

Підсистема GUI – система графічного інтерфейсу, необхідного для наглядного та зрозумілого використання можливостей системи.

Роль – графічне представлення використання можливостей

Юзкейси підсистеми GUI

1. Відкриття вікна

Діючі особи:

1. Window
2. Handler
3. Користувач

Мета: вивести користувачу на екран вікно

Основний сценарій:

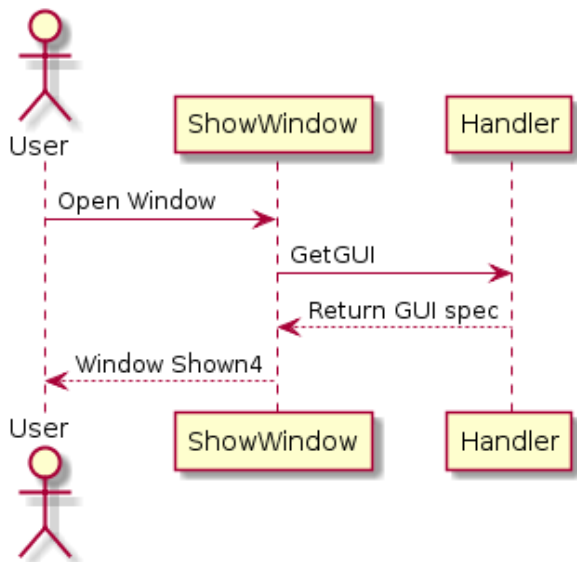
1. Користувач відкриває вікно
2. Window запитує у Handler розмітку GUI
3. Window отримує розмітку GUI
4. Користувач бачить вікно

Код у PlantUML:

```
@startuml
actor User
participant ShowWindow
participant Handler
```

```
User -> ShowWindow: Open Window
ShowWindow -> Handler: GetGUI
Handler --> ShowWindow: Return GUI spec
ShowWindow --> User: Window Shown4
```

```
@enduml
Діаграма у PlantUML:
```



2. Показати розклад

Діючі особи:

1. Користувач
2. Window
3. Handler

Мета: показати користувачу розклад

Основний сценарій:

1. Користувач відкриває вікно(юзкейс Відкриття вікна) та вводить номер аудиторії
2. Window передає номер в Handler
3. Handler перевіряє номер на валідність
4. Handler отримує розклад (юзкейс BaseLevel Отримання розкладу)
5. Handler форматує дані в словник, де ключами є аудиторії, а значенням – викладач, час пари та групи.
6. Handler передає відформатований розклад в Window
7. Window виводить розклад на екран в поле Label

Альтернативний сценарій:

- 3.1 Номер аудиторії не валідний
- 3.2 Handler передає помилку в Window
- 3.3 Window виводить Label з текстом «Неправильно введений номер аудиторії»

Код у PlantUML:

@startuml

actor User

```

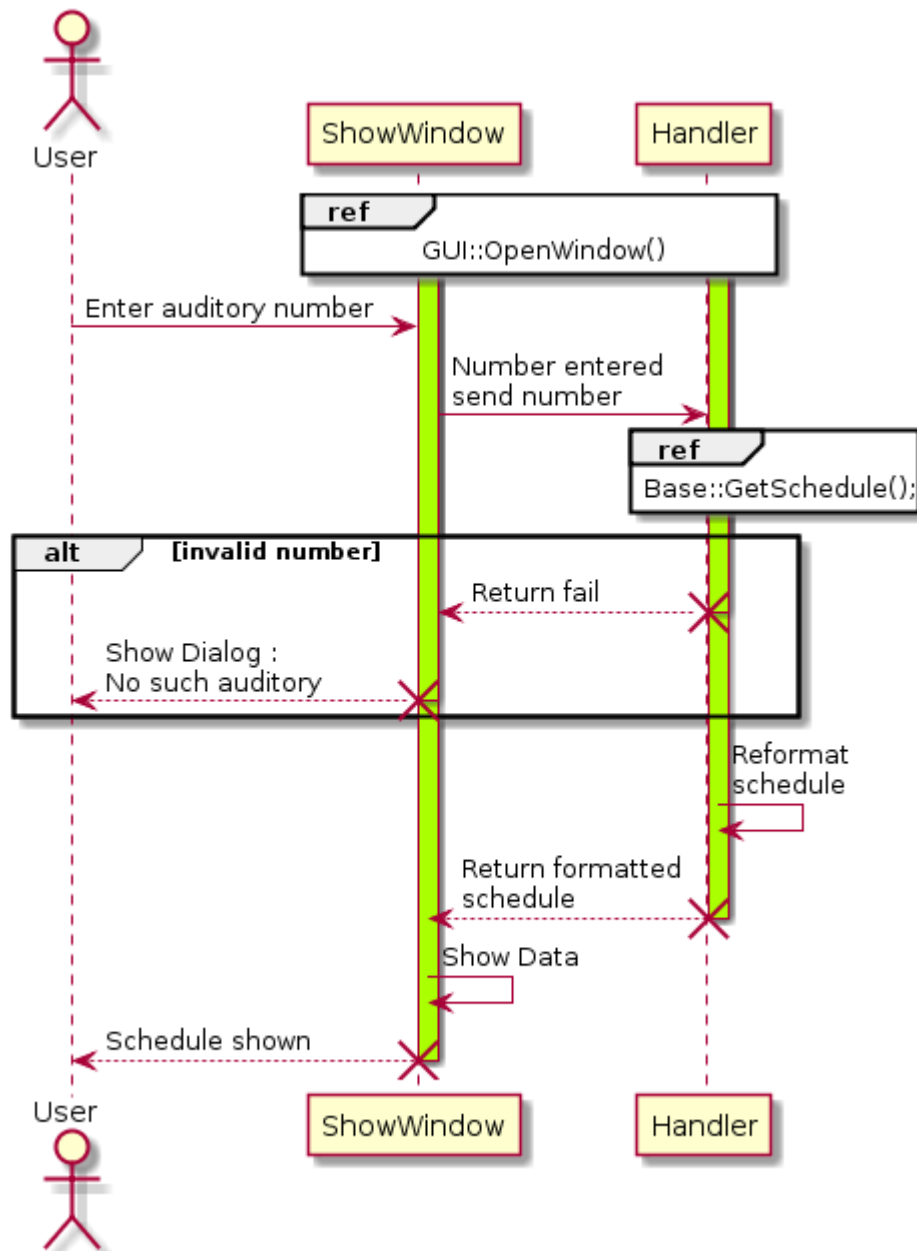
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
activate Handler
activate Handler #aaff00
User -> ShowWindow : Enter auditory number

ShowWindow -> Handler : Number entered\nsend number
ref over Handler : Base::GetSchedule();
alt invalid number
Handler --> ShowWindow : Return fail
destroy Handler
ShowWindow --> User : Show Dialog :\nNo such auditory
destroy ShowWindow
end
Handler -> Handler : Reformat\nschedule
Handler --> ShowWindow : Return formatted\nschedule
destroy Handler
ShowWindow -> ShowWindow : Show Data
ShowWindow --> User : Schedule shown
destroy ShowWindow

