

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри )

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНФОРМУВАННЯ  
ПРО СТАН ПОГОДИ  
WEATHER FORECASTING SOFTWARE

Виконав: студент(ка) 4 курсу, групи КНТ-218

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Сарана О.О.

(прізвище та ініціали)

Керівник Олійник А.О.

(прізвище та ініціали)

Рецензент Гофман Є.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»  
(повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ  
Кафедра програмних засобів  
Ступінь вищої освіти бакалавр  
Спеціальність 122 Комп'ютерні науки  
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
С.О. Субботін  
“      ”                      2022 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

Сарани Олега Олеговича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для інформування про стан погоди. Weather Forecasting Software

керівник проєкту (роботи) Олійник Андрій Олександрович, д.т.н., професор,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | Прізвище, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | Олійник А.О., професор                    |                |                           |
| Нормоконтроль       | Липовець М.В., асистент                   |                |                           |

7. Дата видачі завдання “04” квітня 2022 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|----------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                               | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                               | Розділ 2     |
| 4     | Розробка архітектури програми.                                       | 2 тиждень                               | Розділ 3     |
| 5     | Розробка програми.                                                   | 3-4 тижні                               | Розділи 3, 4 |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                               | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                               | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                               |              |
| 9     | Захист роботи.                                                       | 8 тиждень                               |              |

**Студент(ка)**

\_\_\_\_\_  
( підпис )      Сарана О.О.  
(прізвище та ініціали)

**Керівник проєкту (роботи)**

\_\_\_\_\_  
( підпис )      Олійник А.О.  
(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:  
88 с., 36 рис., 5 табл., 4 дод., 21 джерел.

JAVASCRIPT, MONGODB, БАЗА ДАНИХ, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГНОЗ ПОГОДИ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ.

Об'єкт дослідження – процес розробки програмного забезпечення для подання інформації користувачам про стан погоди.

Предмет дослідження – програмне забезпечення для інформування користувачів про стан погоди.

Метою роботи є розробка програмного забезпечення для інформування користувачів про стан погоди.

Матеріали, методи та технічні засоби: мова програмування JavaScript, інтегроване середовище розробки Atom, система управління базами даних MongoDB.

Результати. Розроблено програмне забезпечення для надсилання користувачам прогнозу погоди у месенджері Telegram за допомогою бота.

Практична цінність роботи полягає у розробці програмного забезпечення для інформування користувачів про стан погоди.

Висновки. Розроблено програмне забезпечення для інформування користувачів про стан погоди.

Галузь використання – прогнозування погоди.

## ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 36 figures, 5 tables, 4 appendixes, 21 sources.

JAVASCRIPT, MONGODB, DATABASE, PROGRAMMING LANGUAGE, SOFTWARE, WEATHER FORECAST, DATABASE MANAGEMENT SYSTEM.

The object of research is the process of software development for provide users with information about the weather.

The subject of the research is software for informing users about the weather.

The purpose of this work is to develop software to inform users about the weather.

Materials, methods and technical tools: JavaScript programming language, Atom integrated development environment, Mongodb database management system.

Results. The software tools for sending weather forecasts to users in the Telegram messenger using a bot have been developed.

The practical value of the work lies in the development of software to inform users about the weather.

Conclusions. The software informing users about the weather is developed.

Scope of use – weather forecasting.

## ЗМІСТ

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
|                                                                                      | С. |
| Перелік скорочень та умовних позначок .....                                          | 8  |
| Вступ .....                                                                          | 9  |
| 1 Аналіз предметної області .....                                                    | 11 |
| 1.1 Поняття про проблему прогнозування погоди .....                                  | 11 |
| 1.2 Дослідження програмних засобів для інформування про стан погоди... 17            | 17 |
| 1.3 Висновки за розділом 1 .....                                                     | 24 |
| 2 Матеріали і методи .....                                                           | 25 |
| 2.1 Вибір мови програмування та інших технологій розробки .....                      | 25 |
| 2.2 Вибір середовища розробки для розробки програмного забезпечення.. 28             | 28 |
| 2.3 Вибір системи керування базою даних .....                                        | 30 |
| 2.4 Висновки за розділом 2 .....                                                     | 33 |
| 3 Опис програми .....                                                                | 35 |
| 3.1 Структура програмного забезпечення для інформування про стан погоди .....        | 35 |
| 3.2 Функціонування програмного забезпечення для інформування про стан погоди..... 37 | 37 |
| 3.3 Опис бази даних .....                                                            | 39 |
| 3.4 Взаємодія користувача з програмою інформування про стан погоди .... 40           | 40 |
| 3.5 Опис серверної частини .....                                                     | 41 |
| 3.6 Висновки за розділом 3 .....                                                     | 43 |
| 4 Експлуатація, тестування та експериментальне дослідження програми.....45           | 45 |
| 4.1 Призначення й умови застосування програми .....                                  | 45 |
| 4.2 Характеристики програми .....                                                    | 46 |
| 4.3 Інструкція по експлуатації програми..... 46                                      | 46 |
| 4.3.1 Звернення до програми..... 46                                                  | 46 |
| 4.3.2 Вхідні й вихідні дані .....                                                    | 46 |
| 4.3.3 Повідомлення..... 47                                                           | 47 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 4.4 Тестування та результати роботи програмного забезпечення ..... |    |
| для інформування про стан погоди .....                             | 47 |
| 4.5 Висновки за розділом 4 .....                                   | 50 |
| Висновки .....                                                     | 52 |
| Перелік джерел посилання .....                                     | 55 |
| Додаток А Технічне завдання .....                                  | 57 |
| Додаток Б Опис програми .....                                      | 61 |
| Додаток В Текст програми .....                                     | 65 |
| Додаток Д Слайди презентації .....                                 | 80 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

БД – база даних;

ПЗ – програмне забезпечення;

ПП – програмний продукт;

ОП – оперативна пам'ять;

ОС – операційна система;

СКБД – система керування базами даних.

## ВСТУП

Сьогодні важливу роль відіграють сервіси прогнозування погоди. Майже кожен, хто підключається до Інтернету, щодня перевіряє погоду, щоб уникнути непередбачуваних ситуацій [1]–[6]. Прогнози погоди стають все точнішими, а сайти, які надають відповідну інформацію, оновлюються майже щогодини. Люди хочуть відчувати зміни і мати можливість планувати свої дії відповідно до ситуації. Прогнозування погоди як онлайн-засіб надає можливість користувачам вчасно отримувати відповідну інформацію [1]–[6].

Крім того, останнім часом все більшої популярності набувають соціальні мережі, а також різні месенджери, зокрема Telegram, Viber, Facebook та ін. Telegram представляє собою багатоплатформну програму для обміну повідомленнями, яка дозволяє обмінюватися повідомленнями та медіа-файлами в багатьох форматах та є досить популярною у наш час [1]–[8]. За допомогою спеціального API сторонні розробники можуть створити спеціальний обліковий запис, який може керувати ботом прогнозування погоди у відповідному месенджері. Звичайні боти реагують на певні команди в приватних і групових чатах і можуть шукати в Інтернеті або виконувати інші завдання, які використовуються в розважальних або ділових цілях.

Тому актуальним є завдання розроблення програмного забезпечення у вигляді Telegram-боту для інформування про стан погоди.

Таким чином, метою роботи є розробка програмного забезпечення для інформування користувачів про стан погоди.

Для досягнення мети роботи потрібно розв'язати такі завдання:

- здійснити аналіз предметної області та програмних засобів для інформування про стан погоди;
- виконати проектування програмного забезпечення у вигляді Telegram-боту для інформування про стан погоди;
- розробити програмне забезпечення для інформування про стан погоди;

– здійснити тестування розробленого програмного забезпечення для інформування про стан погоди.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Поняття про проблему прогнозування погоди**

Прогнозування погоди здійснюється на основі застосування принципів фізики, підкріплених різноманітними статистичними та емпіричними методами. Крім безпосередньо прогнозування атмосферних подій, прогноз погоди включає прогнози змін поверхні Землі, викликаних атмосферними умовами (наприклад, сніговим і льодовим покривом, штормовими припливами, повенями). Вимірювання та ідеї як основа для прогнозування погоди є досить важливими та впливають на багатьох людей. У наш час, прогнозування погоди виконується в основному, шляхом спостережень і прогнозування відповідних змін. Сучасні інструменти для вимірювання температури, тиску, вітру та вологості, навіть із застосуванням найскладніших методів чисельного прогнозування та з використанням суперкомп'ютерів, все одно вимагає серії вимірювань стану атмосфери – поточної картини температури, вітру та інших фундаментальних елементів [1]–[6].

Оскільки спостереження є надзвичайно важливими для прогнозування погоди, метеорологічні вимірювання та опис прогнозу погоди – це поєднання історичних даних про погоду з методами та технологіями чисельного аналізу та прогнозування. Проте наявність помилок у результатах прогнозування погоди вказує на необхідність розробки нових, кращих методів вимірювань та створення нових методів та програмних засобів прогнозування та інформування щодо стану погоди [1]–[6].

З середини 20 століття цифрові комп'ютери дозволили обчислювати зміни атмосферних умов математично й об'єктивно, тобто таким чином, що кожен може отримати однаковий результат з однакових початкових умов. Широке поширення цифрових моделей прогнозування погоди призвело до появи нових професій, зокрема комп'ютерних експертів, експертів з обчислень і статистики, метеорологів робота яких пов'язана з аналізом атмосферних явищ. Крім того, розширена здатність обробляти та аналізувати дані про

погоду підштовхнула давню зацікавленість метеорологів у забезпеченні більшої кількості спостережень з більшою точністю. Технологічний прогрес призвів до все більшої залежності від дистанційного зондування, зокрема збору даних за допомогою спеціально обладнаних супутників на орбіті Землі. Прогноз погоди в значній мірі покладається на використання чисельних моделей, інтегрованих з високошвидкісними суперкомп'ютерами, за винятком деяких короткочасних прогнозів, зокрема тих, що стосуються місцевої грозової активності, які виконуються безпосередньо експертами з радіолокації [1]–[6].

Останнім часом потужність обчислювальних машин суттєво зросла, що дозволило метеорологічним бюро генерувати більш складні ансамблі прогнозів, тобто набори кількох моделей, результати яких обмежують діапазон невизначеності одним прогнозом [1]–[6].

Раніше прогнозуванням користувалися переважно ті, хто займався сільським господарством. Якщо можна передбачити довгострокові погодні умови, очевидно, що посадку та збирання врожаю можна краще спланувати та проводити більш ефективно. На початку двадцятого століття прогнозування погоди стало важливим інструментом для авіації. Під час Другої світової війни відкриття на великих висотах дуже сильних вітрових течій (реактивних течій, які можуть впливати на швидкість літаків) і загальна чутливість до погодних умов військових дій у Європі викликали особливий інтерес до прогнозування погоди. Однією з найвідоміших проблем прогнозування воєнного часу була операція «Оверлорд», вторгнення союзних сил на материкову Європу в Нормандії [1]–[6].

Наприкінці прогнозування погоди набуло безпрецедентного зростання для комерційних компаній. Маркетингові організації та магазини наймають консультантів з прогнозу погоди, щоб допомогти спланувати розпродажі та рекламні кампанії, починаючи від снігових шин і покрівельних матеріалів до літнього одягу та курортів. Багато океанських лайнерів, а також військово-морські судна використовують оптимальні оцінки маршруту судна для

планування своїх маршрутів у важких морях, щоб мінімізувати втрачений час, потенційні пошкодження та споживання палива. Так само авіакомпанії ретельно враховують атмосферні умови під час планування далеких рейсів, щоб уникнути найсильнішого зустрічного вітру та подорожувати з найсильнішим попутним вітром. На міжнародну торгівлю такими продуктами харчування, як пшениця, кукурудза, боби, цукор, какао та кава, можуть серйозно вплинути новини погоди. З цієї причини банки, товарні трейдери та харчові компанії часто звертаються до синоптиків з проханням попередити про можливість таких раптових змін. Вартість усіх видів товарів і послуг, будь то намети для заходів на відкритому повітрі чи пластикові кришки для щоденних газет, можна знизити або виключити, якщо заздалегідь отримати достовірну інформацію про погоду [1]–[6].

Принципи та методика прогнозування погоди короткостроковий прогноз на основі минулого досвіду про погодні явища оцінюють майбутній стан погоди. У короткостроковому прогнозуванні використовуються радіолокаційні та супутникові спостереження за місцевими атмосферними умовами, швидко обробляються та відображаються комп'ютерами для прогнозування погоди на декілька годин наперед. Оскільки метеорологи можуть вимірювати атмосферні умови для великих територій та аналізувати їх за допомогою комп'ютера, вони можуть робити дещо довгострокові прогнози із значною вправністю. Використовуючи моделі, які швидко, точно та статистично обґрунтовано застосовують накопичений досвід, метеорологи у наш час можуть робити об'єктивні прогнози. На відміну від суб'єктивних оцінок минулого, об'єктивні оцінки є послідовними і можуть бути переглянуті, переоцінені та покращені [1]–[6].

Інший метод передбачає використання даних про минулі погодні явища та події для прогнозування значень певних елементів погоди для певного місця та періоду часу. Він долає слабкі сторони чисельних моделей шляхом розробки статистичних зв'язків між прогнозами моделі та спостережуваною погодою. Ці відносини потім використовуються для перекладу прогнозів

моделі безпосередньо в конкретні прогнози погоди. Наприклад, чисельна модель може взагалі не передбачати появу приземних вітрів, і будь-які вітри, які вона прогнозує, завжди можуть бути дуже сильними. Засоби цього методу можуть автоматично виправляти помилки в швидкості вітру і створювати дуже точні прогнози утворення вітру в певному місці [1]–[6].

Короткострокові прогнози часто втрачають точність, оскільки прогнозисти намагаються дивитися далі в часі. Результати прогнозування є досить прийнятними для періодів приблизно 12 годин, а іноді і до двох діб. Все більша кількість короткострокових методів прогнозування стає економічно мотивованою до практичного застосування [1]–[6].

Сповіднення або інформування про погоду — це особливий вид короткострокового прогнозування, який використовується у т.ч. і для збереження людського життя у випадку катастроф, які можна спрогнозувати заздалегідь. Попередження про погоду видають урядові та військові організації в усьому світі щодо всіх типів загрозливих погодних явищ: тропічних штормів, які по-різному називають ураганами, тайфунами або тропічними циклонами залежно від місця; великі океанічні шторми, які охоплюють сотні кілометрів за межами тропіків і іноді посилюють вітри, які можна порівняти з тропічними штормами; а на суші — повені, сильні вітри, тумани, хуртовини, ожеледиця. Особливі зусилля докладаються для попередження про град, блискавки та пориви вітру, пов'язані з сильними грозами, які іноді називають сильними локальними бурями або просто суворими погодними умовами. Прогнози та попередження також робляться щодо ураганів, які є інтенсивними, вихровими штормами, які представляють собою найсуворіший кінець шкали погоди. Особливо в найбільших системах (іноді їх називають максі-торнадо) ризик знищення майна, травм і смерті на шляху торнадо надзвичайно високий [1]–[6].

