

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Екскурсійний чат-бот на платформі AWS»

Виконала студентка 4-го курсу, групи ПІ-55
спеціальності 121 «Інженерія програмного
забезпечення»

_____ М.О. Чмут

Керівник к.т.н., доцент, доцент кафедри ПІЗ

_____ В.П.Петров

Рецензент к.т.н., доцент, КІ та КБ

_____ Г.С. Семенюк

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

ФАКУЛЬТЕТ _____
(назва прописними)

КАФЕДРА _____
(назва прописними)

СПЕЦІАЛЬНІСТЬ _____
(назва прописними)

ЗАТВЕРДЖУЮ

Зав. кафедри _____

(назва кафедри)

« _____ » _____ 20____ р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу

Студента _Чмут Марії Олександрівни _____

Тема роботи: _«Екскурсійний чат-бот на платформі AWS» _____

Затверджена Наказом університету від « _____ » _____ 20____ р. № _____

Термін здачі студентом закінченої роботи _____

Вихідні дані роботи (зазначається предмет і об'єкт дослідження) _____

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1		2.03.20	2.03.20
2		6.04.20	6.04.20
3		4.05.20	4.05.20

Керівник _____
(підпис)

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	2 березня 2020 – 5 квітня 2020	
2	Проектування структури системи	6 квітня 2020 – 12 квітня 2020	
3	Написання програмного коду	13 квітня 2020 – 26 квітня 2018	
4	Тестування	27 квітня 2020 – 4 травня 2020	
5	Захист	4 червня 2020	

Студент

(підпис)

Керівник

(підпис)

РЕФЕРАТ

В структуру випускної роботи бакалавра входить пояснювальна записка та програмний комплекс. Пояснювальна записка до випускної роботи містить № сторінок, № зображення та № таблиць. Програмний комплекс складається з мобільного додатку для операційних систем Android/iOS та серверного рішення на базі платформи AWS.

Метою кваліфікаційної роботи бакалавра є дослідження тематики машинного навчання чат-боту для галузі мистецтвознавства на платформі AWS.

КЛЮЧОВІ СЛОВА: ЧАТ-БОТ, AMAZON, DYNAMODB, LEX, LAMBDA.

					ІПЗ.КР.Б – 121 – 20 – ІЗ						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Чмут М.О.			Екскурсійний чат-бот на платформі AWS			Літ.		Арк.	Аркуші
Керівник		Морозов А.В.									
Рецензент											
н. контр.											
Зав. каф.											
					Пояснювальна записка			Житомирська політехніка, група ПІ-55			

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ЕКСКУРСІЙНОГО ЧАТ-БОТУ НА ПЛАТФОРМІ AWS	7
1.1. Постановка задачі.....	7
1.2. Аналіз аналогів програмного продукту	8
1.3. Вибір архітектури екскурсійного чат-боту	12
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	14
Висновки до першого розділу	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ЕКСКУРСІЙНОГО ЧАТ-БОТУ НА ПЛАТФОРМІ AWS	19
2.1. Визначення варіантів використання та структури системи	19
2.2. Розробка бази даних та структури файлової системи	24
2.3. Проектування та реалізація алгоритмів роботи та функціоналу системи.....	24
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ СИМУЛЯЦІЇ	25
3.1. Порядок встановлення та налаштування системи.....	25
3.2. Структура інтерфейсу. Інтерфейс та порядок роботи з системою симуляції. .	25
3.3. Тестування роботи системи	25
ВИСНОВКИ.....	26
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	27
ДОДАТКИ.....	28

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AWS – Amazon Web Services

API – Application Programming Interface.

ОС – Операційна система.

UI – User Interface.

MVVM – Model View Viewmodel.

CRUD – Create Read Update Delete.

					ІПЗ.КР.Б – 121 – 20 – ПЗ	
						4

ВСТУП

Актуальність теми. Серед останніх тенденцій цифрової індустрії все більше уваги привертають чат-боти. З великою кількістю досліджень в цій області програмування стає все більш людиноподібним, а не тільки суто машинно-автоматизованим. Поєднання швидкої реакції і послідовних зв'язків робить чат-боти більш привабливою альтернативою веб-додаткам. В загальних рисах, чат-бот – це не що інше, як програмне забезпечення, яке виконує певні автоматичні дії. Іншими словами, бот – це комп'ютерна програма, яка призначена для спілкування з користувачами Інтернету.

Основна ціль чат-боту – це вести дискусію або розмову з людиною на певну тематику. Наприклад, будь-який користувач може задати боту запит або питання і той відповість або виконає дію відповідно запиту, якщо він є доречним. Чат-бот взаємодіє у форматі, подібному до обміну миттєвими повідомленнями, основна властивість якого - це здатність комп'ютера розуміти людську мову або текст. Чатбот - це просто комп'ютерна програма, яка принципово імітує людські розмови. Він надає відповіді на основі поєднання заздалегідь визначених сценаріїв та програм машинного навчання. Коли користувач задає питання, чат-бот відповість на підставі наявної на той момент бази даних. Якщо в розмові введено поняття, яке не запрограмовано для розуміння, що також є частиною машинного програмування, чат-бот або відхилить розмову, або потенційно передасть спілкування користувачу для зміни запиту. Наприклад, якщо запитати Amazon's Alexa, Apple Siri або Microsoft's Cortana: "Яка погода?", чат-бот відповість згідно останніх звітів про погоду, до яких він має доступ.

Таким чином, чатбот буде поступово розростатися в масштабі і набувати актуальності. Саме тому чат-боти користуються популярністю у багатьох підприємствах з розвиненою ІТ-інфраструктурою. Основна ціль використання чат-ботів в таких компаніях – це покращення досвіду роботи з клієнтами. Розробники використовують журнали розмов з чат-ботів для аналізу запитів користувачів. Це дозволяє їм зрозуміти потреби клієнтів аби покращити сервіс

підприємства. Також, чат-боти актуальні не тільки в підприємствах та бізнесі, але й у інших галузях, будь то медицина, освіта або мисцетво. Чат-бот можна підлаштувати під будь яку галузь, в залежності від людських потреб, тому тематика чат-ботів є і буде актуальною і надалі.

Метою кваліфікаційної роботи бакалавра є дослідження тематики машинного навчання чат-боту для галузі мистецтвознавста на платформі AWS.