```

@enduml

Діаграма у PlantUML:



3. Додавання тимчасових прав доступу

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію додавання прав доступу

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та вводить UID та час доступу
2. Window показує діалогове вікно для підтвердження операції
3. Window передає дані в Handler

4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Додавання тимчасових прав доступу)
6. Handler повертає до Window оновлені права доступу
7. Window відображає оновлені права доступу в полі Label та виводить повідомлення про успішне завершення операції

Альтернативний сценарій:

- 2.1 Користувач відхилив підтвердження операції
- 2.2 Повернути користувача на початкове вікно
- 4.1 UID або час доступу не валідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введений UID або час доступу»

Код у PlantUML:

@startuml

actor User

ref over ShowWindow, Handler: GUI::OpenWindow()

activate ShowWindow

activate ShowWindow #aaff00

activate Handler

activate Handler #aaff00

User -> ShowWindow : Enter UID and access time

ShowWindow --> User : Show Dialog : \nconfirmation

ShowWindow -> Handler : Data entered\nsend data

alt declining

ShowWindow --> User: Back to Window

end

ref over Handler : Base::AddTempLaws();

alt invalid data

Handler --> ShowWindow : Return fail

destroy Handler

ShowWindow --> User : Show Dialog : \nInvalid UID or schledule

destroy ShowWindow

end

Handler --> ShowWindow : Return updated\nlaws

destroy Handler

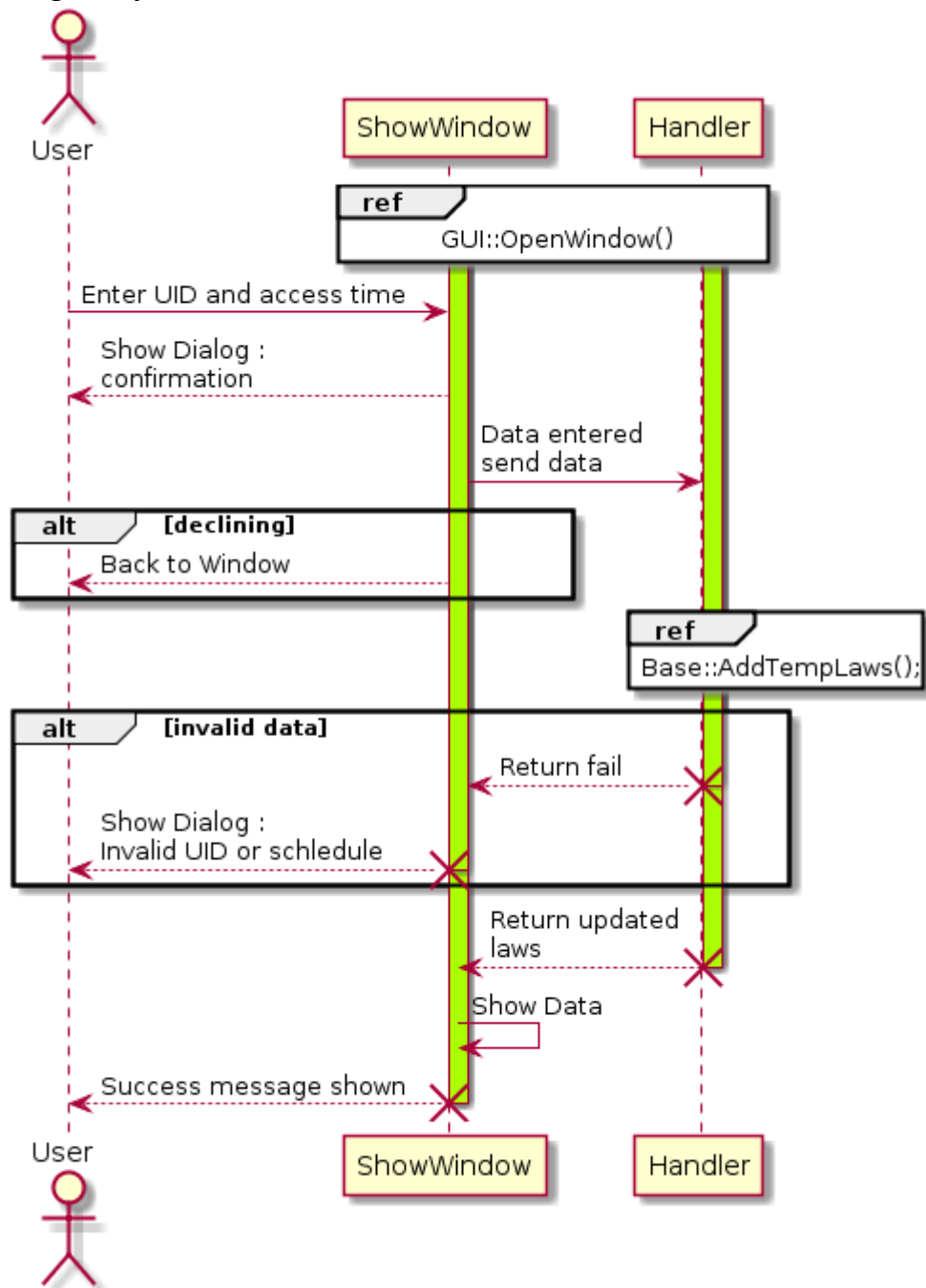
ShowWindow -> ShowWindow : Show Data

ShowWindow --> User : Success message shown

destroy ShowWindow

@enduml

Діаграма у PlantUML:



Додавання викладача

Діючі особи:

1. Користувач
2. Window
3. Handler

Мета: забезпечити візуалізацію додавання викладача

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та вводить UID та розклад нового викладача у форму в вигляді «Пара-група-номер пари»
2. Window показує діалогове вікно для підтвердження операції
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Додавання викладача)
6. Handler повертає до Window оновлений розклад
7. Window відображає оновлений розклад в полі Label у вигляді «Доданий викладач-пара-група-номер пари»

Альтернативний сценарій:

- 2.1 Користувач відхилив підтвердження операції
- 2.2 Повернути користувача на початкове вікно
- 4.1 UID або розклад не валідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введений UID або розклад»

Код у PlantUML:

@startuml

actor User

ref over ShowWindow, Handler: GUI::OpenWindow()

activate ShowWindow

activate ShowWindow #aaff00

activate Handler

activate Handler #aaff00

User -> ShowWindow : Enter teacher's UID and schedule

ShowWindow --> User : Show Dialog : \nconfirmation

ShowWindow -> Handler : Data entered\nsend data

alt declining

ShowWindow --> User: Back to Window

end

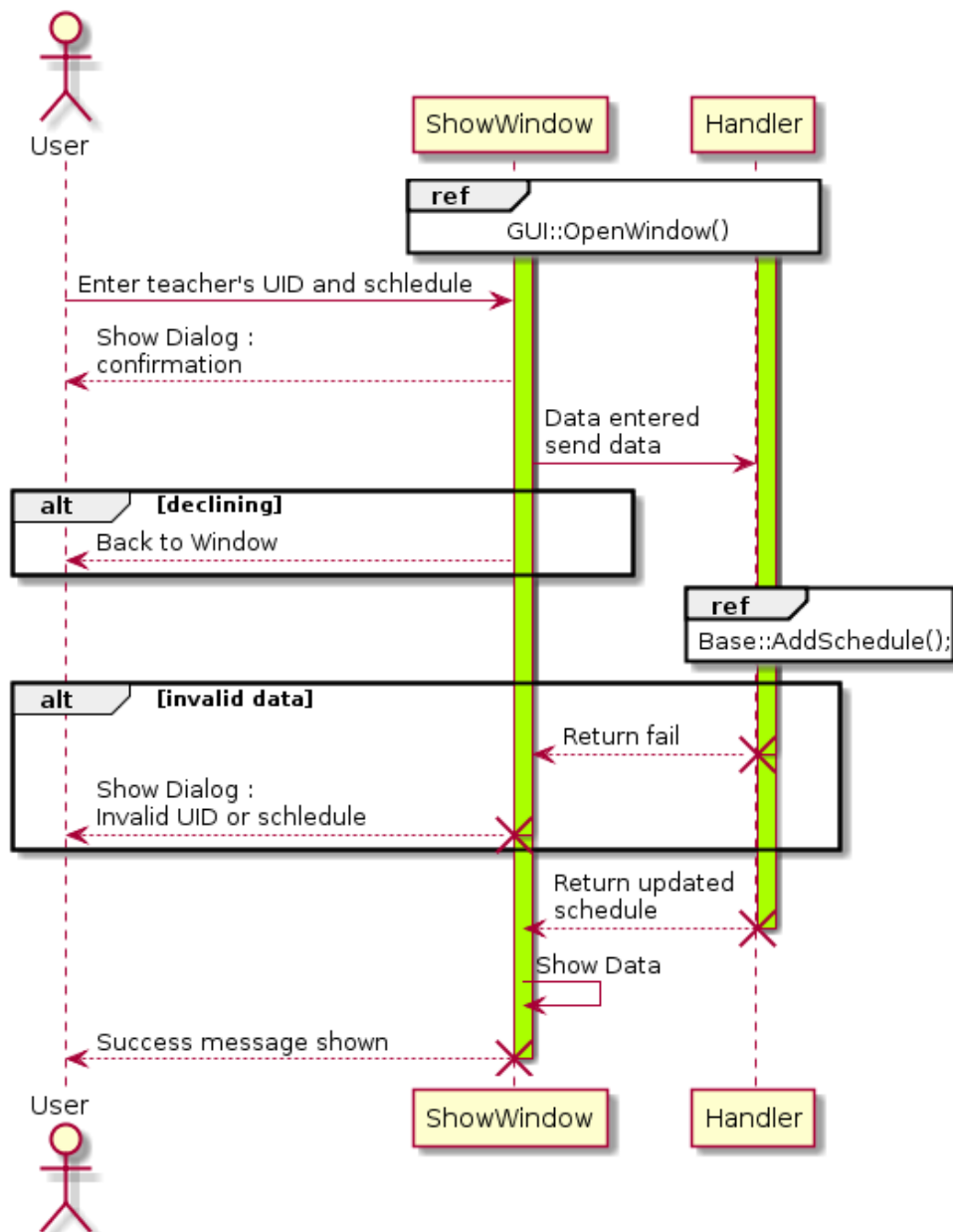
ref over Handler : Base::AddSchedule();

alt invalid data

Handler --> ShowWindow : Return fail
 destroy Handler
 ShowWindow --> User : Show Dialog : \nInvalid UID or schledule
 destroy ShowWindow
 end

Handler --> ShowWindow : Return updated\nschedule
 destroy Handler
 ShowWindow -> ShowWindow : Show Data
 ShowWindow --> User : Success message shown
 destroy ShowWindow
 @enduml

Діаграма у PlantUML:



Зміна аудиторії

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію зміни аудиторії

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та обирає форму «Зміна аудиторії»
2. Користувач вводить нову аудиторію та UID викладача
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Зміна розкладу)
6. Handler повертає до Window оновлений розклад
7. Window відображає оновлений розклад в полі Label та виводить повідомлення про успішне завершення операції

Альтернативний сценарій:

- 4.1 UID або аудиторія не валідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введений UID або аудиторія»

