**DEVELOPMENT OF AN CONDOMINIUM MANAGEMENT SYSTEM**

**Developed by**

Shohel Pramanik

**0121150110050**

Report submitted to Lincoln University College Training Centre, Main Campus Wisma Lincoln, No. 12, 14, 16 & 18, Jalan SS 6/12, 47301 Petaling Jaya, Selangor Darul Ehsan, Malaysia.

December, 2017

**DECLARATION**

I am **Shohel Pramanik**, with student ID: **0121150110050,** hereby declare that the project was done by me and all the content therein are of my own fabrication.

----------------------------  
**Signature & Date**

**CERTIFICATION**

This report was submitted to Lincoln University College Training Centre and it has been accepted as a fulfilment of the requirement for the level advancement in the Software Development/ Network and Server training. The members of the evaluation committee were as follows:

1. **Deputy Registrar/Head of Training Centre**

Name: **Murtadho M. Alao**

Signature and Date:

1. **Software Development Lecturer**

Name: **Dr. Munir Kolapo Yahya-Imam**

Signature and Date:

1. **Network and Server Lecturer**

Name: **Aashu Jha**

Signature and Date:

**ABSTRACT**

Investment in Condominium maintenance is huge all over the world. Maintenance management involves obtaining maximum benefit from the investment made on the maintenance activities. Maintenance in buildings in each country is on the increase regardless of size, type, location, and ownership. The current maintenance management procedures are, however, condition and reactive based. The weaknesses in the current procedures are the primary problems because they do not explicitly link maintenance needs with building performance with respect to the building users. The building users measure the performance of a building with various criteria. The condition of a building is just one of those criteria. This research establishes the need for a building maintenance management system that is based on the concept of value in the effort to lead towards the optimisation of building maintenance programmes. Maintenance management that is value based allows users to be proactively put at the centre of maintenance management during decision-making processes and takes into consideration both the objective and subjective requirements of users. However, this study was based on Condominium Management System to make our work is more easy and comfortable.

**INTRODUCTION**

The purpose of this paper is to look at the maintenance and maintenance management of Condominium in developed countries. Residential condominiums are frequently constructed as apartment buildings, but there has been an increase in the number of "detached condominiums", which look like single-family homes but in which the yards, building exteriors, and streets are jointly owned and jointly maintained by a community association. Unlike apartments, which are leased by their tenants, condominium units are owned outright. Additionally, the owners of the individual units also collectively own the common areas of the property, such as hallways, walkways, laundry rooms, etc.; as well as common utilities and amenities, such as the HVAC system, elevators and so on. To control everything like security, maintenance, finance, stuff, owner, tenants record are need a system which can make everything easy. A Condominium management system that is value based provides a clear definition of the goals of the maintenance programmes and policies as well as establishing the problems the maintenance is expected to solve.

**METHODOLOGY**

Select Project Topic

Project Module

Plan of the Project

Hardware Tools

Software Tools

Project Requirements

Develop the Project

Test

Presentation

Documentation

**Figure 1: Methodology Framework**

**Project Topic Selection**

For managing a Condominium activities like Transaction, Apartment record, stuff record with their activities, maintenance, security, parking and so on I am proposing a software application that take care all the important things. Implementation of this Software will help the Users in several ways.

**System Module**

**HR Module:**

* Log in/Logout
* Landlord Record View
* Employee Record View
* Condominium Map
* Apartment Details view
* Management Rules view (HR Activities, Finance Activities, Admin Activitie, Security Activities, Maintenance activities and Cleaner Activities)
* Update and view Hiring for vacancy
* Tenant Record View

**Administrator Module:**

* Login/Logout
* Create/Update/View Landlord Registration
* Create/Update/View Employe Registration
* Upload/Update/Preview Staff Activities
* Create/Update Bloack/Level/Flat Information

**Finance Module:**

* Login/Logout
* Create/View Payment Record
* Create/View Expense Record
* Create/View Salary Record
* Create/View Balance Sheet

**Stuff Module:**

* Login/Logout
* Staff Check in/ Checkout
* Staff Check in/Checkout Record View
* Security Guard Checking(Enter/Exit) Record Create/View for Staff
* Security Guard Checking(Enter/Exit) Record Create/View For Landlord
* Security Guard Checking(Enter/Exit) Record Create/View For Tenants
* Security Guard Checking(Enter/Exit) Record Create/View For Visitor

**Flat Owner/Landlord Module:**

* Login/Logout
* Create/Update/Delete Tenant Record
* Search/Over View Flat/Apartment Status

**Planning**

* Requirement gathering and analysis.
* Design.
* Implementation or coding.
* Testing.
* Deployment.
* Maintenance.

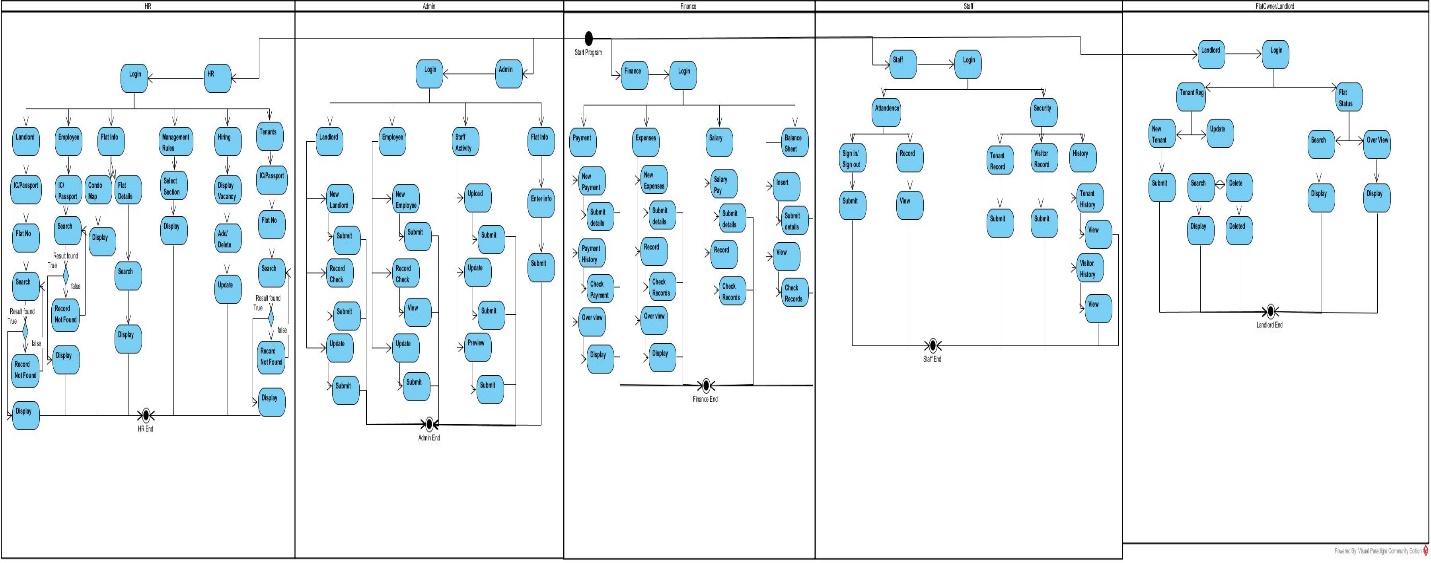
**Hardware tools**

* CPU : Intel(R) Core(TM)2 Duo
* RAM : 4.00GB
* Monitor : 13-inch Color Monitor
* Keyboard : Full-size backlit keyboard with 78 (U.S.) Keys
* Mouse : Apple Mouse

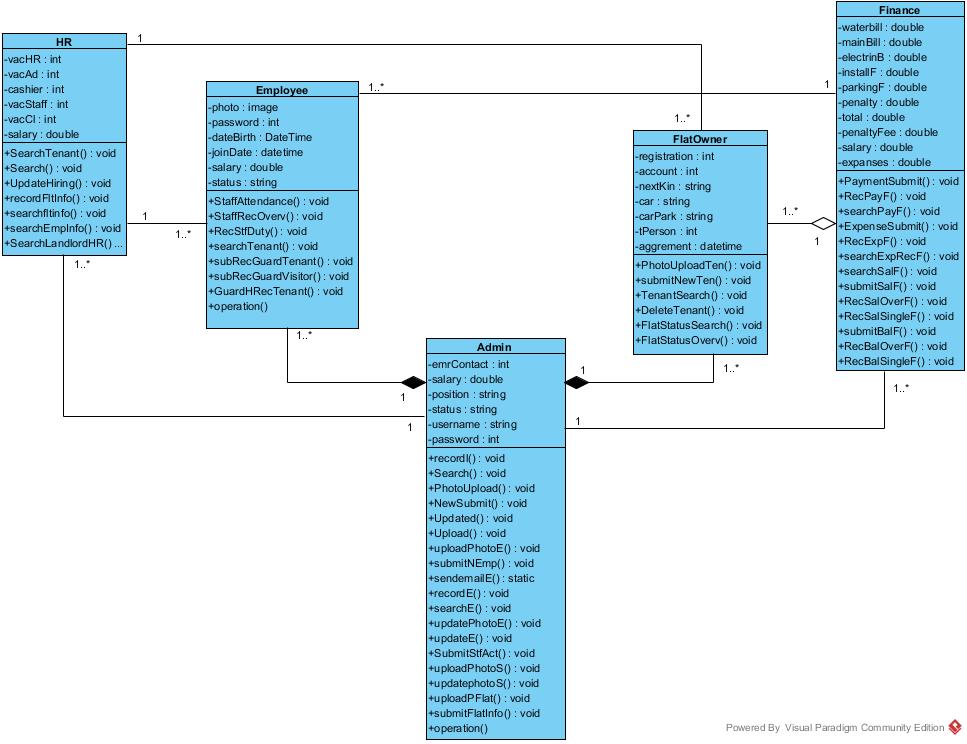
**Software tools**

* Operating System : Windows 7 Professional
* Designing : Windows Form
* Technology : Visual Studio 2015
* Front End : C#(.Net)
* Server Side Scripting : SQL Client
* Back End : SQL Server Management Studio

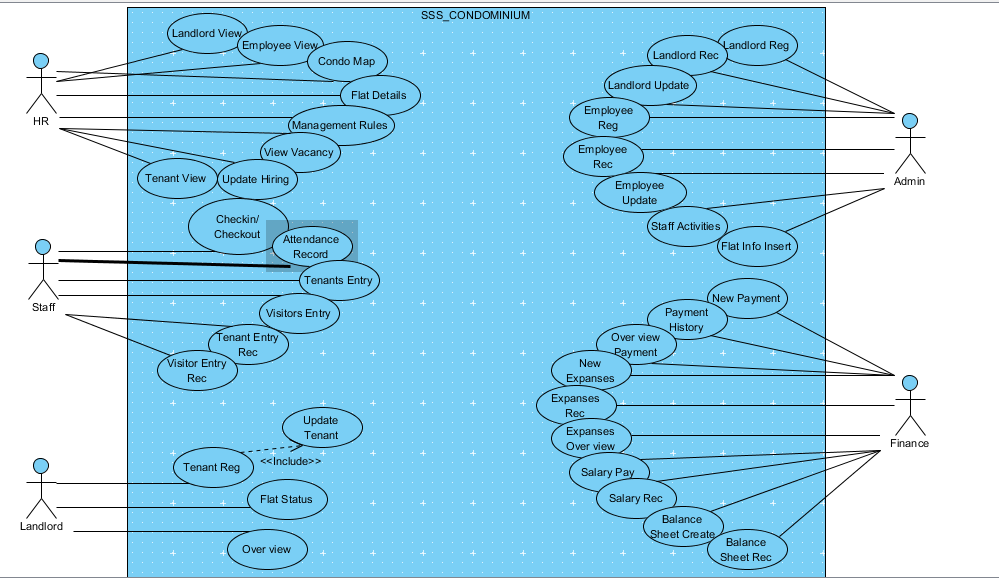
**DESIGN AND IMPLEMENTATION**

****

**Figure 2: Activity Diagram**

****

**Figure 3: Class Diagram**

****

**Figure 4: Use Case Diagram**

Use Case: User Login

Primary Actors:

 HR

 Admin

 Finance

 Staff

 FlatOwner

Precondition: MSSQL Sever active.