Встановлена мета обумовлює наступні завдання:

- вивчення особливості роботи чат-ботів та їх машинного навчання на платформі AWS;
- проведення аналізу процесу комунікації між користувачем та ботом;
- визначення архітектури та узагальненої структури чату;
- обґрунтування та вибір засобів реалізації чат-боту;
- проектування та розробка чат-боту на платформі AWS, включаючи усі можливі опції Amazon для боту.

Об'єктом дослідження є технологія розробки чат-боту на платформі AWS.

Предметом дослідження є поцес спілкування користувача та бота на тему мистецтва, а також використання бота в якості персонального гіда в музеях.

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ЕКСКУРСІЙНОГО ЧАТ-БОТУ НА ПЛАТФОРМІ AWS

1.1. Постановка задачі

Детально вивчивши особливості роботи екскурсійних чат-ботів та його внутрішніх процесів, можна виділити основні задачі, які потребують автоматизованого рішення. Розглянемо наведений нижче список задач.

1. Розробка інформаційної системи для надання інформації стосовно картин, авторів та жанрів мистецтва, представлених у мережі музеїв, підключених до системи. Відповідно цієї задачі, чат-бот повинен надавати необхідну інформацію користувачу відповідно його запиту, якщо той валідний. Інформація може стосуватися не тільки музейного об'єкту, але і його жанру та автора.

2. Облік експонатів музею. Для зручності виведення інформації вона повинна зберігатися в базі даних. Для кожного об'єкта повинна бути своя власна таблиця, де всі дані повинні бути чітко структурованими.

3. Забезпечення надання інформації у максимально людино-подібній формі. Відповідно цій задачі, основна ціль даного проекту – це зробити інтерфейс спілкування людина – машина якомога непомітним. Для цього буде застосовуватися машинне навчання за допомогою деяких сервісів Amazon.

Варто зазначити, що цей перелік охоплює лише ключові етапи роботи, та може бути змінений, в залежності від особливостей підприємства та його робочих практик.

Кожна сформована задача вимагає розробки комплексного та масштабного рішення. Однак для досягнення поставленої мети необхідно їх всі врахувати. Тому для виконання поставлених задач основна частина роботи припадає на серверну частину додатку, а також на заповнення та адміністрування бази даних. Потреби в користувацькому інтерфейсу немає, оскільки реалізація чат-боту буде в меседжерах з уже готовим інтерфейсом для кожної із платформ, обраних для деплоя бота.

1.2. Аналіз аналогів програмного продукту

Під час дослідження процесу роботи екскурсійного чат-боту, було вивчено існуючі продукти, які можуть бути аналогом для системи, що розробляється. Також для більш детального аналізу було розглянуто чат-боти, побудовані на платформі AWS. Останнє обумовлене тим, аби при розробці власного продукту врахувати усі переваги та недоліки AWS чат-ботів при побудові свого боту.

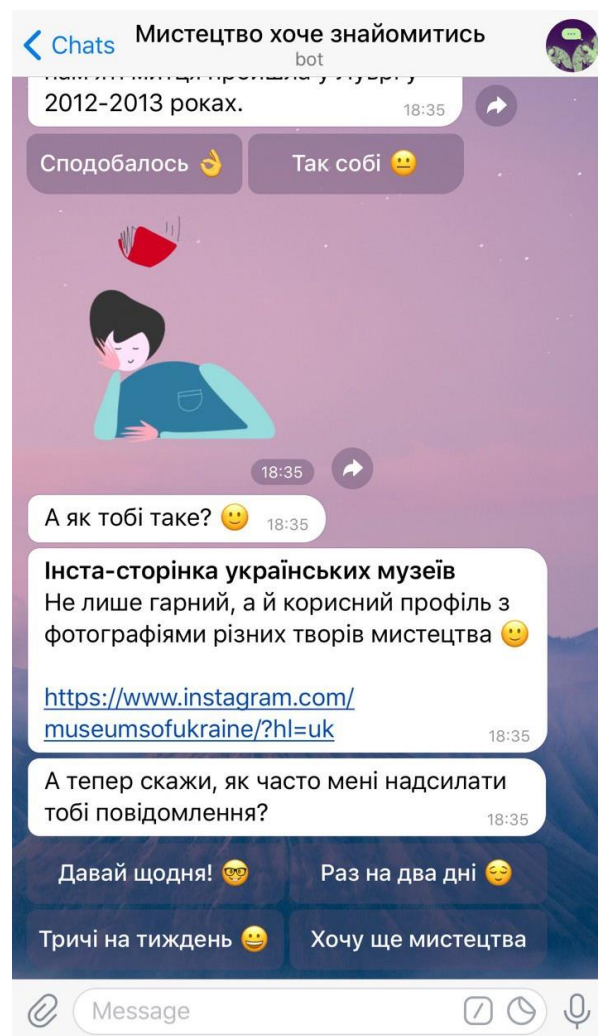
Враховуючи популярність чат-ботів для різних галузей, існує велика кількість пропозицій і для галузі мистецтвознавства. Розглянемо декілька з них.

Date with Art – це чат-бот про мистецтво, розроблений спільно з Digital-агенцією Postmen та Українським культурним фондом. Основна ідея бота полягає в тому, щоб залучити людей, які цікавляться мистецтвом та стати активним учасником соціально-культурних процесів в Україні. Головний засіб комунікації – чат-бот “Date with art” в Telegram та Facebook Messenger, який працює в трьох напрямках:

- пропонує вибрати цікаві для тебе рубрики: живопис, скульптура, театр, танець, книговидавництва;
- надсилає мистецький твір, який ти оцінюєш — подобається чи не дуже;
- запитує, як часто надсилати повідомлення і підбирає наступні мистецькі твори уже за твоїми вподобаннями.

