Miracle: A Property Management System

CST2550 Group Coursework Report
Team Leader: Jebinaxar Michael

1. Introduction

This project, titled "Miracle: A Property Management System," is a console-based application designed to handle property and tenant management operations such as adding, assigning, and vacating tenants in properties using a SQL Server backend. The goal was to simulate real-world operations of a property management system while practicing software development principles, data structure design, and database integration.
This report outlines the system design, choice of data structures, algorithm analysis, testing approach, and reflections from the project development process.

2. Design

2.1 Data Structures and Justification

The application utilizes `List<T>` as the primary data structure to manage Property and Tenant records in memory during runtime. The decision was made to use lists because of their simplicity, dynamic sizing, and efficient linear access for small to medium datasets typical in educational simulations.

2.2 Algorithm Design and Pseudocode

Below is sample pseudocode for adding a new tenant to the database:

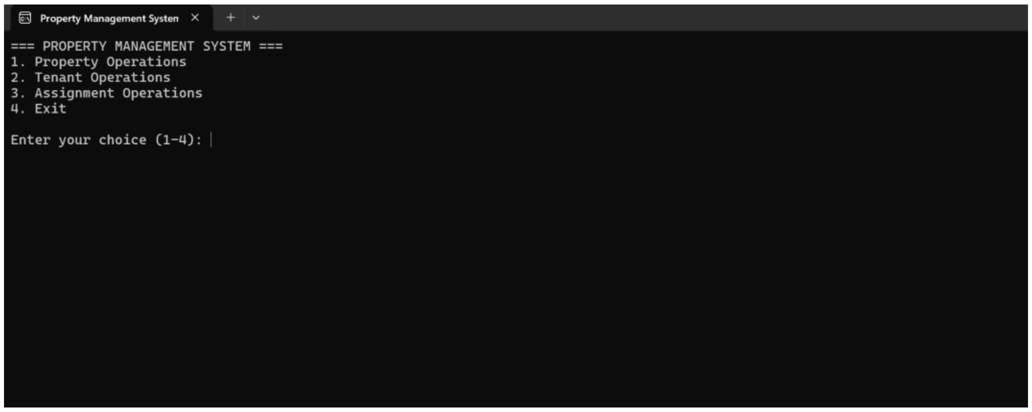
```
FUNCTION AddTenant(Tenant)
  CONNECT to database
  IF TenantID already exists THEN
    DISPLAY "Tenant already exists"
    RETURN
  ELSE
    INSERT tenant into Tenants table
    DISPLAY "Tenant added successfully"
END FUNCTION
```

Time-complexity analysis:

- Searching: O(n) – for checking duplicate IDs in list or DB.
- Insertion: O(1) – direct insert using SQL command.