Код у PlantUML:

```
@startuml
actor User
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
activate Handler
activate Handler #aaff00
User -> ShowWindow : Choose "Change auditory"
User -> ShowWindow : Enter new auditory, UID

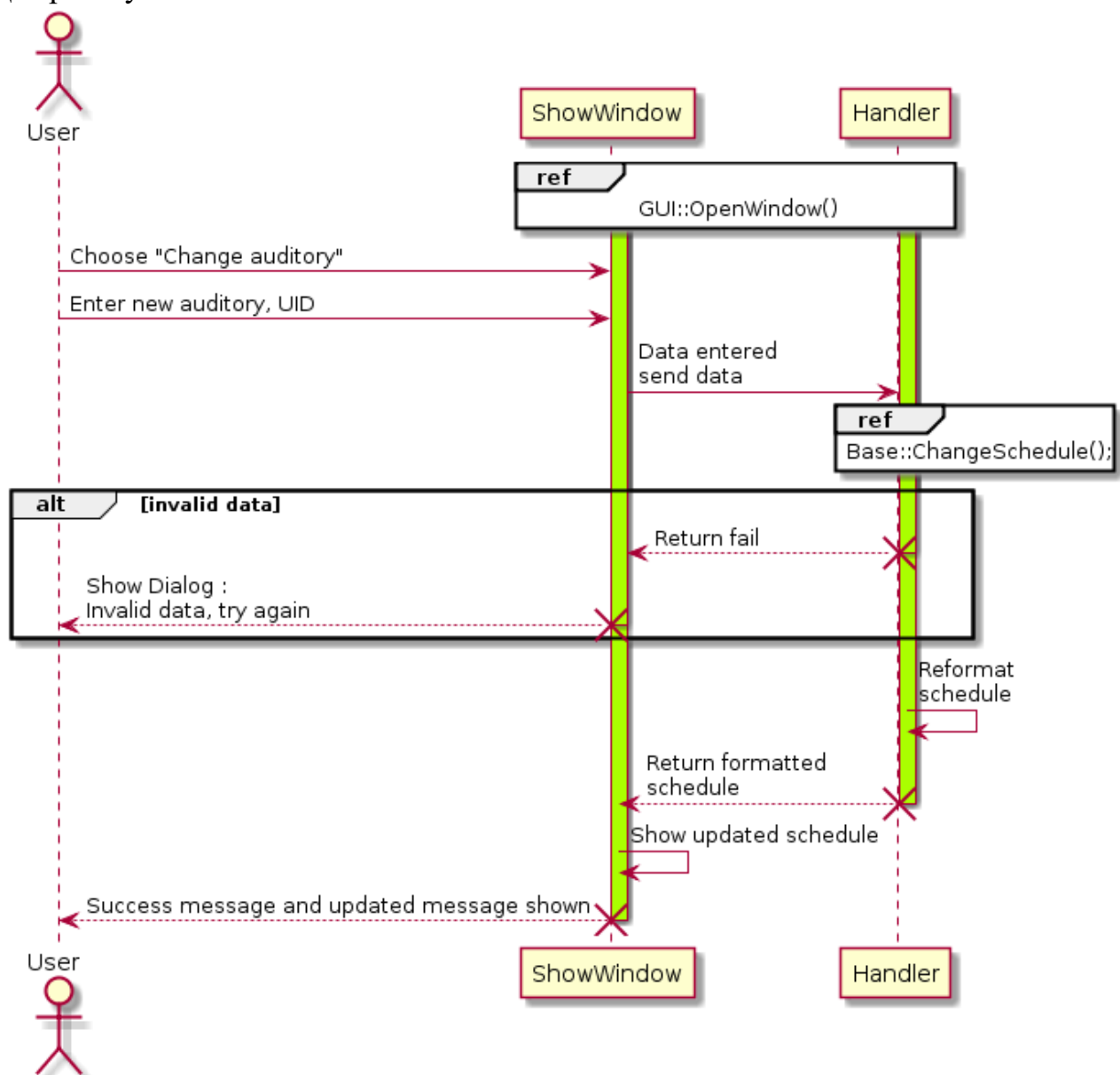
ShowWindow -> Handler : Data entered\nsend data
ref over Handler : Base::ChangeSchedule();
alt invalid data
Handler --> ShowWindow : Return fail
destroy Handler
```

```

ShowWindow --> User : Show Dialog :\nInvalid data, try again
destroy ShowWindow
end
Handler -> Handler : Reformat\nschedule
Handler --> ShowWindow : Return formatted\nschedule
destroy Handler
ShowWindow -> ShowWindow : Show updated schedule
ShowWindow --> User : Success message and updated message shown
destroy ShowWindow
@enduml

```

Діаграма у PlantUML:



Видалити викладача

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію видалення викладача

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та вводить UID, Ім'я, Прізвище викладача викладача, якого треба вилучити
2. Window показує діалогове вікно для підтвердження операції
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Видалення викладача)
6. Handler повертає до Window повідомлення про успішну операцію
7. Window відображає повідомлення про успішну операцію в полі Label

Альтернативний сценарій:

- 2.1 Користувач відхилив підтвердження операції
- 2.2 Повернути користувача на початкове вікно
- 4.1 UID, ім'я та прізвище не валідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введений UID або Ім'я»

Код у PlantUML:

```
@startuml
actor User
```

```
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
```

