the Expense which requires modification and click the Edit button at the end of the Expense.

- 3 The page will display the Edit Expense details form with the existing details from the selected Expense.
- 4 Update the necessary fields.
- 5 Once the required details have been modified click the Update button to save the changes.
- 6 If the Expense has been updated and saved successfully, the page will display the Customer Expenses page which will reflect the updated details .
- 7 The Cancel button can be clicked at any time to discard any changed and return to the Customer Expenses page.

Deleting a Expense

- 1 On the main shift page click the Expenses button, the page displaying all expenses in the system will load.
- 2 Locate the Expense which requires deletion and click the Remove button at the end of the Expense.
- 3 A prompt will appear to confirm the action of deleting the selected Expense
- 4 Click Cancel to void the action of deleting the selected Expense.
- 5 Click Ok to confirm that the Expense must be permanently deleted.
- 6 Once confirmed, the Expense will be deleted.
- 7 The page will automatically load all the remaining Expenses

System documentation

System overview

The RMS (Restaurant-management system) is used to manage restaurants more effectively. This system will enable the business **manage** customers and customer-transactions **more efficiently**, and conduct analysis on the menu and the customers. This means that **the business can more informed decisions and adjustments** to their menu, pricing of food-items, ingredient-expenses, and other business tactics.

Problem definition

Smaller-scale restaurants may still record transactions from customers on pieces of paper. There is difficulty in recording the transactions, and calculating revenue for the business, because of all the papers, related to each customer-transaction, that need to be stored.

A huge portion of smaller-scale restaurants still record their transactions with customers manually on pieces of paper. In turn, this paper-based approach to storage of important receipts often leads to a multitude of operational issues.

These issues include:

- unnecessary consumption of space and time consuming, because of the process of physically storing all the papers related to each customer transaction
- a high level of difficulty keeping track of all the transactions and attempting to gather and do calculations on them
- unnecessary inefficiency by not having the ability to view data in real-time in a technologically advanced world, slowing the decision-making process

The main users who will benefit from using this system are the:

- busboys
 - o will use the system to check when to clean the tables
- hosts
 - o will use the system to check the availability of tables
- waiters
 - o will use the system to record the customer's order
- kitchen staff
 - o will use the system to check the order to prepare
 - o will notify the waiters once the food is ready
- managers (administrators)

- business insights
 - keep track of employees
 - track revenue
 - generate reports
 - keep track operational effectiveness
- restaurant owner(s)
 - will use the system to track revenue, and generate reports

This system will enable the business manage customers and customer-transactions more efficiently and conduct analysis on the menu and the customers. This means that the business can more informed decisions and adjustments to their menu, pricing of food-items, ingredient-expenses, and other business tactics.

Functional Requirements

- The system must record transactions
- The manager must be able to add, update and delete employee information
- The system must calculate be able to calculate salaries
- The system should be able to manage and display employee's shifts
- The system should notify users when stock gets low
- Utilize and database and DBMS
- The system must allow waiters to enter, modify, and track customer orders.
- The system must provide hosts with real-time updates on table status.
- The system must verify staff credentials before granting access.

Non-functional Requirements

- Attractive:
 - Aesthetically pleasing (have modern user-interface)
- Performance:
 - The system must update orders and table statuses swiftly
- Security:
 - Only authorized users can access and modify orders, table statuses, and system settings.
 - Users need to log in and have an account to access the system
- Usability:
 - The system must have an intuitive, user-friendly interface suitable for touchscreen use.

Solution proposal

A software application for a restaurant-management system

This system will benefit small restaurant owners, who do not have access to the best management systems.

The purpose of the system

- to enhance the current management system
- to calculate profits/loses and more efficiently
- to increase staff efficiency

Key stakeholders

- restaurant owners
- managers
- staff
- customers
- software-development team

The application will have the following core features:

- Sign in and login features
- Employee management
- Table management
- Order management
- Revenue calculations
- Customer management (table reservations)

Who will benefit from this system?