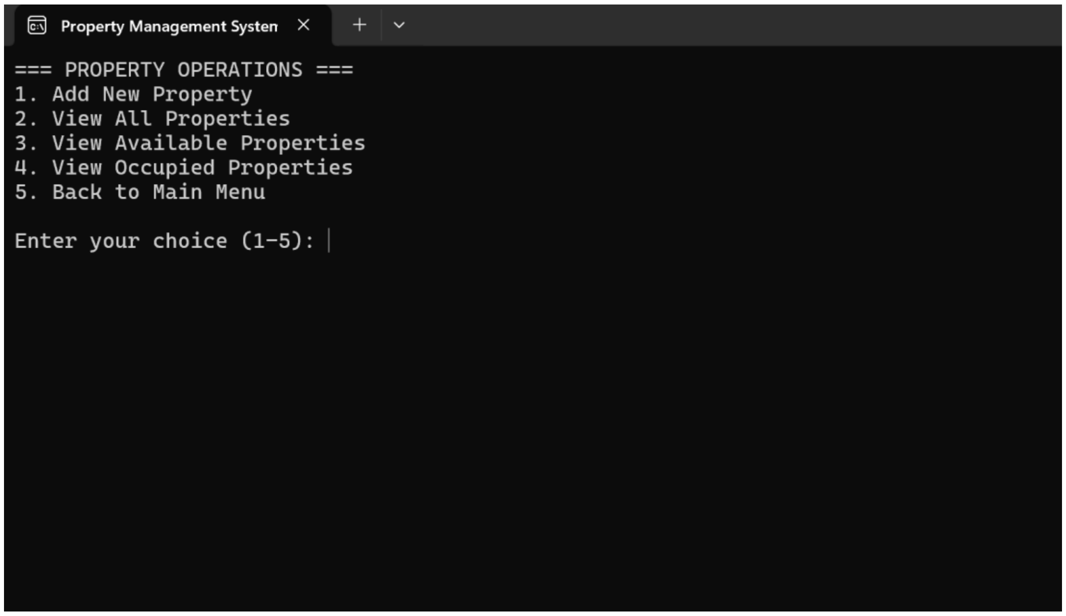
Demonstration Screenshots

Main Menu – Entry point of the system with main operations



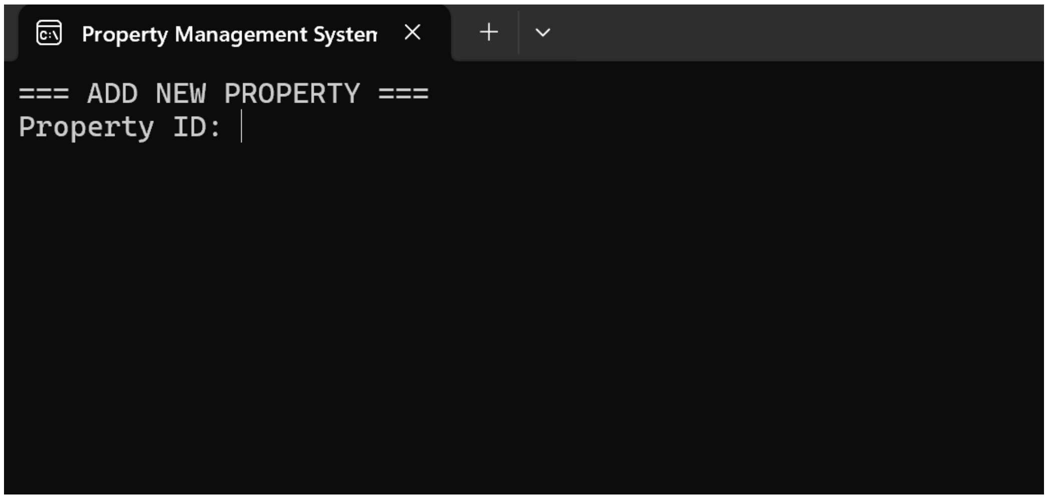
```
=== PROPERTY MANAGEMENT SYSTEM ===
1. Property Operations
2. Tenant Operations
3. Assignment Operations
4. Exit

Enter your choice (1-4):
```

Property Operations – Options to manage properties



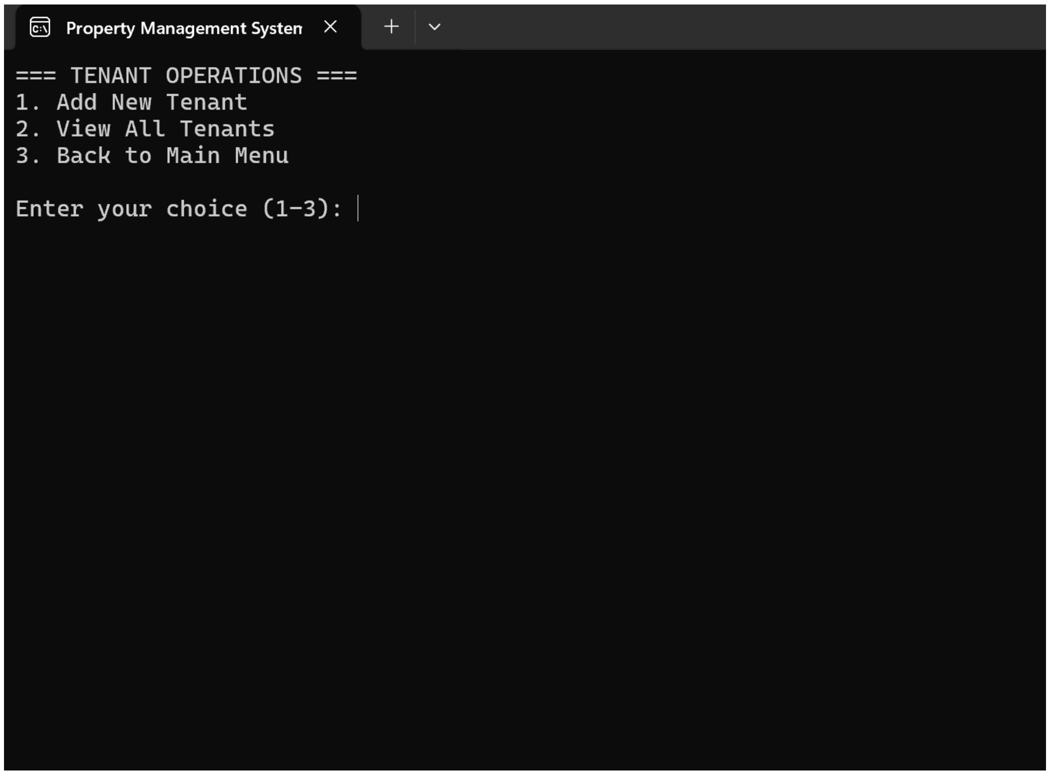
```
=== PROPERTY OPERATIONS ===
1. Add New Property
2. View All Properties
3. View Available Properties
4. View Occupied Properties
5. Back to Main Menu

Enter your choice (1-5):
```