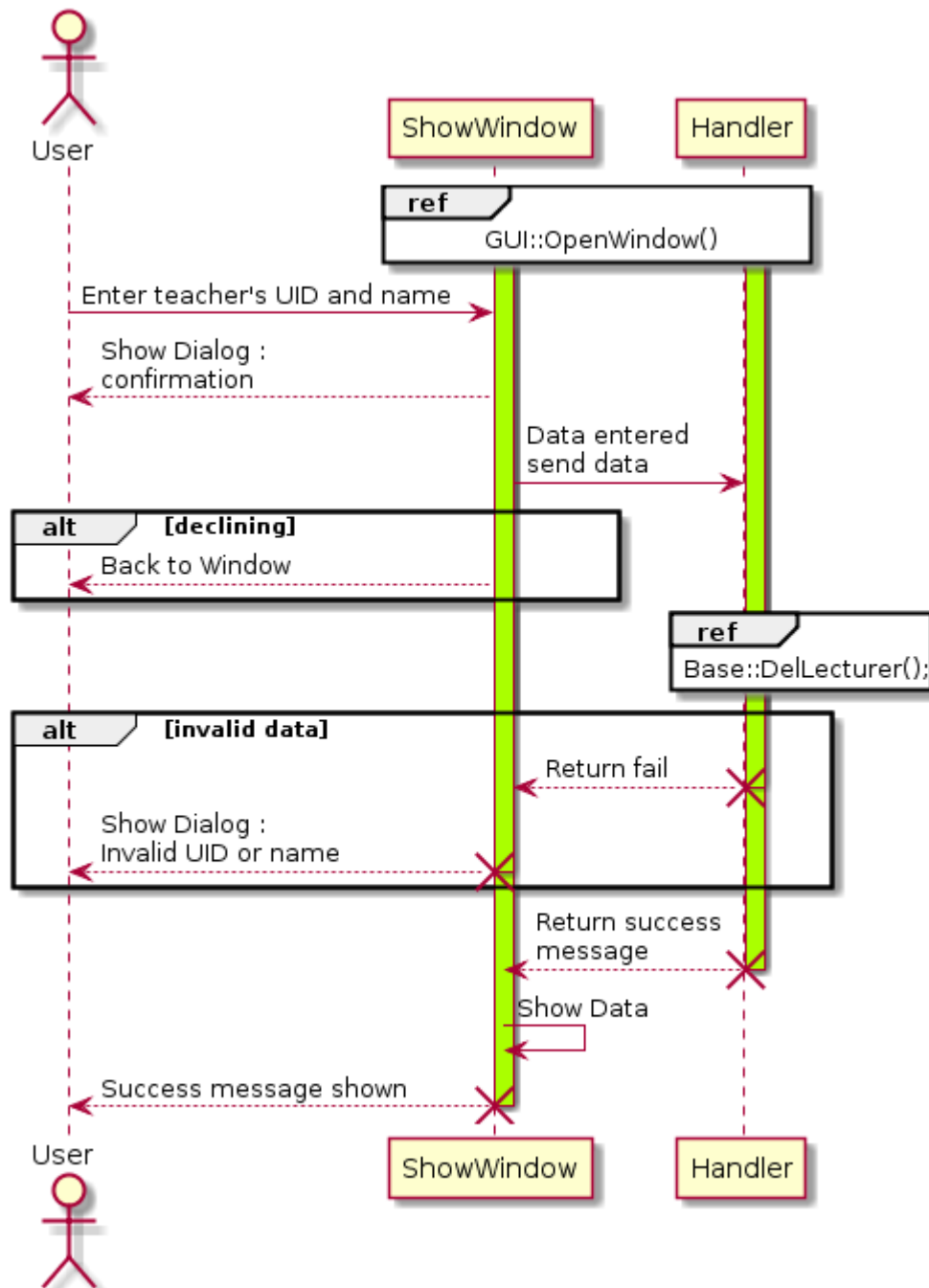
```
activate Handler
activate Handler #aaff00
```

```
User -> ShowWindow : Enter teacher's UID and name
ShowWindow --> User : Show Dialog : \nconfirmation
ShowWindow -> Handler : Data entered \nsend data
alt declining
ShowWindow --> User: Back to Window
end
```

```
ref over Handler : Base::DelLecturer();
alt invalid data
Handler --> ShowWindow : Return fail
destroy Handler
ShowWindow --> User : Show Dialog :\nInvalid UID or name
destroy ShowWindow
end

Handler --> ShowWindow : Return success\nmessage
destroy Handler
ShowWindow -> ShowWindow : Show Data
ShowWindow --> User : Success message shown
destroy ShowWindow
@enduml
```

Діаграма у PlantUML:



Додати розклад

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію додавання розкладу

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та вибирає графу «Додати розклад»

2. Користувач вводить розклад у виді таблиці з 3х колонок – Пара, Група, Номер пари
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Додати розклад)
6. Handler повертає до Window повідомлення про успішну операцію
7. Window відображає повідомлення про успішну операцію в полі Label

Альтернативний сценарій:

- 4.1 Дані з однієї або декількох колонок невалідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введені дані»

Код у PlantUML:

```
@startuml
actor User
```

```
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
```

```
activate Handler
activate Handler #aaff00
```

