

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №6

3 курсу

«Інженерія програмного забезпечення»

студента 2 курсу

групи ПП-22

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Шевлюк Вікторії Віталіївни

Перевірила:

Доц. Бойко Юлія Петрівна

Київ 2022

Тема: моделювання складних програмних систем

Мета роботи – дослідження процесів та набуття практичних навичок з розробки моделі складної програмної системи шляхом виконання конкретного проекту.

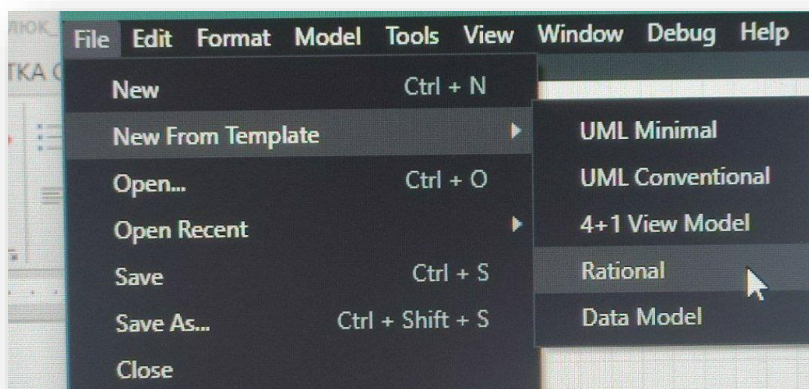
Завдання: Програмно реалізувати систему надання електронних послуг для перегляду можливих івентів та бронювання квитків на обраний івент.

Хід роботи:

У ході цієї лабораторної роботи ми створимо систему, завдяки якій можна буде переглядати івенти певного регіону різних категорій (театральні постанови, концерти, фестивалі, виставки тощо).

Для роботи я використовую програму StarUml.

Спочатку потрібно створити новий проект з темплейту Rational:



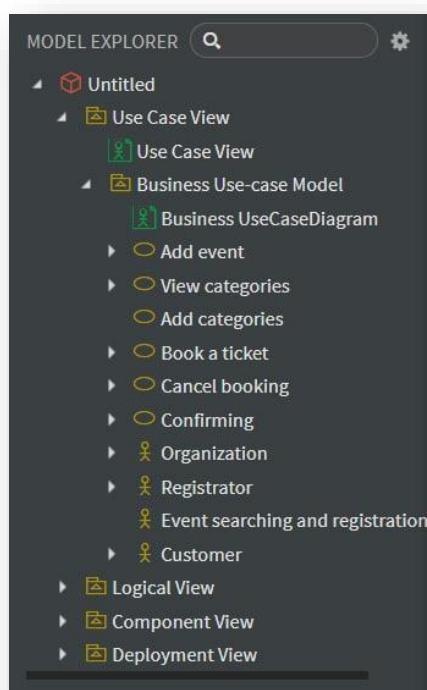
Тепер, коли у нас є новий пустий проект, почнемо зі створення дійових осіб і варіантів використання. Дійові особи для мого проекту – Користувач, Організація, Реєстратор (або адміністратор) , Система пошуку та реєстрації на івенти (ESR System).

Усіх їх можна додати за допомогою «Use Case View → Business Use-case Model» у браузері. Для цього клацаємо правою кнопкою миші на «Business Use-case Model» та обираємо пункт «New / Actor».

Вводимо ім'я дійової особи (внизу у вікні, де показані деталі щойно створеного елементу, вводимо назву у полі Name).

Тепер додамо варіанти використання. Клацнемо правою кнопкою миші на пакеті Business Use Case Model уявлення Use Case View в браузері. Обираємо в меню пункт New / Use Case. Новий варіант використання під назвою NewUseCase з'явиться в браузері. Зліва від нього буде видно піктограма варіанти використання UML у вигляді еліпса. Введемо назву аналогічним чином як для дійових осіб.

Отримуємо наступний список:



Далі побудуємо діаграму варіантів використання. Клацнемо правою кнопкою миші на пакеті Business Use Case Model уявлення Use Case View в браузері. З спливаючого меню оберемо пункт New / Use Case Diagram.

