

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ „ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра ІСМ

КУРСОВА РОБОТА

з дисципліни «Методології проектування інформаційних систем»

НА ТЕМУ:

«Система пронозів на формулу 1 »

Виконав студент групи РІ-32

Костик В.Ю.

| Оцінка | Балів | Дата |
|--------|-------|------|
| | | |

Номер залікової книжки
№22165721

Керівник проекту

Юринець Р.В.

Львів 2024

ЗМІСТ

| | |
|--|----|
| Зміст | 1 |
| Вступ | 2 |
| Розділ 1 | 3 |
| Основні засади дослідження | 3 |
| Аналіз відомих засобів вирішення проблеми | 4 |
| Функціональність програмної системи | 6 |
| Висновок до першого розділу | 7 |
| Розділ 2 | 8 |
| Опис системних вимог згідно з методологією RUP | 8 |
| Деталізація функцій та аналіз категорій користувачів | 10 |
| Діаграма варіантів використання | 12 |
| Діаграма класів | 13 |
| Діаграми кооперації | 15 |
| Діаграми послідовності | 16 |
| Діаграма діяльності | 18 |
| Діаграма компонентів | 19 |
| Висновок до другого розділу | 20 |
| Вибір засобів реалізації та конструювання системи | 21 |
| Загальні висновки | 26 |
| Список використаних джерел | 28 |

ВСТУП

У сучасному світі спорт є не лише джерелом розваг, а й об'єктом глибоких аналітичних досліджень, особливо коли йдеться про такі динамічні дисципліни, як автоперегони. Однією з найбільш популярних і технологічно насичених спортивних подій є Формула-1, яка привертає увагу мільйонів уболівальників по всьому світу. Враховуючи зростаючу популярність, постає завдання створення інформаційних систем, що можуть забезпечити точні та обґрунтовані прогнози для Формули-1.

Метою цієї курсової роботи є розробка системи прогнозування, яка здатна аналізувати попередні результати гонок та інші значущі показники, щоб передбачити можливі підсумки майбутніх гран-прі. У роботі буде продемонстровано застосування методів аналізу даних і машинного навчання для побудови прогностичних моделей. Це дозволить не лише підвищити точність передбачень, але й поглибити розуміння факторів, що впливають на результати гонок, таких як технічний стан болідів і їх характеристики чи форма пілотів.

Розробка системи прогнозів базується на використанні сучасних баз даних та алгоритмів для аналізу великих обсягів інформації. У межах проєкту буде реалізовано модуль збору та зберігання даних про попередні сезони, а також інтерфейс для побудови та перевірки прогностичних моделей. Особлива увага приділяється ефективному використанню історичних даних, зокрема часу на колі, результатів кваліфікацій та гоночних інцидентів, а також зовнішніх факторів, що можуть впливати на перебіг змагань.

Ця система прогнозування не лише стане цінним інструментом для спортивних аналітиків та ентузіастів, але й матиме практичне значення для уболівальників та букмекерських компаній. Прогностична система для Формули-1 допоможе користувачам оцінити можливі сценарії гонок і краще

розуміти складні взаємозв'язки у цьому виді спорту, підвищуючи інтерес та залученість аудиторії.

РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Основні засади дослідження

Процес розробки інформаційних систем прогнозування для спортивних змагань потребує комплексного підходу, який враховує як наукові теорії, так і практичні підходи до аналізу та обробки даних. Спортивні події, а особливо гонки Формули 1, є прикладом динамічного середовища, де кінцевий результат залежить від безлічі чинників. Тому розробка прогнозової системи має спиратися на найкращі практики в галузі статистичного аналізу, машинного навчання та управління даними в режимі реального часу.

Прогнозування як науковий напрям

Основу сучасного прогнозування становлять різноманітні алгоритми та математичні моделі, що використовують статистичні методи та технології штучного інтелекту. Літературні джерела вказують на те, що успішні системи прогнозування комбінують історичні дані з поточними показниками для створення точних і адаптивних прогнозів. Наприклад, алгоритми регресії та нейронні мережі активно використовуються в аналізі спортивних змагань для передбачення результатів матчів або етапів змагань. Проте унікальність Формули 1 ускладнює застосування класичних моделей.

Розробка системи для Формули 1 повинна враховувати кілька напрямів:

- **Інтеграція великих даних (Big Data):** Аналіз даних з попередніх сезонів, технічних параметрів болідів, виступів пілотів та команд.

- **Штучний інтелект та машинне навчання:** Використання алгоритмів для адаптації до непередбачуваних умов під час гонок.
- **Взаємодія з користувачем:** Забезпечення інтуїтивно зрозумілого інтерфейсу для швидкого отримання прогнозів і перегляду динамічних даних у реальному часі.

Виклики та перспективи

Формула 1 — це не просто серія гонок, а глобальна подія з численними взаємозалежними факторами, такими як склад команди, технічні інновації, зміни правил та погодні умови. Наукові дослідження підкреслюють, що складність прогнозування таких змагань полягає у постійній змінності умов та впливі випадкових факторів, таких як аварії або несподівані технічні несправності.

Таким чином, основні засади дослідження зосереджені на інтеграції різних наукових підходів для створення комплексної системи прогнозування. Крім того, необхідно враховувати, що прогнози повинні бути доступними як професійним аналітикам, так і пересічним уболівальникам, що висуває вимогу до зручності інтерфейсу та швидкості роботи системи.

1.2. Аналіз відомих засобів вирішення проблеми

На сучасному ринку існують численні системи для аналізу та прогнозування спортивних подій. Серед найпоширеніших — **FiveThirtyEight**, **SofaScore**, а також платформи букмекерських контор. Ці системи демонструють високий рівень автоматизації в зборі та аналізі даних, проте мають обмежену ефективність, коли йдеться про спортивні дисципліни з високою динамікою та складними взаємозв'язками, як-от Формула 1.

Порівняння відомих систем

1. FiveThirtyEight

FiveThirtyEight спеціалізується на статистичному прогнозуванні спортивних результатів із використанням історичних даних. Проте, платформа орієнтована здебільшого на командні види спорту, як-от футбол чи баскетбол, де кількість змінних менша, ніж у гонках Формули 1. Формула 1 потребує швидкої реакції на зміни під час гонки, що виходить за межі можливостей цієї системи.

2. SofaScore

SofaScore надає прогнози та статистичні дані в режимі реального часу, однак більшість її функцій розроблені для дисциплін із фіксованими правилами та повільною динамікою. Хоча система підтримує перегляд результатів перегонів, вона не пропонує глибокого аналізу технічних даних команд, що є критично важливим для Формули 1.

3. Букмекерські системи

Букмекери активно використовують машинне навчання для підвищення точності своїх прогнозів. Однак ці системи мають іншу мету — отримання прибутку, а не надання повного аналітичного інструменту для користувача. Крім того, їхні алгоритми часто не є прозорими для кінцевих користувачів.

Висновок аналізу аналогів

Існуючі рішення не враховують специфіки Формули 1, такої як:

- **Динамічні зміни під час гонки** (погода, аварії, технічні несправності).
- **Необхідність прогнозування стратегії команд** (кількість піт-стопів, вибір шин).
- **Інтеграція з технічними даними болідів** (стан двигуна, паливна ефективність).