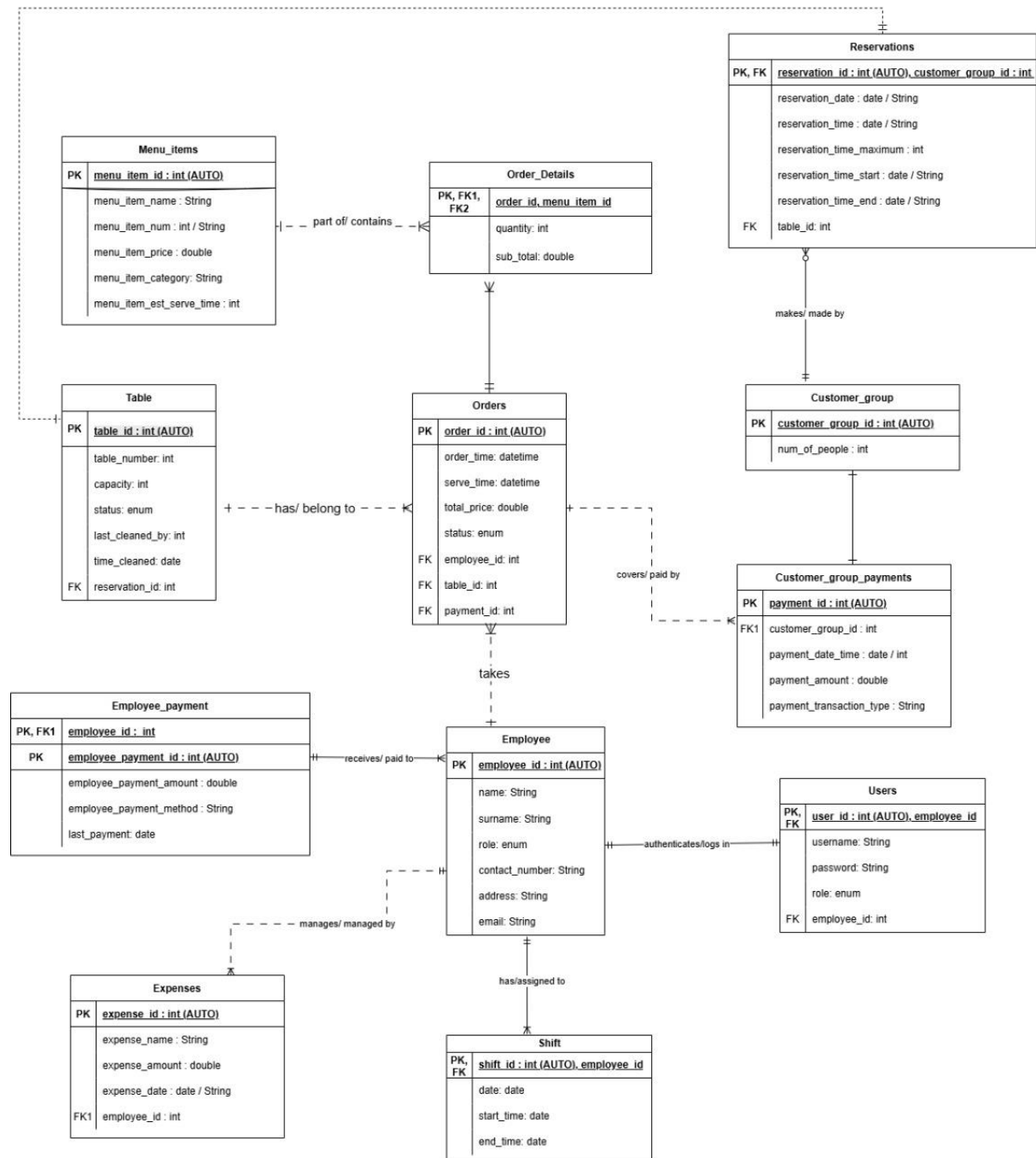
The application is for people who manage a restaurant.

- Owners
- Manager

.

Database-design

ERD



Entities (tables)

Number of entities: 12

- 1) Order
- 2) Order_details
- 3) Menu_item
- 4) Table
- 5) Reservation
- 6) Customer_group
- 7) Customer_group_payment
- 8) Employee
- 9) Employee_payment
- 10) Expenses
- 11) Shift
- 12) User

Tools and software used

Category	Tool
IDE	<ul style="list-style-type: none">- Visual Studio Code- IntelliJ
Front-end	<ul style="list-style-type: none">- HTML- CSS- JavaScript
Frameworks	<ul style="list-style-type: none">- Spring Boot- Bootstrap
Back-end	<ul style="list-style-type: none">- MySQL Workbench
VCS	<ul style="list-style-type: none">- Git- GitHub
Other communication tools	<ul style="list-style-type: none">- WhatsApp- Google Meet

Front-end

For the frontend of our application, we will be using a combination of languages helps use create a friendly environment for the hard workers who would not need to worry about how to manage their daily tasks. We will use JavaScript, CSS, and the backbone of every website, HTML5. To help create the welcoming atmosphere with a friendly atmosphere, we will be using CSS for style and use JavaScript to make it more useful with the method functionality to make it more functional and easier to use.

Backend

MySQL will essentially help with DBMS side of the website to help store some essential elements of our website. These two web development languages are highly effective at DBMS, and we therefore see it as a perfect choice for creating the entire front end of our application.

We will then be using SQL queries to send and retrieve user data from the database, as SQL is reliable and easy to use. With PHP, this will ensure that users of our product are able to create and login to their accounts.