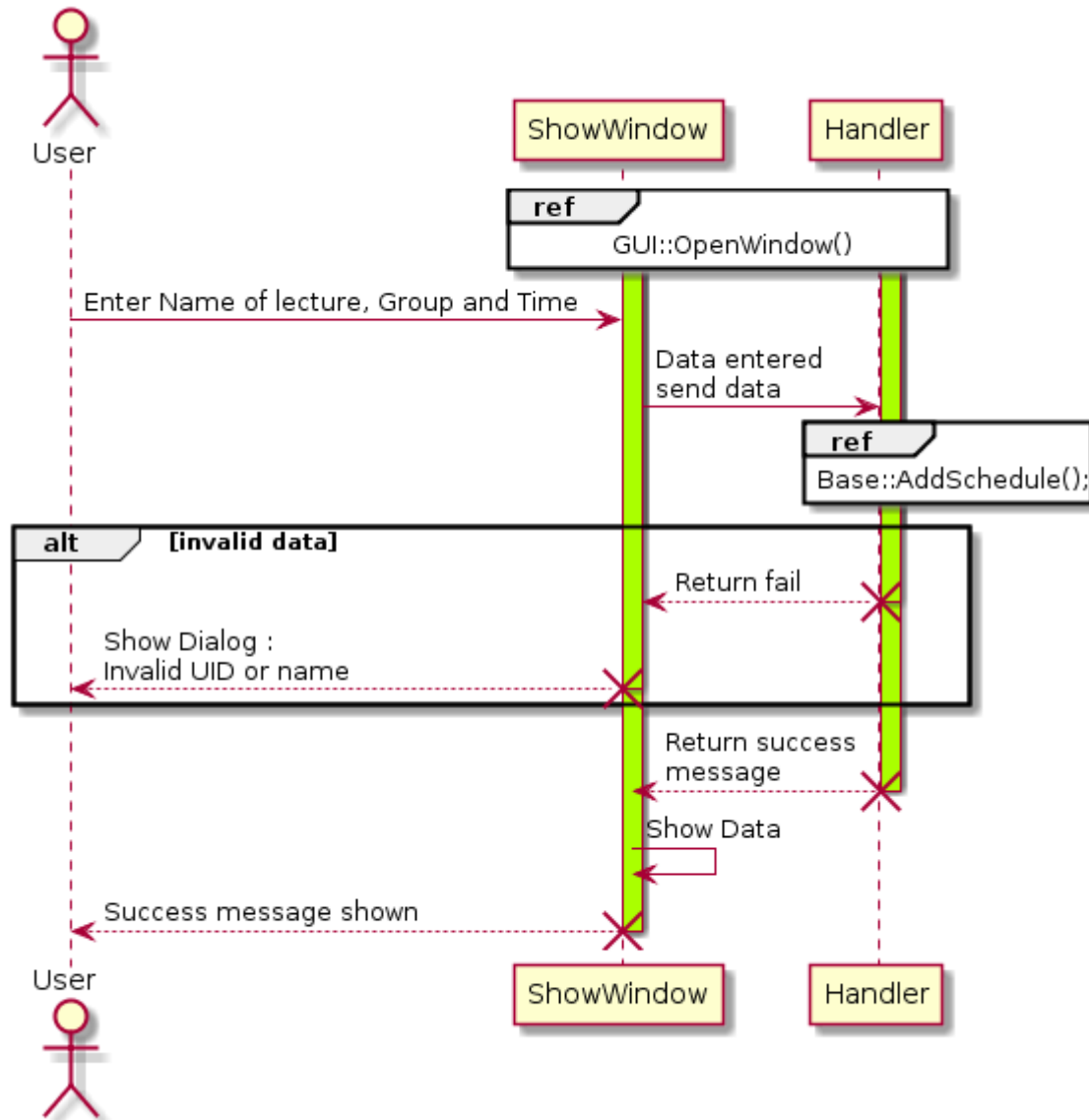
```
User -> ShowWindow : Enter Name of lecture, Group and Time
```

```
ShowWindow -> Handler : Data entered\nsend data
```

```
ref over Handler : Base::AddSchedule();
alt invalid data
Handler --> ShowWindow : Return fail
destroy Handler
ShowWindow --> User : Show Dialog :\nInvalid UID or name
destroy ShowWindow
end
```

```
Handler --> ShowWindow : Return success\nmessage
destroy Handler
ShowWindow -> ShowWindow : Show Data
ShowWindow --> User : Success message shown
destroy ShowWindow
@enduml
```

Діаграма у PlantUML:



Видалити розклад

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію видалення розкладу

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна) та вибирає графу «Видалити розклад»
2. Користувач вводить Ім'я, Прізвище викладача, розклад якого треба видалити
3. Window показує діалогове вікно для підтвердження операції
4. Window передає дані в Handler
5. Handler перевіряє дані на валідність
6. Handler передає дані в BaseLevel (юзкейс BaseLevel Видалити розклад)
7. Handler повертає до Window повідомлення про успішну операцію
8. Window відображає повідомлення про успішну операцію в полі Label

Альтернативний сценарій:

- 2.1 Користувач відхилив підтвердження операції
- 2.2 Повернути користувача на початкове вікно
- 4.1 Ім'я або прізвище викладача невалідні
- 4.2 Handler передає помилку в Window
- 4.3 Window виводить Label з текстом «Неправильно введене ім'я або прізвище»

Код у PlantUML:

```
@startuml
actor User
```

```
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
```

```
activate Handler
activate Handler #aaff00
```

```
User -> ShowWindow : Enter teacher's name and surname
ShowWindow --> User : Show Dialog : \nconfirmation
ShowWindow -> Handler : Data entered \nsend data
alt declining
ShowWindow --> User: Back to Window
end
ref over Handler : Base::DelSchedule();
alt invalid data
Handler --> ShowWindow : Return fail
```