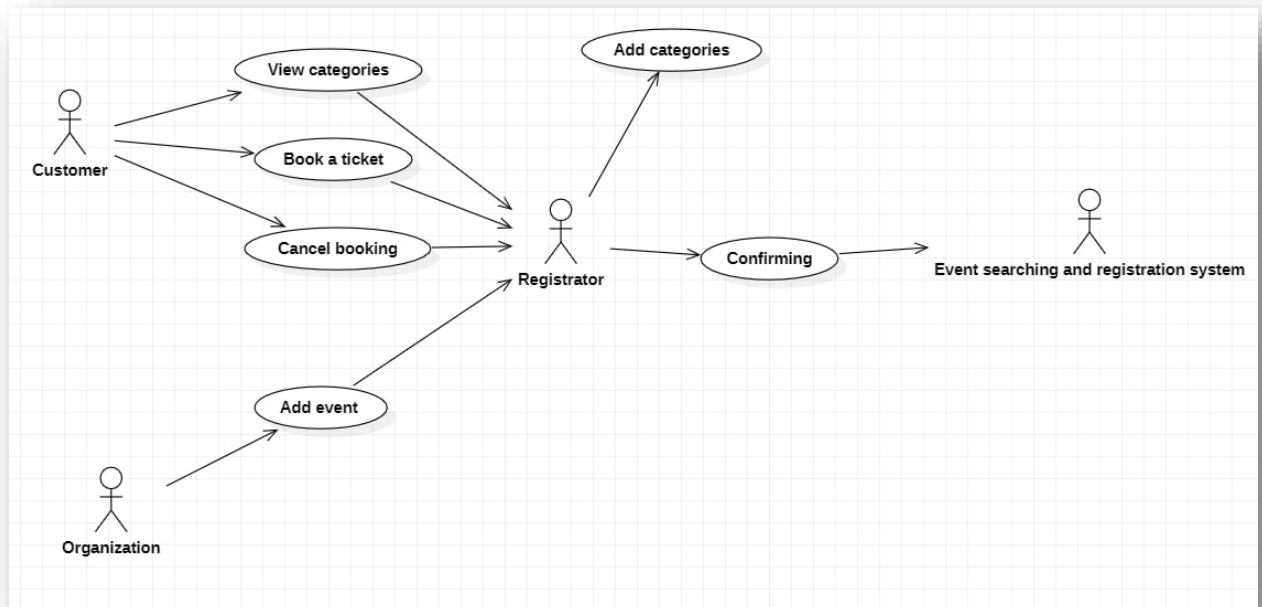
Виділяємо нову діаграму, а потім вводимо назву (Business Use Case Diagram).

Двічі клацаємо на назві цієї діаграми в браузері, щоб відкрити її. У вікні діаграм відкриється порожнє вікно Business Use Case Diagram.

Щоб помістити дійову особа або варіант використання на діаграму, перетягнемо його мишею з браузера на діаграму варіантів використання.

За допомогою кнопки Directed Association панелі інструментів намалюємо асоціацію між діючими особами і варіантами використання.

Розміщуємо елементи діаграми і отримуємо наступний результат:



В пакеті Use-Case Model створимо пакет Use-Case Realizations, потім всередині нього - 4 пакети: Use-Case Realization - Login, Use-Case Realization Book a ticket, Use-Case Realization – Confirm Registration, Use-Case Realization – Create event.

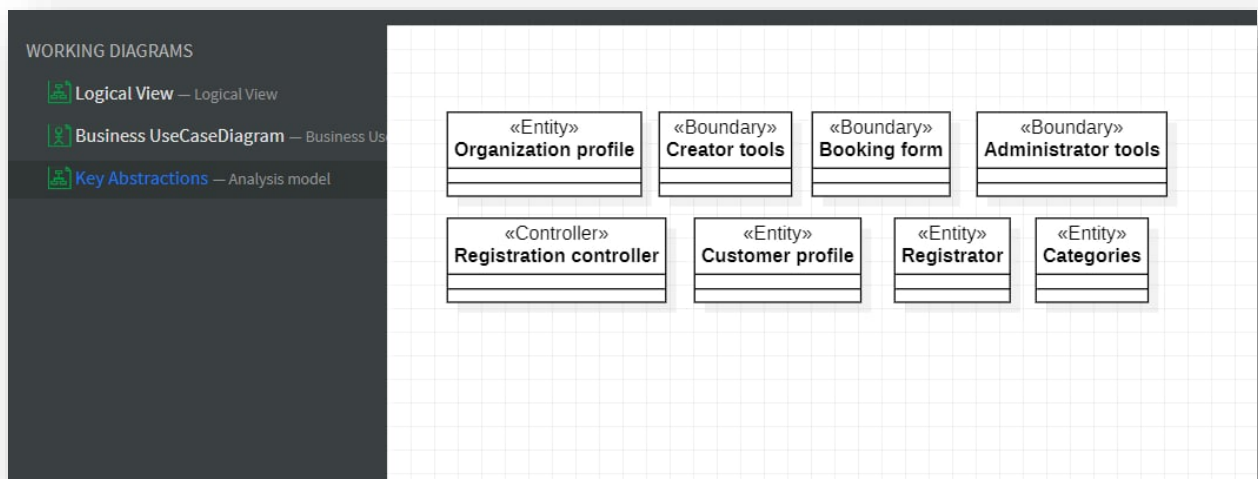
Ці пакети в подальшому знадобляться для створення діаграм.

