

Amandla High School Locker Booking System

Portfolio

Portfolio

November 2025

Module ICT3715

INFORMATION AND COMMUNICATION TECHNOLOGY PROJECT

Welcome to the Portfolio

This Portfolio consists of 38 pages

The Portfolio is compulsory

Instructions:

1. Complete this Front Page (page 1).
2. Complete the Plagiarism Pledge (page 2). Your Portfolio will not be assessed without this.

STUDENT NUMBER (Student completes)									
4	2	6	8		9	7	2		4

IDENTITY NUMBER (Student completes)													
8	7	0	7	0	9	5	9	6	4	0	8	8	

PLAGIARISM PLEDGE BY THE STUDENT

- I have read Unisa's plagiarism policy.
- I understand Unisa's plagiarism policy.
- I agree to abide by Unisa's plagiarism policy.
- I have read the direct copying, plagiarism, and "patch-writing" document.
- I understand what direct copying, plagiarism, and "patch-writing" is.
- I undertake to avoid copying directly, plagiarism and patch writing.
- All academic work, written or otherwise, that I submit is expected to be the result of my own skill and labour.
- I understand that, if I am guilty of the infringement of breach of copyright/plagiarism or unethical practice, I will be subject to the applicable disciplinary code as determined by Unisa.
- The marker has the right to refuse to assess the assignment and the system if plagiarism is detected.

Student name[s] and Surname	Matome Bernard Mogobu
Student number	42689724
Date	03 November 2025
Student signature	

Additional student instructions:

1. No handwritten Portfolios will be accepted.
2. Incorrect file format Portfolio will not be considered.
3. Make sure that you did complete the instructions on page 1 of this document (template).
4. Remove everything that is placed in brackets [...].
5. Make sure that your Table of Content is updated.
6. Save the document as PDF, e.g., 12345678_ICT3715_Portfolio.pdf, (replace 1234568 with your student number).
7. When you are done submit 24 hours before your examination date.
8. Instructions where to submit will follow soon.

Plagiarism is a violation of academic integrity. Unisa has a zero tolerance for plagiarism and/or any other forms of academic dishonesty.

Here you can add your references that you have used e.g., information taken from the Internet, textbooks ...

1. Database Systems Design, Implementation and Management, 13th Edition
Authors: Carlos Coronell and Steven Morris.
2. Relational Database Design and Implementation, 4th Edition
Author: Jan L. Harrington.
3. Johnson, J. (2007). GUI bloopers 2.0: Common user interface design don'ts and do's. Elsevier.
4. Shneider man, B., Plaisant, C., Cohen, M., Jacobs, S., Elmquist, N., & Diakopoulos, N. (2016). Designing the user interface: Strategies for effective human-computer interaction (6th ed.). Pearson.
5. DuRocher, D. 2021, HTML & CSS QuickStart Guide, ClydeBank Media LLC,
6. Larsen, R. 2013, Beginning HTML and CSS, John Wiley & Sons
7. Hong, P. (2018). Practical Web Design: Learn the Fundamentals of Web Design with HTML5, CSS3, Bootstrap, jQuery, and Vue.js. Germany: Pack Publishing.
8. Jones, A. R. (2006). Mastering ASP.NET with Visual C#. Germany: Wiley.
9. Metzler, N. (2018). C# For Beginners: An Introduction to C# Programming with Tutorials and Hands-On Examples. Italy: CRB Publishing.
10. Casabona, J. (2020). HTML and CSS: Visual QuickStart Guide. United Kingdom: Pearson Education.
11. Lock, D. (2020). The Essentials of Project Management. United Kingdom: Taylor & Francis.
12. Rosenau, M. D., Githens, G. D. (2011). Successful Project Management: A Step-by-Step Approach with Practical Examples. Germany: Wiley.
13. Rakos, J., Dhanraj, K., Kennedy, S., Fleck, L., Jackson, S., Harris, J. (2015). The Practical Guide to Project Management Documentation. Germany: Wiley.
14. <https://www.w3schools.com>

15. <https://app.diagrams.net/>
 16. <https://www.visual-paradigm.com/>
 17. <http://intranet.sita.co.za> Fellow Senior Software Development Colleague.
Wellington Molemane (Application Development and Maintenance-ADM)
 18. <http://www.dod.mil.za/> Fellow Software Development Colleague. Sharon
Seroka (ADM) DOD Unique
 19. <https://learn.microsoft.com/en-us/training/browse/>
 20. <http://intranet.sita.co.za> Irene Kgokane-Project Manager DOD client.
 21. <http://intranet.sita.co.za> Linky Tabane-Database Administrator DOD.
-

Table of Contents

1. User Guide	6
Introduction.....	9
Scope and Purpose	9
Quick Start Guide.....	10
Process Parent Registration and Login.....	11
Steps of Process Parent Registration and Login	12
Process Administrato Login	12
Steps of Process Administrato Login	12
2. Business Plan.....	13
System Overview.....	14
Technology requirements	15
Hardware.....	15
Software.....	16
Obstacles	16
Deployment	16
Installation	16
Training.....	17
Testing.....	17
Backup and recovery.....	17
Documentation.....	18
Reporting.....	18
Evidence.....	18
3. Assessment 3	21
4. Assessment 2	29
5. Assessment 1.....	37

1. User Guide

User Guide for Amandla High School Locker Booking System

November 2025

Instructions:

- ❖ Change everything that is in brackets []
- ❖ Add your practical system content to the document.
- ❖ Make sure that your Table of Content is updated.
- ❖ Update the document revision table.
- ❖ Complete all the Process and Process Steps for your practical system in this document

Document Revisions

Date	Version Number	Document Changes
01/07/2025	0.1	Initial Draft
06/08/2025	0.2	Updated Installation process
09/09/2025	0.3	Added reports
03/10/2025	0.4	Updated Business plan
07/10/2025	0.5	Final version

Introduction

Scope and Purpose

❖ Introduce the practical system and its purpose.

Amandla High School Locker Booking System is a digital solution designed to simplify and improve the process of applying or registering, assigning and managing student lockers at Amandla High School. Its main purpose are:

- Eliminate manual paperwork and reduce administrative workload, that is efficiency.
- Prevent double booking of lockers.
- Maintain accurate and up-to-date locker assignment records.
- To ensure that all data on the system is accurate and secured.

❖ Highlighting key features and benefits.