Basic flows:

1. Enter username

2. Enter password

3. Login

Alternate flows: Wrong Username and Password are not granted. It will redirect same login page.



Figure 5: Login Sequence

Use Case: Employee Registration

Primary Actors: Admin

Precondition: MSSQL Sever active.

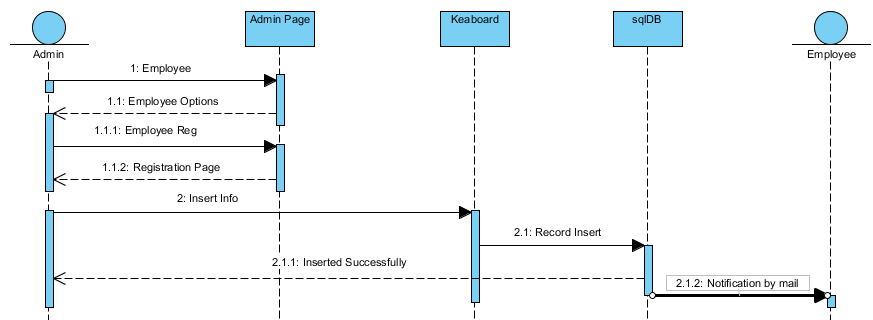
Basic flows:

1. Enter Employee Full Information

2. Submit To Create Record

3. Email Notification to Employee

Alternate flows: All Required Field Have to be fillup.

****

**Figure 6: Employee Registration Sequence**

Use Case: Employee Information Update

Primary Actors: Admin

Precondition: MSSQL Sever active.

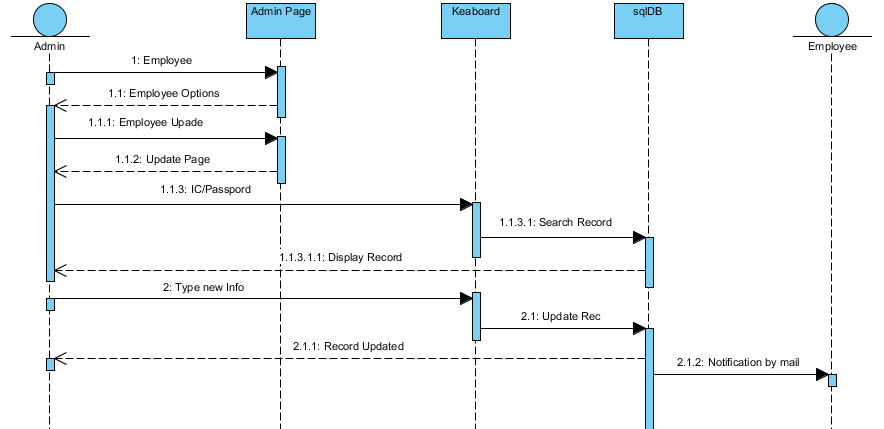
Basic flows:

1. Enter new information

2. Submit To update Record

3. Email Notification to Employee

Alternate flows: Any field can be update.

****

**Figure 7: Employee Update Sequence**

Use Case: Employee Record View

Primary Actors: Admin

Precondition: MSSQL Sever active.

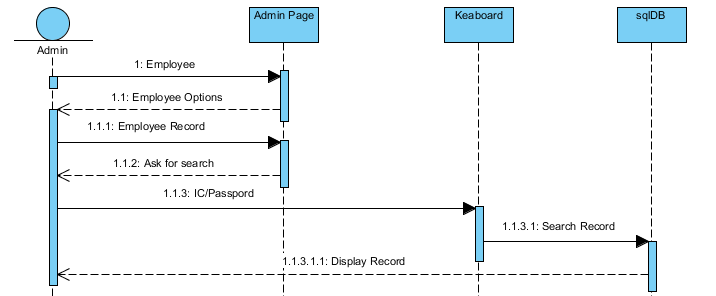
Basic flows:

1. Employ I’d/IC

2. Search

3. Display Record

Alternate flows: Incorrect I’D/IC cannot be access.

****

**Figure 8: Employee Record Sequence**

Use Case: Flat Information Add

Primary Actors: Admin

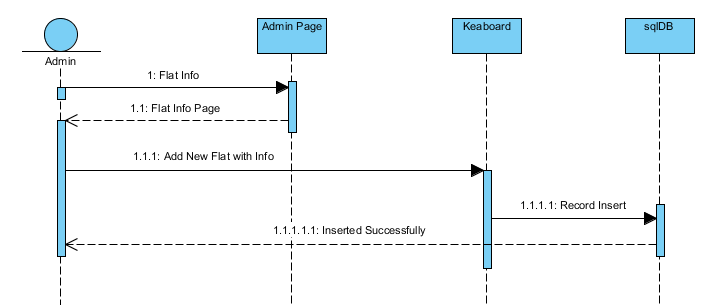
Precondition: MSSQL Sever active.

Basic flows:

1. Enter Flat Full Information

2. Submit To Create Record

Alternate flows: All Required Field Have to be fillup.

****

**Figure 9: Flat Info Add Sequence**

Use Case: Landlord Registration

Primary Actors: Admin

Precondition: MSSQL Sever active.

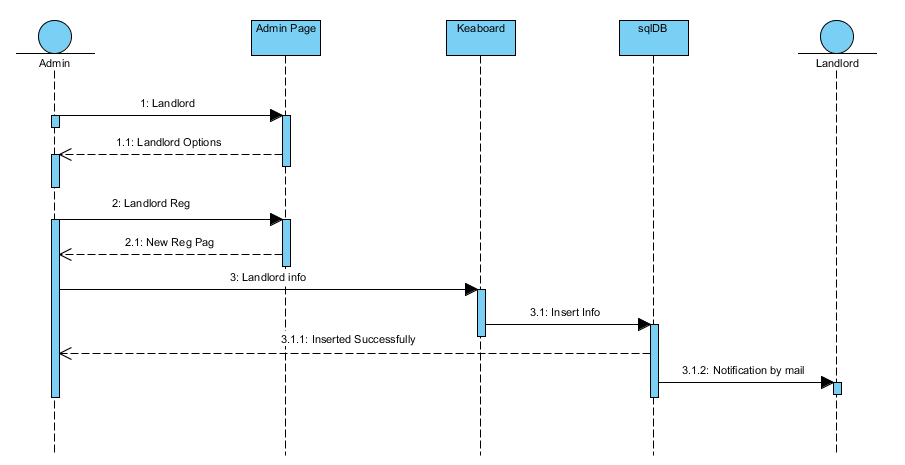
Basic flows:

1. Enter Landlord Full Information

2. Submit To Create Record

3. Email Notification to Landlord

Alternate flows: All Required Field Have to be fillup.

****

**Figure 10: Landlord Registration Sequence**

Use Case: Employee Record Update

Primary Actors: Admin

Precondition: MSSQL Sever active.

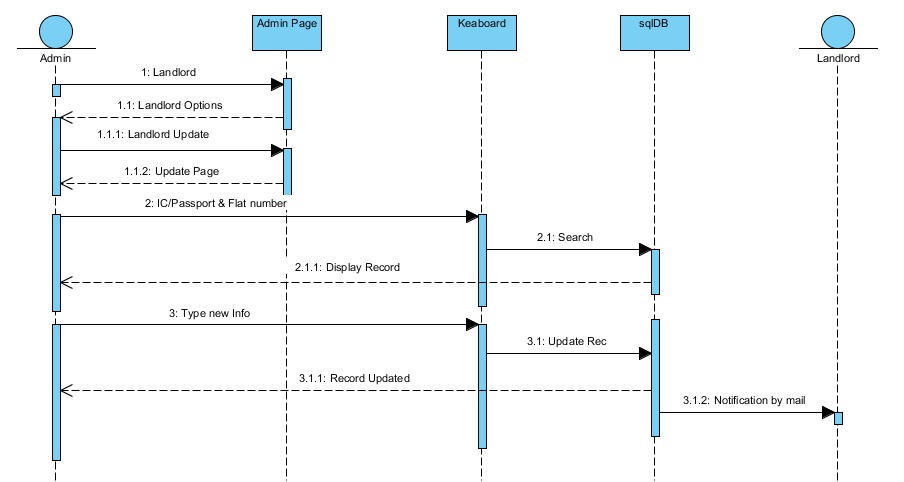
Basic flows:

1. Enter Landlord IC and Landlord Flat Number

2. Submit To update Record

3. Email Notification to Employee

Alternate flows: Incorrect IC or Flat number cannot be accepted

****

**Figure 11: Landlord Registration Update Sequence**

Use Case: Landlord Record view

Primary Actors: Admin

Precondition: MSSQL Sever active.

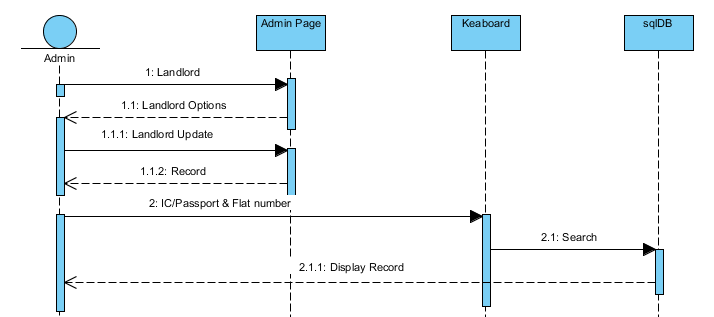
Basic flows:

1. Enter Landlord IC and Flat number

2. Search the record

3. Display record

Alternate flows: Incorrect IC or Flat number cannot be accepted

****

**Figure 12: Landlord Registration Record Sequence**

Use Case: Management Rules/ Staff Activities

Primary Actors: Admin

Precondition: MSSQL Sever active.

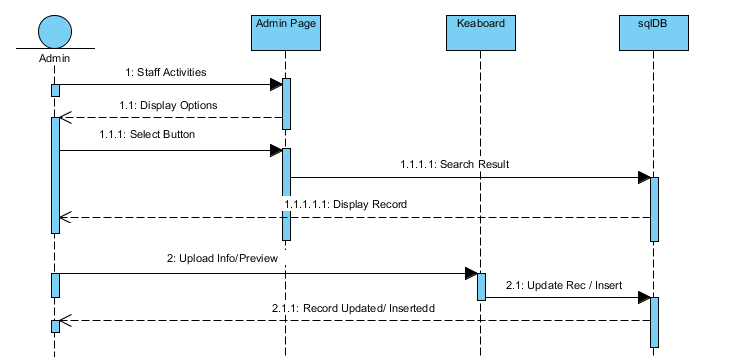
Basic flows:

1. Upload file

2. Update file

3. Preview file

Alternate flows: At first need to check the Radiobutton.

