

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПРОГНОЗУВАННЯ ПІД ЧАС ПЛАНУВАННЯ РОБОТИ ПРОКАТУ
ВЕЛОСИПЕДІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
BICYCLE RENTAL SERVICE PLANNING

Виконав: студент(ка) 2 курсу, групи КНТ-110м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Рябчун Д.О.

(прізвище та ініціали)

Керівник Льовкін В.М.

(прізвище та ініціали)

Рецензент Терещенко Е.В.

(прізвище та ініціали)

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

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Рябчуна Дмитра Олександровича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів прогнозування під час планування роботи прокату велосипедів
Research and Software Implementation of Methods for Bicycle Rental Service Planning

керівник проєкту (роботи) Льовкін Валерій Миколайович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “5” листопада 2021 року №435

2. Строк подання студентом проєкту (роботи) 13 грудня 2021 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Основні рішення щодо реалізації компонентів системи. 5. 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
1-5 Основна частина	Льовкін В.М., доцент		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “7” вересня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз та дослідження предметної області	1-2 тиждень	
2	Розробка та затвердження технічного завдання	3 тиждень	Технічне завдання
3	Дослідження методів розв’язання проблеми	4 тиждень	
4	Проектування програмного забезпечення	5 тиждень	
5	Розроблення структури програми та модулів	6-7 тиждень	Структурна схема
6	Розроблення програмного коду	8 тиждень	Текст програми
7	Тестування та відлагодження програмного забезпечення	9 тиждень	Програма
8	Розробка розділів пояснювальної записки	10-11 тиждень	Пояснювальна записка
9	Розробка графічного матеріалу	12 тиждень	Графічний матеріал
10	Захист магістерської роботи	16 тиждень	

Студент(ка)

(підпис)

Рябчун Д.О.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Льовкін В.М.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 94 с., 31 рис., 3 табл., 3 дод., 20 джерел.

ПРОКАТ ВЕЛОСИПЕДІВ, ПРОГНОЗУВАННЯ ПОПИТУ, ПЛАНУВАННЯ РОБОТИ, АНАЛІЗ ЧАСОВИХ РЯДІВ, СЕЗОННА МОДЕЛЬ АВТОРЕГРЕСІЙНОГО ІНТЕГРОВАНОГО КОВЗНОГО СЕРЕДНЬОГО.

Об'єкт дослідження – процес планування роботи прокату велосипедів.

Предмет дослідження – методи прогнозування під час планування роботи прокату велосипедів.

Мета роботи – розроблення програмного забезпечення підтримки роботи прокату велосипедів для прийняття рішень щодо планування його роботи на основі прогнозування попиту клієнтів.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова PHP, мова Python, модель SARIMA.

Результати. У результаті виконання роботи було комплексно розв'язано проблему шляхом створення методу прогнозування попиту клієнтів та розроблення програмного забезпечення.

Наукова новизна роботи полягає в запропонованому методі прогнозування попиту клієнтів прокату велосипедів, який характеризується використанням моделі SARIMA для забезпечення прогнозування, що дозволяє отримувати основні показники попиту клієнтів для прийняття рішень про планування роботи прокату велосипедів.

Висновки. Розроблена програмна система може використовуватися для підтримки роботи прокату велосипедів щодо фіксації виконуваних робіт та планування майбутньої діяльності.

Галузь використання – стаціонарні прокати велосипедів з одним або декількома пунктами прокату.

ABSTRACT

Explanatory note to the qualifying work of the master: 94 pages, 31 figures, 3 tables, 3 appendixes, 20 sources.

BICYCLE RENTAL, DEMAND FORECASTING, WORK PLANNING, TIME SERIES ANALYSIS, SEASONAL MODEL OF AUTOREGRESSIVE INTEGRATED MOVING AVERAGE.

The object of research is the process of planning the work of bicycle rental.

The subject of research is forecasting methods during planning the work of bicycle rental.

The purpose of the work is to develop software to support the work of bicycle rental to make decisions on planning its work based on forecasting customer demand.

Materials, methods and technical means: object-oriented programming, PHP language, Python language, SARIMA model.

Results. As a result of the work, the problem was comprehensively solved by creating a method of forecasting customer demand and software development.

The scientific novelty of the work lies in the proposed method of forecasting the customer demand for bicycle rentals, which is characterized by the usage of the SARIMA model to provide forecasting, which allows to obtain basic indicators of customer demand for decision making on planning bicycle rental.

Conclusions. The developed software system could be used to support the work of bicycle rental to record the performed work and plan future activities.

Field of usage include stationary bicycle rentals with one or more rental points.

ЗМІСТ

С.

Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження.....	11
1.1 Сучасний стан організації прокату велосипедів.....	11
1.2 Принципи створення програмного забезпечення підтримки роботи прокату велосипедів.....	14
1.3 Сучасні програмні системи підтримки роботи прокатів велосипедів...	16
1.4 Висновки за розділом 1 та постановка завдань дослідження.....	19
2 Матеріали і методи.....	20
2.1 Методи та моделі прогнозування під час планування роботи прокату велосипедів.....	20
2.2 Модель прогнозування SARIMA.....	23
2.3 Метод прогнозування попиту клієнтів прокату велосипедів.....	25
2.4 Висновки за розділом 2.....	31
3 Розробка архітектури програми.....	32
3.1 Вибір структурних рішень.....	32
3.2 Функціональні вимоги до програмної системи.....	33
3.3 Проєктування бази даних.....	38
3.4 Висновки за розділом 3.....	44
4 Основні рішення щодо реалізації компонентів системи.....	45
4.1 Структура розробленої програмної системи.....	45
4.2 Опис модулів реалізованої програмної системи.....	46
4.3 Висновки за розділом 4.....	49
5 Експлуатація, тестування та експериментальне дослідження програми.....	50
5.1 Призначення програми.....	50
5.2 Умови виконання програми.....	50
5.3 Виконання програми.....	50

5.4 Експериментальне дослідження методу прогнозування попиту клієнтів прокату велосипедів.....	57
5.5 Висновки за розділом 5.....	59
Висновки.....	60
Перелік джерел посилання.....	61
Додаток А Технічне завдання.....	64
Додаток Б Текст програми.....	68
Додаток В Слайди презентації.....	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AR	– autoregression;
ARIMA	– autoregressive integrated moving average;
LSTM	– long short term memory;
MA	– moving average;
SARIMA	– seasonal autoregressive integrated moving average.

ВСТУП

На даний момент якісна інфраструктура, якісне життя в місті стали якщо не синонімічними, то дотичними поняттями з велоінфраструктурою. Все це пояснюється тим, що зі збільшенням кількості автомобілів у будь-якому місті виникає проблема користування ними таким чином, щоб це залишалось зручно для себе та оточуючих. Зі збільшенням кількості автомобілів вирішити цю проблему тільки додаванням автомобілів для кожного мешканця, збільшенням довжини якісних доріг у місті, створенням всієї необхідної додаткової інфраструктури неможливо. Користуватися автомобілями в такому випадку стає незручніше, швидкість руху в автомобілі містом знижується. Тоді справді альтернативними засобами стають громадський транспорт та велоінфраструктура. Саме від якості їх організації залежить у результаті якість життя в місті відносно транспортних питань.

До питань забезпечення велоінфраструктури в сучасному місті належать питання організації велодоріжок, паркувань велосипедів та створення прокатів велосипедів. На даний час існують різні типи велопрокатів, проте організація кожного з них потребує відповідної роботи, яка не залежить від форми власності такого велопрокату. І державний, і приватний прокат велосипедів потребують обліку своєї діяльності, що зокрема необхідно для встановлення співвідношень між власними доходами та витратами. Усі ці питання вирішуються на рівні планування діяльності прокату велосипедів. Проте планування не відбувається одномоментно. Планування потребує постійного моніторингу діяльності і актуалізації запланованих показників, а значить і бази таких показників. Тому важливо розробити програмну систему, яка дозволить організувати роботу прокату велосипедів, підтримуючи при цьому виконання щоденних задач, як то фіксацію видачі велосипедів, їх повернення, визначення стану, якому знаходиться кожен конкретний велосипед, тощо та надання з цього даних, необхідних для підтримки планування роботи прокату велосипедів. Задля

цього потрібно як розробити систему, так і забезпечити відповідний математичний апарат для підтримки роботи цієї системи з забезпеченням актуального рівня.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Сучасний стан організації прокату велосипедів

Прокат велосипедів у різних своїх формах може представляти цікавість як для звичайних мешканців міст (представляється у формі мережі публічних міських прокатів, але є далеко не у всіх містах, зокрема не у всіх містах України, відсутній в Запоріжжі, тоді у випадку відсутності таких прокатів звичайних мешканців можуть зацікавити і окремі прокати, де велосипед береться на прокат на певний строк і таким чином дозволяє вирішити тимчасові проблеми переміщення містом у випадку відсутності адекватного транспорту або тимчасової відсутності власного транспорту – ремонту власного велосипеда, автомобіля), так і для туристів (не обов'язково іноземних, це можуть бути і туристи з цієї ж країни або місцеві мешканці, які прагнуть подорожувати власним містом – наприклад, островом Хортиця при проживанні в місті Запоріжжя – або містечками, які розташовані близько до місця проживання).

За прогнозами світовий ринок прокату велосипедів досягне 4 млрд. доларів США до 2025 року, при цьому легкий доступ, привабливі ціни та підтримка місцевої влади є одними з ключових факторів, що стимулюють зростання послуг прокату велосипедів, очікується, що використання цифрових платформ постачальниками послуг також сприятиме зростанню популярності послуг прокату в найближчі роки [1].

Історично можна виділити наступні типи велосипедного обміну:

- перший тип був послуги характеризується велосипедами, випадковим чином розміщеними в певних зонах для безкоштовного користування, проте в багатьох випадках такий варіант зазнавав невдачі, оскільки велосипеди були просто пошкоджені або кинуті в канали, або адаптовані для приватного користування;

- другий тип включає фінансову плату, щоб заохотити користувачів повертати позичений велосипед на станцію для повернення стягнутого

спочатку депозиту, до того ж в такому випадку використовується певна сплата за використання велосипеда;

– міські прокати, які характеризуються різноманітними технологічними удосконаленнями, зокрема електронними блокувальними бар'єрами, телекомунікаційними системами, смарт-картами чи картками на магнітній смужі, доступом до мобільного телефону та вбудованими комп'ютерами, в подальшому цей варіант удосконалювався зі створенням мобільних станцій на сонячній енергії, встановленням застосунків для смартфонів [2].