- **User Authentication:** Secure login for both parents and administrators to protect user data and ensure authorized access only.
- **Administrative Dashboard:** Administrator can manage all bookings, approve or deny requests, and monitor locker usage through a centralized system.
- **Parent Dashboard:** Parent can be able to apply for their student, submit proof of payment online or by replying to email from school. Also to apply for locker for their student.
- **Database Integration:** All locker, student, and booking information is stored in a secure database, ensuring data accuracy, quick retrieval, and easy updates.
- **Email Notification System:** Automated email notifications can be sent to parents upon successful booking, approval, or cancellation, keeping users informed. Also to inform them to make payment if booking/application is successful.
- **Search Options:** Enables quick searching of lockers or students, making it easier for administrator to manage records efficiently.

❖ Introduce the purpose of the user guide.

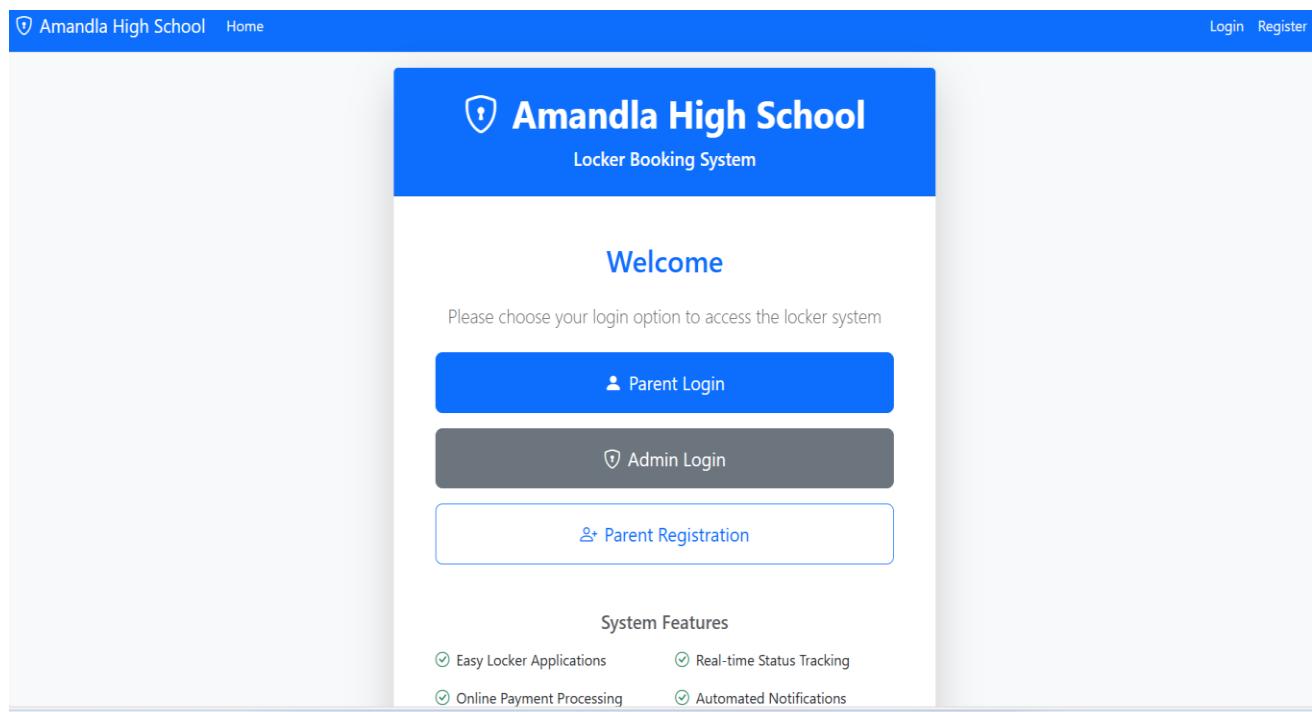
- The **purpose of the user guide** is to provide clear and detailed instructions for users of the **Amandla High School Locker Booking System**, ensuring that both parents and administrators can use the system effectively and efficiently.

Quick Start Guide

Provide an overview of the functionality of the practical system and how the user will interact with the practical system.

The system provides main interfaces: for Parents and another for administrators, each tailored to their specific needs. However, Parents needs to first register on the system.

The user (Parent or Admin) will click on their respective login pages to access the system.



Parent Registration Process

The screenshot shows the 'Parent Registration' page of the Amandla High School Locker Booking System. At the top, there's a blue header bar with the school logo and 'Home' link on the left, and 'Login' and 'Register' links on the right. Below the header is a large blue box containing the registration form. The form fields include:

- Parent ID Number:** A text input field with placeholder text "Enter your ID number".
- First Name:** A text input field with placeholder text "Enter your first name".
- Surname:** A text input field with placeholder text "Enter your surname".
- Mobile Number:** A text input field with placeholder text "Enter your mobile number".
- Home Address:** A text input field with placeholder text "Enter your full home address".
- Email Address:** A text input field with placeholder text "Enter your email address".
- Password:** A text input field with placeholder text "Create a password".
- Confirm Password:** A text input field with placeholder text "Confirm your password".

Below the form fields is a checkbox labeled "I agree to the Terms and Conditions". At the bottom of the blue box is a blue button with the text "Register Account". Outside the blue box, at the bottom center, is a smaller text link "Already have an account? Login here".

Steps of Parent Registration

- Parent ID number
- First Name and Surname
- Mobile number
- Home Address
- Email Address
- password and confirm password
- Lastly click Register.

Process of Parent Login

The screenshot shows the login interface for the Amandla High School Locker System. At the top, there's a blue header bar with the title "Login" and the subtitle "Amandla High School Locker System". Below the header, there's a dropdown menu labeled "I am a:" with "Parent" selected. There are two input fields: "Email Address" containing "liam.Nkosi@gmail.com" and "Password" containing "*****". Below these fields is a large blue "Login" button with a key icon. At the bottom of the form, there are two links: "Don't have an account? Register as Parent" and "[← Back to Home](#)".

Steps of parent login:

- Sign in with the credentials you have registered with.
e.g liam.Nkosi@gmail.com and Password. Parent dashboard will appear, and parent should be able to add their student and apply for locker for 2026 academic year.

Administrator Login Process

The screenshot shows the login interface for the Amandla High School Locker System. It has a similar structure to the parent login page, with a blue header bar, a dropdown menu set to "Administrator", and input fields for "Email Address" (admin@locker.com) and "Password" (*****). A large blue "Login" button is present. At the bottom, there are two links: "Don't have an account? Register as Parent" and "[← Back to Home](#)".

Steps: Administrator username and password then login. once logged in will be able to see admin dashboard and manage students, lockers view reports etc.
e.g admin@locker.com

2. Business Plan

SOFTWARE DEVELOPMENT PROPOSAL | Business Plan

Prepared for: Amandla High School

Prepared by: Matome Bernard Mogobu

Student Number: 42689724

Date: 05 November 2025

Dear, Amandla High School

RE: Enclosed Software Development Proposal

My name is **Matome Bernard Mogobu**, a dedicated software developer with practical experience in designing and implementing database-driven systems using **C#** and **MySQL**. I am confident that my technical background and passion for software development uniquely qualify me to develop a reliable, scalable, and user-friendly system tailored to your school's needs.

