

Лабораторна робота №2

Тема. Behavioral UML.

Мета. Ознайомитися з поведінковими діаграмами UML, засвоїти їх призначення та особливості застосування для моделювання динамічних аспектів програмних систем. Набути практичних навичок побудови діаграм випадків використання, діаграм послідовностей, діаграм діяльності та діаграм станів для опису сценаріїв взаємодії користувача з системою та відображення внутрішніх процесів.

ЗАВДАННЯ

ЗАВДАННЯ 1. Розробити діаграму прецедентів (Онлайн-платформа для організації клубів за інтересами).

ЗАВДАННЯ 2. Statechart Diagram (Двері з доводчиком).

ЗАВДАННЯ 3. Statechart Diagram (Підписка).

ЗАВДАННЯ 4. Activity Diagram (Автоматичне закривання дверей з доводчиком).

ЗАВДАННЯ 5. Activity Diagram (Автоматичне створення підписки)

ЗАВДАННЯ 6. Sequence Diagram (Людина – Двері – Доводчик).

ЗАВДАННЯ 7. Sequence Diagram (Автоматичне створення підписки).

ЗАВДАННЯ 8. Collaboration Diagram (Людина – Двері - Доводчик)

ЗАВДАННЯ 9. Collaboration Diagram (Автоматичне створення підписки)

Хід роботи:

Завдання 1.

Межі системи:

Система реалізує онлайн платформу для клубів за інтересами, вона надає функціонал пошуку, перегляду клубів, вступ до них, можливість власного створення клубу. Система не надає вбудовані інтереси користувачі (наприклад можливість грати в шахи в самій системі), просто є можливість подій, які

					ЛР.ОК24.ПІ231.03.02			
Змін.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Башук О. Ю.			Behavioral UML.	Лім.	Арк.	Аркушів
Перевір.		Жереб Д. В.				Н	1	18
						ХПК		
Н. Контр.								
Затверд.								

попереджають про можливість гри, але в іншій програмі чи в житті.

Намальовано рамки системи разом із назвою:

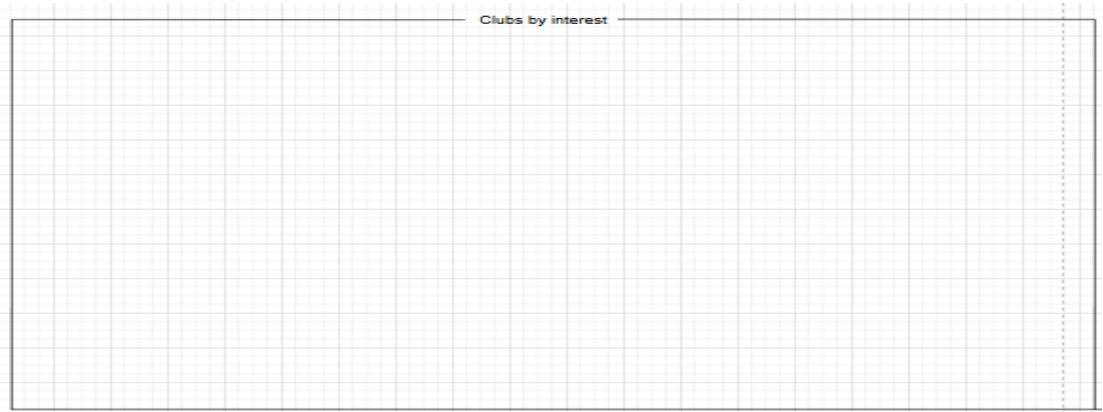


Рисунок 1 – Межі системи із назвою

Визначення акторів:

Визначено акторів системи, діляться вони на реальних користувачів – Гість (мінімальний функціонал), Користувач (базовий функціонал), Власник клубу(розширений користувацький функціонал), Адміністратор(максимальний контроль), системи – Соціальні мережі (реєстрація та взаємодія із користувачами), Система сповіщень (повідомлення подій).

Додано акторів до системи:

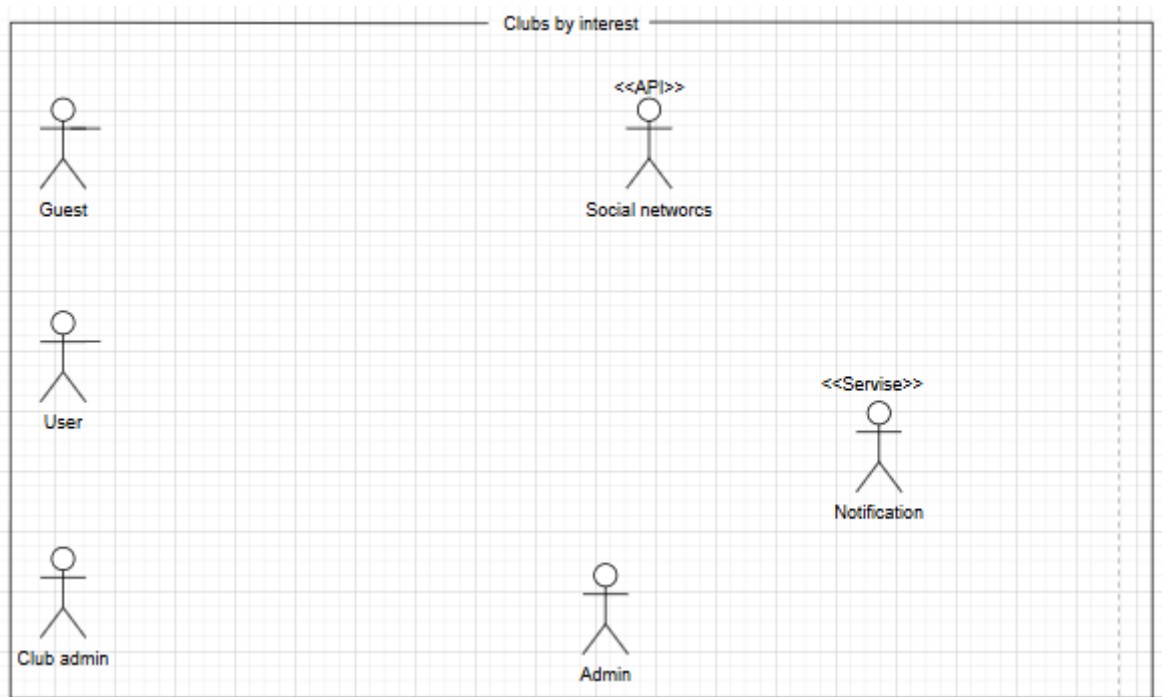


Рисунок 2 – Актори в системі

Виділення прецедентів:

Виділено основні функції системи:

- Sign Up
- Sign In
- Search for clubs
- Join the club
- Leave the club
- Club review
- Edit profile
- Check out famous clubs
- Participation in club events
- View club events
- Add friends
- Add social networks account
- Add friends
- Chating
- Create club
- Delete club
- Edit club
- Notifi participans
- Club moderation
- Create event
- Send an alert
- Manage users
- Block clubs

Додано прецеденти на діаграму:

					ЛР.ОК24.ПІ231.03.02	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

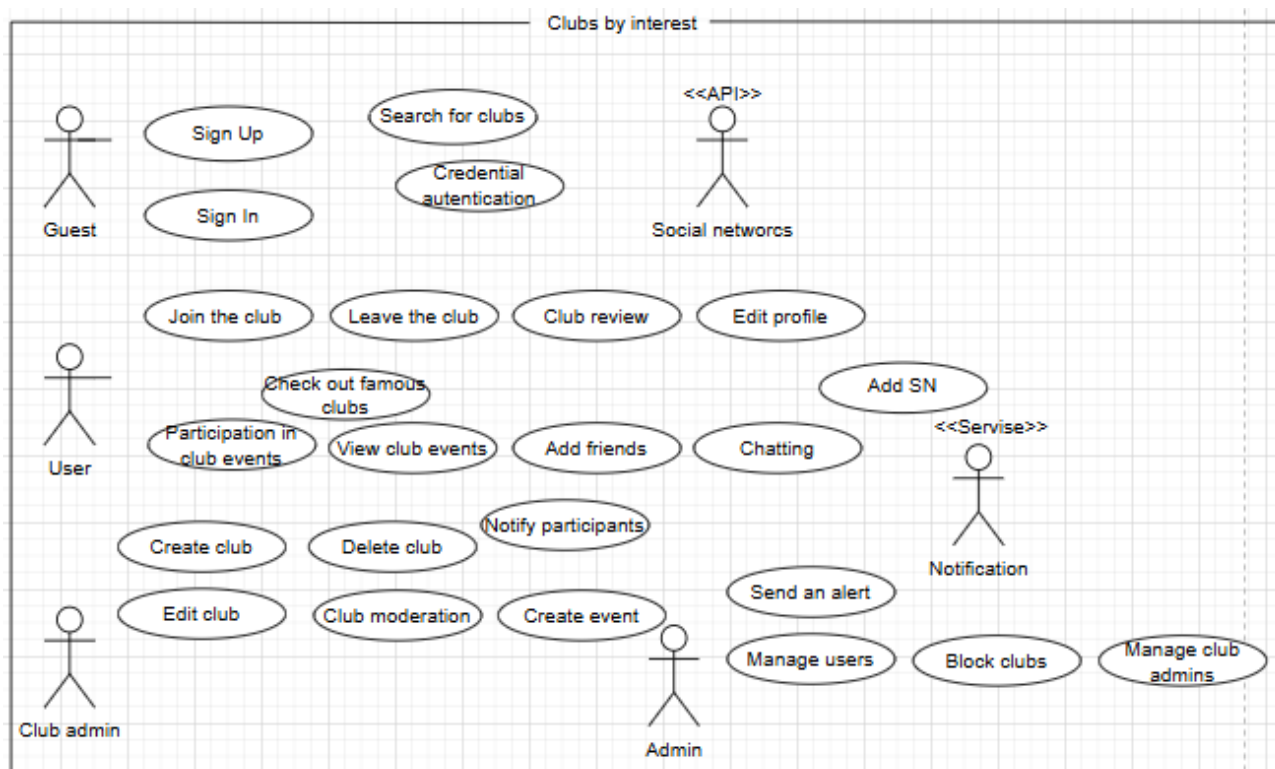


Рисунок 3 – Додано прецеденти

Моделювання зв'язків:

Generalization – Guest базовий актор, має найменший функціонал, який розширює User, який вже увійшов у систему, коли User створює кімнату він стає Club admin, який має можливості керувати клубом, має контроль над усіма користувачами це Admin

Association – Guest має такі можливості: Sign Up, Sign In, Search for clubs, User має можливості Guest та: Join the club, Leave the club, Club review, Edit profile, Check out famous clubs, Participation in club events, View club events, Add friends, Add social networks account, Add friends, Chatting, Club admin має можливості User та: Delete club, Edit club, Notify participants, Club moderation, Create event, Admin має всі функціональні можливості попередніх та: Send an alert, Manage users, Block clubs.