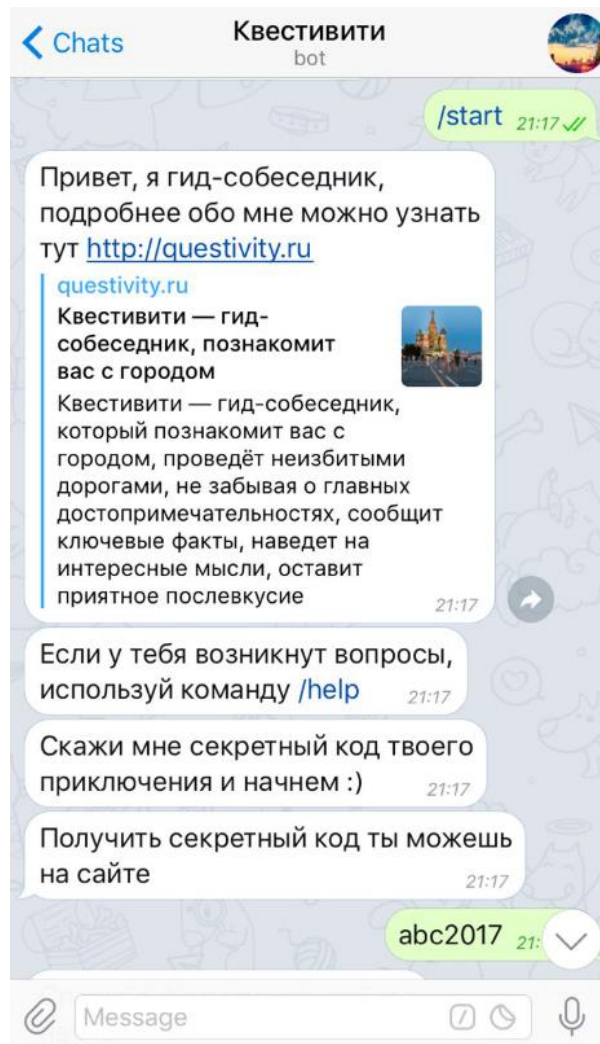
У чат-бота “Date with art” також є сторінки у соціальних мережах Facebook та Instagram. Інтерфейс чат-боту “Date with art” зображено на рисунках 1.1 – 1.2.

Детальніше ознайомитись з продуктом можна за посиланням <https://datewithart.net.ua/>.



Рисунки 1.1-1.2 – Чат-бот “Date with art” в Telegram Messenger

Наступний аналог продукту, який ми розглянемо, це бот-екскурсовід по Москві “Questivity”. Даний чат-бот дозволяє самостійно влаштувати екскурсію по маловідомим визначним місцям районам Москви. Користувач сам обирає, коли відправитися на прогулянку. Все, що потрібно - це телефон із зарядженою батареєю і месенджер. Немає необхідності приїжджати в строго призначений час, ходити з натовпом незнайомих людей і прислухатися до тихого голосу екскурсовода. На прогулянку можна відправитися одному або з друзями. Також у бот встроєна функціональність задавати цікаві питання, на які потрібно шукати відповіді навколо себе.



Рисунки 1.3 – Чат-бот “Questivity” в Telegram Messenger

Інший аналог предмету дослідження – Rov-E чат-бот на платформі AWS, а саме Amazon Lex. Було вирішено протестувати та взяти в аналіз один із чат-ботів на платформі Amazon Lex, оскільки саме на ній буде інтегровано власний продукт. Rov-E - це роботизована система для вивчення зірок, що є близькою копією реальних марсоходів NASA. Rov-E був створений співробітниками NASA, використовуючи Amazon Lex для того, аби надихнути нове покоління для космічних досліджень. Використовуючи Amazon Lex, співробітники NASA тепер можуть просто управляти Rov-E за допомогою голосових команд. Можливість управління діалогом з декількома співрозмовниками дозволяє Rov-E «говорити», захоплюючи відповідаючи на питання учнів про Марс. Інтеграція

з сервісами AWS дозволяє Rov-E підключатися до різних джерел даних для отримання масштабної інформації про дослідження Марса NASA.

Зовнішній вигляд чат-боту Rov-E зображено на рисунку 1.3. Бот інтегрований тільки під веб-сторінку. Детальніше ознайомитись з продуктом можна за посиланням <https://mars.nasa.gov/ask-nasa-mars/#/>.

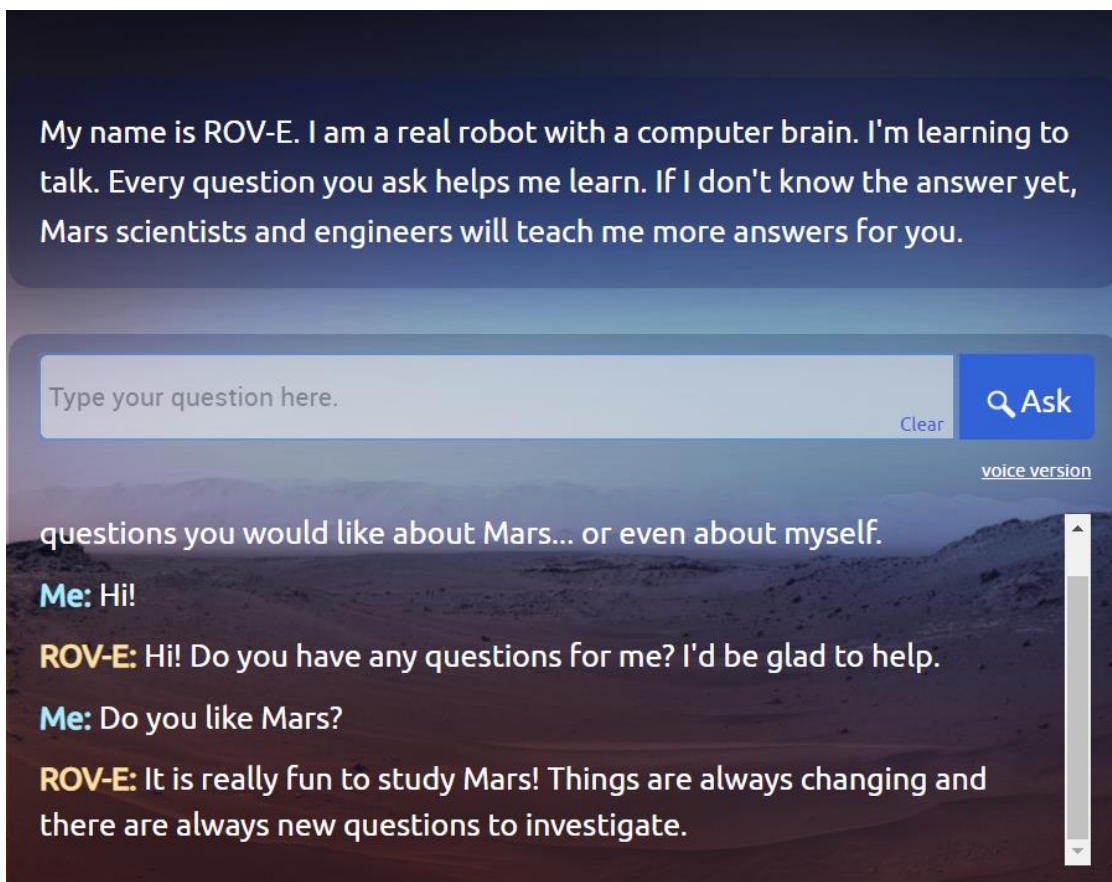


Рисунок 1.3 – Інтерфейс чат-боту Rov-E

Порівнюючи розглянуті системи з предметом дослідження, варто виділити ключову відмінність – в обох системах відсутня ініціація діалогу зі сторони боту. Також в чат-боті “Date with art” інформація фокусується суто на об’єкті мистецтва, тобто на самому творі. В той час як було б доречним надавати інформацію не тільки про об’єкт, але і про художників, види мистецтва, музеї. Тому це буде враховано під час реалізації власного продукту.