Також відстежуються реактивні потоки та сухе повітря, що може об'єднатися, щоб перетворити звичайні грози в рідкісні обертові шторми з похилими вентиляційними отворами повітря, що рухається вгору через схил,

без перешкоди сильних дощів. Ці високошвидкісні повітряні потоки можуть швидко транспортувати велику кількість вологи в холодну верхівку штормів, тим самим сприяючи утворенню великих градів. Град і дощ тягнуть повітря зверху вниз, щоб завершити ланцюг важких, взаємодіючих течій у верхній і нижній течії [1]–[6].

Завдяки точному прогнозуванню складних умов синоптики можуть забезпечити час для мобілізації спеціалізованих мереж спостережень і персоналу. Якщо бурі все-таки розвиваються, на основі прямих спостережень видають спеціальні попередження. Цей двоетапний процес складається із спостереження за ураганом або сильним штормом, який є прогнозом, підготовленим синоптиком за допомогою чисельного методу прогнозування, і попередженням, яке зазвичай видається місцевим об'єктом спостереження. Такий прогноз попереджає певну територію про загрозу, але не намагається визначити, які громади постраждають. На відміну від цього, сповіщення є дуже конкретним і вимагає негайних дій. Різні типи радарів можна використовувати для виявлення великого граду, сильних дощових крапель, відносно відкритої зони швидкого висхідного потоку і навіть обертання під час урагану. Ці індикатори або фактичне спостереження часто викликають попередження про торнадо. Насправді, попередження — це конкретне твердження про те, що небезпека неминуча [1]–[6].

Розширене або довгострокове прогнозування має іншу історію та підхід, ніж короткострокове або середньострокове прогнозування. У більшості випадків у такому випадку не застосовується синоптичний метод прогнозування вперед у часі з конкретної стартової карти. Натомість довгострокові прогнозисти, як правило, використовують кліматологічний підхід, часто маючи справу з широкою картиною погоди за певний період часу, а не намагаючись передбачити щоденні деталі. Більшість довгострокових прогнозів намагаються оцінити відхилення від нормальних умов для даного місяця чи сезону. Такі відриви називаються аномаліями. Карти аномалій не намагаються передбачити погоду на певний день, а

натомість передбачають тенденції (тобто тепліше, ніж зазвичай) на тривалий період часу, наприклад, сезон (тобто весну). Прогнози температурних аномалій надійніші за прогнози опадів, місячні прогнози кращі за сезонні, а прогнозування погоди взимку є дещо точнішим за інші сезони. Використовуються також відомості про стійкість (спекотне літо передуює теплій весні) або антиперсистенції (холодна весна приходить після теплої зими) [1]–[6].

В останній роки підхід до довгострокового прогнозування та очікування різко змінилися, зокрема, вчені почали розглядати аномалії температури поверхні океану як потенційну причину аномалій температури атмосфери в наступні сезони і на великі відстані від місця розташування. Певні повторювані моделі атмосферних потоків у різних частинах світу пов'язані один з одним. Поки були доступні супутникові спостереження, дослідники почали вивчати феномен Ель-Ніньо, а також обґрунтовано утворення аномально теплих або холодних вод океану та аномально високого чи низького атмосферного тиску, які можуть бути взаємопов'язані між собою через величезні глобальні взаємозв'язки. Ці зв'язки і знання про поведінку частин цих великих систем дозволяють прогнозистам робити кращі довгострокові прогнози, принаймні частково, оскільки відповідні властивості змінюються повільно і дещо регулярно. Такий підхід до вивчення зв'язків між атмосферою та океаном може стати початком революційного етапу довгострокового прогнозування. Останнім часом також спостерігається підвищений інтерес до застосування чисельних моделей прогнозування погоди для довгострокового прогнозування. У цьому випадку хвилюють не деталі прогнозу погоди на декілька діб вперед, а об'єктивно прогнозовані аномалії. Надійність довгострокових прогнозів, а також як коротко- та середньострокові прогнози, за останні роки значно покращилася. Тим не менш, багато важливих питань залишаються невирішеними, і для кожного, хто працює в цій галузі, виникають цікаві виклики. Крім того, дуже важливим є інформування людей

про стан погоди, що обумовлює розробку відповідних програмних засобів [1]–[6].

## **1.2 Дослідження програмних засобів для інформування про стан погоди**

У наш час існує досить багато програмних засобів, що використовуються для інформування користувачів про стан погоди.

Для огляду програмних засобів для інформування про стан погоди було обрано програмне забезпечення, у т.ч. телеграм боти, які відповідають за прогнози погоди, а саме: `weatherman_bot`, `meteovunque_bot`, `weatherbot`, `WeatherPlusBot`, `w_weather_bot` [7]–[13].

Програмне забезпечення `weatherman_bot` – програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [7].

Можливості прогнозу погоди програмного забезпечення `weatherman_bot` [7]:

- надсилання погоди щодо обраного міста;
- прогноз погоди на наступний день;
- прогноз погоди на 5 днів.

Переваги програмного забезпечення `weatherman_bot` [7]:

- мультимовність;
- безкоштовний;
- простий інтерфейс.

Недоліки програмного забезпечення `weatherman_bot` [7]:

- не працює увесь час;
- відсутнє щоденне прогнозування.

Програмне забезпечення `meteovunque_bot` – програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [8].

Можливості прогнозу погоди програмного забезпечення meteovunque\_bot [8]:

- надсилання погоди щодо обраного міста;
- прогноз погоди на наступний день;
- прогноз погоди на 5 днів.

Переваги програмного забезпечення meteovunque\_bot [8]:

- використання AI для пришвидшення роботи бота;
- безкоштовний;
- простий інтерфейс.

Недоліки програмного забезпечення meteovunque\_bot [8]:

- не працює увесь час;
- відсутнє щоденне прогнозування.

Програмне забезпечення weatherbot – програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [9].

Можливості прогнозу погоди програмного забезпечення weatherbot [9]:

- можливість пошуку міста за локацією;
- прогноз погоди щодо обраного міста;
- прогноз погоди за обраним містом кожен день;
- прогноз погоди на 3 дні.

Переваги програмного забезпечення weatherbot [9]:

- можливість обрати систему виміру;
- безкоштовний;
- можливість вибору мови.

Недоліки програмного забезпечення weatherbot [9]:

- відсутнє щоденне прогнозування;
- відсутня інструкція використання.

Приклад використання програмного забезпечення weatherbot зображено на рис. 1.1.

## What can this bot do?

Curious about the weather?

Just send me these commands and you'll know a lot better:

-  Get the current weather for a location or city
-  Get a 3-day forecast for a location or city
-  Set current alerts
-  Select a forecast-language
-  Change your unit preferences

Рисунок 1.1 – Робота погодного бота weatherbot

Програмне забезпечення WeatherPlusBot також представляє собою програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [10].

Можливості прогнозу погоди програмного забезпечення WeatherPlusBot [10]:

- надсилання погоди щодо обраного міста;
- прогноз погоди на 5 днів.

Переваги програмного забезпечення WeatherPlusBot [10]:

- мультимовність введення міста;
- безкоштовний;
- простий інтерфейс.

Недоліки програмного забезпечення WeatherPlusBot [10]:

- відсутнє щоденне прогнозування;
- відсутня інструкція використання.

Приклад використання програмного забезпечення WeatherPlusBot зображено на рисунок 1.2.

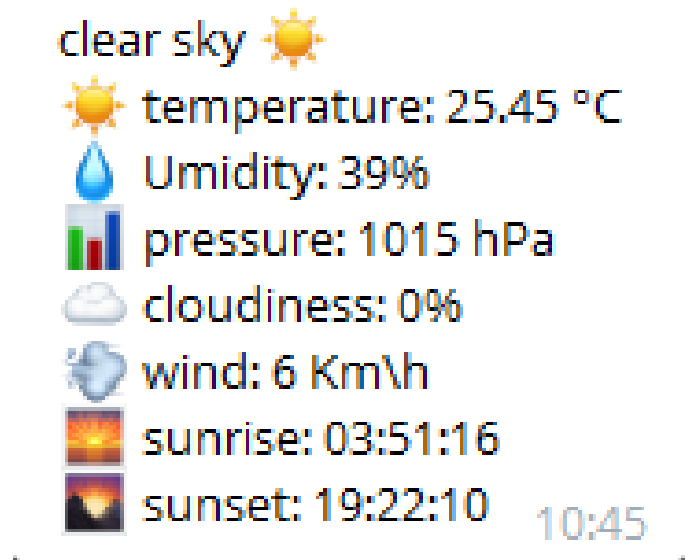


Рисунок 1.2 – Робота погодного бота WeatherPlusBot

Програмне забезпечення `w_weather_bot` також представляє собою програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [11].

Можливості прогнозу погоди програмного забезпечення `w_weather_bot` [11]:

- надсилання погоди щодо обраного міста;
- прогноз погоди на 5 днів.

Переваги програмного забезпечення `w_weather_bot` [11]:

- мультимовність вводу міста;
- безкоштовний;
- простий інтерфейс.

Недоліки програмного забезпечення `w_weather_bot` [11]:

- відсутнє щоденне прогнозування;
- відсутня інструкція використання.

Програмне забезпечення `weatherFor_bot` також представляє собою програмний бот, призначений для інформування про стан погоди за допомогою месенджера телеграм [12].

Можливості прогнозу погоди програмного забезпечення `weatherFor_bot` [12]:

- надсилання погоди щодо обраного міста;
- можливість пошуку міста за локацією;
- можливість обрати місто та час за для щоденного прогнозу погоди.

Переваги програмного забезпечення weatherFor\_bot [12]:

- щоденне прогнозування;
- наявність інструкції щодо використання;
- безкоштовний;
- простий інтерфейс.

Приклад використання програмного забезпечення weatherFor\_bot зображено на рисуюнок 1.3.

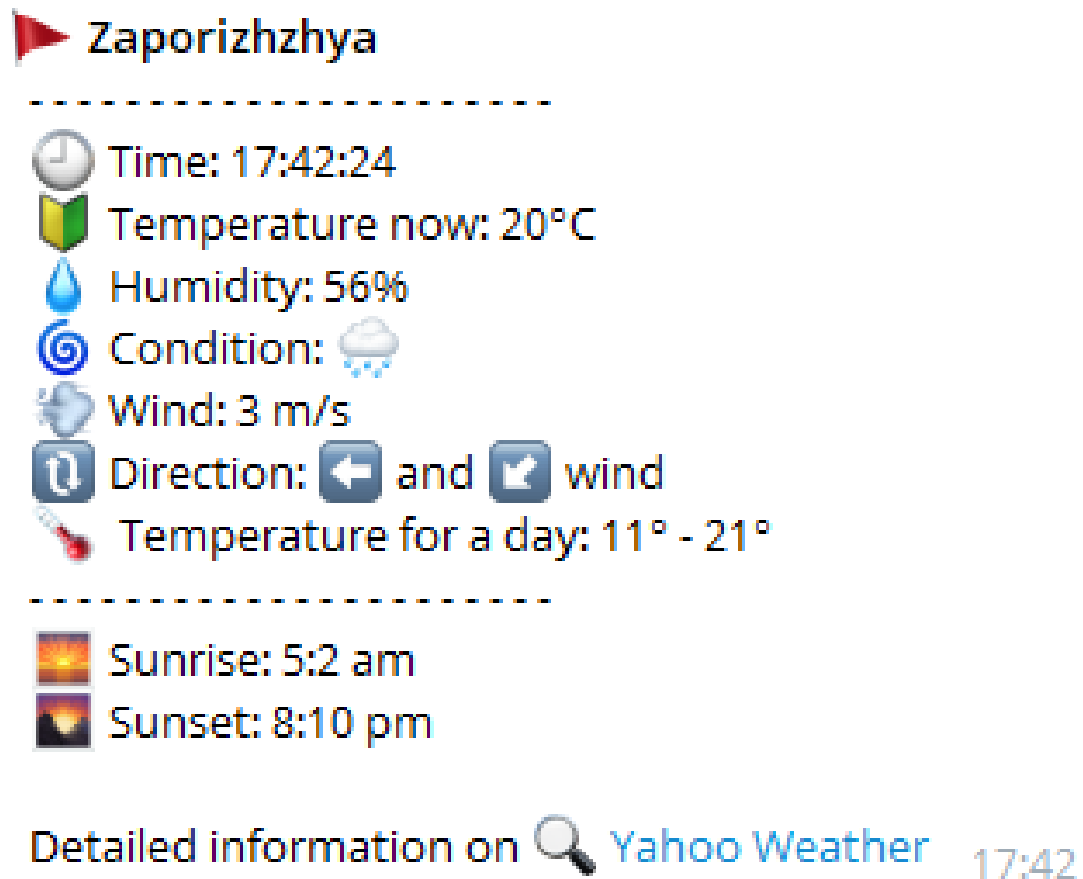


Рисунок 1.3 – Робота погодного бота weatherFor\_bot

Програмне забезпечення Digital Atmosphere — це потужна програма для прогнозування погоди, яка дозволяє створювати детальні карти погоди в режимі реального часу в будь-якій точці світу. Воно широко використовує

методи та алгоритми, які працюють аналогічно, а в деяких випадках і значно краще, ніж дуже дорогі програмні системи Національної метеорологічної служби. Програмне забезпечення Digital Atmosphere працює на будь-якій системі Windows і використовує безкоштовні дані з державних та університетських джерел в Інтернеті (рис. 1.4) [13].

