

Course Number: 420-DW4-AS

Course Title: Web Server Applications Development II

Session: Autumn 2023 Group: 7280

FINAL PROJECT REPORT

Property Rental

Due Date: 25 October

Submitted To

Teacher: Quang Hoang Cao

Developed By

Any Ruiz (2230023)

Jean-Michel Bérubé (2220516)

1. PROJECT DESCRIPTION

The Property Rental Management Web Site allows any potential tenant to search for an apartment suitable for his/her needs. The Web site also helps the property owner and the property manager to facilitate the management tasks of property rental.

2. PROJECT DEVELOPMENT

a. Phase I Analysis

This process phase involved reading and understanding the case study's requirements and scope. We carefully reviewed the documentation, laboratories and tutorials shared by the Professor to choose the right tools for developing the web application.

b. Phase II Design

This phase consisted mainly of making decisions related to the Database design and the website design. We ensured that our decisions during the design phase aligned with the project's objectives.

Database design

The first step of the design process consisted of deciding the database tables and the relationships between them. This step led to the PropertyRentalDB's definition, as shown in Table 1.

Tableau 1. Database Design

Table Name	Properties
Users	UserID (INT, Primary Key, Not Null) FirstName (NVARCHAR(50)) LastName (NVARCHAR(50)) RoleId (INT, Foreign Key to Roles table, Not Null) Phone (NVARCHAR(20)) AddressId (INT, Foreign Key to Addresses table, Not Null)
Roles	RoleId (INT, Primary Key, Not Null) RoleName (NVARCHAR(50))

Table Name	Properties
Addresses	AddressId (INT, Primary Key, Not Null) StreetName (NVARCHAR(100)) StreetNumber (NVARCHAR(20)) City (NVARCHAR(50)) PostalCode (NVARCHAR(10)) Country (NVARCHAR(50)) Province (NVARCHAR(50))
Login	LoginId (INT, Primary Key, Not Null) UserId (INT, Foreign Key to Users table, Not Null) Email (NVARCHAR(100)) Password (NVARCHAR(100))
Apartments	ApartmentID (INT, Primary Key, Not Null) PropertyManagerID (INT, Foreign Key to Users table, Not Null) AddressId (INT, Foreign Key to Addresses table, Not Null) StatusId (INT, Foreign Key to Statuses table, Not Null) BuildingId (INT, Foreign Key to Buildings table, Not Null) NumberOfRooms (INT) Amenities (NVARCHAR(MAX)) Price (DECIMAL(10, 2)) Floor (INT) ConstructionYear (INT) Area (FLOAT)
Statuses	StatusId (INT, Not Null, Primary Key) StatusName (NVARCHAR(50))
Apartment Images	ImageId (INT, Primary Key, Not Null) ApartmentID (INT, Foreign Key to Apartments table, Not Null) ImageURL (NVARCHAR(MAX))
Buildings	BuildingId (INT, Primary Key, Not Null) AddressId (INT, Foreign Key to Addresses table, Not Null)

Table Name	Properties
	NumberOfFloors (INT) ConstructionYear (INT) Amenities (NVARCHAR(MAX))
Appointments	AppointmentId (INT, Primary Key, Not Null) PropertyManagerID (INT, Foreign Key to Users table, Not Null) TenantID (INT, Foreign Key to Users table, Not Null) Timestamp (DATETIME) AddressId (INT, Foreign Key to Addresses table, Not Null)
Messages	MessageId (INT, Primary Key, Not Null) SenderID (INT, Foreign Key to Users table, Not Null) ReceiverID (INT, Foreign Key to Users table, Not Null) Subject (NVARCHAR(100)) MessageBody (NVARCHAR(MAX)) Timestamp (DATETIME) MessageStatusId (INT, Foreign Key to Message Statuses table, Not Null)
Message Status	MessageStatusId (INT, Primary Key, Not Null) Status (INT)
Events	EventId (INT, Primary Key, Not Null) PropertyManagerID (INT) PropertyOwnerID (INT) EventDescription (NVARCHAR(MAX)) ApartmentID (INT) Timestamp (DATETIME) EventTypeID (INT, Foreign Key to Event Types table)
Event Types	EventTypeId (INT, Primary Key) EventTypeName (NVARCHAR(50))

Web Pages Design

At this point, we decided that we needed to create the following list of Views, which also helped us define each role's access on the website.