Тепер перейдемо до створення класів.

Клацнемо правою кнопкою миші на пакеті Analysis Model логічного представлення Logical View. Оберемо в меню пункт New/ Class. Новий клас під назвою NewClass з'явиться в браузері. Введемо ім'я Customer profile. Створимо аналогічним чином класи Booking form, Categories, Creator tools, Registration controller, Administrator tools, Organization profile.

Клацнемо правою кнопкою миші на пакеті *Analysis Model*. У меню, оберемо пункт *New/ Class Diagram*. Назвемо нову діаграму класів *Key Abstractions*.

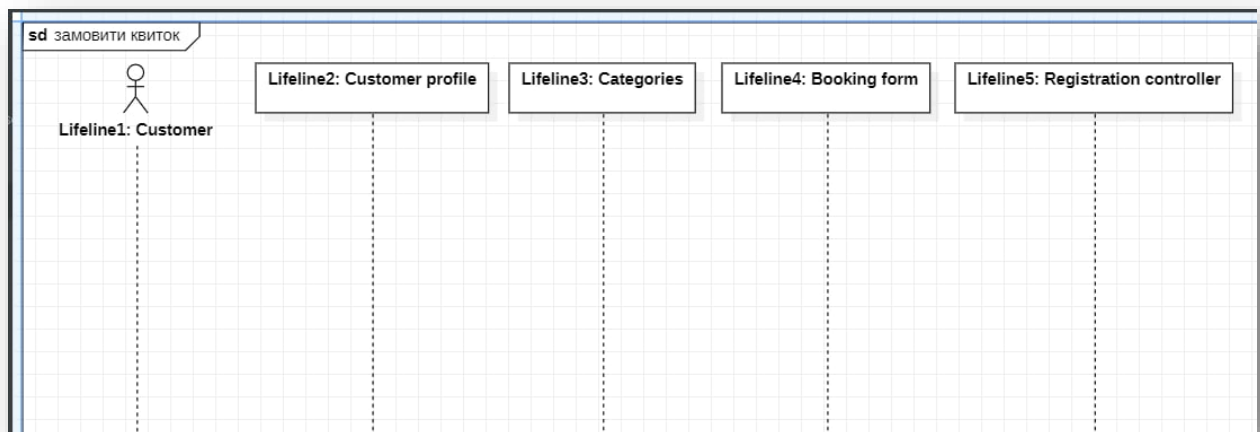
Щоб Розташувати новостворені класи на діаграмі класів *Key Abstractions*, відкриємо її вікно подвійним клацанням і перетягнути класи на відкриту діаграму мишею.




Тепер, коли в нас є дійові особи, варіанти використання і класи, створимо діаграми послідовностей.

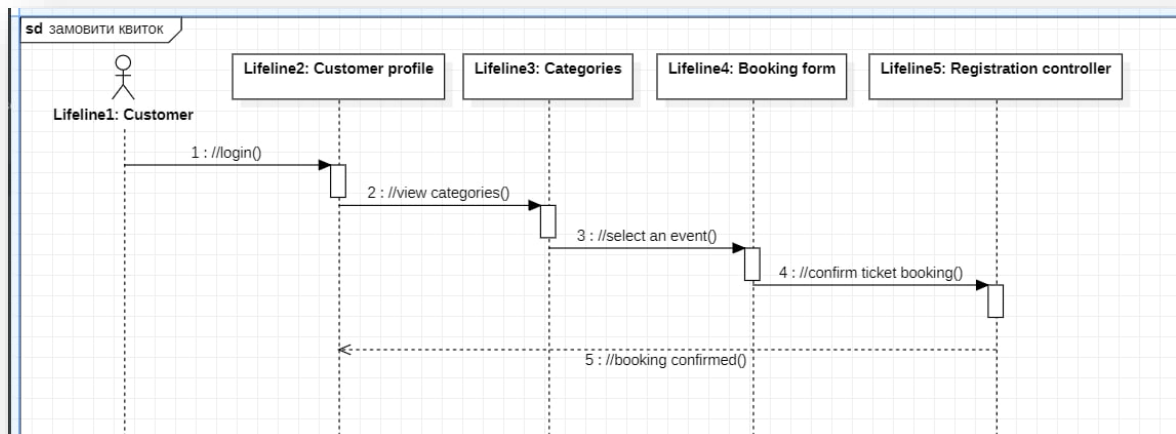
Натискаємо правою кнопкою миші на кооперації *Book a ticket* в пакеті *Use-CaseRealization* - *Book a ticket*. В меню оберемо пункт *New/ Sequence Diagram*. Назвемо нову діаграму *Register for Courses - Book a ticket*. Двічі натискаємо на ній в браузері, щоб відкрити її вікно.

