



**UNIVERSITI TEKNOLOGI MARA (UiTM) SHAH ALAM
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS**

**ICT502 – DATABASE ENGINEERING
OCTOBER 2024 – FEBRUARY 2025**

Neighborly Home System (Neighborly)

FINAL REPORT

PREPARED BY:

NO.	NAME	STUDENT ID
1.	MUHAMAD DANIAL AZIZI BIN MOHD SALEHUDDIN	2024793123
2.	DEAN ARDLEY BIN REZALI	2024793015
3.	SITI SOLEHA BINTI SALEHUDDIN	2024794451
4.	NOR ATIQAH BINTI HASAN BASRI	2024988925

**PREPARED FOR:
DR. NOR DIANA BINTI AHMAD**

Table of Contents

1.0	Company Background	1
2.0	Problem Statement.....	2
3.0	Objectives	2
4.0	Flow Chart	3
5.0	SQL Queries.....	4
6.0	System Development Sample Screen	11
7.0	Conclusion	29
8.0	Appendix.....	30
7.1	Entity Relationship Diagram	30
7.2	3NF Relationship Diagram.....	31
7.3	Data Dictionary	32
7.4	Data Definition Language (DDL).....	36
7.5	Data Manipulation Language (DML).....	44
9.0	References.....	48

1.0 Company Background



Figure 1: Laman Seri Residential, Sek 13 Shah Alam

Laman Seri is a mixed residential and commercial development located in Seksyen 13, Shah Alam, Selangor. Developed by Naza TTDI Sdn Bhd, a prominent property development company in Malaysia, the project reflects the company's dedication to creating well-planned, sustainable communities. The development offers a strategic location with excellent connectivity through major highways such as the Federal Highway and the New Klang Valley Expressway (NKVE), making it easily accessible from various parts of the Klang Valley.

The development of Laman Seri commenced in the early 2000s and was completed over a span of approximately five years. The residential component features a gated and guarded community, prioritizing safety and quality living for its residents. The careful planning of this area ensures an environment conducive to fostering close community relationships while maintaining privacy and security for all homeowners.

In addition to its residential offerings, Laman Seri includes the Laman Seri Business Park, which provides a range of essential services such as dining, retail, and office spaces. This integration of commercial and residential areas makes Laman Seri a self-sufficient community, meeting the diverse needs of its residents and visitors. The combination of accessibility, convenience, and security has positioned Laman Seri as one of the preferred neighborhoods in Shah Alam. Given its structure and facilities, Laman Seri is a fitting choice for implementing a neighborhood management system. The system is intended to enhance communication between residents, optimize resource management, and foster greater community involvement. This initiative aligns with the core values of Laman Seri, supporting its vision of a connected and harmonious living environment.

2.0 Problem Statement

There are 3 problem statement that we discuss for this project which is :

1. Lack of Efficient Communication Between Residents and the Joint Management Body (JMB)

The current communication channels between residents and the JMB are fragmented and inefficient, often resulting in delays in addressing inquiries, complaints, and updates. This hampers the overall satisfaction and coordination within the community.

2. Outdated and Inefficient Visitor Management System

The existing visitor management process relies heavily on manual record-keeping or outdated methods, leading to delays, errors, and potential security vulnerabilities. This not only affects the guards' efficiency but also raises concerns about the safety of the neighborhood.

3. Limited Accessibility to Neighborhood Services and Information

Residents face difficulties in accessing essential services such as bill payments, announcements, and event updates. The lack of a centralized platform creates inconvenience and discourages active community participation.

3.0 Objectives

There are 3 objectives that focus on to develop our system :-

1. To identify the functional and non-functional requirements for the Neighborly Home System to address the challenges faced by residents and the JMB.
2. To design and develop the Neighborly Home System as a user-friendly platform integrating communication, visitor management, and neighborhood services.
3. To evaluate the Neighborly Home System's performance and usability through testing and user feedback.