- Login
 - Form login for all users
- Registration
 - Form Create account for Tenant
 - Form Create account for Property Manager (Admin and Property Owner only)
- Home -> Apartments (public)
 - Button to create an apartment
 - List of apartments
 - Search for an apartment
- Apartment/Id
 - Apartment Details
 - Button to update an apartment -> Redirects to the apartment form (Admin and Property Owner only)
 - Button to delete an apartment -> Redirects to the apartment form (Admin and Property Owner only)
 - Button to make an appointment with the property manager -> Redirects to the appointment form
 - Button to send messages to the property manager -> Redirects to the message form
- Apartment/Create (Admin and Property Owner only)
 - Form to create an apartment
- Apartment/Update (Admin and Property Owner only)
 - Form to update an apartment
- Apartment/Delete (Admin and Property Owner only)
 - Apartment Details
 - Button to delete an apartment
- Create Appointment
 - Form to make an appointment with the property manager

- Date picker
- Send Message
 - Form to send necessary messages to the property manager
- Buildings (Admin and Property Owner only)
 - Button to create a building -> Redirects to the building form
 - List of buildings
- Building/Id (Admin and Property Owner only)
 - Building Details
 - Button to update a building -> Redirects to the building form
 - Button to delete a building -> Redirects to the building form
- Building/Create (Admin and Property Owner only)
 - Form to create a building
- Building/Update (Admin and Property Owner only)
 - Form to update a building
- Building/Delete (Admin and Property Owner only)
 - Building Details
 - Button to delete a building
- Property Managers (Admin and Property Owner only)
 - List of property managers
 - Search for a property manager account
 - Button to see a property manager account -> Redirects to the property manager details
- Property Manager/Id (Admin and Property Owner only)
 - Property Manager Details
 - Button to update a property manager account -> Redirects to the property manager form
 - Button to delete a property manager account -> Redirects to the property manager form
- Property Manager/Create (Admin and Property Owner only)
 - Form to create a property manager account
- Property Manager/Update (Admin and Property Owner only)
 - Form to update a property manager account

- Property Manager/Delete (Admin and Property Owner only)
 - Property Manager Details
 - Button to delete a property manager account
- Potential Tenants (Admin and Property Owner only)
 - List of potential tenants
 - Search for a potential tenant account
 - Button to see a potential tenant account -> Redirects to the potential tenant details
- Potential Tenant/Id (Admin and Property Owner only)
 - Potential Tenant Details
 - Button to update a potential tenant account -> Redirects to the potential tenant form
 - Button to delete a potential tenant account -> Redirects to the potential tenant form
- Potential Tenant/Create (Admin and Property Owner only)
 - Form to create a potential tenant account
- Potential Tenant/Update (Admin and Property Owner only)
 - Form to update a potential tenant account
- Potential Tenant/Delete (Admin and Property Owner only)
 - Potential Tenant Details
 - Button to delete a potential tenant account
- Messages
 - List of messages sent
 - List of messages received
 - Search for a message
 - Button to see a message -> Redirects to the message details
 - Button to send a message -> Redirects to the message form
- Message/Id
 - Message Details
 - Button to answer a message -> Redirects to the create message form
- Message/Create
 - Form to create a message

- Message/Update (Admin and Property Manager only)
 - Form to update a message
- Message/Delete (Admin and Property Manager only)
 - Message Details
 - Button to delete a message
- Events (Admin and Property Manager only)
 - List of events
 - Search for an event
 - Button to see an event -> Redirects to the event details
 - Button to create an event -> Redirects to the event form
- Event/Id
 - Event Details
 - Button to update an event -> Redirects to the event form
 - Button to delete an event -> Redirects to the event form
- Event/Create
 - Form to create an event
- Event/Update
 - Form to update an event
- Event/Delete
 - Event Details
 - Button to delete an event

c. Phase III Implementation

The implementation for the project can be found in the GitHub Repository <https://github.com/JeanMichelBB/final-project-web>.