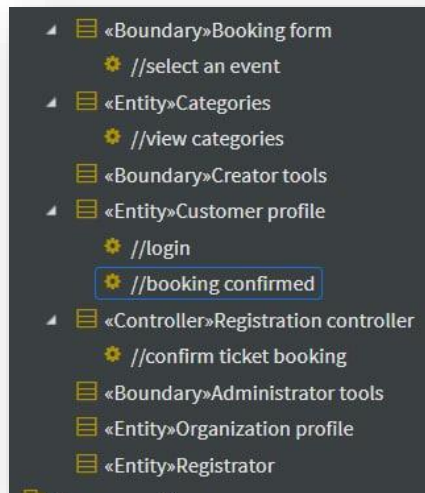
Перетягуємо на діаграму з браузера дійову особу Користувач і класи, з якими він буде взаємодіяти для бронювання квитка.



Тепер додаємо повідомлення. На панелі інструментів натискаємо кнопку  Object Message (Повідомлення об'єкта). Проведемо мишею від лінії життя дійової особи Користувач до лінії життя об'єкта Customer profile. Виділивши повідомлення над стрілкою, введемо його ім'я: `//login()`. Аналогічним чином додамо інші повідомлення до нашої діаграми. Також до повідомлень ми прикріплюємо операції (це можна зробити, клікнувши на стрілку повідомлення і натиснувши add operation)

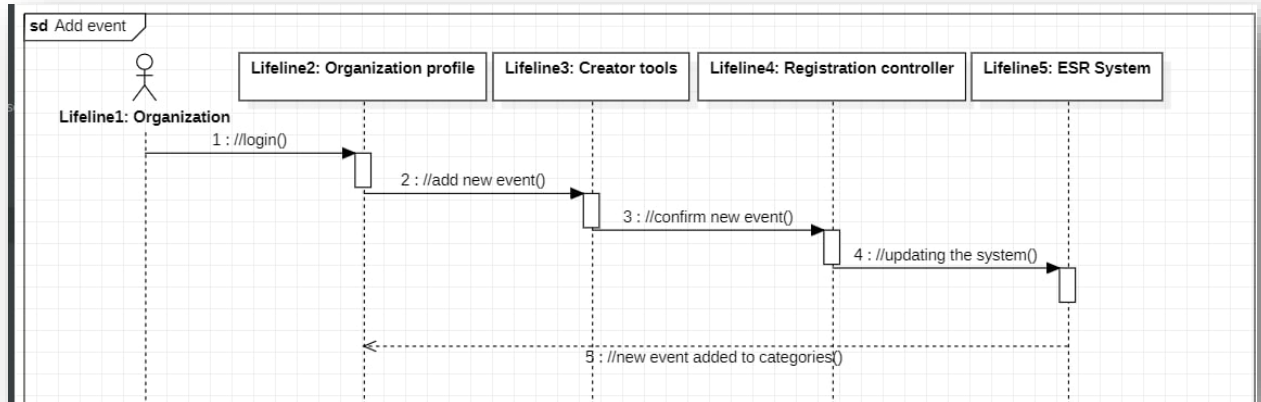
В результаті отримуємо наступну діаграму:





Логіка цієї діаграми: користувач заходить у свій профіль, переглядає категорії та обирає на який івент він хоче замовити квиток. Обирає цей івент, переходить на форму реєстрації, заповнює її та надсилає на підтвердження. В кінці користувач отримує підтвердження реєстрації і квитки.