****

**Figure 13: Staff Activities Sequence**

Use Case: Balance Sheet Create

Primary Actors: Cashier

Precondition: MSSQL Sever active.

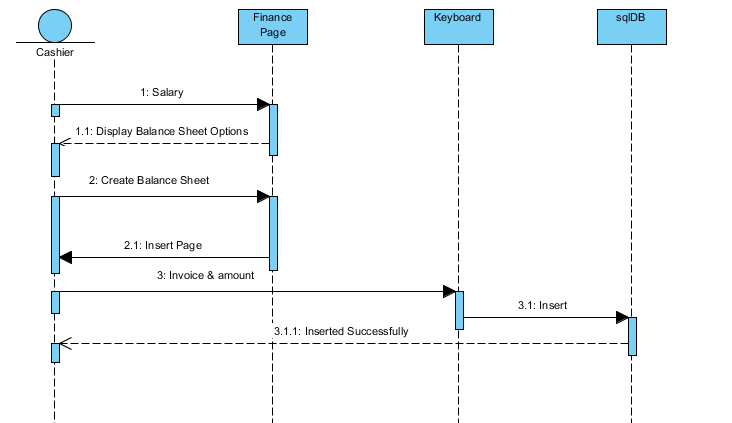
Basic flows:

1. Enter Payment Invoice number and total

2. Enter Expanses Invoice number and Total

3. Enter Salary slip/ Pay slip number and Total

Alternate flows: Cashier IC is requred To Create Balance sheet.

****

**Figure 14: Balance Sheet Create Sequence**

Use Case: Balance Sheet Record view

Primary Actors: Cashier

Precondition: MSSQL Sever active.

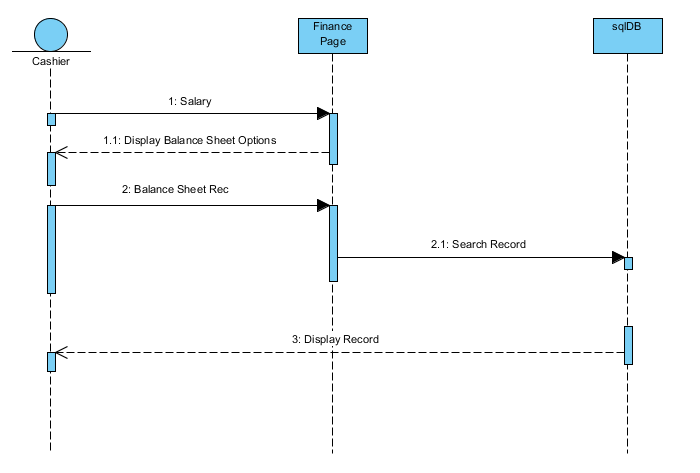
Basic flows:

1. Enter Casier IC

2. Enter Date

3. Display record

Alternate flows: Incorrect IC cannot be accepted

****

**Figure 15: Balance Sheet Record Sequence**

Use Case: New Expanses record create

Primary Actors: Cashier

Precondition: MSSQL Sever active.

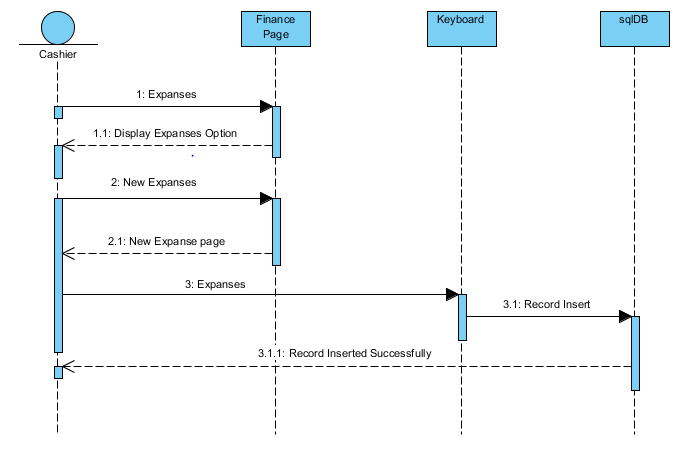
Basic flows:

1. Enter Employee I’d or IC

2. Enter Details

3. Submit

Alternate flows: Incorrect IC cannot be accepted

****

**Figure 16: New Expanses Sequence**

Use Case: Expanses record view

Primary Actors: Cashier

Precondition: MSSQL Sever active.

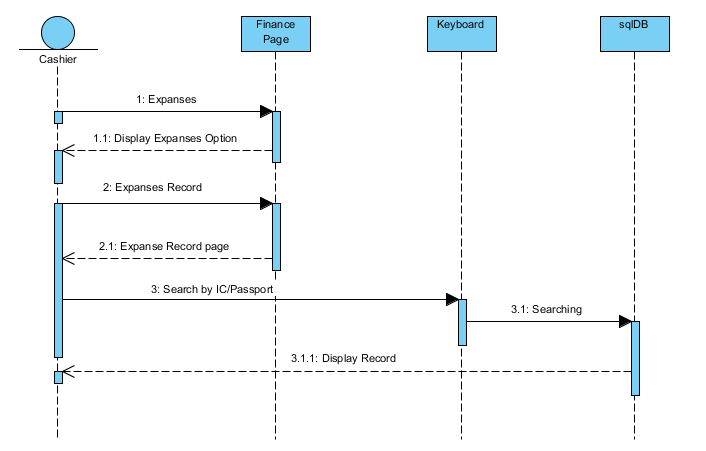
Basic flows:

1. Enter date /IC number

2. Search

3. Display

Alternate flows: Without date cannot be accepted

****

**Figure 17: Expanses Record View Sequence**

Use Case: Expanses record Overview

Primary Actors: Cashier

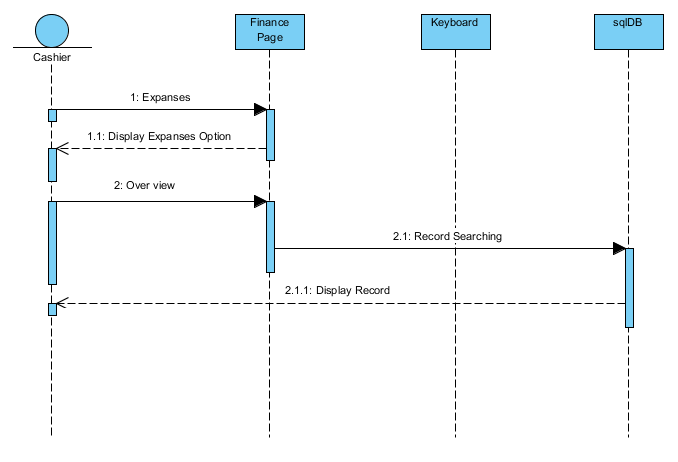
Precondition: MSSQL Sever active.

Basic flows:

1. Enter Cashier I’d or IC

2. Submit

3. Display

****

**Figure 18: Expanses Record Overview Sequence**

Use Case: New Payment record create

Primary Actors: Cashier

Precondition: MSSQL Sever active.

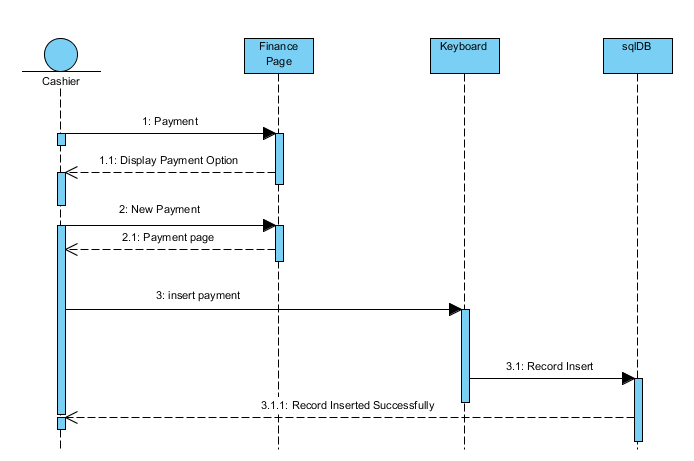
Basic flows:

1. Enter IC and Flat number

2. Enter Details

3. Submit

Alternate flows: IC and Flat number required.

****

**Figure 19: New Payment Sequence**

Use Case: Payment Record view

Primary Actors: Cashier

Precondition: MSSQL Sever active.

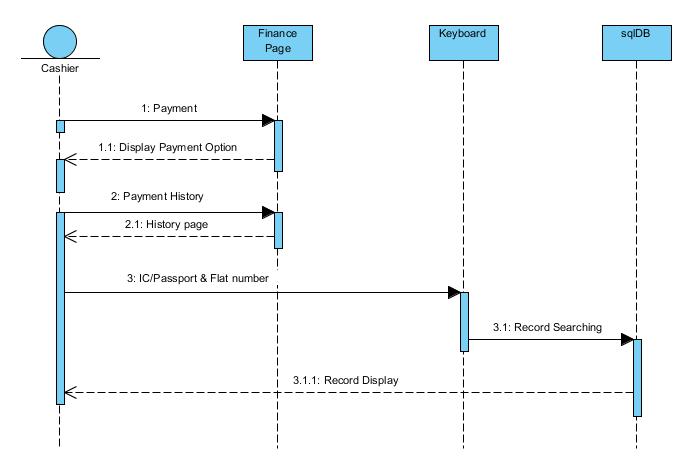
Basic flows:

1. Enter IC and Flat number

2. Enter date

3. Submit

Alternate flows: Incorrect IC and flat number cannot be accepted

****

**Figure 20: Payment Record View Sequence**

Use Case: Payment Record Overview

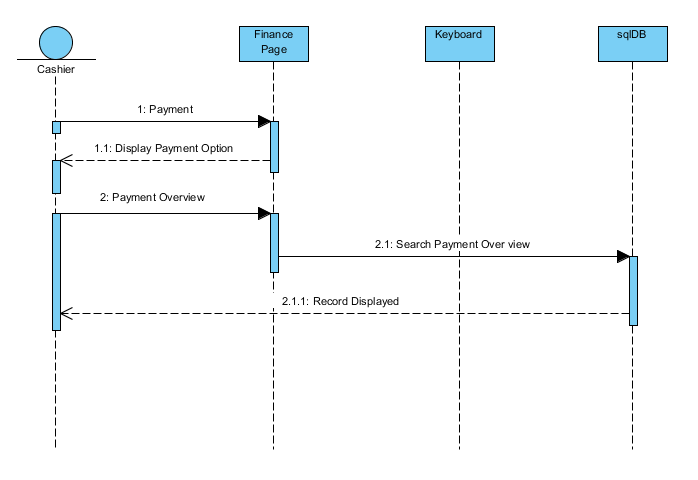
Primary Actors: Cashier

Precondition: MSSQL Sever active.

Basic flows:

1. Search for over view

2. Display

****

**Figure 21: Payment Record Overview Sequence**

Use Case: New Salary record create

Primary Actors: Cashier

Precondition: MSSQL Sever active.