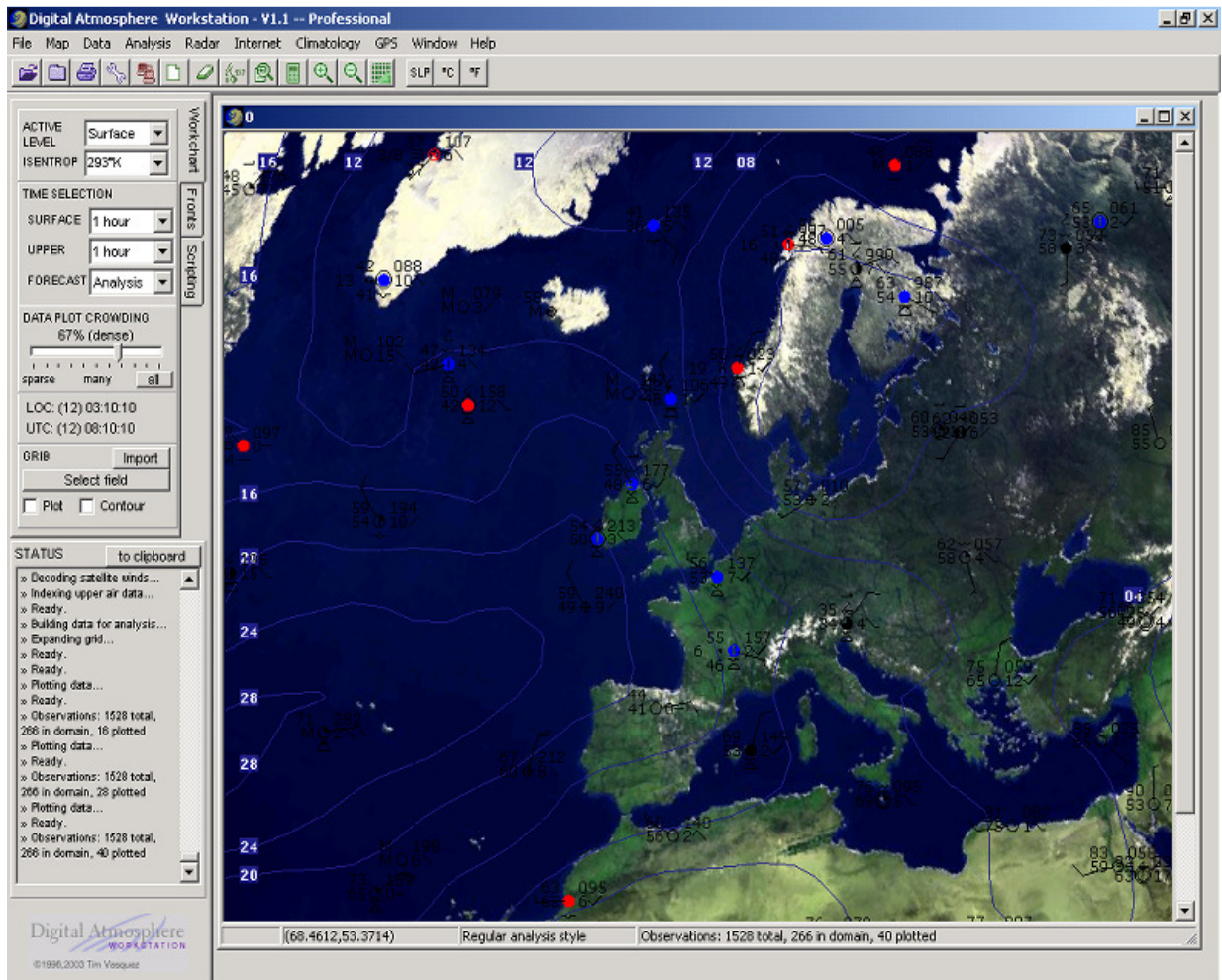


Рисунок 1.4 – Робота програмного забезпечення Digital Atmosphere

Перевагами програмного забезпечення Digital Atmosphere є такі [13]:

- використання ефективних алгоритмів для прогнозування погоди;
- підтримка великої кількості форматів даних;
- зручний інтерфейс.

Недоліки програмного забезпечення Digital Atmosphere є такі [13]:

- незручна система допомоги користувачу;

- необхідність періодичної оплати за користування програмою.

Таким чином, у наш час широко використовується програмне забезпечення для інформування про стан погоди. Порівняльний аналіз програмного забезпечення для інформування про стан погоди наведено у табл. 1.1.

Таблиця 1.1 – Програмне забезпечення для інформування про стан погоди

| Критерій порівняння                   | weatherman_bot | meteovunque_bot | weatherbot | WeatherPlusBot | weather_r_bot | weatherForecast_bot |
|---------------------------------------|----------------|-----------------|------------|----------------|---------------|---------------------|
| Пошук за локацією                     | –              | –               | +          | –              | –             | +                   |
| Мультимовність                        | +              | –               | +          | +              | –             | –                   |
| Прогноз на 3 дні                      | –              | –               | +          | –              | –             | –                   |
| Прогноз на 5 днів                     | +              | +               | –          | +              | +             | –                   |
| Прогноз на тиждень                    | –              | –               | –          | –              | –             | +                   |
| Щоденний прогноз погоди               | –              | –               | +          | –              | –             | +                   |
| Щоденний прогноз погоди в обраний час | –              | –               | –          | –              | –             | +                   |
| Працює весь час                       | –              | –               | +          | +              | –             | +                   |

З приведеної таблиці видно, що не всі програми надають можливість пошуку за локацією, працюють у режимі 24/7, надають можливість щоденного прогнозу погоди на обраний час та ін. Тому доцільною є розробка програмного забезпечення у вигляді телеграм-боту для інформування про стан погоди, яке би виконувало дані функції.

Таким чином, необхідно розробити Telegram-бот для надсилання користувачам прогнозу погоди щодо обраного міста. Обов'язковим також є реалізація щоденного прогнозу погоди. Розроблюваний програмний продукт повинен забезпечувати виконання нижче перерахованих функцій:

- вибір міста користувачем;
- виведення інформації про актуальне значення погоди;
- додавання користувачів до бази даних;
- можливість щоденного автоматичного надсилання погоди користувачу.

### **1.3 Висновки за розділом 1**

Проаналізовано проблему прогнозування погоди. Визначено, що прогнозування погоди здійснюється на основі застосування принципів фізики, підкріплених різноманітними статистичними та емпіричними методами, крім того, безпосередньо прогнозування атмосферних подій, прогноз погоди включає прогнози змін поверхні Землі, викликаних атмосферними умовами (наприклад, сніговим і льодовим покривом, штормовими припливами, повенями). Тому дуже важливим є інформування людей про стан погоди, що обумовлює розробку відповідних програмних засобів.

Визначено, що у наш час існує досить багато програмних засобів, що використовуються для інформування користувачів про стан погоди. Для огляду програмних засобів для інформування про стан погоди було обрано програмне забезпечення, у т.ч. Telegram-боти, які відповідають за прогнози погоди, а саме: `weatherman_bot`, `meteovunque_bot`, `weatherbot`, `WeatherPlusBot`, `w_weather_bot`.

Визначено, що не всі програми надають можливість пошуку за локацією, працюють у режимі 24/7, надають можливість щоденного прогнозу погоди на обраний час та ін. Тому доцільною є розробка програмного забезпечення у вигляді Telegram-боту для інформування про стан погоди, яке би виконувало дані функції.

## **2 МАТЕРІАЛИ І МЕТОДИ**

### **2.1 Вибір мови програмування та інших технологій розробки**

Для створення програмного забезпечення для інформування про стан погоди було обрано мову програмування JavaScript [14]-[15].

Сучасна JavaScript є безпечною мовою програмування. Вона не надає низькорівневого доступу до пам'яті чи центрального процесора, оскільки призначена для браузерів, яким він не потрібен [14]-[15].

Можливості JavaScript залежать від середовища, в якому ця мова працює. Наприклад, Node.JS підтримує читання/запис будь-яких файлів, виконання мережових запитів тощо. У браузері JavaScript можна використовувати все, що стосується керування веб-сторінками, взаємодії користувача та веб-сервера. Наприклад, за допомогою JavaScript можна додавати новий HTML-код на сторінку, змінити наявний вміст і змінити стилі. Ця мова реагує на дії користувача, клацання миші, переміщення вказівника та натискання кнопки. За допомогою JavaScript можна відправляти мережові запити на віддалені сервери, завантажити файли (технології AJAX і COMET), прийняти та встановити файли cookie, ставити запитання гостям та переглядати повідомлення. Також можна працювати зі сховищами даних на стороні клієнта (локальна пам'ять) [14]-[15].

Функції JavaScript у браузері обмежені для безпеки користувачів. Метою є запобігання доступу недобросовісних веб-сайтів до особистої інформації та шкоди інформації користувачів. Прикладами таких обмежень є такі: JavaScript на веб-сторінці не може читати, записувати, копіювати або запускати будь-який файл на жорсткому диску. Операційна система не має прямого доступу до системних функцій. Сучасні браузери дозволяють працювати з файлами, але JavaScript має обмежений доступ і дозволяє користувачеві виконувати лише певні дії, наприклад, перетягувати файл у вікно браузера або вибирати його за допомогою тегу `<input>` [14]-[15].

Існують способи зв'язку з камерами/мікрофонами та іншими пристроями користувача за допомогою JavaScript, але вони потребують спеціальної згоди користувача. Тому сторінка з підтримкою JavaScript не може випадково відкрити веб-камеру, стежити за тим, що відбувається, і надсилати інформацію до інших користувачів. Різні вікна та вкладки не розпізнають одне одного [14]-[15].

Іноді JavaScript використовується для відкриття іншого вікна. Але навіть тоді, якщо JavaScript на одній сторінці надходить з іншого сайту (інший домен, протокол чи порт), користувач не зможе отримати доступ до іншої сторінки. Щоб обійти це обмеження, обидві сторінки повинні прийняти це та містити приватний код JavaScript. Це обмеження все ще необхідне для безпеки користувача. Сторінка користувача не повинна бути доступною або викраденою з будь-якої вкладки браузера. JavaScript може легко взаємодіяти з сервером, на якому розміщена поточна сторінка. Однак можливість отримувати інформацію з інших сайтів/доменів обмежена. Хоча це в принципі можливо, для цього потрібна певна угода з віддаленою стороною (виражена в заголовку HTTP). Ці обмеження не застосовуються, якщо JavaScript використовується поза браузером, наприклад, на сервері [14]-[15].

Сучасні браузери надають доповнення/розширення, які потребують додаткових дозволів [14]-[15].

Суттєвими перевагами JavaScript є такі [14]-[15]:

- повна інтеграція з HTML/CSS.
- простота використання;
- усі основні браузери підтримують JavaScript

Тому це найпоширеніший інструмент для створення інтерфейсу користувача у браузері. Звичайно, JavaScript дозволяє запускати програми не тільки в браузерах, а й на серверах і мобільних пристроях. Синтаксис JavaScript не підходить для всіх потреб. Різним людям потрібні різні здібності. Це природно, тому що проекти різні, і вимоги до них різні. Так, є багато нових мов, які нещодавно було перетворено на JavaScript, перш ніж вони

запрацюють у браузері. Сучасні інструменти роблять переклад на JavaScript настільки швидким і прозорим, що по суті дозволяє розробникам писати код іншою мовою і автоматично конвертувати його в JavaScript [14]-[15].

Таким чином, спочатку JavaScript був розроблений для браузерів, але зараз використовується на багатьох інших платформах. Сьогодні JavaScript унікальний тим, що ця мова є найпопулярнішою мовою браузера, яка повністю інтегрує HTML/CSS. Багато мов можна переключити на JavaScript, щоб забезпечити додаткову функціональність [14]-[15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для інформування про стан погоди наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для інформування про стан погоди

| Критерій порівняння                      | JavaScript | Python | Java | php |
|------------------------------------------|------------|--------|------|-----|
| Швидкість розробки                       | +          | +      | +    | –   |
| Швидкість роботи                         | +          | –      | +    | –   |
| Динамічна типізація                      | +          | +      | –    | +   |
| Кількість готових прикладів              | +          | +      | –    | +   |
| Необхідність серверу при роботі програми | +          | +      | –    | +   |

Таким чином, для реалізації програмного забезпечення для інформування про стан погоди обрано мову програмування JavaScript.

## **2.2 Вибір середовища розробки для розробки програмного забезпечення**

Для створення програмного забезпечення для інформування про стан погоди обрано середовища розробки Atom [16]-[17].

Atom — це сучасний, доступний, текстовий редактор — інструмент, який можна налаштувати, щоб робити будь-що, але й продуктивно використовувати, навіть не торкаючись файлу конфігурації. Кросплатформне редагування Atom працює в різних операційних системах. Його можна використовувати в OS X, Windows або Linux.

Вбудований менеджер пакунків дозволяє встановлювати нові пакети або починати створювати власні. За допомогою Atom Smart Auto Completion Atom можна писати код швидше за допомогою розумного, гнучкого автозаповнення. Браузер файлової системи дозволяє легко переглядати та відкривати окремі файли, цілий проект або декілька проектів в одному вікні. Atom дозволяє розділити інтерфейс на кілька панелей, щоб порівнювати та редагувати код файлів, знайти, переглядати та замінювати текст під час введення у файл або в усіх своїх проектах. Atom дозволяє відкривати та керувати файлами з перегляду дерева, додавати папки проекту з командного рядка, підтримувати натискання клавіш у прив'язках клавіш, відновлювати дані після аварійного вимкнення. Також існує можливість оновлювати середовище вікна Atom з терміналу, виконавши ту саму команду з оновленням середовищем терміналу [16]-[17].

Atom є ядром інструментарію розробника, але зазвичай він не працює самостійно. Atom дозволяє створювати нові гілки, створювати і фіксувати, натискати та витягувати, вирішувати конфлікти злиття, запитів на витяг і багато іншого.

Кросплатформне редагування Atom працює в різних операційних системах. Atom — це настільна програма, створена з інтеграцією HTML, JavaScript, CSS та Node.js. Він працює на Electron, платформі для створення

кроссплатформних додатків за допомогою веб-технологій. Atom поставляється з попередньо встановленими чотирма темами інтерфейсу користувача та вісьмома темами синтаксису як темних, так і світлих кольорів, проте користувач може встановити теми, створені спільнотою Atom, або створити власні теми [16]-[17].

У таблиці 2.2 наведено результати аналізу різних середовищ розробки для створення програмного забезпечення для інформування про стан погоди.

Таблиця 2.2 – Порівняння середовищ розробки для створення програмного забезпечення для інформування про стан погоди

| Критерій порівняння середовищ розробки                                                         | Середовище розробки |                  |         |
|------------------------------------------------------------------------------------------------|---------------------|------------------|---------|
|                                                                                                | Atom                | MS Visual Studio | PyCharm |
| Зручність використання для створення програмного забезпечення для інформування про стан погоди | +                   | +—               | +—      |
| Наявність засобів для роботи з мовою JavaScript                                                | +                   | +—               | —       |
| Засоби редагування файлів проєкту                                                              | +                   | +                | +—      |
| Можливість встановлення власних тем                                                            | +                   | —                | +—      |
| Наявність кроссплатформних засобів створення програмного забезпечення                          | +                   | —                | +       |

Таким чином для створення програмного забезпечення для інформування про стан погоди обрано середовища розробки Atom.

## 2.3 Вибір системи керування базою даних

Для створення програмного забезпечення для інформування про стан погоди обрано систему керування базами даних MongoDB [18]-[19].

MongoDB — це база даних документів з відкритим вихідним кодом, яка забезпечує високу продуктивність, високу доступність і автоматичне масштабування. Запис у MongoDB — це документ, який є структурою даних, що складається з пар полів і значень. Документи MongoDB подібні до об'єктів JSON. Значення полів можуть включати інші документи, масиви та масиви документів [18]-[20].

Переваги використання документів у вигляді баз даних MongoDB є такі [18]-[20]:

- документи (тобто об'єкти) відповідають рідним типам даних у багатьох мовах програмування;
- вбудовані документи та масиви зменшують потребу в дорогих об'єднаннях;
- динамічна схема підтримує плавний поліморфізм.

Високопродуктивна система керування базами даних MongoDB забезпечує високу продуктивність збереження даних. Зокрема, ця система керування базами даних (СКБД) підтримує вбудовані моделі даних, зменшує активність введення-виводу в системі баз даних, індекси підтримують швидші запити і можуть включати ключі від вбудованих документів і масивів. Розширена мова запитів MongoDB підтримує розширену мову запитів для підтримки операцій читання та запису, а також агрегації даних, текстового пошуку. Засіб реплікації MongoDB є досить доступним та називається набором реплік, забезпечує автоматичне перемикання збоїв і резервування даних [18]-[20].

Набір реплік — це група серверів MongoDB, які підтримують той самий набір даних, забезпечуючи резервування та підвищуючи доступність даних. Горизонтальна масштабованість MongoDB забезпечує горизонтальну

масштабованість як частину своєї основної функціональності. Sharding розподіляє дані між кластером машин [18]-[20].

