**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ТЕЛЕГРАМ-ДОДАТКУ СПІЛЬНОЇ ГРИ ПОКЕР МОВОЮ PYTHON»**

Виконав: студент 4 курсу, групи ПД– 41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності)
_____Волощук Р.М._____
(прізвище та ініціали)
Керівник ____Негоденко О.В._____
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)

Київ – 2022

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

_____Негоденко О.В.

“____” _____2022 року

**З А В Д А Н Н Я**

**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА**

Волощук Роман Миколайович

(прізвище, ім’я, по батькові)

1. Тема роботи: «Розробка телеграм-додатку спільної гри покер мовою Python»_____

Керівник роботи: Негоденко Олена Василівна, кандидат технічних наук, доцент _____

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «16» лютого 2022 року №22.

2. Строк подання студентом роботи «01» червня 2022 року

3. Вхідні дані до роботи:

3.1 Науково-технічна література з питань, пов’язаних з програмним забезпеченням щодо розробки Telegram-додатків.

3.2 Офіційна документація мови Python.

3.3 Офіційна документація VisualStudio Code.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Аналіз актуальності та проблематики розроблюваного веб-сервісу.

4.2 Аналіз та вибір інструментів для реалізації продукту.

4.3 Проектування програмного продукту.

4.4 Висновки.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

5.1 Титульний слайд.

5.2 Мета, об’єкт, предмет, ноукова новизна дослідження.

5.3 Аналіз аналогів.

5.4 Технічні завдання.

5.5 Програмні засоби та інструменти реалізації.

5.6 Архітектура системи

5.7 Use case та діаграма класів програми

5.8 Приклад застосування

5.9 Висновки.

5.10 Кінцевий слайд.

6. Дата видачі завдання «11» березня 2022р.

# КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів бакалаврської роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Аналіз предметної галузі | 11.04.2022 | Виконано |
| 2 | Аналіз потреб та вимог потенційної аудиторії майбутнього продукту | 12.04.2022 | Виконано |
| 3 | Підбір науково-технічної літератури | 15.04.2022 | Виконано |
| 4 | Аналіз ринку існуючих інструментальних засобів, їх функціональних недоліків, недоліків з огляду потреб потенційних користувачів продукту. | 18.04.2022 | Виконано |
| 5 | Аналіз та вибір інструментів для розробки додатку | 21.04.2022 | Виконано |
| 6 | Проектування та реалізація | 16.05.2022 | Виконано |
| 7 | Вступ, висновки, реферат | 20.06.2022 | Виконано |
| 8 | Розробка обов‘язкових демонстраційних матеріалів | 29.05.2022 | Виконано |
| 9 | Попередній захист роботи | 01.06.2022 | |
| 10 | Здача роботи | 03.06.2022 | |

Студент _____ Волощук Р.М.

( підпис ) (прізвище та ініціали)

Керівник роботи _____ Негоденко О.В.

( підпис ) (прізвище та ініціали)

# РЕФЕРАТ

Текстова частина бакалаврської роботи 52 с., 23 рис., 15 джерел

Ключові слова: Telegram, Python, покер, телеграм-додаток, чат-бот, Aiogram.

Об’єкт дослідження – комунікація користувачів в процесі гри в покер.

Предмет дослідження — створення телеграм-додатків для забезпечення комунікації користувачів в процесі гри в покер.

Мета роботи — спрощення процедури проведення сеансів гри в покер у групі людей, зібраних онлайн чи офлайн, за рахунок телеграм-додатку у месенджері Телеграм.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Проаналізувати вимоги, потреби, технічні можливості потенційної аудиторії даного програмного продукту, переваги та недоліки існуючих рішень;
2. Провести аналіз інструментів та програмних засобів реалізації, що покажуть найкращу продуктивність системи, під час застосування продукту в реальних умовах, та що спричинять найбільшу зручність під час написання продукту;
3. Спроектувати та розробити телеграм-додаток для створення та проведення сеансів гри в покер серед учасників чату в месенджері Телеграм;
4. Провести тестування телеграм-додатку та ігрової системи на виявлення помилок в роботі.

Розроблено систему для гри в покер мовою Python з використанням для взаємодії з системою месенджеру Телеграм та його можливості імплементування чат-ботів. Для зберігання даних використовується MongoDB.

**ЗМІСТ**

ВСТУП ..... 9

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ..... 11

1.1 Загальний аналіз родини карткових ігор Покер ..... 11

1.2 Аналіз правил та особливостей настільної гри в покер (Техаський холдем) ..... 14

1.3 Аналіз потреб потенційної цільової аудиторії продукту ..... 18

1.4 Аналіз існуючих інструментальних засобів для проведення гри в покер ..... 20

1.5 Огляд засобів реалізації продукту для створення та проведення сеансів гри в покер ..... 25

РОЗДІЛ 2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ..... 34

2.1 Огляд мови програмування ..... 34

2.2 Огляд середовища розробки ..... 38

РОЗДІЛ 3 ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ 41

3.1 Проектування діаграми послідовностей ..... 41

3.2 Проектування внутрішньої будови класів ..... 43

3.3 Розробка системи розрахунку переможця ..... 48

3.4 Тестування програмного продукту ..... 50

ВИСНОВКИ ..... 57

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ..... 59

Додаток А ..... 61

ПРЕЗЕНТАЦІЯ ДО ЗВІТУ ..... 61

# ВСТУП

***Актуальність теми.*** В сучасному світі комунікація є одним з ключових способів от римання та передачі інформації. Сучасний підхід до вдосконалення методів комунікації вимагає достовірної, своєчасної та повної інформації, яку можна отримати шляхом впровадження новітніх інформаційних технологій. Такі технології сприяють розвитку та ефективному використанню електронних джерел інформації, дозволяючи не лише оперувати застарілою інформацією, але й ефективніше володіти останніми відкриттями та результатами.

Впровадження сучасних інформаційних технологій (IT), застосування інформаційних систем у різних  галузях життя забезпечують принципово новий рівень засвоєння й узагальнення інформації, поширення та її використання.
Для виходу комунікацій на більш новітній рівень важливим є покращення такої функції як мобільність.

Мобільність дозволяє комунікаціям бути більш пластичними, гнучкими, та не мати прив’язку до місця розташування. А, враховуючи сучасні реалії, для людей це пріорітетна функція комунікацій.

Саме тому для людей дуже важливим є можливість проведення свого вільного часу онлайн. За останні роки попит на онлайн ігри, серед яких є онлайн покер надзвичайно зріс.

Ігри, в котрих ключовим елементом є ігрова валюча, в Інтернеті мають величезні переваги у зручності перед фізичними завдяки електронному, а не звичайному управлінню. У випадку з покером, великі пули гравців і пов’язані з ними мережеві ефекти також допомогли грі досягти такого розміру, якого немає в офлайн-світі.

Об’єкт дослідження – комунікація користувачів в процесі гри в покер.

Предмет дослідження — створення телеграм-додатків для забезпечення комунікації користувачів в процесі гри в покер.

Мета роботи — спрощення процедури проведення сеансів гри в покер у групі людей, зібраних онлайн чи офлайн, за рахунок телеграм-додатку у месенджері Телеграм.

Для досягнення поставленої мети слід виконати наступні завдання:

- Провести аналіз предметної галузі та існуючих рішень;
- Провести аналіз аналіз потреб та вимог потенційної аудиторії майбутнього програмного продукту;
- Обрати середовище розробки;
- Розробити робочий прототип додатку;
- Провести тестування системи.

*Практичне значення* отриманих результатів полягає у розробці функціоналу для роботи API Telegram з використання бібліотеки Aiogram та мови програмування Python.

# РОЗДІЛ 1
# АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Загальний аналіз родини карткових ігор Покер

Покер — це родина спортивних карткових ігор, у яких гравці роблять ставку на те, яка комбінація найкраща відповідно до правил цієї конкретної гри. Хоча перший відомий варіант гри використовував лише 20 карт, у наші часи більш типово використовувати звичайну колоду. Однак у регіонах, де короткі колоди зустрічаються частіше, у гру можна грати з 32, 40 або 48 картами замість стандартної колоди. Таким чином, ігри в покер можуть відрізнятися за дизайном колоди, кількістю карт у грі, кількістю карт, що вручаються лицевою стороною вгору або обличчям вниз, а також числом карт, яке використовується всіма гравцями; тим не менш, правила всіх покерних ігор включають принаймні одне спільне правило для більш ніж одного раунду ставок.

Покер набув популярності з початку 20-го століття і перетворився з розважальної діяльності, обмеженої невеликими групами ентузіастів, до широко популярної діяльності як для учасників, так і для глядачів, включаючи онлайн, з багатьма професійними гравцями та багатомільйонними призами турнірів.

Витоки покеру були предметом бурхливих дискусій. Можливо, назва гри походить від французького poque, яке, у свою чергу, походить від німецького pochen (що дослівно перекладається як «стукати»), хоча неясно, чи лежать витоки самої назви покер в іграх. які носять це ім’я. Це дуже схоже на гру під назвою Nas, в котру грають у Персії, і цілком можливо, що перські моряки були тими, хто представив її французьким іммігрантам у Новому Орлеані. Спільнотою часто вважається, що покер походить від гри Primero, яка була популярна в епоху Відродження, а також французької гри Brelan. Англійська карткова гра, відома як brag (раніше називалася bragg), безпомилково походить від brelan і включала елементи блефу (хоча на той час ця концепція була відома в інших іграх). Цілком

можливо, що будь-яка з цих попередніх ігор мала великий вплив на еволюцію покеру в тому вигляді, в якому всі знають його сьогодні.

Рисунок 1.1.1 – Сеанс гри в покер, ілюстративно.

Покер має багато варіацій, всі вони відповідають схожій схемі гри і загалом використовують ту саму ієрархію рейтингу комбінацій. Існує чотири основні сімейства варіантів, які в основному згруповані за умовами карткової роздачі та ставок:

1. Стріт

Кожному гравцеві роздається повна роздача, тобто п’ять карт, і гравці роблять ставку в одному раунді, з можливістю підвищення та повторного підвищення. Це найстаріша покерна родина. Корінь гри, якою зараз грали, була гра, відома як Primero, яка перетворилася на гру з трикарткою, дуже популярну джентльменську гру під час Війни за незалежність США, яка користується попитом у Великобританії і сьогодні.

2. Стад покер

Карти роздаються у заздалегідь обумовленому поєднанні, картами, що роздаються лицем вгору та картами, що роздаються лицем вниз, після кожного раунду ставок. Це наступна за віком родина. У міру того, як покер прогресував від трьох до п’ятикарткових роздач, їм часто роздавали одну карту за раз, або обличчям вниз, або обличчям вгору, з раундом ставок між кожною. Найпопулярніший сьогодні варіант стад, семикартковий стад, де Дилер роздає кожному гравцеві дві додаткові карти (три обличчям вниз, чотири обличчям вгору), з яких вони повинні скласти найкращу можливу комбінацію з 5 карт.

3. Дро-покер

Розіграш п’яти карт: кожному гравцеві роздається повна роздача обличчям вниз. Потім кожен гравець повинен поставити ставку до банку. Потім вони зможуть побачити свої карти і зробити ставку відповідно. Після ставки гравці можуть скинути до трьох карт і взяти нові з верхньої частини колоди. Потім відбувається ще один раунд ставок. Врешті, кожен гравець повинен показати свої карти, і гравець з найкращою комбінацією виграє.

4. Покер зі спільними картами

Також відомий як «флоп-покер», покер із спільними картами є різновидом стад-покеру. Гравцям роздається неповна комбінація карт, а потім у центр столу роздається кілька спільних карт, кожна з яких може використовуватися одним або кількома гравцями, щоб скласти комбінацію з п’яти карт. Техаський холдем і Омаха - це два найбільш відомі варіанти сімейства покеру зі спільними картами, та, загалом, всього покерного ринку.

Зараз, коли у світі говориться про покер, зазвичай, мається на увазі покер сімества покеру зі спільними картами, а саме Техаський Холдем (Texas Hold’em)

Здебільшого гральною валютою у грі не виступають справжні гроші. Для ставок використовуються фішки (англ. Chips). Фішки різних кольорів мають різний номінал встановлений покерним клубом чи виробниками покерного набору.

Рисунок 1.1.2 – Покерні фішки. Гральна валюта у грі

## 1.2 Аналіз правил та особливостей настільної гри в покер (Техаський холдем)

В покері існують різні столи за якими грають гравці. Вони відрізняються сумами Смол Блайнд та Біг Блайнд. Це суми які по суті позначають «бюджет» гравців. Смол Блайнд та Біг Блайнд це обов’язкові ставки, що ставляться автоматично на початку кожного раунду двома гравцями зі столу.

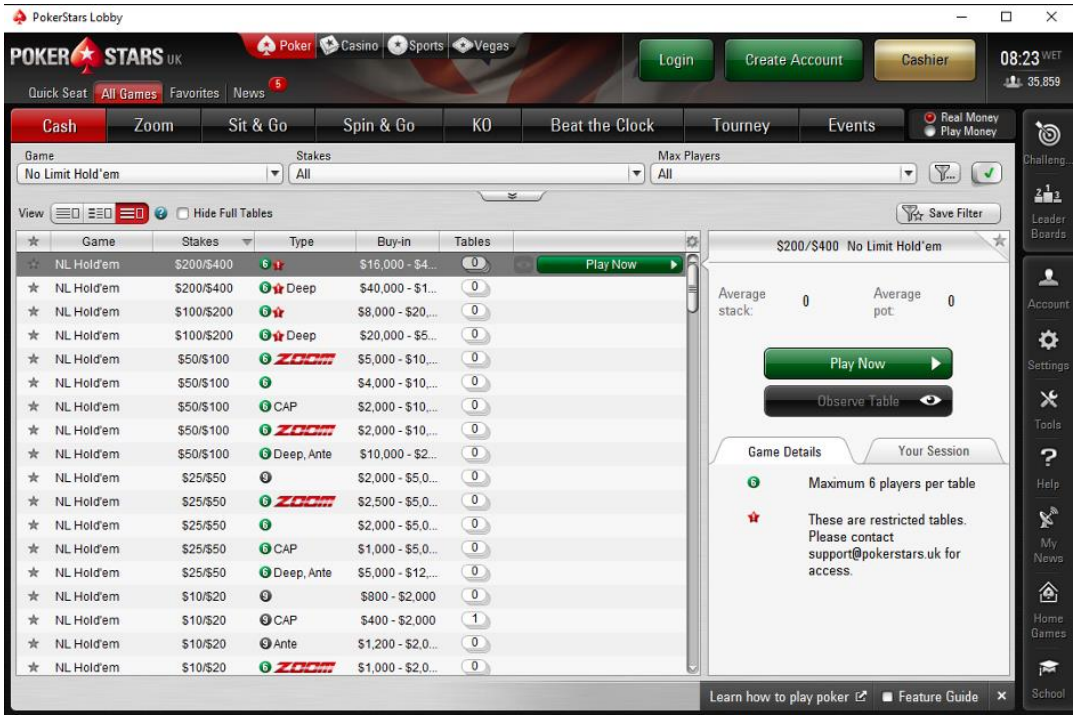



Рисунок 1.2.1 – Інтерфейс вибору стола з різними Смол Блайнд та Біг Блайнд на прикладі найпопулярнішого онлайн-покеру PokerStars.com.