Під час дослідження аналогів програмного продукту, не було знайдено системи, що реалізує повний перелік функціоналу предмету дослідження.

Враховуючи особливості та відмінності програмного продукту та його аналогів, можна зробити висновок, що вони не лише частково виконують аналогічні задачі, але й доповнюють одне одного, тому можуть бути інтегровані. Наприклад, екскурсійний чат-бот може бути поєднаний з машинним навчання боту Rov-E та його голосовими повідомлення від Amazon Polly.

1.3. Вибір архітектури екскурсійного чат-боту

Оскільки система передбачає одночасну участь багатьох користувачів, централізоване зберігання даних та їх поширення між різними користувачами, оптимальним рішенням для розробки становить клієнт-серверна архітектура.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних додатків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Також варто зауважити, що в основі взаємодії клієнт-сервер лежить принцип того, що таку взаємодію починає клієнт, сервер лише відповідає клієнту і повідомляє про те, чи може він надати послугу клієнтові і якщо може, то на яких умовах. Клієнтське програмне забезпечення та серверне програмне забезпечення зазвичай встановлено на різних машинах, але також вони можуть працювати і на одному комп'ютері.

На рисунку 1.4 схематично зображені елементи архітектури системи.

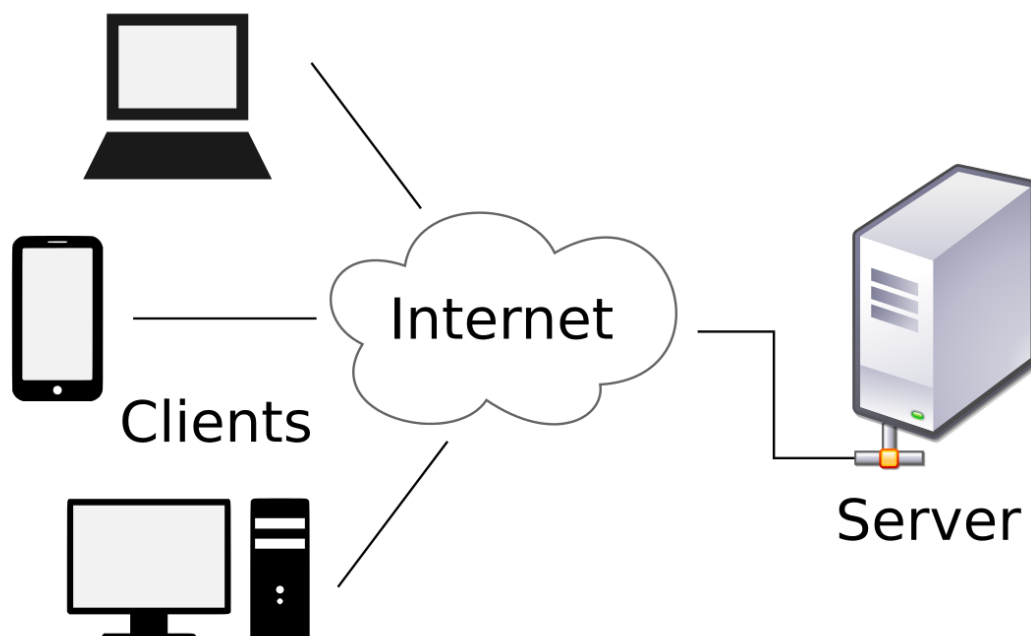


Рисунок 1.4 – Схема роботи клієнт-серверної архітектури

Розглянемо детальніше ключові елементи даної системи.

Сервер представляє собою веб-сервер, що є центральним елементом обробки даних та забезпечує комунікацію між клієнтами. Також, до сервера відноситься база даних – централізоване сховище інформації, якою оперують клієнти.

У системі передбачено декілька типів клієнтських додатків:

- Мобільний додаток – елемент системи, що забезпечує взаємодію майстра з адміністратором. Мобільний додаток надає можливість майстру входити в систему, переглядати матеріали до використання та створювати запис про обслуговування клієнта, який надходить до адміністратора та є основою для формування рахунку клієнта.
- Веб додаток - елемент системи, що власне забезпечує взаємодію з кінцевим користувачем, який обов’язково є адміністратором. Веб додаток знаходиться в режимі постійного моніторингу змін на сервері, він забезпечує отримання сповіщень про новий запис для адміністратора та використовується для аналізу та модифікацій змін у базі даних.

В якості мережі використовується мережа інтернет, що забезпечує комунікацію між усіма архітектурними елементами.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Перед вибором технологій розробки та інструментальних засобів важливо встановити актуальні вимоги до апаратного забезпечення кінцевих користувачів та системи серверу.

Згідно зі статистичними опитуваннями, користувачі смартфонів надають перевагу мобільним додаткам, а не мобільним версіям сайтів. Серед 3.5 тисяч респондентів з різних куточків світу, які брали участь в опитуванні, 85% стверджують, що мобільні додатки набагато зручніше і функціональніші, ніж мобільні версії сайтів. Мобільні додатки спрощують нам життя - ми можемо здійснювати покупки, замовляти їжу з ресторану, читати новини, дізнатися погоду. Вони служать нам як діловий щоденник, органайзер, допомагають вести особистий щоденник, стежити за здоров'ям, читати книгу і т.д. Сьогодні додатки допомагають людям здійснювати банківські операції, оплачувати покупки, рахунки, бронювати місця і квитки. Таким чином, користувачі вважають за краще використовувати мобільні додатки під IOS і Android для взаємодії з компаніями.

Деякі експерти вважають, що додаток зручніше, тільки якщо клієнт використовує його постійно. Мобільний додаток під IOS і Android дійсно зручно, коли воно вже встановлено. Мобільний сайт запускають ті користувачі, кому потрібна терміновість в доступі до інформації. У той же час, для найбільш повної функціональності варто застосовувати або мобільний додаток для ресурсу, або онлайн-сервіси, або адаптивну верстку.