4.0 Flow Chart

4.1 Admin Flow Chart

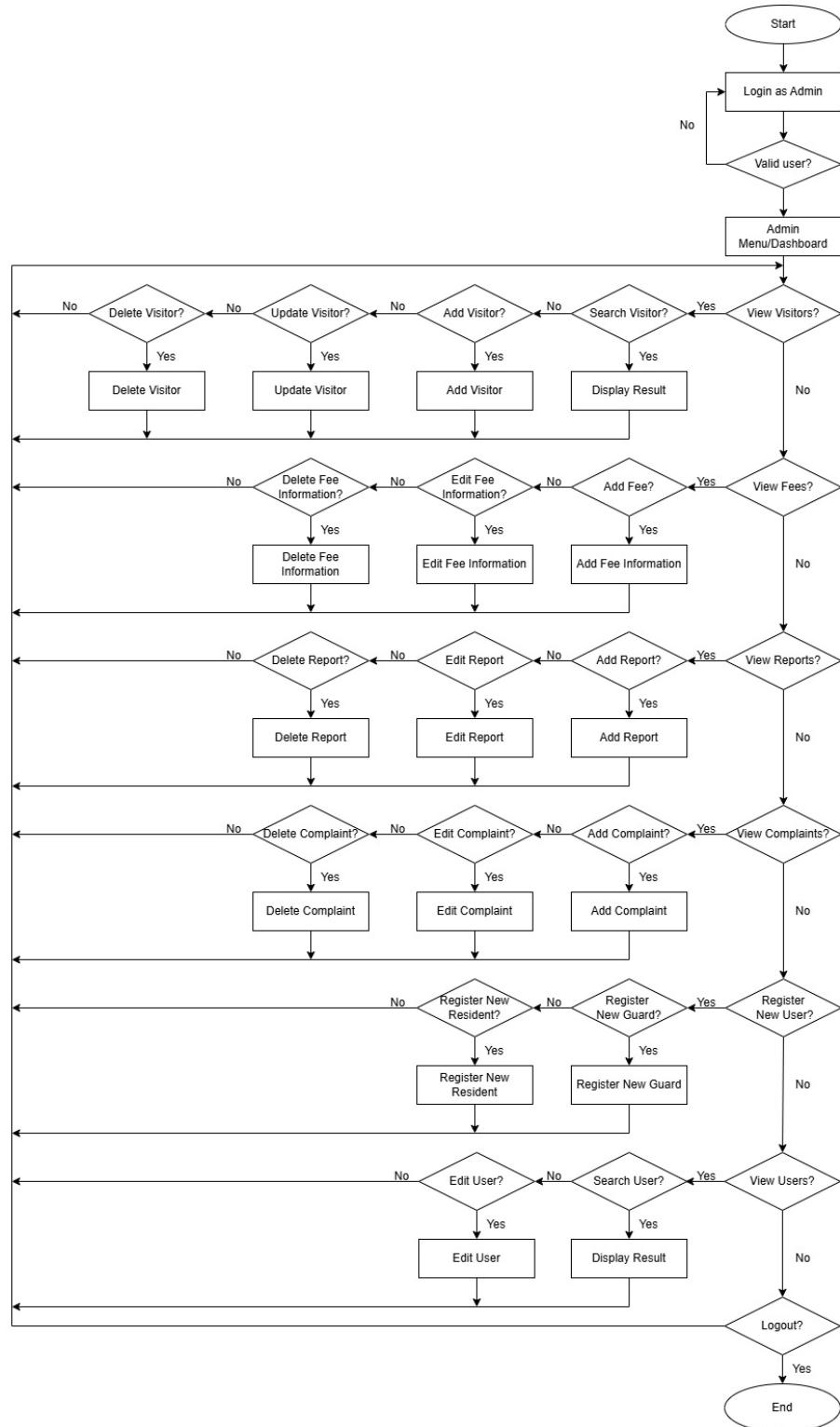


Figure 2: Flow Chart System for Admin

4.2 Guard Flow Chart

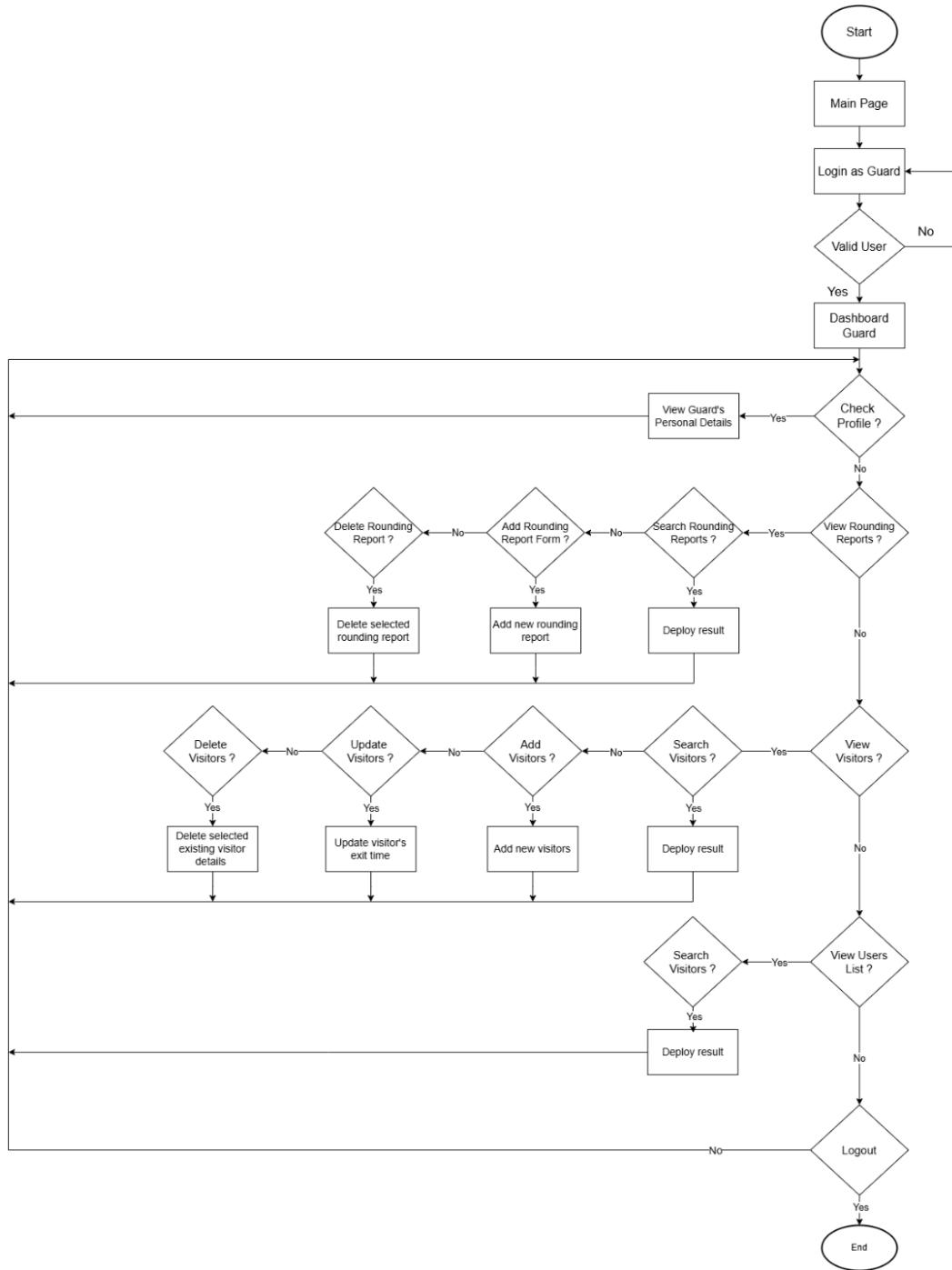


Figure 3: Flow Chart System for Guard

4.2 Resident Flow Chart

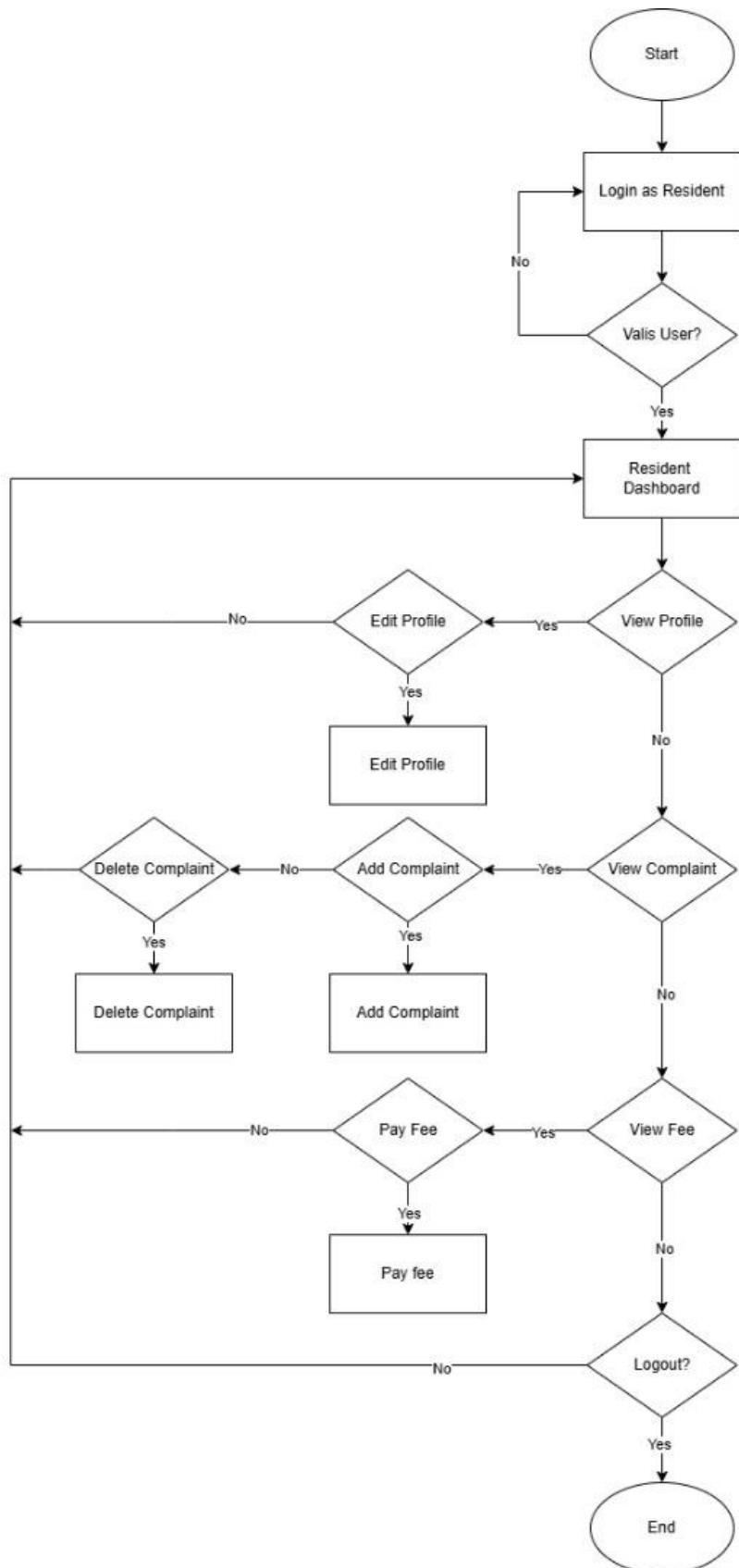


Figure 4: Flow Chart System for Resident

5.0 SQL Queries

1. Find All visitors who entered on the specific date.

```
SELECT visitor_name, no_plate, entryTime, exitTime  
FROM visitor  
WHERE dateOfVisit = '14-09-2024';
```

Result:

The screenshot shows a database interface with a 'Worksheet' tab active. In the code editor, the following SQL query is written:

```
SELECT visitor_name, no_plate, entryTime, exitTime  
FROM visitor  
WHERE dateOfVisit = '14-09-2024';
```

Below the code editor, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is selected, showing the output of the query. The output indicates 'All Rows Fetched: 1 in 0.009 seconds'. The results are displayed in a table with columns: VISITOR_NAME, NO_PLATE, ENTRYTIME, and EXITTIME. The single row returned is:

VISITOR_NAME	NO_PLATE	ENTRYTIME	EXITTIME
Sofia Ahmad	GHJ6574	14/09/2024 17:30:00.000000000	14/09/2024 18:30:00.000000000

2. Get a list of all complaints along with their status description.

```
SELECT c.complaintID, c.complaint_description, s.status_description  
FROM complaint c  
JOIN status s ON c.statusID = s.statusID;
```

Result:

The screenshot shows a database interface with a 'Worksheet' tab active. In the code editor, the following SQL query is written:

```
SELECT c.complaintID, c.complaint_description, s.status_description  
FROM complaint c  
JOIN status s ON c.statusID = s.statusID;
```

Below the code editor, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is selected, showing the output of the query. The output indicates 'All Rows Fetched: 19 in 0.012 seconds'. The results are displayed in a table with columns: COMPLAINTID, COMPLAINT_DESCRIPTION, and STATUS_DESCRIPTION. The data rows are:

COMPLAINTID	COMPLAINT_DESCRIPTION	STATUS_DESCRIPTION
1	70011 Water leakage in My House	Pending
2	70012 Unauthorized parking at visitor area.	Processing
3	70013 Broken streetlights.	Completed
4	70014 Security guard absent during night shift.	Completed
5	70015 Noisy neighbors during late hours.	Processing

3. Retrieve the details of residents who have registered complaints.

```
SELECT r.userID, r.unit, c.complaint_description  
FROM resident r  
JOIN complaint c ON r.userID = c.userID;
```

Result:

The screenshot shows the Oracle SQL Developer interface. The top pane is titled "Worksheet" and contains the SQL query. The bottom pane is titled "Query Result" and displays the fetched data in a grid format.

	USERID	UNIT	COMPLAINT_DESCRIPTION
1	10002	1	Unauthorized parking at visitor area.
2	10002	1	Dogs barking excessively.
3	10003	2	Car damaged in parking lot.
4	10003	2	Security guard absent during night shift.
5	10004	3	Security guard absent during night shift.
6	10005	4	Noisy neighbors during late hours.
7	10005	4	Noisy neighbors during late hours.
8	10006	5	Noise disturbance from construction.
9	10006	5	Water leakage in My House

4. Count the total number of complaints grouped by complaint type.

```
SELECT ct.comp_type_name, COUNT(*) AS total_complaints  
FROM complaint c  
JOIN complaint_type ct ON c.complaint_type_id = ct.complaint_type_id  
GROUP BY ct.comp_type_name;
```

Result:

The screenshot shows the Oracle SQL Developer interface. The top pane is titled "Worksheet" and contains the SQL query. The bottom pane is titled "Query Result" and displays the fetched data in a grid format.

	COMP_TYPE_NAME	TOTAL_COMPLAINTS
1	Parking	2
2	Others	2
3	Property	7
4	Environment	1
5	Waste Management	1
6	Noise	6

5. List all fees with their respective categories.

```
SELECT f.feeID, fc.fee_category_name, f.fee_amount  
FROM fee f  
JOIN fee_category fc ON f.fee_category_ID = fc.fee_category_ID;
```

Result:

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the SQL query:SELECT f.feeID, fc.fee_category_name, f.fee_amount
FROM fee f
JOIN fee_category fc ON f.fee_category_ID = fc.fee_category_ID;The 'Query Result' tab displays the output in a grid format:

FEEID	FEEL_CATEGORY_NAME	FEEL_AMOUNT
1	30001 Quit Rent	364
2	30002 Quit Rent	425.66
3	30003 Quit Rent	396.11
4	30004 Insurance	60.13
5	30005 Maintenance	371.02
6	30006 Quit Rent	469.48

6. Find all guards working in a specific post location.

```
SELECT u.username, u.name, g.post_location  
FROM guard g  
JOIN users u ON g.userID = u.userID  
WHERE post_location LIKE '%Gym%';
```

Result:

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the SQL query:SELECT u.username, u.name, g.post_location
FROM guard g
JOIN users u ON g.userID = u.userID
WHERE post_location Like '%Gym%';The 'Query Result' tab displays the output in a grid format:

USERNAME	name	POST_LOCATION
1 hakim	Hakim bin Idris	Main Gate, Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j, Swimming Pool, Gym

7. Retrieve residents' details along with their unit and phone numbers.

```
SELECT r.unit, u."name", u.phoneNum  
FROM resident r  
JOIN users u ON r.userID = u.userID;
```

Result:

The screenshot shows the MySQL Workbench interface. The 'Query Builder' tab is active, displaying the SQL query:

```
SELECT r.unit, u."name", u.phoneNum  
FROM resident r  
JOIN users u ON r.userID = u.userID;
```

The 'Query Result' tab shows the output of the query:

UNIT	name	PHONENUM
1 1	Ahmad bin Ismail	0134567890
2 2	Siti binti Amin	(null)
3 3	Zulkifli bin Mohd Noor	0116789012
4 4	Rahman bin Salleh	0125678901
5 5	Fatimah binti Idris	0123456789
6 6	Rosli bin Hamid	0167890123

8. Get the details of guards working a specific shift.

```
SELECT u."name" AS GuardName, g.shift, g.post_location  
FROM users u  
JOIN guard g ON u.userID = g.userID  
WHERE g.shift = 'Night';
```

Result:

The screenshot shows the MySQL Workbench interface. The 'Query Builder' tab is active, displaying the SQL query:

```
SELECT u."name" AS GuardName, g.shift, g.post_location  
FROM users u  
JOIN guard g ON u.userID = g.userID  
WHERE g.shift = 'Night';
```

The 'Query Result' tab shows the output of the query:

GUARDNAME	SHIFT	POST_LOCATION
Hilmi bin Zain	Night	Main Gate, Jalan Memanah 13/55e, Jalan Memanah 13/55f, Jalan Memanah 13/55g, DLaman Seri PlayGround, Dewan Astana

9. Get the number of visitors grouped by purpose of visit.

```
SELECT purposeOfVisit, COUNT(*) AS total_visitors  
FROM visitor  
GROUP BY purposeOfVisit;
```

Result:

The screenshot shows the MySQL Workbench interface. The 'Worksheet' tab is active, displaying the SQL query:SELECT purposeOfVisit, COUNT(*) AS total_visitors
FROM visitor
GROUP BY purposeOfVisit;The 'Query Result' tab shows the output:

PURPOSEOFVISIT	TOTAL_VISITORS
1 Home repair service.	1
2 Delivering a parcel.	1
3 Repair work inspection.	1
4 Delivery of appliances.	1
5 Visiting family.	1
6 Family gathering.	1
7 Dropping off documents.	1

10. List all fees that are pending payment

```
SELECT f.feeID, u."name" AS ResidentName, fc.fee_category_name, f.fee_amount  
FROM fee f  
JOIN users u ON f.userID = u.userID  
JOIN fee_category fc ON f.fee_category_ID = fc.fee_category_ID  
WHERE f.fee_category_id = 80001;
```