Сеанс починається з заходу гравців за стіл. Щоб сісти за стіл вони мають мати при собі як мінімум суму (у фішках) десятикратну сумі Біг Блайнд. На турнірах гра продовжується до тих пір, поки всі фішки не дістануться одному гравцеві. В звичайному ж покері гра кінця не має, грається доти, доки хоч два гравці є за столом.

До того, як почнуть роздавати карти, з двох гравців за столом беруть початкові обов’язкові ставки Small Blind та Big Blind. Вони потрібні для того, щоб прискорювати процес гри, щоб гравці попросту не чекали щораунда карт, які би їм підійшли.

Людина, що роздає карти в покері, називається Дилер. Це може бути або окрема наймана для цього людина, або по черзі кожен з гравців. Існує багато різновидів покеру, проведемо огляд лише найпопулярнішого Texas Hold’em. У Hold’em покері роздається по дві карти кожному гравцеві. Ніхто крім гравця не

повинен бачити його карти, вони роздаються лицем вниз. Після роздачі карт слідує чотири раунди торгів:

1. Preflop
2. Flop
3. Turn
4. River

Рисунок 1.2.2 - Карти гравця, карти на столі під час торгів Flop

Гравець виграє банк (всі суми які поставили гравці) в двох випадках: за підсумком торгів у гравця краща комбінація карт (в покері комбінація карт відома як «рука»); інші гравці скинули карти, не бажаючи приймати ставку гравця.

Всього комбінацій карт існує десять. Вони завжди складаються з п’яти карт, що доступні в наборі гравцеві, тобто дві карти на руках особисті, та спільні п’ять карт на столі. Так комбінації виглядають у порядку старшості, тобто, остання в списку краща за першу:

1. Вища карта. Найслабша рука. Жодної з перерахованих нижче комбінацій у гравця немає. Комбінацією є п’ять найвищих доступних гравцеві карт. Безпосередньо вищою картою називається найвища серед цих п’яти.

2. Пара. Гравцеві доступні дві карти одного рангу (дві п’ятірки, два королі). Комбінацією є п’ять карт, серед яких дві карти одного ранку та три найвищі серед решти.

3. Дві пари. Комбінація, в якій є дві пари карт одного рангу одночасно. П’ятою картою є найвищою серед решти доступних.

4. Сет (трійка). Три карти одного рангу та дві найвищі серед решти доступних.

5. Стріт. П’ять карт довільної масті, чий ранг йде по порядку (наприклад від вісімки до дами).

6. Флеш. П’ять карт довільного ранку чия масть однакова.

7. Фул-Хаус. Одночасно Сет та Пара.

8. Каре. Комбінацією є чотири карти одного рангу та найвища серед решти доступних гравцеві.

9. Стріт-Флеш. П’ять карт однієї масті, чий ранг йде по порядку.

10. Роял флеш, або Флеш-рояль. Найвища та найрідкісніша рука. Комбінація складається з п’яти карт однієї масті підряд від десятки до туза (Стріт-Флеш з рангом карт від 10 до туза).

Рисунок 1.2.3 – Приклади кожної комбінації карт.

Перший раунд торгів Preflop полягає в тому, що гравці ставлять ставки, виходячи з карт, що мають на руках після роздачі. Гравець повинен аналітично оцінити шанси на виграш знаючи лише дві доступні йому карти.

Під час торгів у гравців на вибір є чотири варіанти ходу: Fold (скинути карти), Bet/Raise (підвищити ставку), Check (пропустити хід), Call (підтвердити ставку). Ходять гравці по черзі. Якщо хтось з гравців поставив ставку, то для того, щоб продовжити гру в раунді треба її підтвердити, тобто поставити таку ж суму (Call). Звичайно гравець може і збільшити наявну на столі ставку (Raise). Пропустити хід (Check) можна тільки якщо ніхто з гравців, хто ходив до Вас не поставив ставку в цьому раунді торгів. Скинути карти (Fold), тобто, не продовжувати гру до наступної роздачі карт, можна завжди, але не рекомендується цього робити, коли доступний пропуск ходу (Check), так як в наступних раундах торгів у гравця може перевернутись ситуація на виграшну, а від пропуска ходу гравець нічого не втрачає.

Наступний раунд торгів Flop полягає в тому, що на стіл Дилер викладає три карти з колоди лицем вгору. Так як п’ять із семи доступних гравцеві карт вже йому відомі, ставки в цьому раунді вирішують як складеться гра надалі.

Наступні раунди торгів Turn та River однакові. На стіл Дилер викладає одну карту лицем вгору, гравці ходять по черзі. Торги закінчуються тоді, коли кожен з гравців, в яких є карти зробить хід та ставки у всіх гравців будуть рівними.

По закінченню чотирьох раундів торгів гравці перевертають свої карти та порівнюють комбінації. Той, в кого «рука» краще забирає всі ставки, що ставились у грі (банк).

### 1.3 Аналіз потреб потенційної цільової аудиторії продукту

Основна і, певно, єдина категорія цільових користувачів це компанія друзів, знайомих, чи родичів, що шукають варіант проведення вільного вечору, години. В ході аналізу аудиторії було зрозуміло хід думок та потреб потенційних користувачів. Ці потреби було поділено на чотири категорії:

1. азартна складова;
2. технічні можливості;
3. доступність;
4. ціль процесу гри.

Хоча покер і набрав свою популярність за рахунок того, що люди, граючи в нього, могли заробити одразу велику суму грошей, але бажання грати в покер мають не лише азартні люди. Одна з найбільших властивостей саме покеру полягає в тому, що гравець вчиться розуміти думки іншого гравця. Часто для виграшу в раунді є потреба розпізнати, чи опонент дійсно має причину для високої ставки, чи він попросту блефує. Саме тут до гри підключається логічне мислення, інтуїція та, навіть, психологія. За це багато людей, в тому числі і я, полюбляють цю настільну гру. Тож, з’явилась потреба в майбутньому продукті мінімізувати азарт, залежність до гри в покер.

Рисунок 1.3.1 - Залежна азартна людина, ілюстративно

У ході аналізу технічних можливостей потенційних користувачів мого продукту було виявлено те, що у більшості немає бажання виділяти якусь кількість

пам’яті на пристрої, що планується використовуватись для гри, в яку вони не так часто будуть заходити, так як можливості для зустрічей з друзями обмежені. Тому з’явилась інша потреба: обмежити до мінімуму розмір додатку, або, що краще, розмістити його повністю на сервері.

Іншим параметром для вибору платформи була доступність. Одною з потреб стала можливість роботи додатку на всіх можливих платформах, можливість продовжити гру з іншого пристрою. Також одним із потенційних місць проведення гри є гра під час відпочинку на природі. Тобто, з’явилась ще потреба підтримки місць з обмеженим доступом в мережу Інтернет. Інша потреба у доступності була в тому, щоб користувачі мали можливість користуватися продуктом зі своїх старих пристроїв. Тобто, з’явилась потреба у підтримці широкого діапазону операційних систем.

У ході аналізу потреб користувачів, одним із пунктів опитування було запитання про бажаний настрій гравців, про атмосферу під час гри. У більшості відповіддю була атмосфера, що не зобов’язує, неспішна. Тобто час на хід повинен бути необмежений, щоб ціллю гри стало більше спілкування, аніж виграш. Також продукт має бути однаковий для зручності як для людей, що зібрались в одному приміщенні для гри, так і для людей, що знаходяться на відстані, але спілкуються, до прикладу, онлайн на відео-конференції.

## 1.4 Аналіз існуючих інструментальних засобів для проведення гри в покер

Покер – одна з найпопулярніших ігор у світі, тому на ринку є велика кількість готових крупних проектів. У ході аналізу цього ринку було виявлено кілька основних категорій існуючих засобів :

- наявний фізичний ресурс;
- офлайн-додатки;
- онлайн-додатки.

Наявний фізичний ресурс для карточних ігор та, загалом, настільних ігор це перше, що спадає на думку, навіть якщо ці ігри пов’язані з онлайн-казино, яке в сучасні дні має більшу популярність, ніж відвідування фізичних закладів казино.

Ще нещодавно знайти фізичне місце, де проводились ігри та турніри в покер, було складно. Азартні ігри до серпня 2020 року були заборонені в Україні. Тому місця, де проводились ігри в покер, повинні були бути безкоштовними, щоб виключити варіант виграшу та програшу грошей учасників. Так як власники таких приміщень не отримували тих грошей які зазвичай отримують власники казино, таких місць ставало все менше і менше. З серпня ж 2020 року в Україні можуть легально працювати казино, зали гральних автоматів, онлайн-казино та букмекерські пункти. Вже через місяць після прийняття цього закону готелі в різних містах України отримали перші дозволи на розміщення казино та залів ігрових автоматів.

Суть таких закладів в тому, що вони продають фішки з якими гравці сідають за стіл, та ними грають, їх виграють та програють. Чесні заклади заробляють тільки виконуючи роль обмінного пункту, де існують комісії та різні курси для покупки фішок. Нечесні заклади часто маніпулюють результатами для своєї вигоди.

Звідси виходить і основна проблема офлайн-казино для потенційних користувачів мого вирішення. Мала популярність та велика складність знайти місце проведення ігор в покер не на гроші, тобто без азартної складової цієї популярної гри.

Іншим варіантом є самостійне проведення сеансів гри в покер з друзями. У вільному доступі зараз продаються гральні фішки в магазинах настільних ігор, та іноді в супермаркетах. Недоліком такого варіанту є лише те, що у потенційних користувачів мого вирішення не завжди присутні карти та фішки.  Але саме такий варіант я візьму в якості еталонного для створення свого рішення в віртуальному просторі. У гравців є вибір в кількості фішок, що роздаються гравцям, можливість розпочати гру заново, відсутність азартної складової цієї гри, відсутність обмежень в часі.

Рисунок 1.4.1 - Покерний набір

Наступна категорія існуючих засобів це офлайн-додатки. У офлайн ігор в покер є очевидний мінус для потенційної аудиторії мого продутку. У більшості таких додатках гравцю доведеться грати з комп’ютером. Хоч і деякі з застосунків підтримують мультиплеєр (багатокористувацький), але навіть це унеможливлює гру для компанії, що зібралась не в одному приміщенні, а, до прикладу, на відео-конференції.

Онлайн-додатки це найпопулярніший нині вид проведення сеансу гри в покер. Їх є десятки тисяч, із яких переважна більшість по структурі однакова: режим пробний, тобто ігри на уявну валюту, яку теж потрібно заробляти, але дивлячись рекламу, виконуючи завдання, тощо; режим класичний, тобто ігри на справжні гроші; необхідність в реєстрації; обмежений час на хід.

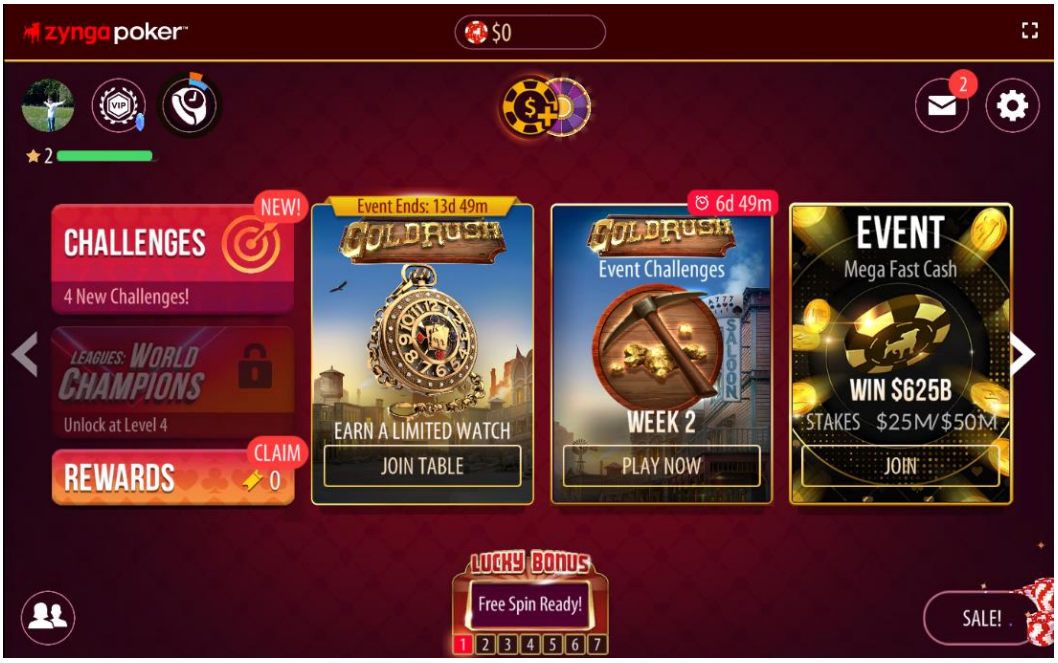



Рисунок 1.4.2 - Інтерфейс найпопулярнішого додатку для гри в покер лише на уявну валюту Zynga Poker.

Під час аналізу функцій додатку, було виявлено, що створити гру з власними налаштуваннями неможливо. Тобто, хоч і гра доступна на великій кількості платформах, вона не підходить потенційним користувачам мого продукту за факторами доступності, цілі процесу гри та фактором технічних можливостей.