Для прикладу можна взяти офіційну інформацію із сайту компанії StatCounter, яка займається збором статистики різного роду трафіку по всьому світі. Згідно графіку, який показано на рисунку 1.5 серед українських користувачів смартфонів найбільша кількість припадає на компанію Xiaomi, на другому місці Samsung, а на третьому місці Apple. У той час як серед планшетів лідує компанія Apple. Однак, програмний продукт, що розробляється буде орієнтований на усі мобільні платформи із тією метою, що чат-бот буде

					ІПЗ.КР.Б – 121 – 20 – ПЗ	
						14

інтегровано під додатки Telegram та Facebook, які уже адаптовані для IOS і Android платформ.

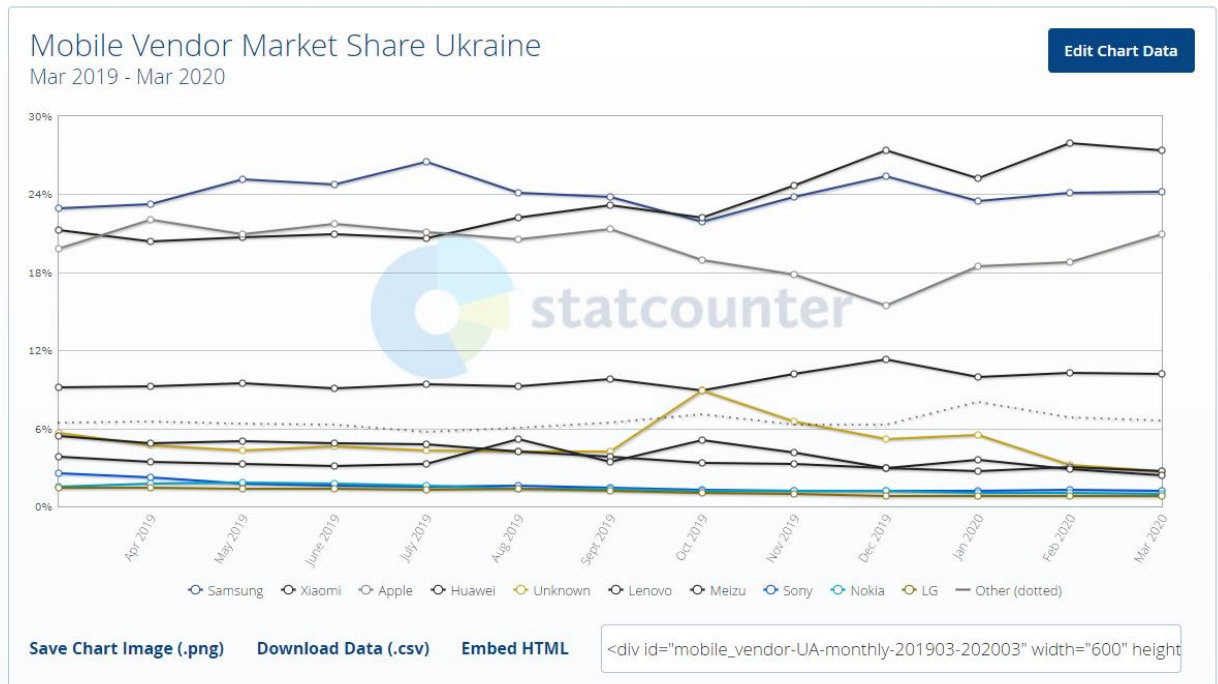


Рисунок 1.5 – Графік ринку мобільних пристроїв з квітня 2019 року по березень 2020 року

Сформуємо вимоги до апаратного забезпечення.

Додаток для мобільних клієнтів повинен функціонувати на сучасних мобільних пристроях з наступними технічними характеристиками:

- операційна система Android 5.1 та вище;
- операційна система iOS 12 та вище;
- мінімальний об'єм оперативної пам'яті 1Гб;
- від 2Гб внутрішньої пам'яті пристрою чи карти пам'яті;
- доступ до системи Інтернет.

Для середовища виконання серверної частини додатку висуваються наступні вимоги:

- операційна система на базі Windows чи Linux;
- підтримка мови Nodejs та C#;
- встановлений сервер;
- стабільний швидкісний доступ до Інтернету;
- оперативна пам'ять – не менше 16 Гб;
- об'єм дискової пам'яті – не менше 500 Гб;

Проаналізувавши встановлені вимоги до апаратного забезпечення, можна переходити до вибору інструментальних засобів розробки системи.

Оскільки для розробки бота буде використовуватися платформа AWS, то і інструменти розробки будуть використовуватися від Amazon. Слід виділити три ключових сервіси для розробки чат-боту:

- Amazon Lex – сервіс для створення голосових і текстових діалогових інтерфейсів в будь-яких додатках. Amazon Lex надає розширені функціональні можливості глибокого навчання, такі як автоматичне розпізнавання мови (ASR), призначене для перетворення мови в текст, і розуміння природних мов (NLU), призначене для визначення сенсу тексту. Це дозволяє створювати привабливі для користувачів додатки з можливостями природного діалогової взаємодії.
- Amazon Lambda - це сервіс бессерверних обчислень, який запускає програмний код у відповідь на певні події і відповідає за автоматичне виділення необхідних обчислювальних ресурсів. AWS Lambda може автоматично запускати програмний код у відповідь на різні події, такі як HTTP-запити через Amazon API Gateway, зміна об'єктів в кошику Amazon S3, оновлення таблиць в Amazon DynamoDB або зміна станів в AWS Step Functions. Код, що запускається в AWS Lambda, називається функцією Lambda. Після того як функція Lambda створена, вона перебуває в стані постійної

готовності до запуску, подібно формулами електронних таблиць. Кожна функція містить користувацький код і деякі дані конфігурації, включаючи ім'я функції і вимоги до ресурсів. Функції Lambda не зберігають стан і ніяк не залежать від базової інфраструктури, тому Lambda може швидко завантажити стільки копій функції, скільки потрібно для масштабування відповідно до кількості вхідних подій.

- Amazon DynamoDB - це нереляційна база даних пар «ключ-значення» і документів, яка забезпечує затримку менше 10 мілісекунд при роботі в будь-якому масштабі. Це надійна повністю керована база даних для додатків в масштабі всього Інтернету, яка працює в декількох регіонах з кількома провідними серверами і має вбудовані засоби забезпечення безпеки, резервного копіювання та відновлення, а також кешування в пам'яті. DynamoDB може обробляти більше 10 трлн запитів в день і справлятися з піковими навантаженнями, що перевищують 20 млн запитів в секунду.