Include – Sign in включає Credential information тому, що вхід включає інформацію користувача, Create event включає Notify participants, тобто при створенні події користувачі одразу отримують сповіщення про неї.

Extend - Check out famous clubs розширює Join the club, тобто при пошуку конкретного клубу можна переглянути відомі клуби, можливо вони

прийдуть до душі, Add SN розширює Edit profile, тобто користувач може додати власну соц. мережу за бажанням.

Текстові специфікації:

- Sign Up: Дозволяє новому користувачеві створити обліковий запис, надавши необхідні реєстраційні дані.
- Sign In: Дозволяє користувачеві увійти до свого облікового запису за допомогою облікових даних.
- Search for clubs: Надає можливість користувачеві знаходити клуби.
- Join the club: Дозволяє користувачеві стати учасником обраного клубу.
- Leave the: Дозволяє учаснику клубу покинути клуб.
- Club review: Надає можливість учаснику клубу залишити оцінку та відгук про клубі.
- Edit profile: Дозволяє користувачеві змінювати свою інформацію та інші налаштування.
- Check out famous clubs: Дозволяє користувачеві переглянути список рекомендованих або найпопулярніших клубів.
- Participation in club events: Дозволяє члену клубу реєструватися та брати участь у заходах, які організовує клуб.
- View club events: Надає користувачам можливість переглядати список подій, організованих клубом.
- Add friends: Дозволяє користувачеві надсилати запити на дружбу іншим користувачам для спілкування.
- Add social networks account: Дозволяє користувачеві додати до свого профілю соціальні мережі.
- Chatting: Надає користувачам можливість обмінюватися повідомленнями.
- Create club: Дозволяє авторизованому користувачеві створити новий клуб, визначивши його назву, опис, правила та категорію.

					ЛР.ОК24.ПІ231.03.02	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

- Delete club: Дозволяє адміністратору клубу видалити свій клуб.
- Edit club: Дозволяє адміністратору клубу змінювати основну інформацію про клуб.
- Notify participants: Система автоматично інформує учасників клубу про створення нової події.
- Club moderation: Дозволяє адміністратору клубу керувати контентом та учасниками.
- Create event: Дозволяє адміністратору клубу організовувати події для учасників свого клубу.
- Send an alert: Надає адміністратору системи можливість розсилати важливі системні повідомлення усім або обраній групі користувачів.
- Manage users: Дозволяє адміністратору системи керувати обліковими записами користувачів, блокувати, редагувати або видаляти їх.
- Block clubs: Надає адміністратору системи право блокувати діяльність клубів, які порушують правила.
- Manage club admins: Дозволяє власнику клубу призначати або позбавляти прав адміністратора інших учасників свого клубу.

Створено кінцевий вигляд діаграм прецедентів:

					ЛР.ОК24.ПІ231.03.02	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

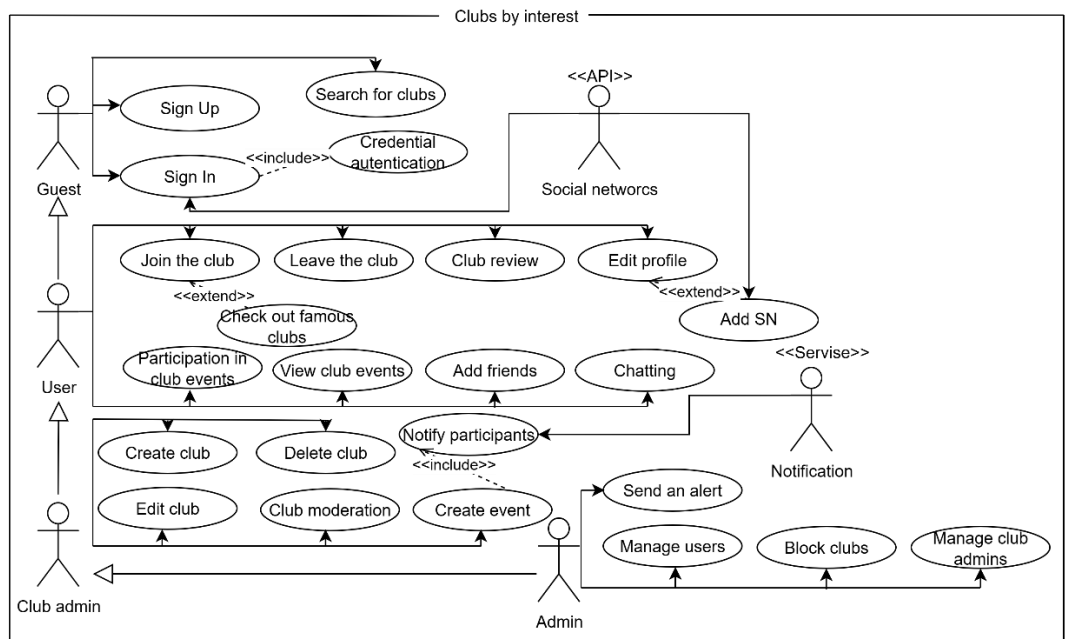


Рисунок 4 – Готова діаграма

Завдання 2.

Межі системи:

Діаграма моделює стани дверей із доводчиком, реакцію на дії людини включаючи дії доводчика.

Стани дверей на діаграмі:

Locked with a key – закриті на ключ, початковий стан дверей.

Unlocked – розблоковані, тобто не закриті на ключ.

Broken – зламані, ключ зламаний в замку.

Closed – стан, коли двері не закриті на ключ, тобто їх можна просто відкрити.

Opening – стан, який описує відкривання дверей.

Open – стан, коли двері повністю відкриті.

Recorded - стан, коли двері зафіксовані на доводчику.