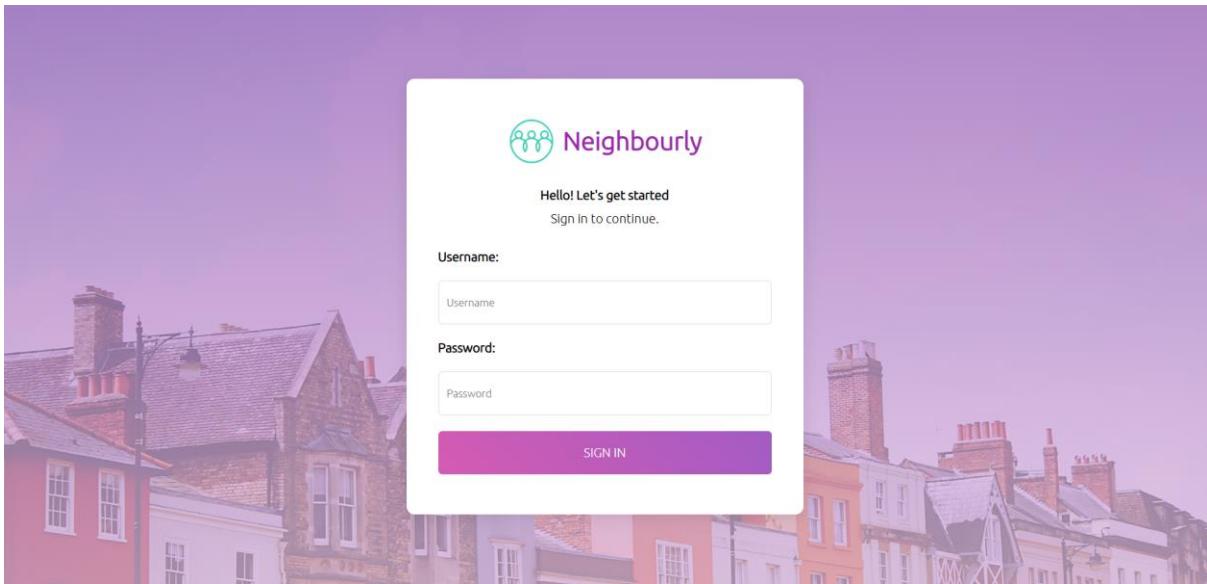
Result:

The screenshot shows the MySQL Workbench interface. The 'Worksheet' tab is active, displaying the SQL query:SELECT f.feeID, u."name" AS ResidentName, fc.fee_category_name, f.fee_amount
FROM fee f
JOIN users u ON f.userID = u.userID
JOIN fee_category fc ON f.fee_category_ID = fc.fee_category_ID
WHERE f.fee_category_id = 80001;The 'Query Result' tab shows the output:

FEEID	RESIDENTNAME	FEEL_CATEGORY_NAME	FEEL_AMOUNT
1	30005 Noraini binti Zainal	Maintenance	371.02
2	30011 Siti binti Amin	Maintenance	371.02
3	30014 Zulkifli bin Mohd Noor	Maintenance	100

6.0 System Development Sample Screen

1. User Login (Admin, Resident And Guard)



2. Dashboard (Admin, Resident and Guard)

Admin Dashboard

A screenshot of the Admin Dashboard. On the left, a sidebar menu lists "Hi, admin", "Dashboard", "Visitor", "Fee", "Report", "Complaints", "Registration", "List User", and "Logout". The main dashboard area has a title "Dashboard" and a "Overview" link. It displays four summary cards: "Visitors" (18), "Complaints" (20), "Reports" (8), and "Fees" (4). Below these are two tables: "Pending Complaints" and "Completed Fees".

Complaints ID	Description	Date	Location	Status	Attachment
70024	Unlit streetlights in the park area.	2025-01-25	City Park	Pending	No Attachment
70026	Window shattered during storm, dangerous shards.	2026-05-01	Apartment 3A	Pending	View
70029	Potholes on main street causing damage to vehicles.	2026-08-21	Main Street	Pending	View
70033	Public park benches are damaged and unsafe.	2026-03-20	Central Park	Pending	View
70036	Elevator repeatedly breaking down in the building.	2026-06-03	Building A, 4th Floor	Pending	View
70039	Overflowing trash bins near the mall entrance.	2026-09-30	Mall Entrance	Pending	View

Fee ID	Fee Category	Date	Amount	Status	Attachment
30014	Insurance	2025-01-22	RM1000	Completed	View

Resident Dashboard

Hi, soleha

Dashboard

Profile

Complaint

Fee

Log Out

Resident Dashboard

WELCOME BACK, SOLEHA!

We're excited to have you back. Your personalized dashboard is ready to help you manage your residence effortlessly.

Explore your options and enjoy your stay!

START EXPLORING

Month	Complaints
January	5.0
February	0.0
March	0.0
April	0.0
May	0.0
June	0.0
July	0.0
August	0.0
September	0.0
October	0.0
November	0.0
December	0.0

Status	Percentage
Completed	~33%
Pending	~55%
Processing	~12%

Your Process Fees

Fee Type	Amount	Status
Insurance	RM 250.0	Pending
Quit Rent	RM 125.0	Pending
Maintenance	RM 222.0	Pending
Insurance	RM 111.0	Pending

Guard Dashboard

Welcome to the Dashboard, hakim!

Your User ID: 10024

Rounding Reports
Submitted Today: 1
Total Reports Submitted 11

Today's Visitors Count
Visitor In: 2
Visitor Out: 1

ID	Visitor's Name	IC/Passport	Plate Number	Entry Time	Exit Time	Date	Purpose	Phone No	Action
20063	Danial	2024	BRE7789	13:09	N/A	2025-01-28	Just Visit My Cousins	0182450835	Edit Delete

3. Read (Admin, Resident And Guard)

Admin

READ data from List of Visitors

List of Visitors

Register ID	Visitor Name	IC Number	Plate Number	Entry Time	Exit Time	Date	Purpose	Phone Number	Action
20002	Kim Gaeul	239402-99-1295	ABC1234	2025-01-01 15:30:00	2025-01-01 16:40:00	2024-12-02 00:00:00.0	Visit Dean Ardley	01243850092	Edit Delete
20022	Liz	040522-99-2965	RD110	2025-01-01 22:30:00	2025-01-01 23:30:00	2025-01-04 00:00:00.0	Visit Dean	01243258853	Edit Delete
20055	Bang Chan	040522-99-2965	VKC2245	2025-01-01 09:05:00	2025-01-01 11:05:00	2025-01-22 00:00:00.0	Visit Dean	01137370938	Edit Delete
20062	Jane Smith	950521-12-3456	XYZ5678	2025-02-10 09:15:00	2025-02-10 11:00:00	2025-02-10 00:00:00.0	Delivery of documents.	0176543210	Edit Delete
20061	John Doe	890123-45-6789	ABC1234	2025-01-15 08:30:00	2025-01-15 10:30:00	2025-01-15 00:00:00.0	Meeting with tenant.	0123456789	Edit Delete
20063	Michael Tan	850409-11-2233	DEF4567	2025-03-05 14:00:00	2025-03-05 15:30:00	2025-03-05 00:00:00.0	Repair and maintenance work.	0198765432	Edit Delete
20064	Emily Wong	990818-08-9876	LMN3456	2025-04-12 12:45:00	2025-04-12 14:00:00	2025-04-12 00:00:00.0	Inspection of facilities.	0112233445	Edit Delete
20065	Sarah Lee	920304-	null	2025-05-20	2025-05-20	2025-05-20	Discussing rental	0101122334	Edit

READ data from List of Fees

List of Fees					
Fee ID	User ID	Fee Date	Fee Amount	Status	Action
30001	10002	2024-12-01	RM364	Processing	Edit Delete
30002	10003	2024-12-02	RM425	Completed	Edit Delete
30003	10005	2024-12-02	RM396	Processing	Edit Delete
30004	10007	2024-12-11	RM60	Completed	Edit Delete
30005	10008	2024-12-12	RM371	Processing	Edit Delete
30006	10010	2024-12-15	RM469	Pending	Edit Delete
30007	10004	2024-12-15	RM363	Processing	Edit Delete
30008	10006	2024-12-17	RM425	Completed	Edit Delete
30009	10009	2024-12-17	RM396	Processing	Edit Delete
30011	10003	2023-12-18	RM371	Processing	Edit Delete
30013	10009	2024-12-19	RM469	Pending	Edit Delete
30014	10004	2025-01-27	RM100	Pending	Edit Delete

READ data from List of Reports

List of Reports							
Report ID	User ID	Date	Location	Remarks	Attachment	Action	
40001	10022	2024-12-01	LakeSide Parking Lot, Laman Seri Seksyen 13	Inspected parking area. No issues found.	View	Edit Delete	
40002	10023	2024-12-02	Jalan Memanah 13/55e, Jalan Memanah 13/55f, Jalan Memanah 13/55g Laman Seri Seksyen 13	All clear.	View	Edit Delete	
40003	10024	2024-12-03	Swimming Pool, Laman Seri Seksyen 13	All clear.	View	Edit Delete	
40004	10022	2024-12-04	Jalan Memanah 13/55a, Jalan Memanah 13/55b, Jalan Memanah 13/55c, Jalan Memanah 13/55d Laman Seri Seksyen 13	Monitored lobby area. Suspicious activity noted.	View	Edit Delete	
40033	10001	2025-01-29	cuba	ghfg	View	Edit Delete	
40005	10023	2024-12-05	Dewan Astana, Laman Seri Seksyen 13	Checked water safety equipment. All in order.	View	Edit Delete	
40006	10024	2024-12-06	Gym Fitness Center, Laman Seri Seksyen 13	Inspected gym equipment. One machine requires maintenance.	View	Edit Delete	
40007	10022	2024-12-07	Jalan Memanah 13/55a, Jalan Memanah 13/55b, Jalan Memanah 13/55c, Jalan Memanah 13/55d Laman Seri Seksyen 13	Reviewed CCTV footage. No anomalies detected.	View	Edit Delete	
40008	10023	2024-12-08	PlayGround, Laman Seri Seksyen 13	Repaired broken light fixtures.	View	Edit Delete	
40009	10024	2024-12-09	Jalan Memanah 13/55h, Jalan Memanah 13/55l, Jalan Memanah 13/55j Laman Seri Seksyen 13	All clear.	View	Edit Delete	
40010	10022	2024-12-10	LakeSide Parking, Laman Seri Seksyen 13	All clear.	View	Edit	

READ data from List of Complaints

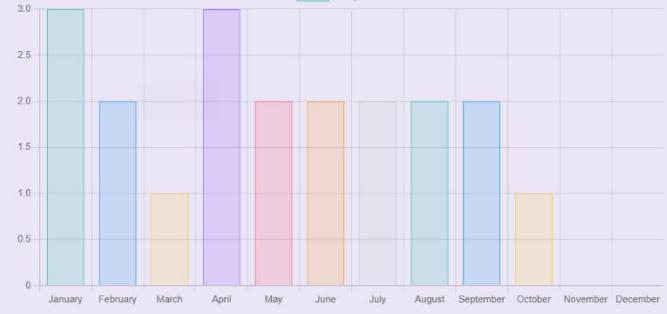
 Hi, admin

- [Dashboard](#)
- [Visitor](#)
- [Fee](#)
- [Report](#)
- [Complaints](#)
- [Registration](#)
- [User List](#)
- [Logout](#)

