

Student: Maciej Mateusz Madejsza
ID: MAD21541198
Date: 21/12/2023
Project: Coursework - Sprint 3
Target: Higher marks - enhanced version

In the following document, I'll do my best to walk you through all coursework requirements I've met with outcome screenshots and descriptions with screenshots of the enhanced version. To notice all extra features and robustness It'll be required however to play around with the program – It's a bit too extended for documentation with screenshots here.

Interface Layout:

The screenshot displays a web application interface with a title bar 'MainWindow' and a toolbar. The interface is divided into three main sections, each outlined with a red border:

- Entities input panel:** This section is used for adding new entities. It contains two main columns:
 - Choose person to add:** Includes a dropdown for 'Landlord', input fields for 'Forename:', 'Surname:', and 'Email:', and checkboxes for 'Flagged' and 'Premium'. An 'Add Person' button is at the bottom.
 - Choose property type:** Includes a dropdown for 'Room', input fields for 'Landlord:', 'Price:', and 'Luxury lvl:', and checkboxes for 'Self check-in', 'Disabled friendly', 'Smoke alarm', 'Separate Entrance', 'Garden', and 'Private key'. An 'Add Property' button is at the bottom.
- Functions panel:** This section is used for performing actions and listing data from the database.
 - Perform action by particular person:** Includes dropdowns for 'Person type:', 'Person:', 'Action:', and 'Target:', and a 'Perform' button.
 - List from database:** Includes dropdowns for 'Entity:', 'Feature:', and 'Person:', and a 'List' button.
- Action Log:** A large green rectangular area at the bottom of the interface, used for displaying the outcome of actions and instructions when needed.

Annotations with callout boxes provide further context:

- A box pointing to the 'Entities input panel' states: 'Area to input entities to the program like in Sprint 2 – users and bookable items'.
- A box pointing to the 'List from database' section states: 'Area to perform more complex actions inc. **booking**'.
- A box pointing to the 'Action Log' area states: 'Area displaying actions outcome and instructions when needed.'

Sprint 3 Basic requirements (3rd – 2.2)

- Button that can respond to a click

- Phone restricted to limit inputs amount for coursework assessment purposes

Choose person to add:

Forename:

Surname:

Email:

☐ Flagged

☒ Premium

Added:

-----Landlord 5-----

ID: 5

Name: Maciej Madejsza

Phone: Phone protected

E-mail: maciej@gmail.com

Premium: Yes

Flagged: No

Choose person to add:

Forename:

Surname:

Email:

☐ Flagged

☐ Premium

Fill up the details first.

Empty form validation

- List box implemented

☐ Flagged

- Plus many other visible everywhere – good way to validate by limiting certain actions based on certain choices

- Use of classes from sprint 1 in 'code behind'

```

// Base class for properties, following DatabaseObject interface (including people and properties). Abstract because never sep
29 references
abstract class PersonBase : IDatabaseObject
{
    // Declaring set of properties held by Landlords and Tenants created based on this base
    protected string ID;
    protected string firstName;
    protected string lastName;
    protected string phone;
    protected bool premiumMember;
    protected bool redFlag;
    // Publicly accessible:
    public string e_mail;

    // Passing access to instance of singleton database (only 1 exists)
    protected Database database = Database.GetInstance();

    // Constructor:
    2 references
    public PersonBase(string firstName, string lastName, string e_mail, string phone, bool premium_member, bool red_flag, string ID)
    {
        this.ID = ID;
        this.firstName = firstName;
        this.lastName = lastName;
        this.e_mail = e_mail;
        this.phone = phone;
        this.premiumMember = premium_member;
        this.redFlag = red_flag;
    }

    //-----Getters-----
    // Method providing all personToListBy details formatted in a block of information
    5 references
    public virtual string GetFullDetails()
    {
        return
            $"-----{this.GetType()} {this.GetID()}-----\n" +
            $"ID: {this.GetID()}\n" +
            $"Name: {this.GetFormattedName()}\n" +
            $"Phone: {this.GetPhoneString()}\n" +
            $"E-mail: {this.e_mail}\n" +
            $"Premium: {(this.IsPremium() == false ? "No" : "Yes")}\n" +
            $"Flagged: {(this.IsRedFlagged() == false ? "No" : "Yes")}\n";
    }
}

```

Part of the Landlord base class with method responsible for printing out the details like above after adding to the database

- List at least three bookable items displayed in the GUI generated by your sprint c# OOP code
 - Like in sprint 2 – we have this ability to input them manually – not generated randomly

Choose property type:

Room

Landlord: Maciej Madejsza (ID:0)
Lisa Haskel (ID:1)

Price: 500

Luxury lvl: 3

☐ Self check-in
☐ Disabled friendly
☒ Smoke alarm

☐ Separate Entrance
☐ Garden
☒ Private key

Add Property

3x

List from database:

Entity: Properties

Feature: All

Person:

List

Room 2

ID: 2
Owner: Maciej Madejsza
Luxury Lvl: 3
Price: 500
Type: To live
Host language: English
Smoke alarm: Yes
Separate entrance: No
Private key: Yes
State: This Room (ID: 2) is FREE!