Closing – стан, який описує процес закривання дверей за допомогою доводчика.

Blocked – стан, який виникає, коли на шляху до закриття виникла перешкода.

Тригери.

Locked with a key – від цього стану є можливість переходу до двох станів, тригер The key broke in the door означає, що при закриванні дверей зламався ключ та двері переходять до стану Broken, Opened with a key – цей тригер означає, що двері успішно відкрились та переходять до стану Unlocked. Після стану Unlocked двері автоматично переходить до стану Closed.

Closed – стан, який може через тригер Lock with a key стати Locked with a key, коли двері закривають на ключ, або перейти до стану Opening при тригеру, який означає початок відкривання дверей start opening після цього стану двері переходять до стану Open.

Open може перейти в стан Recorded через тригер Recorded closer – який означає, що доводчик зафіксували, щоб двері були в певному положенні, далі після цього стану за допомогою тригера Door closer removed from fixation можуть перейти в стан Closing, або одразу від стану Open в стан Closing за допомогою тригера The handle is released.

Closing може перейти в стан Blocked, коли на шляху зустрілась, якась перешкода Encounter on obstacle, коли перешкода зникає Obstacle it seems двері переходять назад до стану Closing.

Коли закриванню нічого не заважає двері із стану Closing переходять до стану Closed звідки можуть продовжити цикл відкривання та закривання або перейти в стан Locked with a key.

Зроблено діаграму станів відповідно до опису:

					ЛР.ОК24.ПІ231.03.02	Арк.
						8
Змін.	Арк.	№ докум.	Підпис	Дата		

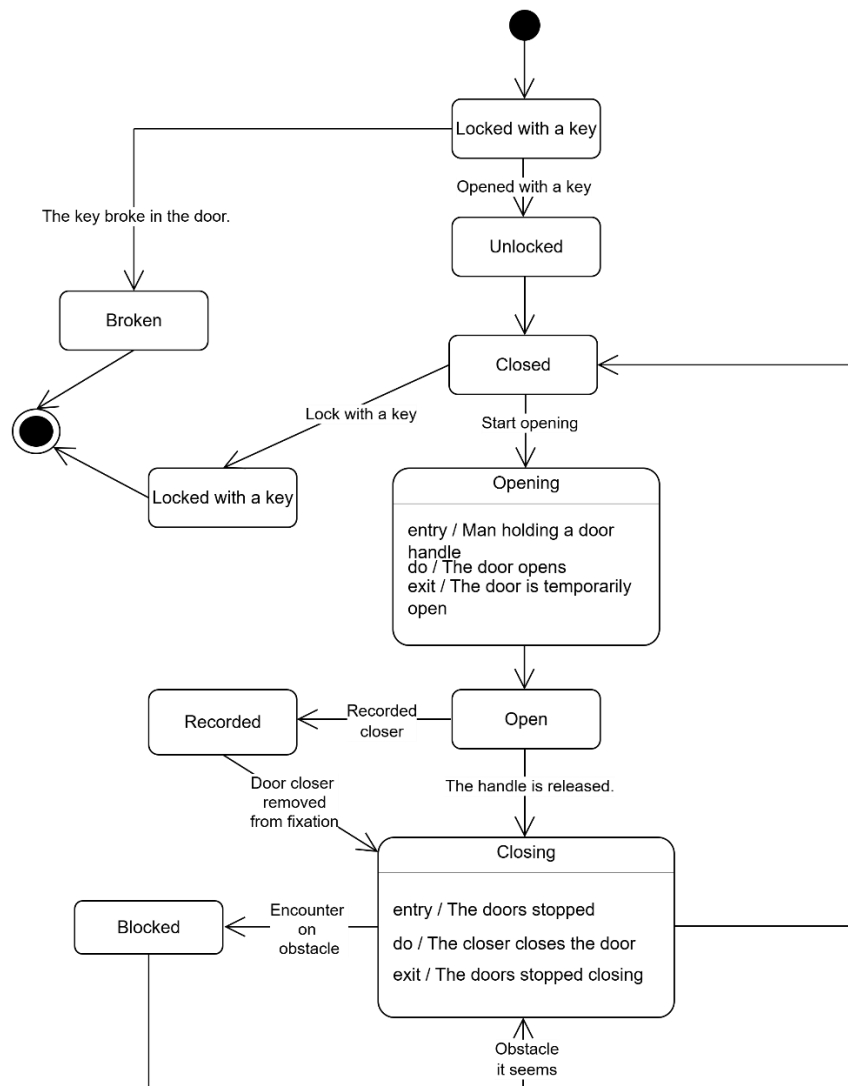


Рисунок 5 – Готова діаграма станів

Завдання 3.

Межі системи:

Діаграма моделює стани через, які проходить підписка користувача.

Стани підписки на діаграмі:

Initialization – початковий стан, де відбувається створення підписки вручну. Система отримує та перевіряє дані користувача.

Integration with the service – стан, що відображає процес інтеграції з зовнішнім сервісом як розгалуження в створенні підписки.

Error integration – стан помилки, в який підписка переходить у разі помилки під час інтеграції.

Created – стан, у якому підписка успішно створена, але ще не активна, тобто очікує дату настання підписки

Actives – композитний стан, який означає, що підписка є активною. Він містить підстани:

- Paid – підписка оплачена на поточний період.
- Awaiting confirmation – система очікує підтвердження оплати за наступний період. Цей стан активується при наближенні дати закінчення періоду оплати.

Overdue – стан, що означає протермінування платежу. Підписка переходить у цей стан, якщо оплату не було підтверджено вчасно.

Cancelled – підписку скасовано.