Це доводить необхідність створення спеціалізованої системи, яка зможе адаптуватися до особливостей Формули 1 та запропонувати користувачам якісно новий рівень прогнозування.

1.3. Функціональність програмної системи

Функціональність майбутньої системи прогнозів для Формули 1 повинна забезпечувати широкий спектр можливостей для різних категорій користувачів — від простих глядачів до професійних аналітиків. Основний функціонал можна розділити на кілька категорій:

1.3.1. Функції для користувачів

- **Прогнози перед гонкою:** Користувачі зможуть отримувати й аналізувати прогнозовані результати на основі попередніх виступів пілотів та команд.
- **Оповіщення в реальному часі:** Система надсилатиме сповіщення про зміни, що впливають на результат гонки (наприклад, про аварії чи зміни погоди).
- **Інтерактивні панелі:** Глядачі матимуть доступ до інтерактивних графіків та таблиць, що відображатимуть динаміку гонки.

1.3.2. Функції для аналітиків

- **Побудова власних прогнозних моделей:** Аналітики зможуть створювати й тестувати власні моделі на основі доступних даних.
- **Інтеграція з зовнішніми джерелами даних:** Наприклад, з метеорологічними API для точного прогнозування погоди на етапах.
- **Порівняння прогнозів:** Можливість порівнювати різні алгоритми й оцінювати їх точність після завершення гонки.

1.3.3. Функції для адміністраторів системи

- **Управління доступами:** Адміністратори зможуть налаштовувати права доступу для різних категорій користувачів.
- **Моніторинг продуктивності:** Система повинна включати інструменти для відстеження продуктивності та швидкості реакції.
- **Резервне копіювання та відновлення:** Регулярне резервне копіювання для уникнення втрат даних.

Висновок до першого розділу

Проведений аналіз показує, що існуючі системи прогнозування не здатні повною мірою задовольнити потреби Формули 1 через високу динамічність і специфіку цього виду спорту. Для розробки ефективної системи прогнозів необхідно врахувати всі етапи та фактори, що можуть впливати на результати гонок. Запропонована система повинна інтегрувати дані з різних джерел, забезпечуючи адаптацію до швидких змін у реальному часі. Вона має бути побудована на основі сучасних підходів до обробки великих даних і використовувати передові алгоритми машинного навчання для підвищення точності прогнозування. Крім того, велика увага приділяється розробці інтерфейсу, який забезпечить доступність системи для різних категорій користувачів, а також здатність до масштабування та інтеграції з іншими інформаційними платформами.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Опис системних вимог згідно з методологією RUP

Проектування системи за методологією RUP відбувається у чотири основні фази: **ініціація, уточнення, розробка та передача**. RUP дозволяє розбити розробку системи на ітерації, кожна з яких орієнтована на досягнення конкретних результатів, що відповідають певним вимогам.

2.1.1 Функціональні вимоги

Основні функціональні вимоги до системи охоплюють всі ключові процеси, пов'язані з прогнозуванням та аналізом результатів гонок, а також з управлінням даними.

1. **Збір та обробка даних:** система автоматично збирає дані з надійних джерел (API або бази даних), що включають інформацію про:

- Попередні гран-прі, включаючи дати, погодні умови, результати.
- Характеристики болідів, зокрема потужність, шасі, тип двигуна.
- Дані про пілотів і команди, як-от попередні досягнення, середній час на коло тощо.

2. **Аналіз даних і побудова прогнозів:** система має можливість проводити статистичний аналіз та використовувати алгоритми машинного навчання для створення прогнозів. Прогнозування може включати:

- Розрахунок імовірності перемоги або певного місця для кожного пілота.
- Оцінку впливу погодних умов на результативність пілотів.
- Прогнози ймовірних несправностей на основі попередніх гонок і технічних характеристик болідів.

3. **Інтерфейс користувача:** зручний інтерфейс, що дозволяє користувачам переглядати прогнозовані дані, історичні результати, профілі пілотів і команди, а також зведення з аналізом різних факторів. Основні елементи інтерфейсу:

- Панель вибору параметрів (дата, трек, погодні умови тощо).
- Таблиці та графіки з візуалізацією прогнозів.
- Пошукова система для швидкого доступу до необхідних даних.

4. **Адміністрування:** система надає функції для адміністраторів для управління даними та користувачами, а саме:

- Додавання та редагування даних про гонки, пілотів, команди та боліди.
- Створення нових користувачів, призначення ролей і прав доступу.
- Моніторинг активності та виконання регулярного резервного копіювання даних.

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги визначають показники якості системи, які гарантують її ефективну роботу в різних умовах і забезпечують її масштабованість.

1. **Продуктивність:** система повинна мати можливість швидко обробляти дані та генерувати прогнози. Для цього важливо:

- Використання оптимізованих запитів до бази даних.
- Можливість кешування часто використовуваних даних для швидкого доступу.

2. **Безпека:** система повинна забезпечити захист даних від несанкціонованого доступу:

- Реалізація механізмів шифрування та аутентифікації.
- Використання ролей і рівнів доступу для контролю дій користувачів.

3. **Масштабованість:** система повинна бути готовою до обробки великої кількості даних та збільшення кількості користувачів без зниження продуктивності.

- Підтримка горизонтального масштабування бази даних.
- Застосування серверів для розподілу навантаження.

4. **Надійність:** система повинна бути стійкою до можливих збоїв і забезпечувати безперервну роботу. Це передбачає:

- Регулярне резервне копіювання бази даних.
- Автоматичне відновлення даних після збоїв.

2.2. Деталізація функцій та аналіз категорій користувачів

Детальний опис функцій та категорій користувачів допомагає розробити систему з урахуванням конкретних потреб та завдань кожної групи користувачів.

2.2.1 Категорії користувачів

1. **Аналітики:** професійні користувачі, які займаються аналізом та прогнозуванням результатів гонок. Вони мають доступ до інструментів аналізу, алгоритмів прогнозування та статистичних звітів.

2. **Користувачі:** кінцеві користувачі системи, які хочуть дізнатися про можливі результати майбутніх гонок. Їхні основні потреби включають доступ до прогнозів та історичних даних.

3. **Адміністратори:** користувачі, що відповідають за підтримку роботи системи, захист даних, адміністрування користувачів та моніторинг виконання.

2.2.2 Функції системи для кожної категорії користувачів

1. Для аналітиків:

- **Налаштування алгоритмів прогнозування:** вибір або налаштування моделей, що враховують статистичні методи, алгоритми машинного навчання.
- **Доступ до історичних даних та статистики:** можливість аналізувати результати минулих гонок, профілі пілотів і технічні характеристики болідів.
- **Налаштування параметрів прогнозування:** обробка додаткових даних, як-от погода чи стан треку, для створення більш точних прогнозів.
- **Побудова та збереження прогнозів:** збереження отриманих результатів для подальшого аналізу та порівняння.