Одним з активних рушіїв розвитку традиційних прокатів велосипедів є туризм. За результатами дослідження, проведеного на основі ініціативи Європейського парламенту, європейського велосипедиста спонукають організувати для себе велосипедні канікули наступні фактори:

- спортивні активності;
- близькість до природи;
- гнучкість;
- здоров'я;
- віддаленість від повсякденного життя;
- інформативність;
- вивчення природи (земель) та людей;
- відпочинок та дозволя;
- культурна спадщина;
- досвід велопогулянок;
- екологічність;
- фізичні вправи;
- пригоди [3].

На даний момент для привертання уваги туристів до прокату необхідно враховувати наступні фактори:

– перелічити на вебсайті зручні місця для їзди на велосипеді після того, до того ж граним фактором є співпраця з організаціями, розташованими

в цих місцях, щоб вони рекламували велосипеди прокату на своєму вебсайті та на місці;

- додати посилання на місцеві маршрути, карти та путівники на своєму вебсайті та згадати про те, що маршрути мають низький трафік, якщо це так;

- додати список місць, де можна їсти та пити по дорозі;

- розповісти про те, як легко дістатися до прокату, скориставшись місцевим транспортом, наприклад, відстань до залізничного вокзалу [3].

Для підвищення ефективності прокату велосипедів необхідно при організації роботи прокату враховувати:

- необхідність тримання велосипедів у належному стані: якщо належним чином обслуговувати велосипеди, вони працюватимуть добре та забезпечать комфортну та безпечну їзду клієнтам, окрім того обслуговування велосипедів – найкращий спосіб зберегти інвестиції у прокат велосипедів, тому що якщо велосипеди не обслуговуються, вони почнуть псуватися;

- необхідність чистити велосипеди: наприклад, більшість компаній, що займаються прокатом велосипедів на пляжі, закликають орендарів не їздити на велосипедах на пляжі біля води, адже морська вода може бути надзвичайно їдкою для сталевих деталей велосипедів, це одна з причин, чому так важливо ретельно чистити кожен велосипед після його повернення (якщо видалити сіль та інше сміття з велосипеда, то ймовірність іржі буде меншою);

- важливість захисту велосипедів: необхідно пропонувати велосипедні замки з кожним із орендованих велосипедів, а також кріплення для велосипедних замків;

- врахування одного зі способів принести додатковий прибуток для бізнесу – послуг повернення велосипедів з прокату для багатоденних орендарів, коли за певну плату можна запропонувати відвідати готель, квартиру чи пляж, щоб забрати орендований велосипед до магазину (прокату);

- пришвидшення процесу оформлення оренди велосипеда, що охоплює етапи від вибору велосипедів до підписання домовленостей, оплати та роздрукування бланків та інших необхідних документів: потрібно зробити все можливе, щоб спростити цей процес, щоб орендарі могли легко отримати велосипед протягом декількох хвилин;

- використання системи онлайн прокату: це один з найкращих способів спростити процес оренди велосипедів, існує безліч різних послуг, які можна використовувати для оренди в Інтернет, при цьому клієнти можуть швидко оформити документи та завантажити необхідну контактну інформацію, не заходячи до магазину, відповідно не витрачаючи час роботи менеджерів та операторів, вони також можуть оплачувати онлайн за допомогою кредитної або дебетової картки, а це означає, що в підсумку сервісу не доведеться витрачати багато часу на обробку їхніх платежів, окрім того це спрощує і процес забирання велосипеда, адже тоді можна підготувати велосипеди ще до приходу клієнта в магазин, а як тільки він з'явиться, він може одразу отримати велосипед і почати його використовувати [4].

1.2 Принципи створення програмного забезпечення підтримки роботи прокату велосипедів

Програмне забезпечення підтримки роботи прокату велосипедів повинно розв'язувати наступні проблеми, які виникають перед власниками прокату:

- втрачений час та втрачені можливості: бронювання телефоном або бронювання офлайн добре підходять для спілкування з клієнтами, при цьому деяким менеджерам подобається мати такий особистий зв'язок, але потрібно враховувати, що відповідь на дзвінки та збирання деталей замовлення забирає багато часу і може стримувати в результаті бізнес, адже окрім потреби в додаткових працівниках варто враховувати, що поки обслуговується бронювання одного велосипеда, інший клієнт, можливо,

захоче зробити групове бронювання на 20 велосипедів, в очікуванні він знайде конкурентів і за вірної організації їх роботи встигне забронювати велосипеди у них, до того ж потрібно враховувати, що це обмежує вікно для прийняття бронювань, оскільки потенційні клієнти зможуть зв'язатися лише у робочий час, для багатьох клієнтів це може стати проблемою, окрім того це стане проблемою для мандрівників з інших країн (тим нагальнішою, чим більша відстань буде між часовими поясами);