Complaints

Overview

Complaints per Month



Month	Complaints
January	~3.0
February	~2.0
March	~1.0
April	~3.0
May	~2.0
June	~2.0
July	~0.5
August	~2.0
September	~2.0
October	~1.0
November	~0.5
December	~0.5

List of Complaints

Complaint ID	User ID	Description	Date	Location	Attachment	Action
70023	10001	Cars blocking the driveway during rush hour.	2026-02-15	Building A, Garage	View	Edit

List of Complaints

Complaint ID	User ID	Description	Date	Location	Attachment	Action
70023	10001	Cars blocking the driveway during rush hour.	2026-02-15	Building A, Garage	View	Edit
70024	10001	Unlit streetlights in the park area.	2025-01-25	City Park	No Attachment	Edit
70025	10001	Broken bench in the community garden.	2026-04-20	Community Garden	View	Edit
70012	10001	Tree Branch	2025-04-19	Backyard	No Attachment	Edit
70013	10001	Graffiti	2025-01-31	Alleyway	View	Edit
70026	10001	Window shattered during storm, dangerous shards.	2026-05-01	Apartment 3A	View	Edit
70027	10001	Neighbor's dog barking incessantly.	2026-06-10	Block 8, Street 2	View	Edit
70028	10001	Flooding in basement due to heavy rain.	2026-07-15	Building H, Basement	View	Edit

READ data from Recent Users



Hi, Danial

- [Dashboard](#)
- [Visitor](#)
- [Fee](#)
- [Report](#)
- [Complaints](#)
- [Registration](#)
- [User List](#)
- [Logout](#)



Total Count

Category	Count
Residents	19
Guards	3
Admins	1

Recent Users

User ID	Username	Name	Email	Role	Salary	Shift	Post Location	House Unit	Vehicle Number	Action
10002	ahmed	Ahmad bin Ismail	0134567890	Resident	-	-	-	1	WXY1234	<button>Edit</button>
10003	siti	Siti binti Amin	null	Resident	-	-	-	2	VZX5678	<button>Edit</button>
10004	zulkifli	Zulkifli bin Mohd Noor	0116789012	Resident	-	-	-	3	XYZ3456	<button>Edit</button>
10005	rahman	Rahman bin Salleh	0125678901	Resident	-	-	-	4	EFG1234	<button>Edit</button>

User ID	Username	Name	Email	Role	Salary	Shift	Post Location	House Unit	Vehicle Number	Action
10010	laila	Laila binti Musa	0135678901	Resident	-	-	-	9	JKL9012	<button>Edit</button>
10011	hafiz	Hafiz bin Ramli	0146789012	Resident	-	-	-	10	MNO1234	<button>Edit</button>
10012	azizah	Azizah binti Yusof	0157890123	Resident	-	-	-	11	PQR5678	<button>Edit</button>
10013	shafiq	Shafiq bin Hassan	0190123456	Resident	-	-	-	12	STU9012	<button>Edit</button>
10014	ain	Ain binti Razak	0172345678	Resident	-	-	-	13	VWX3456	<button>Edit</button>
10015	ismail	Ismail bin Othman	0133456789	Resident	-	-	-	18	YZA5678	<button>Edit</button>
10016	zainab	Zainab binti Kamal	0164567890	Resident	-	-	-	19	BCD9012	<button>Edit</button>
10017	nurul	Nurul binti Saad	0146789012	Resident	-	-	-	20	HJ5678	<button>Edit</button>
10018	faizal	Faizal bin Daud	0187890123	Resident	-	-	-	21	KLM9012	<button>Edit</button>
10019	sofia	Sofia binti Ariffin	0138901234	Resident	-	-	-	22	NOP3456	<button>Edit</button>
10020	adnan	Adnan bin Rahman	0179012345	Resident	-	-	-	26	QRS5678	<button>Edit</button>
10021	alia	Alia binti Mokhtar	0160123456	Resident	-	-	-	27	TUV9012	<button>Edit</button>
10024	hakim	Hakim bin Idris	0123456789	Guard	2000.0	Day	Main Gate, Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j, Swimming Pool, Gym	-	XYZ1234	<button>Edit</button>
10023	hilmi	Hilmi bin Zain	0136789012	Guard	1800.0	Night	Main Gate, Jalan Memanah 13/55e, Jalan Memanah 13/55f, Jalan Memanah 13/55g, Dlaman Seri PlayGround, Dewan Astana	-	JKL7890	<button>Edit</button>
10022	nadia	Nadia binti Osman	0167890123	Guard	1800.0	Day	Main Gate, Jalan Memanah 13/55a, Jalan Memanah 13/55b, Jalan Memanah 13/55c, Jalan Memanah 13/55d, Laman Seri LakeSide	-	DEF4567	<button>Edit</button>
10001	admin	Daniel	0192334567	Admin	2500.0	-	-	-	JKQ6678	<button>Edit</button>

16

Resident

Read of Profile

The screenshot shows the Resident profile page. On the left sidebar, there is a user icon and the text "Hi, siti". Below it are links: Dashboard, Profile (which is highlighted in blue), Complaint, Fee, and Log Out. The main content area has a header "Profile" and a section titled "User Profile". It contains the following data:

Username:	siti
Full Name:	Siti binti Amin
IC Number:	920202-02-2020
Address:	2
Contact:	null
Email:	siti@gmail.com
Vehicle Number Plate:	VZX5678

At the bottom of the profile section is a purple "Update Profile" button.

Read of Complaint

The screenshot shows the Complaints List page. On the left sidebar, there is a user icon and the text "Hi, siti". Below it are links: Complaint, Fee (which is highlighted in blue), and Log Out. The main content area has a header "Complaints List" and a table showing two complaints:

Complaint ID	Complaint Type	Complaint Description	Complaint Date	Complaint Location	Attachment	Status	Action
70014	Others	Security guard absent during night shift.	2024-12-04	Main Gate	View	Completed	Delete
70023	Noise	Car damaged in parking lot.	2024-12-13	Parking Lot LakeSide	View	Processing	Delete

Read of Fee

The screenshot shows the Fee page. On the left sidebar, there is a user icon and the text "Hi, siti". Below it are links: Dashboard, Profile, Complaint, Fee (which is highlighted in blue), and Log Out. The main content area has a header "Fee" and a section titled "Your Fee Information". It contains a table showing two fee entries:

Fee Type	Amount	Date	Your Payment	Remark	Payment Receipt	Status	Action
Maintenance	RM 371.02	2023-12-18	RM 400	I already paid	View	Processing	Pay
Quit Rent	RM 425.66	2024-12-02	RM 600	Done Payment	View	Completed	Pay

Guard

Read of Profile

The screenshot shows the Guard profile page. On the left sidebar, there is a user icon and the text "Hi, drake". Below it are links: Dashboard, Profile (which is highlighted in blue), Rounding Report, Visitor Form, Users List, and Logout. The main content area has a header "Profile" and a section titled "User Profile". It contains the following data:

Username:	drake
Full Name:	bobo
IC Number:	0312445566
Contact:	0182342345
Email:	drake@gmail.com
Shift:	Morning
Post Location:	Playground
Salary:	null

Read of Rounding Report

The screenshot shows a dashboard with a sidebar on the left containing user profile information (Hi, hakim), navigation links (Dashboard, Profile, Rounding Report, Visitor Form, Users List, Logout), and a search bar for the Rounding Report. The main content area displays a table of rounding reports with columns for ID, Date, Location, Remarks, Attachment, and Action (View, Delete). The table lists 16 entries from 2024-12-03 to 2025-01-15.

ID	Date	Location	Remarks	Attachment	Action
40003	2024-12-03	Swimming Pool, Laman Seri Seksyen 13	All clear.	View	Delete
40006	2024-12-06	Gym Fitness Center, Laman Seri Seksyen 13	Inspected gym equipment. One machine requires maintenance.	View	Delete
40009	2024-12-09	Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j Laman Seri Seksyen 13	All clear.	View	Delete
40011	2024-12-12	Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j Laman Seri Seksyen 13	All clear.	View	Delete
40014	2024-12-15	Swimming Pool, Laman Seri Seksyen 13	All clear.	View	Delete
40017	2024-12-18	Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j Laman Seri Seksyen 13	Conducted routine checks on electrical systems.	View	Delete
40019	2024-12-21	Swimming Pool, Laman Seri Seksyen 13	Monitored water tank levels and safety.	View	Delete
40022	2024-12-24	Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j Laman Seri Seksyen 13	Reported issue with access card reader.	View	Delete
40037	2025-01-28	Gym Fitness Area	All Done	View	Delete
40035	2025-01-15	Gym Fitness Area	All Done	View	Delete
40036	2025-01-15	Gym Fitness Area	All Done	View	Delete

Read of Visitors List

The screenshot shows a dashboard with a sidebar on the left containing user profile information (Hi, hakim), navigation links (Dashboard, Profile, Rounding Report, Visitor Form, Users List, Logout), and a search bar for the Visitor list. The main content area displays a table of visitor records with columns for ID, Visitor's Name, IC/Passport, Plate Number, Entry Time, Exit Time, Date, Purpose, Phone No, and Action (Edit, Delete). The table lists 20 entries from 2025-01-01 to 2025-01-28.

ID	Visitor's Name	IC/Passport	Plate Number	Entry Time	Exit Time	Date	Purpose	Phone No	Action
20001	Ali Bin Karim	871210-05-4321	DEF1234	09:00	11:00	2025-01-06	Home repair service.	0134456678	Edit Delete
20002	Rina Kartika	920305-10-6789	LMN9876	10:15	12:30	2025-01-07	Delivering food.	014778899	Edit Delete
20003	Joshua Tan	960712-02-2345	PQR4567	13:00	14:00	2025-01-08	Meeting with a tenant.	0191122334	Edit Delete
20004	Emily Wong	880923-07-1234	XYZ7890	08:30	10:15	2025-01-09	Bringing groceries.	0123344556	Edit Delete
20005	Fadilah Zain	940406-08-5678	ASD3456	16:00	18:00	2025-01-10	Consultation on legal matters.	0175566778	Edit Delete
20006	Zul Afiq	910123-14-7890	QWE9087	14:15	15:45	2025-01-11	Delivery of appliances.	0114455667	Edit Delete
20007	Nora Abdullah	950317-06-1234	TYU5678	11:30	12:15	2025-01-12	Discussing property renovation.	0181122334	Edit Delete
20008	Hafiz Rahman	880511-09-3456	ZXC2345	09:45	10:30	2025-01-13	Inspecting house repairs.	0157788990	Edit Delete
20009	Sofia Ahmad	870802-11-4567	GHJ6574	17:30	18:30	2025-01-14	Family gathering.	0193344556	Edit Delete
20010	Daniel Lee	891205-13-6789	VBN1234	20:00	21:30	2025-01-15	Birthday celebration.	0135566778	Edit Delete
20011	Ahmad Zamri	900101-01-1234	WXY1234	10:00	12:00	2025-01-01	Delivering a parcel.	0123456789	Edit Delete
20012	Norain Binti Hassan	850205-08-5678	JFD2211	08:30	09:15	2025-01-02	Visiting family.	0198765432	Edit Delete
20013	Jonathan Lee	750715-12-3456	XYZ3344	14:00	15:00	2025-01-03	Meeting about a house sale.	0172233445	Edit Delete
20014	Sti Mariam	980430-03-6789	ABC4567	11:00	13:30	2025-01-04	Dropping off documents.	0122233445	Edit Delete
20015	Michael Tan	800312-14-5678	KLM9087	15:30	16:15	2025-01-05	Repair work inspection.	0165678901	Edit Delete
20061	Danial	2024	BRE7789	12:06	00:13	2025-01-14	Just Visit My Cousins	0182450835	Edit Delete
20062	Danial	2024	BRE7789	13:01	01:02	2025-01-28	Just Visit My Cousins	0182450835	Edit Delete

Read of Users List

The screenshot shows a user interface for managing users. On the left, there's a sidebar with navigation links: Dashboard, Profile, Rounding Report, Visitor Form, Users List (which is selected and highlighted in pink), and Logout. The main content area has a search bar at the top with placeholder text "Enter name, ic/passport or phone number". Below the search bar is a table titled "Recent Users" with the following columns: UserID, Name, IC/Passport, Phone Number, Plate Number, Email, and House Unit. The table contains 18 rows of data, each representing a user with their details.