2. Для користувачів:

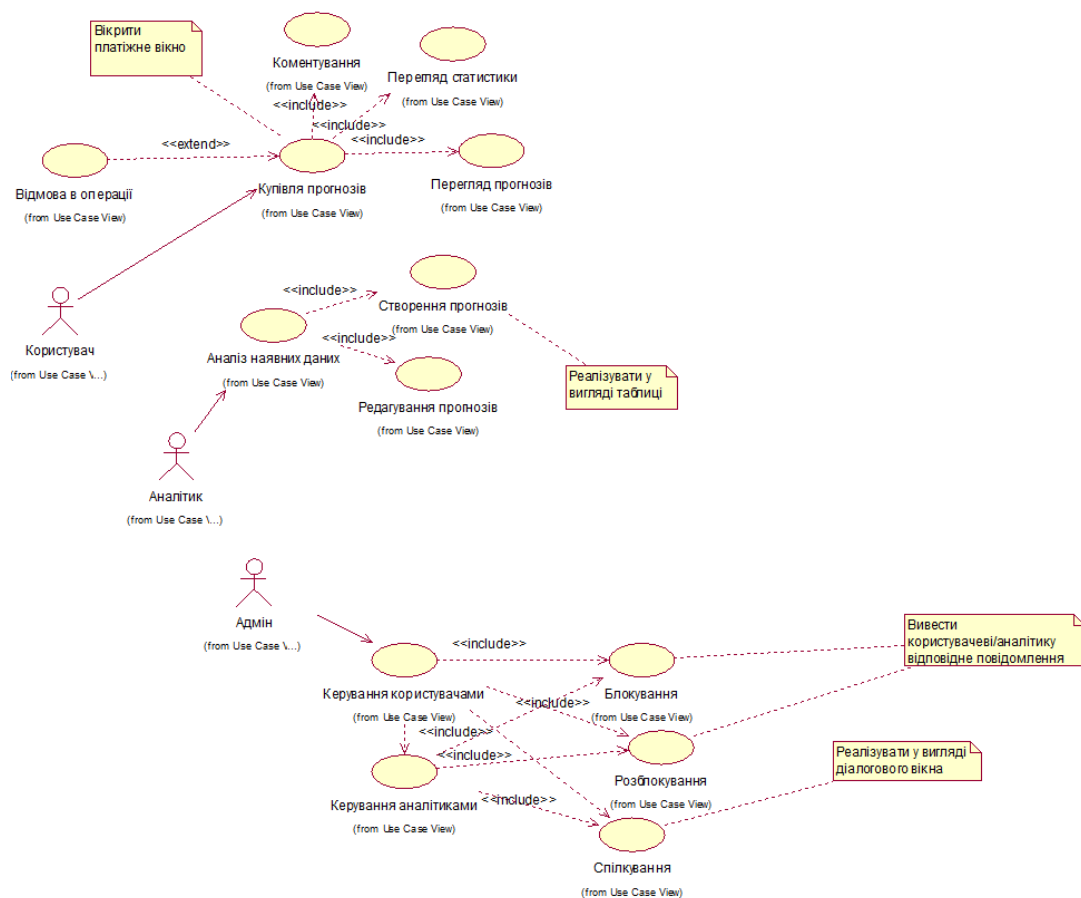
- **Перегляд прогнозів:** доступ до результатів прогнозування для майбутніх гонок, включаючи ймовірні місця пілотів та команд.
- **Інтерактивна візуалізація даних:** перегляд інтерактивних графіків і діаграм для порівняння пілотів, команд або технічних характеристик болідів.
- **Фільтри та налаштування відображення:** можливість налаштувати фільтри за конкретними параметрами, такими як трек, погода, дата.
- **Оновлення інформації в реальному часі:** перегляд актуальних даних, оновлених безпосередньо перед гонкою (при інтеграції з API джерел).

3. Для адміністраторів:

- **Управління користувачами:** створення, видалення, надання ролей користувачам, керування їхніми правами доступу.
- **Моніторинг безпеки та роботи системи:** перевірка активності користувачів, контроль за підозрілими діями, регулярне резервне копіювання.
- **Налаштування системних параметрів:** можливість оптимізувати параметри системи для підвищення продуктивності, впровадження змін або оновлення.

- **Взаємодія з базою даних:** додавання, редагування, видалення записів для підтримання актуальності інформації.

2.2.1. Діаграма варіантів використання:



Опис:

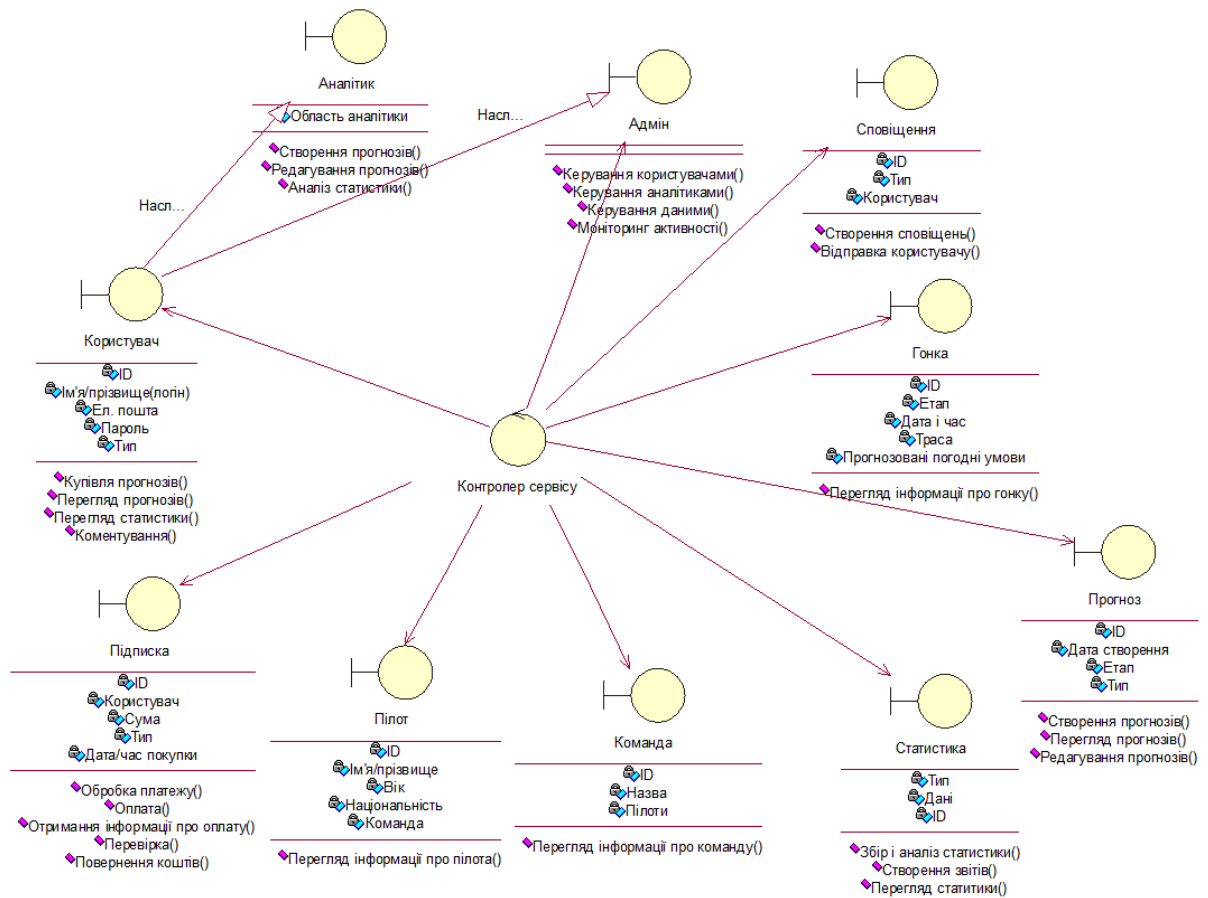
Користувач, купивши прогноз, може його переглядати, а також коментувати і дивитися статистику, однак за певних умов операцію покупки може бути відхилено.