Аналогічну діаграму створимо для кооперації Add event:

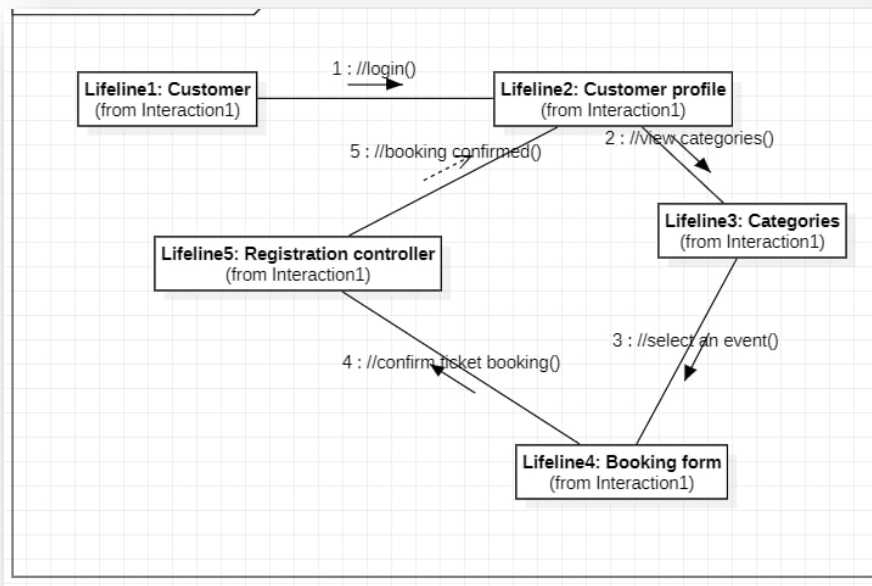


Логіка діаграми: щоб додати новий івент, організації, що проводить його, потрібно спершу зайти в свій кабінет. В цьому кабінеті, за допомогою функціоналу Creator Tools організація вносить усю необхідну інформацію, після чого надсилає запит на підтвердження цієї події. Коли підтвердження отримане, система оновлюється і івент додається до категорій.

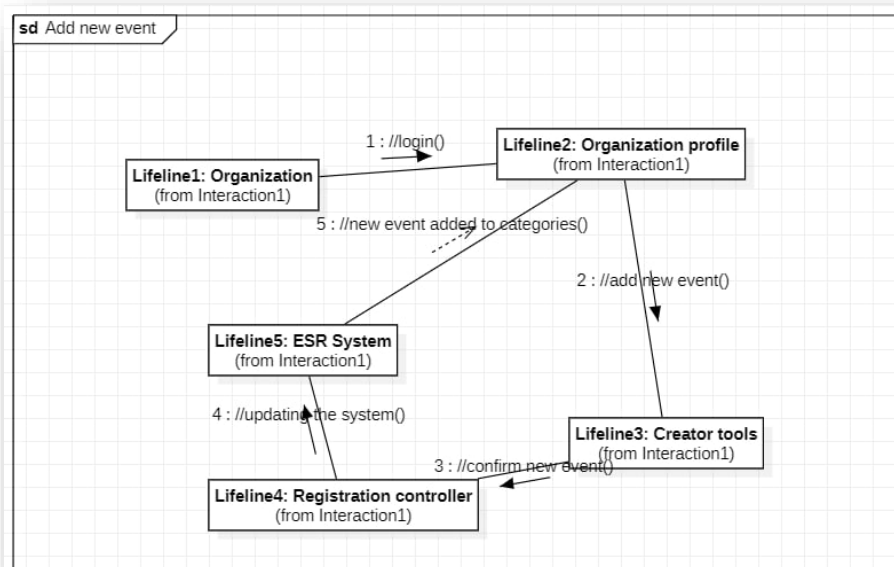
Далі ми створюємо діаграму кооперацій. Для її створення потрібно натиснути правою кнопкою миші на діаграму послідовності, далі натиснути add communication diagram і на вікно, що відкриється, перетягнути об'єкти з нашої діаграми послідовності. Зв'язки між цими об'єктами вибудовуються автоматично на основі попередньої діаграми.





Створимо діаграми кооперацій для обох діаграм послідовностей:


► Book a ticket:



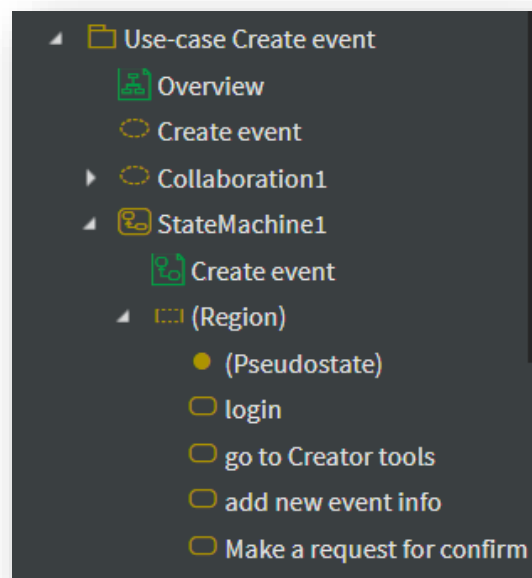
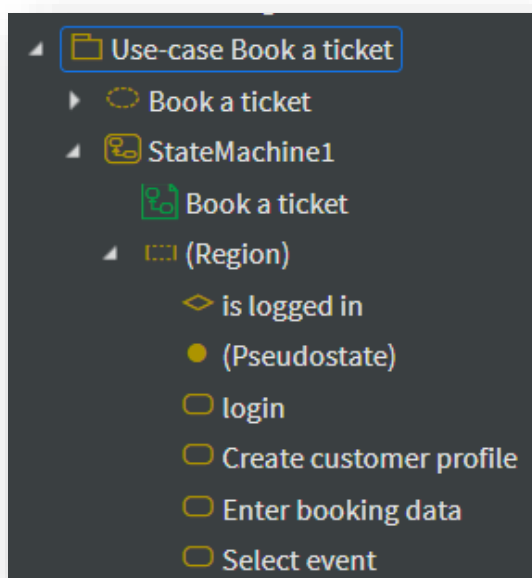
► Add new event:

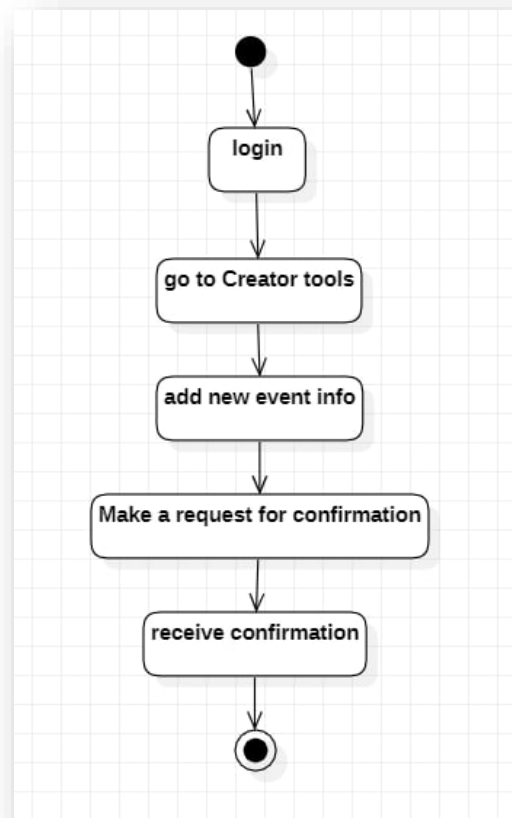
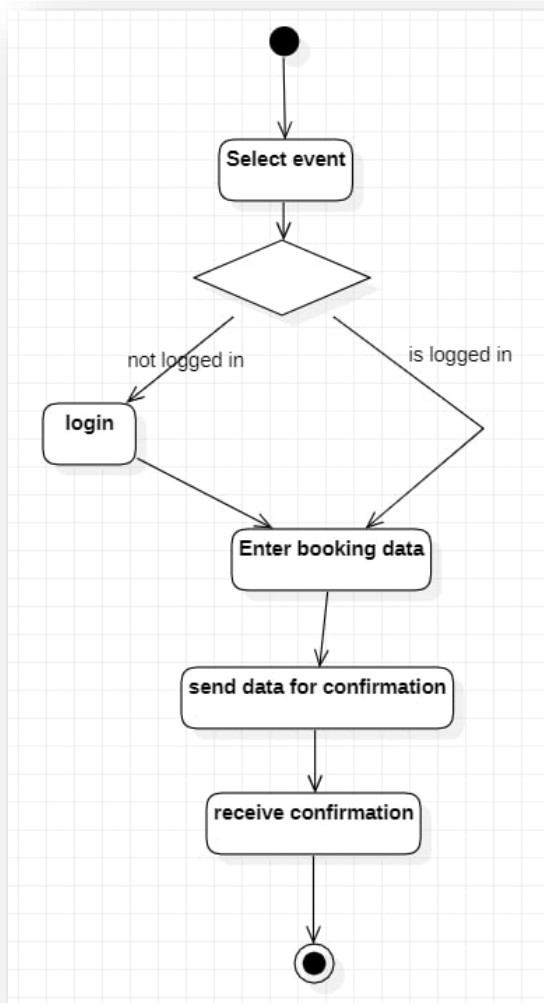


Наступним кроком буд створення діаграми станів. Клацнемо правою кнопкою миші в браузері на потрібному класі Book a ticket в пакеті  Analysis Model логічного представлення. У меню, оберемо пункт *New/ Statechart Diagram*. Даємо ім'я діаграмі станів  Booking. Подвійним клацанням відкриваємо вікно діаграми. На панелі інструментів натискаємо кнопку  State. Натискаємо мишею на діаграмі станів в тому місці, куди хочемо його помістити. Даємо ім'я станам. Натискаємо кнопку  Transition на панелі інструментів. Натискаємо мишею на стані, звідки здійснюється перехід. Проведемо лінію переходу до того стану, де він завершується.