– проблема подвійного бронювання та відсутності велосипедів: електронні таблиці мають великий потенціал для людських помилок, їх легко неправильно прочитати оператору велопрокату, в них легко ввести неправильну інформацію, це може в результаті ускладнити відстеження доступності велосипедів, в кінцевому підсумку може бути створене подвійне бронювання, до того ж кількість таких помилок може збільшитися, якщо організація прокату ускладнена додатковими можливостями (доповнення до замовлень, різна структура цін для різних випадків, робота на різних майданчиках), окрім того зрозуміло, що в такому випадку досить складно відстежувати стан конкретного велосипеда в конкретний момент часу, адже такі можливості відносяться до двох різних просторів функціональних можливостей, в одному з яких знаходиться замовлення, а в іншому – конкретний велосипед, і для того, щоб заповнити дані з одного простору, потрібно з'ясувати дані з іншого простору, що складно для організації в електронних таблицях;

– відстеження технічного обслуговування: важливе правило ведення бізнесу з прокату велосипедів – своєчасний догляд за велосипедами, це забезпечує безпеку клієнтів, надаючи їм найкращий досвід, до того ж забезпечення чистим, якісним обладнанням допомагає виділитися серед конкурентів і продовжувати накопичувати клієнтів, відповідно чим більшою стає кількість велосипедів у наявності в прокаті, тим складнішим стає процес обліку і контролю, при цьому якщо якийсь велосипед буде пропущений, це може призвести до поганого досвіду клієнтів або навіть до травм, тому для

захисту клієнтів та репутації знадобиться спосіб відстежувати обслуговування, у свою чергу програмне забезпечення дозволяє встановити графік обслуговування для кожного велосипеда, після того, як послуга буде виконана, вона записується в журнал технічного обслуговування, що важливо зокрема і для забезпечення документального підтвердження у випадку виникнення конфліктної ситуації [5].

1.3 Сучасні програмні системи підтримки роботи прокатів велосипедів

На даний момент існують готові рішення програмних систем підтримки роботи прокатів велосипедів.

Yoou (рис. 1.1) є однією з таких систем [6]. Вона надає можливість онлайн бронювання, включаючи зокрема і онлайн сплату за бронювання, керування бронюваннями, перегляд бронювань за заданою через календар датою та обраною годиною, за поточним моментом (виконані та такі, що очікують обслуговування), перегляд звітів зі статистикою кількості бронювань за дні тижня та за інші періоди, керування наявними велосипедами. Налаштування тарифів можливо під погодинні періоди, півдобу та добу.

При цьому дана функціональність не є безкоштовною і потребує сплати певних сум за використання, існує наступний набір пакетів для роботи з програмою: 60 фунтів стерлінгів на місяць за використання програми для 11-25 велосипедів, 90 фунтів стерлінгів на місяць за використання для 26-55 велосипедів та 120 фунтів стерлінгів на місяць за варіант роботи для 56-150 велосипедів.

Окрім необхідності сплати за користування програмою протягом всього часу її використання варто також відзначити те, що вона не має ніяких засобів прогнозування, відповідно можна тільки отримати узагальнені дані про виконані бронювання, налаштування тарифів є обмеженими, засоби

внесення додаткових даних (додаткового обладнання, сервісного або будь-якого додаткового обслуговування) є також обмеженими.