This proposal outlines the development plan, technical design, and operational requirements for the software solution.

In summary, my technical background, analytical skills, and understanding of system development life cycles give me the expertise required to design and implement the **Amandla High School Locker Booking System** effectively.

Kind regards,

Matome Bernard Mogobu

Student Number: 42689724

System Overview

The proposed system is a **Locker Management and Administration System** developed using **C# (Windows Forms)** for the front end and **MySQL** for the backend database. The primary aim of this system is to streamline and automate locker allocation, payment tracking, and management processes for administrators, students, and support staff.

Aims

- To replace manual locker recordkeeping with a centralized digital system.
- To improve administrative efficiency and accuracy.
- To provide real-time reports for management decision-making.

Scope

The system includes modules for:

- **Locker Management:** Add, update, delete, and assign lockers.
- **Student Management:** View and manage student profiles.
- **Payment Management:** Track locker fees and payment status.
- **Reports Dashboard:** Generate MIS reports (daily, weekly, and analytical).

System Operation

The system will be installed on the client's computers and connected to a central MySQL server database. Administrators can log in to manage records, generate reports, and confirm locker assignments. The system is designed to handle concurrent users securely.

Design Process

The system follows the **Waterfall Model**, including phases of requirements analysis, system design, implementation, testing, and maintenance. The GUI is designed for simplicity and accessibility, with clear navigation menus and validation checks.

Functionality

- Secure admin authentication.
- Locker CRUD (Create, Read, Update, Delete) operations.
- Waiting list and payment tracking.
- Automated report generation.

Choice of Technology

- **Frontend:** C# (Windows Forms)
- **Backend:** MySQL
- **IDE:** Visual Studio 2022
- **Database Tool:** MySQL Workbench

Distribution and Ownership

The software will be distributed to the client as an executable Windows application.

Ownership of intellectual property and licensing rights will belong to the client after project completion and full payment.

Testing, Support, and Warranties

Thorough testing (unit, integration, and user acceptance testing) will be conducted before deployment. Post-deployment support will be available for one month after handover, including bug fixes and basic system training.

Technology Requirements

Hardware

- The system will run on:
 - 32-bit and 64-bit Windows desktop systems (Windows 8, 10, and 11).
- Minimum system requirements:
 - Intel i3 processor or higher
 - 4GB RAM
 - 500MB disk space
 - Stable internet/network connection (for database synchronization)

Software

- **Programming Language:** C# (.NET Framework 8.0 or later)
 - **Database Management System:** MySQL
 - **Server:** MySQL Community Server 8.x
 - **IDE:** Microsoft Visual Studio
 - **Other Tools:** phpMyAdmin (optional for DB management), GitHub (version control)
 - **XAMPP 8.2.12-0 installed**
-

Obstacles

POTENTIAL RISK	IMPACT	MITIGATION STRATEGY
Integration issues between C# and MySQL	Moderate	Use MySQL Connector/.NET and test database queries early
Data security concerns	High	Implement login authentication and role-based access control
User resistance to new system	Moderate	Provide training and user-friendly documentation
Power or network downtime	Low	Enable offline caching and local backups

Deployment

Installation

The software will be delivered as an installer package (Setup.exe) along with a MySQL database script. Installation steps include:

1. Installing the application using the provided setup wizard.
2. Importing the database schema via MySQL Workbench or command line.
3. Configuring connection settings in app.config.

Updates

Future updates will be distributed through downloadable update files or patches that can be installed without affecting existing data.

Training and Support

Training will be provided to administrative users through:

- A **one-day on-site training session** covering system navigation, reporting, and backup procedures.
 - A **user manual** (digital PDF format).
post-deployment support will be available for one month, with additional support available through service agreements.
-

Testing

Testing will include:

- **Unit Testing:** Each function tested for correctness.
- **Integration Testing:** Ensuring seamless connection between modules.
- **User Acceptance Testing (UAT):** The client verifies that system requirements are met.

The client's role in testing will include providing sample data and verifying system performance.

Backup and Recovery

The system will perform daily automated database backups. Backup files can be stored locally and or in a secure cloud environment (GitHub, Ms onedrive and or Google drive). Recovery will be handled by restoring the latest backup file into MySQL using SQL import commands.

Documentation

The following documents will accompany the system:

- Installation Guide
- User Manual
- Technical Documentation (for administrators)
- Troubleshooting Guide
- System Design Document

All documents will be provided in **PDF format** and accessible from within the system's "Help" menu.

Reporting

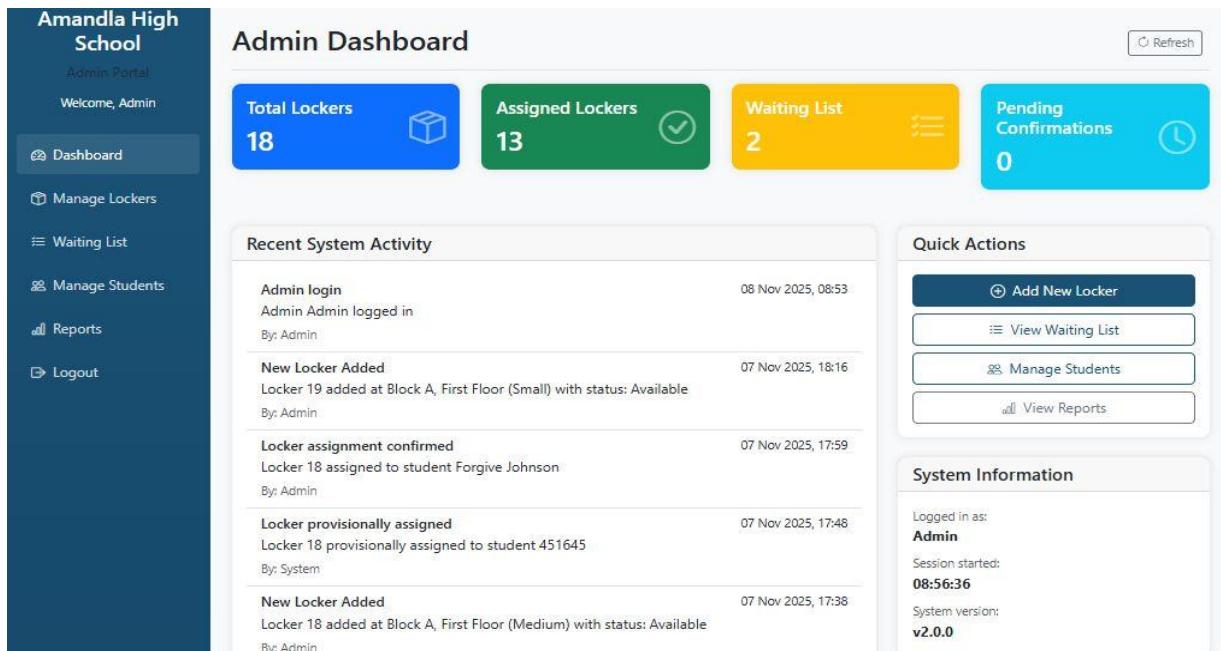
The system will include the following **Management Information System (MIS) reports:**