Archived– стан, до якого переходить підписка для зберігання інформації про неї.

Тригери.

В залежності від тригера, може бути два початкові стани: тригер User has recovered integration – переводить підписку в стан Integration with the service, якщо інтеграція пройшла неуспішно – тригер Error, підписка переходить у стан Error integration та завершує життєвий цикл. Якщо початковим тригером був User selected creation manually підписка переходить в стан Initialization, обидва успішні стани створення підписки переходять в стан Created.

Created може перейти в стан Cancelled при відміні користувачем підписки або в композитний стан Actives при тригері Subscription start date.

Actives є композитним станом, який містить початковий стан Paid, який при настанні тригеру Expiration date переходить в стан Awaiting confirmation.

Пр виникненні тригеру Payment not confirmed підписка переходить в стан Overdue, з якого є два шляхи: Confirmed payment і підписка переходить назад до стану Actives або Not confirmed payment і підписка переходить до стану Cancelled.

					ЛР.ОК24.ПІ231.03.02	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		10

Cancelled при тригері Accepted archiving переходить до стану Archived та завершає свій життєвий або одразу переходить до закінчення без цього тригера.

Зролено відповідну діаграму станів:

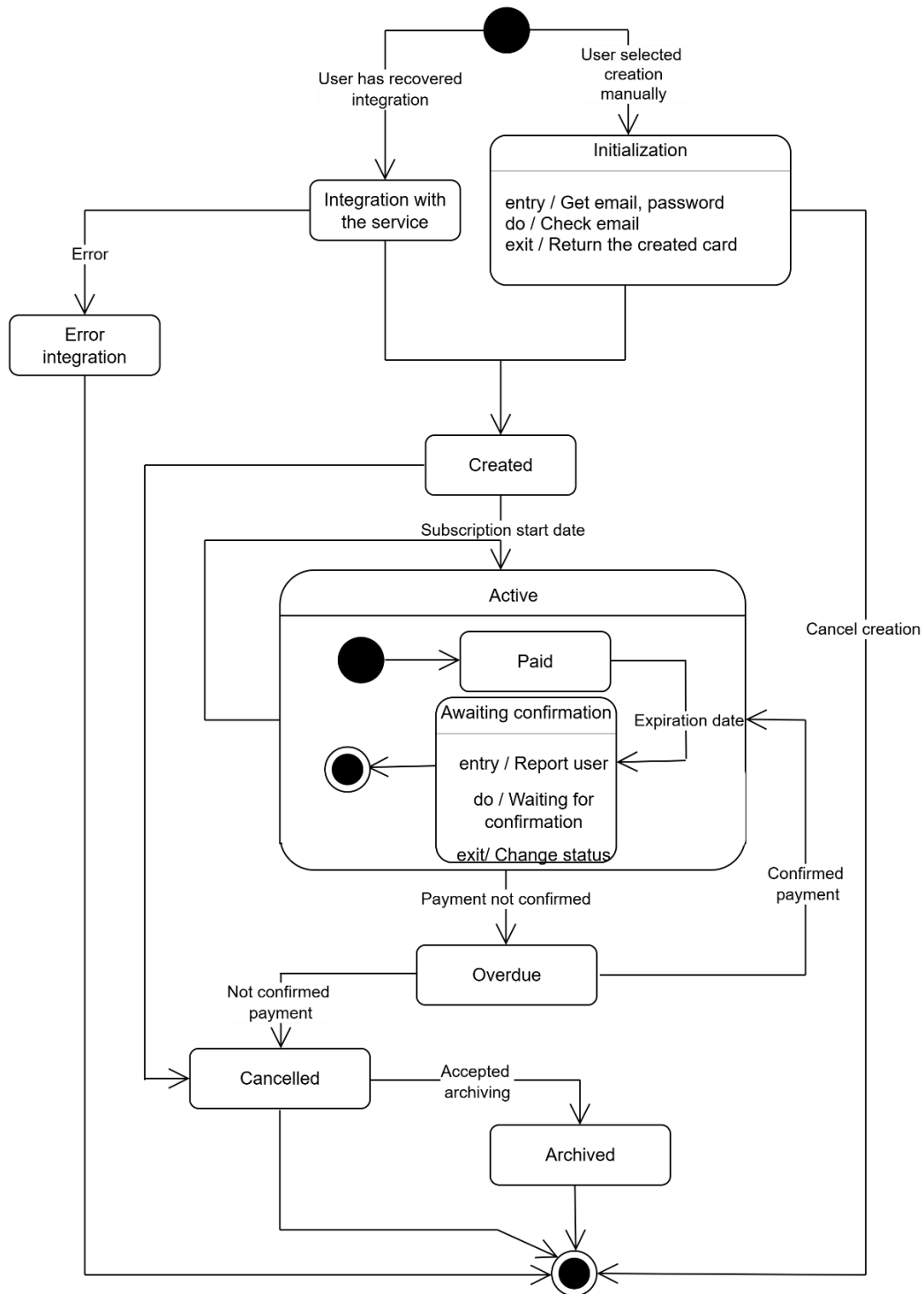


Рисунок 6 – Діаграма станів для підписки

Завдання 4.

Межі процесу.

Процес стартує при перевірці дверей на те чи вони замкнені на ключ та закінчується, коли людина йде та двері автоматично закриваються.

Swimlanes.

Виділено двох учасників системи People – людина, яка взаємодіє із дверима та Closer system – система, яка реагує на взаємодію людини.

Поудовано діаграму активностей.