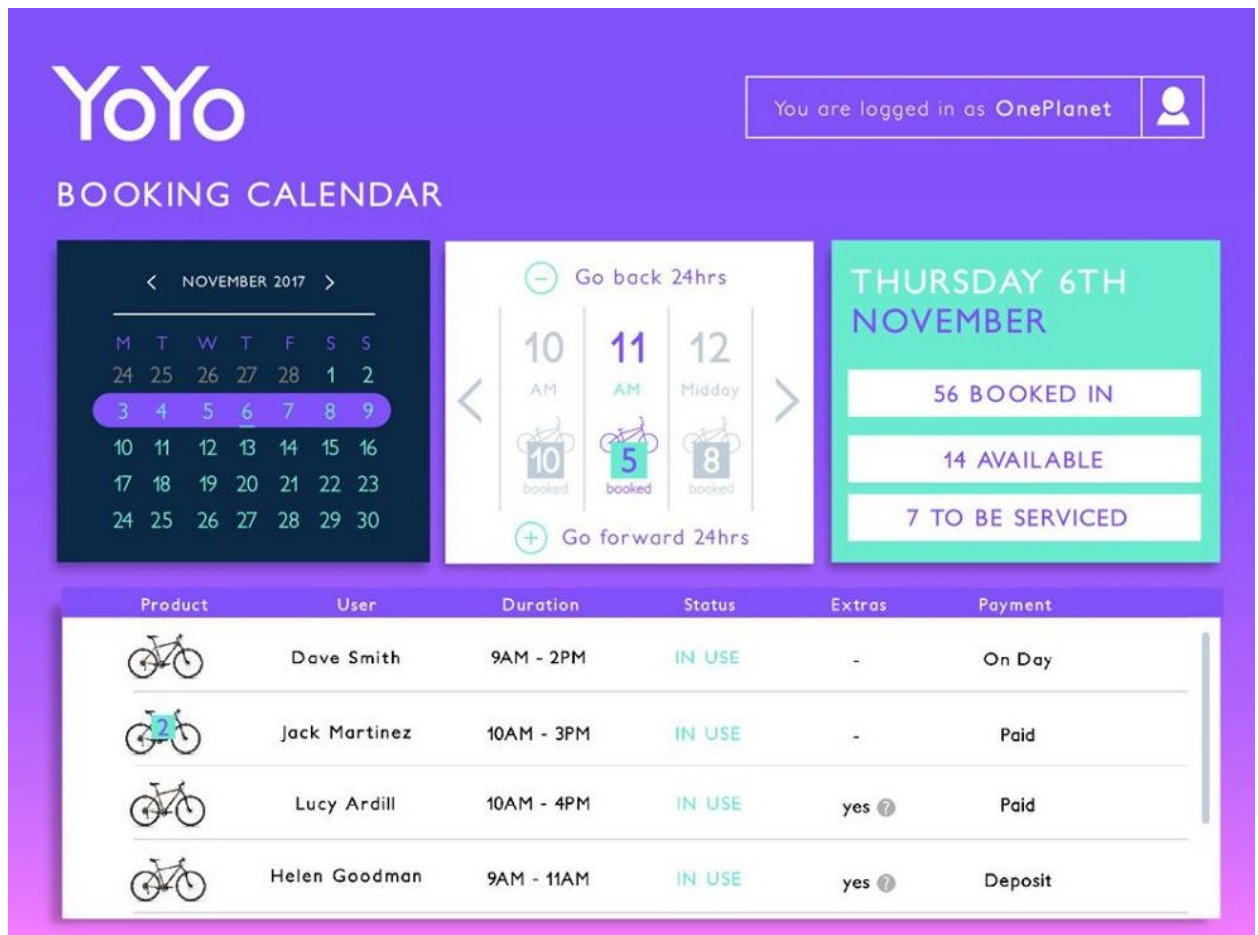


Рисунок 1.1 – Перегляд бронювань в програмі YoYo [6]

Ще однією подібною системою є Bike Rental Manager (рис. 1.2), яка надає наступні функціональні можливості:

- бронювання та внесення даних прокату: внесення для будь-якого бронювання отриманих платежів, редагування та скасування за потреби, внесення нотаток за попереднім замовленням або за обладнанням (можлива фіксація будь-якого обладнання в системі), включаючи зміну велосипеда, педалей, деталей подорожі тощо;
- дошка керування: дозволяє перевіряти, на якому етапі перебуває прокат, скільки велосипедів знаходиться в прокаті на даний момент, скільки

бронювань виконано, коли бронювання починаються, скільки велосипедів потребує обслуговування;

- планування: огляд із керуванням на основі календаря для планування доступу до велосипедів, візуальний огляд вкаже будь-які пропуски в бронюванні, які потрібно заповнити;

- можливість налаштування відповідно до потреб: наприклад, можна обрати, чи може клієнт вибирати при бронюванні конкретний велосипед, чи тільки модель;

- відстеження: контроль над обладнанням включає відповіді на запитання, чи сплачено за велосипед, чи готовий велосипед до прокату, чи відданий велосипед у прокат, коли велосипед потребує обслуговування, чи потребує велосипед обслуговування [7].

The screenshot displays the 'BIKE HIRE' interface. At the top, there's a header 'BIKE HIRE' and a background image of a person riding a bike. Below the header, the 'Step 1/5 - CHOOSE DATES' section is active. It shows a date range from '05/09/2020' to '05/09/2020' with time slots '09:00' and '18:00'. A 'CONFIRM' button is present. The 'Step 2/5 - TYPE OF BIKE' section is also visible, showing a list of bike categories: MOUNTAIN, GRAVEL, HYBRID, DOWNHILL, CHILD, and BEACH CRUISER, each with a corresponding icon.

Рисунок 1.2 – Бронювання в програмі Bike Rental Manager [7]

Щодо даної системи потрібно також зазначити, що при всіх своїх додаткових функціональних можливостях вона все-таки не має засобів прогнозування, відповідно планування за її допомогою можна виконувати тільки для поточної діяльності, а на майбутнє такі можливості відсутні, можна тільки оперувати статистикою, до того ж система також є платною. За

роботу з одиницями товарів (велосипедами, обладнанням) у кількості до 50 потрібно сплатити 59 \$ на місяць або 708 \$ на рік, при кількості до 100 одиниць сплата становить 89 \$ на місяць або 1068 \$ на рік, при кількості до 250 одиниць сплата становить 149 \$ на місяць або 1788 \$ на рік, за більших обсягів вартість є договірною. До того ж за додаткові можливості потрібно сплачувати додатково. Так якщо планується робота в декількох підрозділах (магазинах), то за кожний доведеться сплатити ще по 200 \$.