UserID	Name	IC/Passport	Phone Number	Plate Number	Email	House Unit
10002	Ahmed bin Ismail	901010-10-1010	0134567890	WXY1234	ahmad@gmail.com	1
10003	Siti binti Amin	920202-02-2020	null	VZX5678	siti@gmail.com	2
10004	Zulkifli bin Mohd Noor	880808-08-8080	0116789012	XYZ3456	zul@gmail.com	3
10005	Rahman bin Salleh	950505-05-5050	0125678901	EFG1234	rahman@gmail.com	4
10006	Fatimah binti Idris	950505-05-5050	0123456789	WKL9876	fatimah@gmail.com	5
10007	Rosli bin Hamid	870707-07-7070	0167890123	ABC5432	rosli@gmail.com	6
10008	Noraini binti Zainal	890909-09-9090	0189012345	DEF1234	noraini@gmail.com	7
10009	Azlan bin Rahim	911010-01-1010	0124567890	GHI5678	azlan@gmail.com	8
10010	Laila binti Musa	930303-03-3030	0135678901	JKL9012	laila@gmail.com	9
10011	Hafiz bin Ramli	860606-06-6060	0146789012	MNO1234	hafiz@gmail.com	10
10012	Azizah binti Yusof	850505-05-5050	0157890123	PQR5678	azizah@gmail.com	11
10013	Shafiq bin Hassan	960606-06-6060	0190123456	STU9012	shafiq@gmail.com	12
10014	Ain binti Razak	940404-04-4040	0172345678	VWX3456	ain@gmail.com	13
10015	Ismail bin Othman	850808-08-8080	0133456789	YZA5678	ismail@gmail.com	18
10016	Zainab binti Kamal	890909-09-9090	0164567890	BCD9012	zainab@gmail.com	19
10017	Nurul binti Saad	920202-02-2020	0146789012	HJL5678	nurul@gmail.com	20
10018	Faizal bin Daud	930303-03-3030	0187890123	KLM9012	faizal@mail.com	21

4. Insert (Admin, Resident And Guard)

Admin

Insert Visitor

The screenshot shows a user interface for inserting a visitor. On the left, there's a sidebar with navigation links: Dashboard, Visitor (which is selected and highlighted in pink), Fee, Report, Complaints, Registration, User List, and Logout. The main content area has a form titled "Visitor Form" with the following fields: Visitor ID (10001), Visitor Name (Kim Gaul), IC Number (040522-10-4000), Plate Number (VFD3845), Entry Time (02:20 AM), Exit Time (03:20 AM), Date of Visit (28/01/2025), Purpose of Visit (Visit Dean), and Phone Number (01243258653). At the bottom of the form is a "Submit" button.

Insert Fee

The screenshot shows a user interface for inserting a fee. On the left, a vertical sidebar has a purple header with a user icon and the text "Hi, admin". Below this are several menu items: Dashboard, Visitor, Fee (which is highlighted in a grey box), Report, Complaints, Registration, User List, and Logout. The main content area has a light blue header with the word "Fee". Below it is a "Fee Form" section with the following fields:

- User ID: A dropdown menu with the placeholder "Select".
- Fee Category ID: A dropdown menu with the placeholder "Select".
- Status ID: A dropdown menu with the placeholder "Select".
- Date: A text input field with the placeholder "dd/mm/yyyy".
- Fee Amount: A text input field with the placeholder "0,00".

At the bottom of the form is a purple "Submit" button.

Insert Report

The screenshot shows a user interface for inserting a report. On the left, a vertical sidebar has a purple header with a user icon and the text "Hi, admin". Below this are several menu items: Dashboard, Visitor, Fee, Report (which is highlighted in a grey box), Complaints, Registration, User List, and Logout. The main content area has a light blue header with the word "Report". Below it is a "Report Form" section with the following fields:

- User ID: A dropdown menu with the placeholder "Select".
- Date: A text input field with the placeholder "dd/mm/yyyy".
- Location: A text input field.
- Remarks: A text input field.
- Attachment: A file input field labeled "Choose File" with the placeholder "No file chosen".

At the bottom of the form is a purple "Submit" button.

Below the form is a table titled "List of Reports" with the following columns:

Report ID	User ID	Date	Location	Remarks	Attachment	Action
-----------	---------	------	----------	---------	------------	--------

Insert New Guard

Hi, Danial

Dashboard

Visitor

Fee

Report

Complaints

Registration

User List

Logout

Complaints

Registration

User List

Logout

Registration

Overview

Register New Guard

Register New Resident

Registration New Guard
Please fill all the field required

Name

Email Address

Identification Number

Phone Number

Plate Number

Username

Password

Confirm Password

Shift

Post Location

Salary

Submit **Cancel**

Insert New Resident

Hi, Danial

Overview

Register New Guard Register New Resident

Registration New Resident
Please fill all the field required

Name

Email Address

Identification Number

Phone Number

Plate Number

Username

Password

Confirm Password

Unit House

Logout

Submit Cancel

Resident

Insert Complaint

Complaint Form

Please Fill in the Form to Complaint to Management

Complaint Type

Date

dd/mm/yyyy

Location

Description

Attachment

Choose File No file chosen

Submit Cancel

Insert Fee

Please Fill in the Form to Pay your Fee to Management

Amount

Attachment

No file chosen

Remarks

Guard

Insert Rounding Report

Rounding Report Form
Please fill all informations

Date of Report

Location

Remarks

Attachment

No file chosen

Insert Visitors

Visitor Form
Please fill all informations

Visitor's Name

IC/Passport

Plate Number

Purpose of Visit

Phone Number

Entry Time

Date of Visit

5. Update & Delete (Admin, Resident And Guard)

Admin

Update Visitor and Delete Visitor

 Hi, admin

[Dashboard](#)

Visitor

[Fee](#)

[Report](#)

[Complaints](#)

[Registration](#)

[User List](#)

[Logout](#)

Update Visitor

Register ID:
20077

User ID:
10001

Visitor Name:
Kim Gaeul

Visitor IC:
040522-10-4000

Plate Number:
VFD3845

Entry Time:
03:03 AM

Exit Time:
04:03 AM

Date of Visit:
21/05/2025

Purpose of Visit:
huh

Phone Number:
0123456789

Update

20002	Kim Gaeul	239402-99-1295	ABC1234	2025-01-01 15:30:00	2025-01-01 16:40:00	2024-12-02 00:00:00.0	Visit Dean Ardley	01243850092	Edit Delete
-------	-----------	----------------	---------	------------------------	------------------------	--------------------------	-------------------	-------------	--

Update Fee and Delete Fee

 Hi, admin

[Dashboard](#)

[Visitor](#)

Fee

[Report](#)

[Complaints](#)

[Registration](#)

[User List](#)

[Logout](#)

Fee

Fee Form

Fee ID:
30013

User ID:
10001

Fee Category ID:
80003

Status ID:
50003

Date:
29/01/2025

Fee Amount:
5005

Update

Fee ID	User ID	Fee Date	Fee Amount	Status	Action
30013	10001	2025-01-29	RM5005	Completed	Edit Delete
30014	10001	2025-01-22	RM1000	Completed	Edit Delete

Update Report and Delete Report

Report

Update Report

Report ID: 40047

User ID: 10001

Date: 2025-01-21

Location: House #99

Remarks: none

Attachment: Choose File No file chosen

update

40047 10001 2025-01-21 House #99 none No Attachment

Edit **Delete**

Update Complaint and Delete Complaint

Complaint

Update Complaint

Complaint ID: 70023

User ID: 10001

Status ID: 50003

Complaint Type ID: 60004

Description: Cars blocking the driveway during rush hour.

Date: 2026-02-26

Location: Building A, Garage

Attachment: Choose File No file chosen

update

Complaint ID	User ID	Description	Date	Location	Attachment	Action
70023	10001	Cars blocking the driveway during rush hour.	2026-02-15	Building A, Garage	View	Edit Delete
70024	10001	Unlit streetlights in the park area.	2025-01-25	City Park	No Attachment	Edit Delete

Update User

Hi, Danial

Dashboard

Visitor

Fee

Report

Complaints

Registration

User List

Logout

Edit User Details
Please fill all the field required

Username: aliboy

Name: Dean Ardley

Identification Number: 123455

Phone Number: 01137370938

Email Address: ardleyreza197@gmail.com

Plate Number: VKC2245

Shift: Evening

Post Location: null

Salary: 0.0

Update User Cancel

Resident

Update Profile

Update Profile

Please Fill in the Form to Update Profile to Management

Email

Email

Phone Number

Phone Number

Submit

Cancel

Delete Complaint

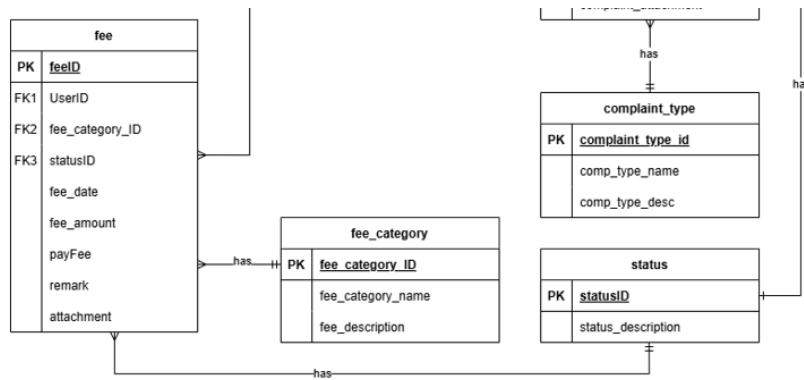
Complaints List							
Complaint ID	Complaint Type	Complaint Description	Complaint Date	Complaint Location	Attachment	Status	Action
70014	Others	Security guard absent during night shift.	2024-12-04	Main Gate	View	Completed	Delete
70023	Noise	Car damaged in parking lot.	2024-12-13	Parking Lot LakeSide	View	Processing	Delete