destroy Handler

ShowWindow --> User : Show Dialog : \nInvalid name or surname

destroy ShowWindow

end

Handler --> ShowWindow : Return success\nmessage

destroy Handler

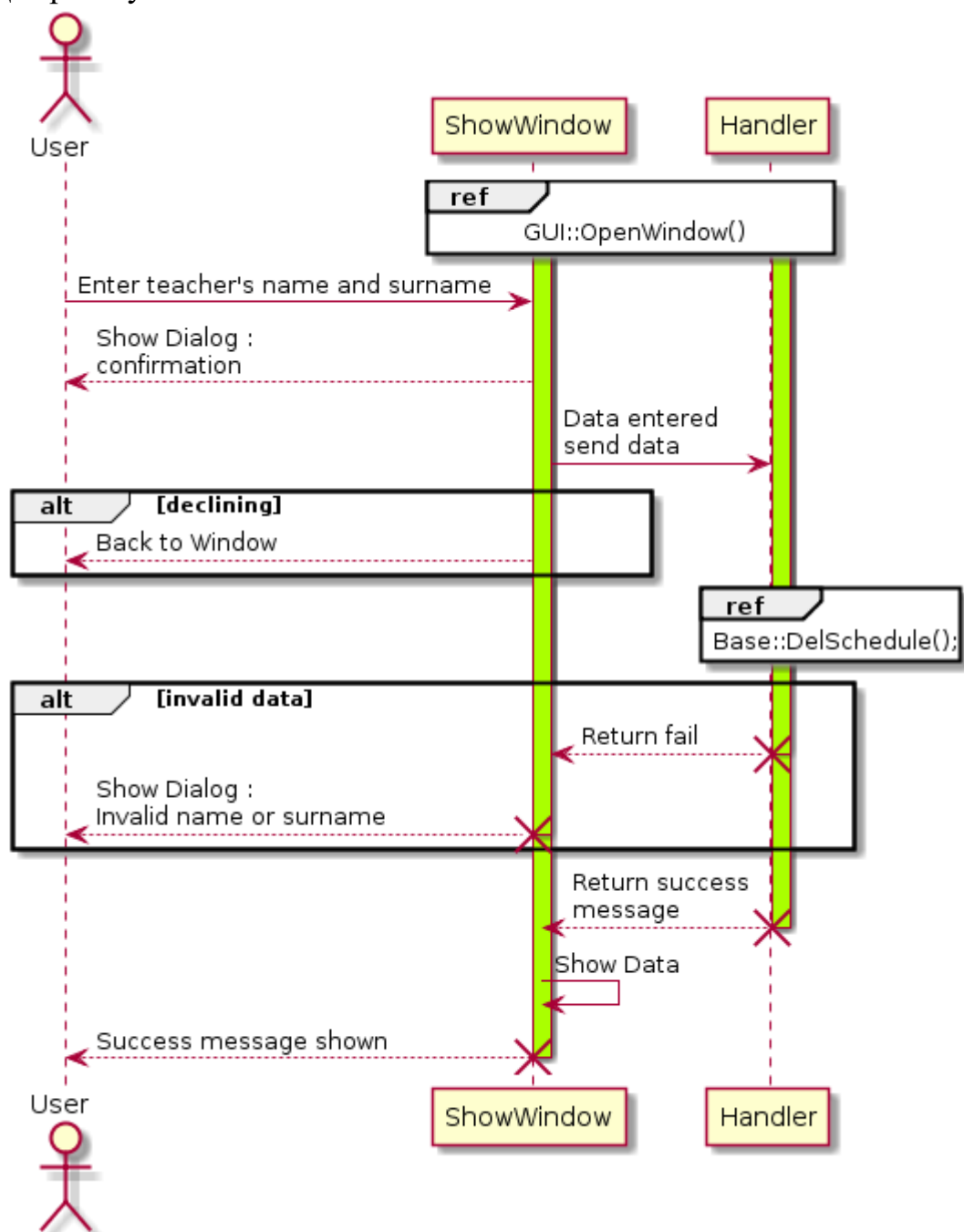
ShowWindow -> ShowWindow : Show Data

ShowWindow --> User : Success message shown

destroy ShowWindow

@enduml

Діаграма у PlantUML:



Надати обладнання

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію надання обладнання

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна)
2. Користувач вводить список вилученого обладнання у відповідну таблицю
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Надати обладнання)
6. Handler повертає до Window повідомлення про успішну операцію
7. Window відображає повідомлення про успішну операцію в полі Label

Альтернативний сценарій:

- 3.1 Формат введенного обладнання невалідний
- 3.2 Handler передає помилку в Window
- 3.3 Window виводить Label з текстом «Неправильно введенне обладнання»

Код у PlantUML:

```
@startuml
actor User
```

```
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
```

```
activate Handler
activate Handler #aaff00
```

```
User -> ShowWindow : Enter equipment list
ShowWindow -> Handler: Data entered\nsend data
```

```
ref over Handler : Base::AddEquipment();
```

```

alt invalid data
Handler --> ShowWindow : Return fail
destroy Handler
ShowWindow --> User : Show Dialog :\nInvalid equipment format
destroy ShowWindow
end

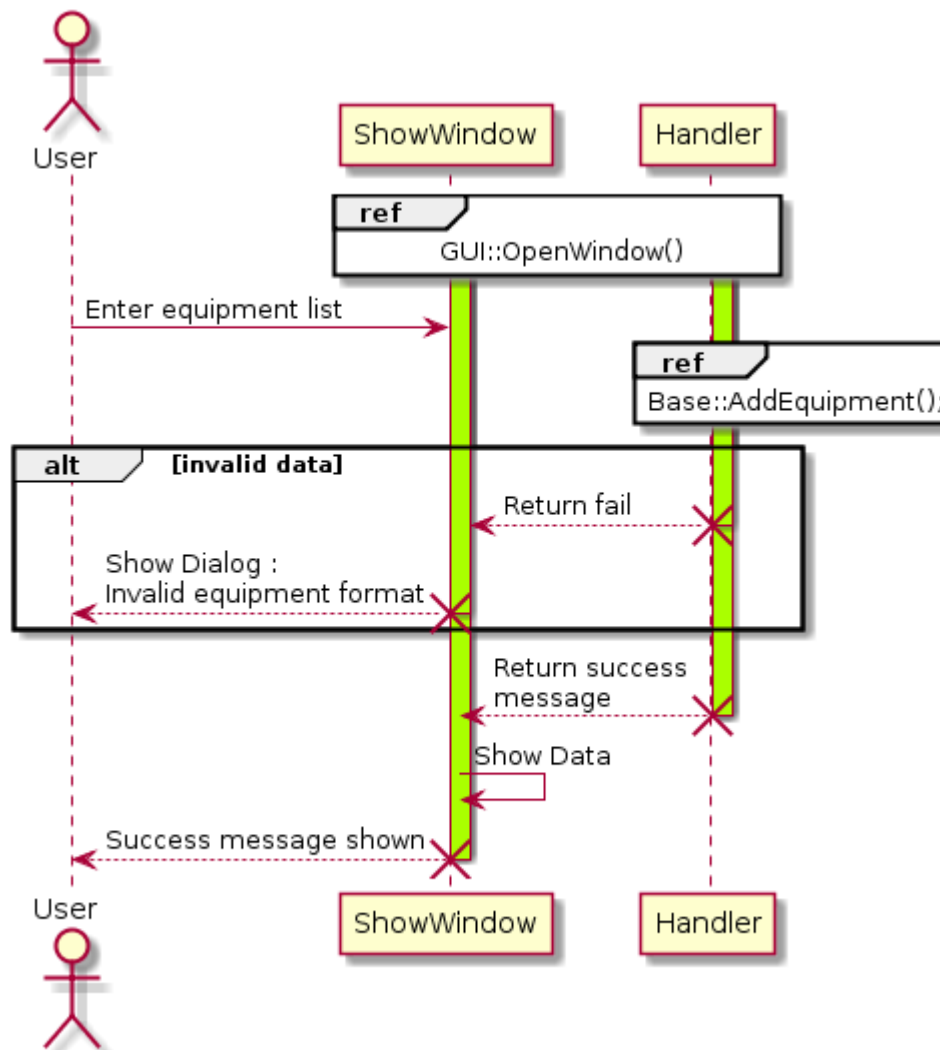
```

```

Handler --> ShowWindow : Return success\nmessage
destroy Handler
ShowWindow -> ShowWindow : Show Data
ShowWindow --> User : Success message shown
destroy ShowWindow
@enduml