This phase involved building the foundation of the application using ASP.NET MVC, Entity Framework and MS SQL Server. We developed custom controllers and models for different areas of the application, like the ***Accounts Controller*** and ***Login Model*** for managing the Login/Signup, the ***Users Role Provider*** for handling the different roles in the whole application, as well as the ***Appointment View Model*** and ***Building View Model*** for handling

information that was coming from the user interface but was not necessarily in the same format in the Database. For instance, for the appointments, we had a Date Picker and a Time selector in the UI but only a field called Timestamp in the Database. Equally, for the building creation form, we had all the fields for the address in the UI, but in the database, we only had the address ID, which involved creating the address to assign it to the corresponding building later.

All of this ensures that the renters, owners, and managers have the best possible user experience. Phase III was a significant step in transforming our concept and design into a workable web application.

Figure 1 shows the diagram with the implementation of the Database in MSSQL Server.

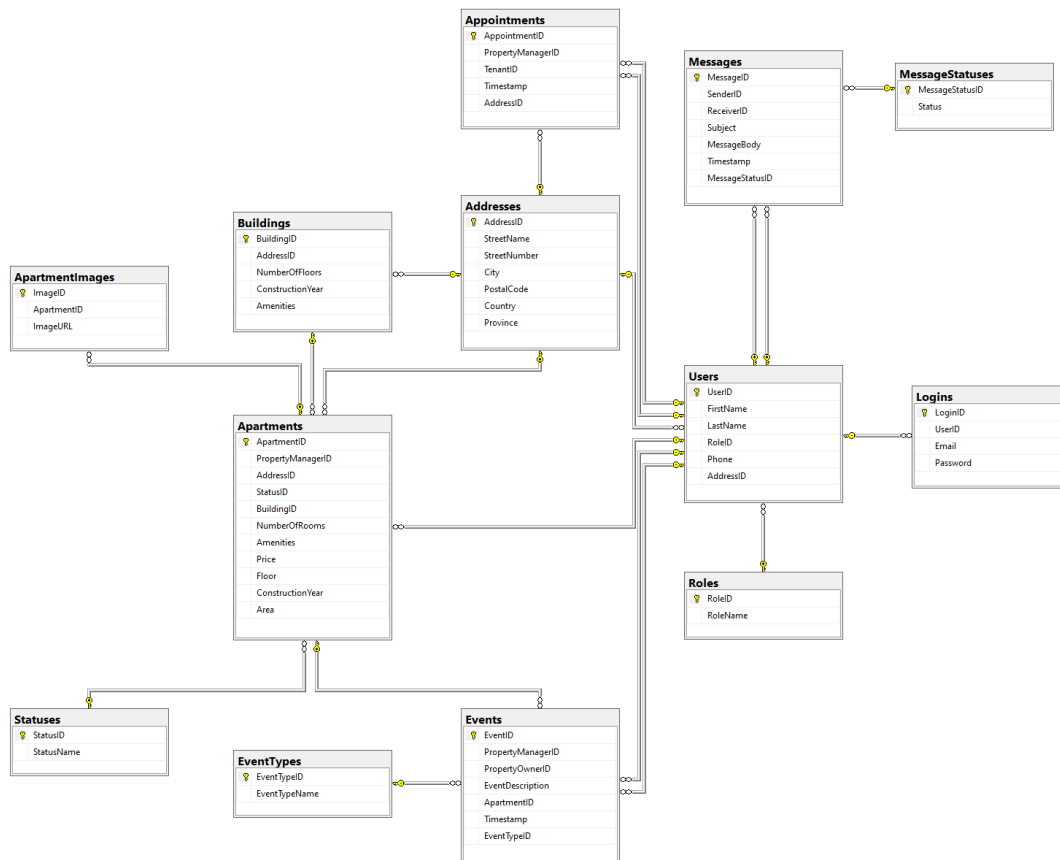


Figure 1. Data base Relationship Diagram

Figure 2 shows the main page view with the list of apartments for a user logged in as an administrator.