Після детального дослідження ринку був виявлений не популярний, але найбільш підходящий потенційній аудиторії готовий продукт. Він називається EasyPoker. Це додаток, за допомогою якого будь-який користувач може створити віртуальну кімнату для гри в покер, налаштувати її та запросити інших гравців. Із виявлених мінусів є наявність платного функціоналу в додатку доступного за підпискою. Без підписки ж у користувача немає можливості відключити спливаючу рекламу, налаштувати гру, тобто встановити кількість фішок, відключити таймер ходу. Також до мінусів можна віднести можливу складність гри з поганим інтернет з’єднанням, необхідність завантажувати додаток. Також із підтримуваних

платформ лише Android та iOS, і ті потребують збірок систем, випущених мінімум 2016 року для Android та 2015 року для iOS.

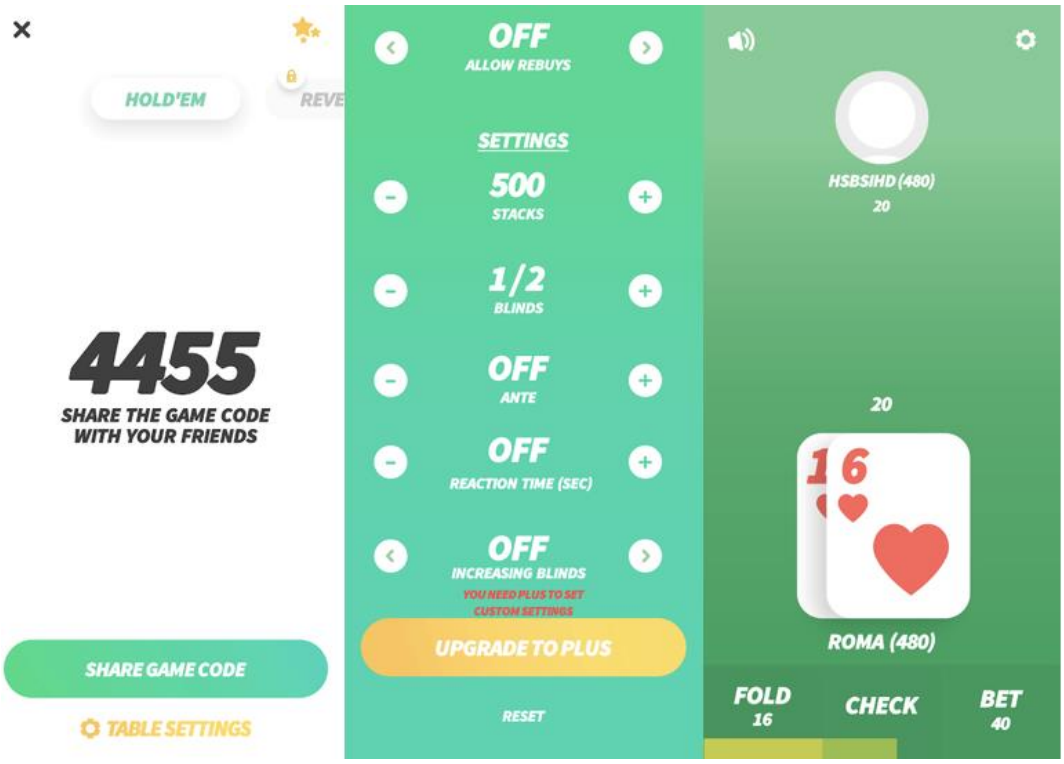



Рисунок 1.4.3 - Інтерфейс додатку EasyPoker

Таблиця 1.4.1 - Зведення результатів аналізу переваг та недоліків онлайн-додатків для проведення сеансів гри в покер.

| Назва | Посилання | Переваги | Недоліки |
|---|---|---|---|
| PokerStars | pokerstars.com | Найбільша аудиторія, можливість створення гри з власними налаштуваннями, багато платформ, що підтримуються | Необхідність наявності стабільного з'єднання, реєстрації, завантаження додатку |
| Zynga Poker | zyngapoker.com | Багато платформ, що підтримуються, гра без реєстрації | Неможливість створення власної гри, неможливість |

| | | | |
|---|---|---|---|
| | | | відключення таймера ходу, |
| EasyPoker | easy.poker | Створення особистої гри, гра без реєстрації, зручний інтерфейс для мобільних пристроїв | Велика доля функціоналу платна |

### 1.5 Огляд засобів реалізації продукту для створення та проведення сеансів гри в покер

Продукт для проведення сеансів гри в покер розробляється для мобільних платформ та настільних комп'ютерів. Вибір мови програмування для розробки продукту залежить від вибору типу додатку, або додатків. Виділимо наступні сучасні види додатків для операційних систем:

– нативні додатки – додатки, розроблені безпосередньо під вибрану платформу, вони зазвичай запускаються без інтернет з'єднання. У розробці нативних додатків офіційна бібліотека для розробки підтримує весь доступний системою функціонал апаратного забезпечення. Нативні додатки є найбільш оптимізованими додатками.

– веб-додатки – додатки, розроблені для інтернет-браузера, обробляють інформацію на сервері, не можуть працювати без інтернету. Так як інтернет-браузер підтримують всі сучасні операційні системи, такі додатки з великою ймовірністю зможуть запуститись на всіх популярних платформах.

– гібридні додатки – додатки, написані для інтернет-браузера, але за допомогою різних технологій можуть завантажуватись на кшталт нативних додатків. Вони також можуть бути завантажені з локального магазину застосунків, працювати без інтернету.

– інші додатки – додатки, створені під готову платформу для роботи на ній. Це можуть бути як плагіни, для популярних застосунків, так і повноцінні програми, що запущені на серверах та приймають зворотній зв'язок з платформ на яких ними користуються.

Таблиця 1.5.1 – Зведення результатів аналізу видів додатків для задоволення потреб користувачів.

| Варіант | Необхідність у завантаженні | Процесинг на сервері | Багатоплат-формність | Необхідність оновлення додатку після програмних змін |
|---|---|---|---|---|
| Нативний додаток | + | - | - | - |
| Веб-додаток | - | + | + | - |
| Гібридний додаток | + | - | + | + |
| Телеграм бот | - | + | + | - |

У даному продукті буде застосована саме остання технологія розробки додатків. Платформа, що буде використана, доступна на великій кількості платформ: Telegram. Оскільки Телеграм, як відомо, потребує мінімального інтернет з’єднання, встановлений уже на більшості мобільних пристроїв молоді України, він доцільно підходить під потреби потенційної аудиторії. Платформа є засобом спілкування з підтримкою автоматизованих допоміжних чат-ботів, створених спільнотою. В якості мови програмування було вибрано найбільш функціональну та найпростішу мову для написання серверів для Телеграм ботів Python.

**Telegram**

Телеграм – безкоштовний крос-платформний, хмарний сервіс миттєвого обміну повідомленнями. Сервіс також надає end-to-end зашифроване з’єднання відео-зв’язком, VoIP, спільне використання файлів і кілька інших функцій. Він був запущений для iOS 14 серпня 2013 і Android 20 жовтня 2013. Сервери Telegram розповсюджені по всьому світу з п'ятьма центрами обробки даних в різних

регіонах, в той час як операційний центр базується в Дубаї в об'єднаних Арабських Еміратах.

Рисунок 1.5.1 – Логотип Телеграм

Клієнтські додатки доступні для настільних і мобільних платформ включаючи офіційні додатки для Android, iOS, Windows, macOS і Linux (Хоча для реєстрації потрібен пристрій iOS або Android і робочий номер телефону).Також є два офіційних веб-додатки Telegram, WebK і WebZ, і численні неофіційні клієнти, які використовують протокол Telegram. Всі офіційні компоненти та додатки Telegram є додатками з відкритим сирцевим кодом, за винятком пропрієтарного сервера, код якого закритий.

Telegram також надає додаткові end-to-end зашифровані бесіди, тобто повідомлення не будуть зберігатись на серверах Телеграму, та можуть бути прочитаними тільки на двох пристроях: що відправляв повідомлення та той, що приймав його. Хмарні чати і групи шифруються між додатком і сервером, так що ISP і інші треті сторони в мережі не можуть отримати доступ до даних, але сервер Telegram може. Користувачі можуть надсилати текстові та голосові повідомлення, здійснювати голосові та відеодзвінки, а також обмінюватися необмеженою кількістю зображень, документів (2 ГБ на файл), локацій користувачів, анімованих та не анімованих стікерів, контактів та аудіофайлів. Користувачі також можуть слідкувати за каналами, які, до прикладу, використовували такі знаменитості, як

Арнольд Шварценеггер і політики, найбільш примітним є канал президента України Володимира Зеленського.

У січні 2021 року Telegram перевершив 500 мільйонів активних користувачів щомісяця. Це був найбільш завантажений додаток по всьому світу в січні 2021 року з 1 мільярдами завантажень по всьому світу станом на кінець серпня 2021 року.

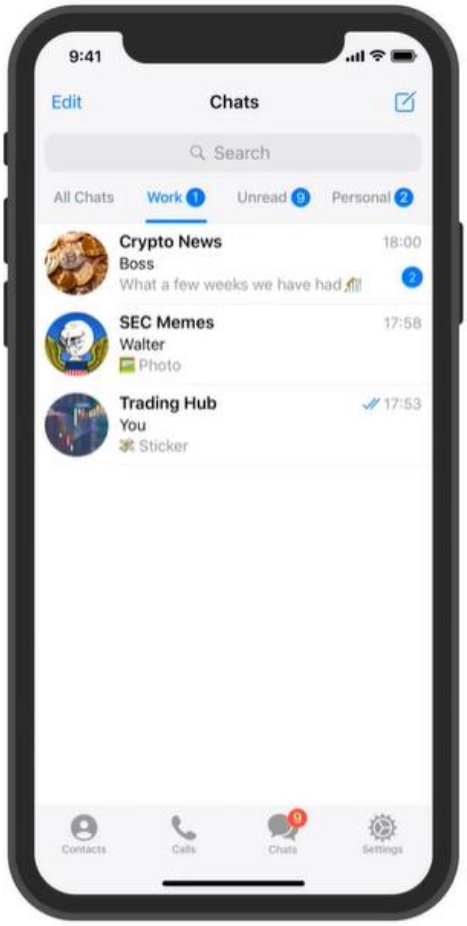



Рисунок 1.5.2 – Інтерфейс додатку Телеграм на платформі iOS

Telegram має загальнодоступний API, за допомогою якого розробники можуть отримати доступ до тих же функцій, що й офіційні програми Telegram, для створення власних програм обміну повідомленнями. У лютому 2015 року творці неофіційного клієнта Whatsapp+ випустили додаток Telegram Plus, пізніше перейменований на Plus Messenger, після того, як їхній оригінальний проект

отримав наказ про припинення піддтримки і відмову від WhatsApp. У вересні 2015 року Samsung випустила програму для обміну повідомленнями на основі цих API.

Telegram також пропонує API, який дозволяє розробникам створювати ботів, які є обліковими записами, контрольованими програмами. Такі боти використовуються, серед іншого, навіть, для емуляції та відтворення старих ігор у програмі або інформування користувачів про наявність вакцини від COVID-19.

Крім того, Telegram пропонує функції для здійснення платежів безпосередньо на платформі, поряд із зовнішнім сервісом, таким як Stripe.

**Телеграм боти**

У червні 2015 року Telegram запустив платформу для сторонніх розробників для створення ботів. Боти — це облікові записи Telegram, якими керують програми. Вони можуть відповідати на повідомлення чи згадки, можуть бути запрошені в групи та можуть бути інтегровані з іншими програмами. Боти також можуть приймати онлайн-платежі, здійснені за допомогою кредитних карток або Apple Pay. Існують також inline-боти, якими можна користуватися з будь-якого екрана чату. Щоб активувати inline-бота, користувач повинен ввести ім’я користувача бота та запит у поле повідомлення. Після цього бот запропонує свій вміст. Користувач може вибрати з цього вмісту та надіслати його в чаті.

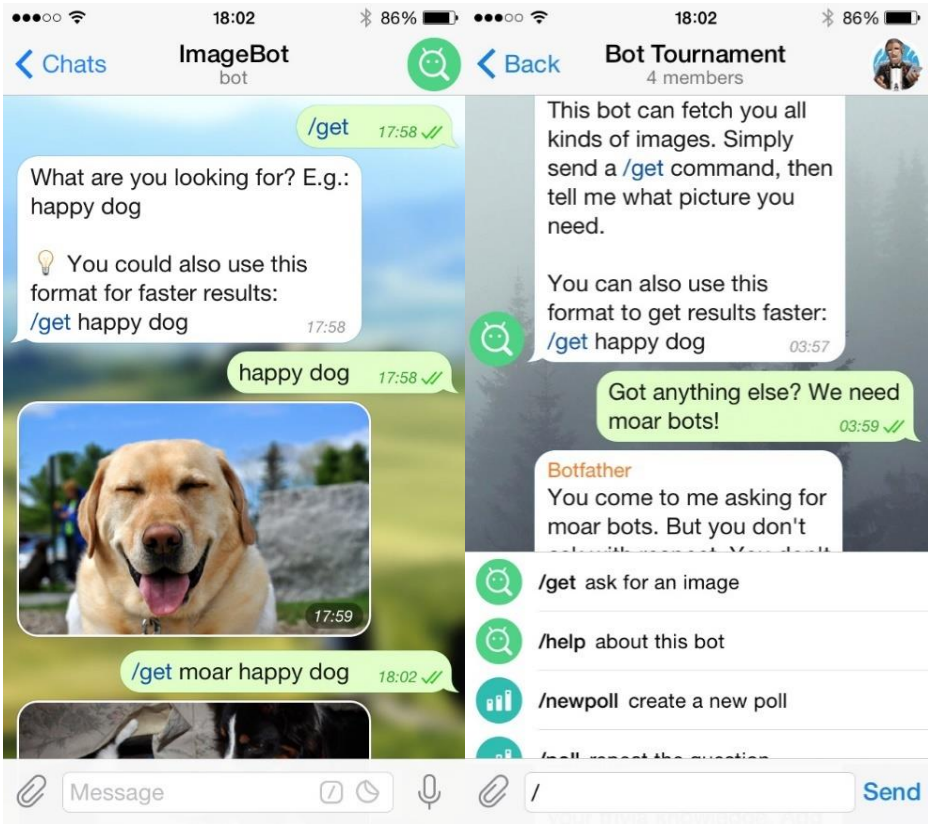



Рисунок 1.5.3 – Інтерфейс ботів в додатку Телеграм

Платежі, здійснені за допомогою Paymentwall, Yandex.Money, Stripe, Ravepay, Razorpay, QiWi та Google Pay, також можуть оброблятися ботами. Ці послуги доступні в багатьох різних країнах. Ігрова платформа Telegram, що може бути запущена через Телеграм ботів підтримує HTML5, тому ігри можна завантажувати так само, як і стандартні веб-сторінки. Ігри з ігрової платформи сумісні з iPhone 4 і пізнішими моделями, а також Android 4.4 і новіше.