Guard

Update Visitor's Exit Time and Delete Visitor

ID	Visitor's Name	IC/Passport	Plate Number	Entry Time	Exit Time	Date	Purpose	Phone No	Action
20041	jiji	03112209444	YTC1928	23:05	10:31	2025-01-21	cimey	0182342345	<button>Edit</button> <button>Delete</button>
20030	atiqah	03112209444	GHADS123	12:50	13:07	2025-01-20	cimey	0199033409	<button>Edit</button> <button>Delete</button>
20031	atiqah	03112209444	GHADS123	13:03	16:31	2025-01-20	cimey	0199033409	<button>Edit</button> <button>Delete</button>
20032	atiqah	03112209444	GHADS123	13:06	16:35	2025-01-20	cimey	0199033409	<button>Edit</button> <button>Delete</button>
20033	soleha	03112209444	GHADS123	16:26	23:05	2025-01-20	cimey	0199033409	<button>Edit</button> <button>Delete</button>
20042	kokok	03112209444	YTC1928	08:57	10:25	2025-01-22	cimey	0182342345	<button>Edit</button> <button>Delete</button>
20043	jomblo	03112209444	YTC1928	10:43	10:45	2025-01-22	cimey	0182342345	<button>Edit</button> <button>Delete</button>

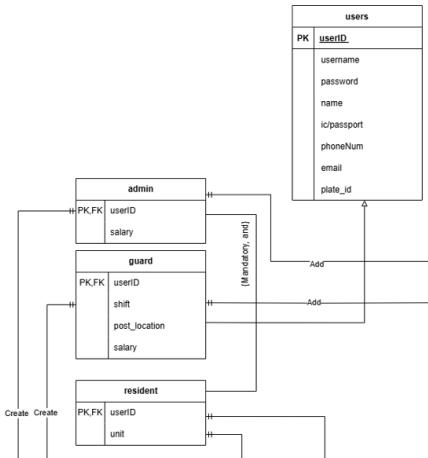
6. Bridge



List of Fees

Fee ID	User ID	Fee Date	Fee Amount	Fee Category	Status	Action
30014	10004	2025-01-27	RM100	Maintenance	Pending	<button>Edit</button> <button>Delete</button>
30011	10003	2023-12-18	RM371	Maintenance	Processing	<button>Edit</button> <button>Delete</button>
30005	10008	2024-12-12	RM371	Maintenance	Processing	<button>Edit</button> <button>Delete</button>
30013	10009	2024-12-19	RM469	Quit Rent	Pending	<button>Edit</button> <button>Delete</button>
30009	10009	2024-12-17	RM396	Quit Rent	Processing	<button>Edit</button> <button>Delete</button>
30008	10006	2024-12-17	RM425	Quit Rent	Completed	<button>Edit</button> <button>Delete</button>
30007	10004	2024-12-15	RM363	Quit Rent	Processing	<button>Edit</button> <button>Delete</button>
30006	10010	2024-12-15	RM469	Quit Rent	Pending	<button>Edit</button> <button>Delete</button>
30003	10005	2024-12-02	RM396	Quit Rent	Processing	<button>Edit</button> <button>Delete</button>
30002	10003	2024-12-02	RM425	Quit Rent	Completed	<button>Edit</button> <button>Delete</button>
30001	10002	2024-12-01	RM364	Quit Rent	Processing	<button>Edit</button> <button>Delete</button>
30004	10007	2024-12-11	RM60	Insurance	Completed	<button>Edit</button> <button>Delete</button>

7. Inheritance



10020	adnan	Adnan bin Rahman	0179012345	Resident	-	-	-	26	QRS5678	<button>Edit</button>
10021	alia	Alia binti Mokhtar	0160123456	Resident	-	-	-	27	TUV9012	<button>Edit</button>
10041	atikah	nurin atikah	0182450835	Guard	2000.0	morning	Unit 5	-	BRE7789	<button>Edit</button>
10024	hakim	Hakim bin Idris	0123456789	Guard	2000.0	Day	Main Gate, Jalan Memanah 13/55h, Jalan Memanah 13/55i, Jalan Memanah 13/55j, Swimming Pool, Gym	-	XYZ1234	<button>Edit</button>
10001	admin	Danial	0192334567	Admin	3200.0	-	-	-	JKQ6678	<button>Edit</button>
10023	hilmi	Hilmi bin Zain	0136789012	Guard	1800.0	Night	Main Gate, Jalan Memanah 13/55e, Jalan Memanah 13/55f, Jalan Memanah 13/55g, DLaman Seri PlayGround, Dewan Astana	-	JKL7890	<button>Edit</button>
10022	nadia	Nadia binti Osman	0167890123	Guard	1800.0	Day	Main Gate, Jalan Memanah 13/55a, Jalan Memanah 13/55b, Jalan Memanah 13/55c, Jalan Memanah 13/55d, Laman Seri LakeSide	-	DEF4567	<button>Edit</button>

8. Extra

i- User triggers for sequence

select * from user_triggers										
All Rows Fetched: 8 in 0.482 seconds										
TRIGGER_NAME	TRIGGER_TYPE	TRIGGERING_EVENT	TABLE_OWNER	BASE_OBJECT_TYPE	TABLE_NAME	COLUMN_NAME	REFERENCING_NAMES	WHEN_CLAUSE	STATUS	
1 TRG_FEE_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	FEE	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
2 TRG_FEE_CATEGORY_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	FEE_CATEGORY	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
3 TRG_COMPLAINT_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	COMPLAINT	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
4 TRG_COMPLAINT_TYPE_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	COMPLAINT_TYPE	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
5 TRG_STATUS_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	STATUS	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
6 TRG_REPORT_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	REPORT	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
7 TRG_VISITOR_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	VISITOR	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		
8 TRG_USERS_ID	BEFORE EACH ROW INSERT	NEIGHBORLY	TABLE	USERS	(null)		REFERENCING NEW AS NEW OLD AS OLD (null)	ENABLED tr		

7.0 Conclusion

In conclusion, the development of the Neighborly Home System (Neighborly) has been a significant learning experience, providing valuable insights into real-world system development and teamwork. Throughout this project, we applied theoretical knowledge to practical scenarios, encompassing the full system development lifecycle—from requirements gathering and system design to implementation and testing.

The Neighborly Home System (Neighborly) aims to address the identified challenges within the neighborhood, providing an efficient and user-friendly platform for residents, the Joint Management Body (JMB), and security personnel to collaborate effectively. By incorporating core functionalities such as bill payment, visitor registration, and complaint management, the system demonstrates its potential to enhance community communication and operational efficiency.

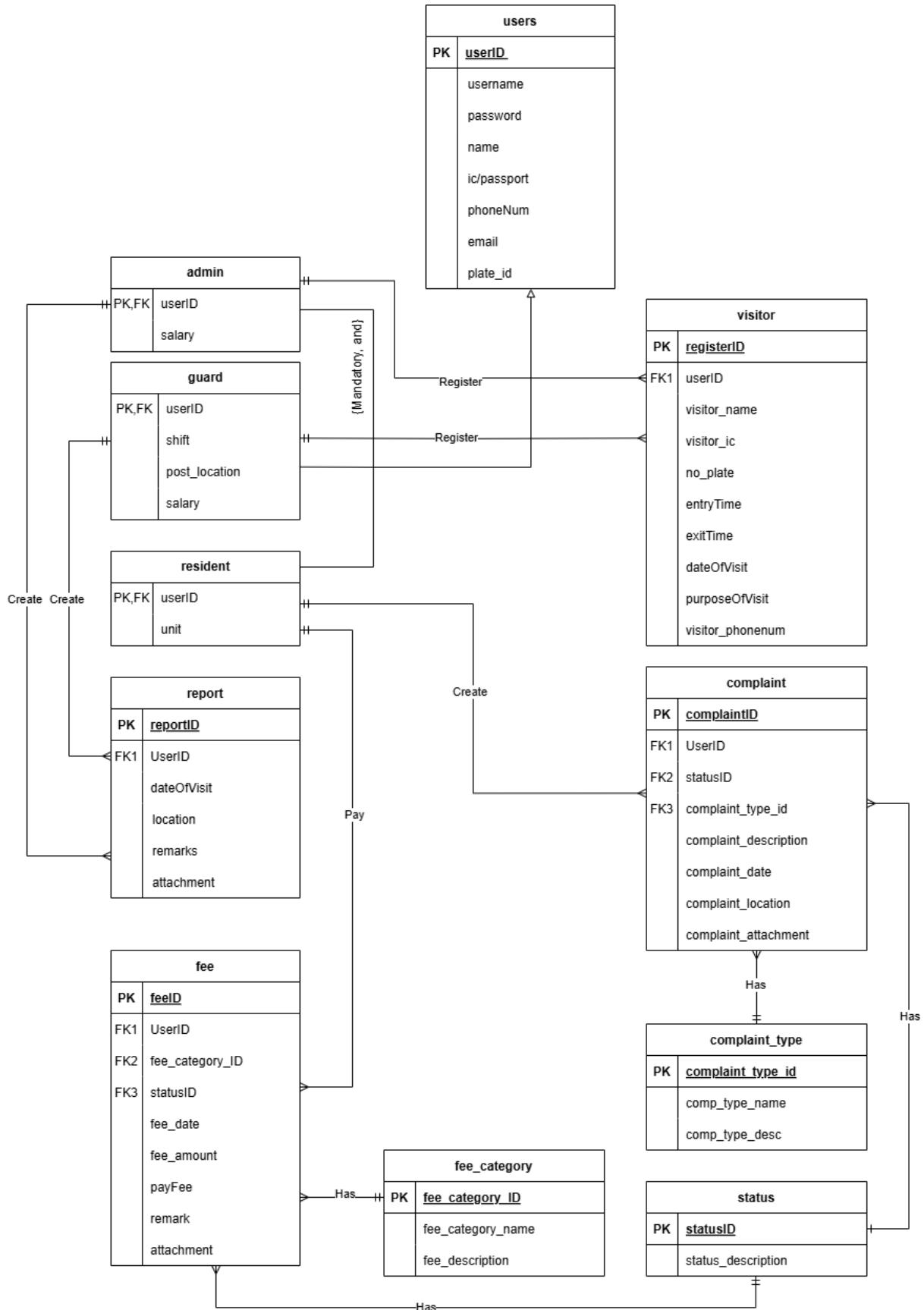
This project allowed us to refine our technical skills in areas such as system architecture, programming, and database management, while also strengthening our soft skills in communication, project management, and problem-solving. Moreover, it underscored the importance of aligning technical solutions with user needs, as well as the value of thorough testing to ensure system reliability.

Although the system meets the primary objectives outlined at the project's inception, we acknowledge that there is room for further improvement. Future enhancements could include integrating advanced features such as predictive analytics, mobile application support, and multilingual functionality to better serve a diverse user base.

Ultimately, this project not only enriched our academic journey but also prepared us for future endeavors in professional environments. The experience gained from collaborating as a team and overcoming challenges together has been invaluable, instilling a sense of achievement and paving the way for continued growth in the field of computer science.