Щоб додати початкове або кінцеве стан використовуємо кнопку  Start State і End State. Двічі клацаємо на переході, щоб відкрити вікно його специфікації. Вводимо умови переходу.

Таким чином створюємо діаграми для Book a ticket та Create event:





Далі створимо діаграму розміщення нашої системи.

Двічі натискаємо на представленні розміщення  Deployment View в браузері. З панелі інструментів перетягуємо на діаграму об'єкт Node. Даємо ім'я пристрою. Подвійне натискання на паралелепіпеді <device name>. Відкриється вікно. У вікні Device Specification задаємо параметр Name = Мережевий принтер. Натискаємо ОК. Подвійне натискання на паралелепіпеді <processor name>. Відкриється вікно Processor Specification. Даємо ім'я процесору: Name = Сервер ESR системи. Призначаємо процесору стереотип, характеристики, планування і процеси. На вкладці General вводимо стереотип Stereotype = Сервер.

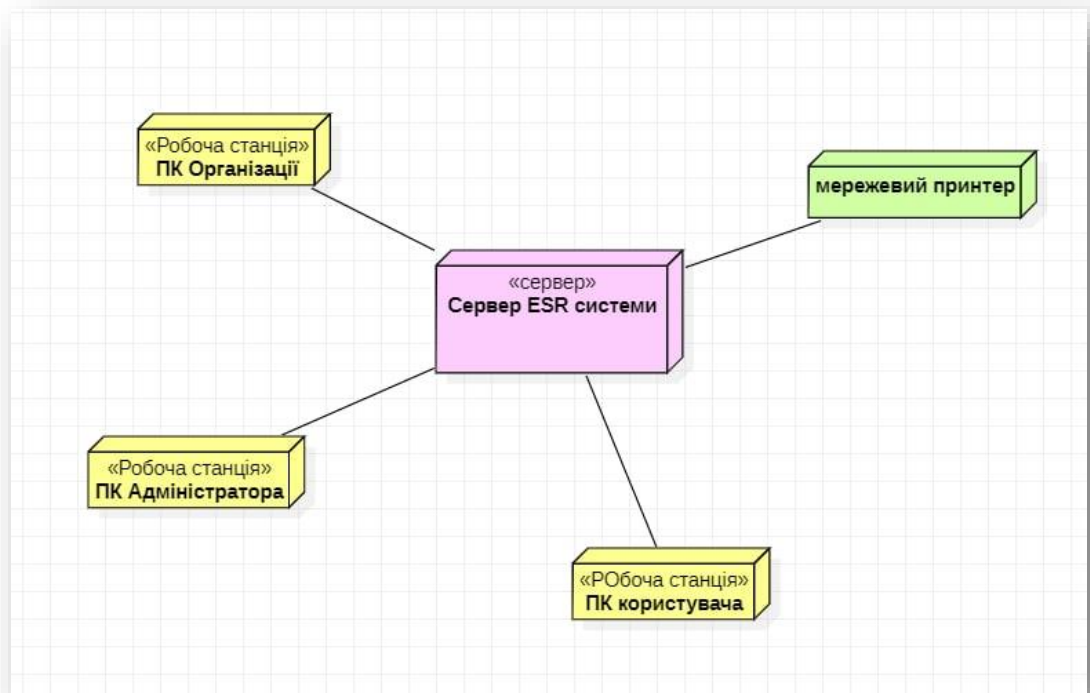
Тепер додаємо зв'язки.

На панелі інструментів обираємо Connection і проводимо зв'язки між нашими блоками.


