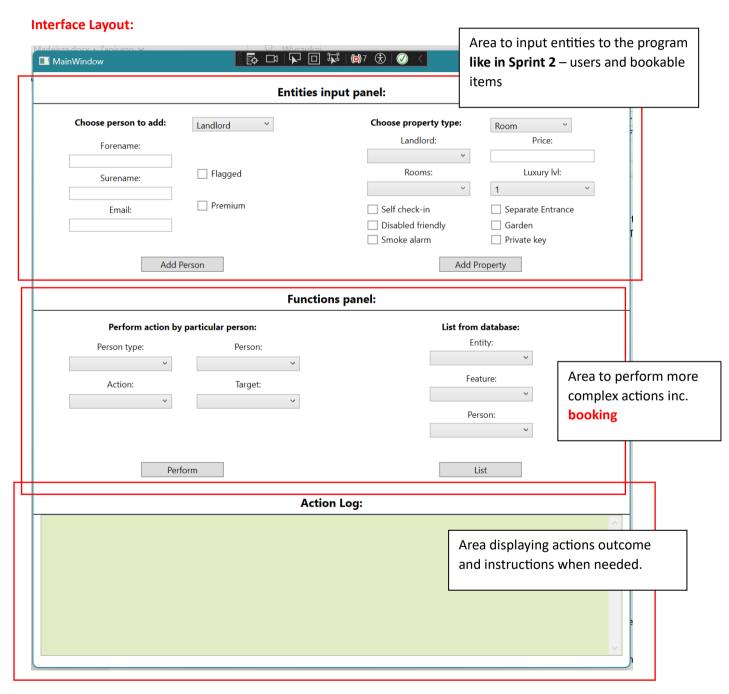
Student: Maciej Mateusz Madejsza

ID: MAD21541198 Date: 21/12/2023

Project: Coursework - Sprint 3

Target: Higher marks - enhanced version

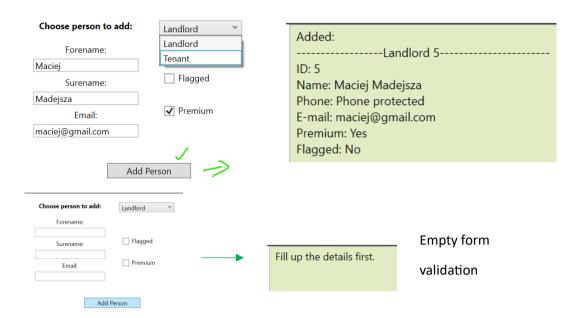
In the following document, I'll do my best to walk you through all coursework requirements I've met with outcome screenshots and descriptions with screenshots of the enhanced version. To notice all extra features and robustness It'll be required however to play around with the program – It's a bit too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



## Sprint 3 Basic requirements (3rd - 2.2)

• Button that can respond to a click

- Phone restricted to limit inputs amount for coursework assessment purposes



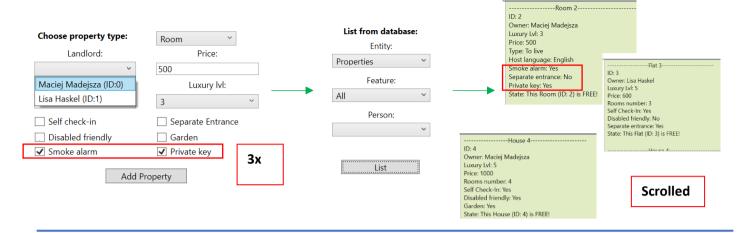
• List box implemented



- 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'

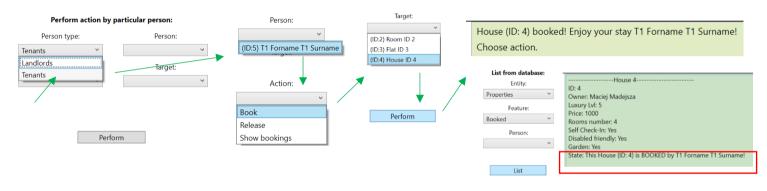
```
### State of properties, retrough database of interact introducting people and properties). Additional Control of the Control
```

- 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

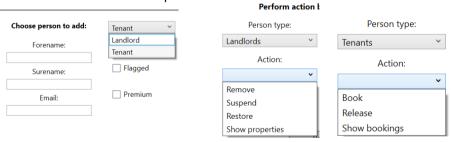


## Advanced requirements (2.1 and above) The above plus...

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



- 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'



- 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: Target: Release Show bookings All Book (ID:4) House ID 4 (ID:2) Room ID 2 (ID:3) Flat ID 3 Perform Perform Actior 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: Person Landlords (ID:0) Maciej Madejszi 🗡 Landlords (ID:0) Maciei Madeiszi Y Restore Target: Action: Action: Target (ID:2) Room ID 2 Suspend (ID:2) Room ID 2 (ID:2) Room ID 2 Nothing Perform suspended yet Perform Perform May be booked (4) Showing only these properties which are SUSPENDED and belong to this person Showing only these properties which are NOT SUSPENDED and belong to this person Showing only these properties which are FREE 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: Choose property type: Room Flat Landlord: Landlord: Maciej Madejsza (ID:0 ~ Maciej Madejsza (ID:0 Y Rooms Luxury Ivl Rooms: Luxury lvl: Separate Entrance Garden Separate Entrance ☐ Private key 2 roperty List from database: Person: List from database: Entity: Entity: -----Room 2---**Properties** Properties Owner: Maciej Madejsza **Properties** Feature: Luxury Lvl: 3 Landlords Price: 500 All Type: To live Scroll down Tenants Feature: Free Host language: English Person: Person Smoke alarm: Yes Booked Person: Separate entrance: No Suspended Private key: Yes Not suspended (ID:0) Maciej Madejsza State: This Room (ID: 2) is FREE! Person (ID:1) Lisa Haskel

(ID:5) T1 Forname T1 Surname

Owner: Maciej Madejsza

Luxury Lvl: 5

Rooms number: 4 Self Check-In: Yes Disabled friendly: Yes Garden: Yes

Price: 1000

-House 4-

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

List

Person

Feature:

Person:

(ID:5) T1 Forname T1:

List

## List from database: Entity: Landlords Feature: The same features set for Landlords and Tenants. Person drop list deactivated List

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