8.0 Appendix

7.1 Entity Relationship Diagram



7.2 3NF Relationship Diagram

USERS (USERID, USERNAME, PASSWORD, NAME, IC_PASSPORT, PHONENUM, EMAIL, PLATE_ID)

ADMIN (USERID*, SALARY)

GUARD (USERID*, SHIFT, POST_LOCATION)

RESIDENT (USERID*, UNIT)

VISITOR (REGISTERID, VISITOR_NAME, VISITOR_IC, NO_PLATE, ENTRYTIME, EXITTIME, DATEOFVISIT, PURPOSEOFVISIT, VISITOR_PHONENUM, USERID*)

REPORT (REPORTID, DATEOFVISIT, LOCATION, REMARKS, ATTACHMENT, USERID*)

STATUS (STATUSID, STATUS_DESCRIPTION)

COMPLAINT_TYPE (COMPLAINT_TYPE_ID, COMP_TYPE_NAME, COMP_TYPE_DESC)

COMPLAINT (COMPLAINTID, COMPLAINT_TYPE_ID, COMPLAINT_DESCRIPTION, COMPLAINT_DATE, COMPLAINT_LOCATION, COMPLAINT_ATTACHMENT, USERID*, STATUSID*)

FEE_CATEGORY (FEE_CATEGORY_ID, FEE_CATEGORY_NAME, FEE_DESCRIPTION)

FEE (FEEID, FEE_CATEGORY_ID, STATUSID, FEE_DATE, FEE_AMOUNT, ATTACHMENT, USERID*)

7.3 Data Dictionary

Table Name	Column Name	Data Type	Description	Constraints
Users	userID	INT	Unique ID for each user	Primary Key
	username	VARCHAR (50)	Username for user login	NOT NULL
	password	VARCHAR (50)	Password for user login	NOT NULL
	name	VARCHAR (100)	Full name of the user	NOT NULL
	ic_passport	VARCHAR (20)	IC or passport number of the user	NOT NULL
	phoneNum	VARCHAR (15)	User's phone number	NULLABLE
	email	VARCHAR (100)	User's email address	NULLABLE
	plate_id	VARCHAR (20)	Vehicle plate ID	NULLABLE
	userID	INT	Unique ID referencing Users	Primary Key, Foreign Key (Users.userID)
Admin	salary	DECIMAL (10, 2)	Salary of the admin	NULLABLE
	userID	INT	Unique ID referencing Users	Primary Key, Foreign Key (Users.userID)
Guard	shift	VARCHAR (50)	Guard's working shift	NULLABLE
	post_location	VARCHAR (100)	Location of guard's assigned post	NULLA
	salary	DECIMAL (10, 2)	Salary of the admin	NULLABLE
Resident	userID	INT	Unique ID referencing Users	Primary Key, Foreign Key (Users.userID)

	unit	VARCHAR (10)	Resident's assigned housing unit	NULL
Visitor	registerID	INT	Unique registration ID for visitors	Primary Key
	userID	INT	Unique ID referencing Users	Foreign Key (Users.userID)
	visitor_name	VARCHAR (100)	Name of the visitor	NOT NULL
	visitor_ic	VARCHAR (20)	IC or passport of the visitor	NOT NULL
	no_plate	VARCHAR (20)	Visitor's vehicle plate number	NULLABLE
	entryTime	TIMESTAMP	Entry time of the visitor	NULLABLE
	exitTime	TIMESTAMP	Exit time of the visitor	NULLABLE
	dateOfVisit	DATE	Date of the visit	NULLABLE
	purposeOfVisit	VARCHAR (1000)	Purpose of the visit	NULLABLE
	visitor_phonenum	VARCHAR (15)	Visitor's phone number	NULLABLE
	registerID	INT	Unique registration ID for visitors	Primary Key
Report	reportID	INT	Unique report ID	Primary Key
	userID	INT	Unique ID referencing Users	Foreign Key (Users.userID)
	dateOfVisit	DATE	Date of the report	NULLABLE
	location	VARCHAR (200)	Location related to the report	NULLABLE

	remarks	VARCHAR (1000)	Remarks or details about the report	NULLABLE
	attachment	VARCHAR (255)	File attachment related to the report	NULLABLE
Status	statusID	INT	Unique status ID	Primary Key
	status_description	VARCHAR (200)	Description of the status	NULLABLE
Complaint_Type	complaint_type_id	INT	Unique complaint type ID	Primary Key
	comp_type_name	VARCHAR (100)	Name of the complaint type	NULLABLE
	comp_type_desc	VARCHAR (1000)	Description of the complaint type	NULLABLE
Complaint	complaintID	INT	Unique complaint ID	Primary Key
	userID	INT	Unique ID referencing Users	Foreign Key (Users.userID)
	statusID	INT	Unique ID referencing Status	Foreign Key (Status.statusID)
	complaint_type_id	INT	Unique ID referencing Complaint Type	Foreign Key (Complaint_Type.complaint_type_id)
	complaint_descript ion	VARCHAR (1000)	Description of the complaint	NULLABLE
	complaint_date	DATE	Date the complaint was made	NULLABLE
	complaint_location	VARCHAR (200)	Location related to the complaint	NULLABLE

	complaint_attachment	VARCHAR(255)	File attachment related to the complaint	NULLABLE
Fee_Category	fee_category_ID	INT	Unique fee category ID	Primary Key
	fee_category_name	VARCHAR(50)	Name of the fee category	NULLABLE
	fee_description	VARCHAR(1000)	Description of the fee category	NULLABLE
Fee	feeID	INT	Unique fee ID	Primary Key
	userID	INT	Unique ID referencing Users	Foreign Key (Users.userID)
	fee_category_ID	INT	Unique ID referencing Fee Category	Foreign Key (Fee_Category. fee_category_ID)
	statusID	INT	Unique ID referencing Status	Foreign Key (Status.statusID)
	fee_date	DATE	Date the fee was charged	NULLABLE
	fee_amount	DECIMAL(10, 2)	Amount of the fee	NULLABLE
	pay_fee	DECIMAL(10, 2)	Status of the fee	NULLABLE
	attachment	VARCHAR(255)	File attachment related to the fee	NULLABLE
	remakrs	VARCHAR(1000)	Remarks or details about the fee	NULLABLE

7.4 Data Definition Language (DDL)

1. Table Users

```
-- Users Table
CREATE SEQUENCE users_seq START WITH 10001 INCREMENT BY 1;

CREATE TABLE users (
    userID INT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    "password" VARCHAR(50) NOT NULL,
    "name" VARCHAR(100) NOT NULL,
    ic_passport VARCHAR(20) NOT NULL,
    phoneNum VARCHAR(15),
    email VARCHAR(100),
    plate_id VARCHAR(20)
);

CREATE OR REPLACE TRIGGER trg_users_id
BEFORE INSERT ON users
FOR EACH ROW
BEGIN
    :NEW.userID := users_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
USERID	NUMBER(38,0)	No	(null)	1	(null)
USERNAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
password	VARCHAR2(50 BYTE)	No	(null)	3	(null)
name	VARCHAR2(100 BYTE)	No	(null)	4	(null)
IC_PASSPORT	VARCHAR2(20 BYTE)	No	(null)	5	(null)
PHONENUM	VARCHAR2(15 BYTE)	Yes	(null)	6	(null)
EMAIL	VARCHAR2(100 BYTE)	Yes	(null)	7	(null)
PLATE_ID	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)

2. Table Admin

```
-- Admin Table
CREATE TABLE admin (
    userID INT PRIMARY KEY,
    salary DECIMAL(10, 2),
    FOREIGN KEY (userID) REFERENCES users(userID)
);
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
USERID	NUMBER(38,0)	No	(null)	1	(null)
SALARY	NUMBER(10,2)	Yes	(null)	2	(null)

3. Table Guard

```
-- Guard Table
CREATE TABLE guard (
    userID INT PRIMARY KEY,
    shift VARCHAR(50),
    salary DECIMAL(10, 2),
    post_location VARCHAR(100),
    FOREIGN KEY (userID) REFERENCES users(userID)
);
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
USERID	NUMBER(38,0)	No	(null)	1	(null)
SHIFT	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
POST_LOCATION	VARCHAR2(1000 BYTE)	Yes	(null)	3	(null)
SALARY	NUMBER(10,2)	Yes	(null)	4	(null)

4. Table Resident

```
-- Resident Table
CREATE TABLE resident (
    userID INT PRIMARY KEY,
    unit VARCHAR(10),
    FOREIGN KEY (userID) REFERENCES users(userID)
);
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
USERID	NUMBER(38,0)	No	(null)	1	(null)
UNIT	VARCHAR2(10 BYTE)	Yes	(null)	2	(null)

5. Table Visitor

```
-- Visitor Table
CREATE SEQUENCE visitor_seq START WITH 20001 INCREMENT BY 1;

CREATE TABLE visitor (
    registerID INT PRIMARY KEY,
    userID INT,
    visitor_name VARCHAR(100) NOT NULL,
    visitor_ic VARCHAR(20) NOT NULL,
    no_plate VARCHAR(20),
    entryTime TIMESTAMP,
    exitTime TIMESTAMP,
    dateOfVisit DATE,
    purposeOfVisit VARCHAR(1000),
    visitor_phonenum VARCHAR(15),
    FOREIGN KEY (userID) REFERENCES users(userID)
);

CREATE OR REPLACE TRIGGER trg_visitor_id
BEFORE INSERT ON visitor
FOR EACH ROW
BEGIN
    :NEW.registerID := visitor_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
REGISTERID	NUMBER(38,0)	No	(null)	1	(null)
USERID	NUMBER(38,0)	Yes	(null)	2	(null)
VISITOR_NAME	VARCHAR2(100 BYTE)	No	(null)	3	(null)
VISITOR_IC	VARCHAR2(20 BYTE)	No	(null)	4	(null)
NO_PLATE	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
ENTRYTIME	TIMESTAMP(6)	Yes	(null)	6	(null)
EXITTIME	TIMESTAMP(6)	Yes	(null)	7	(null)
DATEOFVISIT	DATE	Yes	(null)	8	(null)
PURPOSEOFVISIT	VARCHAR2(1000 BYTE)	Yes	(null)	9	(null)
VISITOR_PHONENUM	VARCHAR2(15 BYTE)	Yes	(null)	10	(null)

6. Table Report

```
-- Report Table
CREATE SEQUENCE report_seq START WITH 40001 INCREMENT BY 1;

CREATE TABLE report (
    reportID INT PRIMARY KEY,
    userID INT,
    dateOfVisit DATE,
    "location" VARCHAR(200),
    remarks VARCHAR(1000),
    attachment VARCHAR(255),
    FOREIGN KEY (userID) REFERENCES users(userID)
);

CREATE OR REPLACE TRIGGER trg_report_id
BEFORE INSERT ON report
FOR EACH ROW
BEGIN
    :NEW.reportID := report_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
REPORTID	NUMBER(38,0)	No	(null)	1	(null)
USERID	NUMBER(38,0)	Yes	(null)	2	(null)
DATEOFVISIT	DATE	Yes	(null)	3	(null)
location	VARCHAR2(200 BYTE)	Yes	(null)	4	(null)
REMARKS	VARCHAR2(1000 BYTE)	Yes	(null)	5	(null)
ATTACHMENT	VARCHAR2(255 BYTE)	Yes	(null)	6	(null)

7. Table Status

```
-- Status Table
CREATE SEQUENCE status_seq START WITH 50001 INCREMENT BY 1;

CREATE TABLE status (
    statusID INT PRIMARY KEY,
    status_description VARCHAR(200)
);

CREATE OR REPLACE TRIGGER trg_status_id
BEFORE INSERT ON status
FOR EACH ROW
BEGIN
    :NEW.statusID := status_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
STATUSID	NUMBER(38,0)	No	(null)	1	(null)
STATUS_DESCRIPTION	VARCHAR2(200 BYTE)	Yes	(null)	2	(null)

8. Table Complaint_Type

```
-- Complaint Type Table
CREATE SEQUENCE complaint_type_seq START WITH 60001 INCREMENT BY 1;

CREATE TABLE complaint_type (
    complaint_type_id INT PRIMARY KEY,
    comp_type_name VARCHAR(100),
    comp_type_desc VARCHAR(1000)
);

CREATE OR REPLACE TRIGGER trg_complaint_type_id
BEFORE INSERT ON complaint_type
FOR EACH ROW
BEGIN
    :NEW.complaint_type_id := complaint_type_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
COMPLAINT_TYPE_ID	NUMBER(38,0)	No	(null)	1	(null)
COMP_TYPE_NAME	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
COMP_TYPE_DESC	VARCHAR2(1000 BYTE)	Yes	(null)	3	(null)