1.4 Висновки за розділом 1 та постановка завдань дослідження

Виконано огляд основних принципів організації сучасних прокатів велосипедів, створення програмного забезпечення підтримки роботи прокату велосипедів, проаналізовано сучасні програмні системи підтримки роботи прокатів велосипедів, включаючи Yoyo та Bike Rental Manager, що є платними засобами і не мають інструментів прогнозування.

У кваліфікаційній роботі магістра потрібно розв'язати такі завдання:

- дослідження методів прогнозування під час планування роботи прокату велосипедів;
- розробка технічного завдання і вибір засобів розробки;
- проєктування структури програмної системи;
- розробка програмної системи підтримки роботи прокату велосипедів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Методи та моделі прогнозування під час планування роботи прокату велосипедів

Правильне планування роботи прокату велосипедів є дуже важливою діяльністю. Від того, на скільки якісно і обґрунтовано виконано дану діяльність, залежить фінансовий успіх організації загалом. Звичайно, що основою планування є визначення кількості пунктів прокатів, кількості велосипедів, на що безпосередньо впливає попит клієнтів. Розуміючи наявні фінансові ресурси, можна звичайно запланувати роботу з відповідною кількістю пунктів та велосипедів у них, але фактично може виявитися, що попиту на таку кількість велосипедів немає. Тому спочатку має бути визначено, яка потреба виникає у клієнтів, а далі вже відрегульовано наявними фінансовими ресурсами її, що в результаті дозволить визначити основні показники діяльності прокату.

У роботі [8] випадкові ліси (random forest) та лінійна регресія досліджуються для прогнозування попиту прокату велосипедів. В якості параметрів моделі використовуються дата, для якої реалізується прогнозування, погода (використовуються 3 типи погоди), температура повітря, те, на скільки відчувається температура повітря, швидкість вітру та вологість повітря. За результатами дослідження було виявлено, що саме випадкові ліси дозволили отримати кращі результати: точність прогнозування складала 82 % (детальніших даних в дослідженні не наводилось) для моделі на основі випадкових лісів та 50 % (коефіцієнт кореляції R складав 0,572, а стандартна помилка дорівнювала 146,32) для лінійної регресії.

Подібне дослідження було також виконано в роботі [9], де досліджувалось використання регресійних моделей для прогнозування попиту на наступний день, включаючи лінійну регресію, регресію екстремального градієнтного бустингу та регресію на основі випадкових

лісів. Кращі результати, хоча і доволі співставні (для двох останніх регресійних моделей), показали випадкові ліси. Окрім того в роботі було досліджено важливість ознак, найбільшу з яких мала відповідна година дня, далі слідувала температура, а вже після цього вологість, сонячна радіація, сезонність і ознаки, що достатньо мало впливали на підсумковий результат (дощ, день тижня, швидкість вітру, видимість та сніг). Тобто загалом дані результати вказують на те, що саме послідовність значень найбільше впливає на поточне отримане значення разом з загальними погодними умовами (сенса використання сезонності, врахування окремих природних явищ, які до того ж протягом дня можуть змінювати свою активність, дещо занижений).

У роботі [10] прогнозування реалізовувалось на наступні 6 годин (прогнозування 12 значень за відповідні періоди у півгодини) на основі моделі, побудованої з використанням рекурентних нейронних мереж, при цьому прогнозування відбувається для кожної окремої станції прокату, а в параметрах моделі окрім кількості прокатів використовується також просторова інформація, що визначає станцію, для якої відбувається прогноз. Порівняння даної моделі в дослідженні відбувалось з моделлю регресії Пуасона, яку модель на основі рекурентних нейронних мереж перевершила у точності прогнозування за дослідженням на вибірці даних. Але покращення відбулось для середнього результату, в той же час модель продемонструвала гірші результати у вихідні та години пік.

У роботі [11] розглядається проблема визначення надлишкового попиту. Під надлишковим попитом у системах прокату велосипедів розуміють, наприклад, скільки клієнтів намагалися орендувати велосипед зі станції, на якій вже не було велосипедів, порожньої станції), оскільки цей вид попиту не записаний безпосередньо у журнали поїздок на велосипеді, то виконується моделювання, яке ґрунтується на визначенні необхідних часових сегментів у даних про наявність велосипедів, при цьому щоб передбачити чистий сукупний попит (який включає спостережуваний та надлишковий попит), використовується модель регресії Скеллама за допомогою оцінки

максимальної ймовірності, але дане дослідження піднімає загалом важливе питання, для якісного розв'язання якого необхідно зафіксувати відповідні дані і виконувати не моделювання, а прогнозування, зокрема для цього в системи обміну велосипедами можуть використовувати відповідний мобільний застосунок для визначення звернень користувачів, які не були задоволені системою [11].

Методи прогнозування, представлені в роботах [8]-[10], дозволяють прогнозувати потребу прокату велосипедів на наступний день або на годину чи декілька годин наперед. При цьому необхідно розуміти, що планування має відбуватися на основі усереднених показників. Тобто прогнозування має виконуватися на певний період (наприклад, місяць або сезон). Найменшим таким періодом є місяць. Потреби у прогнозуванні на кожен день саме для планування немає: таке прогнозування впливає саме на результати операційної діяльності, а не на планування.