Для написання усього коду буде використовуватися Amazon Lambda. Є два варіанти написання коду:

- Використовуючи веб-інтерфейс Amazon;
- Локально, але після завершення написання коду потрібно буде задеплоїти функцію Lambda на сервер Amazon Lambda.

Amazon Lambda підтримує декілька мов для написання коду: .Net Core 3.1 (C#/PowerShell), Java 8, Node.js 10.x, Python, Ruby, Go. Для розробки було обрано мову C# та локальне написання коду. В якості середовища розробки мобільного додатку використовується Microsoft Visual Studio, інтерфейс якого зображено на рисунку 3.2. Однак для написання коду спеціально для Amazon необхідно встановити додаткове розширення в Visual Studio, яке називається AWS Toolkit for Visual Studio. Тоді для розробки у обраному середовищі будуть доступні усі необхідні інструменти для сервісу AWS.

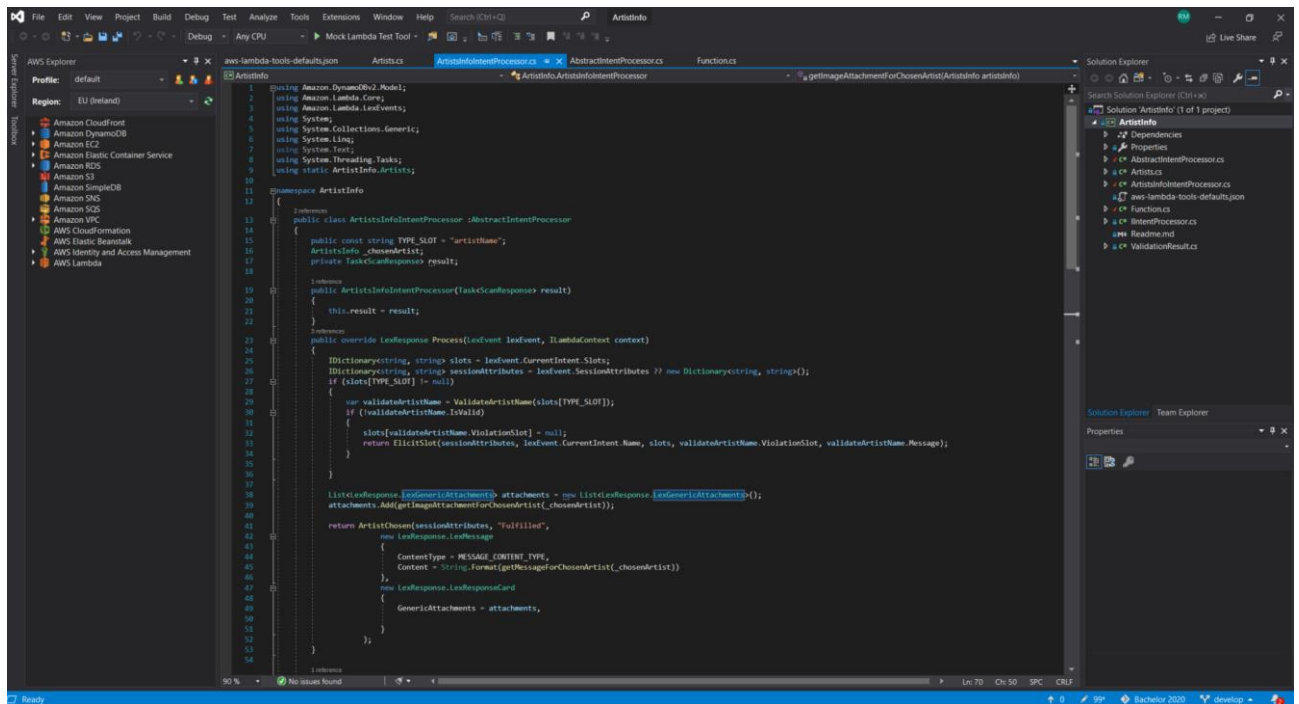


Рисунок 3.2 – Інтерфейс Microsoft Visual Studio

Після визначення апаратних вимог та технологій розробки, можна переходити до власне проектування системи, визначення варіантів використання, основних алгоритмів та визначення майбутньої архітектури окремих клієнтів.

Висновки до першого розділу

У першому розділі кваліфікаційної роботи проведено аналіз галузі роботи екскурсійних чат-ботів та досліджено існуючі аналоги ботів на тематику мистецтвознавства. Було сформовано вимоги до системи та на основі цих вимог обрано основну архітектуру системи, здійснено вибір інструментів для розробки. Було вирішено використовувати Microsoft Visual Studio з додатковим розширенням для AWS. Для реалізації завдання кваліфікаційної роботи необхідною умовою є розробка на платформі AWS і з застосуванням сервісів, описаних у підрозділі 1.4. Також, було обрано базу даних для збереження даних, яка вже інтегрована під платформу AWS та тоже застосовуватися при написанні функцій Amazon Lambda.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ЕКСКУРСІЙНОГО ЧАТ-БОТУ НА ПЛАТФОРМІ AWS

2.1. Визначення варіантів використання та структури системи

Для проведення проектування системи спершу необхідно визначити варіанти використання користувачами та скласти перелік вимог до системи.

Бізнес вимоги:

1. Основні цілі: проект створюється для забезпечення користувачам мобільних пристроїв на базі операційних систем (ОС) Android та iOS можливості самостійно вивчати експонати мистецтва без допомоги гіда. Роль гіда буде виконувати чат-бот.

2. Можливості: створення чат-боту для проведення онлайн-екскурсій.

3. Представлення проекту: проект розробляється в вигляді чат-боту для планшетів та телефонів на базі ОС Android та iOS, та буде доступний для використання в додатках Telegram та Facebook Messenger, як можна попередньо встановити із App Store або Google Play Market.

Вимоги користувачів:

1. Введення назви експонату;
2. Перегляд наданої інформації;
3. Оцінка наданої інформації;
4. Можливість обрати наступний експонат;
5. Можливість повернутися в головне меню;
6. Можливість перегляду інформації про музей;
7. Можливість перегляду інформації про художників;
8. Можливість перегляду інформації про жанр мистецтва.

Функціональні вимоги:

1. Пошук та перегляд даних;
2. Доповнення даних або їх видалення;
3. Кешування даних.