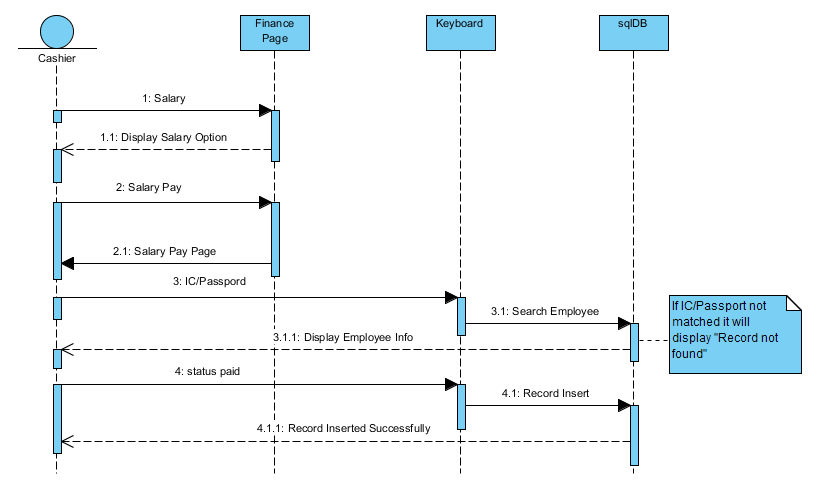
Basic flows:

1. Enter IC /Employee I’d

2. Enter Details

3. Submit

Alternate flows: IC/I’d is required.

****

**Figure 22: New Salary Sequence**

Use Case: Salary record view

Primary Actors: Cashier

Precondition: MSSQL Sever active.

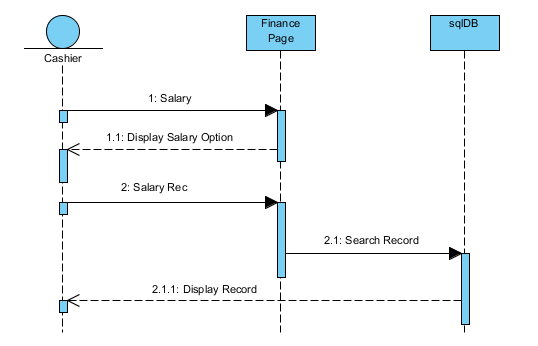
Basic flows:

1. Enter IC or Employee I’d

2. Enter date

3. Submit

Alternate flows: IC is required.

****

**Figure 23: Salary Record View Sequence**

Use Case: New Flat Add

Primary Actors: Flat Owner

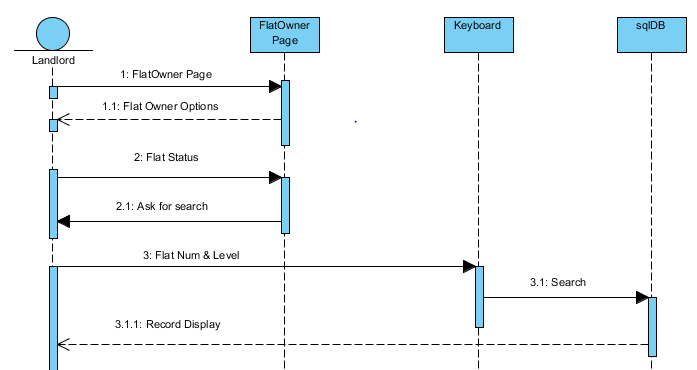
Precondition: MSSQL Sever active.

Basic flows:

1. Enter Flat details

3. Submit

Alternate flows: All fields are required.

****

**Figure 24: Flat Status Search Sequence**

Use Case: Flat Status view

Primary Actors: FlatOwner

Precondition: MSSQL Sever active.

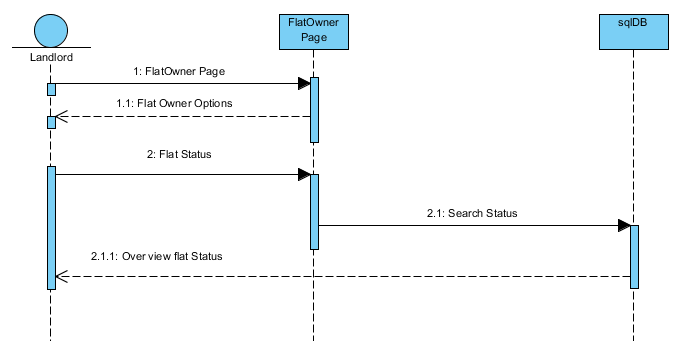
Basic flows:

1. Enter Flat number

2. Search

3. Display

Alternate flows: Flat number must required.

****

**Figure 25: Flat Status Overview Sequence**

Use Case: New Tenant record create

Primary Actors: Flat Owner

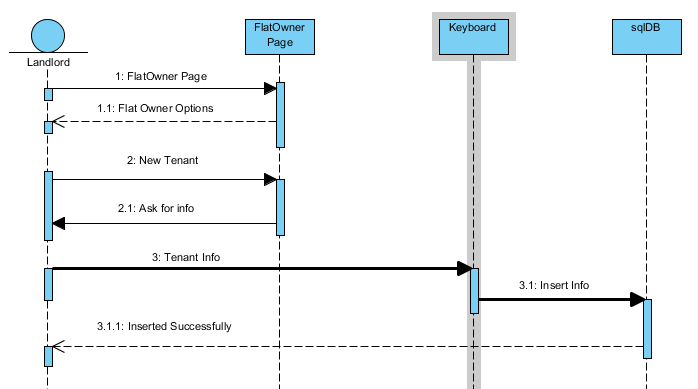
Precondition: MSSQL Sever active.

Basic flows:

1. Enter Tenant Information

2. Submit

Alternate flows: All fields are required to fill.

****

**Figure 26: Tenant Registration Sequence**

Use Case: New Tenant record update

Primary Actors: Flat Owner

Precondition: MSSQL Sever active.

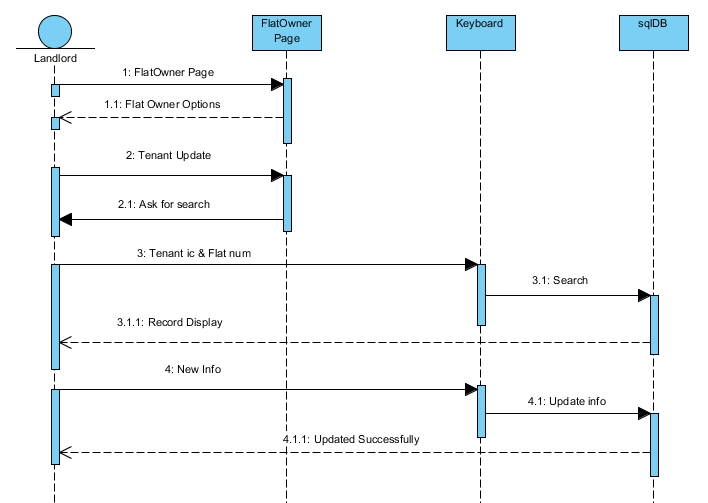
Basic flows:

1. Enter Tenant IC and Flat number

2. Search

3. Enter new info

Alternate flows: IC and flat number are required to fill.

****

**Figure 27: Tenant Registration Update Sequence**

Use Case: New Tenant record update

Primary Actors: HR

Precondition: MSSQL Sever active.

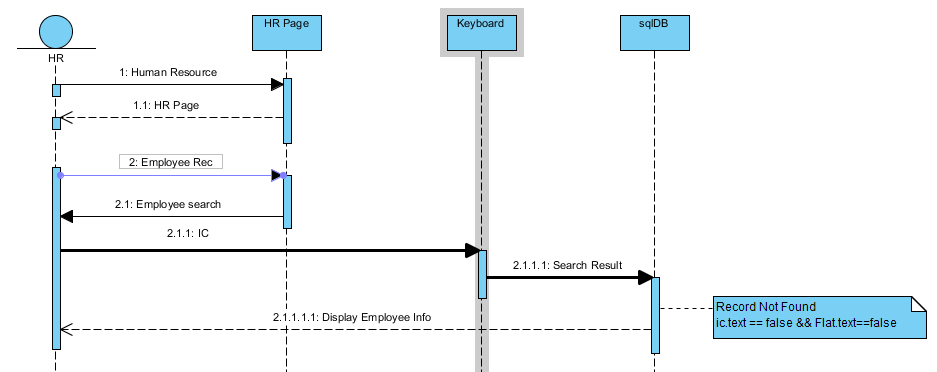
Basic flows:

1. Enter Tenant IC and Flat number

2. Search

3. Display

Alternate flows: IC and flat number are required to fill.

****

**Figure 28: Tenant Record View Sequence**

Use Case: Condominium Overview

Primary Actors: HR

Precondition: MSSQL Sever active.

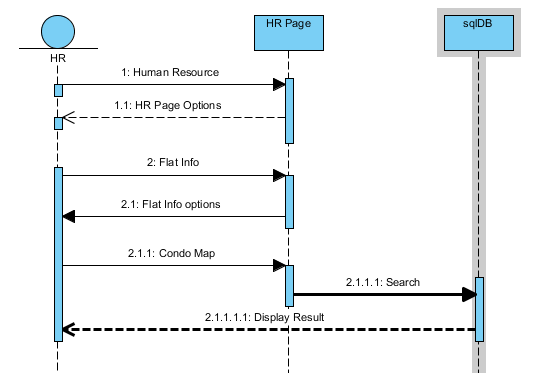
Basic flows:

1. Enter Flat number

2. Search

3. Display

Alternate flows: Flat number are required to fill.

****

**Figure 29: Condominium Overview Sequence**

Use Case: Flat Details

Primary Actors: HR

Precondition: MSSQL Sever active.

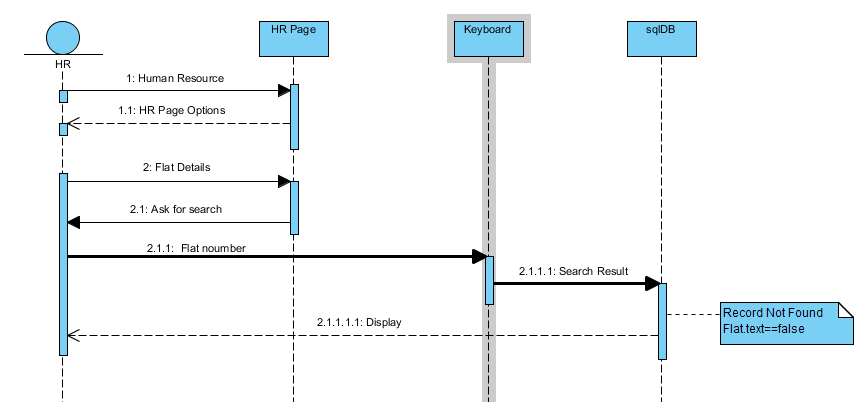
Basic flows:

1. Enter Flat number

2. Search

3. Display

Alternate flows: Flat number are required to fill.