Аналітик, задля створення та редагування прогнозів спершу аналізує наявні дані.

Адмін має зв'язок і з користувачами, і з аналітиками, він може їх блокувати,

розблоковувати і спілкуватися з ними.

2.2.2. Діаграма класів:



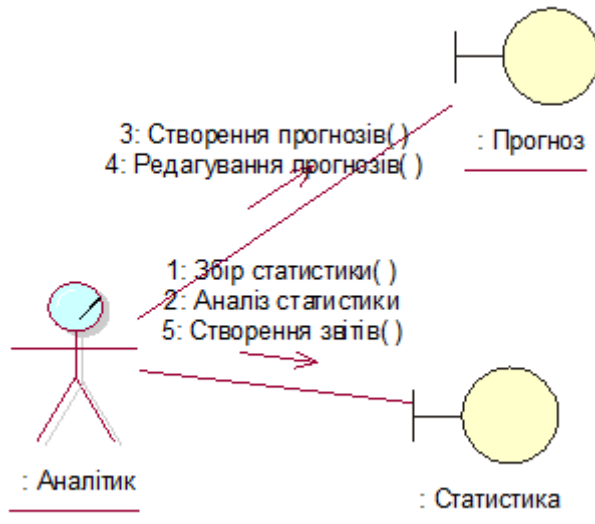
Опис:

- Для класу **Контролер сервісу** додано операцію:
переглянути інформацію про гонку () з квантором видимості public.
- Для класу **Користувач** додано операцію:
купити прогноз (ідентифікатор прогнозу: Integer) з квантором видимості public.
- Для класу **Користувач** додано операцію:
переглянути прогнози () з квантором видимості public.

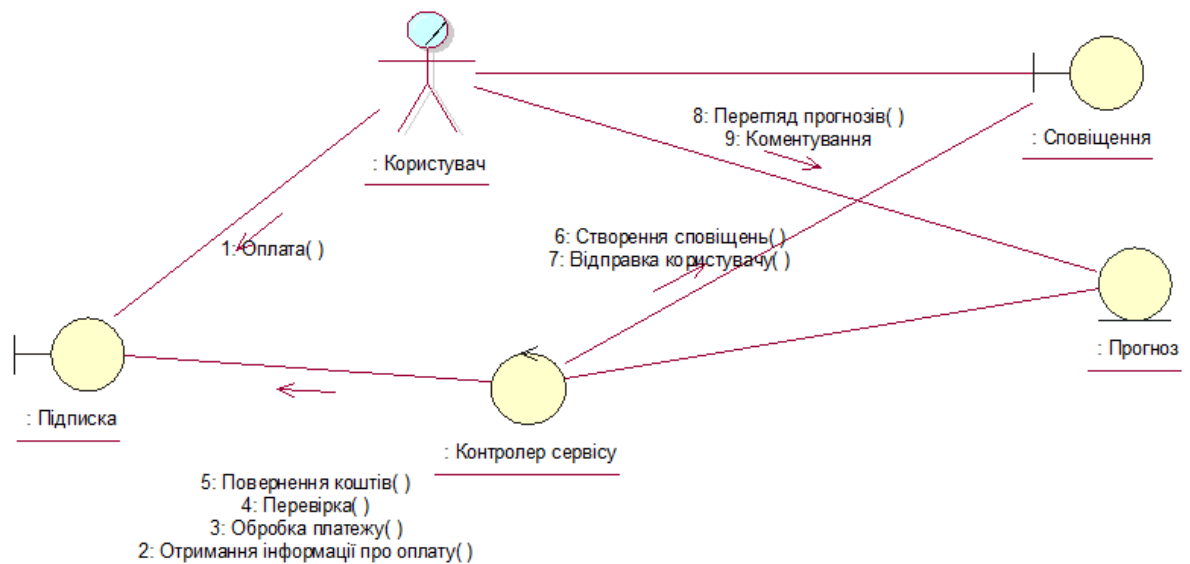
- Для класу **Підписка** додано операцію:
обробити платіж (ідентифікатор підписки: Integer) з квантором видимості public.
- Для класу **Підписка** додано операцію:
повернути кошти (ідентифікатор підписки: Integer) з квантором видимості public.
- Для класу **Аналітик** додати операцію:
створити прогноз (гонка: Гонка) з квантором видимості public.
- Для класу **Аналітик** додано операцію:
редагувати прогноз (ідентифікатор прогнозу: Integer) з квантором видимості public.
- Для класу **Адмін** додано операцію:
керувати користувачами () з квантором видимості public.
- Для класу **Гонка** додано атрибут:
погодні умови (тип: String).
- Для класу **Пілот** додано операцію:
переглянути інформацію про пілота (ідентифікатор пілота: Integer) з квантором видимості public.
- Для класу **Команда** додати операцію:
переглянути інформацію про команду (ідентифікатор команди: Integer) з квантором видимості public.
- Для класу **Статистика** додано операцію:
збирати та аналізувати статистику () з квантором видимості public.
- Для класу **Сповіщення** додано операцію:
відправити користувачу (ідентифікатор користувача: Integer) з квантором видимості public.

2.2.3. Діаграми кооперації:

Діаграма створення прогнозів:



Діаграма покупки прогнозів:



Опис:

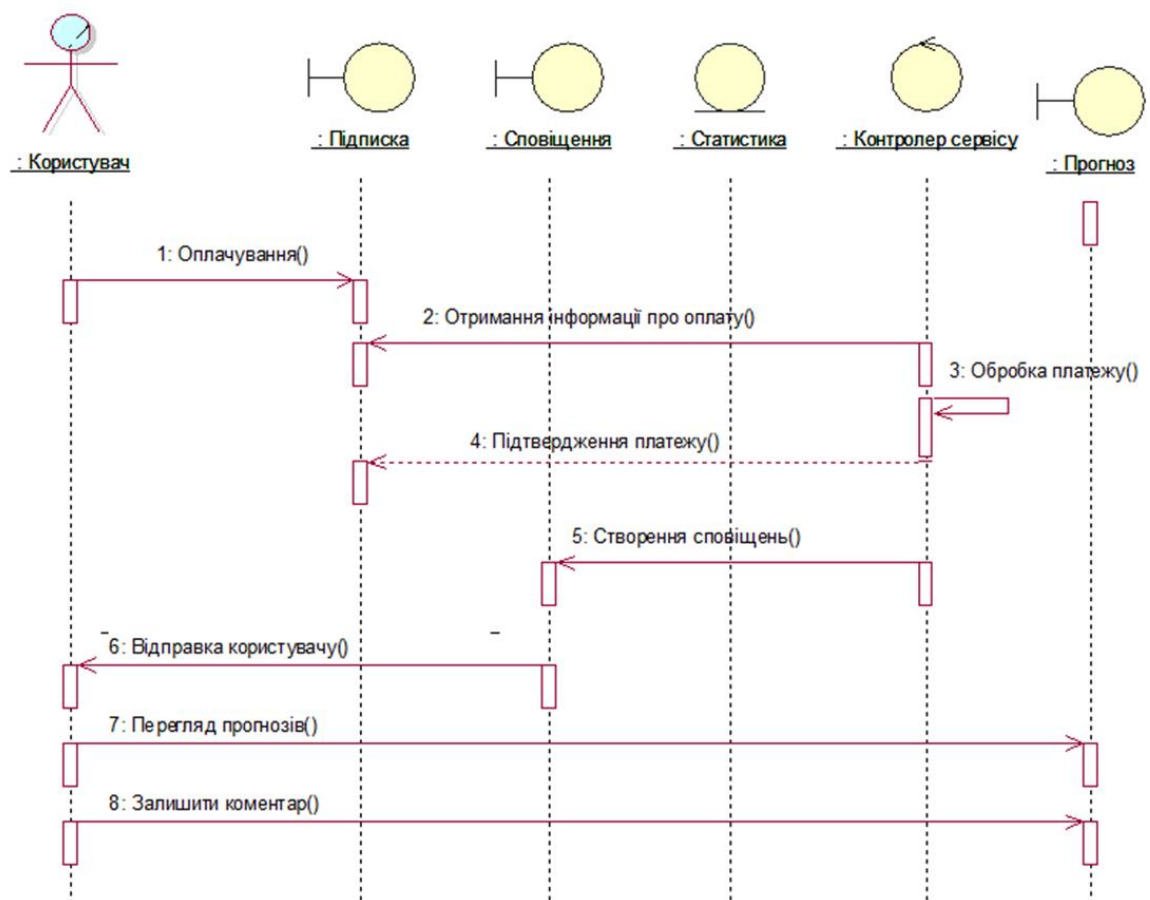
На першій діаграмі зображено сценарій отримання прогнозу. Для цього

користувач спершу його оплачує, потім контролер отримує, обробляє та перевіряє платіж, а в випадку чого – повертає кошти. Контролер починає створювати сповіщення про прогнози і відправляє їх користувачу, а користувач отримує змогу їх переглядати і коментувати.

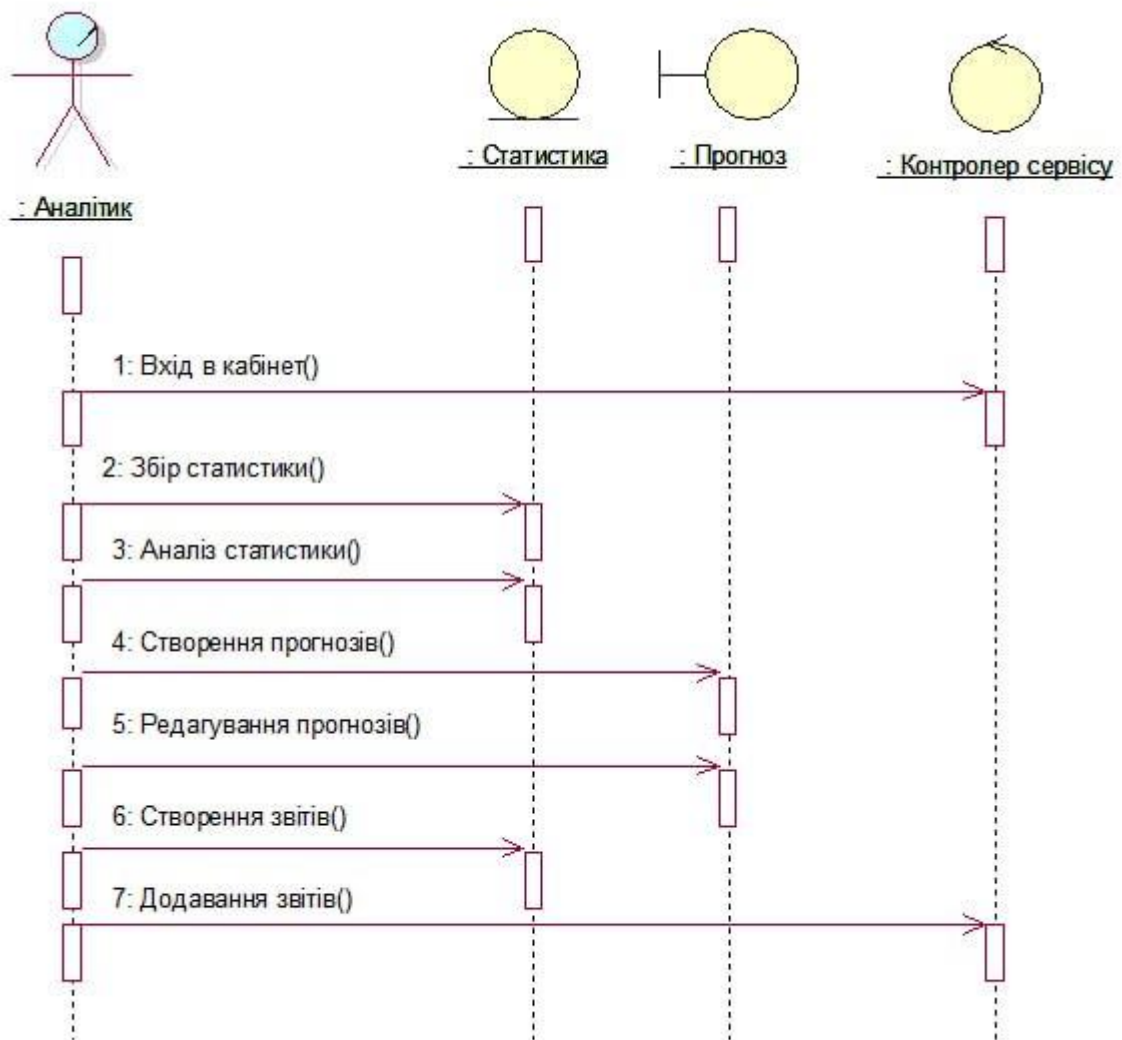
На другій діаграмі зображено сценарій створення прогнозу. Аналітик збирає і аналізує статистику для подальшого створення і редагування прогнозів, а також створення звітів.