1. **Daily Reports:** Students on the waiting list and assigned lockers.
2. **Weekly Reports:** Total locker usage by grades
3. **Payment Tracking Reports:** Outstanding and completed payments.
4. **Locker Utilization Reports:** To show in percentage sign

Evidence

Below are examples of developed user interfaces included in the current version of the system:

1. **Admin Dashboard:** Displays real-time locker and payment statistics.



The screenshot shows the Admin Dashboard interface for Amandla High School. The dashboard features a header with the school name and a "Refresh" button. Below the header are four summary cards: "Total Lockers" (18), "Assigned Lockers" (13), "Waiting List" (2), and "Pending Confirmations" (0). The main area contains two sections: "Recent System Activity" and "Quick Actions".

Recent System Activity	
Admin login	08 Nov 2025, 08:53
Admin Admin logged in	
By: Admin	
New Locker Added	07 Nov 2025, 18:16
Locker 19 added at Block A, First Floor (Small) with status: Available	
By: Admin	
Locker assignment confirmed	07 Nov 2025, 17:59
Locker 18 assigned to student Forgive Johnson	
By: Admin	
Locker provisionally assigned	07 Nov 2025, 17:48
Locker 18 provisionally assigned to student 451645	
By: System	
New Locker Added	07 Nov 2025, 17:38
Locker 18 added at Block A, First Floor (Medium) with status: Available	
By: Admin	

Quick Actions	
+ Add New Locker	
≡ View Waiting List	
⌘ Manage Students	
✉ View Reports	

System Information	
Logged in as:	Admin
Session started:	08:56:36
System version:	v2.0.0

2. Locker Management Screen: Add, edit, and delete locker details.

Manage Lockers

Locker Management

Total Lockers	Assigned	Available	Maintenance
18	13	1	2

Show 10 lockers per page

Search lockers:

Locker ID	Location	Size	Status	Assigned To	Actions
#1	Block A, Ground Floor	Small	Assigned	Olivia Wilson (110216)	
#10	Near Library	Small	Assigned	Forgive Moatshe (420182)	
#11	Block C, Ground Floor	Small	Assigned	Anekie Moatshe (386815)	
#12	Block C, Ground Floor	Medium	Reserved	Not assigned	
#14	Block D, Ground Floor	Small	Assigned	Andisa Nkosi (638043)	
#15	Block D, Ground Floor	Medium	Assigned	Dakalo Sesame (519818)	
#16	Block A, Ground Floor	Small	Assigned	Emmanuel Wilson (972595)	
#17	Block A, Ground Floor	Medium	Assigned	Olivia Johnson (617741)	
#18	Block A, First Floor	Medium	Assigned	Forgive Johnson (451645)	

3. Students List Screen: Displays student information linked to lock assignments.

Manage Students

Student Management

Total Students	With Lockers	Waiting List	No Locker
57	5	1	52

Show 10 students per page

Search students:

Student No	Name	Grade	Locker	Parent	Status	Actions
895067	Alice Taylor	Grade 9	No locker assigned	Jane Taylor	No Locker	
407505	Emily Anderson	Grade 9	No locker assigned	Liam Anderson	No Locker	
S1001	Emma Smith	Grade 9	No locker assigned	John Smith	No Locker	
916977	James Taylor	Grade 9	No locker assigned	Robert Taylor	No Locker	
390412	James Wilson	Grade 9	No locker assigned	Robert Wilson	No Locker	
309568	Jane Wilson	Grade 9	No locker assigned	Ethan Wilson	No Locker	
739637	John Nkosi	Grade 9	No locker assigned	John Nkosi	No Locker	
366289	Liam Anderson	Grade 9	No locker assigned	Olivia Anderson	No Locker	
878977	Noah Anderson	Grade 9	No locker assigned	Michael Anderson	No Locker	

4. Reports Module: Generates PDF and on-screen MIS reports.

Total Lockers	Assigned	Available	Maintenance
18	13	1	2

5. Payment Management Screen: Track payment history and outstanding fees.

Status	Count	Total Amount
confirmed	6	R 600.00
pending	4	R 400.00

Each GUI is designed for ease of use, clarity, and consistency, ensuring administrators can perform all operations efficiently.

Prepared by: Matome Bernard Mogobu

Student Number: 42689724

Date: 10 November 2025

Signature: 

3. Assessment 3

Question 1 [20]

Management Information Systems [MIS] Reports

Create the following MIS Reports

Refer to Task 10.

1.1 Locker Usage by Grade Report (10)

1.1.1 Locker usage for Grade 8 and Grade 11

```

String gradeQuery = """
    SELECT s.grade_no AS grade, COUNT(s.locker_id) AS assigned
    FROM student s
    WHERE s.grade_no IN ('Grade8', 'Grade11') AND s.locker_id IS NOT NULL
    GROUP BY s.grade_no
""";
```

```

List<Map<String, Object>> gradeData = jdbcTemplate.queryForList(gradeQuery);

// Total lockers
int totalLockers = jdbcTemplate.queryForObject(
    "SELECT COUNT(*) FROM locker", Integer.class);

return Map.of(
    "success", true,
    "data", Map.of(
        "gradeData", gradeData,
        "totalLockers", totalLockers
    ),
    "message", "Locker usage by grade report generated"
);
}
```

This report provides insights into locker allocation and utilization among students in **Grade 8 and Grade 11**.

Key Benefits for the School:

1. Resource Allocation

- Helps administrators determine if there are enough lockers for students in these grades.
- Identifies if some grades have **overutilized** or **underutilized** lockers.

2. Fair Distribution

- Ensures equitable access to lockers across different grades.

- Helps prevent situations where some students are left without lockers while others have spares.

3. Future Planning

- If **Grade 8** has high demand, the school may need to **increase locker capacity** for incoming students.
- If **Grade 11** has low usage, lockers could be **reallocated** to other grades.

4. Conflict Resolution

- If locker shortages occur, the school can implement a **waiting list** or **priority system** based on grade needs.