MongoDB — це база даних документів з відкритим вихідним кодом і провідна база даних NoSQL, тобто без використання мови SQL. MongoDB написана на C++. MongoDB — це кросплатформна документно-орієнтована база даних що забезпечує високу продуктивність, високу доступність і легку масштабованість [18]-[20].

MongoDB працює над концепцією колекцій та документів. База даних у MongoDB є фізичним контейнером для колекцій. Кожна база даних отримує власний набір файлів у файловій системі. Колекція – це група документів MongoDB. Це еквівалент таблиці у реляційній СКБД. Колекції не застосовують схему. Документи в колекції можуть мати різні поля. Як правило, усі документи в колекції мають схоже або споріднене призначення. Документ – це набір пар ключ-значення. Документи мають динамічну схему. Динамічна схема означає, що документи в одній колекції не повинні мати однаковий набір полів або структуру, а загальні поля в документах колекції можуть містити різні типи даних [18]-[20].

Будь-яка реляційна база даних має типову схему, яка показує кількість таблиць і зв'язок між цими таблицями. Хоча в MongoDB немає концепції відносин, переваги MongoDB перед реляційними СКБД можна описати так: MongoDB — це база даних документів, в якій одна колекція містить різні документи. Кількість полів, вміст і розмір документа можуть відрізнятися від одного документа до іншого. Структура окремого об'єкта зрозуміла. Без складних з'єднань. MongoDB підтримує можливість глибокого запиту та динамічні запити до документів, використовуючи мову запитів на основі документів, яка майже настільки ж потужна, як SQL. MongoDB легко масштабувати. Перетворення/відображення об'єктів програми в об'єкти бази даних не потрібні [18]-[20].

MongoDB дозволяє використовувати внутрішню пам'ять для зберігання (віконного) робочого набору, що забезпечує швидший доступ до даних.

MongoDB має документно-орієнтоване сховище, дані зберігаються у вигляді документів у форматі JSON. Його можна індексувати за будь-якими атрибутами [18]-[20].

Галузі використання баз даних, створених у MongoDB такі [18]-[20]:

- великі дані;
- управління контентом і доставка;
- мобільна та соціальна інфраструктура;
- управління даними користувача;
- центри (хаби) даних.

MongoDB — це документно-орієнтована база даних. Ця система керування базами даних передбачає організацію даних щодо шаблонів запитів, параметрів індексування, обробки поліморфних об'єктів у коді, виконання ручного об'єднання на клієнті та ін. MongoDB також являє собою базу даних, яка висвітлює використання кількох серверів двома способами: набори реплік і шардінг [18]-[20].

Система керування базами даних MongoDB розповсюджує єдину систему логічної бази даних між кластером машин. MongoDB використовує розділення на основі діапазону для розподілу документів на основі певного ключа сегмента [18]-[20].

MongoDB автоматично балансує дані, пов'язані з кожним шаром. MongoDB дозволяє вмикати та вимикати для кожної колекції (таблиці) дані, що безпеку запису, обробку основного перемикавання, ключі сегментів і балансування сегментів [18]-[20].

MongoDB також надає просту та досить зручну структуру для виконання операцій картування/зменшення або агрегації на кількох комп'ютерах. Це призводить до можливості проекції об'єктів та основних примітивів агрегації [18]-[20].

У таблиці 2.3 наведено порівняння систем керування базами даних для інформування про стан погоди.

Таблиця 2.3 – Порівняння систем керування базами даних для інформування про стан погоди

| Критерій порівняння СКБД                                    | Системи керування базами даних |       |            |
|-------------------------------------------------------------|--------------------------------|-------|------------|
|                                                             | MongoDB                        | MySQL | PostgreSQL |
| Зручність для створення ПЗ для інформування про стан погоди | +                              | +—    | —          |
| Кросплатформність                                           | +                              | +     | +—         |
| Висока продуктивність                                       | +                              | +—    | +          |
| Засоби безпеки даних                                        | +                              | +     | +—         |
| Вартість використання                                       | +                              | +     | +—         |

Таким чином для розробки програмного забезпечення для інформування про стан погоди обрано СКБД MongoDB.

## 2.4 Висновки за розділом 2

Для створення програмного забезпечення для інформування про стан погоди було обрано мову програмування JavaScript, яка є безпечною мовою програмування та не надає низькорівневого доступу до пам'яті чи центрального процесора, оскільки призначена для браузерів, яким він не потрібен. Крім того, функції JavaScript у браузері обмежені для безпеки користувачів.

Для розроблення програмного забезпечення для інформування про стан погоди обрано середовище розробки Atom, яке являє собою сучасний, доступний, текстовий редактор — інструмент, який можна налаштувати, щоб робити будь-що, але й продуктивно використовувати, навіть не торкаючись файлу конфігурації. Міжплатформне редагування Atom працює в різних

операційних системах. Його можна використовувати в OS X, Windows або Linux.

Для створення програмного забезпечення для інформування про стан погоди обрано систему керування базами даних MongoDB, яка дозволяє створювати бази даних документів з відкритим вихідним кодом, і яка забезпечує високу продуктивність, високу доступність і автоматичне масштабування. Документи MongoDB подібні до об'єктів JSON, значення полів можуть включати інші документи, масиви та масиви документів.

### 3 ОПИС ПРОГРАМИ

#### 3.1 Структура програмного забезпечення для інформування про стан погоди

Проект програмного забезпечення для інформування про стан погоди у вигляді набору файлових модулів розташовано в каталозі WeatherFor\_bot.

Файлова структура програмного забезпечення для інформування про стан погоди зображена на рисунку 3.1.

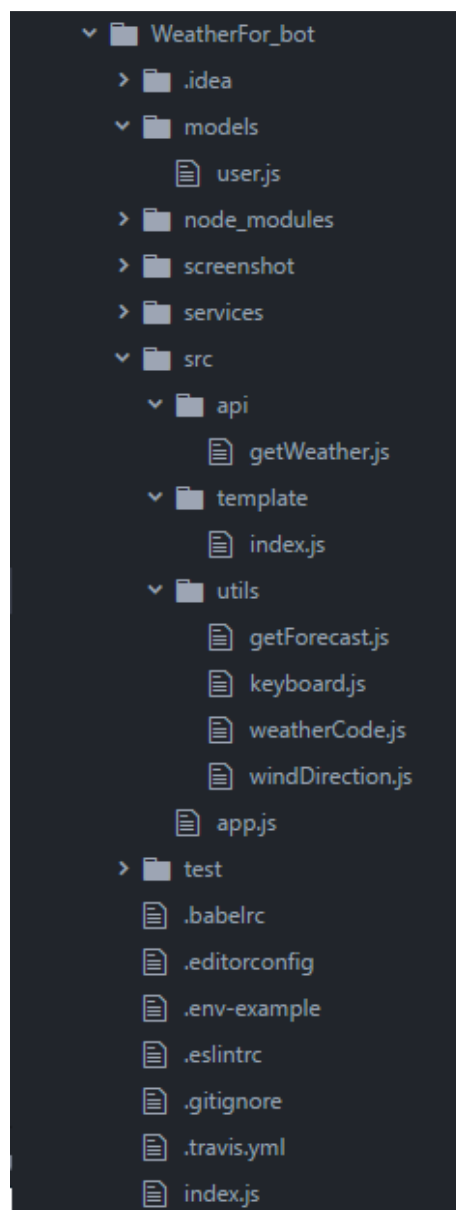


Рисунок 3.1 – Файлова структура програмного забезпечення для інформування про стан погоди WeatherFor\_bot

Структурна схема програмного забезпечення для інформування про стан погоди WeatherFor\_bot продемонстрована на рисунку 3.2.

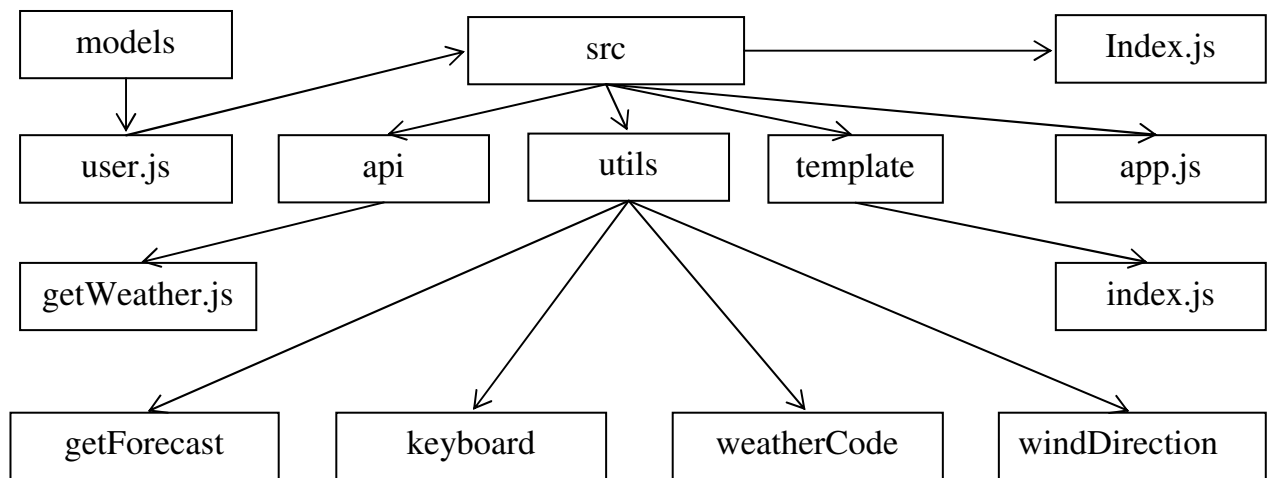


Рисунок 3.2 – Структурна схема програмного забезпечення для інформування про стан погоди

Відповідно до структурної схеми програмне забезпечення для інформування про стан погоди складається з таких компонентів та скриптів:

- user.js – скрипт, який знаходить в папці models відповідає за створення моделі «User» у базі даних MongoDB;
- папка node\_modules – містить потрібні проекту node бібліотеки, за для розробки на іншому ПК треба лише прописати одну команду для встановлення усіх бібліотек;
- папка screenshot – містить скріншот прикладу роботи боту;
- getWeather.js – містить скрипт, який надсилає запит через Yahoo API та отримує прогноз погоди;
- файл index.js у папці template – містить шаблон для відправки повідомлень користувачам;
- getForecast.js – містить скрипт, який оброблює отриманий від запиту прогноз погоди та надсилає його користувачу;

- keyboard.js – містить скрипт, який створює кнопки після повідомлень за для перегляду прогнозу на тиждень та повернення до прогнозу на один день;
- weatherCode.js – містить набір іконок для зображення стану погоди;
- windDirection.js – містить скрипт, який оброблює напрямок вітру та надсилає оформлену інформацію користувачу;
- app.js – містить скрипт, який відповідає за отримання та надсилання повідомлень користувачу ботом, також скрипт виконує підключення до бази даних MongoDB, додає користувачів та редагує їх за потреби. Також цей скрипт відповідає за функцію forecast, тобто щогодинне надсилання прогнозу погоди згідно користувачів у базі даних;
- файл index.js у корені папки – містить підключення інших скриптів та є точкою запуску програми.

### **3.2 Функціонування програмного забезпечення для інформування про стан погоди**

Алгоритм роботи програми складається з трьох основних команд та має просту логіку. Робота з програмою розпочинається з команди “/start”. Далі є 3 команди за для подальшої роботи з ботом:

- “/help” – надсилає користувачу інформацію щодо використання бота;
- “/forecast” – надає можливість отримання користувачем щоденного автоматичного прогнозу погоди та додає користувачів у базу даних;
- “/where” – надає можливість отримати прогноз погоди у обраному місті або отримання прогнозу погоди за локацією.

Повну схему алгоритму функціонування програмного забезпечення для інформування про стан погоди зображено на рисунку 3.3.

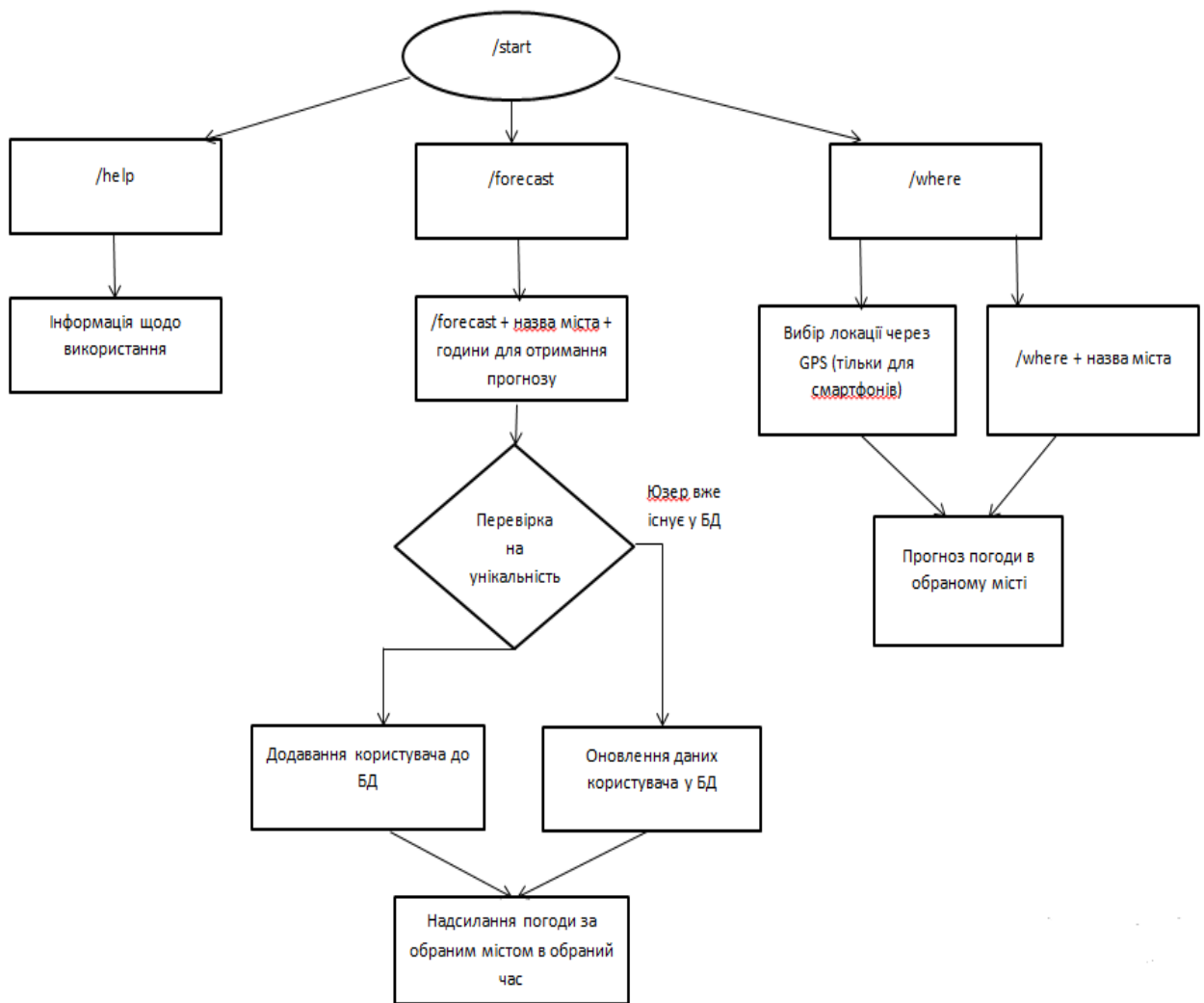


Рисунок 3.3 – Схема функціонування програмного забезпечення для інформування про стан погоди

Алгоритм функціонування програмного забезпечення для інформування про стан погоди є прямим та дуже простим в своїй реалізації. Принцип цього алгоритму полягає у відстеженні повідомлень від користувачів та надсилання прогнозу погоди за обраним містом.