Окрім безпосередньо наявності фінансових ресурсів, попиту, варто також розуміти, що велосипеди потребують постійного обслуговування, тому навіть якщо в певний період виникає більша потреба у велосипедах, їх обслуговування в проміжні періоди може призвести до того, що витрати на велосипеди можуть перевищувати доходи від їх використання. Тому важливо прогнозувати попит у розрізі загальної або середньої кількості прокатів за весь період, середньої кількості прокатів за святкові або вихідні дні протягом такого періоду та мінімальної (максимальної) кількості виконаних прокатів за день.

Для реалізації прогнозування часових рядів використовуються наступні моделі:

- модель авторегресії (autoregression (AR)): авторегресивна модель передбачає майбутню поведінку на основі минулої поведінки, використовується для прогнозування, коли існує деяка кореляція між значеннями в часовому ряду та значеннями, які передують їм і є їх наступниками;

- модель авторегресивної інтегрованої ковзаючої середньої (autoregressive integrated moving average (ARIMA)): модель ARIMA є одними з найбільш широко використовуваних підходів для прогнозування часових рядів, хоча фактично це не модель, а клас моделей, який пояснює певний часовий ряд на основі його власних минулих значень, тобто власних затримок та помилок затриманих прогнозів, так що це рівняння можна використовувати для прогнозування майбутніх значень;
- модель сезонної авторегресивної інтегрованої ковзаючої середньої (seasonal autoregressive integrated moving average (SARIMA)) – це модель, що розширює базову модель ARIMA і дозволяє включати сезонні моделі;
- модель long short term memory (LSTM) – це довготривала короткострокова пам'ять, тип рекурентної нейронної мережі, яка може вивчити залежність порядку між елементами в послідовності, її часто використовують для вирішення задач прогнозування часових рядів;
- модель DeepAR, розроблена компанією Amazon, є імовірнісною моделлю прогнозування на основі авторегресивних рекурентних нейронних мереж [12];
- модель N-BEATS: N-BEATS – це користувацький алгоритм глибокого навчання, який базується на зворотному та прямому залишкових посиленнях для одновимірного прогнозування точок часових рядів [13];
- модель temporal fusion transformer (від компанії Google) – це нова архітектура, орієнтована на увагу, яка поєднує високоефективне багатогоризонтне прогнозування з інтерпретаційним розумінням часової динаміки [14]-[15].

2.2 Модель прогнозування SARIMA

У моделі множинної регресії реалізується прогнозування змінної, що представляє інтерес, за допомогою лінійної комбінації окремих факторів прогнозування, у свою чергу модель AR виконує прогнозування значення

змінної, що представляє інтерес, використовуючи лінійну комбінацію минулих значень змінної, при цьому термін AR вказує на те, що це регресія змінної відносно неї самої, відповідно авторегресійна модель порядку p може бути записана на основі формули (2.1):

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t, \quad (2.1)$$

де ε_t – білий шум;

$y_{t-1}, y_{t-2}, \dots, y_{t-p}$ – значення прогнозованої змінної на минулій, попередній та p -й часових відмітках;

y_t – прогнозована величина;

$c, \varphi_1, \varphi_2, \dots, \varphi_p$ – коефіцієнти [16].

Замість використання минулих значень прогнозованої змінної в регресії, модель ковзної середньої (moving average (MA)) використовує помилки минулого прогнозу в регресійно-подібній моделі порядку q на основі формули (2.2):

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}, \quad (2.2)$$

де $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q}$ – значення помилки прогнозування на минулій, попередній та q -й часових відмітках;

$c, \theta_1, \theta_2, \dots, \theta_q$ – коефіцієнти [16].

Якщо поєднати диференціювання з AR та моделлю MA, то буде отримано несезонну модель ARIMA, яку можна представити відповідно на основі формули (2.3):

$$y'_t = c + \varphi_1 y'_{t-1} + \varphi_2 y'_{t-2} + \dots + \varphi_p y'_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}, \quad (2.3)$$

де $y'_t, y'_{t-1}, \dots, y'_{t-p}$ – різниця часового ряду (може мати декілька порядків, тобто різниця може братися декілька разів, ця кількість виражається ступенем моделі d) [16].

У результаті кожна конкретна ARIMA модель визначається трійкою параметрів ARIMA (p, d, q).

Часовий ряд можна розбити на 3 складові:

- тенденція – рух даних, направлений вгору та вниз з плином часу протягом великого періоду часу;
- сезонність – сезонна різниця в значеннях;
- шум – зміни через випадкові проміжки часу [17].

Сезонність у часовому ряді – це звичайна модель змін, що повторюються протягом S періодів часу, де S визначає кількість періодів часу, поки шаблон не повториться знову, модель SARIMA включає як несезонні, так і сезонні фактори в мультиплікативну модель (2.4):

$$ARIMA(p, d, q) \cdot (P, D, Q)_S, \quad (2.4)$$

де p – порядок несезонної AR;

d – ступінь несезонної диференціації;

q – несезонний порядок MA;

P – порядок несезонної AR;

D – ступінь несезонної диференціації;

Q – несезонний порядок MA;

S – часовий діапазон повторюваного сезонного шаблону [18].