Завдяки впровадженню IFTTT користувачі можуть використовувати сервіси Інтернету речей (IoT), які підтримують двосторонню взаємодію в Telegram.

Функція соціального входу, яка була розроблена Telegram і стала доступною для користувачів у лютому 2018 року, отримала назву Telegram Login. Він включає в себе віджет веб-сайту, який можна вставляти на веб-сайти та дає змогу користувачам входити на сторонній веб-сайт за допомогою свого облікового запису Telegram. Цей віджет можна розмістити на веб-сайтах. Шлюз відповідає за

передачу імені в Telegram, імені користувача та зображення профілю користувачів власнику веб-сайту; однак номер телефону користувачів залишається приватним.

**Python**

Python — одна з найпопулярніших мов програмування. За даними популярного серед програмістів ресурсу Stackoverflow, Python входить до трійки найпопулярніших та затребуваних мов протягом 5 останніх років. Це пов'язують з його універсальністю і простотою синтаксису. Мова відрізняється мінімальним набором символів і проста для вивчення та використання щодо інших мов. В силу універсальності дозволяє виконувати різну роботу: писати плагіни та скрипти, розробляти ігри, мобільні програми, онлайн-сервіси, супроводжувати ПЗ. Python — універсальна мова програмування, але найбільш застосовувана у наступних напрямках:

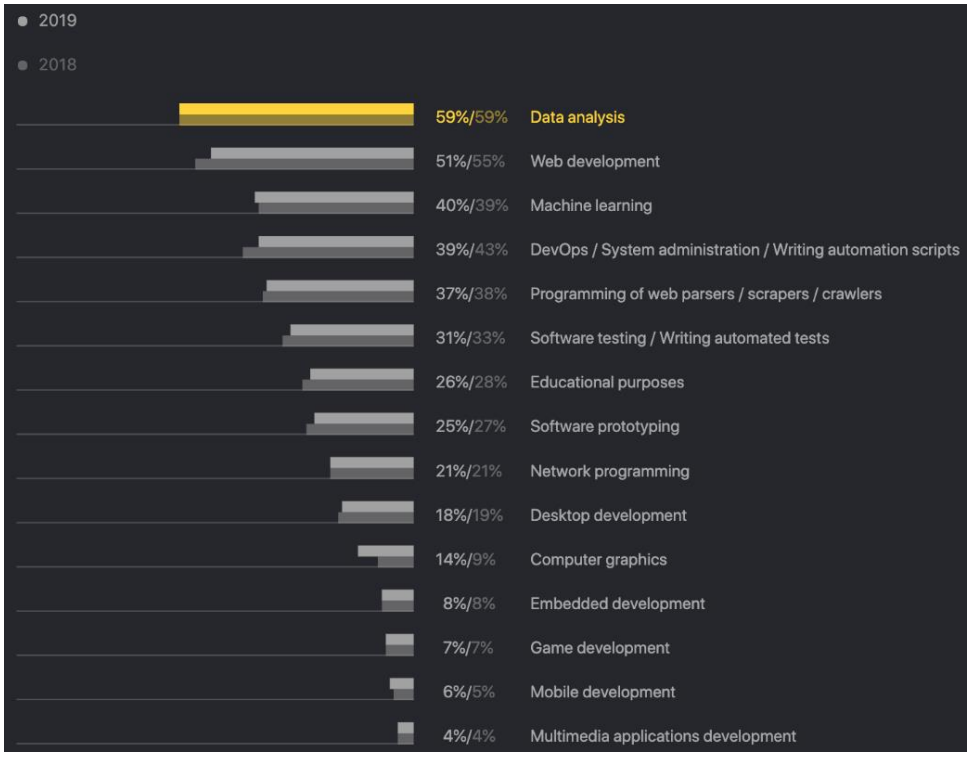



Рисунок 1.5.4 – Напрямки використання мови Python за даними опитування JetBrains 2019 р.

За результатами аналізу використовуваних мов програмування ботів в месенджері Telegram було виявлено, що найбільшу популярність мають боти, написані саме на Python. Про це свідчить кількість репозиторіїв на платформі GitHub.

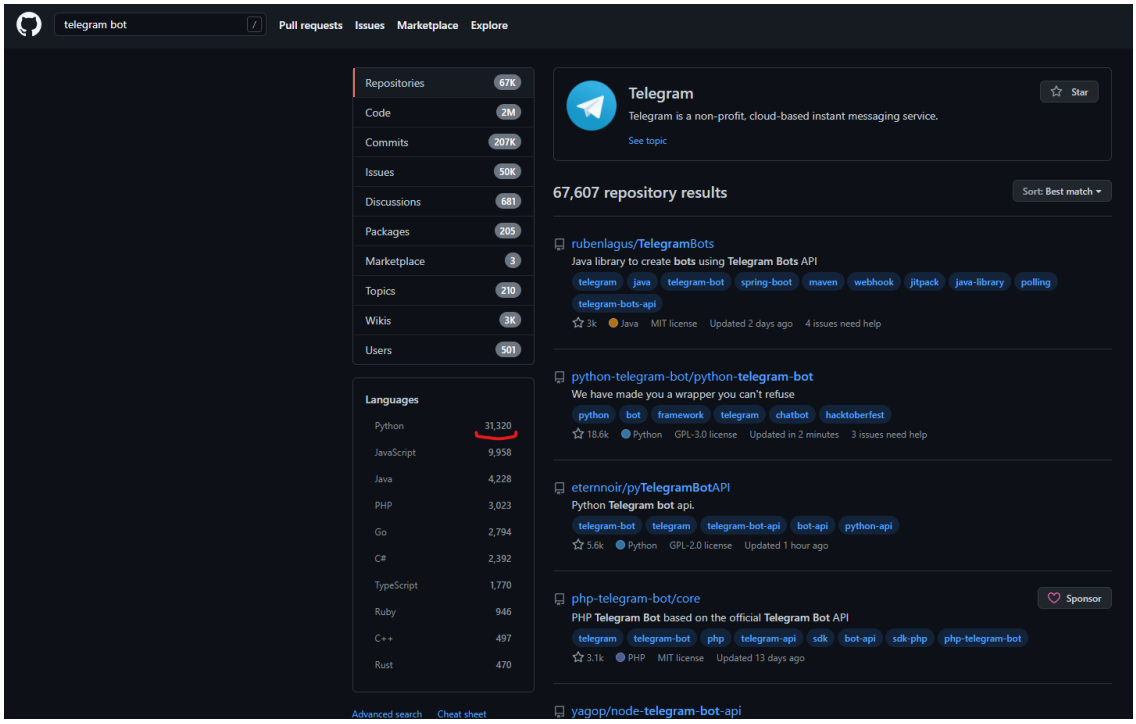



Рисунок 1.5.5 – Пошуковий запит «Telegram bot» на платформі GitHub.

Оскільки, для швидкої відповіді від сервера потрібно обробляти запити до бота від різних користувачів одночасно, потрібно зробити функціонал сервера асинхронним. У зв’язку з тим, що стандартна бібліотека є синхронною, можна використати:

– технологію Python “Threading”, тобто ядра, потоки. Технологія полягає в тому, що розробник вручну налаштовує виконання функцій на різних вітках обробки процесора.

– бібліотеку Aiogram. Aiogram – фреймворк для розробки Телеграм ботів, що основана на іншій технології aiohttp.

Оскільки функціонал Aiogram дуже близький до стандартної офіційної бібліотеки pyTelegramBotAPI, використаємо саме цей фреймворк для розробки асинхронного коду. Асинхронність – це процес обробки введення/виводу, що дозволяє продовжити обробку інших завдань, не чекаючи завершення попереднього завдання.

# РОЗДІЛ 2
# АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

## 2.1 Огляд мови програмування

Python (найчастіше вживане прочитання — «Па́йтон») – мова програмування з відкритим сирцевим кодом написаним мовою програмування С. Творець назвав мову на честь телевізійного шоу «Літаючий цирк Монті Пайтона». Велика кількість жартів з шоу присутня в офіційній документації, прикладах та підручниках.

Пайтон — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Мова була створена голандцем Гвідо ван Россумом 1990 року. Інтерпритована мова програмування це мова, яку не потрібно компілювати перед запуском. Програма під назвою інтерпретатор запускає код Python практично на будь-якому комп’ютері. Це означає, що програміст може змінити код і швидко побачити результати. Це також означає, що Python повільніший за компільовану мову, як-от С, оскільки він не запускає машинний код безпосередньо.

Головні переваги Python – велика спільнота програмістів і висока швидкість розробки. Завдяки лаконічній мові програмування і відмінним бібліотекам проєкти на Python робити дешевше, швидше та простіше. Широкий набір функцій дозволяє вирішувати не тільки типові для інтернет-додатків завдання, а й втілювати в життя унікальні ідеї.

Python – найпопулярніша мова програмування станом на середину 2022 року, згідно з Stack Overflow Trends.

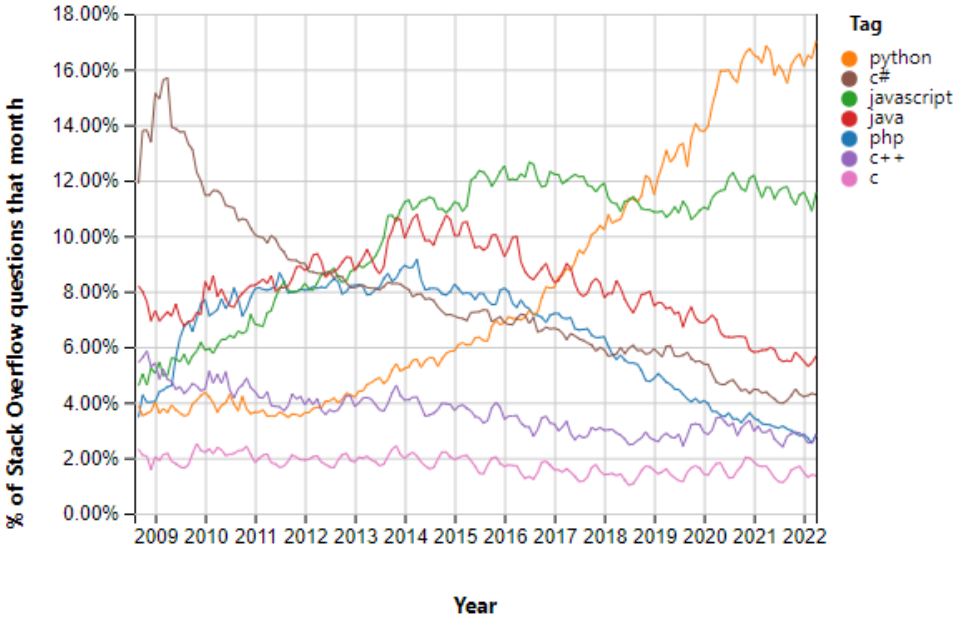



Рисунок 2.1.1 Динаміка зросту кількості обговорень на платформі Stack Overflow

Мова володіє надчистим синтаксисом та лаконічністю. Лаконічний, читабельний код легше читати, інспектувати, виправляти та підтримувати. У порівнянні з іншими сучасними мовами програмування Python виглядає сучаснішим, так як код, написаний мовою Python, зазвичай, значно коротший за код, що виконує ті ж дії, написаний іншими мовами програмування. До прикладу можна привести функцію, що приймає одне чисельне значне та друкує в консоль прямокутний рівнобедрений трикутник з катетами, що дорівнюють цьому значенню. Порівняємо дві однакові функції, але написані різними мовами програмування:

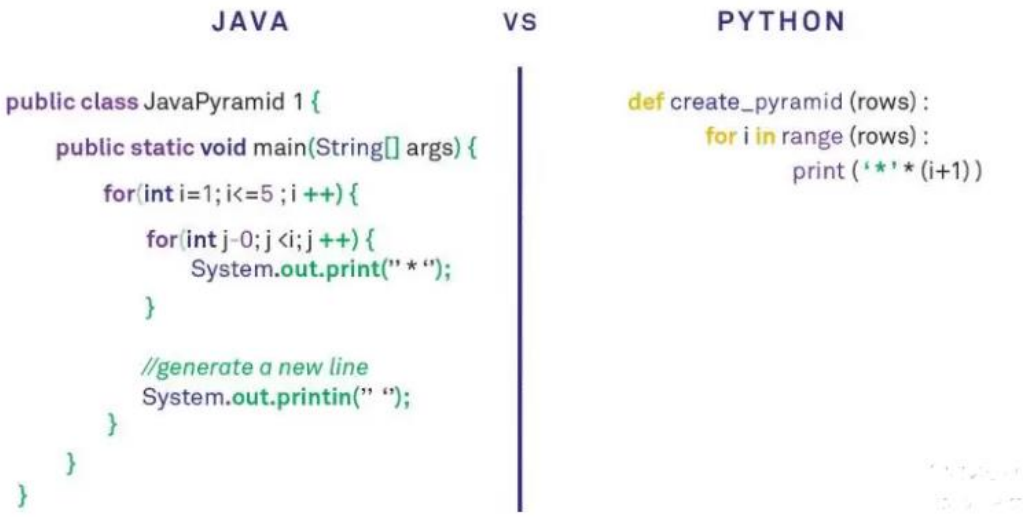



Рисунок 2.1.2 – Порівняння обсягу коду написаних мовами Java та Python

Python — хороша мова програмування для початківців. Це мова високого рівня, а це означає, що програміст може зосередитися на тому, що він повинен зробити, мова не вимагає знань апаратного забезпечення. Написання програм на Python займає менше часу, ніж на інших мовах.

Python був натхненний іншими мовами програмування, такими як C, C++, Java, Perl і Lisp.

Розробники Python намагаються не робити дрібних виправлень, які називаються патчами, до несуттєвих частин довідкової програми CPython, що зробило б Пайтон швидшим, бо коли швидкість виконання важлива, програміст Python може вчасно перекласти частину роботи програми на інші частини, написані не інтерпретованими мовами програмування, наприклад C або компілятором «на ходу» PyPy. Він перекладає скрипт Python на C і здійснює виклики API рівня C безпосередньо до інтерпретатора Python.