Нефункціональні вимоги до системи:

					ІПЗ.КР.Б – 121 – 20 – ПЗ	
						19

1. *Сприйняття.* Час необхідний для ознайомлення користувача зі способом роботи чат-боту має становити не більше пів години для досвідчених Android та iOS користувачів та не більше ніж півтори години для користувачів, які не мають досвіду роботи з Android або iOS додатками.

2. *Надійність.* Система не повинна помилково закінчувати роботу без попередження для користувача. Помилки роботи користувача повинні бути оброблені системою та забезпечені відповідним повідомленням про неправильне використання додатку.

3. *Продуктивність.* Система має підтримувати роботу користувача з чат-ботом протягом всього часу безпосередньої взаємодії та підтримувати роботу запитів користувача в фоновому режимі. Система повинна забезпечувати синхронізацію даних під час роботи різних користувачів.

4. *Можливість експлуатації.* Екскурсійний чат-бот має автоматично оновлюватися на мобільному пристрої користувача за наявності дозволу зі сторони операційної системи та обов'язково попереджувати користувача про оновлення.

Визначимо основні типи користувачів системи, так званих акторів. Під час роботи з об'єктом дослідження було виділено такі актори:

- Відвідувач – користувач чат-боту;
- AWS Адміністратор - зареєстрований та аутентифікований користувач платформи Amazon, який є адміністратором боту та має доступ до бази даних.

На рисунку 2.1 зображено діаграму варіантів використання системи, на якій схематично показані актори та основні дії, які виконують ці актори при роботі з системою.

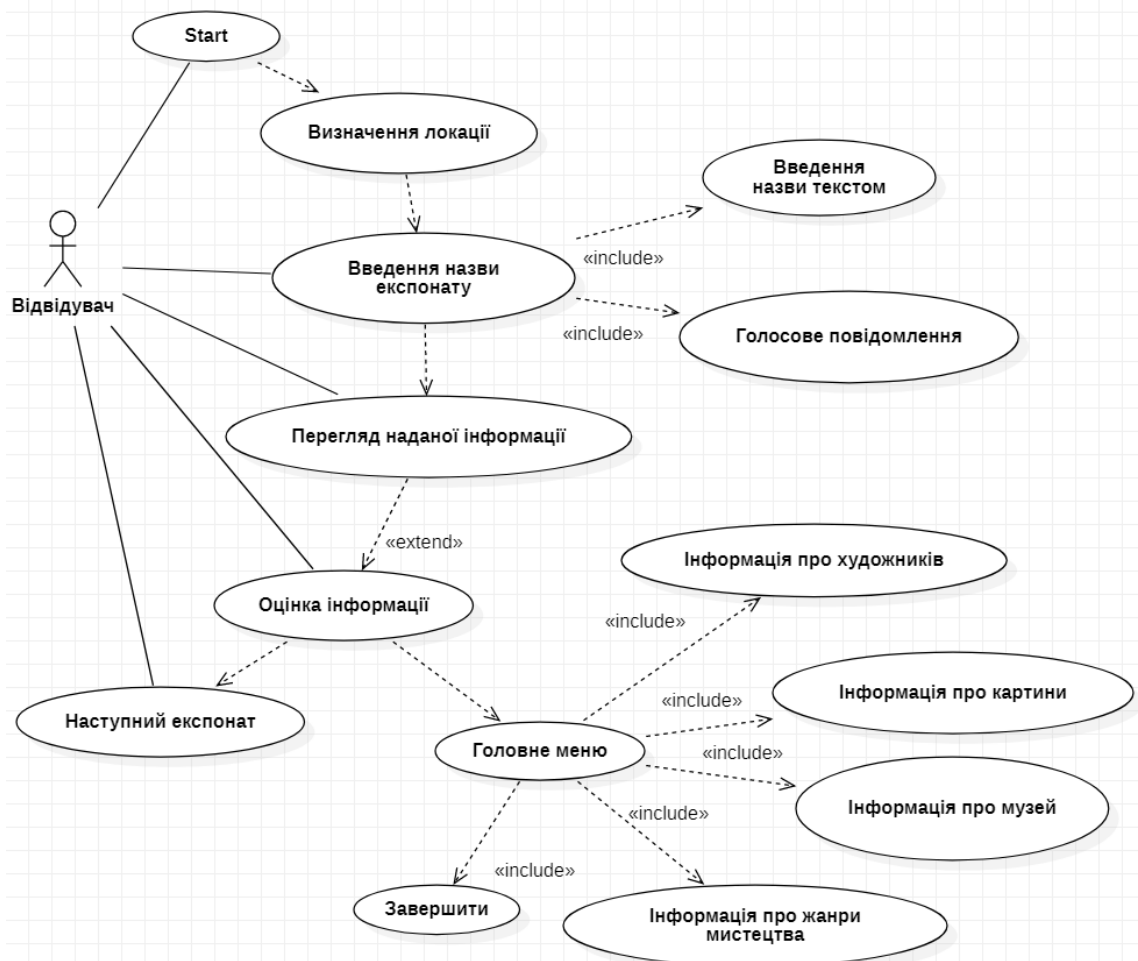


Рисунок 2.1 – Діаграма варіантів використання для користувача чат-боту



Рисунок 2.2 – Діаграма варіантів використання для адміністратора

Для опису варіантів використання застосовується діаграма варіантів використання (use case diagram) - діаграма, що відображує відносини, які існують між акторами і варіантами використання. Вона представляє собою єдиний засіб, що дає можливість замовнику, кінцевому користувачеві і розробнику спільно обговорювати функціональність та поведінку системи.

Розглянемо детальніше наведені варіанти використання:

- Виведення інформації про експонат. Після того, як чат-бот отримав запит від користувача про назву експоната, система повинна знайти ці дані в базі даних та вивести інформацію у зручному для користувача вигляді.
- Враження від експонату. Оскільки критика – це невід’ємна частина мистецтва, в даному випадку чат-бот надає право користувачеві оцінити надану інформацію про експонат. Незалежно від того, якою буде оцінка користувача, в обох випадках після оцінювання система запропонує користувачеві вибір – продовжити оцінювати нові експонати або вивести головне меню бота.
- Головне меню. Отримавши від користувача запит про виведення головного меню, система оброблює запит. На вибір користувача, чат-бот пропонує: переглянути інформацію про художників, картини, жанри мистецтва або ж завершити сесію.
- Інформація про художників. Даний запис виводить з бази даних усю доступну інформацію про художників: загальна інформація, цікаві факти з життя, кількість картин, тощо.
- Інформація про картини. Система надає користувачеві повну інформацію про картину з текстовим описом та зображенням.
- Інформація про жанри мистецтва. Даний запис виводить із системи інформацію про обраний вид мистецтва, також пропонує розглянути художників цього жанру.
- Завершити. Даний запит завершує сесію з чат-ботом.