1.2 Locker Booking Summary Report (10)

1.2.1 Summarize the number of lockers booked for period January 2026 to June 2026

```
@RequestParam(defaultValue = "2026-01-01") String startDate,
@RequestParam(defaultValue = "2026-06-30") String endDate {
```

```
String query = """
SELECT
    b.booking_id,
    CONCAT(s.name, ' ', s.surname) AS student_name,
    s.grade_no,
    l.locker_id,
    l.location,
    b.booking_date,
    b.status
FROM booking b
JOIN student s ON b.student_id = s.student_no
LEFT JOIN locker l ON b.locker_id = l.locker_id
WHERE b.booking_date BETWEEN ? AND ?
ORDER BY b.booking_date
""";
```

```
List<Map<String, Object>> bookings = jdbcTemplate.queryForList(
    query, startDate, endDate);
```

```
return Map.of(
    "success", true,
    "data", bookings,
    "message", "Booking summary report generated"
);
```

This report summarizes locker bookings over a **6-month period**, showing trends and demand patterns.

Key Benefits for the School:

1. Demand Analysis

- Identifies **peak booking periods** (e.g., start of the academic year).
- Helps predict future demand for lockers.

2. Operational Efficiency

- Allows the school to **schedule maintenance** during low-usage periods.
- Helps in **staff allocation** (e.g., more admin support during high-demand months).

3. Financial Planning

- If bookings are declining, the school may need to **promote locker rentals**.
- If bookings are consistently high, the school could consider **expanding locker facilities**.

4. Student Experience

- Ensures students get lockers when they need them.
- Reduces complaints about **unavailability** or **long waiting times**.

The screenshot shows a left sidebar with navigation links: Admin Portal, Dashboard, Manage Lockers, Waiting List, Manage Students, and Reports. The main content area has two sections: 'Locker Usage by Grade' and 'Locker Booking Summary'.

Locker Usage by Grade

Grade	Lockers Assigned	Percentage Used
Grade11	2	13.33%
Grade8	1	6.67%

Locker Booking Summary

Student	Grade	Locker	Booking Date	Status
Liam Nkosi	Grade10	1	Feb 5, 2026	waiting
Olivia Wilson	Grade11	3	Jan 1, 2026	waiting

Question 2 [20]

Graphical User Interfaces

Create the following MIS Reports

Refer to Task 11.

It is now time to start with the “coding” and create the front-end Graphical User Interfaces (GUI).

1. Create a Graphical User Interface for the parent, or the administrator to login to the system.
2. When you are done, make a screenshot of the interface, and also provide the code for the GUI.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Locker System Login</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
    .login-container {
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        background-color: #f8f9fa;
    }

    .login-card {
        width: 400px;
        padding: 30px;
        background: white;
        border-radius: 8px;
        box-shadow: 0 0 20px rgba(0,0,0,0.1);
    }

    .login-card h2 {
        text-align: center;
        margin-bottom: 20px;
        color: #343a40;
    }

    .form-group {
        margin-bottom: 15px;
    }

    .btn-login {
        width: 100%;
        padding: 10px;
        margin-top: 10px;
    }

    .nav-tabs {
```

```
margin-bottom: 20px;
border-bottom: 1px solid #dee2e6;
}

.nav-link {
color: #495057;
}

.nav-link.active {
font-weight: 500;
color: #0d6efd;
}

```

</style>

</head>

<body>

<div class="login-container">

<div class="login-card">

<h2>Locker System Login</h2>

<ul class="nav nav-tabs" id="loginTabs">

<li class="nav-item">

Parent

<li class="nav-item">

Admin

<div class="tab-content">

<div id="parent" class="tab-pane active">

<form th:action="@{/login}" method="post">

<input type="hidden" name="userType" value="parent">

<div class="form-group">

<label for="parentEmail">Email</label>

<input type="email" class="form-control" id="parentEmail" name="email" required>

</div>

<div class="form-group">

<label for="parentPassword">Password</label>

<input type="password" class="form-control" id="parentPassword" name="password" required>

</div>

<button type="submit" class="btn btn-primary btn-login">Login</button>

</form>

</div>

<div id="admin" class="tab-pane fade">

```
<form th:action="@{/login}" method="post">
    <input type="hidden" name="userType" value="admin">
    <div class="form-group">
        <label for="adminEmail">Email</label>
        <input type="email" class="form-control" id="adminEmail" name="email"
required>
    </div>
    <div class="form-group">
        <label for="adminPassword">Password</label>
        <input type="password" class="form-control" id="adminPassword"
name="password" required>
    </div>
    <button type="submit" class="btn btn-primary btn-login">Login</button>
</form>
</div>
</div>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

The screenshot shows a login interface titled "Locker System Login". At the top, there are two tabs: "Parent" (which is currently selected) and "Admin". Below the tabs are two input fields: "Email" and "Password", each with a corresponding input box. At the bottom is a large blue button labeled "Login".

Parent Dashboard

Welcome

Welcome to the Amandla High School Locker Booking System.

From here you can:

- Apply for a locker for your student
- View locker assignments
- Make payments
- View your students' information

Quick Actions

Apply for Locker

Make Payment

Recent Notifications

Welcome to the system

Thank you for registering with the Amandla High School Locker Booking System.

Activate Windows Today
Go to Settings to activate Windows.

4. Assessment 2

Question 1 [10]

Cleaning the data

1.1 Download the data file (Parent_Student_Data) from Additional Resources on the myModules 2025 Site.

Refer to Task 8 regarding cleaning data.

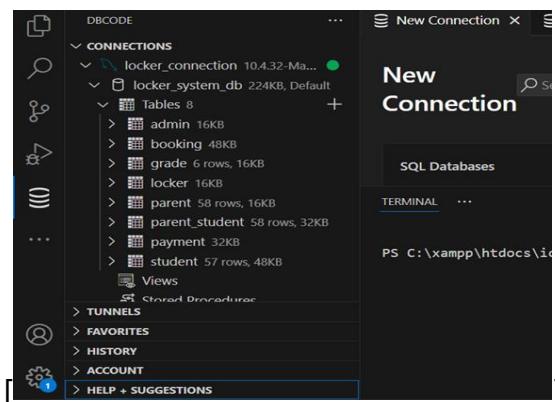
1.2 Follow the process to clean the data. Attached the clean data file to the end of your assessment.