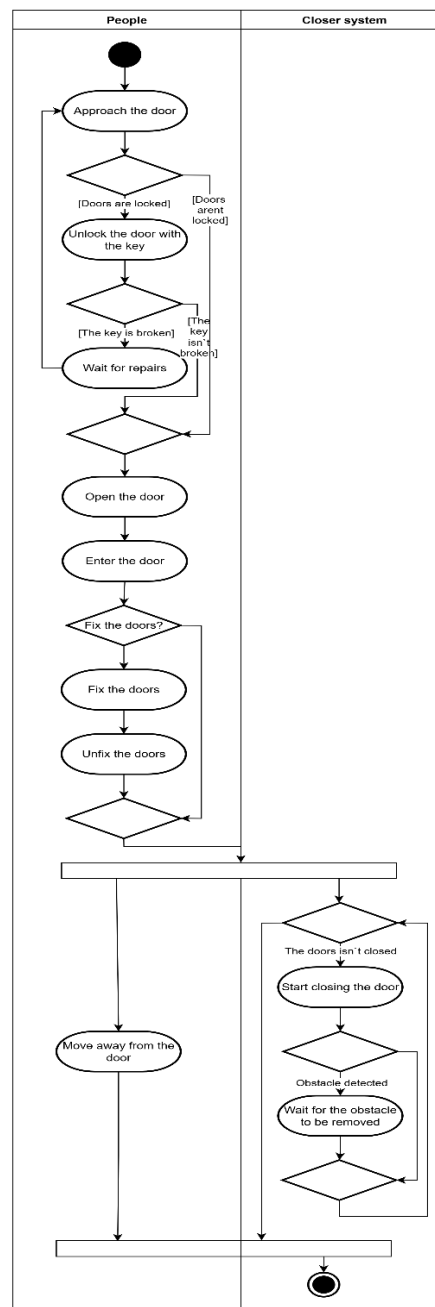


Рисунок 7 – Діграма активностей.

Завдання 5

Межі процесу.

Процес стартує при запуску застосунку.

Swimlanes.

Виділено двох учасників системи User – людина, яка взаємодіє із дверима та System – система, яка реагує на взаємодію людини.

Побудовано діаграму активностей.

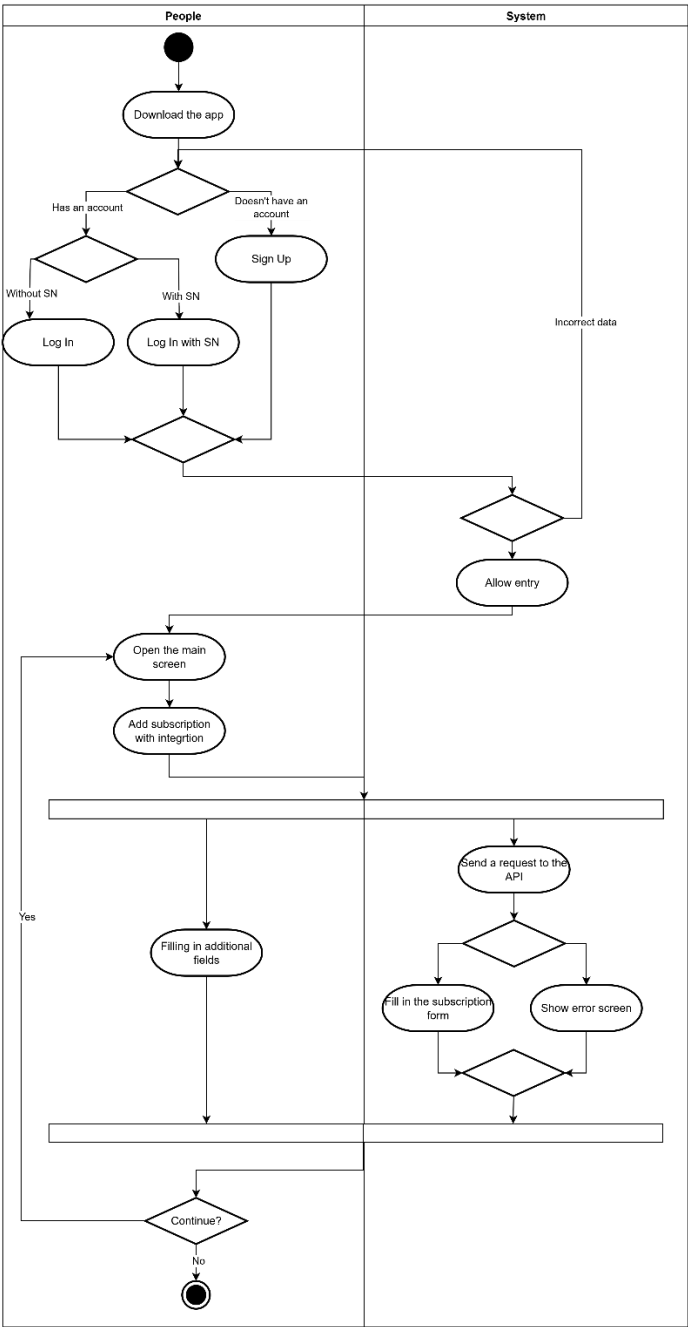


Рисунок 8 – Діаграма активностей для автоматчного додавання підписки

Завдання 6.

Визначення учасників.

Визначено трьох основних учасників: Людина – головний ініціатор дій, Двері – головний об’єкт, з яким взаємодіє користувач, доводчик – об’єкт, який реагує на дії користувача.

Послідовність дій.

Користувач перевіряє, чи двері відкриті.

Користувач відкриває двері.

Користувач відкриває двері – двері сигналізують доводчику про початок переходу в відкритий стан.