Property Rental INC ®

[Home](#) [Messages](#) [Appointments](#) [Users](#) [Buildings](#) [Events](#)

Logout

Logged in as: admin@example.com

Apartments

[Create New](#)

Number of Rooms	Price	Amenities	Area	Floor	Property Manager	
2	1200,00	Balcony, Furnished, Cats allowed	800	2	Mary Brown	Edit Details Delete
3	1500,00	Balcony, Patio, Hard wood floor	1100	4	Mary Brown	Edit Details Delete
1	900,00	Patio, Hard wood floor	500	10	Bob Johnson	Edit Details Delete
4	2000,00	Balcony, Jacuzzi, Patio, Hard wood floor	1000	10	Bob Johnson	Edit Details Delete

Developed by Any and Jean-Michel

© 2023 - Property Rental INC ®

Figure 2. Apartments List View as an Admin

Figure 3 shows the list of Appointments, and Figure 4 the form for creating a Building in the website.

Property Rental INC ®

[Home](#) [Messages](#) [Appointments](#) [Users](#) [Buildings](#) [Events](#)

Logout

Logged in as: admin@example.com

Appointments

[Create New](#)

Date	Time	Address	Property Manager	
2023-10-17	9:00 AM	123 Main Street Apt 101 Cityville Stateville	Mary Brown	See details Reschedule Cancel

Developed by Any and Jean-Michel

© 2023 - Property Rental INC ®

Figure 3 View with the List of Appointments

Property Rental INC ®
Home Messages Appointments Users Buildings Events
Logout
Logged in as: admin@example.com

Edit Building

Street Name

123 Main Street

Street Number

Apt 101

City

Cityville

Postal Code

12345

Country

Countryland

Province

Stateville

Number Of Floors

5

Construction Year

2010

Amenities

Parking, Gym, Pool

Building Name

The Mountain Palace

Save

Back to List

Developed by Any and Jean-Michel
© 2023 - Property Rental INC ®

Figure 4 View with the form for creating a Building

In this step, we tried to make the website as user-friendly as possible. Using Bootstrap and Razor pages, we created a visually clean, consistent, and functional design. We incorporated all necessary features during the implementation phase, such as property management tools, property visualization, user registration, login, and more.

d. Phase IV Testing the program

This phase consisted of testing the different use cases, and edge cases for all the users and browsers, comparing with the application's functional requirements. Table 2 shows a summary of the results obtained.

Tableau 2 Functional Requierements

User	Functional Requirement	Result
Property Owner	Create/Update Delete/List any property manager account	Working

	Update/Delete any potential tenant account	Working
	Complete control of the Web Site	Working
Property Manager	Perform CRUD operations related to buildings	Working
	Perform CRUD operations related to apartments	Working
	Keep track of apartments status	Working
	Schedule potential tenants 'appointments	Working
	Respond to potential tenants 'messages	Working
	Report any events to the property owner when necessary	Working
Tenant	Create an online account through Property Rental Management Web Site	Working
	View any apartment suitable for their needs	Working
	Make an appointment with the property manager	Working
	Send necessary messages to the property manager	Working

3. CONCLUSIONS

We have a user-friendly "Property Rental Management Web Site" project. We selected technology and database architecture carefully. We still have room for improvement, but given the short time and our amount of knowledge, we feel satisfied with the results.

We learned how to do data validation, manage user roles, authentication, and authorization, and handle different user actions depending on the HTTP request and the URLs visited by the user. We finally could see how the Models interact with the Controllers and with the Views. Finally, we learned how important it is to do a High-Level Analysis and Design before implementing any solution.

4. RESOURCES

- Visual Studio - Integrated Development Environment (IDE) for project development.
- SQL Server Management Studio (SSMS) - Database management and query tool.
- SQL Server Express - A relational database management system used in the project.
- Visual Studio Code - Code editor for additional development and scripting.
- GitHub - Version control and collaboration platform for code management.
- ASP.NET MVC - The chosen development framework for web application development.
- Bootstrap - The responsive web design framework used to create a user-friendly interface.
- ASP.NET MVC Pattern Documentation - [ASP.NET MVC Pattern | .NET \(microsoft.com\)](#)
- Authentication and authorization in MVC - [Authentication and Authorization in MVC - Dot Net Tutorials](#)
- Forms Authentication in ASP.NET MVC - [FormsAuthentication in ASP.NET MVC - Dot Net Tutorials](#)
- Role-based Authentication in ASP.NET MVC - [Role-Based Authentication in ASP.NET MVC - Dot Net Tutorials](#)