****

**Figure 30: Flat Details Sequence**

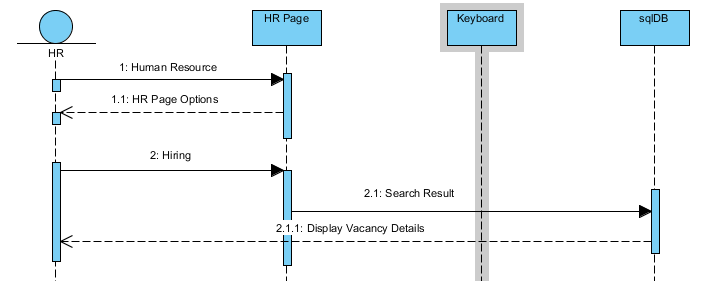
Use Case: Vacancy view

Primary Actors: HR

Precondition: MSSQL Sever active.

Basic flows:

1. Search
2. Display

****

**Figure 31: View Vacancy Sequence**

Use Case: Update Vacancy

Primary Actors: HR

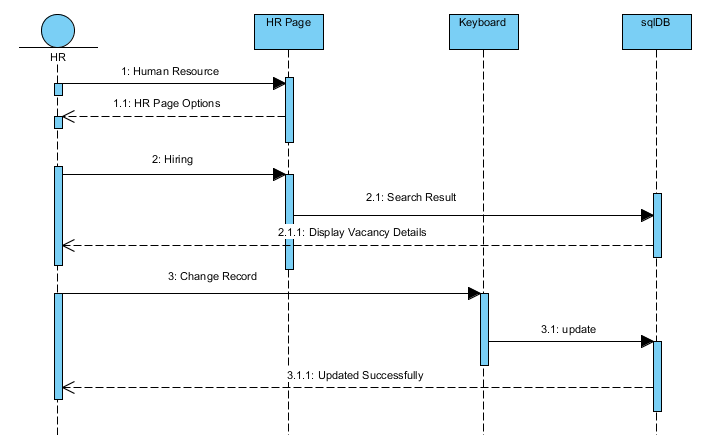
Precondition: MSSQL Sever active.

Basic flows:

1. Search Vacancy

2. Enter new vacancy

3. Update record

****

**Figure 32: Update Vacancy Sequence**

Use Case: Landlord Record View

Primary Actors: HR

Precondition: MSSQL Sever active.

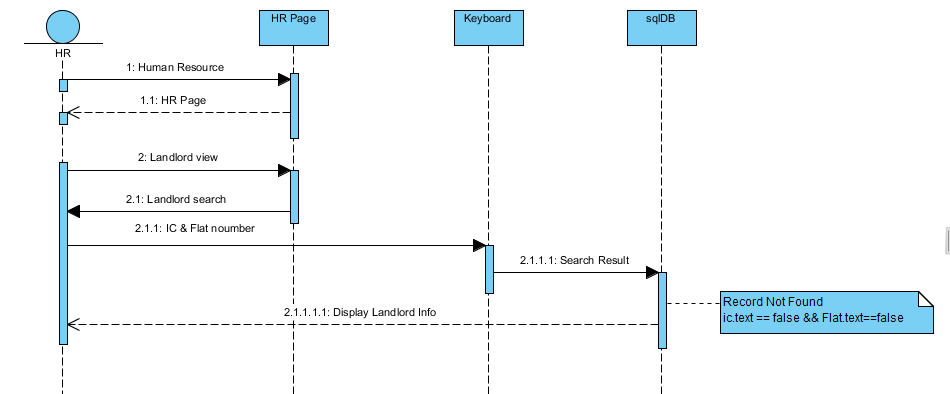
Basic flows:

1. Enter IC and Flat number

2. Search

3. Display

Alternate flows: IC and Flat number are required to fill.

****

**Figure 33: Landlord Record View Sequence**

Use Case: Management Rules

Primary Actors: HR

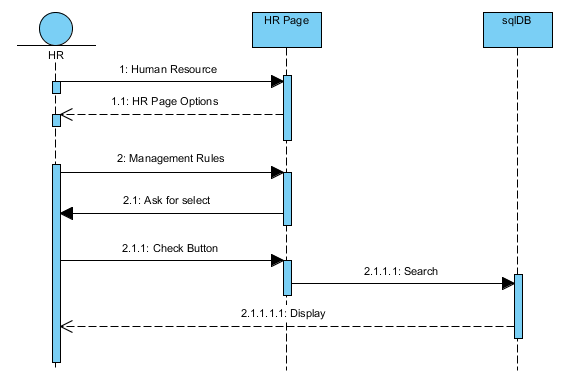
Precondition: MSSQL Sever active.

Basic flows:

1. Select Employee sector

2. Preview file

3. Display

****

**Figure 34: Management Rules Sequence**

Use Case: Tenant Record View

Primary Actors: HR

Precondition: MSSQL Sever active.

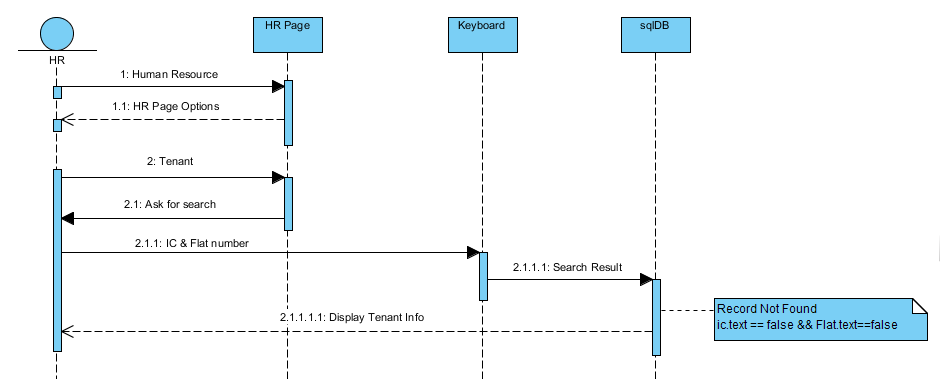
Basic flows:

1. Enter IC and Flat number

2. Search

3. Display

Alternate flows: IC and Flat number are required to fill.

****

**Figure 35: Tenant Record View Sequence**

Use Case: Staff Check in / Check out

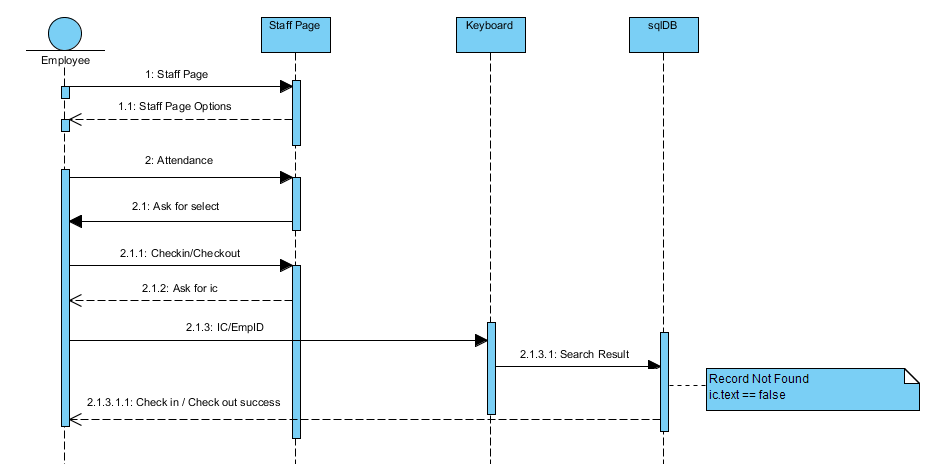
Primary Actors: Staff

Precondition: MSSQL Sever active.

Basic flows:

1. Enter IC

2. Sign in or Sign out

****

**Figure 36: Check in / Checkout Sequence**

Use Case: Attendance Record

Primary Actors: Staff

Precondition: MSSQL Sever active.

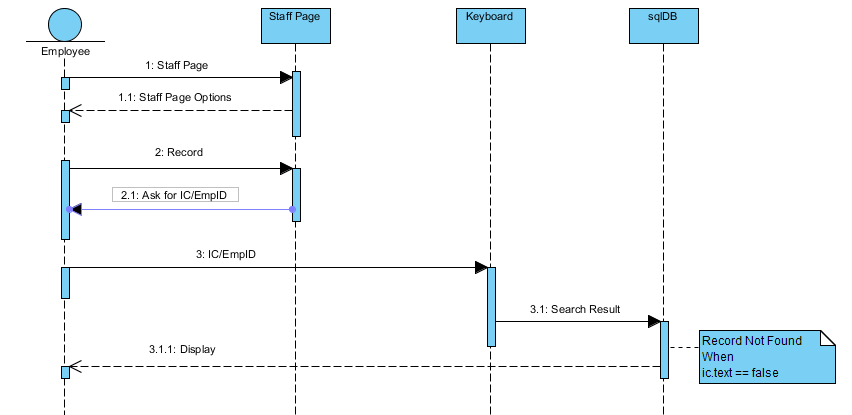
Basic flows:

1. Enter IC

2. Sign in or Sign out

3. Display

Alternate flows: IC and date are required to fill.

****

**Figure 37: Check in / Checkout Record View Sequence**

Use Case: Tenant Entry/exit

Primary Actors: Security

Precondition: MSSQL Sever active.

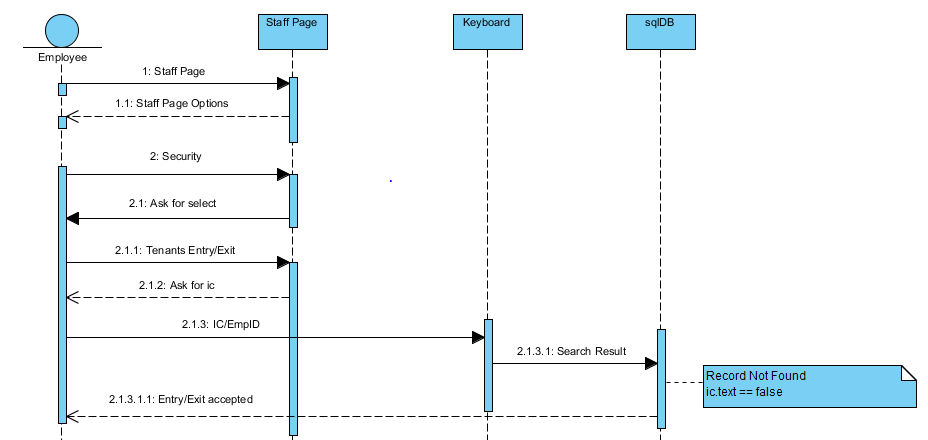
Basic flows:

1. Enter IC

2. Check identity

3. Exit/Entry

Alternate flows: IC is required to fill.

****

**Figure 38: Tenant Entry Sequence**

Use Case: Tenant Entry/exit History

Primary Actors: Security

Precondition: MSSQL Sever active.

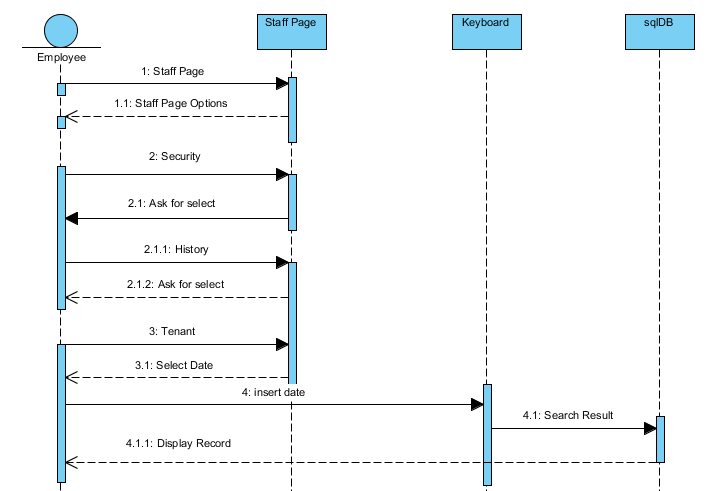
Basic flows:

1. Enter IC / Date

2. Search

3. Display Record

Alternate flows: IC/Date is required to fill.