9. Table Complaint

```
CREATE SEQUENCE complaint_seq START WITH 70001 INCREMENT BY 1;

CREATE TABLE complaint (
    complaintID INT PRIMARY KEY,
    userID INT,
    statusID INT,
    complaint_type_id INT,
    complaint_description VARCHAR(1000),
    complaint_date DATE,
    complaint_location VARCHAR(200),
    complaint_attachment VARCHAR(255),
    FOREIGN KEY (userID) REFERENCES users(userID),
    FOREIGN KEY (statusID) REFERENCES status(statusID),
    FOREIGN KEY (complaint_type_id) REFERENCES complaint_type(complaint_type_id)
);

CREATE OR REPLACE TRIGGER trg_complaint_id
BEFORE INSERT ON complaint
FOR EACH ROW
BEGIN
    :NEW.complaintID := complaint_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
COMPLAINTID	NUMBER(38,0)	No	(null)	1	(null)
USERID	NUMBER(38,0)	Yes	(null)	2	(null)
STATUSID	NUMBER(38,0)	Yes	(null)	3	(null)
COMPLAINT_TYPE_ID	NUMBER(38,0)	Yes	(null)	4	(null)
COMPLAINT_DESCRIPTION	VARCHAR2(1000 BYTE)	Yes	(null)	5	(null)
COMPLAINT_DATE	DATE	Yes	(null)	6	(null)
COMPLAINT_LOCATION	VARCHAR2(200 BYTE)	Yes	(null)	7	(null)
COMPLAINT_ATTACHMENT	VARCHAR2(255 BYTE)	Yes	(null)	8	(null)

10. Table Fee_Category

```
-- Fee Category Table
CREATE SEQUENCE fee_category_seq START WITH 80001 INCREMENT BY 1;

CREATE TABLE fee_category (
    fee_category_ID INT PRIMARY KEY,
    fee_category_name VARCHAR(50),
    fee_description VARCHAR(1000)
);

CREATE OR REPLACE TRIGGER trg_fee_category_id
BEFORE INSERT ON fee_category
FOR EACH ROW
BEGIN
    :NEW.fee_category_ID := fee_category_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
FEE_CATEGORY_ID	NUMBER(38,0)	No	(null)	1	(null)
FEE_CATEGORY_NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
FEE_DESCRIPTION	VARCHAR2(1000 BYTE)	Yes	(null)	3	(null)

11. Table Fee

```
-- Fee Table
CREATE SEQUENCE fee_seq START WITH 30001 INCREMENT BY 1;

CREATE TABLE fee (
    feeID INT PRIMARY KEY,
    userID INT,
    fee_category_ID INT,
    statusID INT,
    fee_date DATE,
    fee_amount DECIMAL(10, 2),
    payfee DECIMAL(10, 2),
    remarks VARCHAR(1000),
    attachment VARCHAR(255),
    FOREIGN KEY (userID) REFERENCES users(userID),
    FOREIGN KEY (fee_category_ID) REFERENCES fee_category(fee_category_ID),
    FOREIGN KEY (statusID) REFERENCES status(statusID)
);

CREATE OR REPLACE TRIGGER trg_fee_id
BEFORE INSERT ON fee
FOR EACH ROW
BEGIN
    :NEW.feeID := fee_seq.NEXTVAL;
END;
/
```

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
FEEID	NUMBER(38,0)	No	(null)	1	(null)
USERID	NUMBER(38,0)	Yes	(null)	2	(null)
FEE_CATEGORY_ID	NUMBER(38,0)	Yes	(null)	3	(null)
STATUSID	NUMBER(38,0)	Yes	(null)	4	(null)
FEEDATE	DATE	Yes	(null)	5	(null)
FEEMOUNT	NUMBER(10,2)	Yes	(null)	6	(null)
ATTACHMENT	VARCHAR2(255 BYTE)	Yes	(null)	7	(null)
PAYFEE	NUMBER(10,2)	Yes	(null)	8	(null)
REMARK	VARCHAR2(1000 BYTE)	Yes	(null)	9	(null)

7.5 Data Manipulation Language (DML)

1. Users Table

```
INSERT INTO users (username, "password", "name", ic_passport, phoneNum, email, plate_id)  
VALUES ('amir_hasan', 'katalaluan123', 'Amir Hasan', '900101-14-5678', '0192345678',  
'amir.hasan@example.com', 'WXD1234');
```

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
-- Insert a new user into the users table  
INSERT INTO users (username, "password", "name", ic_passport, phoneNum, email, plate_id)  
VALUES ('amir_hasan', 'katalaluan123', 'Amir Hasan', '900101-14-5678', '0192345678', 'amir.hasan@example.com', 'WXD1234');
```

Below the editor, the "Script Output" tab is selected, showing the result of the query:

Query Result | Script Output | Task completed in 0.041 seconds

1 row inserted.

At the bottom, a preview of the inserted row is shown in a table:

26	10044	amir_hasan	katalaluan123	Amir Hasan	900101-14-5678	0192345678	amir.hasan@example.com	WXD1234
----	-------	------------	---------------	------------	----------------	------------	------------------------	---------

```
DELETE FROM users WHERE username = 'amir_hasan';
```

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
-- Delete a user by username  
DELETE FROM users  
WHERE username = 'amir_hasan';
```

Below the editor, the "Script Output" tab is selected, showing the result of the query:

Script Output | Task completed in 0.043 seconds

1 row deleted.

2. Admin Table

```
INSERT INTO admin (userID, salary) VALUES (10001, 3200.00);
```

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
INSERT INTO admin (userID, salary)  
VALUES (10001, 3200.00);
```

Below the editor, the "Script Output" tab is selected, showing the result of the query:

Script Output | Task completed in 0.038 seconds

1 row inserted.

At the bottom, a preview of the inserted row is shown in a table:

USERID	SALARY
10001	3200

3. Guard Table

```
DELETE FROM guard WHERE userID = 10002;
```

The screenshot shows the SQL editor with the following code:

```
DELETE FROM guard
WHERE userID = 10002;
```

Below the editor is the Script Output window, which displays:

Script Output X
Task completed in 0.034 seconds

1 row deleted.

4. Table Resident

```
UPDATE resident SET unit = '47' WHERE userID = 10003;
```

The screenshot shows the SQL editor with the following code:

```
UPDATE resident
SET unit = '47'
WHERE userID = 10003;
```

Below the editor is the Script Output window, which displays:

Script Output X
Task completed in 0.039 seconds

1 row updated.

USERID	UNIT
10002	1
10003	47

5. Visitor Table

```
INSERT INTO visitor (userID, visitor_name, visitor_ic, no_plate, entryTime, dateOfVisit, purposeOfVisit, visitor_phonenum)
VALUES (10004, 'Nur Aina', '930504-08-5678', 'JND9087', SYSTIMESTAMP, SYSDATE, 'Ziarah keluarga', '0178765432');
```

The screenshot shows the SQL editor with the following code:

```
-- Insert a new visitor
INSERT INTO visitor (userID, visitor_name, visitor_ic, no_plate, entryTime, dateOfVisit, purposeOfVisit, visitor_phonenum)
VALUES (10004, 'Nur Aina', '930504-08-5678', 'JND9087', SYSTIMESTAMP, SYSDATE, 'Ziarah keluarga', '0178765432');
```

Below the editor is the Script Output window, which displays:

Script Output X
Task completed in 0.034 seconds

1 row inserted.

20064	10004 Nur Aina	930504-08-5678 JND9087	28/01/2025 07:30:42.336000000 (null)	28/01/2025 Ziarah keluarga	017
-------	----------------	------------------------	--------------------------------------	----------------------------	-----

6. Report Table

```
DELETE FROM report WHERE reportID = 40001;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command: `DELETE FROM report WHERE reportID = 40001;`. Below the code editor is a toolbar with icons for script, edit, run, and refresh. To the right of the toolbar is a message bar stating "Task completed in 0.038 seconds". Underneath the message bar, the output pane displays the message "1 row deleted.".

7. Status Table

```
UPDATE status SET status_description = 'Telah Diluluskan'  
WHERE statusID = 50001;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command: `UPDATE status SET status_description = 'Telah Diluluskan' WHERE statusID = 50001;`. Below the code editor is a toolbar with icons for script, edit, run, and refresh. To the right of the toolbar is a message bar stating "Task completed in 0.036 seconds". Underneath the message bar, the output pane displays the message "1 row updated.".

Below the output pane, a results grid is shown with two columns: STATUSID and STATUS_DESCRIPTION. The single row contains the value 50001 under STATUSID and 'Telah Diluluskan' under STATUS_DESCRIPTION.

8. Complaint Table

```
INSERT INTO complaint (userID, statusID, complaint_type_id, complaint_description,  
complaint_date, complaint_location, complaint_attachment)  
VALUES (10006, 50001, 60001, 'Kawasan lorong belakang terlalu bising waktu malam.',  
SYSDATE, 'Lorong Blok B', 'bising_lorong.jpg');
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command: `INSERT INTO complaint (userID, statusID, complaint_type_id, complaint_description, complaint_date, complaint_location, complaint_attachment) VALUES (10006, 50001, 60001, 'Kawasan lorong belakang terlalu bising waktu malam.', SYSDATE, 'Lorong Blok B', 'bising_lorong.jpg');`. Below the code editor is a toolbar with icons for script, edit, run, and refresh. To the right of the toolbar is a message bar stating "Task completed in 0.028 seconds". Underneath the message bar, the output pane displays the message "1 row inserted.".

70027	10002	50002	60002 Unauthorized parking at visitor area.	02/01/2025	Visitor Parking Lot	complaint_17.pdf
70032	10006	50001	60001 Kawasan lorong belakang terlalu bising waktu malam.	28/01/2025	Lorong Blok B	bising_lorong.jpg
70029	10004	50003	60001 Security guard absent during night shift.	04/01/2025	Main Gate	complaint_18.pdf

9. Fee Table

```
DELETE FROM fee WHERE feeID = 30001;
```

The screenshot shows a MySQL Workbench interface. In the main editor window, a SQL command is written:

```
DELETE FROM fee  
WHERE feeID = 30001;
```

The command is highlighted with a yellow background. Below the editor is a toolbar with icons for script, preview, results, and history. The title bar of the results tab is "Script Output". To the right of the toolbar, the status message "Task completed in 0.032 seconds" is displayed. In the results pane, the output "1 row deleted." is shown.

9.0 References

- Coronel, C., Morris, S., & Rob, P. (2019). *Database systems: Design, implementation, & management* (13th ed.). Cengage Learning.
- Oracle Corporation. (2023). *SQL basics: An introduction to SQL syntax*. Retrieved from <https://www.oracle.com/sql/>
- Teorey, T. J., Lightstone, S., Nadeau, T., & Jagadish, H. V. (2011). *Database modeling and design: Logical design* (5th ed.). Morgan Kaufmann.
- Lucidchart. (n.d.). *Entity relationship diagram (ERD) guide: Learn about ER diagrams*. from <https://www.lucidchart.com/pages/er-diagrams>
- Murach, J., & Murach, M. (2019). *Murach's SQL server 2019 for developers*. Mike Murach & Associates.
- Connolly, T., & Begg, C. (2020). *Database systems: A practical approach to design, implementation, and management* (7th ed.). Pearson.