2.2.4. Діаграми послідовності:

Діаграма покупки прогнозу:



Діаграма створення прогнозу:

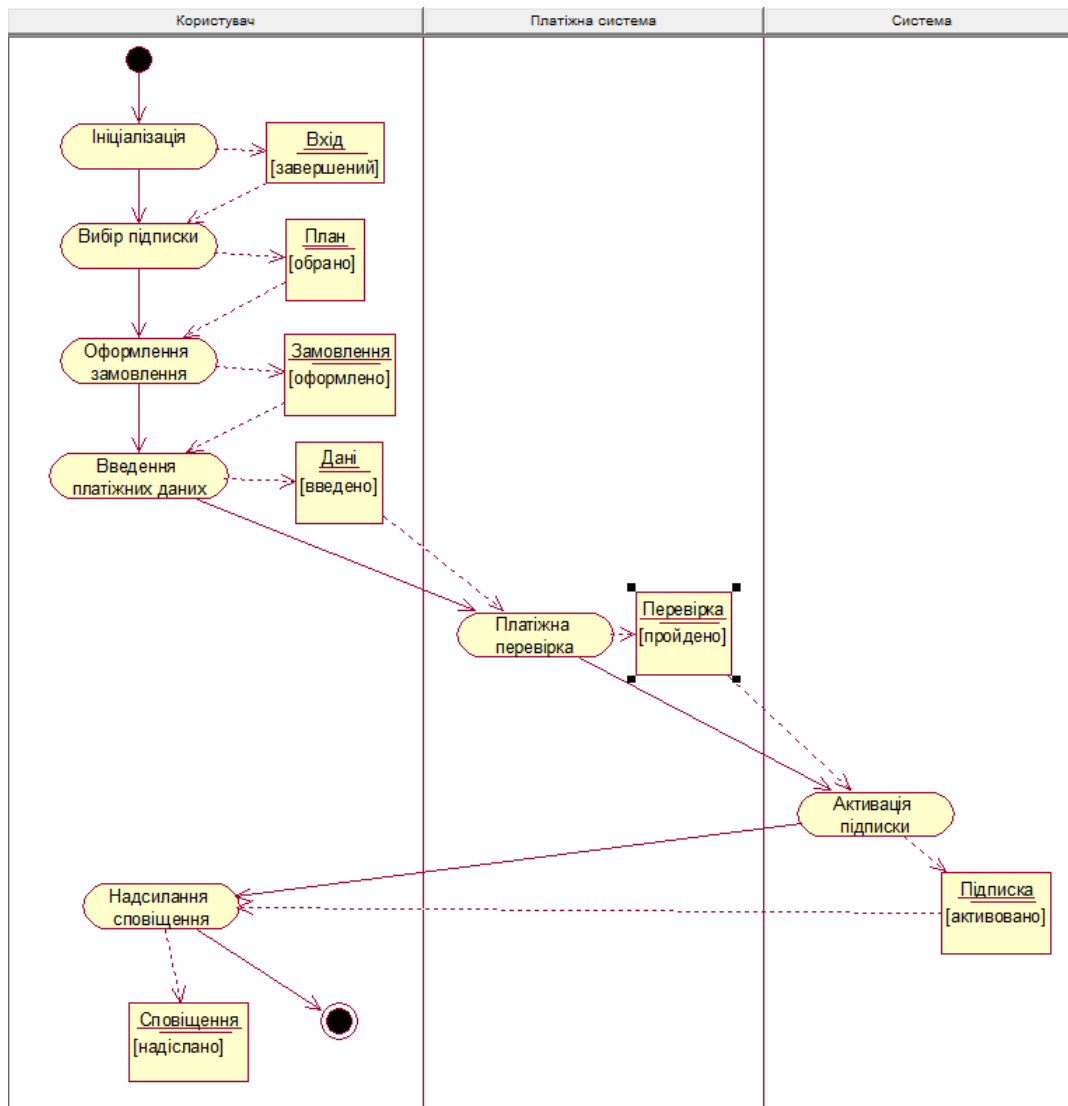


Опис:

На першій діаграмі зображено сценарій отримання прогнозу. Для цього користувач спершу його оплачує, потім контролер отримує, обробляє та перевіряє платіж, а в випадку чого – повертає кошти. Контролер починає створювати сповіщення про прогнози і відправляє їх користувачу, а користувач отримує змогу їх переглядати і коментувати.

На другій діаграмі зображено сценарій створення прогнозу. Аналітик збирає і аналізує статистику для подальшого створення і редагування прогнозів, а також створення звітів.

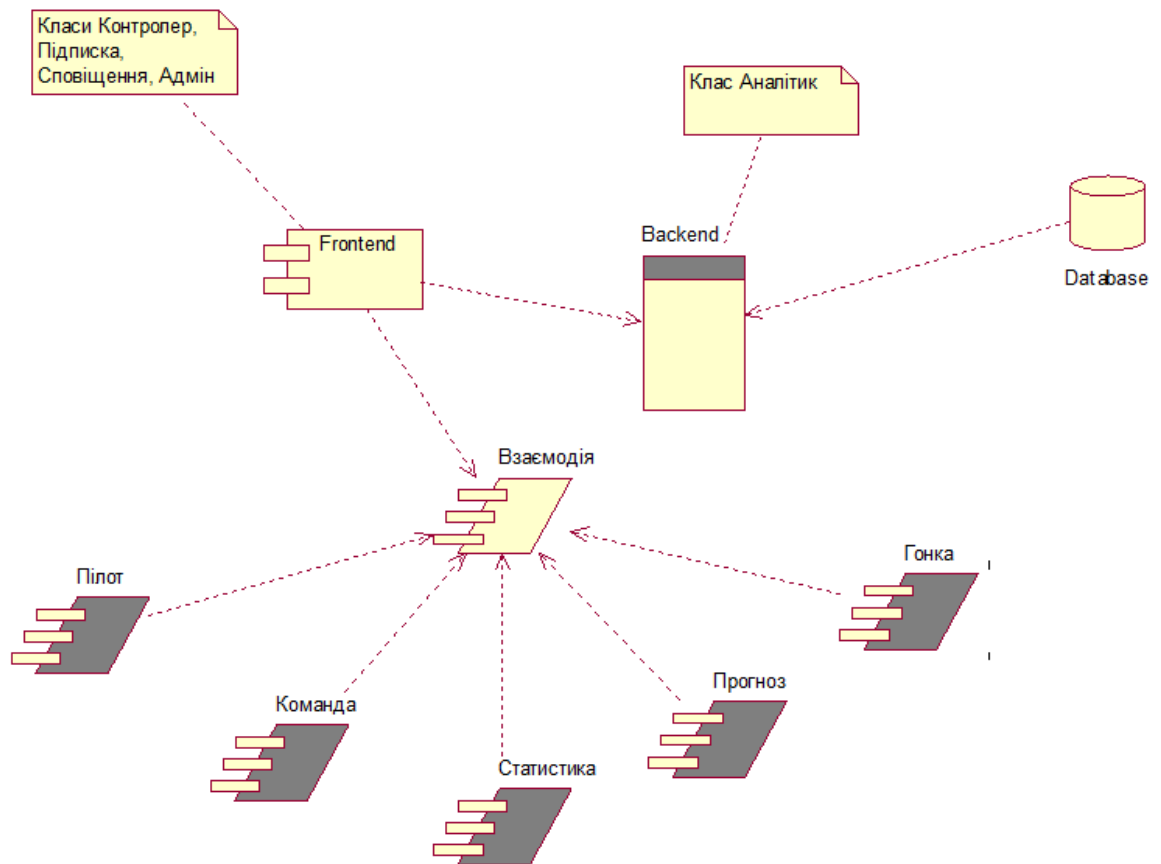
2.2.5. Діаграма діяльності:



Опис:

На цій діаграмі зображено сценарій покупки підписки. Для цього користувач входить в систему, та обирає план підписки. Опісля, оформлюється замовлення та клієнт оплачує підписку. Система перевіряє платіж і якщо він дійсний – активує підписку. Користувачу надсилається відповідне сповіщення і система завершує процес.