Користувач перестає тримати двері та йде, не очікуючи закривання – двері надсилають сигнал двоводчину на закривання.

Побудовано відповідну діаграму послідовностей.

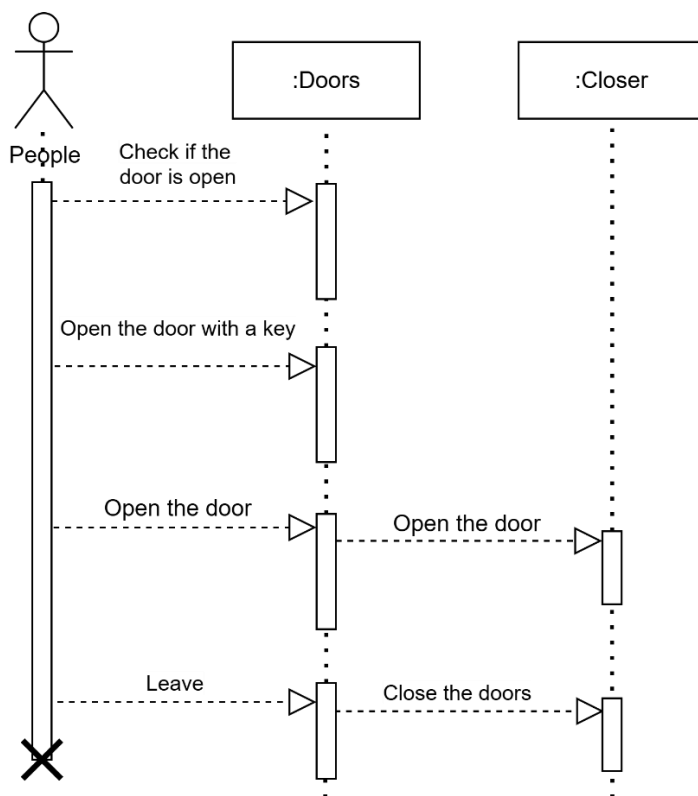


Рисунок 9 – Діаграма послідовностей відкривання та закривання дверей.

Завдання 7.

Визначення учасників.

					ЛР.ОК24.ПІ231.03.02	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		14

Визначено шість основних учасників: Користувач – головний ініціатор дій, Програмний інтерфейс – граничний об’єкт, який надає взаємодію із програмою, Контролер підписок – граничний об’єкт, який контролює роботу всіх підписок, Підписка – об’єкт, API системи підписок – граничний об’єкт, за допомогою якого автоматизується процес створення, БД – головне місце, де зберігаються підписки.

Послідовність дій.

Користувач надсилає запит на перегляд всіх підписок до Програмного інтерфейсу - Програмний інтерфейс надсилає цей запит до контролера - контролер - отримує записи із БД та повертає до Графічного інтерфейсу, а той в свою чергу відображає їх користувачеві.

Користувач починає створення автоматичної підписки до Програмного інтерфейсу – Програмний інтерфейс передає запит Контролеру – Контролер створює підписку, надсилає запит на отримання інформації до API та передає стан Графічному інтерфейсу – Графічний інтерфейс відображає стан користувачеві. API - повертає інформацію до Підписки – Підписка передається до Контролера та надсилає асинхронний запит до БД на збереження. Контролер – передає інформацію до Графічного інтерфейсу, а той в свою чергу повертає результат користувачеві.

Створено відповідну діаграму послідовностей.

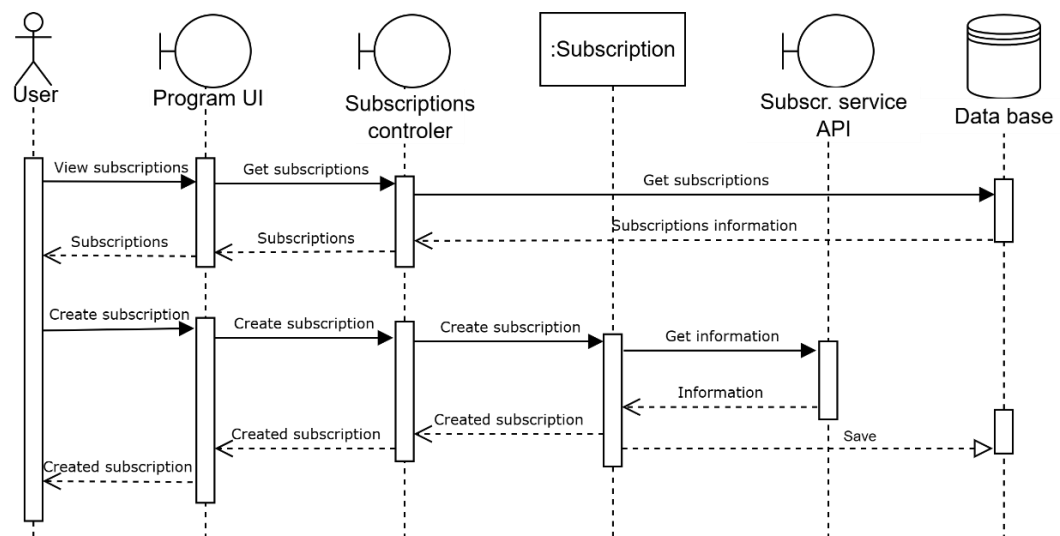


Рисунок 10 – Діаграма послідовностей для підписки

Завдання 8.

Визначення учасників.

Визначено трьох основні учасників: Людина – головний ініціатор дій, Двері – головний об’єкт, з яким взаємодіє користувач, доводчик – об’єкт, який реагує на дії користувача.

