

Міністерство освіти і науки України  
Національний авіаційний університет  
Факультет кібербезпеки, комп'ютерної і програмної інженерії  
Кафедра комп'ютеризованих систем управління

Звіт  
з проектно-технологічної практики

студента 3 курсу СП-325 групи  
напряму 123 «Комп'ютерна інженерія»  
Клокуна Владислава Денисовича

База практики: ТОВ «Смарт Медіа Інвест»

Керівник практики:  
від університету — старший викладач Кашкевич Іван Фуркатович  
від бази практики — керівник проектів Чезганов Олександр Сергійович

Київ 2019

Підстава для проходження практики: робочий навчальний план № РС-4-6.05010202/12 підготовки фахівців за напрямом підготовки 6.050102 «Комп'ютерна інженерія», договір на проведення практики між університетом та ТОВ «Смарт Медіа Інвест», наказ ректора № 687/ст від 10.04.2019.

Термін проходження практики: з 03.06.2019 по 23.06.2019

Індивідуальні завдання: розробити програмний продукт — чат-бота для платформи Telegram.

ВІДМІТКА  
про проходження практики

Прибув на базу практики

«\_\_\_\_\_» \_\_\_\_\_ 2019 р. (підпис) \_\_\_\_\_ М. П.

Вибув з бази практики

«\_\_\_\_\_» \_\_\_\_\_ 2019 р. (підпис) \_\_\_\_\_ М. П.

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК ПРАКТИКИ

| №<br>п/п | Об'єкт практики та види робіт   | Термін виконання |            |
|----------|---|------------------|------------|
|          |   | початок          | закінчення |
| 1        | Знайомство з інфраструктурою і забезпеченням організації  | 03.06.2019       | 05.06.2019 |
| 2        | Постановка і аналіз задачі. Вибір необхідних інструментів для розробки. Затвердження специфікації | 06.06.2019       | 07.06.2019 |
| 3        | Розробка програмного продукту   | 10.06.2019       | 14.06.2019 |
| 4        | Розробка рішення для розгортання програми   | 17.06.2019       | 18.06.2019 |
| 5        | Тестування і запуск розробленого програмного продукту   | 19.06.2019       | 21.06.2019 |

Керівник практики від університету: \_\_\_\_\_

## **ЗМІСТ**

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Характеристика бази практики</b>                            | <b>4</b> |
| 1.1      | Загальні відомості про організацію . . . . .                   | 4        |
| 1.2      | Сфера діяльності . . . . .                                     | 4        |
| <b>2</b> | <b>Зміст та результати виконаних робіт</b>                     | <b>5</b> |
| 2.1      | Знайомство з інфраструктурою і забезпеченням організації . . . | 5        |
| 2.2      | Постановка і аналіз задачі . . . . .                           | 7        |
| 2.3      | Вибір інструментів . . . . .                                   | 8        |

## **1. ХАРАКТЕРИСТИКА БАЗИ ПРАКТИКИ**

### **1.1. Загальні відомості про організацію**

Організація ТОВ «Смарт Медіа Інвест» зареєстрована за адресою 47201, Тернопільська область, Зборівський район, місто Зборів, вулиця Б. Хмельницького. За класифікатором видів економічної діяльності вона є рекламним агентством, займається створенням, запуском, супроводом, просуванням і підтримкою медіапроектів у соціальних мережах на кшталт «Вконтакте», «Instagram» і «Telegram».

### **1.2. Сфера діяльності**

Щоб підтримувати медіапроект, необхідно збирати, створювати, оформлювати і публікувати інформацію, на якій він спеціалізується, розважати користувачів проекту, а також тримати його у належному стані. Для досягнення цих цілей і покращення результатів використовують різні інструменти: для розваги користувачів — мультимедійні матеріали і інтерактивні заходи, для підтримки проекту — спеціальні програмні продукти.

За кожною зі складових відповідають певні структурні підрозділи, які тісно взаємодіють між собою. Пошуком і створенням інформації, розробкою ідей, креативу, а також інтерактивних заходів займаються редактори. Мультимедійні матеріали для оформлення розроблених ідей надають дизайнери. За створення і запуск інструментів, необхідних для організації інтерактивних розважальних заходів і підтримки медіапроектів, відповідають розробники. Крім цього, розробники відповідають за налаштування і підтримку апаратно-програмної інфраструктури, яка необхідна для правильної роботи вже запущених продуктів, у справному стані.

Інструменти, створенням і запуском яких займаються розробники, можна класифікувати так:

- розважальні інструменти:
  - чат-боти,
  - боти-коментатори,
- супровідні інструменти:
  - очищувачі,
  - спеціалізовані сценарії (живі обкладинки тощо),
  - інструменти збору статистики.

Щоб зрозуміти, що входить в обов'язки розробників, розглянемо кожен клас цих інструментів детальніше. *Розважальні інструменти* призначені для того, щоб розважити користувача, безпосередньою взаємодією з ним. Прикладом розважального інструменту є *чат-бот* — програмний продукт, який веде діалог

лог (чат) з користувачем відповідно до заданого сценарію або специфікації. Наприклад, за допомогою чат-бота можна реалізувати текстову пригоду, головним героєм якого буде кожен окремий користувач, організувати лотерею або віртуального помічника, який відповідатиме на питання користувача.

*Бот-коментатор* — це програмний продукт, який додає коментарі у спеціальних дошках для обговорення у соціальних мережах залежно від певної умови. Наприклад, якщо користувач залишає коментар до запису про товар, в якому питає про деталь, відомому боту-коментатору, цей бот може залишити коментар з відповіддю на поставлене питання.

Наступною категорією є *супровідні інструменти*, тобто інструменти, призначені для автоматизованої підтримки представництв медіапроекту у певному стані. Першим представником супровідних інструментів є *очищувач* — програмний продукт який очищує ту чи іншу складову присутності медіапроекту у соціальній мережі: записи на дошці обговорень та її коментарі, зміст фото-, аудіо- і відеоальбомів тощо.

*Спеціалізований сценарій (або скрипт)* — це програмний продукт, який виконує різноманітні вузькоспеціалізовані дії в залежності від свого призначення. Поширеним прикладом спеціалізованого сценарію є «жива обкладинка» — програмний продукт, який змінює обкладинку представництва медіапроекту у соціальній мережі в залежності від певних параметрів: кількості записів, створених користувачами, проведених дій, переглядів тощо.

*Інструмент збору статистики* — це програмний продукт, який збирає статистику різних представництв медіапроекту у соціальних мережах, оброблює зібрані дані та експортує їх у потрібний формат. Результати роботи цих інструментів використовуються для аналізу, відстеження і планування стану та життєздатності проекту, а також для звітності.

Отже, як бачимо, компанія займається повним циклом управління медіа-проектами. Для цього вона використовує набір різноманітних інструментів, до яких також входять програмні інструменти, за створення і запуск яких відповідає підрозділ розробників, в якому автор проходив практику.

## **2. ЗМІСТ ТА РЕЗУЛЬТАТИ ВИКОНАНИХ РОБІТ**

### **2.1. Знайомство з інфраструктурою і забезпеченням організації**

Організована у компанії інфраструктура для розробки програмних рішень складається з апаратного і програмного забезпечення, а також віддалених сервісів. Віддалені (або *хмарні*) сервіси використовуються для координації команди і збереження її напрацювань. Організація використовує такі сервіси:

- хостинг для репозитаріїв початкового коду GitHub;
- система організації документообігу Google Suite: Docs, Sheets і Slides;

— дошка завдань Trello.

Хостинг для репозитаріїв початкового коду GitHub потрібен, щоб зберігати і відстежувати версії початкового коду усіх програмних продуктів і інструментів, які створюються і використовуються в організації. Система організації документообігу Google Suite використовується для звітності та обміну документами між працівниками компанії. На дошці завдань Trello відстежують завдання, які зараз поставлені перед працівниками, і оцінюють їх завантаженість, щоб планувати наступні дії.

До апаратного забезпечення входить виділений сервер, розташований за межами компанії, на якому встановлене програмне забезпечення, необхідне для розробки продуктів, якими користується організація, зокрема:

- операційна система GNU/Linux Ubuntu LTS;
- веб-сервер Nginx;
- система керування базами даних MongoDB;
- інтерпретатор мови програмування Python;
- система управління контейнерами Docker.

Розглянемо вищезазначене програмне забезпечення детальніше. Операційна система GNU/Linux Ubuntu LTS дозволяє встановити всі засоби, необхідні для правильної роботи потрібних продуктів, наприклад, Docker Server Community Edition. На робочій машині встановлена серверна версія дистрибутиву, що дозволяє зменшити обсяг пам'яті, який займає система, бо в ній встановлюється лише набір мінімально необхідних компонентів. Позначка LTS (Long-Term Support) означає, що ця версія дистрибутиву буде підтримуватись тривалий час, тобто в разі виникнення проблем з системою можна розраховувати на необхідну підтримку протягом декількох років з моменту випуску даної версії.

На операційній системі встановлений веб-сервер Nginx, необхідний для запуску програмних продуктів, які надають послуги в мережі Інтернет, зокрема веб-додатків та веб-сайтів. На одному сервері водночас можуть працювати одразу декілька продуктів, тому веб-сервер виступає як зворотний проксі (англ. *reverse proxy*), тобто отримує запити від зовнішніх клієнтів і перенаправляє їх на інші, локальні веб-сервери.

Система керування базами даних MongoDB використовується для збереження і управління постійними відносно структурованими даними, які створюються під час роботи веб-додатків.

Основною мовою програмування для розробки продуктів в організації є Python, тому на сервері встановлений її інтерпретатор. З його допомогою розробники створюють модулі та пакети на мові програмування Python і запускають їх.

Щоб зручно розгортати продукти і запускати їх в ізольованому середовищі, яке можна точно відтворити на будь-якій сумісній платформі, використовують систему управління контейнерами Docker.

## 2.2. Постановка і аналіз задачі

Для проходження проектно-технологічної практики була поставлена така задача: розробити чат-бота для платформи «Telegram», який надсилає казки своїм користувачам. Коли користувач запускає бота, він повинен привітати його і надіслати випадкову казку. Якщо користувач просить надіслати ще, бот надсилає наступну випадкову казку. Також, бот повинен регулярно розсилати казки своїм користувачам: щодня в певний час.

Щоб наочно зобразити високорівневу функціональність бота, була розроблена UML-діаграма варіантів використання чат-бота (рис. 1).

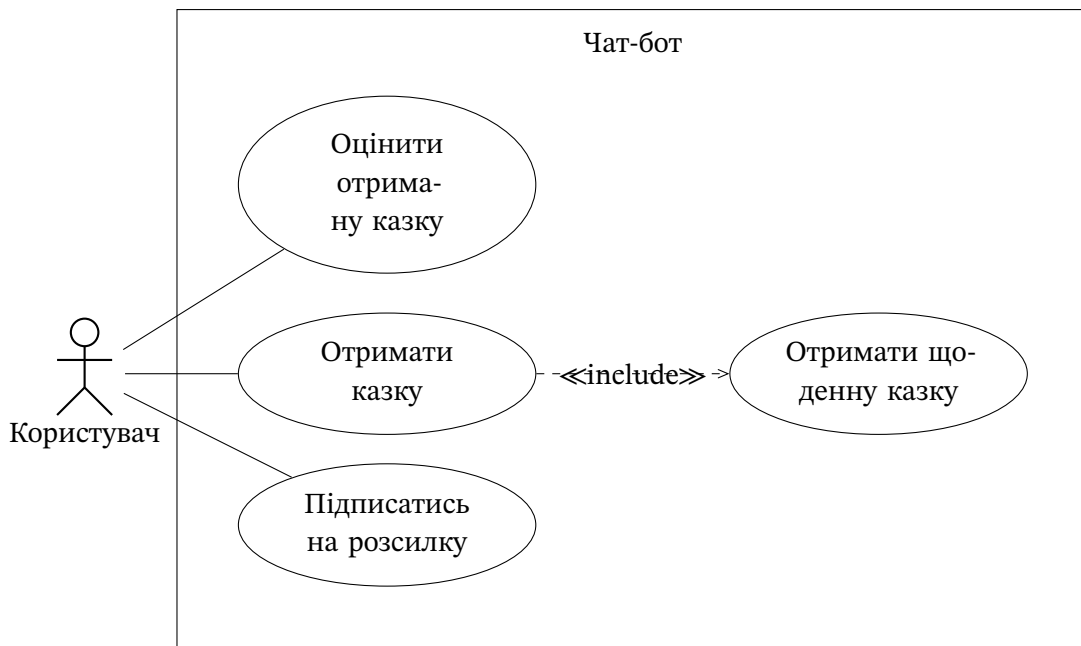


Рис. 1: Діаграма варіантів використання чат-бота

Розроблена діаграма варіантів використання допомогла затвердити вимоги до програмного продукту з працівниками, які розробляли ідею даного проекту, і знадобилась під час розробки архітектури програмного продукту.

Щоб проаналізувати задачу, розділимо її на підзадачі. В результаті декомпозиції отримаємо таку ієрархію:

1. Регулярно надсилати випадкову казку.
2. Обробити запит користувача:
  - 2.1. Визначити тип запиту.
  - 2.2. Додати користувача в список активних користувачів чат-бота.
  - 2.3. Дати відповідь на запит:
    - 2.3.1. Надіслати повідомлення-привітання.

- 2.3.2. Надіслати повідомлення з інструкцією, як користуватись чат-ботом.
- 2.3.3. Обрати випадкову казку.
- 2.3.4. Надіслати текст обраної казки.
- 2.3.5. Додати казку у список казок, що подобаються.

### 2.3. Вибір інструментів

Як відомо зі знайомства з інфраструктурою компанії, основною мовою програмування для створення продуктів є Python, тому для розробки чат-бота буде використана саме вона.

Розглянемо діаграму варіантів використання (рис. 1), з якої видно, що при взаємодії з системою (Чат-ботом) актор (Користувач) може виконувати такі дії:

- Підписатись на розсилку, тобто дати згоду на те, щоб бот працював з користувачем, який до нього звертається.
- Отримати казку, тобто надіслати запит чат-боту, щоб він надіслав користувачу казку.
- Отримати щоденну казку, яка є розширенням варіанту використання «Отримати казку». Якщо користувач підписався на розсилку, бот надсилає щоденну казку за заданим розкладом.
- Оцінити отриману казку, тобто мати змогу виразити, що казка сподобалась користувачу.

Основною можливістю є взаємодія з чат-ботом. Він повинен працювати на платформі Telegram, тож необхідно використати її програмний інтерфейс (англ. *application programming interface, API*). Програмний інтерфейс Telegram, а тому і робота більшості чат-ботів побудована на основі двох дій: отримання і обробки оновлень та постановки запитів до серверів Telegram API.

Отримання оновлень дозволяє зрозуміти, які дії користувачі або інші треті сторони проводять з ботом. Наприклад, які повідомлення надсилають, які кнопки натискають та інше. Запити дозволяють визначити, які дії повинен провести сам чат-бот: яке повідомлення відправити у відповідь, яку інформацію отримати або оновити тощо.

Щоб організувати взаємодію з платформою Telegram, скористаємось зовнішнім пакетом «Python Telegram Bot» (знаходиться за адресою <https://github.com/python-telegram-bot/python-telegram-bot>), який надає модулям, написаним на мові програмування Python зручний доступ до інтерфейсу програмування платформи Telegram.

Цей модуль дозволяє отримувати оновлення за допомогою двох методів: довгого опитування (англ. *long polling*) і веб-гачка (англ. *webhook*).

Щоб кожен день розсилати казки користувачам, які звернулись до бота і підписались на розсилку, необхідно вести їх облік. Хорошою практикою при



розробці веб-додатків є принцип відсутності стану: дії, які виконує програма — веб-додаток, не повинні залежати від її внутрішнього стану. Отже, облікові дані користувачів варто зберігати зовні, за логічними межами програми.

Так як облікові дані користувачів будуть часто змінюватись (будуть з'являтися нові користувачі, оцінки тощо), для збереження цих даних зручно використати базу даних. В організації вже використовують систему керування базами даних MongoDB, тому для вирішення поставленої задачі використаємо її.

Щоб отримати доступ до бази даних з Python-модуля, використаємо пакет PyMongo, який надає офіційний розробник системи керування базами даних MongoDB.