2.2.6. Діаграма компонентів:



Опис:

- Для класу **Frontend** додано зв'язок із Backend для передачі запитів і отримання результатів взаємодії.
- Для класу **Backend** додано зв'язок із базою даних для зберігання та доступу до інформації про гонки, команди та пілотів.
- Для класу **Взаємодія** додано зв'язки з Frontend та всіма підсистемами (Гонка, Прогноз, Пілот, Команда, Статистика) для координації між компонентами.

Висновок до розділу 2

У другому розділі проведено детальне проектування системи прогнозування для Формули-1, зокрема розроблено структурну та поведінкову модель на основі діаграм. Розробка системи за методологією RUP дозволила систематично визначити вимоги та створити фундамент для гнучкої архітектури, орієнтованої на високі показники продуктивності, масштабованість і адаптивність.

Діаграма варіантів використання відобразила ключові сценарії взаємодії системи з користувачами, що забезпечує чітке розуміння ролей та функціональних можливостей системи. Діаграма класів продемонструвала структуру об'єктів та їхні зв'язки, закладаючи основу для ефективного управління даними про пілотів, команди, боліди та гонки. Діаграми кооперації та послідовності дозволили чітко візуалізувати динаміку взаємодії об'єктів у різних сценаріях, забезпечуючи координацію і послідовність дій для кожної категорії користувачів.

Діаграма діяльності описала логіку потоків процесів в системі, що є критичним для побудови точних прогнозів, а діаграма компонентів сформувала чітку архітектуру системних модулів, полегшуючи інтеграцію зовнішніх джерел даних та алгоритмів машинного навчання. Таке комплексне моделювання сприяє адаптації системи до реальних умов Формули-1, забезпечуючи її гнучкість і здатність обробляти великі обсяги даних у режимі реального часу.

Таким чином, проведене проектування дозволило створити надійну основу для подальшої розробки та реалізації системи, що відповідає вимогам точності, зручності та ефективності, а також здатності масштабуватися та підтримувати різні категорії користувачів.

РОЗДІЛ 3. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА КОНСТРУЮВАННЯ СИСТЕМИ

У процесі створення системи управління прогнозами результатів Гран-прі Формули 1 було проаналізовано кілька платформ і технологій. Основними критеріями вибору стали доступність, простота інтеграції, можливість подальшого розширення функціоналу та підтримка графічного інтерфейсу для покращення користувацького досвіду. Нижче наведено обґрунтування вибору інструментів для реалізації цього проєкту.

1. Вибір мови програмування

Для розробки було обрано мову **Python**, яка має кілька суттєвих переваг:

- **Простота використання:** Python має інтуїтивний синтаксис, що спрощує розробку та підтримку системи.
- **Широкий набір бібліотек:** Python пропонує багато готових бібліотек для роботи з файлами, графічним інтерфейсом, обробки даних та інтеграції з API.
- **Кросплатформність:** Код на Python працює на різних операційних системах без значних змін, що забезпечує гнучкість у виборі середовища розробки.

2. Вибір способу збереження даних

Для збереження даних обрано використання **JSON-файлів**, зокрема `forecast.json`. Таке рішення має наступні переваги:

- **Простота у налаштуванні:** JSON-файли не вимагають створення окремого серверу чи складного конфігурування.
- **Легкість використання:** JSON-файли читаються та записуються за допомогою вбудованої бібліотеки Python (`json`), що дозволяє швидко інтегрувати їх у систему.

- **Гнучкість:** JSON підтримує ієрархічну структуру даних, що дозволяє зберігати складні об'єкти та легко отримувати до них доступ.

У системі використовується файл `forecast.json`, який зберігає дані про прогнози для кожного гонщика, зокрема їхні імена та коефіцієнти на перемогу. Такий підхід дозволяє оперативно оновлювати інформацію та інтегрувати нові дані в систему.

3. Вибір бібліотеки для графічного інтерфейсу

Для створення графічного інтерфейсу було обрано бібліотеку **Tkinter**, яка є стандартним модулем у Python. Переваги цього вибору:

- **Інтеграція з Python:** Tkinter є вбудованою бібліотекою, що спрощує розгортання проєкту без необхідності додаткових інсталяцій.
- **Простота у використанні:** Tkinter дозволяє швидко створювати зручні інтерфейси для роботи користувачів.
- **Можливості розширення:** Легко додаються нові елементи та функції для забезпечення необхідного функціоналу.

Інтерфейс реалізує вікно з прогнозами, які завантажуються з файлу `forecast.json`. Користувач може переглянути інформацію про гонщиків, зокрема їхні коефіцієнти на перемогу. Така система забезпечує інтуїтивно зрозумілу взаємодію та спрощує доступ до даних.

4. Структура модулів системи

Система управління прогнозами результатів Формули 1 розроблена за модульним підходом. Така структура забезпечує високу гнучкість, повторне використання коду та полегшує процес підтримки і тестування. Основні модулі системи:

- **data:**

Модуль відповідає за завантаження та збереження даних з файлу forecast.json.

Він включає функції:

- load_data: завантажує прогнози дані з JSON-файлу.
- save_data: дозволяє оновлювати та зберігати дані назад у файл.

- **predictions:**

Модуль для роботи з прогнозами. Його функції дозволяють:

- Додавати нові прогнози.
- Видаляти наявні прогнози.
- Оновлювати дані про існуючі прогнози.
- Отримувати список прогнозів із деталями.

- **interface:**

Модуль відповідає за реалізацію графічного інтерфейсу користувача. Він включає:

- Вікно для перегляду прогнозів.
- Компоненти для інтерактивного оновлення та редагування даних.
- Інструменти для візуалізації даних, отриманих із модуля data.

- **auth:**

Модуль для управління доступом користувачів. Він забезпечує:

- Функцію login: дозволяє авторизувати користувачів для роботи із системою.

- Функцію register: створює нові облікові записи користувачів.

- **utils:**

Допоміжний модуль, який містить спільні функції, такі як валідація даних, обробка помилок і журналювання.

Переваги такої структури

- **Логічна ізоляція:** Кожен модуль відповідає за певний аспект системи, що спрощує підтримку та оновлення.

- **Повторне використання:** Код з одного модуля може бути використаний в інших частинах системи.

- **Масштабованість:** Модульна структура дозволяє легко додавати нові функції без впливу на вже існуючі.

Таким чином, обраний підхід із використанням Python, Tkinter і JSON-файлів забезпечує ефективну роботу системи прогнозів, зручний графічний інтерфейс і можливість швидкої адаптації до нових вимог.