Позначення зв’язків.

Користувач комунікує з дверима (зв’язок асоціації), двері в свою чергу комунікують з доводчиком (зв’язок агрегації та композиції).

Послідовність повідомлень.

Користувач 1:Check if the doors are open.

Далі Користувач 2:Open the door with a key.

Наступним кроком Користувач 3:Opening the doors, в результат цієї взаємодії двері до доводчика 4:Go to opening state

Користувач до дверей 5:Leave the doors, двері до доводчика 6:Start closing.

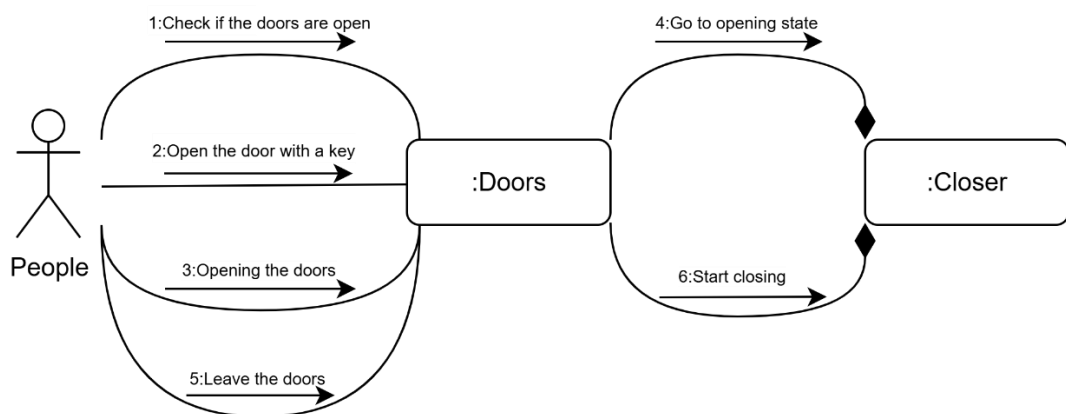


Рисунок 11 – Діаграма комунікації.

Завдання 9.

Визначення учасників.

Визначено шість основні учасників: Людина – головний ініціатор дій, ProgramUI – головний об’єкт, з яким взаємодіє користувач, Subscription

controller – об’єкт, який відповідає за виконання дій, які запитує користувач, Subscription – об’єкт, який вміщує в собі інформацію, Subscription service API – API для отримання інформації про підписку та DB – база даних для збереження інформації про підписки користувача

Позначення зв’язків.

Взаємодія між об’єктами відбувається за допомогою зв’язку асоціації та синхронний запитів, окрім відправлення інформації про підписку до БД.

Послідовність повідомлень.

User до ProgramUI: 1:View subscriptions, далі ProgramUI до Subscription controller: 2:Get subscriptions, Subscription controller до DB: 3:Get subscriptions, далі інформація повертається до користувача: DB до Subscription controller: 4:Subscriptions information, Subscription controller до ProgramUI: 5: Subscriptions, ProgramUI до User: 6: Subscriptions.

Створення підписки за допомогою інтеграції відбувається в такому порядку: User до ProgramUI: 7:Create subscription, ProgramUI до Subscription controller: 8:Create subscription, Subscription controller до Subscription: 9:Create subscription, Subscription до Subscription service API: 10:Get information, Subscription service API повертає Subscription: 11:Information, Subscription повертає інформацію до Subscription controller: 12:Create subscription та асинхронно до DB: 13:Save subscription. Subscription controller повертає ProgramUI: 14: Created subscription, ProgramUI повертає кінцевий результат до User: 15: Created subscription.

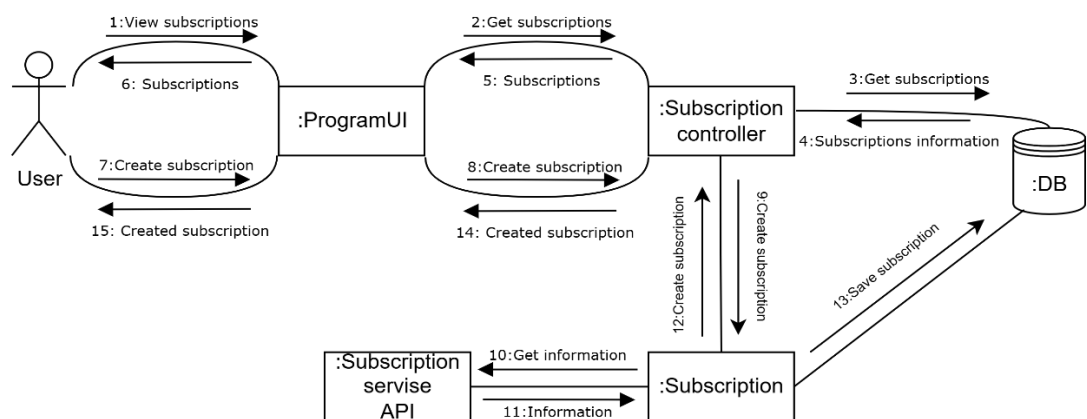


Рисунок 12 – Діаграма комунікації для підписки

Висновок.

Я ознайомився з поведінковими діаграмами UML, засвоєно їх призначення та особливості застосування для моделювання динамічних аспектів програмних систем. Я набув практичних навичок побудови діаграм випадків використання, діаграм послідовностей, діаграм діяльності та діаграм станів для опису сценаріїв взаємодії користувача з системою та відображення внутрішніх процесів

					ЛР.ОК24.ПІ231.03.02	Арк.
						18
Змін.	Арк.	№ докум.	Підпис	Дата		