****

**Figure 39: Tenant Entry History Sequence**

Use Case: Visitor Entry/exit History

Primary Actors: Security

Precondition: MSSQL Sever active.

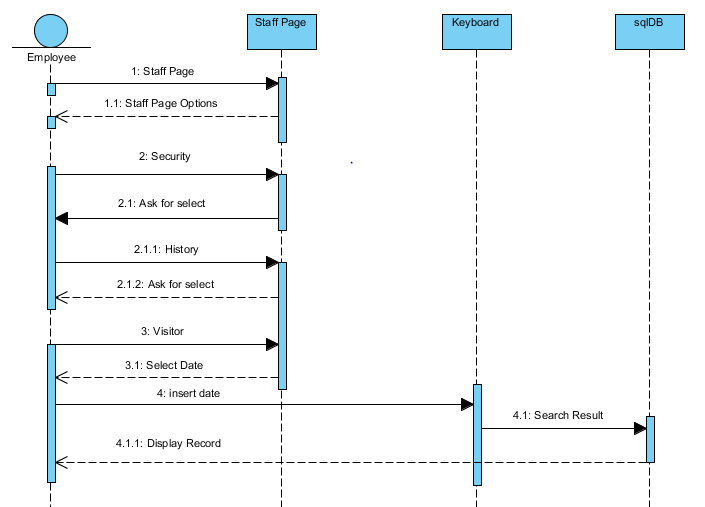
Basic flows:

1. Enter IC / Date

2. Search

3. Display Record

Alternate flows: IC/Date is required to fill.