Python використовується сотнями тисяч програмістів і в багатьох місцях. Його стандартна бібліотека складається з багатьох функцій, які постачаються разом із Python після його встановлення. У Інтернеті є багато інших доступних бібліотек, офіціних, чи бібліотек, написаних спільнотою, які дають змогу мові Python робити

більше речей. Ці бібліотеки роблять його надпотужною мовою; завдяки яким він може робити багато різних речей.

Деякі речі, для яких Python часто використовується:

- Веб-розробка
- Наукове програмування:
    1. Наука про дані
    2. Машинне навчання
    3. Чисельний аналіз
    4. Статистика
- Програми з графічним інтерфейсом робочого столу
- Мережне програмування
- Програмування ігор
- Створення складних алгоритмів
- Скрипти автоматизації
- Штучний інтелект
- Аудіо та відео програми

З версії 3.5 Python підтримує асинхронний код. Асинхронний додаток продовжує працювати і обробляти запити під час очікування відповіді від сервера.

Так вдається уникнути простою додатку, який виникає при послідовній обробці запитів.

Асинхронна модель підвищує продуктивність і час відклику веб-додатків. Тому виходячи з нашого досвіду Пітон підходить навіть для високонавантажених сайтів і проектів, де важлива швидкість обробки складних запитів.

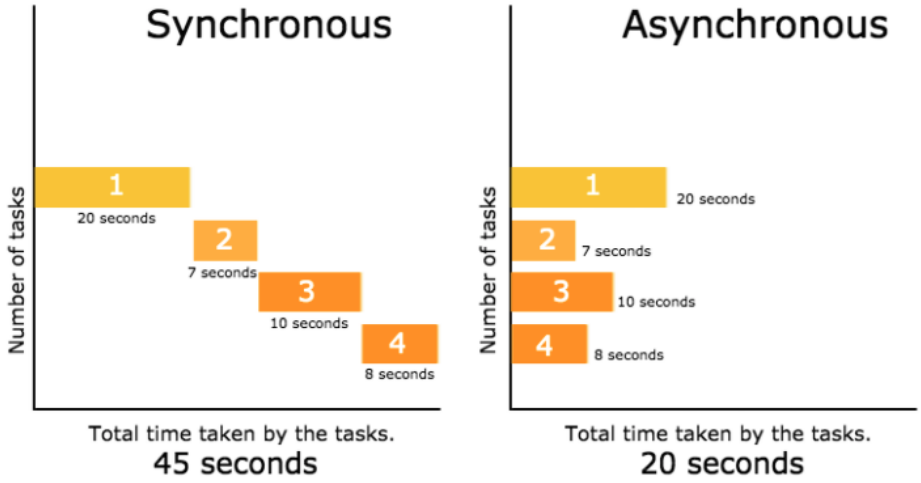



Рисунок 2.1.3 – Ілюстративне порівняння часу виконання Синхронного та асинхронного коду.

## 2.2 Огляд середовища розробки

У сучасних умовах у розробці програмного забезпечення приймає дуже важливу роль також застосунок, в якому ми програмуємо, пишемо код. Середовищем розробки було обране забезпечення, розроблюване компанією Майкрософт та відкритою спільнотою. Саме такий продукт з відкритим вихідним кодом є найпопулярнішим середовищем розробки, згідно з Stack Overflow 2021 Developer Survey, та одним із найзручніших для розробки мовою Python.

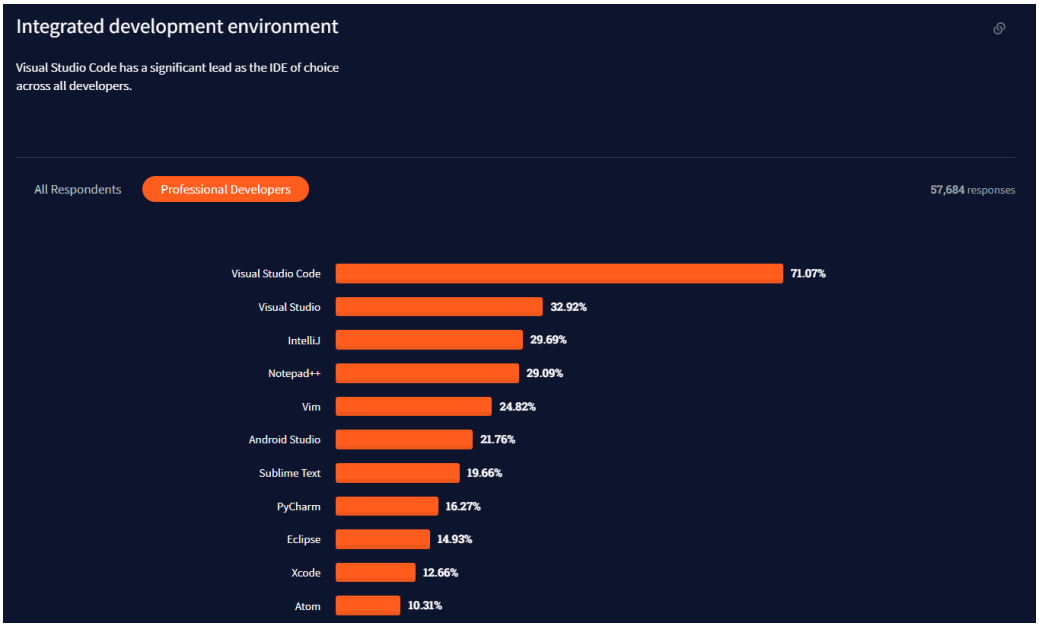



Рисунок 2.2.1 – Опитування професійних розробників про середовище розробки, що ті використовують на платформі Stack Overflow 2021 року

Visual Studio Code, також відомий як VS Code, — це редактор вихідного коду, створений Microsoft для Windows, Linux і macOS. Функції включають підтримку налагодження, виділення синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають додаткові функції.

Visual Studio Code – це розповсюдження публічного репозиторію «Code – OSS» зі спеціальними налаштуваннями від Microsoft, випущеним під традиційною ліцензією на продукт Microsoft.

Visual Studio Code поєднує в собі простоту редактора коду з тим, що потрібно розробникам для їх основного циклу редагування-складання-налагодження. Він забезпечує повну підтримку редагування коду, навігацію та розуміння, а також легке налагодження, багату модель розширюваності та легку інтеграцію з існуючими інструментами.

VS Code — це редактор коду, який можна використовувати з різними мовами програмування, включаючи найпопулярніші Java, JavaScript, Go, Node.js, Python,

C++ і Fortran. Він побудований на фреймворку Electron, який використовується для розробки веб-додатків Node.js, які працюють на механізмі макета Blink. Visual Studio Code використовує той самий компонент редактора (під кодовою назвою «Monaco»), що використовується в Azure DevOps.

Із коробки редактор включає базову підтримку більшості поширених мов програмування. Ця базова підтримка включає підсвічування синтаксису, узгодження дужок, згортання коду та фрагменти, що налаштовуються. Visual Studio Code також постачається з IntelliSense для JavaScript, TypeScript, JSON, CSS і HTML, а також підтримує налагодження для Node.js. Підтримка додаткових мов може бути забезпечена безкоштовними розширеннями на VS Code Marketplace. Щодо Маркетплейсу, що доступний в редакторі, то в ньому доступна численна кількість аматорських та професійних проектів, серед яких крім розширень також доступні також теми, інструменти для тестування, віддаленого підключення тощо. До прикладу, на запит “Python” на Маркетплейсі знаходиться близько 700 додатків, серед яких є як безкоштовні, так і платні розширення.

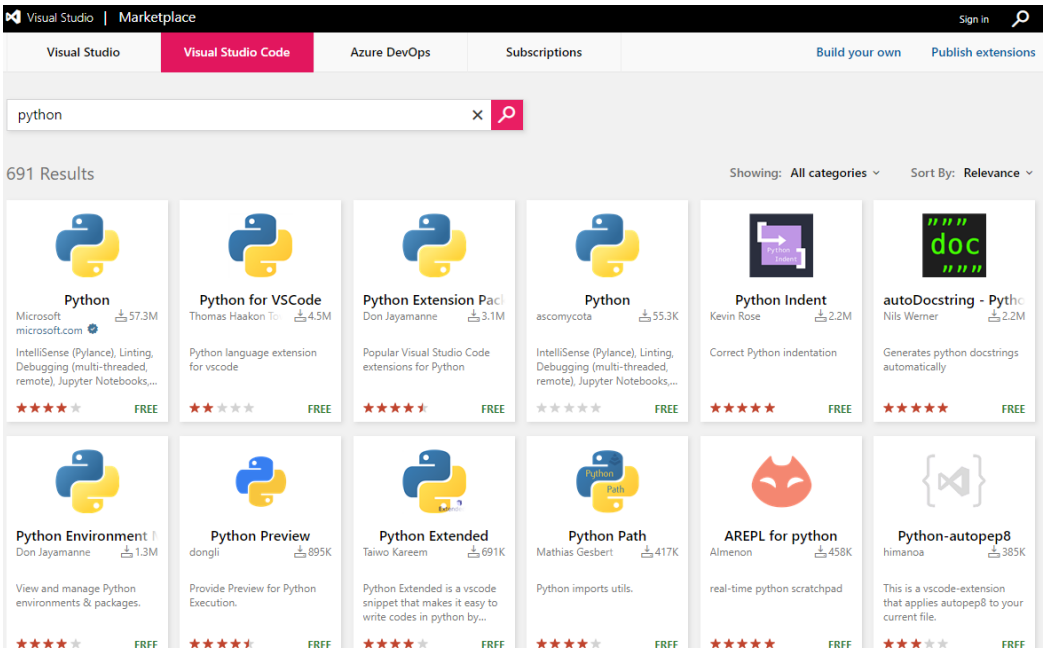



Рисунок 2.2.2 – Пошуковий запит “Python” на VS Code Marketplace

# РОЗДІЛ 3
# ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Проектування діаграми послідовностей

Прецедентні діаграми, діаграми, які забезпечують більш повне уявлення логіки ситуацій використання, можна уточнити за допомогою діаграм послідовності. Це чудовий інструмент для документування випадків використання проекту.

Діаграма послідовності – діаграма, що призначена для представлення взаємодії між елементами моделі програмної системи та повідомлень між ними впорядкованих за часом. Взаємодія – зв’язок між елементами, між якими відбувається обмін інформацією, або міжелементне керування одне одним.

У реальному середовищі такі діаграми використовуються для переходу від вимог до програмного забезпечення на етапі проектування. На відміну від інших діаграм взаємодії, діаграма послідовності візуалізує порядок рузу повідомлень по часу, але не відображає структурні відношення між процесами. Розробка діаграми послідовності здійснюється в ході детального проектування, В подальшому ця діаграма може бути використана для тестування застосунків, або окремих модулів. Для опису програмної системи використовується загальний механізм моделювання взаємодії, які можуть бути представлені з різною мірою деталізації і орієнтовані як на розробника програмної системи, так і на кінцевого користувача.

Об’єкти, які взаємодіють всередині сценарію, повідомлення, якими вони обмінюються, і результати повернення, пов’язані з повідомленнями, часто відображаються на діаграмах послідовності. Графічне зображення реалізації UML діаграми являє собою представлення на горизонтальній вісі площини елементів взаємодії. Кожен з яких має власну лінію життя. З лінії життя виходять горизонтальні стрілки вбік інших ліній життя. Це позначення взаємодій. Вертикальна вісь відповідає за плин часу, чим нижче знаходиться взаємодія, тим пізніше вона повинна відбуватись. Для проектування деталей взаємодій і опису

послідовностей подій в мові UML 2.0 використовують спеціальний термін «траєкторія». При об’єднанні двох і більше траєкторій, при умові, що події з різних траєкторій проходять у будь-якому порядку, а події з одніє і тієї ж траєкторії у визначеному порядку, називається траєкторія, що чергується. Повідомлення зображуються у вигляді горизонтальних стрілок, які можуть мати ім’я повідомлення. Лінія життя об’єкта зображується пунктирною лінією, що йде вертикально вниз від елементів взаємодії. Вона представляє існування об’єкта прогягом певного періоду часу. Для позначення моменту знищення об’єкта застосовуються символи хреста, оберненого як буква «х». Іноді можна зустріти діаграму послідовностей, що містять елемент Актор. Це використовується для відображення взаємодії з користувачем. Також для відображення періоду часу, який є «активним», тобто період, під час якого дія виконувалась, використовується прямокутник, витягнутий по лінії життя на потрібний період.

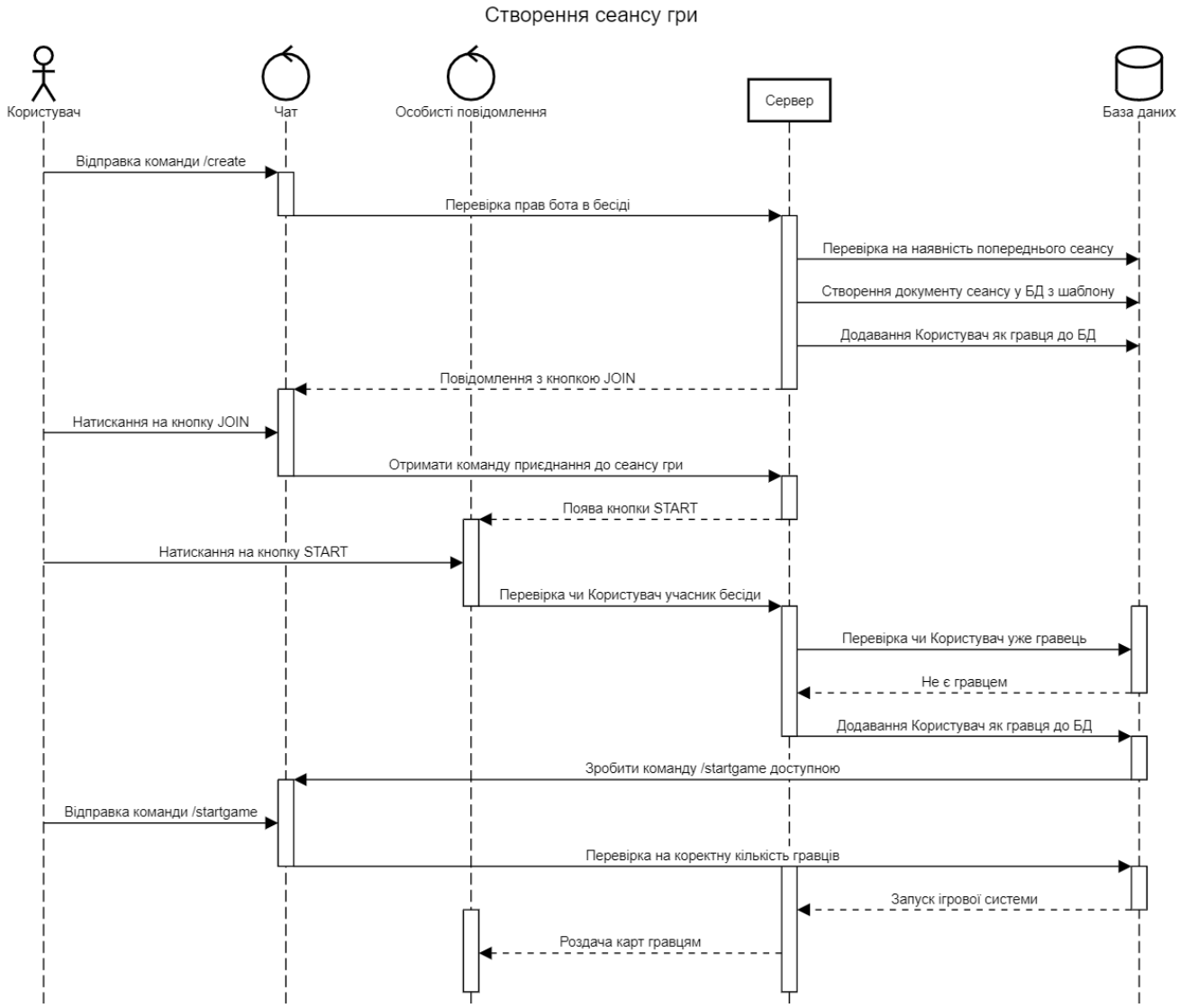



Рисунок 3.1.1 – Діаграма послідовності створення сеансу гри.