Після аналізу функціональних вимог ми можемо виділити необхідні сутності, що забезпечать реалізацію програмного комплексу системи.

					ІПЗ.КР.Б – 121 – 20 – ПЗ	
						22

На діаграмі класів (рисунок 2.2) зображені основні сутності системи та їх взаємозв'язки.

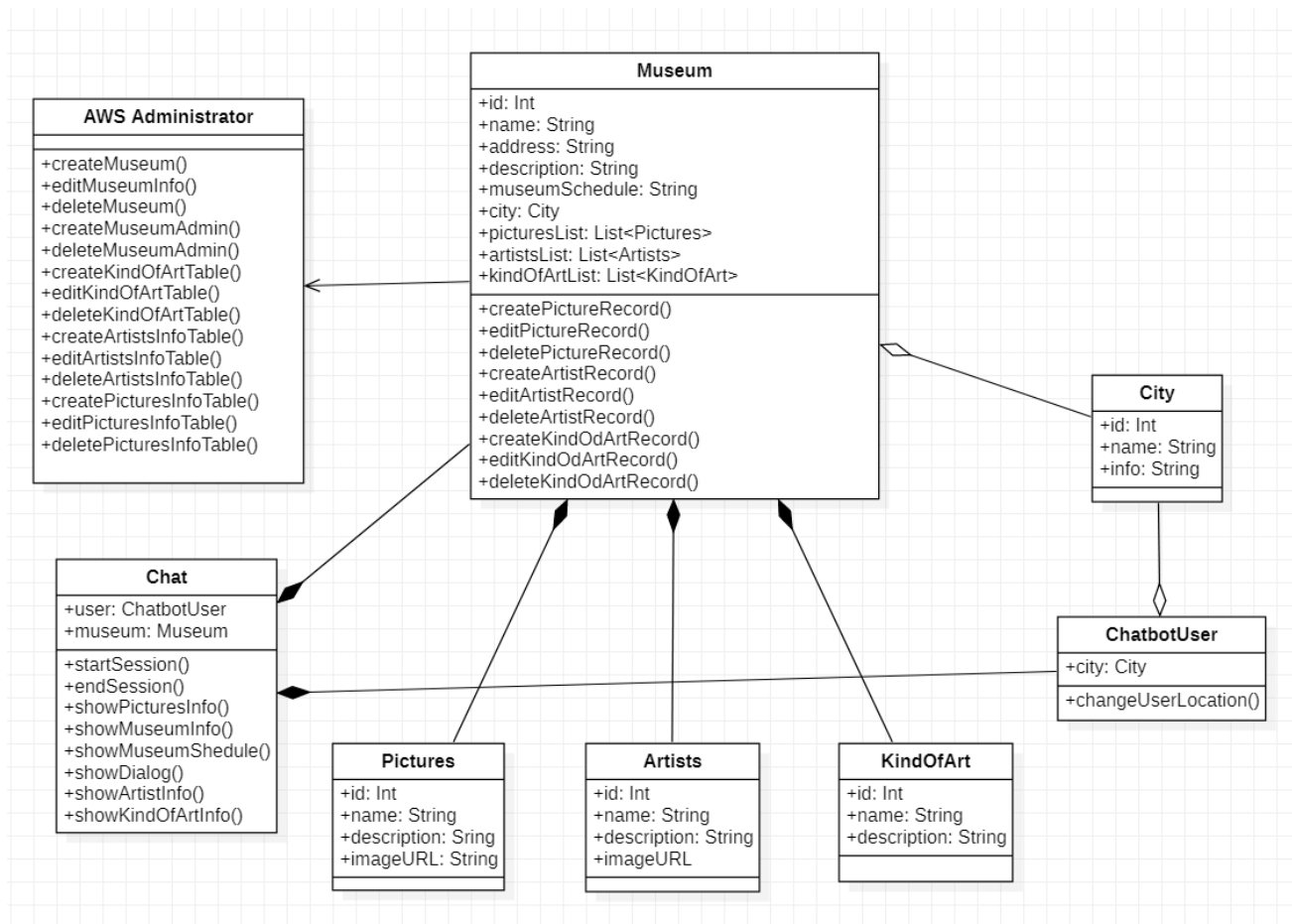


Рисунок 2.2 – Діаграма класів сутностей системи

Діаграма класів це - структурна діаграма мови UML, що показує загальну структуру, ієрархію класів системи, їх кооперації, атрибутів (полів або властивостей), методів, інтерфейсів і взаємозв'язків між ними. Така діаграма застосовується не лише для документування та візуалізації, але й для конструювання за допомогою прямого чи зворотного способів проектування.

Основними елементами діаграми є класи та зв'язки між ними. Класи характеризуються за допомогою атрибутів і операцій. Атрибути показуються щонайменше назвою, також може бути показано їх тип, початкове значення і інші властивості.

Операції (методи) також показуються принаймні назвою, крім того, може бути показано їх параметри і типи значень, які буде повернуто. Операції, як і атрибути, може бути показано з областю видимості.

Виділяють три основних типи зв'язку між класами: композиція, агрегація та успадкування.

Агрегація - асоціація типу «ціле-частина». Агрегація описує, яким чином клас, який грає роль цілого, складається з інших класів, які грають роль частин.

Композиція - асоціація, яка відповідає сильній агрегації, що означає, що частини зв'язку «ціле-частина» не можуть існувати без цілого, а створюються лише у межах цілого.

Наслідування - один з основних принципів об'єктно-орієнтованого програмування, де клас «наслідує» всі атрибути і операції батьківського класу. Також, клас що «наслідує» може перевизначити чи змінювати деякі з батьківських елементів, а також створювати свої операції та атрибути.

2.2. Розробка бази даних та структури файлової системи

2.3. Проектування та реалізація алгоритмів роботи та функціоналу системи

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ СИМУЛЯЦІЇ

3.1. Порядок встановлення та налаштування системи.

3.2. Структура інтерфейсу. Інтерфейс та порядок роботи з системою симуляції.

3.3. Тестування роботи системи

ВИСНОВКИ

					ПЗ.КР.Б – 121 – 20 – ПЗ	
						26

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

					ІІЗ.КР.Б – 121 – 20 – ІІЗ	
						27

ДОДАТКИ

					ПЗ.КР.Б – 121 – 20 – ПЗ	
						28