Алгоритм створення бази даних та роботи з файлами реалізовано в файлі `app.js`.

Данні додані до таблиці зберігаються в базі даних. За для оновлення даних використовується пошук по таблиці користувачів за їх `id`.

### 3.3 Опис бази даних

В програмі використана модель даних СКБД MongoDB. У такій моделі бази даних досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій. Опис сутностей представляється за допомогою прямокутних таблиць. Опис таблиці «Користувачі» наводиться у таблиці 3.1.

Таблиця 3.1 – Таблиця «Користувачі»

| Назва поля  | Тип даних | Розмір     |
|-------------|-----------|------------|
| Id          | Лічильник | Довге ціле |
| userName    | Текстовий | 32         |
| firstName   | Текстовий | 32         |
| lastName    | Текстовий | 32         |
| cityName    | Текстовий | 32         |
| weatherTime | Числовий  | 32         |
| userId      | Числовий  | Довге ціле |

Модель «Користувачі» у базі даних MongoDB складається з полів, які вказані у таблиці 3.1. Приклад подання даних у розробленій базі даних зображено на рисунку 3.4.

```

_id: ObjectId("5b0d49e0a19a150e9c8f08f2")
userName: Dima
lastName: ""
cityName: "Zaporizhia"
weatherTime: 21
last: null
userId: 277746
firstName: "Dmitry"
created: 2022-03-29 15:38:56.467
__v: 0

```

Рисунок 3.4 – Приклад подання даних таблиці «Користувачі» у розробленій базі даних

### 3.4 Взаємодія користувача з програмою інформування про стан погоди

Також роботу програми інформування про стан погоди зображено на рис. 3.5 за допомогою діаграми прецедентів, поданої у нотації UML.

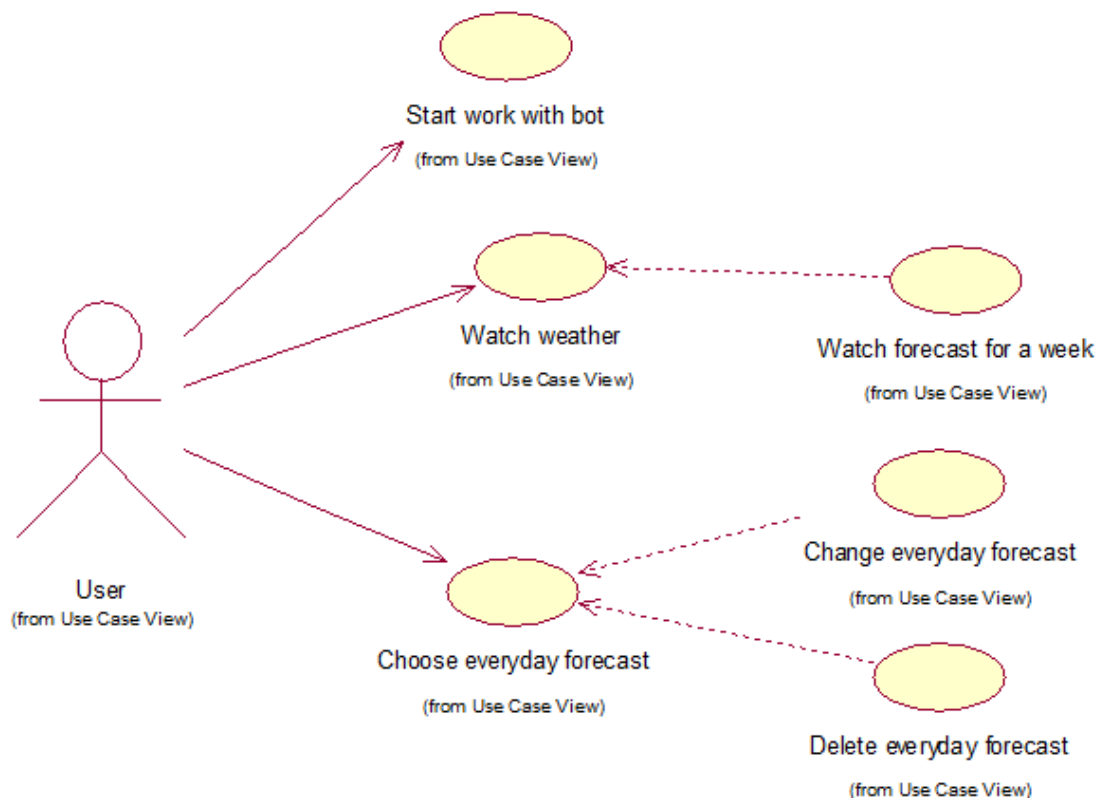


Рисунок 3.5 – Діаграма прецедентів програмного забезпечення для інформування про стан погоди

На рис. 3.5 прецедент відповідає окремому сервісу програмного забезпечення для інформування про стан погоди, визначає один з варіантів її використання і описує типовий спосіб взаємодії користувача з програмним забезпеченням для інформування про стан погоди.

На рис. 3.6 зображено діаграму кооперації програмного забезпечення для інформування про стан погоди.

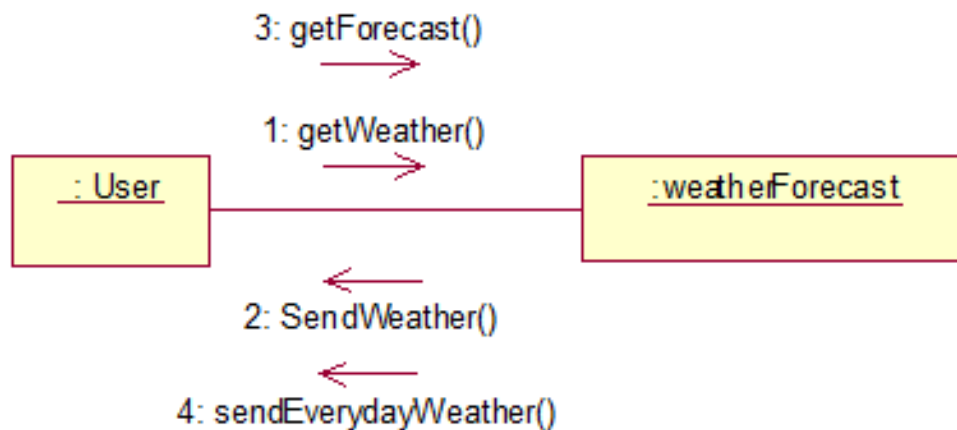


Рисунок 3.6 – Діаграма кооперації програмного забезпечення для інформування про стан погоди

### 3.5 Опис серверної частини

Програмне забезпечення для інформування про стан погоди у вигляді погодного телеграм боту повинно працювати на сервері. Для цього використовується локальний сервер Apache.

Apache є кросплатформним ПО, підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS [21].

Основними перевагами Apache вважаються надійність і гнучкість конфігурації. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СКБД для аутентифікації користувачів, модифікувати повідомлення про помилки і т. д.

Також при розробці використовується локальний сервер Node.js.

Node.js – програмна платформа, заснована на движку V8 перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js додає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду. Node.js застосовується переважно на сервері, виконуючи роль веб-сервера [21].

Сервер node.js починає свою роботу з команди node та назви файлу, який є точкою входу. Приклад використання цього серверу зображено на рисунку 3.7.

```
D:\WebWork\Telegram\WeatherFor_bot>node index  
Successfully connected
```

Рисунок 3.7 – Завантаження серверу node.js

Єдиною вимогою до серверу є швидка обробка запитів користувачів та достатньо потужності за для великої завантаженості [21].

Програмний продукт надсилає запити через API до сервісу погоду. Можливості Weather API зображено на рисунку 3.8.

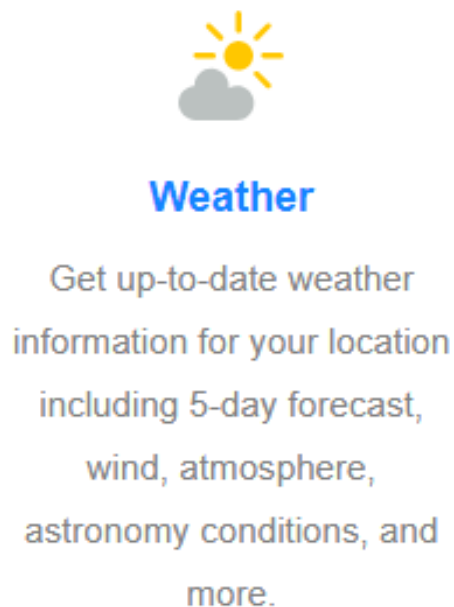


Рисунок 3.8 – Можливості Weather API

Також на рис. 3.9 зображено приклади простих запитів з офіційного сайту Weather API та результату отриманого від сайту щодо прогнозу погоди за обраними містами і т.д. [21].

### Sample Weather API Response

**Examples**  
Forecast for Nome, AK  
Wind for Chicago, IL  
Current conditions for San Diego, CA  
One word conditions for Dallas, TX  
Sunset time for Hawaii  
Weather in Greenland

**Responses**  
☒ JSON ☐ XML  
YQL Query   
  
Test  
Response  

```
{
  "query": {
    "count": 1,
    "created": "2022-03-29T20:34:47Z",
    "lang": "ru-RU",
    "results": {
      "channel": {
        "units": {
          "distance": "mi",
          "pressure": "in",
          "speed": "mph",

```

Endpoint

Рисунок 3.9 – Приклади запитів та результату через Yahoo Weather API

Під час першого разу роботи користувача з ботом після натискання кнопки “/start” створюється чат (рис. 3.10.). Надалі для роботи з ботом потрібно лише зайти у створений чат.



Рисунок 3.10 – Створений чат

## 3.6 Висновки за розділом 3

Розроблене програмне забезпечення для інформування про стан погоди має досить просту структуру. Запропоновано структурну схему програмного забезпечення для інформування про стан погоди. Основними компонентами, скриптами та модулями програми є: скрипт user.js, папка node\_modules, папка screenshot, скрипт getWeather.js, файл index.js, скрипт getForecast.js, скрипт keyboard.js, файли weatherCode.js, windDirection.js, app.js, index.js.

Алгоритм роботи програми складається з трьох основних команд та має просту логіку. Робота з програмою розпочинається з команди “/start”. Далі є 3 команди за для подальшої роботи з ботом: “/help”, “/forecast”, “/where”.

Функціонування програмного забезпечення для інформування про стан погоди полягає у відстеженні повідомлень від користувачів та надсилання прогнозу погоди за обраним містом.

В програмі використана модель даних СКБД MongoDB. У такій моделі бази даних досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій.

Наведено схеми взаємодії користувача з програмою інформування про стан погоди у вигляді діаграми прецедентів та діаграми кооперації, поданих у нотації UML.

## **4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **4.1 Призначення й умови застосування програми**

Програма для інформування про стан погоди призначена для швидкого отримання прогнозу погоди в обраному місті. Програма інформування про стан погоди написана на мові JavaScript. База даних програмного забезпечення для інформування про стан погоди створена в MongoDB.

Програмне забезпечення для інформування про стан погоди забезпечує виконання таких функцій:

- вибір міста користувачем;
- інформування про актуальне значення погоди;
- додавання користувачів до бази даних;
- можливість щоденного автоматичного надсилання погоди користувачу.

Для експлуатації програмного забезпечення для інформування про стан погоди необхідно таке апаратне забезпечення:

- процесор з тактовою частотою 1.0 ГГц або вище;
- об'єм оперативної пам'яті 1Г та більше;
- 100 Мб вільного місця на диску або більше;
- наявність пристроїв для введення та виведення інформації.

Функціональні характеристики розробленого програмного забезпечення для інформування про стан погоди наведено у технічному завданні (додаток А). Опис програмного забезпечення для інформування про стан погоди та текст програмного забезпечення для інформування про стан погоди наведені у додатках Б та В, відповідно.

## **4.2 Характеристики програми**

Програма для інформування про стан погоди розроблена за допомогою мови програмування високого рівня JavaScript та інтегрованого середовища розробки Atom. В якості системи керування БД у програмі для інформування про стан погоди обрана система MongoDB. Програма для інформування про стан погоди базується на сервері. Програма для інформування про стан погоди працює постійно.

Програма для інформування про стан погоди у вигляді Telegram боту weatherFor не надає користувачу можливості зміни своїх параметрів.

## **4.3 Інструкція по експлуатації програми**

### **4.3.1 Звернення до програми**

Для початку роботи з програмним забезпеченням для інформування про стан погоди потрібно знайти бота в пошуку месенджеру Telegram та натиснути кнопку “Start”. Після цього створиться підключення користувача до бота та він розпочне свою роботу у створеному чаті.

### **4.3.2 Вхідні й вихідні дані**

Вхідними даними до програмного забезпечення для інформування про стан погоди є дані, введені користувачем з клавіатури та отримані дані з серверу. Вхідними даними програмного забезпечення для інформування про стан погоди також є кнопки, завдяки яким є можливість обрати знайти за локацією місто чи переглянути прогноз погоди на тиждень.

Вихідними даними програмного забезпечення для інформування про стан погоди (рис. 4.1) програми є інформація щодо прогнозу погоди за обраним містом яку отримує користувач від боту.

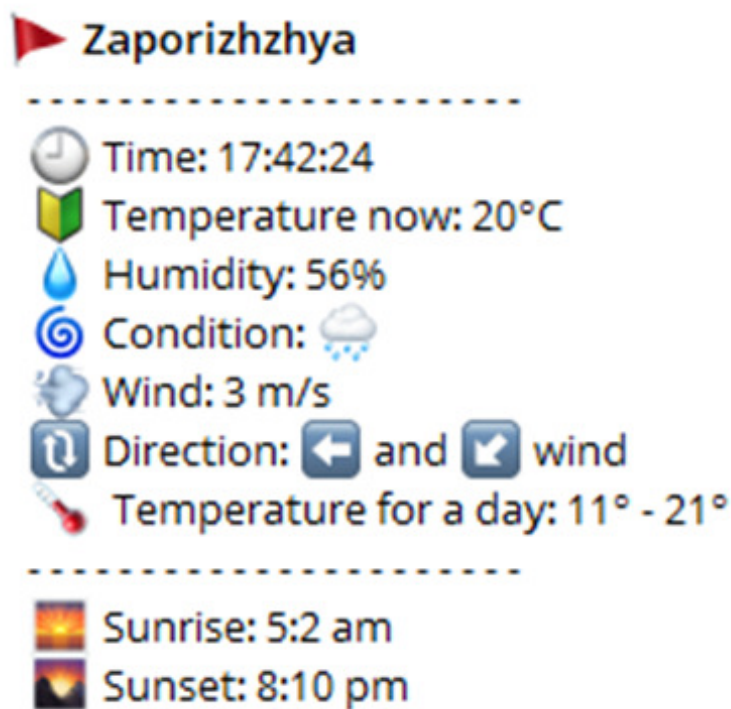


Рисунок 4.1 – Вихідні дані

#### 4.3.3 Повідомлення

Програмне забезпечення для інформування про стан погоди може видавати повідомлення про помилку у разі не підключення до бази даних.