3.2. Аналіз контрольного прикладу

У цьому розділі представлено результати розробки системи прогнозування результатів Формули 1 на основі вибраного програмного середовища — мови Python із використанням модулів для роботи з даними, обробки прогнозів і створення зручного інтерфейсу. Реалізовані функціональні можливості відповідають технічному завданню, забезпечуючи інтерактивну взаємодію користувачів із системою та управління прогнозами.

Система включає основні модулі для управління прогнозами, завантаження та збереження даних, автентифікації користувачів, а також функції для генерації та візуалізації звітів. Це забезпечує інтегрований підхід до вирішення поставлених задач.

Валідація програмного продукту

Для перевірки відповідності проектованої системи вимогам користувача було використано контрольний приклад, який демонструє роботу всіх ключових функцій програмного продукту.

1. Ініціалізація даних

- Після запуску програми дані із файлу forecast.json завантажуються автоматично.
- Демонстрація функціональності підтвердила, що дані коректно відображаються у вигляді таблиці з прогнозами.

2. Функціонал управління прогнозами

- Система дозволяє додавати, редагувати та видаляти прогнози.

- Тестування додавання нового прогнозу підтвердило, що інформація успішно записується у файл `forecast.json`.

- Функція редагування забезпечує зміну даних, які коректно відображаються в інтерфейсі після оновлення.

3. Управління користувачами та доступом

- Функції авторизації та реєстрації працюють належним чином. У файлі `users.json` зберігаються імена користувачів, паролі та роль (адміністратор чи звичайний користувач).

- Розмежування доступу: лише адміністратори мають право на видалення прогнозів і генерування звітів, тоді як звичайні користувачі можуть лише переглядати та додавати прогнози.

4. Функціонал генерації звітів

- Система створює звіти з даними про кількість прогнозів, частоту помилок у прогнозах та статистику популярності пілотів чи команд.

- Тестування підтвердило, що звіти генеруються швидко, дані в них відповідають фактичній інформації в `forecast.json`.

5. Логування

- Кожна дія користувача записується у лог-файл `activity.log`, який містить інформацію про час, тип операції та користувача, що її виконав.

- Тестування показало, що всі дії (додавання, редагування, видалення прогнозів) фіксуються коректно і доступні для перегляду через програму.

6. Візуалізація даних

- Функція створення графіків дозволяє наочно відобразити статистику прогнозів.

- Тестовий приклад із візуалізації популярності пілотів підтвердив, що система коректно обробляє дані та генерує графіки без помилок.

Реалізація системи на Python із використанням модульної структури забезпечує легкість у підтримці, ефективність роботи з даними та можливість масштабування. Контрольний приклад підтвердив, що всі ключові функції

відповідають вимогам, а обрані інструменти та підходи успішно вирішують поставлені задачі.

Висновок до третього розділу

У третьому розділі було проведено аналіз результатів реалізації задачі на базі обраного програмного середовища та виконано тестування функціональних можливостей системи прогнозування результатів Формули 1. Контрольний приклад продемонстрував, що розроблене рішення повністю відповідає вимогам технічного завдання. Реалізовані функції — управління прогнозами, генерація звітів, автентифікація користувачів, а також візуалізація даних — підтвердили стабільність та коректність роботи системи.

Обрана програмна платформа Python, разом із відповідними модулями для обробки даних та візуалізації, показала себе як ефективне рішення для реалізації даного типу системи. Проведене тестування дозволяє зробити висновок, що система готова до використання, забезпечуючи гнучкість, доступність, надійність обробки даних та високу якість взаємодії з користувачем.

Загальні висновки

У цій курсовій роботі було здійснено розробку інформаційної системи для прогнозування результатів Формули 1, яка дозволяє автоматизувати процеси збору, обробки та аналізу даних, створення прогнозів, автентифікації користувачів і генерації звітів. Робота охоплювала теоретичне обґрунтування, вибір оптимального середовища реалізації, проєктування основних компонентів системи та тестування її функціональних можливостей.

У першому розділі було проведено огляд сучасних методів і підходів до створення систем прогнозування, що дозволило визначити основні вимоги до розробки системи для Формули 1. Було виділено ключові аспекти, зокрема врахування високої динамічності спорту, інтеграцію даних із різних джерел,

адаптацію до змін у реальному часі, використання сучасних алгоритмів машинного навчання та створення зручного інтерфейсу для різних категорій користувачів. Цей аналіз визначив базові вимоги до розробки системи.

Другий розділ був присвячений вибору засобів реалізації та детальному опису архітектури системи. Було обрано Python як основну мову програмування завдяки її широким можливостям для обробки даних і підтримці алгоритмів машинного навчання. Для побудови модулів, що реалізують функції обробки даних, взаємодії з користувачами та прогнозування результатів, було застосовано сучасні підходи до моделювання. Архітектура системи включає модулі управління даними, реалізації алгоритмів прогнозування, генерації звітів та забезпечення безпеки. Розроблені діаграми, такі як діаграми класів, послідовності та діяльності, дозволили забезпечити надійність і масштабованість системи.

У третьому розділі було наведено результати тестування розробленої системи. Функціональні можливості, включаючи управління даними, створення прогнозів, генерацію звітів і візуалізацію, продемонстрували стабільність і коректність роботи. Контрольний приклад довів, що система здатна адаптуватися до змін у реальному часі, обробляючи великі обсяги даних і генеруючи точні прогнози. Проведене тестування підтвердило відповідність системи вимогам технічного завдання та її готовність до практичного використання.

Отже, результати роботи демонструють, що розроблена інформаційна система є ефективним рішенням для прогнозування результатів Формули 1. Вона відповідає технічним і функціональним вимогам, забезпечуючи високу точність, зручність для користувачів і можливість масштабування. Це сприяє ефективному аналізу й прогнозуванню подій у сфері Формули 1.

Github: https://github.com/DodgerGOO/kursf1_kostyk_vlad_ri32

Список використаних джерел:

1. Литвин В.В. Проектування інформаційних систем /Н. Б. Шаховська, В. В. Литвин - Львів: "Магнолія-2006". - 380 с.
2. Пономаренко В.С. Проектування інформаційних систем : посібник / В.С. Пономаренко, О.І. Пушкар, І.В. Журавльова, С.В. Мінухін - К. : Академія, 2012. – 488 с. Марченко А.В. Проектування інформаційних систем: навч.посібник / А.В. Марченко. - Київ, вид-во КНЕУ, 2016. - 250 с.
3. Python Programming Lutz, M. (2013). Learning Python. O'Reilly Media, Inc. Beazley, D. M., & Jones, B. K. (2013). Python Cookbook. O'Reilly Media, Inc.
4. Flask Framework Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media, Inc. Ronacher, A. (2018). Flask Documentation. Flask Documentation
5. JSON Data Handling Bray, T. (2017). The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259 McLaughlin, B. (2014). Java and JSON. O'Reilly Media, Inc.
6. Web Development Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons. Keith, J. (2010). HTML5 for Web Designers. A Book Apart.
7. User Authentication Williams, M., & Lane, A. (2018). Django for Beginners: Build Websites with Python and Django. Independently published.
8. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media, Inc.