****

**Figure 40: Visitor Entry History Sequence**

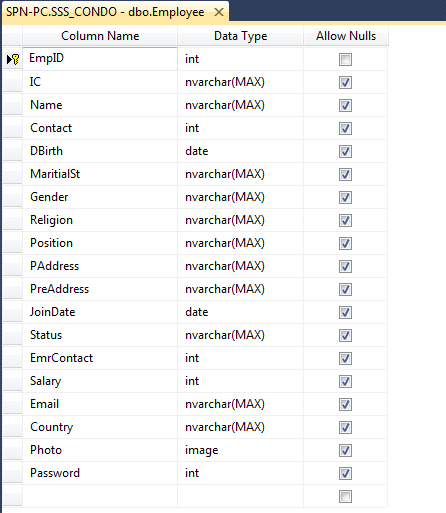


Figure 41: Employee DB Design

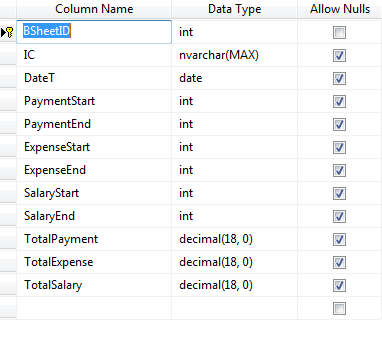


Figure 42: Balance Sheet DB Design

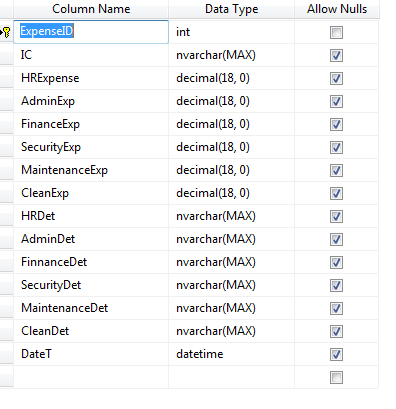


Figure 43: Expanses DB Design

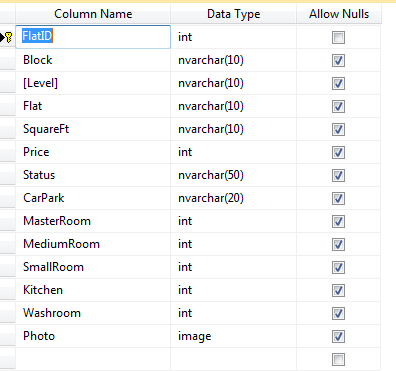


Figure 44: Flat Info DB Design

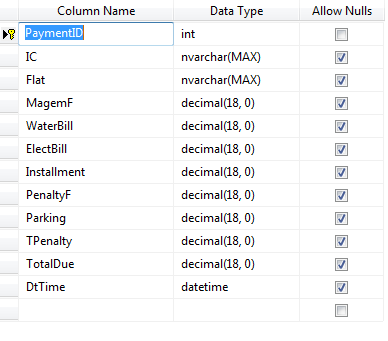


Figure 45: Payment DB Design

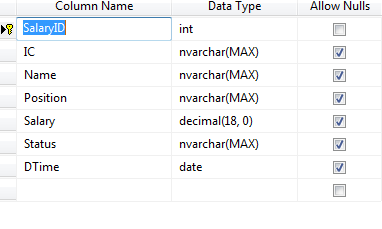


Figure 46: Salary DB Design

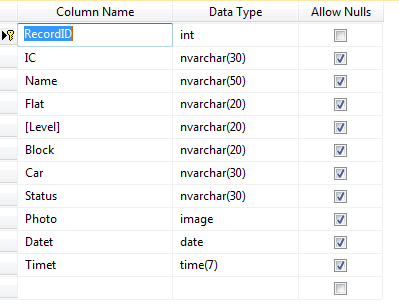


Figure 47: Security Guard Tenant DB Design

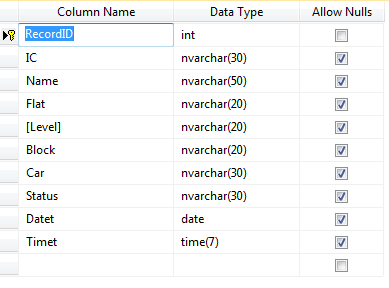


Figure 48: Security Guard Visitor DB Design

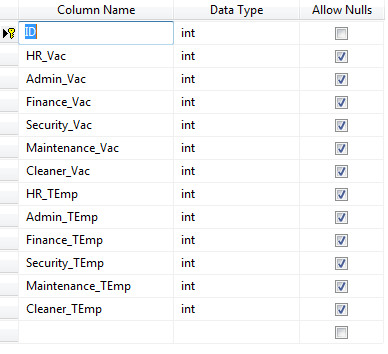


Figure 49: Hiring DB Design

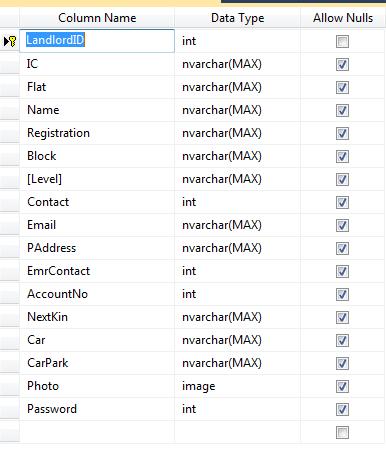


Figure 50: Landlord DB Design

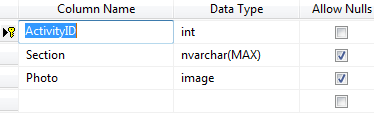


Figure 51: Staff Rules DB Design

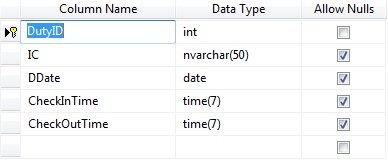


Figure 52: Staff Attendance DB Design

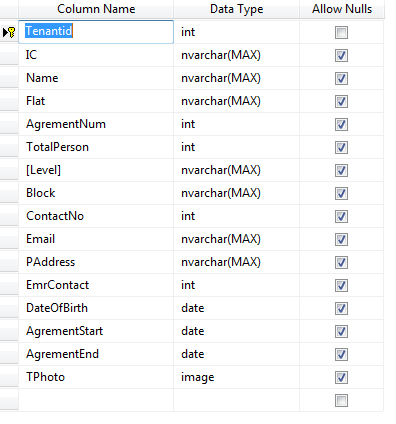


Figure 53: Tenant Registration DB Design

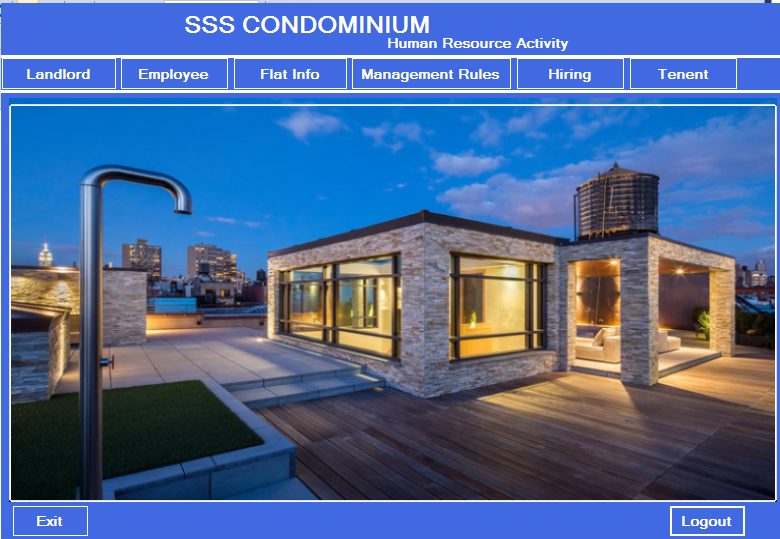
**TEST**

****

**Figure 41: Splash**

****

**Figure 42: Login Page**

****

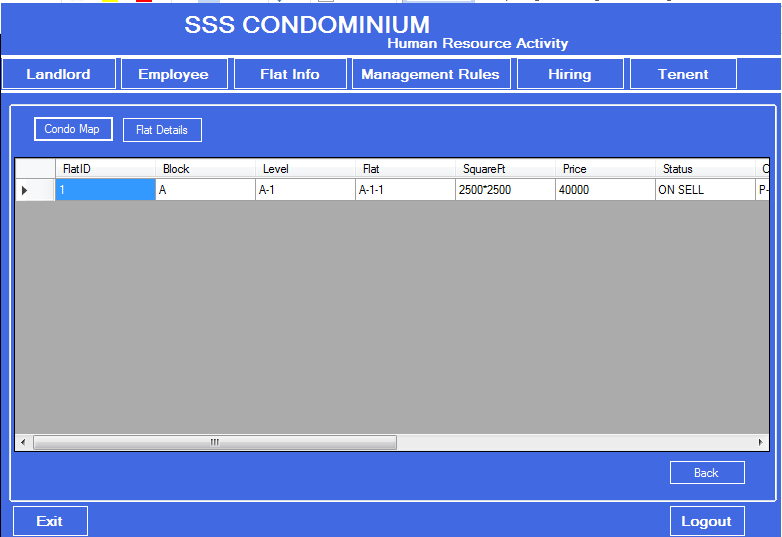
**Figure 43: HR Page**

****

**Figure 44: Landlord History**

****

**Figure 45: Employee History**

****

**Figure 46: Condo Map**

****

**Figure 47: Flat Details**

****

**Figure 48: Management Rules**

****

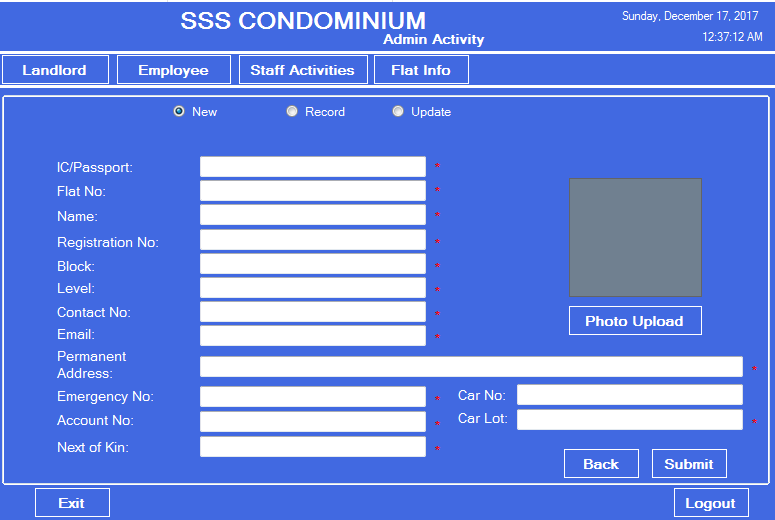
**Figure 49: Vacancy**

****

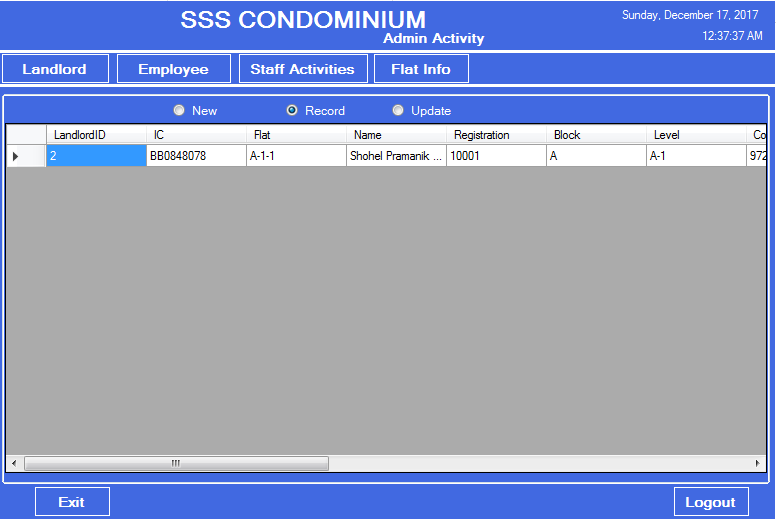
**Figure 50: Tenant Record**

****

**Figure 51: Admin Page**

****

**Figure 52: Landlord Registration**

****

**Figure 53: Landlord Record**

****

**Figure 54: Landlord Record Update**

****

**Figure 55: Employee Registration**

****

**Figure 56: Employee Record**

****

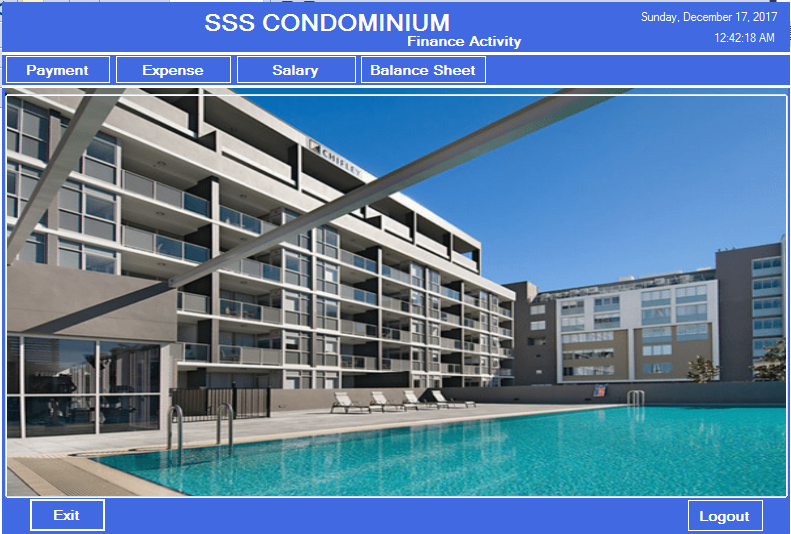