Дії програміста у разі виникнення цієї помилки: перевірити справність підключення до бази даних. Також помилка може виникати у випадку неправильного написання міста користувачем програмного забезпечення для інформування про стан погоди. В цьому випадку програмне забезпечення для інформування про стан погоди оброблює виняткову ситуацію та надає користувачу повідомлення за для виправлення помилки.

#### 4.4 Тестування та результати роботи програмного забезпечення для інформування про стан погоди

Для запуску програмного забезпечення для інформування про стан погоди потрібно упевнитися у відповідності характеристик комп'ютера,

завантажити програмне забезпечення та встановити його в необхідний каталог, далі знайти точку запуску проекту `index.js` у корені проекту, що показано на рис 4.2, та здійснити його запуск.

| Ім'я          | Тип                | Размер |
|---------------|--------------------|--------|
| .idea         | Папка с файлами    |        |
| models        | Папка с файлами    |        |
| node_modules  | Папка с файлами    |        |
| screenshot    | Папка с файлами    |        |
| services      | Папка с файлами    |        |
| src           | Папка с файлами    |        |
| test          | Папка с файлами    |        |
| .babelrc      | Файл "BABELRC"     | 1 КБ   |
| .editorconfig | Файл "EDITORCO..." | 1 КБ   |
| .env-example  | Файл "ENV-EXAM..." | 1 КБ   |
| .eslintrc     | Файл "ESLINTRC"    | 1 КБ   |
|               | Текстовый докум... | 1 КБ   |
| .travis.yml   | Файл "YML"         | 1 КБ   |
| index         | файл JavaScript    | 1 КБ   |
| LICENSE       | Файл               | 2 КБ   |
| package       | Файл "JSON"        | 2 КБ   |
| package-lock  | Файл "JSON"        | 288 КБ |
| README        | Файл "MD"          | 2 КБ   |

Рисунок 4.2 – Точка запуску – файл `index.js`

Робота з програмою для інформування про стан погоди розпочинається відразу після запуску файлу `index.js`. Далі завантажується база даних та у випадку вдалого підключення виводиться повідомлення у консолі (рис. 4.3).

```
Successfully connected
```

Рисунок 4.3 – Повідомлення про підключення до БД

Програма для інформування про стан погоди може видавати повідомлення про помилку за наявності неправильного вводу міста чи надсилання користувачем повідомлень без назви міста, приклад помилки зображений на рис. 4.4.

```
TypeError: Cannot read property 'channel' of null
    at D:/WebWork/Telegram/WeatherFor_bot/src/api/getWeather.js:18:33
    at <anonymous>
    at process._tickCallback (internal/process/next_tick.js:188:7)
```

Рисунок 4.4 – Повідомлення про помилку

Дії оператора програмного забезпечення для інформування про стан погоди у випадку виникнення цієї помилки: не звертати уваги, оскільки помилка оброблюється у програмі та користувач отримує повідомлення щодо правильного вводу, у випадку частого повтору схожих помилок повідомити програмісту.

Тестування програмного забезпечення для інформування про стан погоди проведено за таким сценарієм:

- користувач спробує запросити прогноз погоди за неіснуючим містом;
- якщо програма працює правильно, то користувач отримає повідомлення що треба перевірити назву міста;
- користувач спробує видалити свій профіль з бази даних, коли він у ній відсутній;
- користувач спробує обрати час надсилання погоди менший ніж 1 та більший ніж 24.

Для того щоб розпочати роботу з програмою, потрібно запустити файл index.js. Користувачі можуть переглянути інструкцію щодо використання за допомогою команди “/help” (рис.4.5).

Enter / There is a command prompt available, you can use the location to query the weather,  
 Or type /where yourCityName to find weather information for your city.  
 City you have to write in English.  
 If you want everyday forecast just type /forecast yourCityName yourTime. Time is GMT+3.  
 Example: /forecast Lviv 10 23:47

Рисунок 4.5– Команда “/help”

Після запуску програми, користувачі мають можливість писати Telegram боту та отримувати прогноз погоди відповідно до обраного міста за допомогою команди “/where”.

За для отримання щоденного прогнозу погоди потрібно використовувати команду “/forecast” з назвою міста та часом для отримання користувачами погоди.

За для очищення щоденного прогнозу погоди потрібно використовувати команду “/clear”.

Якщо виникла помилка щодо правильності вводу міста – треба перевірити правильність назви на англійській мові та повторити з правильною назвою. Приклад помилки зображено на рисунку 4.6

Choose position for your query 23:49

Рисунок 4.6– Помилка за відсутності вводу міста

## 4.5 Висновки за розділом 4

Описано програмне забезпечення для інформування про стан погоди.

Визначено мінімальні системні вимоги необхідні для коректної роботи програми для інформування про стан погоди та описано поширені помилки роботи з програмою для інформування про стан погоди та надано інструкції щодо їх вирішення.

Наведено опис процесу тестування програмного забезпечення для інформування про стан погоди та дано інструкції дій користувача у випадку виникнення нестандартних ситуацій. Також було описано процес роботи з програмою для інформування про стан погоди.

Результати тестування програмного забезпечення для інформування про стан погоди підтвердили можливість його використання за призначенням.

## ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для подання інформації користувачам про стан погоди.

Проаналізовано проблему прогнозування погоди. Визначено, що прогнозування погоди здійснюється на основі застосування принципів фізики, підкріплених різноманітними статистичними та емпіричними методами, крім того, безпосередньо прогнозування атмосферних подій, прогноз погоди включає прогнози змін поверхні Землі, викликаних атмосферними умовами (наприклад, сніговим і льодовим покривом, штормовими припливами, повенями). Тому дуже важливим є інформування людей про стан погоди, що обумовлює розробку відповідних програмних.

Визначено, що у наш час існує досить багато програмних засобів, що використовуються для інформування користувачів про стан погоди. Визначено, що не всі програми надають можливість пошуку за локацією, працюють у режимі 24/7, надають можливість щоденного прогнозу погоди на обраний час та ін. Тому доцільною є розробка програмного забезпечення у вигляді телеграм-боту для інформування про стан погоди, яке би виконувало дані функції.

Для створення програмного забезпечення для інформування про стан погоди було обрано мову програмування JavaScript, яка є безпечною мовою програмування та не надає низькорівневого доступу до пам'яті чи центрального процесора, оскільки призначена для браузерів, яким він не потрібен. Крім того, функції JavaScript у браузері обмежені для безпеки користувачів.

Для розроблення програмного забезпечення для інформування про стан погоди обрано середовище розробки Atom, яке являє собою сучасний, доступний, текстовий редактор — інструмент, який можна налаштувати, щоб робити будь-що, але й продуктивно використовувати, навіть не торкаючись

файлу конфігурації. Міжплатформне редагування Atom працює в різних операційних системах. Його можна використовувати його в OS X, Windows або Linux.

Для створення програмного забезпечення для інформування про стан погоди обрано систему керування базами даних MongoDB, яка дозволяє створювати бази даних документів з відкритим вихідним кодом, і яка забезпечує високу продуктивність, високу доступність і автоматичне масштабування. Документи MongoDB подібні до об'єктів JSON, значення полів можуть включати інші документи, масиви та масиви документів.

Розроблене програмне забезпечення для інформування про стан погоди має досить просту структуру. Запропоновано структурну схему програмного забезпечення для інформування про стан погоди. Основними компонентами, скриптами та модулями програми є: скрипт `user.js`, папка `node_modules`, папка `screenshot`, скрипт `getWeather.js`, файл `index.js`, скрипт `getForecast.js`, скрипт `keyboard.js`, файли `weatherCode.js`, `windDirection.js`, `app.js`, `index.js`.

Алгоритм роботи програми складається з трьох основних команд та має просту логіку. Робота з програмою розпочинається з команди `"/start"`. Далі є 3 команди за для подальшої роботи з ботом: `"/help"`, `"/forecast"`, `"/where"`.

Функціонування програмного забезпечення для інформування про стан погоди полягає у відстеженні повідомлень від користувачів та надсилання прогнозу погоди за обраним містом.

В програмі використана модель даних СКБД MongoDB. У такій моделі бази даних досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій.

Наведено схеми взаємодії користувача з програмою інформування про стан погоди у вигляді діаграми прецедентів та діаграми кооперації, поданих у нотації UML.

Визначено мінімальні системні вимоги необхідні для коректної роботи програми для інформування про стан погоди та описано поширені помилки

роботи з програмою для інформування про стан погоди та надано інструкції щодо їх вирішення.

Наведено опис процесу тестування програмного забезпечення для інформування про стан погоди та дано інструкції дій користувача у випадку виникнення нестандартних ситуацій. Також було описано процес роботи з програмою для інформування про стан погоди.

Результати тестування програмного забезпечення для інформування про стан погоди підтвердили можливість його використання за призначенням.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Weather forecasting [Electronic resource]. – Access mode: <https://www.britannica.com/science/weather-forecasting/Principles-and-methodology-of-weather-forecasting>.
2. World Weather Information Service [Electronic resource]. – Access mode: <https://worldweather.wmo.int/en/home.html>.
3. Weather forecasting – ScienceDaily [Electronic resource]. – Access mode: [https://www.sciencedaily.com/terms/weather\\_forecasting.htm](https://www.sciencedaily.com/terms/weather_forecasting.htm).
4. How weather forecasts are created - Met Office [Electronic resource]. – Access mode: <https://www.metoffice.gov.uk/weather/learn-about/how-forecasts-are-made>.
5. Weather forecasting and prediction [Electronic resource]. – Access mode: <https://www.accessscience.com/content/weather-forecasting-and-prediction/742600>.
6. The Challenges and Complexities of Weather Forecasting [Electronic resource]. – Access mode: <https://www.weather.gov/car/weatherforecasting>.
7. Weatherman [Electronic resource]. – Access mode: [https://ru.botostore.com/c/weatherman\\_bot/](https://ru.botostore.com/c/weatherman_bot/).
8. Meteovunque [Electronic resource]. – Access mode: [https://tginfo.ru/bot/news/meteovunque\\_bot](https://tginfo.ru/bot/news/meteovunque_bot).
9. Weatherbot [Electronic resource]. – Access mode: <https://ru.botostore.com/c/weatherbot/>.
10. WeatherPlusBot [Electronic resource]. – Access mode: <https://botostore.com/c/weatherplusbot/>.
11. W\_weather\_bot [Electronic resource]. – Access mode: [https://telegram-robot.ru/bot/w\\_weather\\_bot/](https://telegram-robot.ru/bot/w_weather_bot/).
12. WeatherFor\_bot [Electronic resource]. – Access mode: [https://telegram-robot.ru/bot/weatherFor\\_bot/](https://telegram-robot.ru/bot/weatherFor_bot/).

13. Digital Atmosphere [Electronic resource]. – Access mode: <http://www.weathergraphics.com/da/>.

14. Современный учебник JavaScript [Электрон. ресурс]. – Режим доступа: <https://learn.javascript.ru/>.

15. JavaScript [Электрон. ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript>.

16. Visual Studio Code Vs Atom: Який редактор коду кращий [Електрон. ресурс]. – Режим доступу: <https://uk.myservername.com/visual-studio-code-vs-atom>.

17. Atom IDE [Electronic resource]. – Access mode: <https://www.linkedin.com/pulse/atom-ide-ajay-dubedi>.

18. MongoDB [Electronic resource]. – Access mode: <https://www.mongodb.com>.

19. The MongoDB Database Tools Documentation [Electronic resource]. – Access mode: <https://docs.mongodb.com/database-tools/>.

20. A Review on Various Aspects of MongoDB Databases [Electronic resource]. – Access mode: <https://www.ijert.org/a-review-on-various-aspects-of-mongodb-databases>.

21. Робота з сервером Node.js [Електрон. ресурс]. – Режим доступу: <https://nodejs.org/uk/docs>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Програмне забезпечення для інформування про стан погоди може використовуватися для швидкого отримання прогнозу погоди в обраному місті.

### **A.1 Підстава для розробки**

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для інформування про стан погоди», затверджене наказом Національного університету «Запорізька політехніка» № 102 від 27 квітня 2022 р.

### **A.2 Призначення розробки**

Програмний продукт для інформування про стан погоди призначений для надання інформації користувачам прогнозу погоди в обраному місті.

### **A.3 Основні вимоги до програми, що розробляється**

#### **A.3.1 Вимоги до функціональних характеристик**

Програмне забезпечення для інформування про стан погоди повинно задовольняти такі функціональні характеристики:

- вибір міста користувачем;
- інформування про актуальне значення погоди;
- додавання користувачів до бази даних;
- можливість щоденного автоматичного надсилання погоди користувачу.

### **А.3.2 Вимоги до інтерфейсу програми**

Інтерфейс програми для інформування про стан погоди повинен бути зручним для користувачів програми. Взаємодія користувача з програмою для інформування про стан погоди можлива за допомогою команд у месенджері Телеграм.

### **А.3.3 Вимоги до надійності**

Програмне забезпечення для інформування про стан погоди повинно забезпечувати роботу у критичних ситуаціях.

### **А.3.4 Умови експлуатації**

Для експлуатації програмного забезпечення для інформування про стан погоди необхідно апаратне забезпечення з параметрами, наведеними у наступному підпункті.

### **А.3.5 Вимоги до складу та параметрів технічний засобів**

Для експлуатації програмного забезпечення для інформування про стан погоди необхідні такі апаратні засоби:

- процесор з тактовою частотою 1.0 ГГц або вище;
- об'єм оперативної пам'яті 1Г та більше;
- 100 Мб вільного місця на диску або більше;
- наявність пристроїв для введення та виведення інформації.

### **A.3.6 Вимоги до маркування і пакування**

Програма для інформування про стан погоди може бути записана на різних носіях інформації. На упаковці повинна бути назва програми – «Програмне забезпечення для інформування про стан погоди».

### **A.4 Вхідні дані до роботи**

Вхідними даними до програмного забезпечення для інформування про стан погоди є дані, введені користувачем з клавіатури та отримані дані з серверу. Вхідними даними програмного забезпечення для інформування про стан погоди також є кнопки, завдяки яким є можливість обрати знайти за локацією місто чи переглянути прогноз погоди на тиждень.

Вихідними даними програмного забезпечення для інформування про стан погоди програми є інформація щодо прогнозу погоди за обраним містом яку отримує користувач від боту.

**ДОДАТОК Б**  
**Опис програми**

## **Б.1 Загальні відомості**

Програмне забезпечення для інформування про стан погоди розроблено за допомогою мови програмування JavaScript та інтегрованого середовища розробки Atom. В якості системи керування БД у програмі для інформування про стан погоди обрана система MongoDB.

## **Б.2 Функціональне призначення**

Програма для інформування про стан погоди призначена для швидкого отримання прогнозу погоди в обраному місті.

Програмне забезпечення для інформування про стан погоди забезпечує виконання таких функцій:

- вибір міста користувачем;
- інформування про актуальне значення погоди;
- додавання користувачів до бази даних;
- можливість щоденного автоматичного надсилання погоди користувачу.

## **Б.3 Опис логічної структури**

Програмне забезпечення для інформування про стан погоди складається з таких компонентів та скриптів:

- user.js – скрипт, який знаходить в папці models відповідає за створення моделі «User» у базі даних MongoDB;
- папка node\_modules – містить потрібні проекту node бібліотеки, за для розробки на іншому ПК треба лише прописати одну команду для встановлення усіх бібліотек;
- папка screenshot – містить скріншот прикладу роботи боту;

- `getWeather.js` – містить скрипт, який надсилає запит через Yahoo API та отримує прогноз погоди;
- файл `index.js` у папці `template` – містить шаблон для відправки повідомлень користувачам;
- `getForecast.js` – містить скрипт, який оброблює отриманий від запиту прогноз погоди та надсилає його користувачу;
- `keyboard.js` – містить скрипт, який створює кнопки після повідомлень за для перегляду прогнозу на тиждень та повернення до прогнозу на один день;
- `weatherCode.js` – містить набір іконок для зображення стану погоди;
- `windDirection.js` – містить скрипт, який оброблює напрямок вітру та надсилає оформлену інформацію користувачу;
- `app.js` – містить скрипт, який відповідає за отримання та надсилання повідомлень користувачу ботом, також скрипт виконує підключення до бази даних MongoDB, додає користувачів та редагує їх за потреби. Також цей скрипт відповідає за функцію `forecast`, тобто щогодинне надсилання прогнозу погоди згідно користувачів у базі даних;
- файл `index.js` у корені папки – містить підключення інших скриптів та є точкою запуску програми.

#### **Б.4 Використані технічні засоби**

Для експлуатації програмного забезпечення для інформування про стан погоди необхідно таке апаратне забезпечення:

- процесор з тактовою частотою 1.0 ГГц або вище;
- об'єм оперативної пам'яті 1Г та більше;
- 100 Мб вільного місця на диску або більше;
- наявність пристроїв для введення та виведення інформації.

## **Б.5 Виклик і завантаження**

Для початку роботи з програмним забезпеченням для інформування про стан погоди потрібно знайти бота в пошуку месенджеру Telegram та натиснути кнопку “Start”. Після цього створиться підключення користувача до бота та він розпочне свою роботу у створеному чаті.

## **Б.6 Вхідні та вихідні дані**

Вхідними даними до програмного забезпечення для інформування про стан погоди є дані, введені користувачем з клавіатури та отримані дані з серверу. Вхідними даними програмного забезпечення для інформування про стан погоди також є кнопки, завдяки яким є можливість обрати знайти за локацією місто чи переглянути прогноз погоди на тиждень.

Вихідними даними програмного забезпечення для інформування про стан погоди програми є інформація щодо прогнозу погоди за обраним містом яку отримує користувач від боту.

**ДОДАТОК В**  
**Текст програми**

models/user.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const modelName = 'User';
const modelSchema = Schema(
  {
    userId:
      {
        type: Number,
        required: true,
        index: {unique: true}
        //access: Number
      },
    userName:
      {
        type:String,
        index: {unique: true},
        default: ''
      },

    firstName:
      {
        type:String,
        required:true
      },

    lastName:
      {
        type:String,
        default: ''
      },

    cityName:
      {
```

```

    type:String,
    required:true,
    default: null
  },

  weatherTime:
  {
    type:Number,
    required:true,
    default: 8
  },

  created:
  {
    type: Date,
    default: () => new Date()
  },

  last:
  {
    type: Date,
    default: null
  }
});

const User = mongoose.model(modelName, modelSchema);
module.exports = User;

```

#### src/api/getWeather.js

```

import fetch from 'isomorphic-unfetch';

const getWeatherInfo = (query, u = 'c') => {
  const arg =
    typeof query === 'object'

```

```

    ? `(${query.latitude}, ${query.longitude})`
    : query;
    const api = `https://query.yahooapis.com/v1/public/yql?q=select
* from weather.forecast where u='${u}' AND woeid in (select woeid
from geo.places(1) where text="${arg}")&format=json`;

    return fetch(api)
      .then(response => response.json())
      .then(json => {
        if (json['error']) {
          throw json;
        }

        return json.query.results.channel;
      });
  };

const getWeather = async city => {
  try {
    const result = await getWeatherInfo(city);
    return result;
  } catch (err) {
    console.log(err);
  }
};

export default getWeather;

```

**src/template/index.js**

```

export const weatherTemplate = (response, date, time, direction)
=>
`
  □ *${response.location.city}*
  - - - - -
  Time: ${time}
  Temperature now: ${response.item.condition.temp}°C

```

```

Humidity: ${response.atmosphere.humidity}%
Condition: ${weatherEmoji(response.item.condition.code)}
Wind: ${Math.round(response.wind.speed/3.6)} m/s
Direction: ${direction}
Temperature for a day: ${response.item.forecast[0].low}° -
${response.item.forecast[0].high}°
- - - - -
Sunrise: ${response.astronomy.sunrise}
Sunset: ${response.astronomy.sunset}

Detailed information on [Yahoo Weather](${response.link})
`;

```

```

export const forecastTemplate = (text, link) =>
`☀ Weather for a week

${text}

Details [Yahoo Weather](${link})`;

```

```

utils/getForecast.js
import format from 'date-fns/format';
import weatherEmoji from './weatherCode';
import { forecastTemplate } from '../template';

const getForecast = response => {
  let text = [];
  let fiveDayForecast = '';
  let result;
  const days = response.item.forecast.slice(0, 8);

  days.map(day => {
    let date = format(day.date, 'YYYY-MM-DD');
    let message = `
    • *${date} - ${day.day}*

```

```

□ The lowest and the highest temperature ➡ `${day.low}°C -
`${day.high}°C
Weather conditions ➡ `${weatherEmoji(day.code)}
`;

    return text.push(message);
  });
text.map(day => (fiveDayForecast += `${day} \n`));
result = forecastTemplate(fiveDayForecast, response.link);

return result;
};
export default getForecast;

utils/windDirection.js

const guessDirection = input => {
  const keyArray = Object.keys(windObj);
  const filtered = keyArray.filter(value => {
    return value >= parseInt(input);
  });

  return filtered[0];
};

const windDirection = input => {
  const direction = guessDirection(input);

  if (windObj[input]) return `${windObj[input]} wind`;
  if (direction - 45 < 45) return `${windObj[direction]}
Northerly wind`;

  return `${windObj[direction]} and ${windObj[direction - 45]}
wind`;
};

```

```
export default windDirection;
```

**src/app.js**

```
import TelegramBot from 'node-telegram-bot-api';
import format from 'date-fns/format';
import getWeather from './api/getWeather';
import windDirection from './utils/windDirection';
import getForecast from './utils/getForecast';
import { today, fiveDays } from './utils/keyboard';
import { startMessage, weatherTemplate, helpMessage,
russianHelpMessage } from './template';
import cron from 'node-cron';

const TOKEN = '552522210:AAE3gXa94ol_Wifi6mQRxri7gc-FAPgerbA';
const bot = new TelegramBot(TOKEN, { polling: true });
const mongoose = require('mongoose');
let todayWeatherInfo;
let forecastInfo;
const User = require('../models/user');

bot.onText(new RegExp(`(.*)`), (msg, [source, match]) =>
{
  console.log(msg.chat);
  console.log(`text: `, msg.text);
  console.log(`\n`);
});

bot.onText(/\/start/, message => {
  const chatId = message.chat.id;
  return bot.sendMessage(chatId, startMessage, { parse_mode:
'markdown' });
});
```

```

bot.onText(/\//help/, message => {
  const chatId = message.chat.id;
  return bot.sendMessage(chatId, helpMessage, { parse_mode:
'markdown' });
});

```

```

bot.onText(/\//хелп/, message => {
  const chatId = message.chat.id;
  return bot.sendMessage(chatId, russianHelpMessage, {
parse_mode: 'markdown' });
});

```

```

bot.onText(/\//location/, message => {
  const chatId = message.chat.id;
  const text = 'Click to take your position!';
  const replyMarkup = JSON.stringify({
    keyboard: [
      [
        {
          text: '•Take position',
          request_contact: false,
          request_location: true
        }
      ]
    ],
    callback_data: 'location',
    resize_keyboard: true,
    one_time_keyboard: true
  });
});

```

```

return bot.sendMessage(chatId, text, {
  reply_markup: replyMarkup,
  parse_mode: 'markdown'
});
});

```

```
//##### Звичайний
пошук погоди
```

```
bot.onText(/\//where( (.+))?/, (message, match) => {
  const chatId = message.chat.id;
  const city = match[1];
  if (match[1] === undefined) {
    bot.sendMessage(chatId, 'Choose position for your query');
    return;
  }

  getWeather(city)
    .then(response => {
      const direction = windDirection(response.wind.direction);
      const date = format(new Date(), 'YYYY-MM-DD');
      const time = format(new Date(), 'HH:mm:ss');
      const message = weatherTemplate(response, date, time,
direction);
      forecastInfo = getForecast(response);
      todayWeatherInfo = message;

      return message;
    })
    .then(message => {
      bot.sendMessage(chatId, message, {
        reply_markup: fiveDays,
        parse_mode: 'markdown'
      });
    })
    .catch(err => {
      if (err.name === 'TypeError') {
        bot.sendMessage(chatId, 'API Error, please try again!');
      }
    });
});
```

```
});
```

```
//##### Пошук через  
локацію
```

```
bot.on('location', message => {
  const chatId = message.chat.id;

  getWeather(message.location)
    .then(response => {
      const direction = windDirection(response.wind.direction);
      const date = format(new Date(), 'YYYY-MM-DD');
      const time = format(new Date(), 'HH:mm:ss');
      const message = weatherTemplate(response, date, time,
direction);
      forecastInfo = getForecast(response);
      todayWeatherInfo = message;

      return message;
    })
    .then(res => {
      bot.sendMessage(chatId, res, {
        reply_markup: fiveDays,
        parse_mode: 'markdown'
      });
    })
    .catch(err => {
      if (err.name === 'TypeError') {
        bot.sendMessage(chatId, 'API Error, please try to
reposition! ');
      }
    });
});
```

```
bot.on('callback_query', message => {
```

```

if (message.data === 'forecast') {
  const optional = {
    reply_markup: today,
    parse_mode: 'markdown',
    chat_id: message.message.chat.id,
    message_id: message.message.message_id
  };

  bot.editMessageText(forecastInfo, optional);
}

if (message.data === 'today') {
  const optional = {
    reply_markup: fiveDays,
    parse_mode: 'markdown',
    chat_id: message.message.chat.id,
    message_id: message.message.message_id
  };

  bot.editMessageText(todayWeatherInfo, optional);
}
});

//##### Прогноз
погоды кожного дня в зазначений час

mongoose.connect('mongodb://localhost/weatherBot', function
(err) {
  if (err) {
    throw err;
    console.log('some error');
  }
  console.log('Successfully connected');

  var task = cron.schedule('1 1 * * * *', function(){
    let date = new Date();

```

```

let hoursNow = date.getHours();
    User.aggregate(
    [
        { $match: {weatherTime : hoursNow} },
        { $group : {_id: "$cityName", userid: { $push: "$userId" }}
    ]
    )
    .then(cities =>
    {
        if(cities.length==0) {
            console.log(`\n`);
            console.log(`Сейчас ${hoursNow} часов.`)
            console.log(`\n`);
        }
        else {
            console.log(`\n`);
            console.log(`Сейчас ${hoursNow} часов.`)
            console.log(cities);
            console.log(`\n`);
        }
        cities.forEach((city)=>{
            getWeather(city._id)
                .then(response => {
                    const direction =
windDirection(response.wind.direction);
                    const date = format(new Date(), 'YYYY-MM-DD');
                    const time = format(new Date(), 'HH:mm:ss');
                    const message = weatherTemplate(response, date,
time, direction);
                    forecastInfo = getForecast(response);
                    todayWeatherInfo = message;
                    return message;
                })
                .then(message => {
                    city.userid.forEach((id)=>{

```

```

        bot.sendMessage(id, message, {
            reply_markup: fiveDays,
            parse_mode: 'markdown'
        });
    }); //foreach 2
})
.catch(err => {
    if (err.name === 'TypeError') {
        bot.sendMessage(chatId, 'API Error, please try
again! ');
        count += 1;
    }
});
}); //forEach 1
})
.catch(err => {console.log(err)});
}); //cron

bot.onText(/\\/forecast( (.+)?)?/, (message, match) => {

    const chatId = message.chat.id;
    let userMessage = [];
    userMessage = match[0].split(' ');
    const city = userMessage[1];
    let weatherTime = userMessage[2];
    let count = 0;

    if (userMessage[1] === undefined) {
        bot.sendMessage(chatId, 'Choose position for your query');
        return;
    }

    if (userMessage[2] === undefined) {
        bot.sendMessage(chatId, 'Choose your time or your time is
8:00 GTM+3');
    }

```

```

    weatherTime = 8;
}

if(parseInt(userMessage[2])<0 || parseInt(userMessage[2])>=24){
    bot.sendMessage(chatId, 'Change your time from 0 to 23 or
your time is 8:00 GTM+3');
    weatherTime = 8;
}
var user = new User({
    userId: chatId,
    userName: message.chat.username,
    firstName: message.chat.first_name,
    lastName: message.chat.last_name,
    cityName: `${city}`,
    weatherTime: `${weatherTime}`
});

User.find({userId:chatId})
.exec(function(err, users) {
    if (err) throw err;

    if (users.length==0) {
        user.save(function(err) {
            if (err) throw err;

            console.log(`User successfully saved.`);
        });
    }
    else {
        User.update({userId:chatId}, { $set: {    userId: chatId,
            userName: message.chat.username,
            firstName: message.chat.first_name,
            lastName: message.chat.last_name,
            cityName: `${city}`,
            weatherTime: `${weatherTime}`
        } })
    }
}

```

```

        .then((result) => {console.log(result)})
        .catch((e) => {console.log(e)});
        console.log(`User successfully saved.`);
    };
}) // find
if(count === 1) {
    bot.sendMessage(chatId, 'API Error, please check your cityName
and try again!');
    count = 0;
    return;
}
else{

bot.sendMessage(chatId, `Bot will send you weather in ${city}
everyday at ${weatherTime} (GTM+3).`);
bot.sendMessage(chatId, `If you wanna clear your forecast use
\`/clear\` command.`);
    }
}); //forecast

bot.onText(/\`/clear/, message => {
    const chatId = message.chat.id;
    User.remove({userId:chatId})
        .then((result) => {console.log(result)})
        .catch((e) => {console.log(e);
        bot.sendMessage(chatId, `Your forecast already clear`);
        });
        bot.sendMessage(chatId, `Your forecast successfully
removed`);
        console.log(`User forecast successfully removed.`);

    }); //clear
    });

```

**ДОДАТОК Д**  
**Слайди презентації**

**Міністерство освіти та науки України  
Національний університет «Запорізька політехніка»**

## **Програмне забезпечення для інформування про стан погоди**

Сарана Олег Олегович

ст. групи КНТ-218

керівник: Олійник Андрій Олександрович

д.т.н., професор НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

## **Об'єкт, предмет та мета роботи**

**Об'єкт дослідження:** процес розробки програмного забезпечення для подання інформації користувачам про стан погоди.

**Предмет дослідження:** програмне забезпечення для інформування користувачів про стан погоди.

**Мета роботи:** розробка програмного забезпечення для інформування користувачів про стан погоди.

Рисунок Д.2 – Слайд 2

## Постановка завдання

Розроблення програмного забезпечення для інформування користувачів про стан погоди.

Застосунок повинен виконувати такі функціональні вимоги:

- вибір міста користувачем;
- виведення інформації про актуальне значення погоди;
- додавання користувачів до бази даних;
- можливість щоденного автоматичного надсилання погоди користувачу.

3

Рисунок Д.3 – Слайд 3

## Порівняння існуючих аналогів

| Критерій порівняння                   | weather man_bot | meteovunque_bot | weatherbot | WeatherPlusBot | w_weather_bot | weatherFor_bot |
|---------------------------------------|-----------------|-----------------|------------|----------------|---------------|----------------|
| Пошук за локацією                     | –               | –               | +          | –              | –             | +              |
| Мультиязычність                       | +               | –               | +          | +              | –             | –              |
| Прогноз на 3 дні                      | –               | –               | +          | –              | –             | –              |
| Прогноз на 5 днів                     | +               | +               | –          | +              | +             | –              |
| Прогноз на тиждень                    | –               | –               | –          | –              | –             | +              |
| Щоденний прогноз погоди               | –               | –               | +          | –              | –             | +              |
| Щоденний прогноз погоди в обраний час | –               | –               | –          | –              | –             | +              |
| Працює весь час                       | –               | –               | +          | +              | –             | +              |

4

Рисунок Д.4 – Слайд 4

## Порівняння мов програмування

| Критерій порівняння                      | JavaScript | Python | Java | php |
|------------------------------------------|------------|--------|------|-----|
| Швидкість розробки                       | +          | +      | +    | –   |
| Швидкість роботи                         | +          | –      | +    | –   |
| Динамічна типізація                      | +          | +      | –    | +   |
| Кількість готових прикладів              | +          | +      | –    | +   |
| Необхідність серверу при роботі програми | +          | +      | –    | +   |

5

Рисунок Д.5 – Слайд 5

## Порівняння середовищ розробки

| Критерій порівняння середовищ розробки                                                         | Середовище розробки |                  |         |
|------------------------------------------------------------------------------------------------|---------------------|------------------|---------|
|                                                                                                | Atom                | MS Visual Studio | PyCharm |
| Зручність використання для створення програмного забезпечення для інформування про стан погоди | +                   | +-               | +-      |
| Наявність засобів для роботи з мовою JavaScript                                                | +                   | +-               | –       |
| Засоби редагування файлів проекту                                                              | +                   | +                | +-      |
| Можливість встановлення власних тем                                                            | +                   | –                | +-      |
| Наявність кросплатформених засобів створення програмного забезпечення                          | +                   | –                | +       |

6

Рисунок Д.6 – Слайд 6

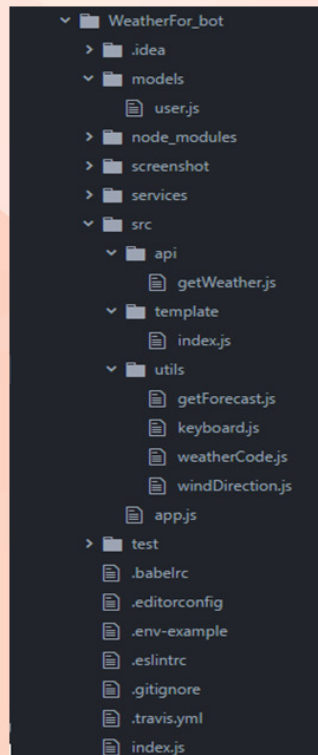
## Порівняння систем керування базами даних

| Критерій порівняння СКБД                                    | Системи керування базами даних |       |            |
|-------------------------------------------------------------|--------------------------------|-------|------------|
|                                                             | MongoDB                        | MySQL | PostgreSQL |
| Зручність для створення ПЗ для інформування про стан погоди | +                              | +–    | –          |
| Кросплатформеність                                          | +                              | +     | +–         |
| Висока продуктивність                                       | +                              | +–    | +          |
| Засоби безпеки даних                                        | +                              | +     | +–         |
| Вартість використання                                       | +                              | +     | +–         |

7

Рисунок Д.7 – Слайд 7

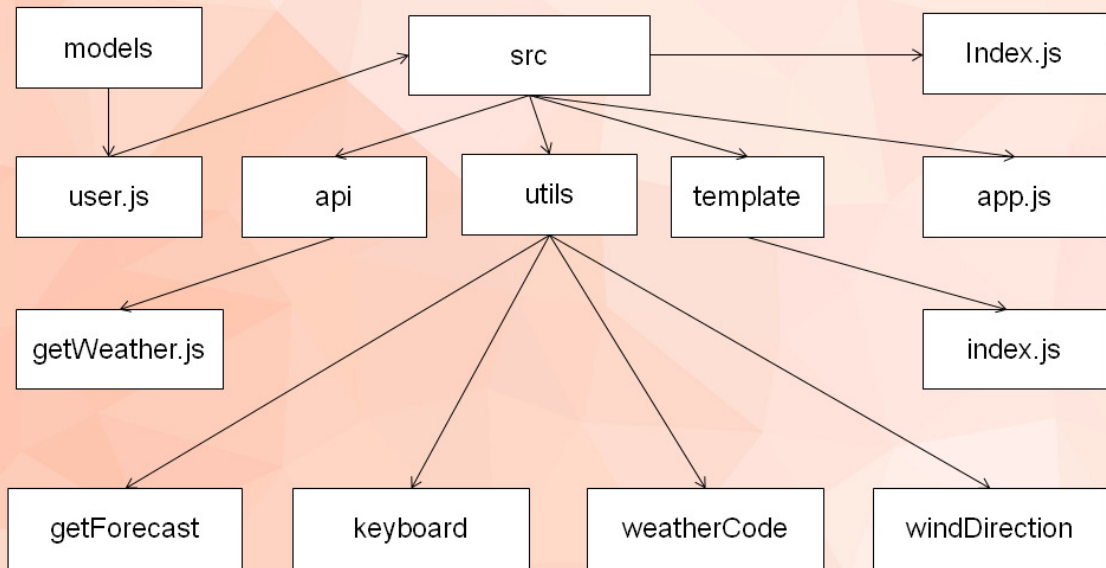
## Файлова структура програми



8

Рисунок Д.8 – Слайд 8

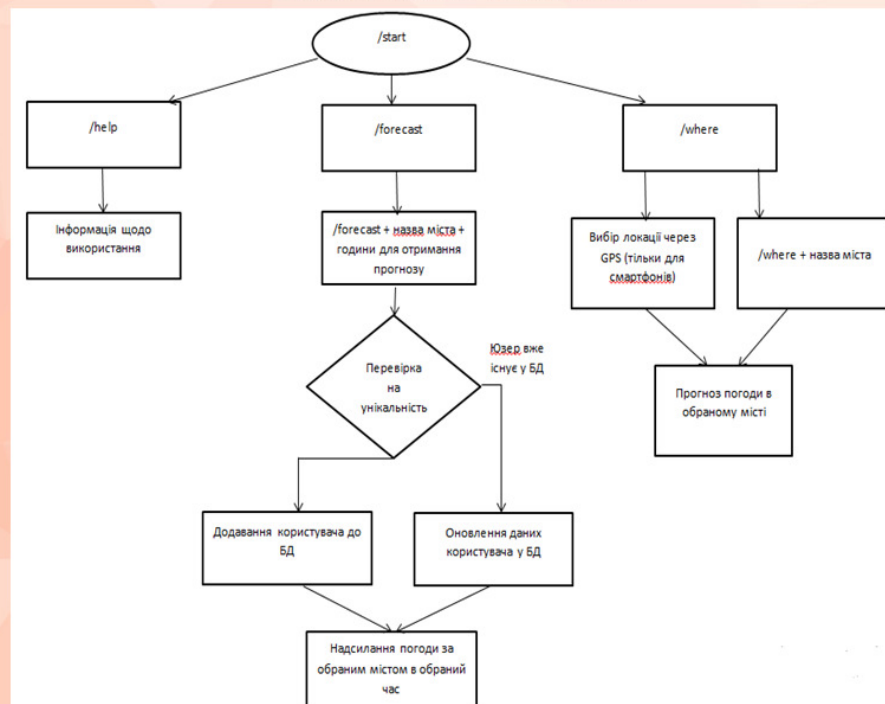
## Структурна схема програмного забезпечення для інформування про стан погоди



9

Рисунок Д.9 – Слайд 9

## Схема функціонування програмного забезпечення



10

Рисунок Д.10 – Слайд 10

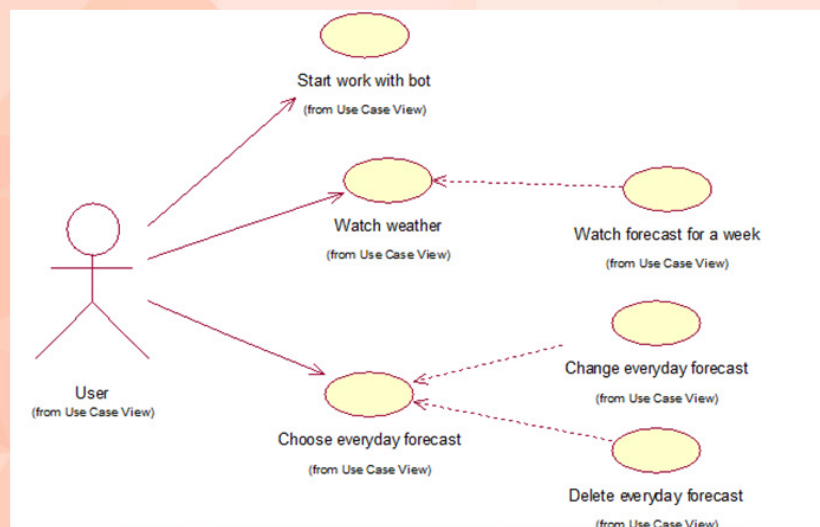
## Приклад подання даних таблиці «Користувачі» у розробленій базі даних

```
_id: ObjectId("5b0d49e0a19a150e9c8f08f2")
userName: Dima
lastName: ""
cityName: "Zaporizhia"
weatherTime: 21
last: null
userId: 277746
firstName: "Dmitry"
created: 2022-03-29 15:38:56.467
__v: 0
```

11

Рисунок Д.11 – Слайд 11

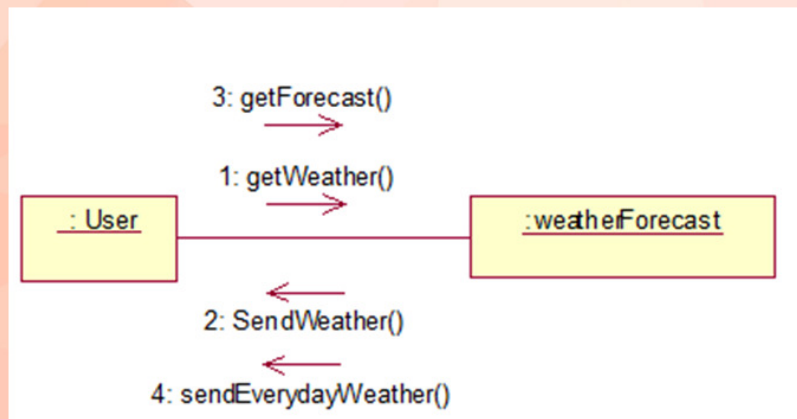
## Діаграма прецедентів програмного забезпечення для інформування про стан погоди



12

Рисунок Д.12 – Слайд 12

## Діаграма кооперації програмного забезпечення для інформування про стан погоди



13

Рисунок Д.13 – Слайд 13

## Робота з програмою

```
D:\WebWork\Telegram\WeatherFor_bot>node index
Successfully connected
```

```
TypeError: Cannot read property 'channel' of null
    at D:\WebWork\Telegram\WeatherFor_bot/src/api/getWeather.js:18:33
    at <anonymous>
    at process._tickCallback (internal/process/next_tick.js:188:7)
```

```
Enter / There is a command prompt available, you can use the
location to query the weather,
Or type /where yourCityName to find weather information for your
city.
City you have to write in English.
If you want everyday forecast just type /forecast yourCityName
yourTime. Time is GTM+3.
Example: /forecast Lviv 10
```

23:47

### ▶ Zaporizhzhya

```

Time: 17:42:24
Temperature now: 20°C
Humidity: 56%
Condition: 
Wind: 3 m/s
Direction:  and  wind
Temperature for a day: 11° - 21°

Sunrise: 5:2 am
Sunset: 8:10 pm
  
```

14

Рисунок Д.14 – Слайд 14

## Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для подання інформації користувачам про стан погоди.

Проаналізовано проблему прогнозування погоди. Визначено, що прогнозування погоди здійснюється на основі застосування принципів фізики, підкріплених різноманітними статистичними та емпіричними методами, крім того, безпосередньо прогнозування атмосферних подій, прогноз погоди включає прогнози змін поверхні Землі, викликаних атмосферними умовами (наприклад, сніговим і льодовим покривом, штормовими припливами, повенями).

Для створення програмного забезпечення для інформування про стан погоди було обрано мову програмування JavaScript, яка є безпечною мовою програмування та не надає низькорівневого доступу до пам'яті чи центрального процесора, оскільки призначена для браузерів, яким він не потрібен. Крім того, функції JavaScript у браузері обмежені для безпеки користувачів.

15

Рисунок Д.15 – Слайд 15

## Висновки

Для розроблення програмного забезпечення для інформування про стан погоди обрано середовище розробки Atom, яке являє собою сучасний, доступний, текстовий редактор — інструмент, який можна налаштувати, щоб робити будь-що, але й продуктивно використовувати, навіть не торкаючись файлу конфігурації. Міжплатформне редагування Atom працює в різних операційних системах.

Розроблене програмне забезпечення для інформування про стан погоди має досить просту структуру. Запропоновано структурну схему програмного забезпечення для інформування про стан погоди. Основними компонентами, скриптами та модулями програми є: скрипт `user.js`, папка `node_modules`, папка `screenshot`, скрипт `getWeather.js`, файл `index.js`, скрипт `getForecast.js`, скрипт `keyboard.js`, файли `weatherCode.js`, `windDirection.js`, `app.js`, `index.js`.

Алгоритм роботи програми складається з трьох основних команд та має просту логіку. Робота з програмою розпочинається з команди `"/start"`. Далі є 3 команди для подальшої роботи з ботом: `"/help"`, `"/forecast"`, `"/where"`.

Функціонування програмного забезпечення для інформування про стан погоди полягає у відстеженні повідомлень від користувачів та надсилення прогнозу погоди за обраним містом.

Результати тестування програмного забезпечення для інформування про стан погоди підтвердили можливість його використання за призначенням.

16

Рисунок Д.16 – Слайд 16