Question 2 [20]

Create the database

2.1 Create the database structure and database tables for your system according to your ERD diagram.

2.2 Import the data from the data file into your database.



[

The screenshot shows three separate database connections in MySQL Workbench:

- Connection 1: locker_connection - locker_system_db - parent**

#	abc ParentID	Name	Surname	Email	MobileNo	Address
1	1008113111123	Ethan	Wilson	ethan.wilson...	0737248373	636 Cedar Street
2	202019503196	Jane	Taylor	jane.taylor...	0781998910	911 Ash Street
3	3018701182	Mary	Johnson	mary.j@gmail...	0832345678	456 Oak Ave
4	306052541142	Emma	Anderson	emma.anderso...	0721206388	704 Willow Stre...
5	4108192658003	Olivia	Anderson	olivia.ander...	0714962168	404 Cedar Street
6	4109137427178	James	Thomas	james.thomas...	0789012345	246 Redwood Cou...
7	4204287984104	Liam	Nkosi	liam.nkosi@g...	0744083306	433 Birch Street
8	4207012946087	Emma	Smith	emma.smith@g...	0749100685	718 Ash Street
9	4409125431129	John	Nkosi	john.nkosi@g...	0799536039	436 Elm Street
10	4510283636054	Noah	White	noah.white@g...	0777393881	604 Pine Street
11	4608148192183	John	Anderson	john.anderso...	0777769694	789 Willow Stre...
12	4609139102037	Olivia	Smith	olivia.smith...	0763380717	216 Oak Street
13	4788037500102	Ethan	Wilson	ethan.wilson...	0718425293	500 Birch Street
14	4803233540087	Ethan	Nkosi	ethan.nkosi@g...	0722944641	336 Willow Stre...
15	4809018674075	Robert	Taylor	robert.taylo...	0743877000	162 Ash Street
16	4809209872196	Robert	Smith	robert.smith...	0768346836	757 Cedar Street
17	4901218928125	Michael	Thomas	michael.thom...	0711790361	104 Elm Street
18	5004038523192	Linda	Anderson	linda.anders...	0778901234	135 Spruce Stre...

- Connection 2: locker_connection - locker_system_db - parent_student**

#	abc ParentID	abc StudentNo
1	1008113111123	647927
2	202019503196	895067
3	3018701182	357874
4	306052541142	528834
5	4108192658003	366289
6	4109137427178	910983
7	4204287984104	720053
8	4207012946087	558556
9	4409125431129	739637
10	4510283636054	801075
11	4608148192183	977032
12	4609139102037	801734
13	4788037500102	309568
14	4803233540087	382254
15	4809018674075	916977
16	4809209872196	271520
17	4901218928125	922628
18	5004038523192	200443

- Connection 3: locker_connection - locker_system_db - student**

#	abc StudentNo	GradeNo	Name	Surname	LockerID
1	110216	Grade9	Olivia	Wilson	(null)
2	163688	Grade8	Emma	Wilson	(null)
3	173463	Grade9	Michael	Anderson	(null)
4	200443	Grade11	Ava	Anderson	(null)
5	217985	Grade10	Liam	Johnson	(null)
6	221375	Grade11	Ava	Taylor	(null)
7	248256	Grade10	Liam	Nkosi	(null)
8	264399	Grade8	Robert	Thomas	(null)
9	271520	Grade8	Alice	Smith	(null)
10	277788	Grade10	Olivia	Nkosi	(null)
11	287688	Grade8	David	Nkosi	(null)
12	309568	Grade9	Jane	Wilson	(null)
13	312754	Grade10	Liam	Smith	(null)
14	317750	Grade10	Ava	Smith	(null)
15	357874	Grade8	Liam	Johnson	(null)
16	366289	Grade9	Liam	Anderson	(null)
17	382254	Grade8	Michael	Nkosi	(null)
18	399412	Grade9	James	Wilson	(null)

[Please make sure that the screenshots is readable!]

[
CREATE DATABASE locker_system_db;
USE locker_system_db;

```
CREATE TABLE Grade (
    GradeNo VARCHAR(10) PRIMARY KEY
);
```

```
CREATE TABLE Locker (
    LockerID INT PRIMARY KEY,
    Location VARCHAR(50),
    Size VARCHAR(20),
    Status VARCHAR(20)
);
```

```
CREATE TABLE student (
    StudentNo VARCHAR(10) PRIMARY KEY,
    GradeNo VARCHAR(10),
    Name VARCHAR(50),
    Surname VARCHAR(50),
    LockerID INT,
    FOREIGN KEY (GradeNo) REFERENCES Grade(GradeNo),
    FOREIGN KEY (LockerID) REFERENCES Locker(LockerID)
);
```

```
CREATE TABLE parent (
    ParentID VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(50),
    Surname VARCHAR(50),
    Email VARCHAR(100),
    MobileNo VARCHAR(15),
    Address TEXT
);
```

```
CREATE TABLE Payment (
    PaymentID INT AUTO_INCREMENT PRIMARY KEY,
    ParentID VARCHAR(20),
    Amount DECIMAL(10,2),
    PaymentDate DATE,
    ProofOfPayment VARCHAR(255),
    FOREIGN KEY (ParentID) REFERENCES Parent(ParentID)
);
```

```
CREATE TABLE Booking (
```

```
BookingID INT AUTO_INCREMENT PRIMARY KEY,  
StudentID VARCHAR(10),  
LockerID INT,  
BookingDate DATE,  
Status VARCHAR(20),  
FOREIGN KEY (StudentID) REFERENCES Student(StudentNo),  
FOREIGN KEY (LockerID) REFERENCES Locker(LockerID)  
);
```

```
CREATE TABLE Admin (  
AdminID INT AUTO_INCREMENT PRIMARY KEY,  
Name VARCHAR(50),  
Email VARCHAR(100),  
Password VARCHAR(255)  
);
```

```
CREATE TABLE parent_student (  
ParentID VARCHAR(20),  
StudentNo VARCHAR(10),  
PRIMARY KEY (ParentID, StudentNo),  
FOREIGN KEY (ParentID) REFERENCES Parent(ParentID),  
FOREIGN KEY (StudentNo) REFERENCES Student(StudentNo)  
);...]
```

Cleaned Data:



Cleaned_Parent_Student_Data (1).csv

Please see below table its same as data inside excel spreadsheet attached.