## 3.2 Проектування внутрішньої будови класів

У інженерії програмного забезпечення діаграма класів це вид статичної діаграми структур, що описує модельну структуру системи, показуючи класи розроблені в системі, їх атрибути, методи та взаємозв'язки між іншими об'єктами.

Діаграма класів є основним архітектурним елементом моделювання об’єктно-орієнтованого програмного продукту. Використовують діаграму для загального моделювання структури програми та для детального моделювання перенесення моделей у програмний код.

На схемах класи представлені прямокутними фігурами, які містять в собі три відділення:

- У верхньому відділенні міститься назва класу. Назва зазвичай написана жирним шрифтом з великої літери, текст відцентрований.
- Середній відсік – це атрибути класу. Вирівнюються за лівим краєм, весь текст пишеться малим регістром.
- Третє відділення містить методи, які наявні у класі. Вирівнюються за лівим краєм, весь текст пишеться малим регістром.

При проектуванні системи ряд класів ідентифікується та згруповується у схему класів, яка допомагає визначити статичні відносини між ними. При детальному моделюванні класи концептуального проекту часто поділяються на ряд підкласів.

Залежність – це зв’язок між залежними та незалежними елементами моделей. Залежність відображається у вигляді пунктирної лінії зі стрілкою, яка вказує напрямок від клієнта до постачальника.

Асоціація являє собою родину посилань. Двійкова асоціація (з двома кінцями) зазвичай представляється у вигляді рядка. Асоціація може зв’язувати будь-яку кількість класів. Потрійною асоціацією називається асоціація з трьома ланками. Асоціаціям можна присвоїти ім’я, а кінці асоціацій можна прикрашати іменами ролей, кратністю, показниками власності, видимістю чи іншими властивостями.

Розрізняють чотири різні типи асоціацій: односпрямована, двонаправлена, агрегаційна (включає в себе агрегацію композиції) та рефлексивна. Найбільш поширеними є двонаправлені та односпрямовані асоціації.

До прикладу, клас польоту асоціюється з класом літака двонаправлено. Асоціація представляє статичне відношення, яке ділиться між об'єктами двох класів.

Агрегація є більш точним варіантом зв'язку "має", ніж асоціація. Це зв’язок, який символізує частину мети або відносин. Агрегація може мати назву й мати ті самі характеристики, що й асоціація, як різновид асоціації. Агрегація, з іншого

боку, не може мати більше двох класів; воно повинно мати бінарне відношення. Крім того, під час впровадження існує невелика різниця між агрегаціями та асоціаціями, і діаграма може повністю пропустити зв’язки агрегації.

Агрегація може відбуватися, коли клас є колекцією або контейнером для інших класів, але життєві цикли включених класів не сильно залежать від контейнера. Коли контейнер знищено, вміст контейнера залишається.

Сукупність — це семантично розширений об’єкт, який розглядається як одиниця в багатьох діях, незважаючи на те, що він фізично складається з кількох менших об’єктів. Це тому, що сукупності надано додаткове семантичне значення.

Приклад: Бібліотека та студенти. Студент може існувати без бібліотеки, зв'язок між студентом і бібліотекою буде називатись агрегацією.

Це свідчить про те, що один із двох споріднених класів, відомих як підклас, вважається більш конкретним екземпляром іншого, відомого як супертип, тоді як інший, відомий як суперклас, вважається більш узагальненим підкласом. У застосуванні це означає, що кожен екземпляр підтипу також є екземпляром суперкласу, до якого він належить.

Форма порожнистого трикутника в кінці суперкласу рядка (або дерева рядків), який зв’язує його з одним або кількома підтипами, є графічним зображенням узагальнення UML. Цю форму можна знайти в кінці дерева рядка.

Зв’язки успадкування та зв’язки «є» — це ще дві назви відносин узагальнення.

У контексті відносин узагальнення термін «батько» відноситься до суперкласу, який також відомий як базовий клас або базовий тип.

Підтип, який є дочірнім іншим підтипом у відношенні спеціалізації, також відомий як "дочірній" підклас, похідний клас або похідний тип, клас успадкування або тип успадкування.

Наприклад, "дуб - це тип дерева" і "автомобіль - це тип транспортного засобу" обидва приклади класифікацій.

Для представлення узагальнень підходять лише діаграми класів і діаграми використання.

Зв'язок реалізації — це тип з'єднання, яке може використовуватися в моделюванні UML. Цей тип відносин визначається як такий, у якому один елемент моделі (клієнт) реалізує поведінку, описану іншим елементом моделі (постачальником).

Графічне зображення реалізації UML являє собою трикутник з отвором посередині, який знаходиться в кінці дерева рядків або інтерфейсу пунктирної лінії, що пов’язує його з одним або кількома реалізаторами. На терміналі інтерфейсу пунктирної лінії, який зв’язує його з користувачами, пряма стрілка позначає з’єднання. У цих компонентних діаграмах використовується умова діаграми, відома як "м'яч і розетка" (реалізатори ставлять кульку або льодяник, а користувачі показують розетку). Реалізації можна відобразити лише на діаграмах класів або компонентів. Зв'язок між класами, інтерфейсами, компонентами та пакетами, який встановлює зв'язок між елементом замовника та елементом постачальника, називається реалізацією. З відношення реалізації, яке існує між інтерфейсами та класами чи компонентами, зрозуміло, що клас чи компонент дійсно втілюють на практиці операції, які надаються інтерфейсом.

Залежність — це слабкіший тип спілкування, який показує, що один клас залежить від іншого класу, оскільки перший клас використовує другий клас у певний момент часу. Якщо незалежний клас використовується як змінна параметра або локальна змінна методу в залежному класі, то ці два класи вважаються залежними один від одного. Це не те саме, що й асоціація, в котрій екземпляр залежного класу вважається атрибутом незалежного класу. Бувають випадки, коли зв’язок між двома класами не дуже міцний. У них взагалі немає жодних змінних-членів, реалізованих якимось чином. Натомість їх можна використовувати у формі аргументів під час виклику функції-члена.

Фраза «А володіє В» зазвичай використовується для опису цього зв’язку.

Представлення асоціацій в UML - це лінія, яка з'єднує два класи, які з'єднані один з одним. На обох кінцях лінії додаються додаткові символи. Наприклад, ми можемо використовувати наконечник стрілки, щоб показати, що загострений кінець можна побачити з хвоста стрілки, вказавши, що він видимий. Ми можемо

позначити право власності шляхом розміщення м'яча, роль, яку виконують складові цієї мети, назвавши роль, а кількість зустрічей цієї сутності, вказавши їхню множинність (діапазон об'єктів, залучених до асоціації з іншого кінця ).

Моделювання довговічної інформації, яку система обробляє, і, в деяких випадках, поведінки, яка пов'язана з цією інформацією, є обов'язком класів сутностей. Недоцільно називати їх таблицями бази даних або будь-яким іншим типом сховища даних.

Вони показані у вигляді кіл, які мають коротку лінію, додану до нижньої частини кола. З іншого боку, їх можна зобразити як звичайні класи стереотипу «сутність», накладеним на назву класу.

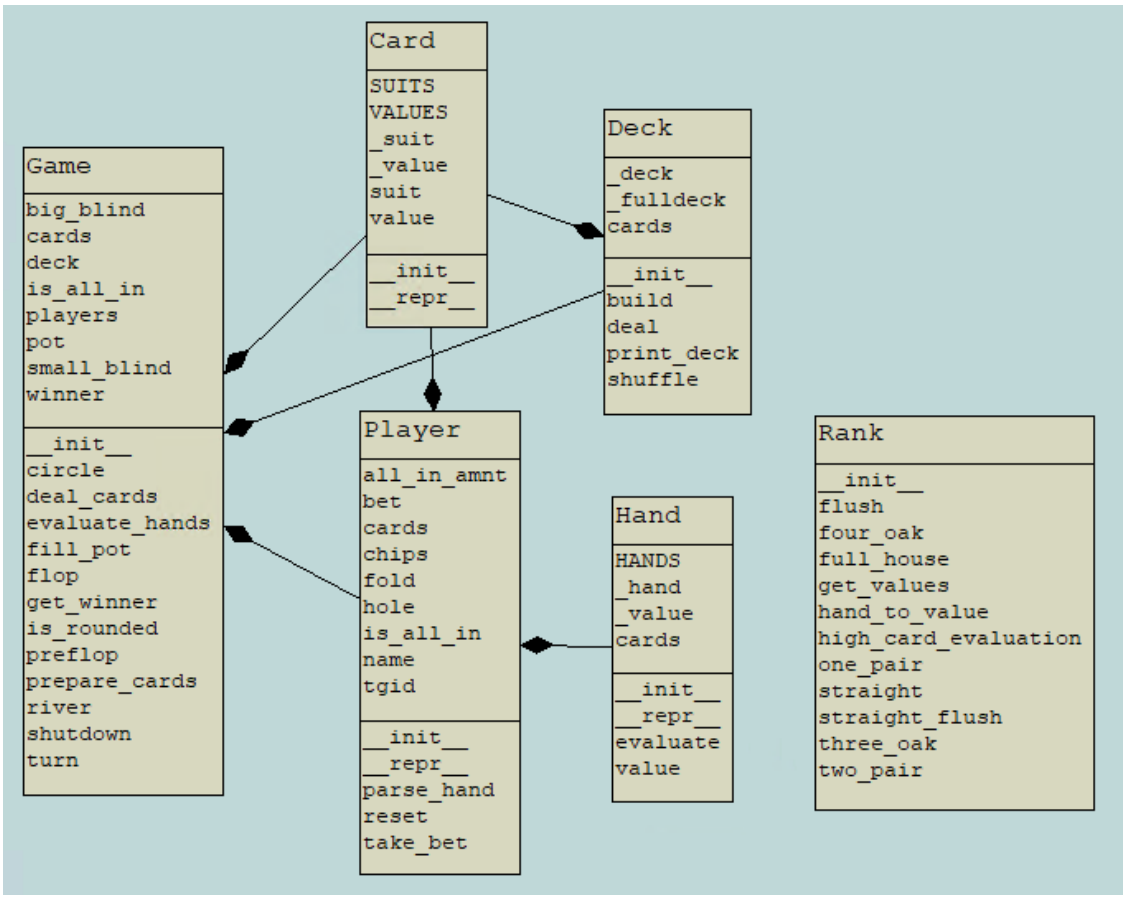



Рис. 3.2.1 — Діаграма класів ядра покер в Телеграм додатку

### 3.3 Розробка системи розрахунку переможця

Цікавості гра набуває лише тоді, коли в ній може бути переможець. На просторах інтернету не вдалося знайти систему розрахунку комбінації карт, що буде зручного зі сторони архітектури додатку. На мою думку, комбінація карт повинна зберігатись у одній змінній, щоб до неї можна було застосувати логічні більше та менше для знаходження вищої комбінації карт. Численними тестуваннями було знайдене рішення для зберігання комбінації карт у списку з трьох елементів типу Integer, тобто цілих чисел. Перше число позначатиме комбінацію карт від 0 до 9, де 0 – це комбінація «Вища карта», тобто найнижча комбінація, та 9 – комбінація «Роял Флеш». Друге число зберігатиме номінал основної карти у комбінації, тобто у «Вища карта» номінал вищої карти, у «Стріт» номінал карти, якою починається порядок, у «Фул Хаус» номінал карти, яких у комбінації є три, згідно з правилами покеру.

| 1st cell | Hand Name | 2nd cell | Hand Main Card | 3rd cell | High Card Evaluation |
|---|---|---|---|---|---|
| high card | 0 | 2 | 2 | 2 | 0 |
| one pair | 1 | 3 | 3 | 3 | 1 |
| two pair | 2 | 4 | 4 | 4 | 2 |
| three of a kind (set) | 3 | 5 | 5 | 5 | 4 |
| straight | 4 | 6 | 6 | 6 | 8 |
| flush | 5 | 7 | 7 | 7 | 16 |
| full house | 6 | 8 | 8 | 8 | 32 |
| four of a kind (quads) | 7 | 9 | 9 | 9 | 64 |
| straight flush | 8 | 10 | 10 | 10 | 128 |
| royal flush | 9* | J | 11 | J | 256 |
| | | Q | 12 | Q | 512 |
| | | K | 13 | K | 1024 |
| | | A | 14 | A | 2048 |

Рисунок 3.3.1 – Таблиця системи розрахунку комбінації «Три клітинки»

У покері часто трапляється ситуація, коли у гравців однакова комбінація карт. У такому випадку оголошувати нічию ще рано, адже є ще «Кікер». Так у покері називають другорядну карту, що може стати виграшною у комбінації. Наприклад, у гравців однакова комбінація «Пара» двох десяток. Тоді для визначення переможця ми поглянемо на карту «Кікер», тобто найвища серед решти трьох карт у комбінації. Якщо ж і ці карти у гравців однакового номіналу, то дивимось на наступну карту, і так по-порядку. Нічию оголошують лише якщо всі п’ять карт однакові. Тому третє число в нашій змінній, що зберігатиме прораховану комбінацію карт, варте особливої уваги, адже вимогами до числа було те, щоб, порівнюючи однакові комбінації з однаковими номіналами основних карт, це число зберігало номінали всіх решти карт і при цьому при порівнянні відповідало вимогам правил щодо порівняння спочатку вищих за номіналом карт по-порядку.