```

Діаграма y PlantUML:



Вилучити обладнання

Діючі особи:

1. Користувач
2. Window
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуалізацію вилучення обладнання

Основний сценарій:

1. Користувач відкриває вікно(юзкейс GUI Відкриття вікна)
2. Користувач вводить список отриманого обладнання у відповідну таблицю
3. Window передає дані в Handler
4. Handler перевіряє дані на валідність
5. Handler передає дані в BaseLevel (юзкейс BaseLevel Вилучити обладнання)
6. Handler повертає до Window повідомлення про успішну операцію
7. Window відображає повідомлення про успішну операцію в полі Label

Альтернативний сценарій:

- 3.1 Формат введеного обладнання невалідний
- 3.2 Handler передає помилку в Window
- 3.3 Window виводить Label з текстом «Неправильно введене обладнання»

Код у PlantUML:

```
@startuml
actor User
```

```
ref over ShowWindow, Handler: GUI::OpenWindow()
activate ShowWindow
activate ShowWindow #aaff00
```

```
activate Handler
activate Handler #aaff00
```

```
User -> ShowWindow : Enter equipment list
ShowWindow -> Handler: Data entered\nsend data
ref over Handler : Base::DelEquipment();
alt invalid data
```

```

Handler --> ShowWindow : Return fail
destroy Handler
ShowWindow --> User : Show Dialog : \nInvalid equipment format
destroy ShowWindow
end

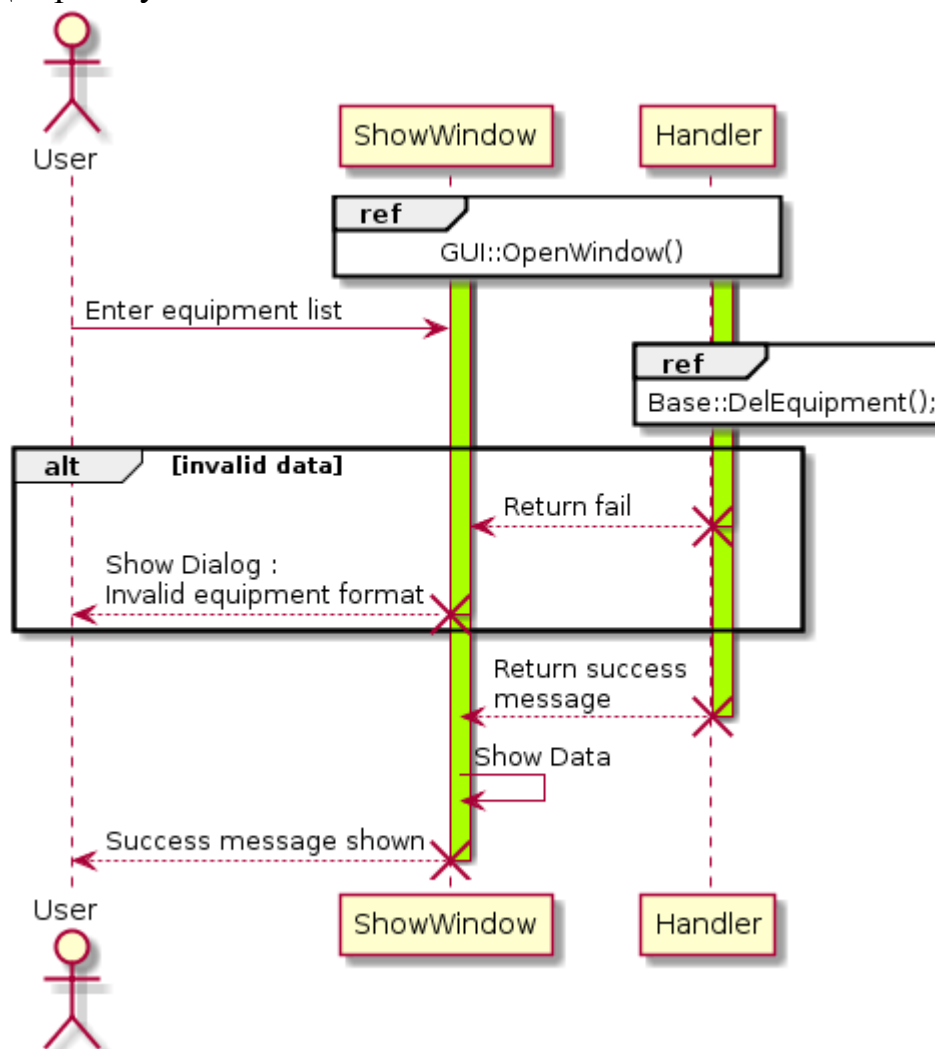
```

```

Handler --> ShowWindow : Return success\nmessage
destroy Handler
ShowWindow -> ShowWindow : Show Data
ShowWindow --> User : Success message shown
destroy ShowWindow
@enduml

```

Діаграма у PlantUML:



Вивести на консоль повідомлення

Діючі особи:

1. Sender
2. Console
3. Handler
4. Підсистема BaseLevel

Мета: забезпечити візуального розуміння про успішність операції

Основний сценарій:

1. Console отримує дані (юзкейс BaseLevel отримати дані)
2. Console передає дані в Handler
3. РАА перевіряє дані на валідність
4. Handler передає сигнал про успішність в Console
5. Console відображає повідомлення "Access allowed"

Альтернативний сценарій:

- 3.1 Дані не валідні
- 3.2 Handler передає сигнал про помилку в Console
- 3.3 Console відображає повідомлення "Access denied"

Код у PlantUML:

@startuml

actor Sender

participant Console

participant Handler

activate Console

activate Console#aaff00

activate Handler

activate Handler #aaff00

ref over Sender, Console: Base::GetData()

Console -> Handler: Send data

ref over Handler : Base::Validate();

alt data is valide:

Handler --> Console: Return success\signal

destroy Handler

Console -> Console: Show "access allowed"

Console --> Sender : Success message shown

```

    destroy Console
else data isn't valide:
    Handler --> Console: Return failure\nsignal
    destroy Handler
    Console -> Console: Show "access denied"
    Console --> Sender : Failure message shown
    destroy Console
end
@enduml

```

Діаграма у PlantUML:

