

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ПРИРОДОКОРИСТУВАННЯ**  
**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ**  
**ТЕХНОЛОГІЙ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КВАЛІФІКАЦІЙНА РОБОТА**

другого (магістерського) рівня вищої освіти

на тему: «Розробка telegram бота для інформаційної підтримки навчального процесу у Львівському національному університеті природокористування із вибором ефективних засобів»

Виконав: студент 6 курсу групи Іт-61

Спеціальності 126 «Інформаційні систем  
технології»

(шифр і назва)

Фіялковський Віталій Іванович

(Прізвище та ініціали)

Керівник: к.т.н., в.о. доцента Падюка Р.І.

(Прізвище та ініціали)

Рецензент: к.т.н., доцента Тимочко В.О.

(Прізвище та ініціали)

Дубляни 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ МЕХАНІКИ ТА ЕНЕРГЕТИКИ  
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти

Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

д.т.н., проф. А. М. Тригуба

\_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

на кваліфікаційну роботу студенту

Фіялковський Віталій Іванович

1. Тема роботи: «Розробка чат-боту розкладу занять у Львівському національному аграрному університеті для месенджера Telegram»

Керівник роботи Падюка Роман Іванович, доцент

затверджені наказом по університету від 26.01.2022 року № 21/к-с.

2. Строк подання студентом роботи 10.06.2021 р.
3. Вихідні дані до роботи
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Вступ.

1. Аналіз предметної області
2. Постановка задачі
3. Проектування інформаційної системи чат-боту з розкладу занять у Львівському національному аграрному університеті для месенджера Telegram
4. Практичне використання інформаційної системи

5. Охорона праціВисновки та пропозиціїСписок використаної літератури

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових схем та моделей): особливості планування розробки чат-боту розкладу занять у Львівському національному аграрному університеті для месенджера Telegram; огляд систем аналогів; результат вибору методів та засобів реалізації проекту; результат проектування інформаційної системи; реалізація та практичне використання чат-бота

## 6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3,6	<i>Падюка Р. І., доцент кафедри ІСТ</i>		
4	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

## 7. Дата видачі завдання

26 січня 2022р.

## Календарний план

№з/п	Назва етапів кваліфікаційного проекту	Термін виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	20.03.2021	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	25.03.2021	
3	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	26.03.2021	
4	<i>Написання розділу «Охорона праці»</i>	05.04.2021	
5	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	07.04.2021	
6	<i>Завершення роботи в цілому</i>	28.05.2021	

Студент

Фіялковський В. І.

Керівник роботи

Падюка Р. І.

УДК 004.9 : 631.1

Розробка telegram бота для інформаційної підтримки навчального процесу у Львівському національному університеті природокористування із вибором ефективних засобів

Фіялковський В. І. Кафедра інформаційних технологій – Дубляни, Львівський НУП, 2022.

Кваліфікаційна робота: ст. 54, список літературних джерел та інших джерел 30, таблиць: 1.

Виконано аналіз предметної області завдяки ідентифікації ідей проєкту, був проведений огляд інформаційної системи аналогів та побудований загальний аналіз роботи інформаційної системи.

У роботі побудований зміст та структуру виконаної роботи, мета задачі та спосіб її вирішення. За допомогою інформаційних систем було побудовано діаграми декомпозиції, контекстну та діаграма варіантів використання. Створено базу даних для системи та змодельована архітектура системи.

На основі задачі було побудовано систему, яка виконує функцію розкладу у месенджері «Telegram».

Значення даної роботи полягає у розробленні бота, який у месенджері «Telegram» буде надавати підтримки користувачам, виходячи з вибраних категорій. Це дозволить користувачам набагато швидше отримувати потрібно їм інформацію.

Розроблено заходи щодо охорони праці

## ЗМІСТ

ВСТУП.....	7
1. АНАЛІЗ СТАНУ ПИТАННЯ В ТЕОРІЇ ТА ПРАКТИЦІ.....	9
1.1. Аналіз чат-ботів .....	9
1.2. Чинні аналоги телеграм боту .....	11
1.3. Аналіз чинних засобів підтримки навчального процесу .....	14
2. ОБГРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ РОЗВ’ЯЗАННЯ ЗАДАЧІ .....	16
2.1. Чому вибір випав саме на чат-бот.....	16
2.2. Вибір засобів реалізації.....	16
2.3. Аналіз інструментарію розв’язання задачі з чат-ботом .....	19
3. РЕЗУЛЬТАТИ РОЗВ’ЯЗАННЯ ЗАДАЧІ.....	23
3.1. Результат побудови задачі.....	23
3.2. Етапи розвитку функціонала бота для підвищення ефективності навчального процесу в ЛНУП.....	25
3.3. Аналіз притоку користувачів удосконаленого бота.....	26
3.4. Результат опитування студентів стосовно функціонала розробленого бота.....	29
4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	32
4.1 Структурно функціональний аналіз процесу розробки інформаційної системи .....	32
4.2. Моделювання процесу виникнення травм та аварій.....	32
4.3. Розробка логічно-імітаційної моделі травм .....	33
4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях .....	37
5. ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ.....	40
5.1. Скільки часу пішло на створення бота .....	40

5.2. Розрахунок трудомісткості розроблення та використання .....	40
5.3. Вартість розробленого продукту.....	41
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТОК А .....	47
ДОДАТОК Б.....	51

## ВСТУП

Як і інші види мов програмування, Java дає свій спосіб вираження понять. У разі успішного використання цього засобу подальше використання буде значно простішим та гнучкішим, що дасть змогу не використовувати інші варіанти використання, оскільки нові функціонали потребують складнішої роботи з мовою програмування. Щоб працювати з мовою програмування Java без проблем, ви повинні навчитись взаємодіяти з нею та мати хоча б базові знання з програмування.

Завдяки цьому, можна розкрити усі особливості даної мови, що дає нам змогу виконати конкретну задачу, яка нам поставлена. Це дасть нам змогу, більш детально пояснити роботу з даною мовою програмування та перенести вас до моменту, коли функціонування Java стане більш доступною для розуміння. Протягом часу, що буде відведено на виконання поставленої задачі, ви можете стикнутись з різними видами проблем, але вирішення кожної з них дасть вам змогу розвинути краще розуміння мови, що надалі дасть змогу впоратись з будь-якою головоломкою, яка може з'явитись під час виконання роботи.

У нашій роботі передбачається, що ви працювали з мовами програмування і розумієте їх. Ви повинні знати, що програма містить у собі сукупність операторів, функцій, макроси, і різні типи операторів, а саме: оператор "if" та циклічні конструкції "while" та інші. Проте, ви могли дізнатись про це в інших мовах програмування, наприклад у програмуванні макросами. Коли ви покращуєте свої знання з програмування то незабаром вам буде легко почати працювати з базовими ідеями програмування, ви почнете розуміти будову будь-якої задачі. Ці нові знання дають вам змогу працювати з мовами програмування, які вам були новими до цього часу, а саме мови: C та C ++.

Проте ми будемо проводити презентацію роботи концепції об'єктноорієнтованого програмування та основний функціонал керування Java. Хоч мова програмування Java має подібний функціонал з C та C++, слід не забувати що у кожній з них є свій функціонал, яка відрізняє їх між собою.

Усі мови програмування використовують абстракції. Можна стверджувати, що будь-яку складну проблему, яку ви можете вирішити завдяки певному виду абстракції або її якості.

Асемблерна мова - це функціональна абстракція стандартної машини. Багато так званих нових мов програмування, що появились з часом (наприклад, Python, C++ та C), були кращими варіантами асемблерної мови. Ці мови містили у собі значну кількість нового функціоналу та покращень порівняно з мовою асемблер, але їх основна задача все одно потребує від вас виконання поставленої задачі з точки зору структури комп'ютера, а не структури проблеми, яку вам потрібно виправити. Програмісту потрібно провести лінію зв'язку між структурою машини та структурою проблеми, яка легко вирішується. Зусилля, які нам потрібні для виконання поставленої задачі, і те містить у собі декілька мов програмування, створює програми, проблемні для написання та обслуговування. Отож, даний ефект створив цілу галузь "методів програмування" для розв'язання такої задачі.

Отже, метою даного кваліфікаційного проєкту є створення «Telegram» бота для інформаційної підтримки користувача у навчальному закладі.

Під час виконання даного проєкту потрібно виконати поставлені задачі, а саме:

- Провести аналіз чинних аналогів
- Сформулювати функціональні вимоги для створення Бота
- Створити модель роботи інформаційної системи
- Обрати мову програмування та спосіб її реалізації
- Спроекувати рекомендацію для спрощеного користування ботом
- Виконати реалізацію проєкту в програмному середовищі та провести

тестування.

**Об'єктом** дослідження роботи є створення «Telegram» бота для отримання інформаційної підтримки користувачу

**Предметом** дослідження є бот який реалізує інформаційну підтримку користувача у месенджері «Telegram»



# 1. АНАЛІЗ СТАНУ ПИТАННЯ В ТЕОРІЇ ТА ПРАКТИЦІ

## 1.1. Аналіз чат-ботів

Чат-бот – це інформаційний помічник, який використовує віртуальну комунікацію з користувачами завдяки повідомленню і має велику кількість унікальних функцій. Інформаційного чат-бота можна використати як для відправлення інформації, так і для її збору. На сьогодні такого типу месенджери користуються великою популярністю. Це пов'язано зі значним ростом мобільного інтернету, що супроводжується покращенням швидкодії, низькою ціною і широким розповсюдженням смартфонів.

На цей момент існує великий різновид чат-ботів в клієнті по обміну повідомленнями, наприклад, в категоріях: розваги, фото і відео, новини, наука, ігри, соціальні комунікації, онлайн інструменти. В цих категоріях можна знайти потрібного нам чат-бота, та отримати потрібну нам інформацію. Крім того, у користувача є можливість збільшити його інструментарій завдяки цікавим опціям і отримувати інформацію від них. При цьому йому не немає потреби встановлювати різного типу додатки для отримання потрібної інформації, тому, що це все можна знайти в одному додатку – в месенджері.

Створення чат-бота в месенджері Telegram може дати можливість:

- отримувати сповіщення про новини, якими інтересується користувач надсилаючи йому відповідний контент, як тільки він опублікується;
- взаємодіяти з іншими послугами: бот може збагатити чати контентом від зовнішніх служб, а саме Gmail bot, Image bot, GIF bot, bot IMDB, Wiki bot, Music bot, Youtube bot, GitHub bot;
- приймати грошову подяку для автора бота
- створювати спеціальні інструменти, бот може надати сповіщення, прогнози погоди, переклади, форматування або інші послуги.

Тема чат-ботів є досить актуальною, про що свідчать численні публікації останніх років. Створення Telegram-бота засобами мови Java розглядається в статті Антонюка О.В [1], питаннями розробки Telegram-бота для інформаційної підтримки саме навчального процесу займався Коляструк Б.І. [2], Цукрук В.І. розглянув особливості розробки ігрового бота на основі месенджера Telegram [4].

До переваг месенджерів входить: охоплення більшого відсотка користувачів, можливість надсилати іншим різний контент.

Також головними перевагами месенджерів є можливість завжди залишатися на зв'язку, якщо є інтернет, що дає змогу цілодобово підтримувати розмову з користувачем.

У таких систем є такі недоліки:

- Боти не завжди розуміють, що від них саме потрібно користувачеві. Боти дають відповіді тільки на прості та точно сформульовані запитання.
- У бота немає емоцій. Деяким клієнтам важливий «живий» діалог, у якому оператор проявляє людські риси характеру. Бот поки ще не здатний на подібне.
- Якщо користувач робить орфографічні чи стилістичні помилки, система може хибно розпізнати текст і, як наслідок, неправильно відреагувати на запит.
- Легко може бути заблокована користувачем за розсилання спаму.

Сьогодні боти помічники можуть взяти на себе частину функцій HR-працівників, менеджерів, консультантів, продавців, працівників банку та інших фахівців. Проте повністю замінити людину вони поки не здатні.

## 1.2. Чинні аналоги телеграм боту

### Аналог №1 – «CHDTU Timetable Bot»

Бот був зроблений студентом Черкаського державного технологічного університету. Вигляд чат-бота на рисунку 1.1.

**Переваги:** Переваг немає так, як це ідентична платформа, більше інформації немає.

**Недоліки:** Недоліки даної роботи полягають у тому, що користуватись ботом зможуть невелика кількість людей так, як використано будову «Telegram api» , яка дає змогу користувачу вибирати, що йому потрібно. Та проблема полягає у тому, якщо робити так для усього університету, то буде дуже багато інформації і розташування почне зміщуватись, що приведе до незручного користування ботом. Судячи з виконаної роботи, розклад створений для однієї групи.

### Аналог №2 – «NuppBot»

Бот був зроблений студентами Полтавської політехніки імені Юрія Кондратюка. Вигляд чат-бота на рисунку 1.2.

**Переваги:** Перевага полягає у тому, що користувач ставить розклад отримування сповіщень, у них це 20 хв до початку пари.

**Недоліки:** Не зручне користування, лишня інформація на одному повідомленні від бота заважає отримати точну інформацію, яку задав користувач при запиті, використовується лише українські літери, що зазвичай не зручно, коли є можливість обирати розклад по переходах. Немає практичної рекомендації для нових користувачів.



Рисунок 1.1 – Головна сторінка чат-бота CHDTU Timetable Bot

### Аналог №3 – «ForStudyBot»

Бот був створений завдяки Одеському національному політехнічному університету. Вигляд чат-бота на рисунку 1.3.

**Переваги:** Творці бота не вказали його переваги.

**Недоліки:** Дуже багато інформації, яка розташована по секторам, база даних буде забита інформацією, яка не цікавить студента якому потрібен розклад.

Ключовими факторами для майбутнього бота є зручний та зрозумілий інтерфейс, надання практичних рекомендацій для користувача, збільшена кількість груп користувачів, можливість збереження розкладу в БД.

Отож, у кожного з аналогів є свої переваги та недоліки, які ми виправимо при побудованому чат-бота для Львівського НУП.

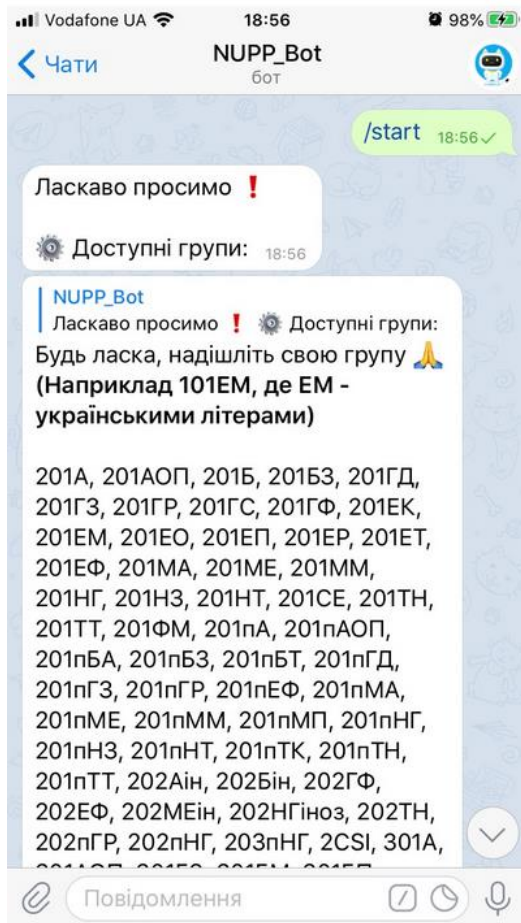


Рисунок 1.2 – Головна сторінка чат-бота «NuppBot»

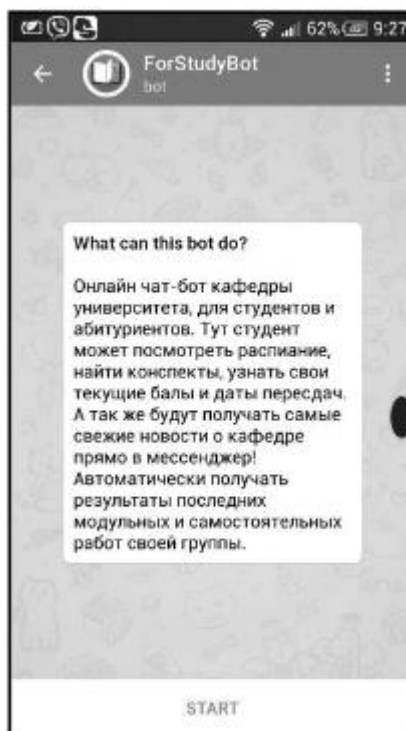


Рисунок 1.3 – Головна сторінка чат-бота «ForStudyBot»

Побудуємо таблицю порівняння для аналогічних ботів в месенджері telegram з розкладом для студентів та майбутньої інформаційної системи:

Таблиця 1.1 – Порівняльні характеристики аналогічних чат-ботів

Критерії Боти	Зручність інтерфейсу	Надання практичних рекомендацій	Кількість груп користувачів	Можливість збереження результатів в БД
CHDTU Timetable Bot	+	-	-	-
NuppBot	-	+	+	-
ForStudyBot	-	+	-	+

Судячи з результатів проведеного тесту аналогів показаних у Таблиці 1.1, можна привести висновок, що жоден з аналогів не має повну підтримку бота та користувача, а тому для створення бота для Львівського національного університету природокористування потрібно використати перераховані факти.

### 1.3. Аналіз чинних засобів підтримки навчального процесу

Медіатека - це середовище, в якому користувачі можуть легко та ефективно працювати з різними інформаційними ресурсами (надруковані видання і листові матеріали, аудіо та відеодані, машинописна інформація), які перебувають як у бібліотеках, так і за її межами.

Соціальна мережа - інтерактивний, багатокористувацький вебсайт, контент якого доповнюється самими учасниками мережі. Сайт являє собою автоматичне соціальне середовище, що дозволяє обмінюватись думками групі користувачів, об'єднаних спільним інтересом.

Інтернет аудиторія дозволяє заощадити час, минаючи етап навчання студентів до нового комунікативного простору, яке дозволяє побудувати

неформальне спілкування між викладачем і студентами та допомагає організувати орієнтоване навчання. Достатньо високий рівень взаємодії викладача та студента забезпечується покращенням навчального процесу, що виходить за рамки аудиторних занять. Можливість об'єднання індивідуальних і групових форм роботи сприяє покращенню засвоєнню матеріалу.

## **Висновки**

Завдяки новим інноваційним підходам пов'язаним з використанням організації соціальних мереж в освітніх цілях, почалось стимулювання пізнавальної активності студентів і, як результат, підвищується якість освіти. Навчатися потрібно там – де весело і цікаво, що покращує відчуття студента і дає йому приріст сил для науки. Освіта повинна йти в ногу з часом і відстежувати останні тенденції спілкування між студентом та викладачами, це дозволить більш легко та невимушено давати нові знання студентам. Впровадження соціальних мереж в навчальний процес може стати переломною ланкою для істотно підвищеної якості освіти не тільки в ВНЗ, але і в школах.

## **2. ОБГРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ РОЗВ'ЯЗАННЯ ЗАДАЧІ**

### **2.1. Чому вибір випав саме на чат-бот**

Отже, чому саме чат-бот? Відповідь на питання досить просте, це покращена взаємодія студента з навчальним процесом та побудова розкладу навчання. Бот є досить простий у використанні, що дає змогу студенту, викладачу або працівнику ВНЗ легко почати з ним працювати. Це легкий спосіб отримати швидко інформацію від викладача, яка потрібна студенту для навчання, наприклад: методичні рекомендації, книжки чи посилання на сайт з потрібною інформацією для студента. На сьогодні студенти користуються месенджерами для особистих потреб, таких як спілкування з іншими, пошук інформацій або ж для забави. Завдяки чат-боту, університет може покращити доступ до інформації студенту або ж викладачу.

### **2.2. Вибір засобів реалізації**

Для того, щоб зробити коректну та стабільну роботу інформаційної підтримки, необхідно буде вибрати, які саме інструменти взаємодії системи реалізації ми будемо використовувати. Для виконання поставленої задачі, ми вибрали мову програмування для створення бота – Java, інструментом управління та розуміння проєкту було обрано Maven, комплексна модель для конфігурації системи Spring Framework, хмарна платформа, яка дозволяє створити віддалене з'єднання користувачів без взаємодії хоста, доставляти інформацію, контролювати та масштабувати програми - Heroku та база даних, яку надав сайт heroku PostgreSQL.

#### **Java**

Для того, щоб розробити інформаційну систему підтримки потрібно використати мову програмування Java. Java – мова програмування, яка є швидка у



використанні безпечна у роботі з системами та надійна у використанні, приклад роботи на рисунку 2.1.

```

1   public class Dog extends Mammal{
2       public static void main(String args[]){
3           Animal a = new Animal();
4           Mammal m = new Mammal();
5           Dog d = new Dog();
6
7           System.out.println(m instanceof Animal);
8           System.out.println(d instanceof Mammal);
9           System.out.println(d instanceof Animal);
10      }
11  }
```

Рисунок 2.1 – Приклад використання Java

## Maven

Maven - інструмент управління та розуміння проєкту. Завдяки якому тепер можна побудувати та керувати будь-яким проєктом на основі Java. Даний інструмент покращить роботу з проєктом, приклад роботи на рисунку 2.2. Також перевагами є:

- Контролювання збірки системи
- Створення звіту роботи
- Документація, яка надходить з центральної частини
- Адаптивне використання
- Додаткові функції взаємодії підсистеми XML

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>TelegramBot</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.telegram</groupId>
      <artifactId>telegrambots</artifactId>
      <version>3.6</version>
    </dependency>
  </dependencies>
</project>
```

## Рисунок 2.2 – Приклад використання Maven

### Spring Framework

Spring Framework (Spring) – дозволяє нам забезпечувати коректну роботу моделі для програмування та конфігурацію системи на основі мови програмування Java у будь-який спосіб розгортання проєкту. Spring є інфраструктурна підтримка для додатків, приклад роботи на рисунку 2.3.

```
server.port=5432
localeTag=ua-UA

pingtask.url= https://www.google.com
pingtask.perion= 1200000

telegrambot.botUserName=@StudyLnauBot
```

Рисунок 2.3 – Приклад використання Spring Framework

### PostgreSQL

PostgreSQL – це сильна об'єктно-реляційна система база даних з відкритим кодом, що надало їй міцну репутацію надійності та продуктивності. Використовується для доступу до бази даних, у нашому випадку вона створена у хмарному хостингу «Heroku», приклад роботи на рисунку 2.4

```
Class.forName("com.mysql.jdbc.Driver");
try (Connection conn = DriverManager.getConnection(url, username, password);
    Statement stat = conn.createStatement()) {
    stat.executeUpdate( sql: "drop table schedule");
    stat.executeUpdate( sql: "CREATE TABLE IF NOT EXISTS schedule(id MEDIUMINT NOT NULL AUTO_INCREMENT, lesson char(120),`groups` char(120),d
    stat.executeUpdate( sql: "INSERT INTO schedule (lesson,`groups`,day,par) VALUES ('Веб-технології та веб-дизайн 55M','IT-41','Понеділок','Д
```

Рисунок 2.4 – Приклад запису бази даних

### Heroku

Heroku - яка дозволяє створити віддалене з'єднання користувачів без взаємодії хоста, доставляти інформацію, контролювати та масштабувати програми

– це дозволяє знайти найшвидший спосіб перейти від ідеї до URL-адреси, приклад роботи на рисунку 2.5.

```
Building on the Heroku-20 stack
Using buildpack: heroku/gradle
Gradle app detected
Spring Boot detected
Installing JDK 1.8... done
Building Gradle app...
executing ./gradlew build -x check
```

Рисунок 2.5 – Приклад деплою на Heroku

У нашому проєкті застосування heroku дає змогу працювати чат-бота підтримки в автономному режимі без нашого втручання в його роботу, проте дана платформа не дає змогу повністю підтримувати бота в постійному автономні тому у надалі варто перейти на платформу, яка зможе це виправити. Завдяки Heroku ми можемо користуватись базою даних, яка створюється на самому проєкті хостингу, ще однією перевагою є те що платформа синхронізується з github, що дає змогу синхронізувати код у обох місцях одночасно.

### 2.3. Аналіз інструментарію розв’язання задачі з чат-ботом

Тепер, коли ви знаєте деякі з багатьох варіантів використання розмовного чат-бота зі штучним інтелектом, ось деякі з найкращих інструментів для розмовного штучного інтелекту на ринку для створення чат-ботів.

#### 1. Kore.ai

Kore.ai — це розмовний інструмент, який надає рішення автоматизації для клієнтів, співробітників і агентів. Його платформа пропонує функції для покращення досвіду роботи з клієнтами, агентами, співробітниками та пошуком. Kore.ai має приклади використання в багатьох галузях, включаючи роздрібну торгівлю, засоби масової інформації, розваги, банківську справу, охорону здоров’я, авіакомпанії, роздрібну торгівлю, страхування тощо. Згідно з документацією Kore.ai, його основні функції:

- Запатентований багатосторонній НЛП-підхід

Цей розмовний інструмент штучного інтелекту використовує багатокомпонентний підхід до обробки природної мови (NLP). Цей підхід дає змогу створювати інтелектуальних ботів, які розуміють людські розмови, наміри, емоції, настрої, контекст і відповідно виконують завдання

- Аналітика ботів і інформаційна панель

Ви отримуєте інструменти візуалізації та даних для відстеження та оцінки продуктивності вашого бота, що допомагає вам глибше зрозуміти поведінку співробітників, клієнтів і ботів. Ви можете відстежувати розмовну, функціональну аналітику та залучення ботів із централізованої інформаційної панелі, щоб зрозуміти взаємодію користувачів.

Без кодова платформа Kore.ai надає вам безпечне та масштабоване наскрізне рішення, яке дає змогу створювати та тестувати розмовні чат-боти. Хоча у Kore.ai є безплатна версія та безплатна пробна версія, вам доведеться одноразово сплатити 500 доларів США за корпоративні функції.

Плюс:

- Це дозволяє без особливих зусиль створювати складні чат-боти, додавати канали тощо. Ви можете використовувати візуальний інтерфейс розробки, щоб скидати та перетягувати компоненти для створення програми без будь-якого технічного досвіду.

- Це полегшує розробникам створення контекстно-залежних розмов корпоративного рівня без необхідності писати код.

Мінуси:

- У Kore.ai немає серйозних недоліків. Користувачі зазвичай мають проблеми лише з інтеграцією каналів.

## 2. Boost.Ai

Boost.Ai — це розмовний інструмент, який використовується для створення чат-ботів для автоматизації внутрішніх каналів підтримки та обслуговування клієнтів. Це також збільшує дохід і підвищує ефективність роботи людей через онлайн-розмови. Він пропонує рішення для таких галузей, як банківська справа, телекомунікації, страхування тощо. Деякі з його основних функцій включають:

- Статус позову:

Boost.Ai пропонує повний огляд продуктивності вашого розмовного чат-бота за допомогою багатоцільової інформаційної панелі. Це дає вам змогу відстежувати рейтинги відгуків, якість конверсії та інші відповідні показники ефективності для оптимізації UX.

- Детальний аналіз повідомлень:

Це дає вам детальний аналіз кожного повідомлення користувача, яке ви отримуєте, включно з важливими словами та оцінкою прогнозів, таким чином дозволяючи вам краще зрозуміти свого чат-бота.

Крім того, ви маєте розширену аналітику розмов, необмежену масштабованість і конструктор бесід без коду, що означає, що ви можете створювати розширені потоки розмов за лічені хвилини.

Плюси:

- Boost.Ai відомий своєю першокласною технологією, завдяки якій надзвичайно легко навчати та підтримувати бота; його також легко налаштувати та використовувати.
- За допомогою Boost.Ai менеджери можуть отримати глибоке уявлення про важливі показники, такі як дані розмов і оцінки якості. Він також підтримує інтеграцію зі сторонніми програмами, такими як eGain, Zendesk.

Мінуси:

- Єдиною проблемою, яка, здається, турбує Boost.Ai, є відсутність зв'язку щодо дорожньої карти продукту.

## **Висновки**

Завдяки ботам ми покращуємо своє життя, вони спрощують нам доступ до інформації, яка нам потрібна, на цей час вони вже існують усюди де ми тільки можемо це побачити, вони допомагають людям робити замовлення без розмови з консультантами, перевіряти доступні квитки на поїзда або літаки, а зараз вони помагають нам дізнатись про ситуації зі світлом, раніше ще був бот який існував аби повідомляти водіям де зараз є пальне. Боти можуть бути створені на різних мовах програмування, та зв'язує їх одне, це підтримка користувачів.

### 3. РЕЗУЛЬТАТИ РОЗВ'ЯЗАННЯ ЗАДАЧІ

#### 3.1. Результат побудови задачі

Наша задача полягала у тому щоб розробити telegram бота для інформаційної підтримки навчального процесу у Львівському національному університеті природокористування із вибором ефективних засобів. Ми створили бота у месенджері, отримали ключ котрий дасть нам змогу почати працювати з ним та для його майбутніх модифікації.

Почали розробляти алгоритм за яким користувач буде отримувати потрібну йому інформацію. Зробили поділ потрібної інформації для користувача на розклад навчання та посилання на сайт, реквізити, опитування і конференції в zoom.

Було створено окремий розділ куди користувач міг звернутись, щоб отримати відповідь на своє питання, також надавалась можливість користувачу запропонувати свою змінну в бота, а саме що корисного вони б хотіли бачити у ньому. Для більш точної інформації була створена контекстна діаграма IDEF0, діаграма варіантів використання, структура бази даних та архітектура інформаційної системи(рис. 3.1-3.4).

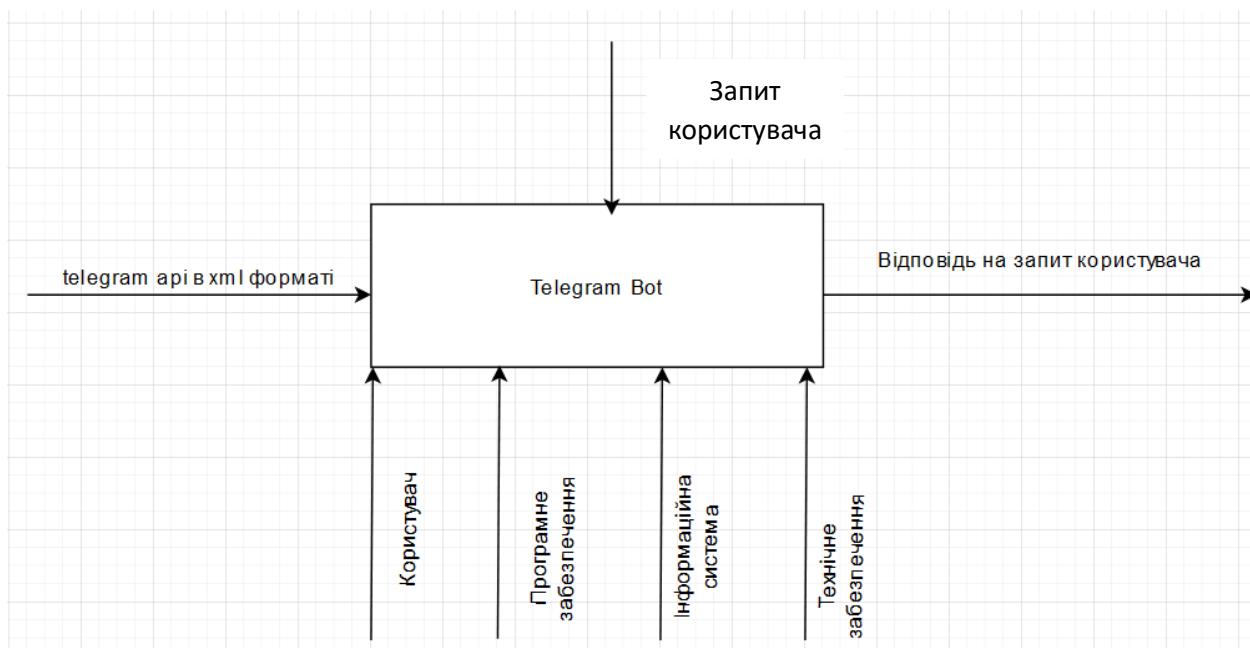


Рисунок 3.1 – Контекста діаграма IDEF0

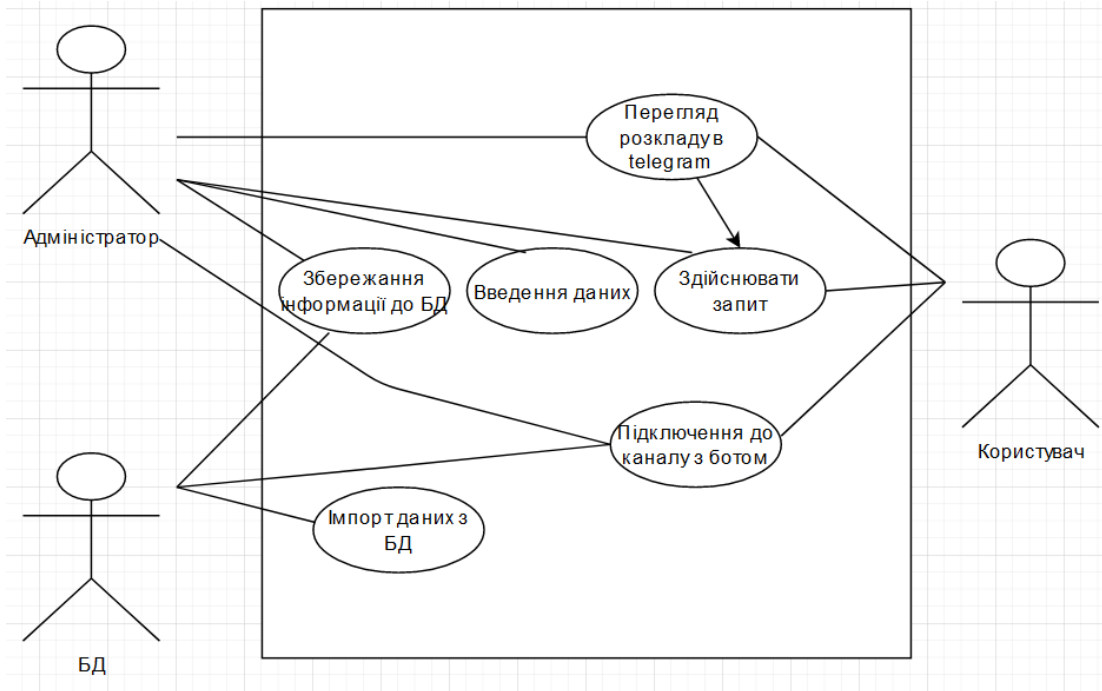


Рисунок 3.2 – Діаграма варіантів використання

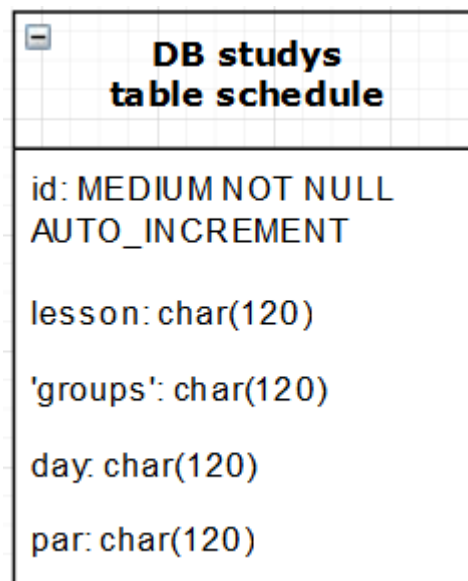


Рисунок 3.3 – Структура бази даних інформаційної системи



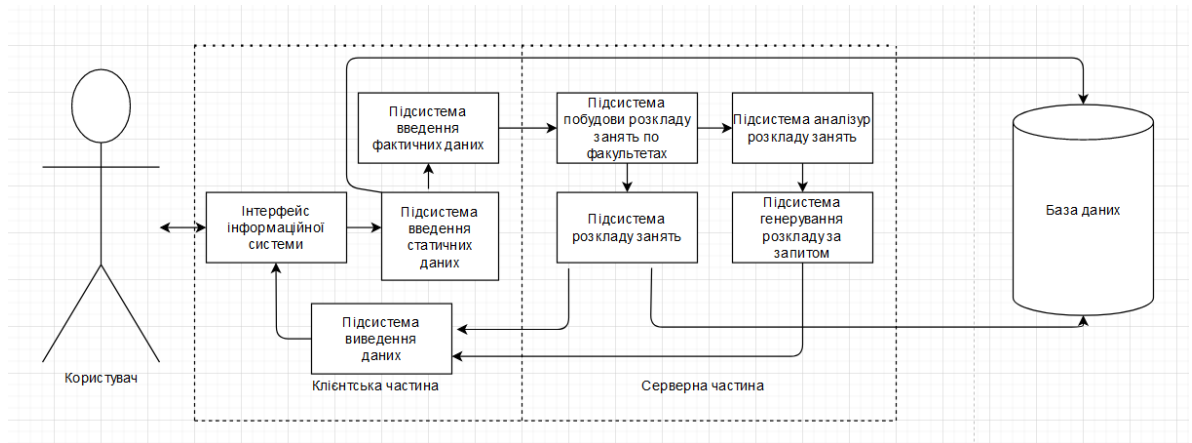


Рисунок 3.4 – Схема архітектури інформаційної системи

### 3.2. Етапи розвитку функціонала бота для підвищення ефективності навчального процесу в ЛНУП

Отже, етапи розвитку функціоналу, для більш точного аналізу розберемо їх детальніше:

- Першим етапом нашої роботи було зробити функціонал, який не буде користувачу нав'язувати непотрібну інформацію, а лише те що йдеться в запитах.

Розбили їх на категорії, а саме реквізити, посилання на інші ресурси університету, розклад та опитування для користувачів. Понад усе користувачів турбувала проблема з отриманням реквізитів, отже ми внесли їх в нашу бд, щоб користувачу не потрібно було шукати їх в інтернеті або ж робити телефонний дзвінок в університет. Також для зручного пошуку приєднання до конференції zoom, було створено зручне посилання на кожного з викладачів. Також для студентів які бажають спостерігати за «життям» університету, ми додали посилання на Instagram та facebook. Розклад залишили так, як було задумано, оскільки це досить зручно для користувача.

- Другим етапом була зміна дизайну

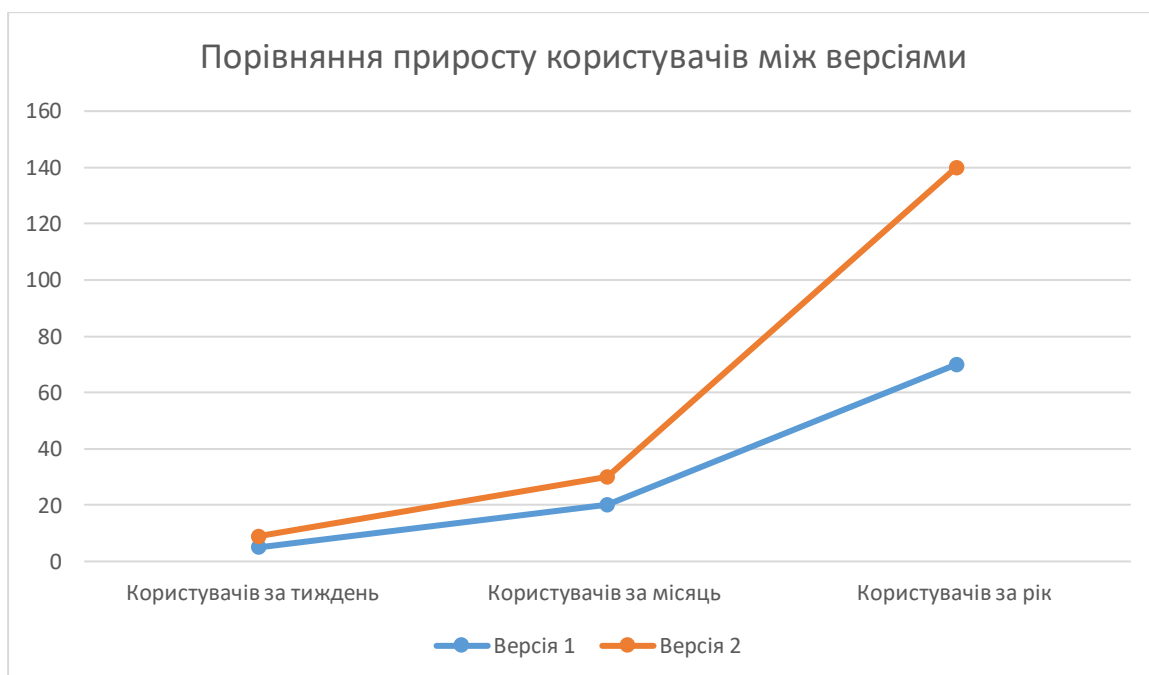
Оскільки виходять постійно нові оновлення telegram з дизайном, нам потрібно було змінити наш, так як був конфлікт між старою версією та новою. Ми переробили дизайн з нуля, добавивши нові розділи в головне меню, для зручності вибору, нове меню під запитом користувача, таким чином змінили перехід з текстового вибору до вибору пунктами. А завдяки тому, що кожен запит є унікальним номером, користувач не отримує зайвої інформації. Так, як тепер в телеграмі є теми, які залежно від свого функціонала могли міняти колір тексту або його вигляд, нам потрібно було додати й цю функцію до бота, так що тепер незалежно від типу теми користувача, проблем зі зміщенням тексту не буде.

- Третім етапом стало створити підтримку для користувача

Оскільки в кожного користувача можуть бути свої запити, які ми не враховували до загальних, ми вирішили що потрібно зробити розділ підтримки, куди б користувачі могли написати. Також, щоб користувачу було не обов'язково давати запит в підтримку, ми створили розділ з опитування, в якому кожен користувач міг написати що він хоче побачити нового в боті або чого йому не вистарчає. В кінці місяця, ми розбираємо усі запити й вирішуємо, що більше користувачу зараз потрібно і по цим даним оновляємо нашого бота.

### **3.3. Аналіз притоку користувачів удосконаленого бота**

Щоб показати як сильно впливає удосконалення бота на користувача, ми побудували діаграму, на якій покажемо, як змінилась динаміка приросту користувачів між першої та другою версією бота(Діаграма 1).



Діаграма 1 – Порівняння приросту користувачів нової та старої версії

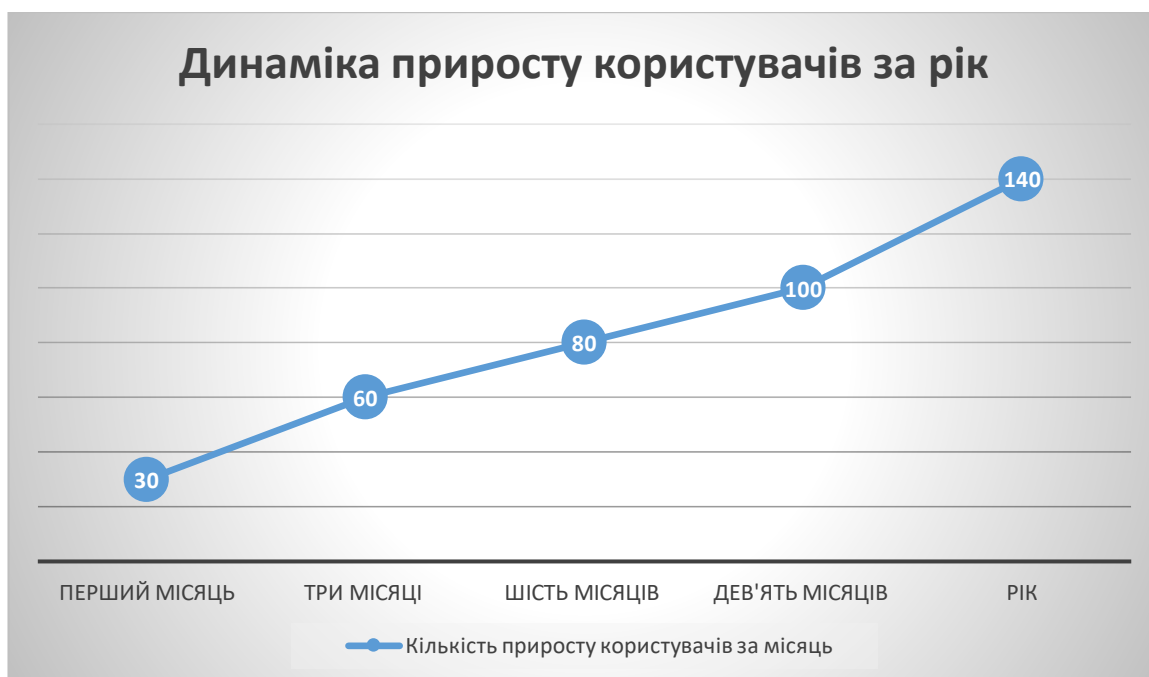
Нами були побудовані діаграми використання бота. Впродовж першого тижня запуску, бота почало використовувати все більше користувачів, за даними які ми отримали у перший день це було всього 2 користувачі, а вже на 7 день це було 9 користувачів, це показує результат зробленої нами роботи і те що користувачам починає подобатись отримувати підтримку у месенджері телеграм. Ми продовжили спостерігати за тим, скільки нових користувачів ми отримаємо за місяць та рік(Діаграма 2, 3, 4). Якщо брати до уваги першу версію бота, то це значні змінити тому, що на початках створення бота, за тиждень ми набрали 4 нових користувача, за тиждень 20 та за рік 70. За даними з діаграм стає зрозуміло, що удосконалення бота користувачам подобається і вони все більше рекомендують нас.



Діаграма 2 – Приріст користувачів впродовж тижня



Діаграма 3 – Приріст користувачів впродовж місяця



Діаграма 4 – Приріст користувачів впродовж місяця

За даними діаграм ми можемо спостерігати не великий, але приріст кількості користувачів, які використовують підтримку в телеграмі. Якщо взяти до уваги те, що розповсюдженням інформації про бота підтримку не було, то ці дані дають нам зрозуміти, що користувачі знайшли його особисто або завдяки знайомим, які ним користувались з перших тижнів запуску. Особливу увагу надає те, що на старій версії бота, такого приросту не було, поки ми не почали його удосконалювати до другої версії.

### **3.4. Результат опитування студентів стосовно функціонала розробленого бота**

Нами проведено анкетування серед користувачів бота, щоб отримати інформацію що б вони хотіли змінити, а що залишити в боті.

Користувач	Що змінити?				
Михайло	Все подобається				
Сергій	Змінити дизайн				
Oleh	Добавте реквізити				
Andrii	Нічого не міняйте				
Олег	.....				
Василь	Все подобається				
Андрій	Добавте нову групу				
Сергій	Внесіть зміни, в розклад у нас викладач помінявся з іншим парами				
Віталік	Добавте іспити в розклад				
Микола	Хай буде як є				
Євген	Норм				
Євген	Хоч так				
Назар	Буду користуватись				
Ярослав	Добавте автоматичне надсилання на кожен день				
Максим	все добре				
Володимир	Все подобається				
Vasyl	нічого не змінюйте				
Олексій	.....				
Роман	поміняйте текст				

Рис 3.5 – Опитування користувачів

Отже, по даній інформації опиту студентів, побудуємо діаграму 4, яка допоможе нам зрозуміти, чи розроблений нами чат-бот повністю задовольняє користувачів: що спочатку слід змінити. Задамо користувачам такі варіанти відповіді на питання анкетування:

1. Нічого не змінювати
2. Змінити функціонал розкладу, посилання на інші ресурси університету, систему підтримки користувача
3. Оптимізувати теми від telegram для коректної роботи бота
4. Додати нові розділи підтримки для користувача
5. Змінити дизайн
6. Підвищити швидкодію та роботу з багатьма користувачами одночасно
7. Оптимізувати інтерфейс під IOS



Діаграма 4 – Результат розподілу

За результатом розділення голосів, більшість користувачів готові використовувати бота таким, яким він є зараз. Та щоб покращити результат підтримки бота, ми вирішили змінити функціонал та додати те чого користувачам не хватало, щоб в наступному опитуванні, кількість користувачів, які бажають змін для бота значно зменшилось або ж не залишилось.

## **4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

### **4.1 Структурно функціональний аналіз процесу розробки інформаційної системи**

Умови в яких перебуває розробник інформаційної системи характеризуються певною травмо- та аварійною небезпекою. Охорона праці безпосередньо на робочому місці – в приміщенні з комп'ютерною технікою та іншим обладнанням, значним чином впливає на показники використання робочого часу, а відтак і на продуктивність розробника в цілому.

Таким чином, створення безпечних умов праці є одним із важливих шляхів підвищення ефективності виконання процесу розробки інформаційних систем.

Для окреслення груп чинників, котрі характеризуються тією особливістю, що зумовлюють виникнення травмонебезпечних та аварійно небезпечних умов розглянемо види технологічних фаз і операцій, що мають місце в процесі розробки інформаційної системи. Технологічні фази: 1) підключення персонального комп'ютера до електромережі; 2) зміна системи охолодження персонального комп'ютера; Технологічні операції: 1) робочий стан ПК на низьких частотах; 2) робочий стан ПК на максимальних частотах;

Можливими травмонебезпечними чинниками є: 1) ураження електричним струмом; 2) випромінювання; 3) перенапруження зору під час роботи з електронними пристроями;

### **4.2. Моделювання процесу виникнення травм та аварій**

Використання методу, що розроблений Д.Хенлі і Х.Кумамото, дає можливість шляхом побудови “дерева” відмов і помилок операторів різних систем вести математичну обробку моделі з метою одержання ймовірності виникнення таких випадкових подій, як аварія, травма, катастрофа [43].



Наведемо основні принципи побудови логічно-імітаційної моделі. Окреслюється досліджувана технологічна операція, під час виконання якої вже були раніше або можуть статися аварії, виробничі травми чи катастрофи. В графічній інтерпретації, модель за своєю формою нагадує крону дерева, тому вона і одержала назву “дерево відмов і помилок”. В свою чергу кінцеві події називають базовими.

Для окреслення та характеризування тієї чи іншої події в побудові логіко-імітаційних моделей застосовують різні символи. Як правило, побудова моделі починається з головної події, а наступні розміщують зверху вниз, аж до базових подій.

Кожен блок рисунку, позначений відповідним номером, означає подію або окремий етап побудови моделі: 1 – відмова (аварія, травма) системи – головна подія; 2 – послідовність подій, що приводять до відмови системи; 3 – послідовність подій зображується за допомогою логічних операторів; 4 – усі вхідні і вихідні події, що входять до моделі, зображуються у вигляді прямокутників з відповідними написами всередині; 5 – послідовний підхід до базових подій, частоти виникнення яких відомі; 6 – базові події зображують у вигляді кружечків з написами всередині, вони є межею аналізу побудованої моделі.

### **4.3. Розробка логічно-імітаційної моделі травм**

Шляхом проведення аналізу, кожний із логічних процесів формування та можливого виникнення травмонебезпечних та аварійних ситуацій, можна виокремити та знайти подію з якої починається небезпечний процес, ще до виникнення небезпечних наслідків.

Методикою оцінки рівня небезпеки робочих місць, машин, виробничих процесів та окремих виробництв передбачено пошук об’єктивного критерію рівня небезпеки для конкретного об’єкта [7]. Таким показником вибрана ймовірність виникнення аварії, травми залежно від явища що досліджується.

Наведемо методику побудови логічно-імітаційної моделі.

За даними виробництва визначаємо ймовірність базових подій. Наприклад, базова подія “стан контролю з охорони праці”. Для визначення ймовірності ми повинні встановити, наскільки (у відсотках) від ідеального рівня здійснюється відповідний контроль на об’єкті. Якщо буде встановлено, що такий рівень контролю становить 50% або 30%, то ймовірність відповідно дорівнює 0,5 і 0,3. При відсутності контролю ймовірність “не здійснення контролю” становитиме 1, якщо контроль ідеальний, то відповідно ймовірність дорівнює 0.

Після обчислення ймовірності всіх подій, розміщених у ромбах, і базових подій, починаючи з лівої нижньої гілки “дерева”, позначаємо номерами всі випадкові події, що увійшли до моделі.

На цьому можна вважати, що певна модель підготовлена до математичних обчислень ймовірностей випадкових подій логічно-імітаційної моделі

Отже, для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми в процесі вирощування культури складемо перелік базових подій. Вони лежатимуть в основі даної моделі. Кожній події (пункту) присвоїмо певне значення ймовірності його виникнення:

Стан контролю з охорони праці .....	$P_1 = 0,25;$
Несерйозне відношення до роботи з ПК .....	$P_2 = 0,05;$
Неправильне облаштування робочого місця .....	$P_3 = 0,1;$
Неправильна робота електроживлення.....	$P_4 = 0,03;$
Застаріла техніка .....	$P_6 = 0,3;$
Перебіг напруги.....	$P_7 = 0,2;$
Досвід роботи з ПК .....	$P_{12} = 0,5.$
Професійний рівень користувача.....	$P_{13} = 0,5;$

Психофізіологічний стан користувача.....  $P_{14} = 0,3$ ;

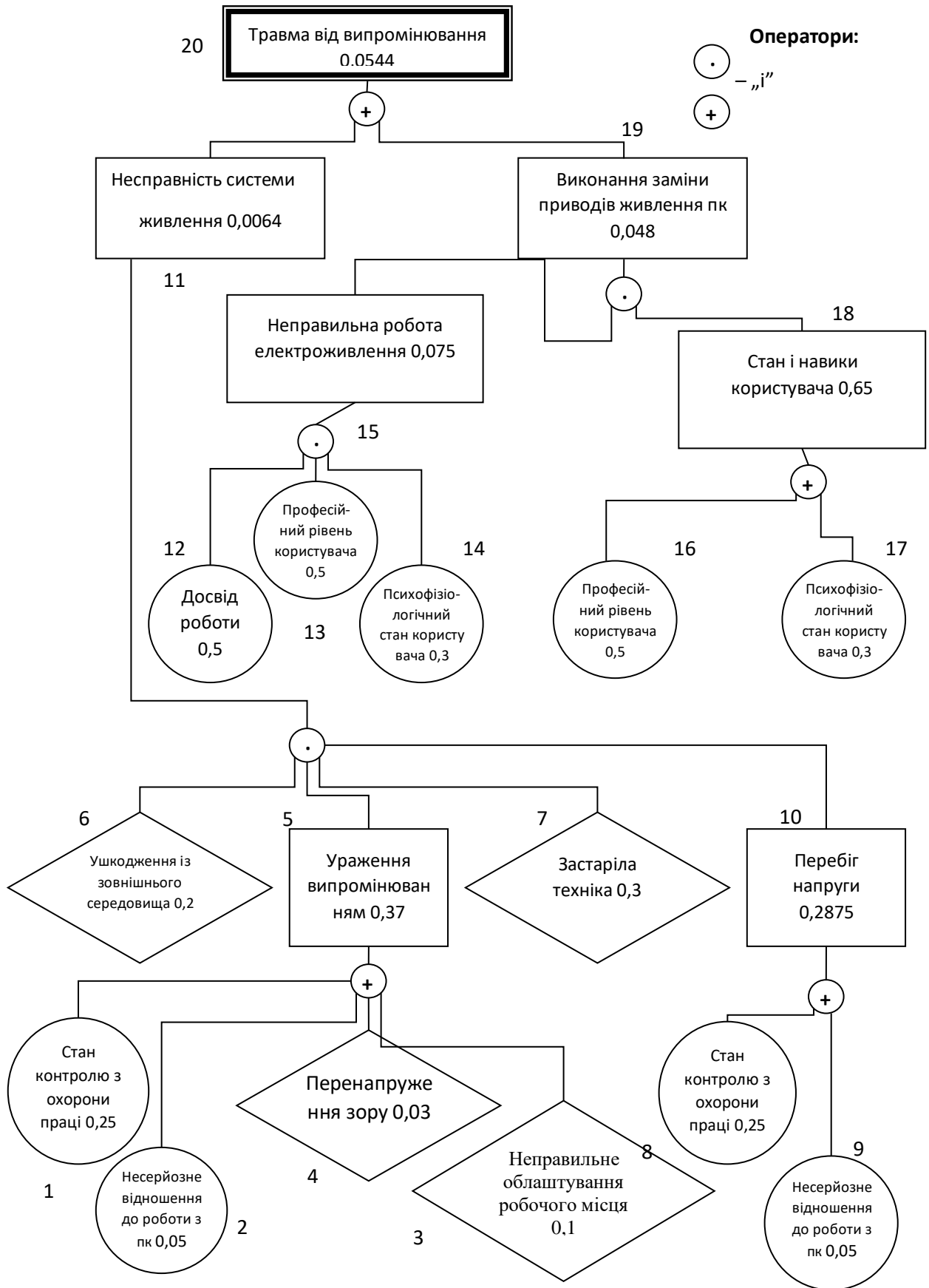


Рис.4.1. - Матриця логічних взаємозв'язків між окремими подіями травмонебезпечної ситуації [7]

На основі наведених подій будемо матрицю логічних взаємозв'язків між окремими пунктами, графічна інтерпретація якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що формують логіко-імітаційну модель технологічного процесу розробки інформаційної системи Розглянемо травмонебезпечну ситуацію, що виникає за умови роботи за несправним персональним компютером. Ймовірність виникнення події  $P_5$  визначаємо наступним чином:

$$P_5 = 0,25 + 0,05 + 0,03 + 0,1 - 0,25 \cdot 0,05 - 0,25 \cdot 0,03 - 0,25 \cdot 0,1 - 0,05 \cdot 0,03 - 0,05 \cdot 0,1 - 0,03 \cdot 0,1 + 0,25 \cdot 0,05 \cdot 0,03 \cdot 0,1 = 0,37$$

Ймовірність виникнення події  $P_{10}$  визначаємо так:

$$P_{10} = 0,25 + 0,05 - 0,25 \cdot 0,05 = 0,2875$$

Ймовірність виникнення події  $P_{11}$  визначаємо:

$$P_{11} = 0,2 \cdot 0,37 \cdot 0,3 \cdot 0,3 = 0,0064$$

Ймовірність виникнення події  $P_{15}$  визначаємо наступним чином:

$$P_{15} = 0,5 \cdot 0,5 \cdot 0,3 = 0,075$$

Ймовірність події  $P_{18}$ :

$$P_{18} = 0,5 + 0,3 - 0,5 \cdot 0,3 = 0,65$$

Ймовірність події  $P_{19}$ :

$$P_{19} = 0,075 \cdot 0,65 = 0,0488$$

Ймовірність події  $P_{20}$ :

$$P_{20} = 0,0064 + 0,0488 = 0,0544$$

Таким чином, ймовірність виникнення травми працівника під час несправності комп'ютера є досить мала і становить  $P_{20} = 0,0544$ .

Використання логіко-імітаційних моделей для дослідження аварій і травм та обґрунтування заходів охорони праці, дають можливість знизити ймовірність виникнення аварійних та травмонебезпечних ситуацій.

#### **4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях**

Актуальність проблеми природно-техногенної безпеки для населення і території, зумовлена зростанням втрат людей, що спричиняється небезпечними природними явищами, промисловими аваріями та катастрофами. Ризик надзвичайних ситуацій природного та техногенного характеру невпинно зростає, тому питання захисту цивільного населення від надзвичайних ситуацій на сьогодні є дуже важливе.

Заходи щодо зниження ступеня впливу негативних наслідків аварійних ситуацій здійснюються з метою завчасної підготовки підприємств від надзвичайних ситуацій, та створення умов для підвищення стійкості їх роботи, та проведення своєчасних робіт щодо рятувальних заходів [40].

Відповідальність за організацію цивільної оборони згідно із Законом “Про цивільну оборону України” лягає на керівника підприємства. Керівництво підприємств повинно забезпечити працівників засобами захисту (індивідуального та колективного), створює загони для ліквідації наслідків надзвичайних ситуацій.

У системі цивільного захисту окремого господарства необхідно забезпечити захист населення таким чином:

- можливість укриття населення у захисних спорудах;
- використання засобів індивідуального і медичного захисту;
- будівництво захисних споруд, насадження лісосмуг.

Укриття в захисних спорудах, якому підлягає усе населення відповідно до приналежності, досягається за рахунок створення фонду захисних споруд.

Евакуаційні заходи, які проводяться в містах та інших населених пунктах, які мають об'єкти підвищеної небезпеки, а також у воєнний час, основним способом захисту населення є евакуація і розміщення його у позаміській зоні.

Медичний захист проводиться для зменшення ступеня ураження людей, своєчасного надання допомоги постраждалим та їх лікування, забезпечення епідеміологічного благополуччя в районах надзвичайних ситуацій.

Радіаційний і хімічний захист включає заходи щодо виявлення і оцінки радіаційної та хімічної обстановки, організацію і здійснення дозиметричного та хімічного контролю, розроблення типових режимів радіаційного захисту, забезпечення засобами індивідуального захисту, організацію і проведення спеціальної обробки.

Евакуаційні заходи, які проводяться в містах та інших населених пунктах, які мають об'єкти підвищеної небезпеки, а також у воєнний час, основним способом захисту населення є евакуація і розміщення його у позаміській зоні [40].

## ВИСНОВКИ

1. Як вказують результати досліджень, аналіз умов, обставин та причин різних аварій, травм при роботі з ПК показує, що процеси формування та виникнення цих явищ можна заздалегідь моделювати, застосовуючи метод побудови “дерева відмов” та помилок оператора людино-машинних систем у роботі з персональним комп'ютером.

2. Розглянули травмонебезпечну ситуацію, що виникає за умови роботи з інформаційною системою на персональному комп'ютері приєднаного до не стабільної електромережі, що може призвести до удару струмом і встановили, що

ймовірність виникнення травми виконавця під час роботи з не стабільною системою електроживлення комп'ютера є досить мала і становить  $P_{20} = 0,0544$ .

3. Ризик надзвичайних ситуацій природного та техногенного характеру невпинно зростає, тому питання захисту цивільного населення від надзвичайних ситуацій на сьогодні є дуже важливе.

## 5. ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ

### 5.1. Скільки часу пішло на створення бота

Отже, часу на створення чистого бота пішло 2 тижні, для добавлення нового функціоналу ще 2 тижні, додавання його на веб-хостинг для автоматичної роботи ще один тиждень. Отже, в загальному для створення потрібно було всього 5 тижнів, але щоб бот продовжував функціонувати потрібно постійно підтримувати його роботу на вебхостингу, через те, що нам надали тимчасову автономну роботу, також існує дубляж бота у який ми додаємо нові функції й тестуємо як вони будуть впливати на бота, їх плюси та мінуси. Нами була побудована діаграма Ганта для кращого сприйняття використання часу, яке пішло нам на створення бота.

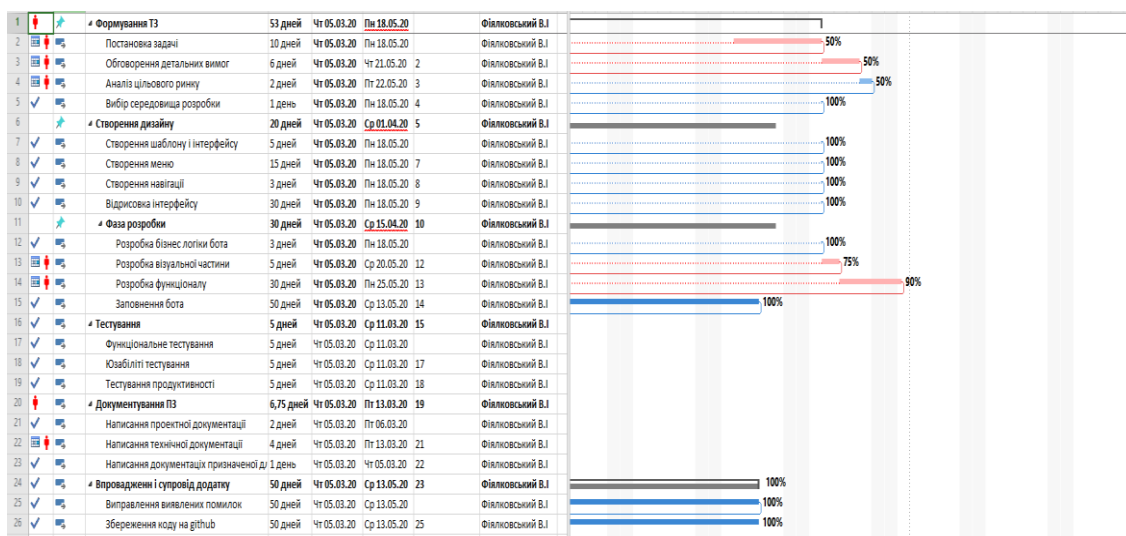


Рис 5.1 – Діаграма Ганта

### 5.2. Розрахунок трудомісткості розроблення та використання

Розраховуючи трудомісткість розроблення бота, проблем як таких не було, телеграм надав ключі без жодних проблем і з розробленням бота нічого критичного не було, але на стадії питання як зробити, щоб він почав працювати без нашого втручання виникла чи мало кількість проблем. Ми вирішили використати вебхостинги для цього. Новою проблемою стало що для його повноцінної роботи



потрібно заплатити, а якщо використати безплатну версію що нам дають, то ми отримуємо обмеження в використанні бд та часі роботи бота.

За нашими розрахунками, той функціонал що надавав вебхостинг нам би вистачило, та час роботи давав про себе знати, якщо до бота не було ніяких запитів, веб-хостинг автоматично переводив його в сонний режим, а коли до нього звертались то потрібно було дочекатись його «пробудження», яке тривало 1-2 години від так званого «сну», щоб дати відповідь користувачу.

### **5.3. Вартість розробленого продукту**

Точну вартість розробленого продукту ми розраховали завдяки інтернету, де на сайтах продавались чат-боти в телеграмі, орієнтована ціна такого бота 1000 грн, але це за уваги якщо використовувати безплатний хостинг, а ціна платного використання 5\$ за 1000 годин автономної роботи бота та збільшеною кількістю функціонала.

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Ми розглянули аналіз ринку на аналоги чат-бота і побудували таблицю порівнянь їх переваг та недоліків, після усіх аналізів було вирішено, що потрібно додати в чат-бота та які системи він повинен виконувати, результатом роботи став побудований план взаємодії користувача з ботом.

Після завершення роботи з аналогами ми почали формувати функціональні вимоги для створення бота, дане завдання потрібне для того, щоб виявити який функціонал ми внесемо в бота підтримку та що він буде виконувати, результатом став побудований функціонал передачі інформаційної підтримки від бота до користувача.

Результат вирішеної задачі, яка була побудована нами, містить діаграми нотації та варіанти використання.

При розробці чат-бота, ми вирішили додати анкетування для користувачів, які б змогли вносити свої ідеї щодо зміни бота, ми побудували графік який показує яка кількість користувачів користувалась ним впродовж першого тижня його створення.

Ми обрали мову програмування Java для розробки чат-бота за її унікальність, інструмент Maven для покращення роботи з ботом, бібліотеку Spring, яка своєю чергою підтримує наш додаток, базу даних PostgreSQL та віддалений сервер Heroku, результатом виконаної роботи являється бот котрий працює в автономному режимі без втручання адміністратора та має в собі базу даних з інформаційною підтримкою для користувача.

Під час завершення роботи ми провели тест на працездатність бота, результатом стала система, яка працює без збоїв і може працювати з будь-яким користувачем.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Студент ЧДТУ створив бот у «Телеграмі» для зручного доступу до розкладу занять. URL: <https://chdtu.edu.ua/news/item/14609-student-chdtu-stvoryv-bot-u-telehrami-dlia-zruchnoho-dostupu-do-rozkladu-zaniat> (дата звернення 20.03.2021)
2. Розклад занять завжди в телефоні: університет запустив на Telegram чат-бота для студентів URL: <https://nupp.edu.ua/news/rozklad-zanyat-zavzhdi-v-telefoni-universitet-zapustiv-na-telegram-chat-bot-dlya-studentiv.html> (дата звернення 20.03.2021)
3. Розробка чат-бота розкладу занять для месенджера Telegram. URL: <https://cutt.ly/snETQtK> (дата звернення 20.04.2021)
4. SQL-Урок 1. Мова SQL. Загальні поняття URL: <https://cutt.ly/tnETvMy> (дата звернення 18.05.2021)
5. Введення в MS SQL Server і T-SQL URL: <https://metanit.com/sql/sqlserver/1.1.php> (дата звернення 18.05.2021)
6. Що таке Spring Framework URL: <https://metanit.com/sql/sqlserver/1.1.php> (дата звернення 18.05.2021)
7. Князевського Б. А., Долина П. А. Охорона праці: Підручник для студентів вузів, 2003.
8. Белов С. В., Ільніцкая А. В., Козьяков А. Ф. Безпека життєдіяльності. 2005.
9. Герберт Шилдт. Java 8 Полное руководство. Девятое издание. 2017, 1376 с.
10. Сьерра Кэти, Бейтс Берт. Изучаем Java. 2016. 718 с.
11. Козлов А. А., Батищев А. В. Телеграм-бот как простой и удобный способ получения информации. Текст научной статьи по специальности «Компьютерные и информационные науки». 2015. №16. 10 с.
12. Козмина Юлиана, Харроп Роб, Шефер Крис, Кларенс Хо. Spring 5 для профессионалов. 2016. 1120 с.

13. Бьюли Алан. Изучаем SQL. 2007. 308 с.
14. Шаббир Ч., Джадип Л., Чинтан М., Кандарп П. MySQL 8 для больших данных. 2018. 226 с.
15. Джош Л., Кеннет Б. Java в облаке. Spring Boot, Spring Cloud, Cloud Foundry. 2019. 624 с.
16. Роберт М. Чистый код. Создание, анализ и рефакторинг. 2019. 464 с.
17. Джошуа Б. Java. Эффективное программирование. Второе издание. 2016. 440 с.
18. Срини Д. Разработка чат-ботов и разговорных интерфейсов. 2016. 340 с.
19. Макконнелл С. Совершенный код. Практическое руководство по разработке программного обеспечения. 2016. 870 с.
20. Томас К., Каролин Б. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 2017. 1440 с.
21. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинг. 544 с.
22. Способы хранения деревьев в реляционных базах данных с использованием ORM Hibernate. URL: <https://habr.com/ru/hub/java/> (дата звернения 20.05.2021)
23. Руководство по возможностям Java версий 8-16. URL: <https://habr.com/ru/post/551590/> (дата звернения 20.05.2021)
24. Spring Boot: быстрое знакомство и старт на примере простого веб-приложения. URL: <https://tproger.ru/articles/spring-boot-bystro-znakomstvo-i-start-na-primere-prostogo-veb-prilozhenija/> (дата звернения 21.05.2021)
25. Обзор модулей Spring для Java. URL: <https://tproger.ru/articles/spring-modules-overview/> (дата звернения 21.05.2021)
26. Записки программиста. Использование Foreign-Data Wrappers в PostgreSQL. URL: <https://eax.me/postgresql-fdw/> (дата звернения 22.05.2021)
27. Roman E. A tribute to Java. URL: <https://elizarov.medium.com/a-tribute-to-java-c184cd97db8f> (дата звернения 25.05.2021)

28. Rygel Louv. Lifecycle. URL: <https://proandroiddev.com/understanding-gradle-the-build-lifecycle-5118c1da613f> (дата звернення 26.05.2021)
29. Smith Tom. DZone Research: The Problems With Java. URL: <https://dzone.com/articles/problems-with-java> (дата звернення 26.05.2021)
30. Язык программирование Java. URL: <https://web-creator.ru/articles/java> (дата звернення 26.05.2021)

## **ДОДАТКИ**

## ДОДАТОК А

### 1. Планування робіт

#### А.1. Деталізація мети методом SMART

Технологія SMART – це основний підхід для роботи з конкретними цілями. Кожна буква даної технології несе в собі мету для виконання цілі, а саме:

S – Specific

M – Measurable

A – Achievable

R – Realistic

T - Timed

S – Інформаційна система розкладу занять в месенджері Telegram

M – Результатом завдання виходить чат бот з розкладом занять для студентів

A – Створення чат-бота з розкладом занять буде здійснено завдяки Spring Framework, Maven, PostgreSQL, Heroku, Java.

R – під час виконання роботи доступні усі технічні інструменти та засоби взаємодії з програмою

T – Завдання має обмеження по часу і повинне бути виконане вчасно

Після проведеного аналізу методом СМАРТ ми отримуємо кінцеву мету, а саме створення чат-бота з розкладом занять для студентів. Проєкт буде завершений вчасно.

#### А.2. Планування змісту структури робіт ІТ-проєкту(WBS)

Виробнича структура – це розбиття головного завдання на декілька окремих цілей, які по ході виконання роботи будуть завершені, щоб у кінцевому результаті

об'єднати їх у готовий продукт. Загальне поняття WBS походить від розбивання завдання на декілька цілей проєкту, кожен етап проходить поступове виконання роботи та наступним етапом загальний обсяг роботи займає менше часу його виконання, що показна на рисунку 5.1

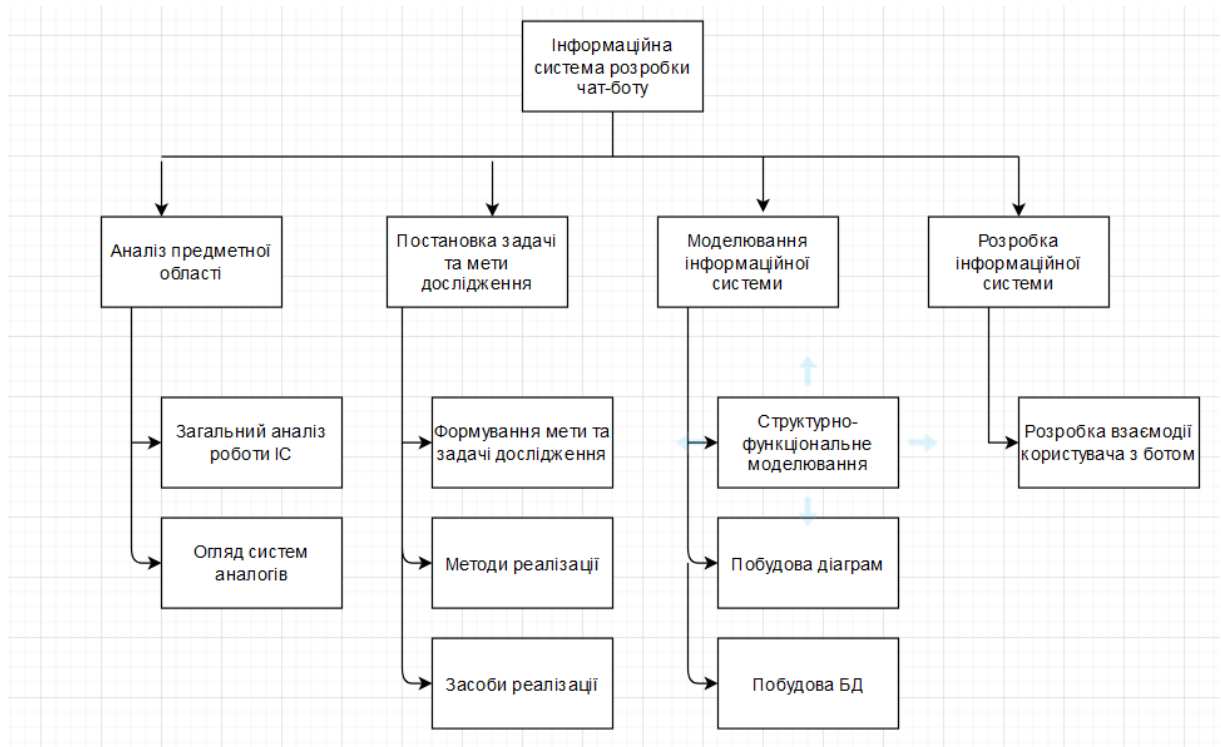


Рисунок А.1 – WBS діаграма проєкту

### А.3. Планування структури організації робіт ІТ-проєкту(OBS)

Суть даної структури полягає у внутрішній взаємодії проєкту. OBS є аналогом WBS і буду діаграми є ідентична. На верхніх рівнях завдання створений елемент над, над яким буду проводитись робота, проте відмінність даної робочої структури полягає у розбивання частини завдань на рівні групи, які своєю чергою виконують меншу кількість робіт, тому роботи меншого рівня WBS є ті ж, що і в OBS.



У розроблені telegram бота для інформаційної підтримки навчального процесу у Львівському національному університеті природокористування із вибором ефективних засобів брав участь один розробник та один керівник.

Графічна структура OBS виглядає як показано на рисунку 5.2

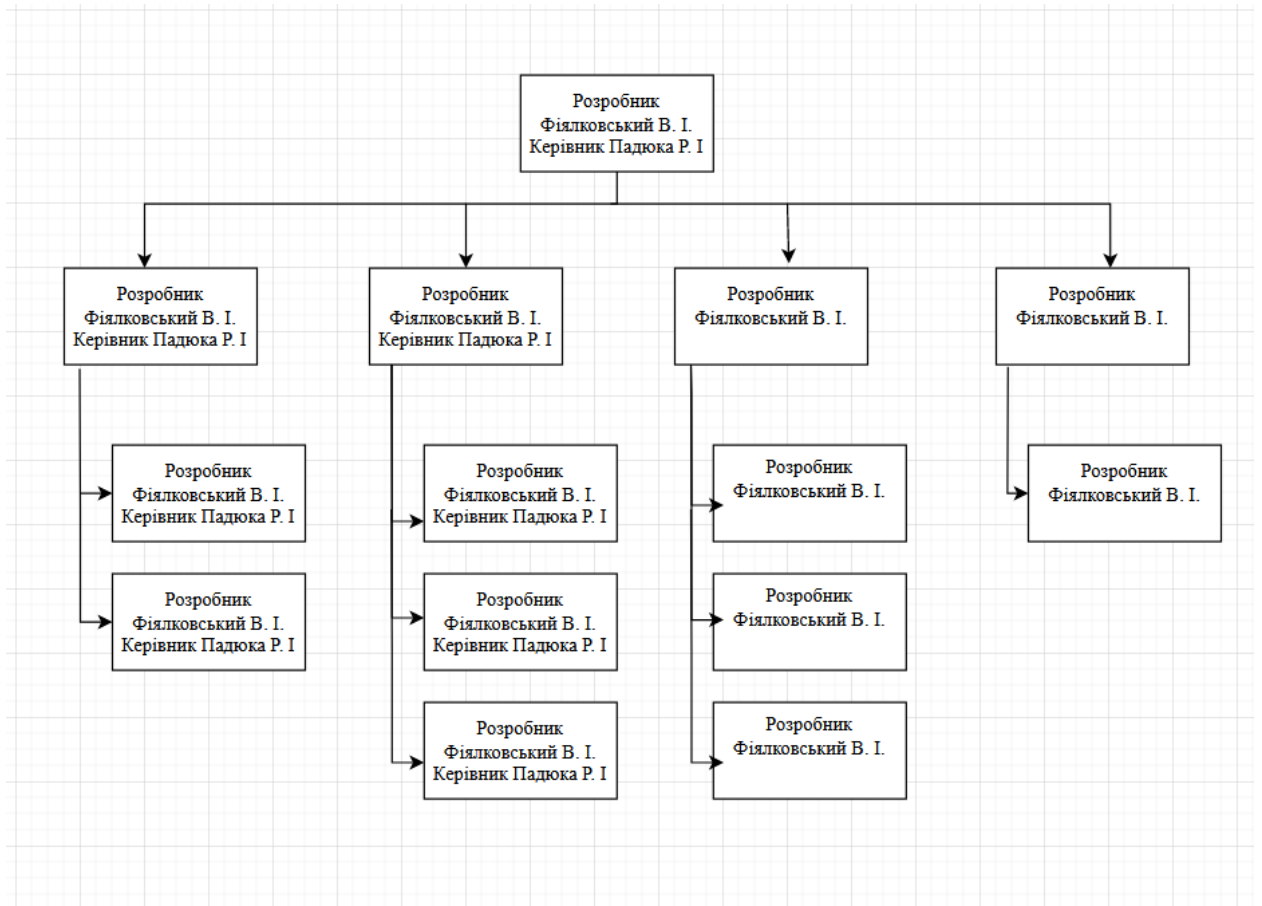


Рисунок А.2 – OBS діаграма проєкту

#### А.4. Планування ризиків ІТ-проєкту

Реалізація проєкту відбувається в умовах зміни інформаційного плану в університеті, що означає зміну функціоналу використання бота.

Що ж ми можемо розуміти під ризиком? Це вплив фактору на завдання проєкту що може призвести до зміни завершення роботи по часі.

В умовах задачі слід контролювати зміну занять у різних спеціальностей, щоб інформація котра буде доставлятись студентам була коректною і не мала помилок. При розробці різних проєктів потрібно слідкувати за ризиками та уміти обійти дану проблему.

Ризики, які можуть стати на перешкоді розробки проєкту:

- Поява того ж продукту
- Обмеження в часі
- Помилка в коді
- Помилка в рекомендації
- Недостатньо умінь для реалізації поставленої задачі
- Збій апаратного та програмного забезпечення

Постановка правильних цілей та моніторинг ризиків дозволить обійтись без проблем при побудові проєкту, а саме розробка інформаційної підтримки навчального процесу у Львівському НУП завдяки telegram боту.

## ДОДАТОК Б

### Б.1. Код розробки чат-бота у програмному середовищі IntelliJ IDEA

```

public class Bot extends TelegramLongPollingBot {

    public static void main (String[]args){
        ApiContextInitializer.init();
        TelegramBotsApi telegramBotsApi = new TelegramBotsApi();
        try {
            telegramBotsApi.registerBot(new Bot());

        } catch (TelegramApiRequestException e) {
            e.printStackTrace();
        }
    }

    public void sendMsg (Message message, String text){
        SendMessage sendMessage = new SendMessage();
        sendMessage.enableMarkdown(true);
        sendMessage.setChatId(message.getChatId().toString());
        sendMessage.setReplyToMessageId(message.getMessageId());
        sendMessage.setText(text);
        try {
            setButtons(sendMessage);
            sendMessage(sendMessage);
        } catch (TelegramApiException e) {
            e.printStackTrace();
        }
    }

    public void onUpdateReceived (Update update){
        Message message = update.getMessage();

    public void setButtons (SendMessage sendMessage){
        ReplyKeyboardMarkup replyKeyboardMarkup = new ReplyKeyboardMarkup();
        sendMessage.setReplyMarkup(replyKeyboardMarkup);
        replyKeyboardMarkup.setSelective(true);
        replyKeyboardMarkup.setResizeKeyboard(true);
        replyKeyboardMarkup.setOneTimeKeyboard(false);

        List<KeyboardRow> keyboardRowList = new ArrayList<>();
        KeyboardRow keyboardFirst = new KeyboardRow();
    }
}

```

```

keyboardFirst.add(new KeyboardButton("/help"));
keyboardFirst.add(new KeyboardButton("/start"));
keyboardFirst.add(new KeyboardButton("Підтримка"));
keyboardFirst.add(new KeyboardButton("Реквізити"));
keyboardFirst.add(new KeyboardButton("Викладачі"));
keyboardFirst.add(new KeyboardButton("Опитування"));

keyboardRowList.add(keyboardFirst);
replyKeyboardMarkup.setKeyboard(keyboardRowList);
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

```

```

public class DataBase {
    public static void main(String[] args) throws ClassNotFoundException,
SQLException {
        String url =
"jdbc:mysql://localhost:3306/studys?useUnicode=true&useJDBCCompliantTimezoneS
hift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";
        Class.forName("com.mysql.jdbc.Driver");
        try (Connection conn = DriverManager.getConnection(url, username, password);
            Statement stat = conn.createStatement()) {
            stat.executeUpdate("drop table schedule");
            stat.executeUpdate("CREATE TABLE IF NOT EXISTS schedule(id
MEDIUMINT NOT NULL AUTO_INCREMENT, lesson char(120),`groups` char(120)
,day char(120),par char(120),PRIMARY KEY (id))");

```

```

server.port=5432
localeTag=ua-UA

```

```

pingtask.ulr= https://www.google.com
pingtask.period= 1200000

```

```

telegrambot.botUserName=@StudyLnauBot

```

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"

```

```

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>TelegramBot</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>1.8</java.version>

  </properties>
  <dependencies>
    <dependency>
      <groupId>org.telegram</groupId>
      <artifactId>telegrambots</artifactId>
      <version>3.6</version>
    </dependency>
    <dependency>
      <groupId>org.apache.poi</groupId>
      <artifactId>poi</artifactId>
      <version>4.1.2</version>
    </dependency>

    <dependency>
      <groupId>org.apache.poi</groupId>
      <artifactId>poi-scratchpad</artifactId>
      <version>4.1.2</version>
    </dependency>

    <dependency>
      <groupId>org.apache.poi</groupId>
      <artifactId>poi-ooxml</artifactId>
      <version>4.1.2</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>

```

```

<version>3.1.2</version>
<configuration>
  <archive>
    <manifest>
      <addClasspath>>true</addClasspath>
      <classpathPrefix>target/</classpathPrefix>
      <mainClass>TelegramBot-1.0-SNAPSHOT</mainClass>
    </manifest>
  </archive>
</configuration>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>appassembler-maven-plugin</artifactId>
  <version>1.1.1</version>
  <configuration>
    <assembleDirectory>target</assembleDirectory>
    <programs>
      <program>
        <mainClass>com.home.server.TelegramBot</mainClass>
        <name> TelegramBot </name>
      </program>
    </programs>
  </configuration>
  <executions>
    <execution>
      <phase>package</phase>
      <goals>
        <goal>assemble</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
  </configuration>
</plugin>
</plugins>
</build>
</project>

```