Тому для задоволення архітектурних вимог по збереженню цих величин в одному числі, це число має бути сумою всіх карт у комбінації, і, при цьому, для задоволення вимог щодо правил гри, карта найвищого номіналу повинна бути більшою за суму всіх решти можливих номіналів карт. Було прийняте рішення по перетворенню номіналів карт від трійки на числа степеню 2. Тобто, карта «Туз» відповідає числу 2048 ($2^{11}$), карта «2» = 0, карта «3» = $2^0 = 1$, «6» = $2^3 = 8$. Додаємо перетворені суми всіх карт, та отримуємо бажане нам число, яке відповідає нашим вимогам.

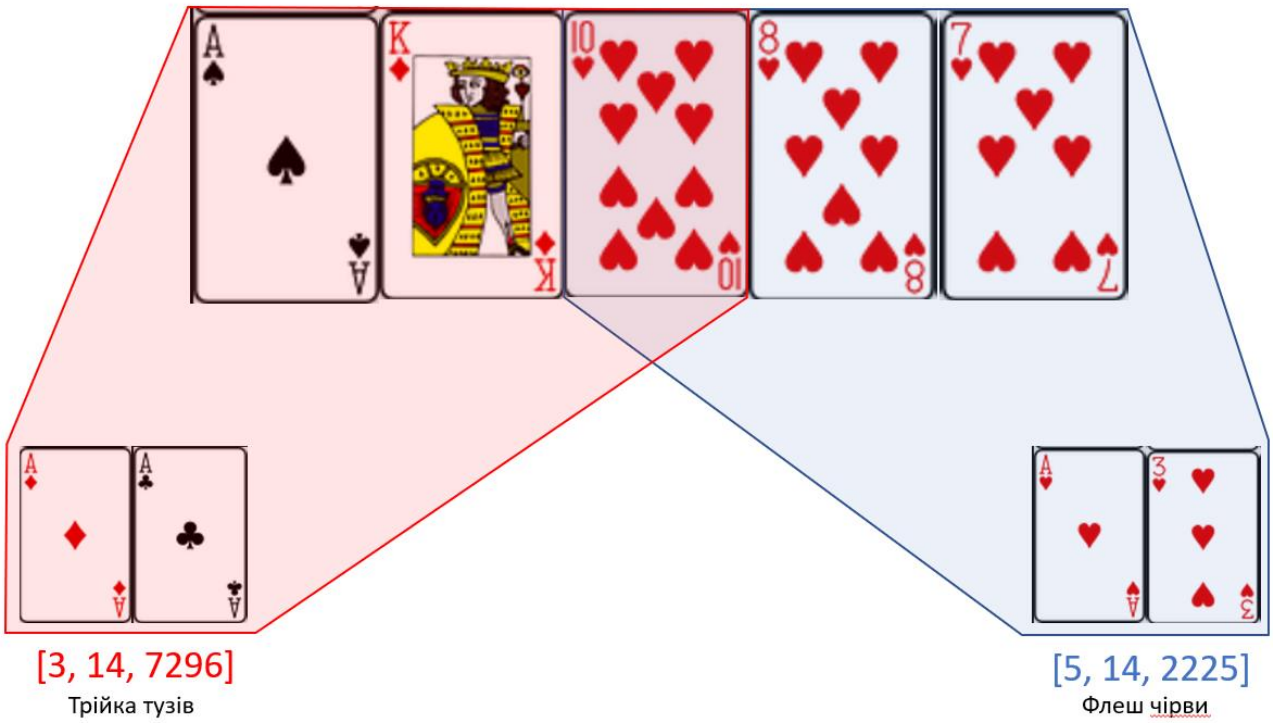



Рисунок 3.3.2 – Приклад розрахунку комбінації

За правилами виграшна комбінаціє має бути та, що синя справа, незважаючи на те, що карти у гравця зліва вищі (див. Рисунок 3.3.2). Порівнювати розрахунки треба починаючи з першого числа у змінній, що зберігає розраховану комбінацію, до третього. Для цього можна створити простеньку функцію, або скористатись вбудованим у мову Python методом max(*[list]*), що приймає у себе послідовність чисел і повертає максимальне з неї. У мові Python логічне порівняння послудовностей чисел порівнює елементи по черзі, та, як тільки знаходить нерівні числа, повертає найвище. Тому до методу max(*[list]*) передамо список змінних, що визначають розраховані комбінації карт, що являють собою списки з трьох елементів кожна. Функція поверне найвищу комбінацію коректно.

### 3.4 Тестування програмного продукту

Процес знаходження артефактів і поведінки програми, що перевіряється, відомий як тестування програмного забезпечення. Тестування програмного забезпечення також може дати організаціям об’єктивну та неупереджену перспективу програми, що дає їм змогу оцінити та усвідомити ризики, пов’язані з використанням програмного забезпечення. Методи тестування можуть включати, але цим не обмежуватись, наступне:

- перевірка вимог до продуктів для визначення їх повноти та точності в різних умовах, включаючи галузеву точку зору, бізнес-перспективу, доцільність і життєздатність впровадження, простоту використання, продуктивність, безпеку, проблеми з інфраструктурою тощо;
- аналіз архітектури та загального дизайну продукту;
- співпраця з розробниками продукту, для покращення техніки кодування, шаблонів дизайну та тестів, які можна розробити як частину коду на основі різноманітних методологій, таких як граничні умови тощо;
- виконання програми або процедури з метою перевірки її поведінки;
- переконання, що інфраструктура розгортання разом з будь-якими відповідними сценаріями та автоматизацією знаходиться в робочому стані;
- участь у виробничих операціях, використовуючи різні методи моніторингу та спостереження.

Результати тестування програмного забезпечення можуть запропонувати користувачам або спонсорам об’єктивну та неупереджену інформацію про якість програмного забезпечення та ймовірність збою програмного забезпечення.

**Несправності та збої**

Помилки програмного забезпечення виникають через наступну процедуру: Помилка робиться програмістом, що призводить до внесення помилки (дефекту, артефакту) у вихідний код програмного забезпечення. У випадку, якщо буде допущена ця помилка, система за певних обставин дасть неточні результати, що в кінцевому підсумку закінчиться збоєм.

Немає гарантії, що кожна помилка призведе до збою. Наприклад, помилки мертвого коду будуть працювати ідеально щоразу. Проблема, яка не розпізнає несправність, може призвести до того, що середовище не буде змінено, коли це повинно було бути. Приклади такого роду змін у середовищі включають роботу програмного забезпечення на іншій апаратній платформі комп'ютера, зміни вихідних даних або взаємодію з іншими видами програмного забезпечення. Одна проблема може призвести до широкого спектру ознак і симптомів відмови.

Помилки в сирцевому коді не завжди є винуватцями помилок програмного забезпечення. Прогалини в вимогах, також відомі як невизнані потреби, є типовою причиною дорогих дефектів. Ці прогалини призводять до того, що програміст пропускає помилки, що призводить до дорогих дефектів. Відсутність нефункціональних потреб, таких як тестованість, масштабованість, ремонтопридатність, продуктивність та інші вимоги, є поширеною причиною прогалин у вимогах безпеки.

**Тестування за допомогою статичних, динамічних та пасивних методів**

Тестування програмного забезпечення може проводитися різними способами. Тестування, яке складається з оглядів, покрокових інструкцій або перевірок, називається статичним тестуванням, тоді як тестування, що включає виконання запрограмованого коду із заздалегідь визначеним набором тестових випадків, називається динамічним тестуванням.

Коли інструменти програмування або текстові редактори перевіряють структуру вихідного коду, або коли компілятори (попередні компілятори) перевіряють граматику та потік даних у рамках статичного аналізу програми, це є прикладом неявного типу статичного тестування. Динамічне тестування відбувається, коли програмне забезпечення запускається самостійно. Під час динамічного тестування окремі фрагменти коду можуть бути застосовані до окремих модулів або функцій ще до того, як програма буде завершена в цілому. Використання заглушок, а також налагодження також є поширеними підходами до написання цих фрагментів коду тестування.

Перевірка є частиною як статичного, так і динамічного тестування, і обидва види тестування передбачають перевірку.

Перевірка роботи системи таким чином, що не передбачає взаємодії користувача з програмою, що тестується, називається пасивним тестуванням. На відміну від активного тестування, пасивне тестування вимагає від тестувальників аналізувати системні журнали та трасування, а не надавати будь-які тестові дані. Щоб прийти до висновку, вони шукають зразки та особливу поведінку. Перевірка часу виконання в автономному режимі та вивчення журналу також є частиною цього процесу.

**Методологія дослідження**

Дослідницьке тестування — це підхід до тестування програмного забезпечення, що передбачає навчання, розробку тестів і виконання їх одночасно. Джем Канер, який вперше використав цей термін у 1984 році, визначає дослідницьке тестування як «стиль тестування програмного забезпечення, який підкреслює особисту свободу та відповідальність окремого тестувальника за постійну оптимізацію якості своєї роботи, враховуючи навчання, пов’язане з тестуванням, дизайн тесту та продуктивність». Дослідницьке тестування та інтерпретація результатів тесту розглядаються як додаткові дії, які виконуються паралельно протягом усього часу проекту.

**Підхід «Коробка»**

Тестування білої скриньки та тестування чорної скриньки — це дві основні категорії, які історично використовувалися для класифікації підходів до тестування програмного забезпечення. При створенні тестових випадків тестувальник може вибрати будь-який з цих двох методів, які використовуються для характеристики точки зору тестера. Методології тестування програмного забезпечення також можуть використовувати гібридну техніку, відому як тестування сірого ящика. Це «розходження» між тестуванням чорних і білих ящиків значною мірою зникло, оскільки ідея тестування сірої коробки, яка генерує тести на основі конкретних компонентів дизайну, стає все більш поширеною. Тестування сірого ящика розробляє тести на основі конкретних елементів дизайну.

Таблиця 3.4.1 — Тестування телеграм-додатку

| № | Тест-кейс | Очікуваний результат | Отриманий результат |
|---|---|---|---|
| 1 | Спроба створити гру командою в особистих повідомленнях користувача з ботом | Помилка. Повідомлення про необхідність виконання команди в повідомленні в груповому чаті | Помилка. Повідомлення про необхідність виконання команди в повідомленні в груповому чаті |
| 2 | Спроба доєднатись до гри гравцеві, якого немає в бесіді, в якій сеанс гри був створений | Помилка. Повідомлення про необхідність перебувати в бесіді, в якій запрошення було сформовано | Помилка. Повідомлення про необхідність перебувати в бесіді, в якій запрошення було сформовано |
| 3 | Спроба приєднання до гри одинадцятого гравця | Помилка. Повідомлення про те, що максимальна кількість гравців: 10 | Помилка. Повідомлення про те, що максимальна кількість гравців: 10 |
| 4 | Спроба створення гри з менше, ніж двома гравцями | Помилка. Повідомлення про неможливість грати самотужки | Помилка. Повідомлення про неможливість грати самотужки |
| 5 | Хід гравця поза чергою / повідомлення іншого гравця (не того, чий хід) | Ігнорування повідомлення системою | Ігнорування повідомлення системою |

| | | | |
|---|---|---|---|
| 6 | Відправка гравцем повідомлення, що не є командою | Ігнорування повідомлення системою | Ігнорування повідомлення системою |
| 7 | Відправка команди в особисті повідомлення бота з користувачем | Помилка. Повідомлення про те, що команда може бути відправлена лише в груповому чаті, адже активних сеансів у користувача може бути кілька | Помилка. Повідомлення про те, що команда може бути відправлена лише в груповому чаті, адже активних сеансів у користувача може бути кілька |
| 8 | Відправка гравцем команди, що недоступна йому на даному ході | Помилка. Повторне очікування коректної команди. Перелік доступних команд в повідомленні | Помилка. Повторне очікування коректної команди. Перелік доступних команд в повідомленні |
| 9 | Команда підвищення ставки без вказування суми ставки | Помилка. Повторне очікування коректної команди. Перелік доступних команд в повідомленні | Помилка. Повторне очікування коректної команди. Перелік доступних команд в повідомленні |
| 10 | Команда підвищення ставки, зі вказаною ставкою більшою за доступну гравцеві суму | Помилка. Повідомлення про кількість доступної гравцеві валюти | Помилка. Повідомлення про кількість доступної гравцеві валюти |
| 11 | Команда підвищення ставки зі вказаною сумою ставки, що є | Помилка. Повідомлення про найменшу доступну | Помилка. Повідомлення про найменшу доступну |

| | | | |
|---|---|---|---|
| | меншою за найменшу доступну за правилами покеру (найбільша ставка на столі + смол блайнд) | суму ставки на даному етапі торгів | суму ставки на даному етапі торгів |

Результати тестування показують, що система повністю функціональна і готова до використання в реальних умовах.

# ВИСНОВКИ

У ході виконання атестаційної роботи були розглянуті потреби, вимоги, технічні можливості потенційної аудиторії продукту. Застосунок має бути мінімальний за розміром, або повністю знаходитись на сервері, має підтримувати велику кількість сучасних операційних систем та мати можливість працювати на застарілих та бюджетних пристроях, з обмеженим доступом до інтернету гравця. Для задоволення цих потреб було обрано як платформу для розробки месенджер Телеграм. Він встановлений у переважної більшості молоді України та підтримує велику кількість пристроїв. Серед вимог була також відсутність азартної складової гри в покер, що була досягнута шляхом повної відсутності монетизації за рахунок користувача та вільної можливості створення нового сеансу гри.

У результаті виконання даної дипломної роботи було розроблено систему для гри в покер мовою Python. Для взаємодії з системою використовується месенджер Телеграм та його можливість імплементування чат-ботів. Для зберігання даних використовується MongoDB.

У процесі виконання кваліфікаційної роботи були виконані чітко поставлені раніше задачі.