Parent Title	Parent ID Number	Parent Name	Parent Surname	Parent Email Address	Parent Home Address	Phone Number	Student Number	Student Name	Student Surname	Student Grade
Mr	7110115033116	John	Smith	john.smith@gmail.com	123 Maple Street	0821234567	331001	Emma	Smith	Grade 9
Mrs	3018701182689	Mary	Bernard Mogobu	mary.johnson@gmail.com	456 Oak Ave	0832345678	357874	Liam	Johnson	Grade 8
Ms	9504200125035	Linda	Brown	linda.brown@gmail.com	789 Pine Rd	0843456789	550037	Olivia	Brown	Grade 10
MR	9411277787016	David	Johnson	david.johnson@gmail.com	321 Maple Lane	0745678901	217985	Liam	Johnson	Grade 10
Mrs	9298654199864	Emma	Wilson	emma.wilson@gmail.com	654 Cedar Drive	0756789012	110216	Olivia	Wilson	Grade 9
Dr	9212011817041	Robert	Taylor	robert.taylor@gmail.com	987 Birch Boulevard	0767890123	764982	Noah	Taylor	Grade 8
Ms	5004038523192	Linda	Anderson	linda.anderson@gmail.com	135 Spruce Street	0778901234	200443	Ava	Anderson	Grade 11
Mr	4109137427178	James	Thomas	james.thomas@gmail.com	246 Redwood Court	0789012345	910983	Ethan	Thomas	Grade 10
Mrs	5612226663035	Robert	Wilson	robert.wilson@gmail.com	189 Pine Street	0735460310	390412	James	Wilson	Grade 9
Mr	9101735471758	Michael	Anderson	michael.anderson@gmail.com	702 Maple Street	0775510858	878977	Noah	Anderson	Grade 9
Ms	4207012946087	Emma	Smith	emma.smith@gmail.com	718 Ash Street	0749100605	558556	Liam	Smith	Grade 10
Mrs	8305068147131	Ava	Smith	ava.smith@gmail.com	599 Spruce Street	0783072551	317750	Ava	Smith	Grade 10
Mr	9505017498148	Ava	Hall	ava.hall@gmail.com	332 Maple Street	0791798774	453633	Emily	Hall	Grade 11
Ms	6602050571067	Liam	White	liam.white@gmail.com	842 Birch Street	0737988829	611784	Jane	White	Grade 8
Mrs	9012129123195	Noah	Hall	noah.hall@gmail.com	190 Willow Street	0743682129	582349	John	Hall	Grade 10
Dr	3060525411428	Emma	Anderson	emma.anderson@gmail.com	704 Willow Street	0721206388	528034	Jane	Anderson	Grade 11
Ms	9507203452020	Robert	Anderson	robert.anderson@gmail.com	875 Maple Street	0739094911	173463	Michael	Anderson	Grade 9
Mr	6806020888197	Michael	Nkosi	michael.Nkosi@gmail.com	897 Cedar Street	0798801632	287868	David	Nkosi	Grade 8
Dr	7807275046086	Liam	Brown	liam.brown@gmail.com	492 Spruce Street	0786828544	908979	Emily	Brown	Grade 10
Dr	4809209872196	Robert	Smith	robert.smith@gmail.com	757 Cedar Street	0768346836	271520	Alice	Smith	Grade 8
Mrs	4204287904104	Liam	Nkosi	liam.Nkosi@gmail.com	433 Birch Street	0744083306	720053	Jane	Nkosi	Grade 10
Dr	6120643431829	Ethan	Nkosi	ethan.Nkosi@gmail.com	852 Cedar Street	0752005034	761953	Sophia	Nkosi	Grade 10
Ms	4809018674075	Robert	Taylor	robert.taylor@gmail.com	162 Ash Street	0743877000	916977	James	Taylor	Grade 9
Mr	5707284141112	Liam	Anderson	liam.anderson@gmail.com	327 Redwood Street	0796115093	407505	Emily	Anderson	Grade 9
Mr	7009119979038	Ava	Wilson	ava.wilson@gmail.com	911 Birch Street	0764281373	652308	Michael	Wilson	Grade 10
Mrs	5106123937064	Alice	Smith	alice.smith@gmail.com	447 Birch Street	7597640456	312754	Liam	Smith	Grade 10
Dr	4510283636054	Noah	White	noah.white@gmail.com	604 Pine Street	0777393801	801075	Ava	White	Grade 11
Ms	8502022675077	Jane	Anderson	jane.anderson@gmail.com	431 Spruce Street	0758356276	825753	Ethan	Anderson	Grade 8

Dr	4409125431129	John	Nkosi	john.Nkosi@gmail.com	436 Elm Street	0799536039	739637	John	Nkosi	Grade 9
Ms	6305123394192	Emma	Johnson	emma.johnson@gmail.com	715 Birch Street	0763177499	902543	Alice	Johnson	Grade 11
Mr	5912127358130	Michael	White	michael.white@gmail.com	238 Oak Street	0743043868	749847	Robert	White	Grade 9
Mrs	6603287420165	Sophia	Hall	sophia.hall@gmail.com	776 Cedar Street	0778621486	611591	Liam	Hall	Grade 8
Mr	6012217265129	John	Nkosi	john.Nkosi@gmail.com	600 Elm Street	0728680821	899951	Liam	Nkosi	Grade 8
Ms	9602046538016	Ava	Thomas	ava.thomas@gmail.com	446 Redwood Street	0711846135	264399	Robert	Thomas	Grade 8
Dr	7303154126003	Ethan	Taylor	ethan.taylor@gmail.com	404 Oak Street	0799314784	509342	Olivia	Taylor	Grade 8
Ms	2020195031966	Jane	Taylor	jane.taylor@gmail.com	911 Ash Street	0781998910	895067	Alice	Taylor	Grade 9
Mrs	5512102556093	Liam	Wilson	liam.wilson@gmail.com	413 Ash Street	0748585981	828859	Ethan	Wilson	Grade 12
Mr	9309202925129	Liam	Smith	liam.smith@gmail.com	310 Maple Street	0742749180	628670	Ethan	Smith	Grade 10
Mrs	8090359780987	John	Smith	john.smith@gmail.com	140 Maple Street	0775223322	508534	Emma	Smith	Grade 10
Ms	9912089374198	Sophia	Hall	sophia.hall@gmail.com	661 Maple Street	0777667807	773539	John	Hall	Grade 8
Dr	6705260699127	David	Johnson	david.johnson@gmail.com	473 Birch Street	0784780625	617741	Olivia	Johnson	Grade 12
Mr	1008113111123	Ethan	Wilson	ethan.wilson@gmail.com	636 Cedar Street	0737248373	647927	Alice	Wilson	Grade 12
Mr	9122667840153	Noah	Nkosi	noah.Nkosi@gmail.com	353 Oak Street	0763407066	277788	Olivia	Nkosi	Grade 10
Mrs	4803233540087	Ethan	Nkosi	ethan.Nkosi@gmail.com	336 Willow Street	0722944641	382254	Michael	Nkosi	Grade 8
Mr	9501303848045	Ava	Wilson	ava.wilson@gmail.com	487 Spruce Street	0714946067	163680	Emma	Wilson	Grade 8
Mr	6711189204048	Ethan	Nkosi	ethan.Nkosi@gmail.com	382 Elm Street	0738510987	248256	Liam	Nkosi	Grade 10
Mrs	6301209154077	Olivia	Hall	olivia.hall@gmail.com	864 Cedar Street	0717403055	582349	James	Hall	Grade 8
Mrs	7811225151184	Emma	Johnson	emma.johnson@gmail.com	483 Oak Street	0787978403	567866	Olivia	Johnson	Grade 9
Dr	4608148192183	John	Anderson	john.anderson@gmail.com	789 Willow Street	0777769694	977032	James	Anderson	Grade 11
Mr	6204032653193	Michael	Taylor	michael.taylor@gmail.com	701 Spruce Street	0721216884	762563	John	Taylor	Grade 11
Dr	4901218928125	Michael	Thomas	michael.thomas@gmail.com	104 Elm Street	0711790361	922628	Sophia	Thomas	Grade 9
Mrs	5206163043195	David	Taylor	david.taylor@gmail.com	744 Maple Street	0749909129	221375	Ava	Taylor	Grade 11
Ms	9108286769008	Noah	Taylor	noah.taylor@gmail.com	566 Ash Street	0787273081	540787	Sophia	Taylor	Grade 8
Ms	4708037500102	Ethan	Wilson	ethan.wilson@gmail.com	500 Birch Street	0718425293	309568	Jane	Wilson	Grade 9
Mr	4609139102037	Olivia	Smith	olivia.smith@gmail.com	216 Oak Street	0763380717	801734	Olivia	Smith	Grade 12
Ms	6104301288034	Emily	Brown	emily.brown@gmail.com	906 Willow Street	0719919743	868018	James	Brown	Grade 10
Dr	8612145459128	Robert	Anderson	robert.anderson@gmail.com	991 Redwood Street	0764100813	694925	Ava	Anderson	Grade 11
Mr	4108192658003	Olivia	Anderson	olivia.anderson@gmail.com	404 Cedar Street	0714962168	366289	Liam	Anderson	Grade 9