Flat 3

ID: 3
Owner: Lisa Haskel
Luxury Lvl: 5
Price: 600
Rooms number: 3
Self Check-In: Yes
Disabled friendly: No
Separate entrance: Yes
State: This Flat (ID: 3) is FREE!

House 4

ID: 4
Owner: Maciej Madejsza
Luxury Lvl: 5
Price: 1000
Rooms number: 4
Self Check-In: Yes
Disabled friendly: Yes
Garden: Yes
State: This House (ID: 4) is FREE!

Scrolled

Advanced requirements (2.1 and above) The above plus...

- User can select a bookable item and an action is performed

Perform action by particular person:

Person type: Tenants

Person: (ID:5) T1 Forname T1 Surname

Target: (ID:4) House ID 4

Action: Book

Perform

List from database:

Entity: Properties

Feature: Booked

Person:

List

House (ID: 4) booked! Enjoy your stay T1 Forname T1 Surname!

Choose action.

House 4

ID: 4
Owner: Maciej Madejsza
Luxury Lvl: 5
Price: 1000
Rooms number: 4
Self Check-In: Yes
Disabled friendly: Yes
Garden: Yes
State: This House (ID: 4) is BOOKED by T1 Forname T1 Surname!

- Advanced features of your choice:

- User - there are 3 types of users with different privileges **in code behind**: Admin, Landlord and Tenant. To keep interface reasonable simple for assessment – only Landlord and Tenant are pulled out to interface in ‘functions panel’

Choose person to add:

Forename:

Surname:

Email:

Person type: Landlord

☐ Flagged
☐ Premium

Perform action

Person type: Landlords

Action: Remove
Suspend
Restore
Show properties

Person type: Tenants

Action: Book
Release
Show bookings

- Design patterns used:
- State pattern: to achieve smooth state instances of the properties and to be able to flexibly implement booking and **suspension features (extra)**.

- Booking, releasing, and suspending change property state

Tenant

Action:

Book

Target:

(ID:2) Room ID 2

(ID:3) Flat ID 3

Release

(ID:4) House ID 4

Show bookings

All

Perform

Perform

Perform

Showing only these properties which are FREE = bookable!

Showing only these properties which are BOOKED by THIS tenant!

Searching limited - Showing only all bookings which belong to this person!

Person type:

Landlords

Person:

(ID:0) Maciej Madejsz

Action:

Remove

Target:

(ID:2) Room ID 2

(ID:2) Room ID 2

Perform

Person type:

Landlords

Person:

(ID:0) Maciej Madejsz

Action:

Suspend

Target:

(ID:2) Room ID 2

(ID:4) House ID 4

Perform

Restore

Nothing suspended yet

Perform

May be booked (4)

Showing only these properties which are FREE and belong to this person!

Showing only these properties which are NOT SUSPENDED and belong to this person!

Showing only these properties which are SUSPENDED and belong to this person!

- Singleton: to achieve 1 common database and monitor the occurrence of objects. That allowed to implement **listing of particular landlord or tenant properties**. It allowed to populate conditionally drop lists based on chosen person or property to allow/restrict different actions.

Choose property type:

Room

Price:

Landlord:

Maciej Madejsza (ID:0)

Rooms:

5

Separate Entrance

Garden

Private key

Add Property

Choose property type:

Flat

Price:

Landlord:

Maciej Madejsza (ID:0)

Rooms:

1

2

List from database:

Entity:

Properties

Properties

Landlords

Tenants

Person:

List

List from database:

Entity:

Properties

Feature:

All

Free

Booked

Suspended

Not suspended

Person

Person:

(ID:0) Maciej Madejsza

(ID:1) Lisa Haskel

(ID:5) T1 Forname T1 Surname

Room 2

ID: 2

Owner: Maciej Madejsza

Luxury Lvl: 3

Price: 500

Type: To live

Host language: English

Smoke alarm: Yes

Separate entrance: No

Private key: Yes

State: This Room (ID: 2) is FREE!

Scroll down

Feature:

Person

Person:

(ID:5) T1 Forname T1 Surname

List

House 4

ID: 4

Owner: Maciej Madejsza

Luxury Lvl: 5

Price: 1000

Rooms number: 4

Self Check-In: Yes

Disabled friendly: Yes

Garden: Yes

State: This House (ID: 4) is BOOKED by T1 Forname T1 Surname!

List from database:

Entity:
Landlords

Feature:
All
Active
Premium
Flagged

List

The same features set for
Landlords and Tenants.

Person drop list
deactivated

- Various validation – based on conditional access to different actions based on user and property type + simple inputs validation.