1. Були проаналізовані вимоги, потреби, технічні можливості потенційної аудиторії даного програмного продукту. У результаті було обрано платформу Телеграм для розробки додатку з проведення сеансів гри в покер.
2. Проведений аналіз інструментів та програмних засобів реалізації, що покажуть найкращу продуктивність системи, під час застосування її в реальних умовах, та що спричинять найбільшу зручність під час написання програмного продукту. У ході аналізу було обрано використання VisualStudio Code як середовища розробки, мову програмування Python, асинхронну бібліотеку Aiogram для взаємодії з API Telegram та систему керування базами даних MongoDB, розташовану в хмарному сховищі.
3. Спроектовано та розроблено телеграм-додаток для створення та проведення сеансів гри в покер серед учасників чату в месенджері Телеграм.

4. Проведено тестування телеграм-додатку та ігрової системи на виявлення помилок в роботі. Було досліджено більшу частину сегментів взаємодії користувача з додатком, Помилок не було виявлено, система працює стабільно та надійно.

Створений додаток має значні перспективи подальших досліджень.
Можна додати функціонал зі зміною активних учасників гри під час сеансу, функціонал пересилання ігрової валюти іншому гравцеві, режим гри “лише фішки”, що передбачає тільки контроль ігрової валюти для гравців, в яких є фізичні гральні карти але немає фішок тощо.

Робота пройшла апробацію на Науково-технічній конференції «Застосування програмного забезпечення в ІКТ», м. Київ, ДУТ, 20 квітня 2022 року.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Official Rules of Poker – CardPlayer [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.cardplayer.com/rules-of-poker
2. How to Play Texas Hold'em Poker: Rules & Hands [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.pokernews.com/poker-rules/texas-holdem.htm
3. Як грати в покер - правила гри в покер для новачків в 2021 [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://cardmates.ua/pravila_pokeru/
4. 5 takeaways from the Python Developer Survey 2019 [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://blog.jetbrains.com/pycharm/2020/05/5-takeaways-from-the-python-developer-survey-2019/
5. Bot Code Examples [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://core.python.org/bots/samples/
6. Telegram (software) [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://en.wikipedia.org/wiki/Telegram_(software)
7. Telegram Bot Platform [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://telegram.org/blog/bot-revolution?setln=en
8. Top 10 moments in poker history [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.casinocitytimes.com/tadas-peckaitis/article/top-10-moments-in-poker-history-67088
9. "History of poker" Роя В. Card Night / Will Roya., 2021. – 203 с.
10. G. van Rossum – Python (programming language) [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.wikizero.com/en/Python_(language)

11. Visual Studio Code - Open Source ("Code - OSS") [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://github.com/microsoft/vscode/blob/main/README.md
12. Stack Overflow Developer Survey 2021 [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://insights.stackoverflow.com/survey/2021#most-popular-technologies-new-collab-tools-prof
13. Introduction to Python [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.w3schools.com/python/python_intro.asp
14. “OMG Unified Modeling LanguageTM (OMG UML), Superstructure” UML Superstructure Specification, v2.3 / Object Management Group, 2010.
15. Діаграми UML для для моделювання процесів і архітектури проекту [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://evergreens.com.ua/ua/articles/uml-diagrams.html

# Додаток А

## ПРЕЗЕНТАЦІЯ ДО ЗВІТУ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

# РОЗРОБКА ТЕЛЕГРАМ-ДОДАТКУ СПІЛЬНОЇ ГРИ ПОКЕР МОВОЮ PYTHON

Виконав студент 4 курсу

Групи ПД-41
Волощук Роман Миколайович
Керівник роботи

к. т. н., доц. Негоденко Олена Василівна

Київ – 2022

1

# АНАЛІЗ АНАЛОГІВ

| Назва | Посилання | Переваги | Недоліки |
|---|---|---|---|
| **PokerStars** | pokerstars.com | Найбільша аудиторія, можливість створення гри з власними налаштуваннями, багато платформ, що підтримуються | Необхідність наявності стабільного з'єднання, реєстрації, завантаження додатку |
| **Zynga Poker** | zyngapoker.com | Багато платформ, що підтримуються, гра без реєстрації | Неможливість створення власної гри, неможливість відключення таймера ходу, |
| **EasyPoker** | easy.poker | Створення особистої гри, гра без реєстрації, зручний інтерфейс для мобільних пристроїв | Велика доля функціоналу платна |

2

# Мета, об’єкт та предмет роботи

- **Мета роботи** – спрощення процедури проведення сеансів гри в покер у групі людей, зібраних онлайн чи офлайн, за рахунок телеграм-додатку у месенджері Телеграм.
- **Об’єкт дослідження** – комунікація користувачів в процесі гри в покер.
- **Предмет дослідження** – створення телеграм-додатків для забезпечення комунікації користувачів в процесі гри в покер.

3

# ТЕХНІЧНІ ЗАВДАННЯ

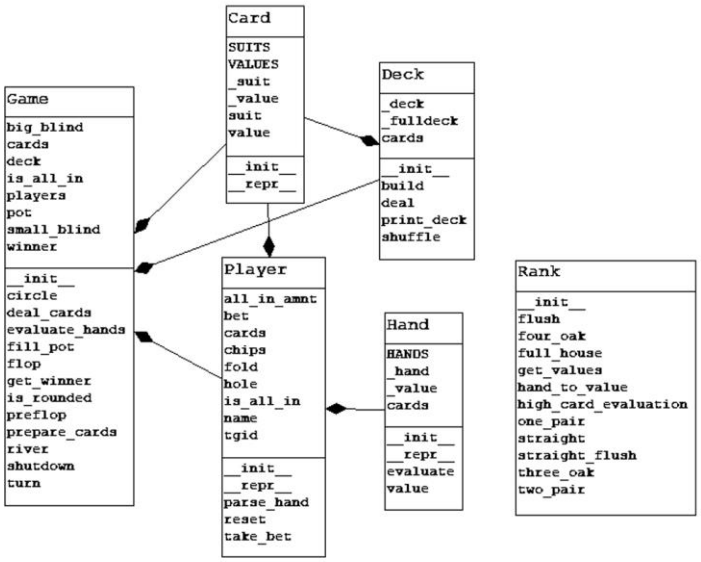
1. Забезпечити взаємодію з ігровою системою через месенджер Телеграм
2. Можливість створення нового сеансу гри користувачем
3. Можливість приєднання чат-бота до існуючих спільнот
4. Можливість вибору стартової кількості фішок гравців
5. Сповіщення про переможця та переможну комбінацію карт гравця
6. Збереження результатів гри в базі даних на сервері
7. Відсутність монетизації за рахунок користувачів

4

# ПРОГРАМНІ ЗАСОБИ ТА ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ

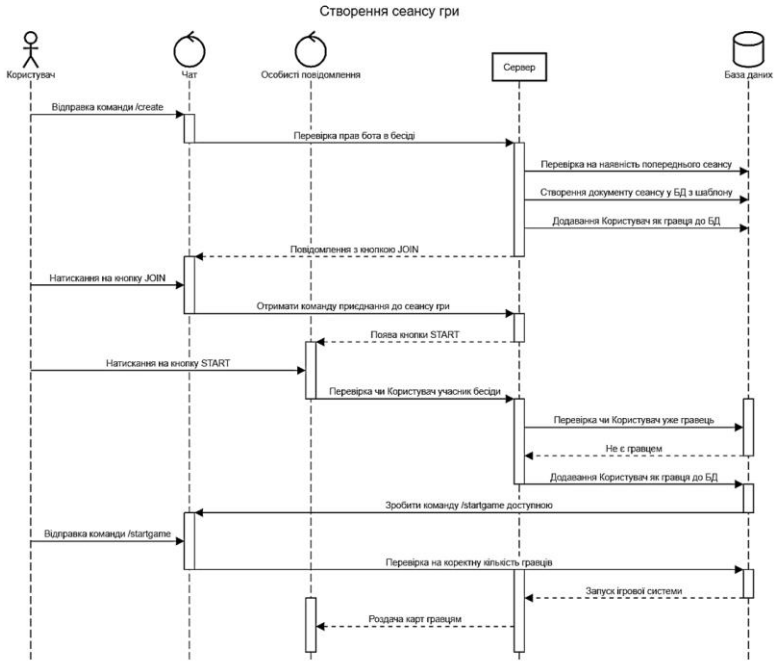
5

# ДІАГРАМА КЛАСІВ ПРОГРАМИ



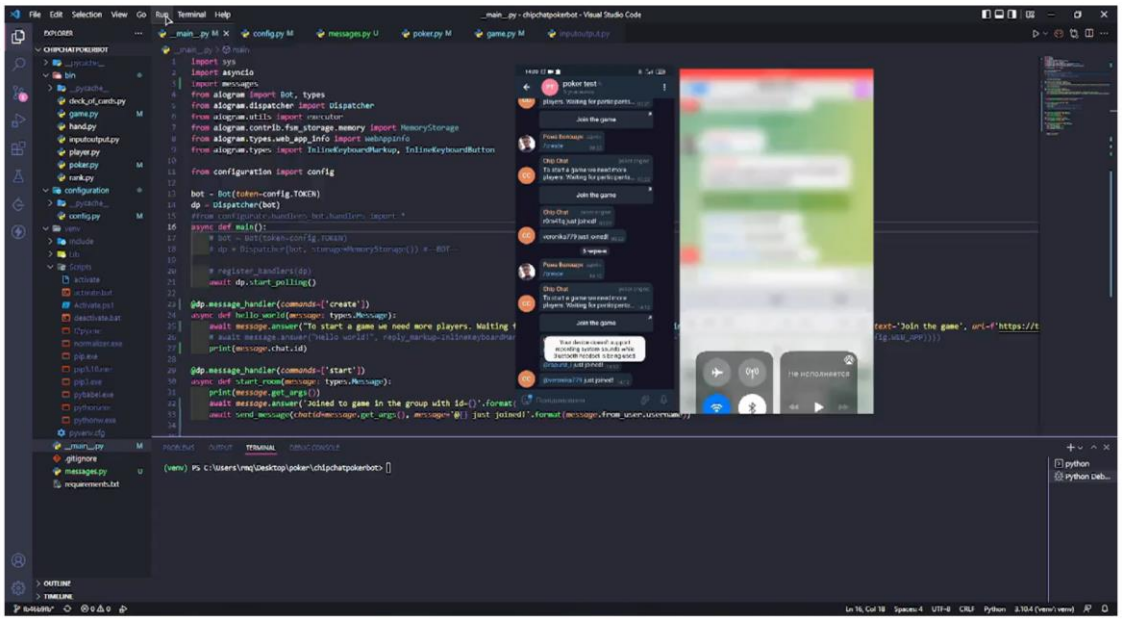

6

# ДІАГРАМА ПОСЛІДОВНОСТІ



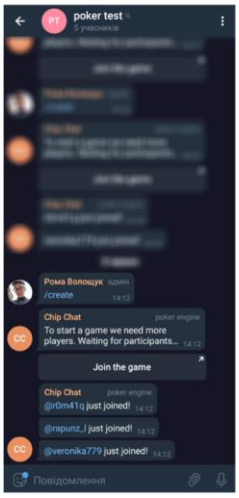

7

# ЕКРАННІ ФОРМИ

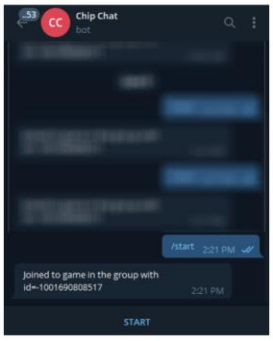


8

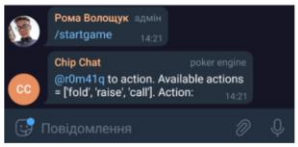
# ЕКРАННІ ФОРМИ



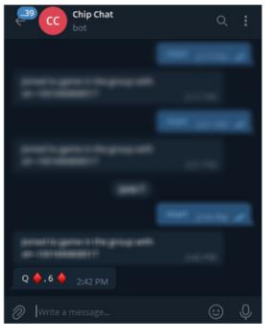

Відповідь на команду /create




Кнопка START (приєднання до гри)



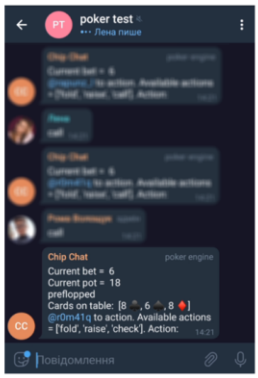

Відповідь на команду /startgame



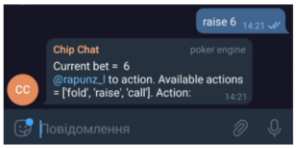

Повідомлення про карти на початку гри



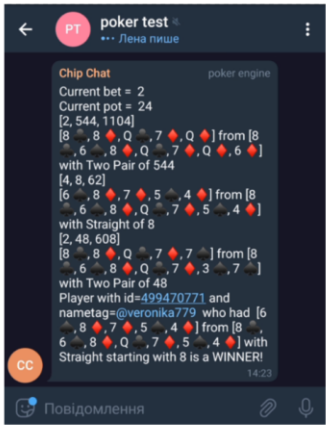
# ЕКРАННІ ФОРМИ




Повідомлення про зміну карт на столі




Відповідь на команду підвищення ставки (Raise n)




Повідомлення про переможну комбінацію

# АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Волощук Р.М. Розробка телеграм-бота для проведення сеансів гри в покер: матеріали Науково-технічної конференції «Застосування програмного забезпечення в ІКТ», м. Київ, ДУТ, 20 квітня 2022 року.



# ВИСНОВКИ

1. Проаналізовані вимоги, потреби, технічні можливості потенційної аудиторії даного програмного продукту. У результаті було обрано платформу Телеграм для розробки додатку з проведення сеансів гри в покер.
2. Проведений аналіз інструментів та програмних засобів реалізації, що покажуть найкращу продуктивність системи, під час застосування продукту в реальних умовах, та що спричинять найбільшу зручність під час написання продукту. У ході аналізу було обрано використання VisualStudio Code як середовища розробки, мову програмування Python, асинхронну бібліотеку Aiogram для взаємодії з API Telegram та систему керування базами даних MongoDB, розташовану в хмарному сховищі.
3. Спроектовано та розроблено телеграм-додаток для створення та проведення сеансів гри в покер серед учасників чату в месенджері Телеграм.
4. Проведено тестування телеграм-додатку та ігрової системи на виявлення помилок в роботі. Було досліджено більшу частину сегментів взаємодії користувача з додатком, помилок не було виявлено, система працює стабільно та надійно.