Об'єкти діаграми:

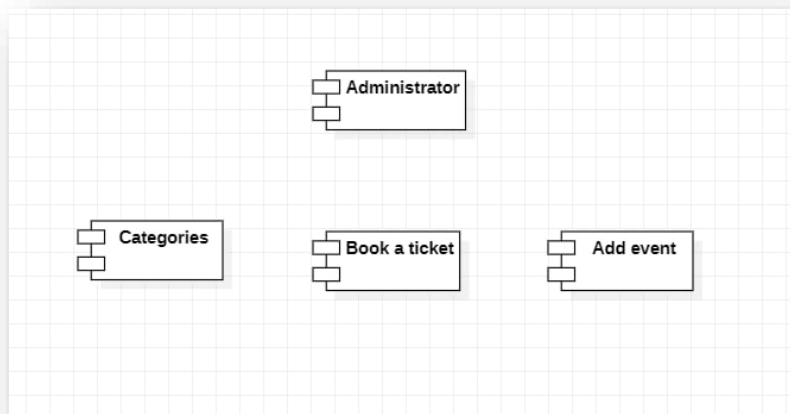


Ось таку діаграму розміщення системи ми отримали в результаті:

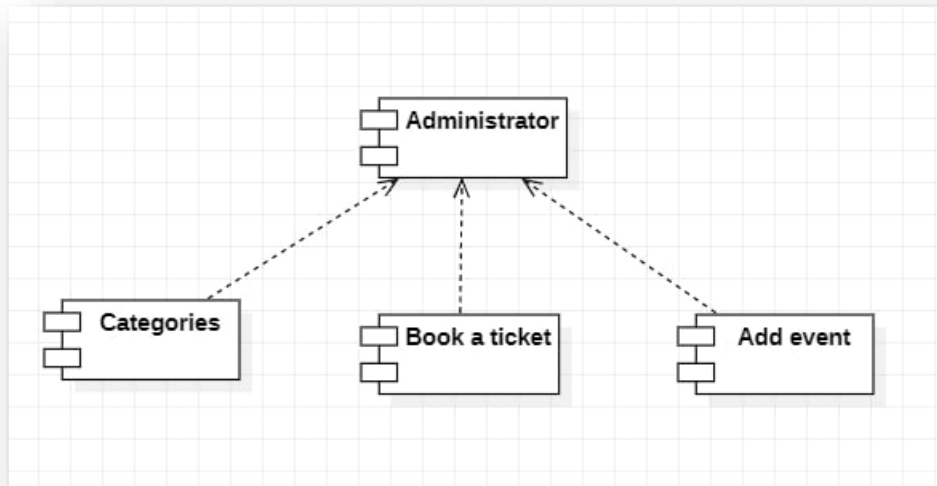


Далі створимо діаграму компонентів.

Двічі натискаємо на головній діаграмі компонентів Main в поданні компонентів  Component View. Відкриється вікно. На панелі інструментів натиснемо кнопку Package Specification. Помістимо специфікацію пакета на діаграму. Введемо ім'я специфікації пакета Name = Administrator і вкажемо у вікні специфікації мову Language = C ++. Створимо аналогічно ще кілька компонентів.



На панелі інструментів натиснемо кнопку Dependency. Проведемо лінію залежності. В результаті отримаємо наступну діаграму:



На завершення, спробуємо створити код на C++ для нашого проекту. Обираємо пакет Logical View, зверху на панелі натискаємо Tools – Generate C++ code. Вказуємо папку, куди програмі зберегти файли з кодом, і потім переглядаємо цю папку:

Имя	Дата изменения
Administrator tools.cpp	5/25/2022 5:54 PM
Administrator tools.h	5/25/2022 5:54 PM
Booking form.cpp	5/25/2022 5:54 PM
Booking form.h	5/25/2022 5:54 PM
Categories.cpp	5/25/2022 5:54 PM
Categories.h	5/25/2022 5:54 PM
Creator tools.cpp	5/25/2022 5:54 PM
Creator tools.h	5/25/2022 5:54 PM
Customer profile.cpp	5/25/2022 5:54 PM
Customer profile.h	5/25/2022 5:54 PM
ESR System.cpp	5/25/2022 5:54 PM
ESR System.h	5/25/2022 5:54 PM
Organization profile.cpp	5/25/2022 5:54 PM
Organization profile.h	5/25/2022 5:54 PM
Registration controller.cpp	5/25/2022 5:54 PM
Registration controller.h	5/25/2022 5:54 PM
Registrator.cpp	5/25/2022 5:54 PM
Registrator.h	5/25/2022 5:54 PM

Як бачимо, код успішно згенерувався.

Висновок: під час цієї лабораторної роботи я дослідила процеси та набула практичних навичок з розробки моделі складної програмної системи шляхом виконання конкретного проекту, а саме - програмно реалізувала систему надання електронних послуг для перегляду можливих івентів та бронювання квитків на обраний івент.