**Figure 57: Employee Update**

****

**Figure 58: Staff Activities**

****

**Figure 59: Flat Info**

****

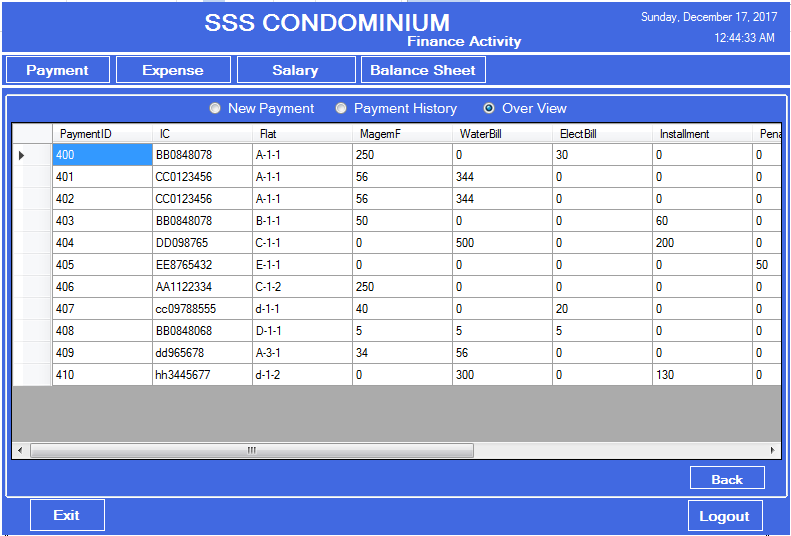
**Figure 60: Finance Page**

****

**Figure 61: New Payment**

****

**Figure 62: Payment History**

****

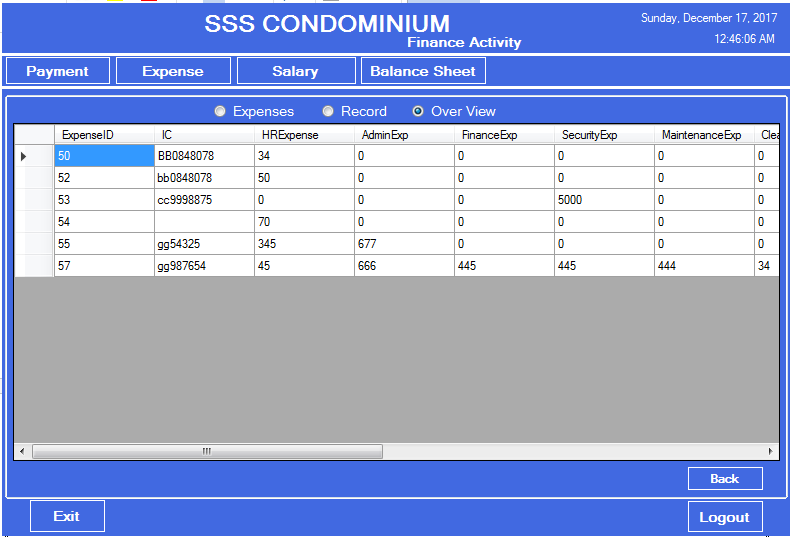
**Figure 63: Over view**

****

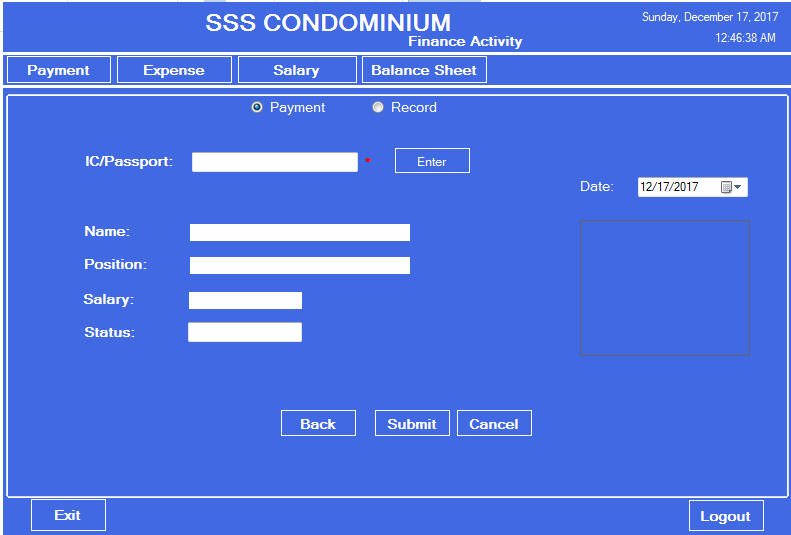
**Figure 64: Expanses**

****

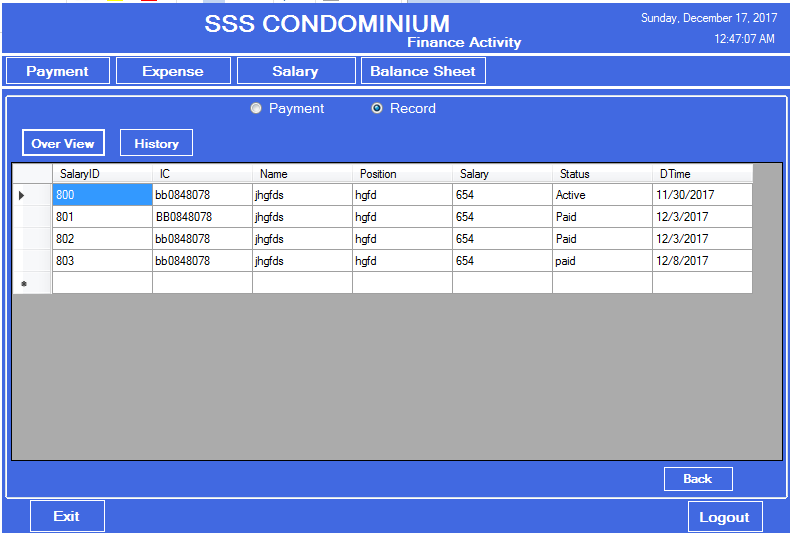
**Figure 65: Expanses Record**

****

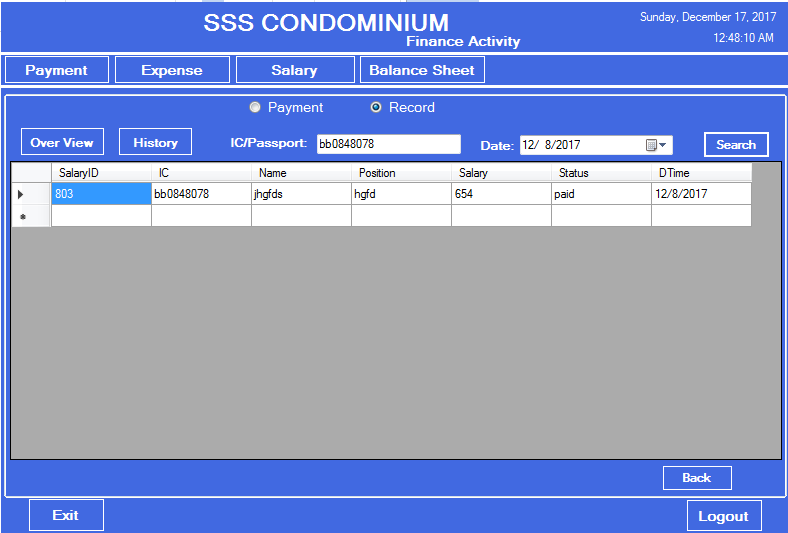
**Figure 66: Over view**

****

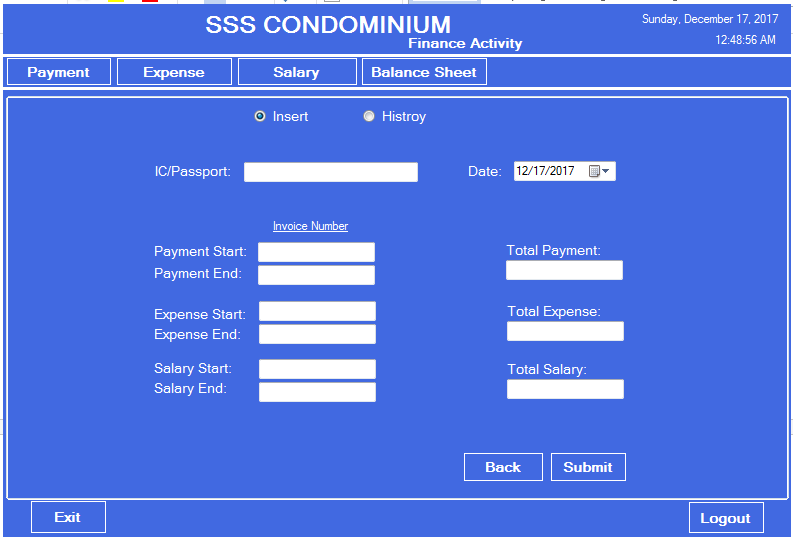
**Figure 67: Salary Payment**

****

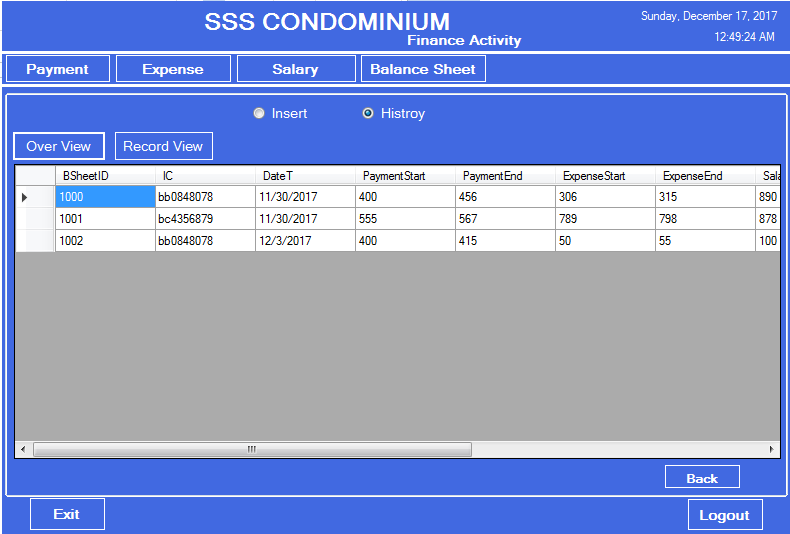
**Figure 68: Salary Record Overview**

****

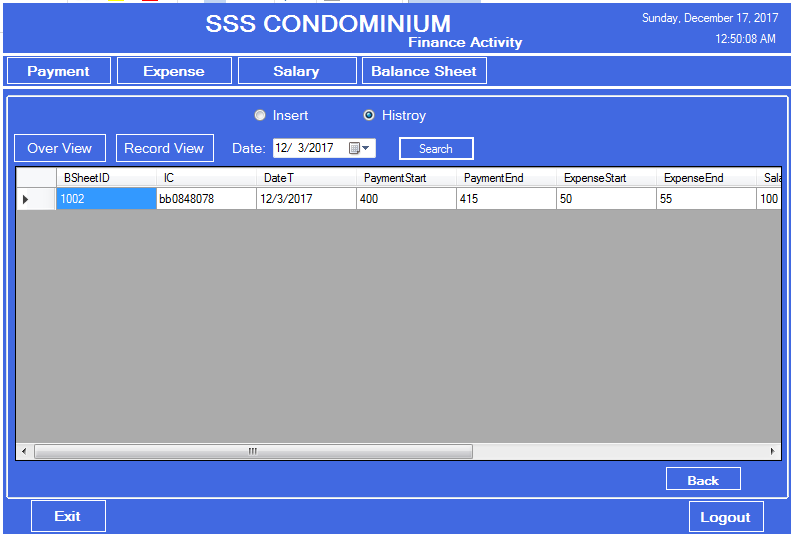
**Figure 69: History**

****

**Figure 70: Balance Sheet**

****

**Figure 71: Balance Sheet Overview**

****

**Figure 72: Balance Sheet Record Search**

****

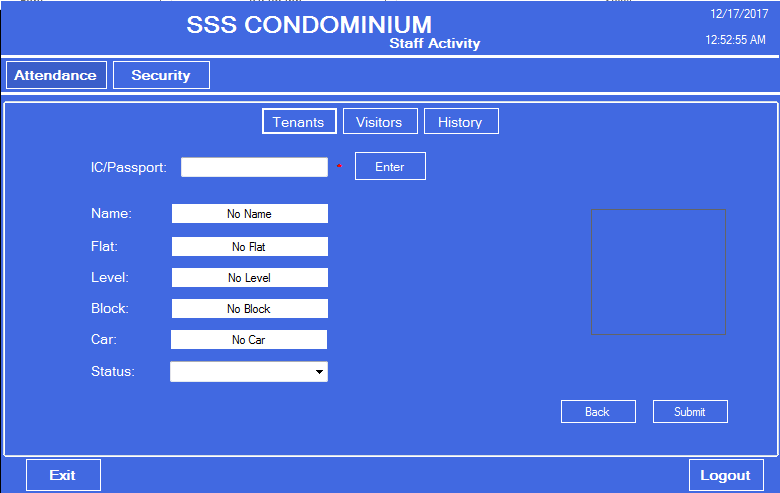
**Figure 73: Staff Page**

****

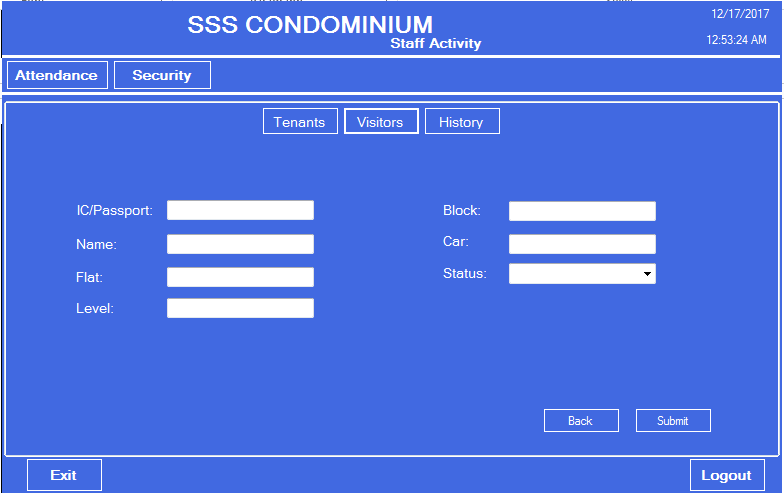
**Figure 74: Check in / Check out**

****

**Figure 75: Check in / Check out Record**

****

**Figure 76: Tenant Entry/Exit**

****

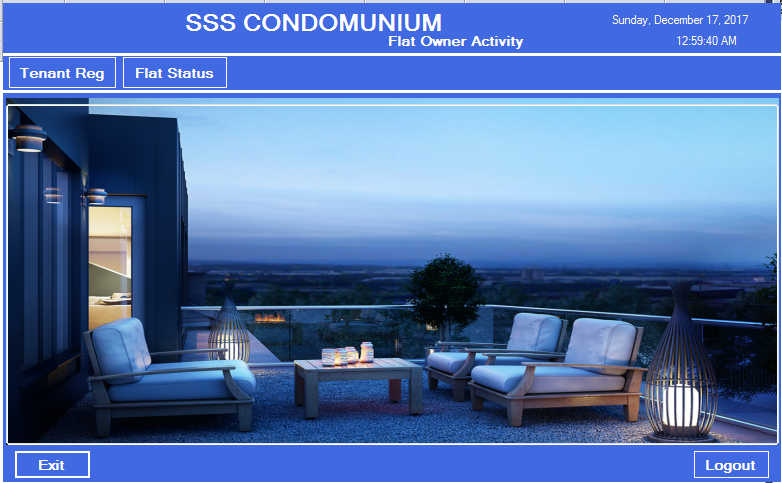
**Figure 77: Visitor Entry/Exit**

****

**Figure 78: Tenant Entry/Exit Record**

****

**Figure 79: Visitor Entry/Exit Record**

****

**Figure 80: Flat Owner Page**

****

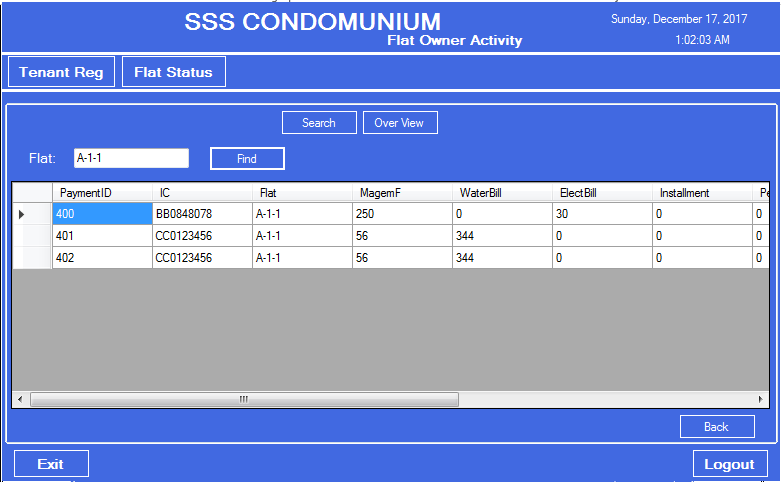
**Figure 81: Tenant Registration**

****

**Figure 82: Tenant Record Update**

****

**Figure 83: Flat Status Search**

****

**Figure 84: Flat Status Overview**

**CONCLUSION AND RECOMMENDATION**

Technology has made significant progress over the years to provide a better solution and will continue to do so for years to come. To maintain a Condominium this system can make easy to control everything securely. This windows based can further be implemented as a mobile app and web application which can be accessed by the Employee or Landlord by online from anywhere.