Matome Bernard Mogobu

Student Number: 42689724

Ms	9108286769008	Noah	Taylor	noah.taylor@gmail.com	566 Ash Street	0787273081	540787	Sophia	Taylor	Grade 8
Mr	5707284141112	Liam	Anderson	liam.anderson@gmail.com	327 Redwood Street	0796115093	407505	Emily	Anderson	Grade 9

5. Assessment 1

Programming Languages (3)

Identified the programming languages that you will use to develop your system.

- **C# (.Net backend Framework)**
- **Blazor/HTML (frontend templating)**
- **Entity Framework(ORM for MySQL)**

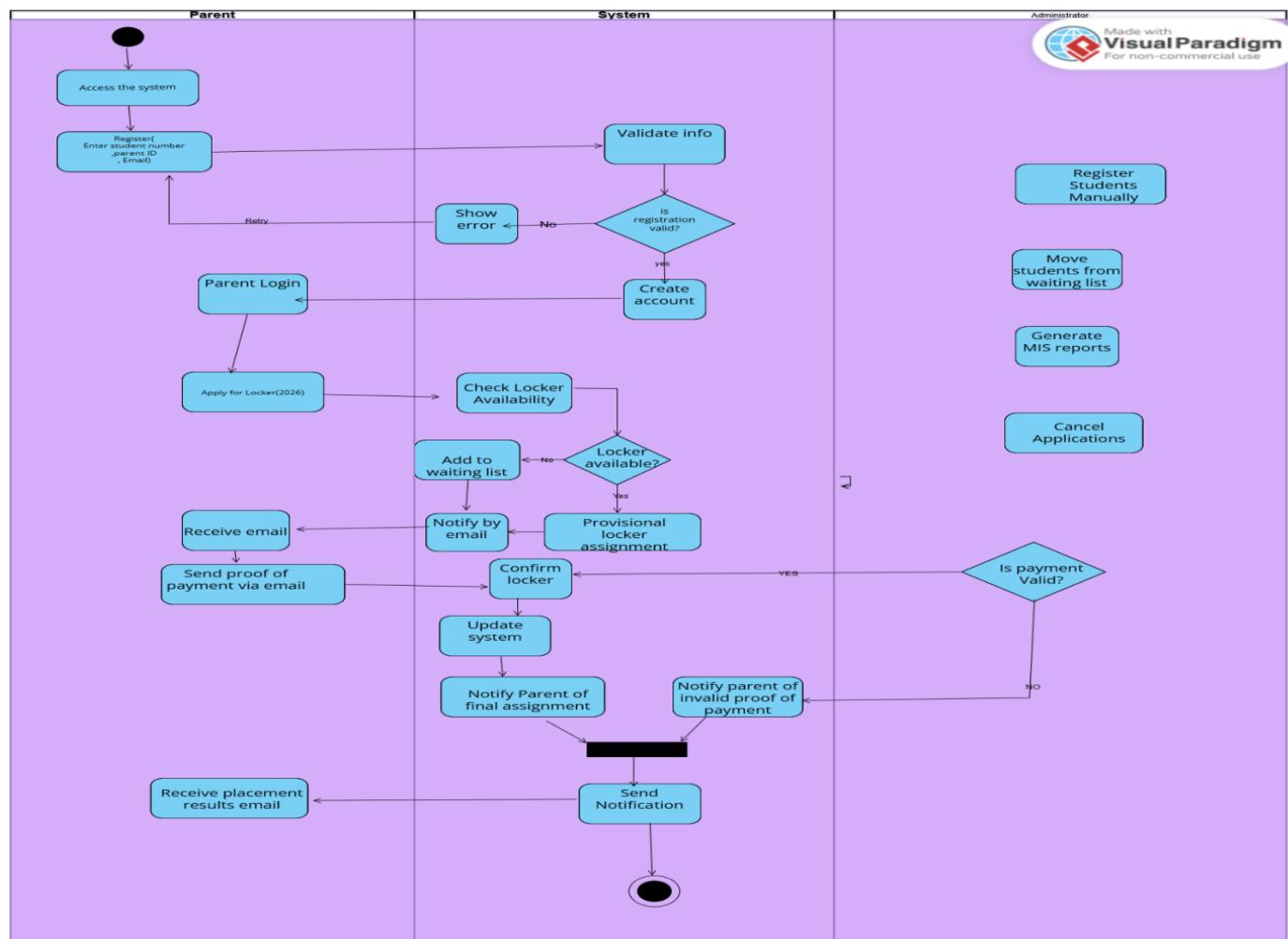
Database Software (1)

Identified the database software that you will use to develop your database.

- **MySQL** (configured in application.properties).

Question 2 [20]

Start by designing your system by completing an Activity Diagram for your system.



Question 3 [20]

Start by designing your database by completing an ERD Diagram for your database.