Version control systems and collaboration

We will be using Git and GitHub, to not only store our project files for the entire team, but to provide version history in case of code breakage, extra backup, and a means for coders to push and pull their changes to work on the same updated version.

We will make use of various platforms to meet, plan and collaborate further than just code. We are currently making use of Google Meets to schedule meetings and scrum meets to see if we are still on the same page and share updates on the projects altogether.

As much as video/voice conferencing is the best choice for meetings, for those times when we need to share ideas and updates when it is out of the meeting schedule, we huddle together on our own WhatsApp Group to communicate and delegate tasks, also to check in and have meetings to discuss the sprints/work that has been done and where to do better, what to change, what still needs to be completed.

Challenges

Scope

Challenge:

Defining the scope and requirements of the application was a significant challenge. Mental health is a broad and sensitive topic and ensuring that our application addressed the right issues without complicating the user experience required careful planning and consultation with group members.

Solution:

We conducted thorough research to narrow down our requirements and regular meetings, and feedback sessions helped us stay aligned with our goals.

Database Integration and CRUD Operations

Challenge:

Integrating the application with a database to perform all CRUD (Create, Read, Update, Delete) operations was technically challenging. Ensuring data consistency, security, and efficient data retrieval were key concerns.

Solution:

We used best practices for the database management system and implemented secure coding practices. Regular testing and debugging were essential to ensure the database operations were reliable and efficient.

User Interface and User Experience (UI/UX) Design

Challenge:

Designing an interface that was user-friendly for individuals with varying levels of technical understanding was challenging. The application needed to be accessible and easy to navigate while providing the necessary resources.

Solution:

We adopted a user-centred design approach, creating rough wireframes that guided us to the final design. Feedback from group members was incorporated collectively to improve the UI/UX.

Technical Challenges and Debugging

Challenge:

Encountering technical issues, such as bugs and performance issues, was expected. Debugging and optimising the application to ensure smooth functionality was time-consuming.

Solution:

We adopted a systematic approach to debugging, using tools and techniques to identify and resolve issues. Collaborative problem-solving sessions were important in overcoming technical issues.

Achievements

ERD Success

Even though the development of the ERD was a difficult milestone and was one of the most important steps to the design of our database, we were able to overcome the challenges. As said, our team had initially faced difficulties as creating relationships with ERD relationship symbols was something we weren't fully skilled in. This became a learning process for all members, as we sat in meetings constantly wanting to create the most well-defined ERD that would lay the foundation for the database.

By visualizing the interaction that would take place between users, doctors, and the admin control panel, we gained a deeper understanding of how features in our application would work. Concentrating on how the database would act with the application and allow the users and administrators to save their credentials was crucial, as was storing, retrieving, and being able to manipulate data smoothly. Once we were satisfied with the design of our ERD, we proceeded to the development of our GUI as we had a stable structure to support our database.

Database

The database gave us the most trouble as team members were unable to connect to databases on their side, and setting up became stressful with limited time on our hands in Term 3. To tackle this problem, we sat for longer periods on Google meetings sorting out each member's issues, and there were a variety of problems. During these sessions, we shared troubleshooting tips and explored different connection methods. By collaborating and sharing our knowledge, we gradually identified the root causes of the connectivity problems and implemented a solution.

Setting up the database was a great challenge for us as a team, but we were able to successfully achieve a functioning application and database.

The success of implementing the CRUD method.

When our application was finally completed, we successfully implemented the CRUD (Create, Read, Update, Delete) method for only a few functions. We shared our progress and worked with our supervisor. However, during our explanation, we quickly realized that we had misunderstood the expectations of implementing the CRUD method.

Doing this, we split the entities among the members, and each member had to test the CRUD method. A few problems came back where the CRUD methods was not implemented correctly on a few tables. We scheduled a meeting to correct these

problems and have successfully done so. This experience reinforced the importance of adaptability and teamwork, as we worked under pressure to meet the project requirements. In the end, the successful implementation of the CRUD method across all tables not only enhanced the overall performance of Wellness Wave but also provided us with invaluable insights into database management and user interface design.

Potential improvements

We could implement the following ideas to enhance our current application.

1. Improve the UI and GUI of our application.

There is not enough brand identity, as the pages are sparse and lack content and logos. The UI can be improved by adding more brand identity with our company's content and logos and by correcting the colors and feeling of unity on each of the pages. Modern UI design improvements to the UI would also improve our user experience greatly.

2. Security

We could improve the apps security by introducing 2FA (2 factor authentication) for both patient users and administrators to protect their accounts. We can also improve how we handle users' sensitive personal information and their mental health status information so that we can abide by the POPI (protection of personal information act) and ensure users trust us with their information.