Add New Property – User inputs property details



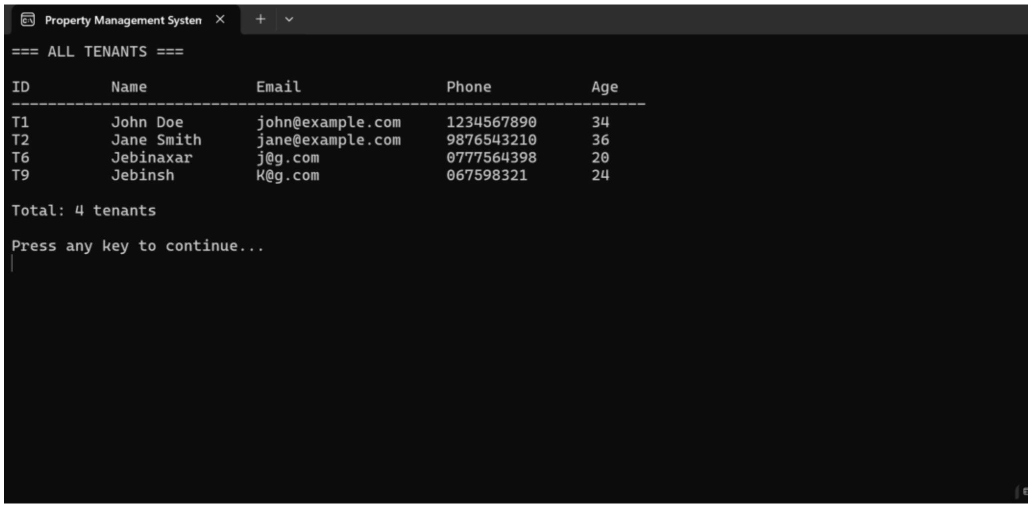
```
=== ADD NEW PROPERTY ===
Property ID:
```

Tenant Operations – Menu to add or view tenants



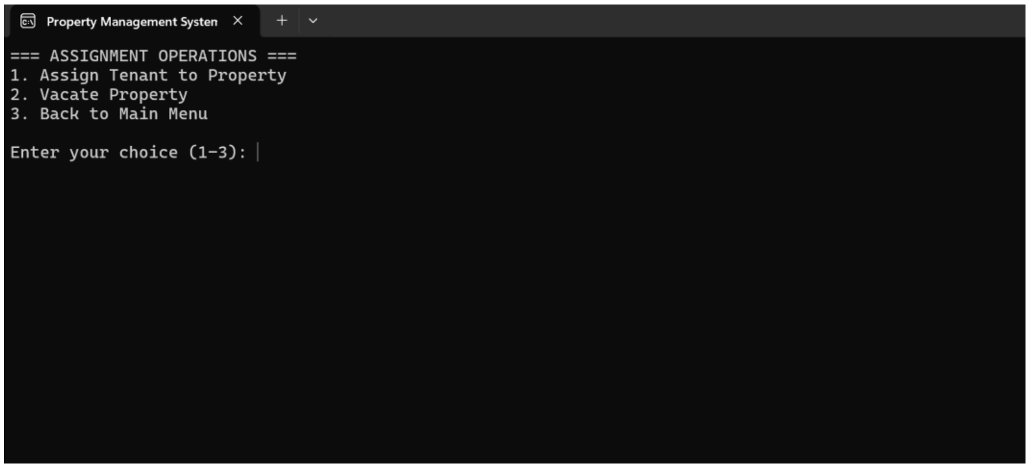
```
=== TENANT OPERATIONS ===
1. Add New Tenant
2. View All Tenants
3. Back to Main Menu

Enter your choice (1-3):
```

All Tenants – Display of all registered tenants



```
=== ALL TENANTS ===

ID          Name            Email                 Phone           Age
-----------------------------------------------------------------------
T1          John Doe        john@example.com      1234567890      34
T2          Jane Smith      jane@example.com      9876543210      36
T6          Jebinaxar       j@g.com               0777564398      20
T9          Jebinsh         K@g.com               067598321       24

Total: 4 tenants

Press any key to continue...
```

Assignment Operations – Assign or vacate tenants



```
=== ASSIGNMENT OPERATIONS ===
1. Assign Tenant to Property
2. Vacate Property
3. Back to Main Menu

Enter your choice (1-3):
```

Occupied Properties – View of currently occupied properties

4. Testing

We used MSTest to perform unit tests on core methods like AddTenant, AddProperty, AssignTenantToProperty, and VacateProperty. Testing was done in isolation, focusing on edge cases like duplicate entries and null values. But, Unfortunately Not the whole code is tested because of the unexpected error and time shortage(Team leader is doing the test not the tester)

6. Conclusion

The Miracle system is a System made in two days, after changing the whole plan. The initial Plan was much Prettier , good to use, a wide range Property management system. Because of the shortage of people and unexpected crisis the team failed in executing the plan. The current document, code, test file etc.. everything is done by the team leader in the final week! Even after my this week, none of my team members except me, were not able to learn or understand whats git, push pull nothing.
As team leader, I failed in making the team progress.

6. References

Microsoft Docs. (n.d.). SqlConnection Class. https://learn.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection
Microsoft Docs. (n.d.). MSTest Framework. https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-mstest