2.3 Метод прогнозування попиту клієнтів прокату велосипедів

Для планування діяльності прокату велосипедів потрібно отримати показники:

– середньої кількості орендованих велосипедів за день: важливо, що під даним показником розглядається не усереднена кількість окремих активних замовлень прокату велосипедів за день, а саме усереднений показник на день протягом всього періоду часу (місяць або сезон), при цьому показник обчислюється як максимальна кількість велосипедів, які перебували в прокаті одночасно за день, відповідно саме цей показник дозволяє визначити, скільки велосипедів потребував прокат у середньому на день роботи;

– максимальної кількості орендованих велосипедів за день: визначається як максимальний протягом усього періоду (місяця, кварталу або року) показник максимальної кількості велосипедів, одночасно орендованих протягом дня.

Сукупність цих двох показників дозволяє в залежності від наявних ресурсів, а також розглядаючи дані показники у динаміці періодів (тобто наприклад, протягом кожного місяця року), вирішувати, яка саме кількість велосипедів потрібна для роботи прокату.

Показник максимальної кількості характеризує дні, коли попит є найвищим, тобто визначає, скільки велосипедів потрібно було прокату, щоб задовольнити весь попит власних клієнтів протягом всього періоду. Проте така кількість велосипедів може знадобитися у святковий або вихідний день, відповідно в інші дні вони будуть простоювати і вимагати сплати за обслуговування. Але всі клієнти за такої кількості велосипедів були б повністю задоволені.

Задача прогнозування попиту клієнтів прокату велосипедів є задачею прогнозування часового ряду, оскільки для її вирішення необхідно визначити значення одного з трьох перелічених вище показників на наступний період, знаючи послідовність його значень за минулі періоди. Наприклад, на сьомий місяць за значеннями попередніх шести місяців або на наступний рік.

Для визначення моделі, яка має використовуватися в основі методу, який розробляється, спочатку було виконано порівняння моделей, аналіз

яких було виконано в підрозділі 2.1 (табл. 2.1).

Таблиця 2.1 – Аналіз методів прогнозування часових рядів для прогнозування попиту клієнтів прокату велосипедів

Модель	Прогнозування часових рядів	Потреба великого обсягу даних	Період прогнозу	Характеристика
ARIMA	+	–	Зазвичай короткостроковий	Не враховує сезонність
SARIMA	+	–	Зазвичай короткостроковий	Враховує сезонність
LSTM	+	+	Придатна для довгострокового	Дозволяє вносити додаткові показники
DeepAR	+	+	Придатна для довгострокового	Розроблена Amazon, створює спільну модель для різних показників
N-BEATS	+	+	Придатна для довгострокового	Глибока нейронна архітектура
Temporal fusion transformer	+	+	Придатна для довгострокового	Розроблена Google архітектура на основі механізму уваги

За результатами аналізу було виділено в якості основних моделей ARIMA та SARIMA. Це пов'язано з тим, що у випадку такого планування

зазвичай немає великого обсягу даних: якщо планування реалізується на місяць (у випадку сезону це ще більш виражено) і наприклад, використовується 6 місяців для прогнозування значення для сьомого, то за 10 років буде накопичено дані тільки за 120 місяців, а враховуючи, що горизонт прогнозування 6 місяців, то виходить вдасться накопичити 114 екземплярів вибірки даних. У випадку сезону виходить відповідно 40 значень за 10 років, відкидаючи при цьому ще й довжину горизонту, тобто навіть менше в підсумку. З такою кількістю даних використання глибоких нейронних мереж непридатне, адже модель не зможе якісно навчитися на такому обмеженому наборі даних.

Моделі ARIMA та SARIMA призначені для короткострокових прогнозів, які розраховані на періоди менше одного року з нормальним діапазоном від одного до трьох місяців [19].

Оскільки модель SARIMA враховує сезонний вплив, а для прокату велосипедів це є важливим фактором, адже зокрема виражено в зимові та літні місяці (для місцин з подібним до України географічним розташуванням) цифри будуть різнитися, то саме модель SARIMA обрана для використання в запропонованому методі з метою виконання прогнозування попиту клієнтів прокату велосипедів.

Для прогнозування значень зазначених показників було запропоновано метод прогнозування попиту клієнтів прокату велосипедів, що складається з наступних етапів (рис. 2.1).

Етап 1. Визначення значень показників середньої та максимальної кількості орендованих велосипедів на день протягом минулих періодів планування тривалості, що дорівнює заданій, на основі історичних даних.

Для обчислення даних показників спочатку потрібно проаналізувати вибірку даних, що характеризує історичні дані. У ній дані зібрані за всіма замовленнями прокату. Відповідно один велосипед може бути у прокаті протягом декількох діб. Для виконання обчислень важливо, щоб на кожен день були зафіксовані окремо всі замовлення, які були активні протягом

цього дня. Тобто якщо замовлення тривале, то у позиціях за кожним з цих днів потрібно внести його. Кожна позиція повинна складатися з часу початку прокату, часу завершення (початок роботи прокату, завершення роботи прокату відповідно), номеру велосипеда.

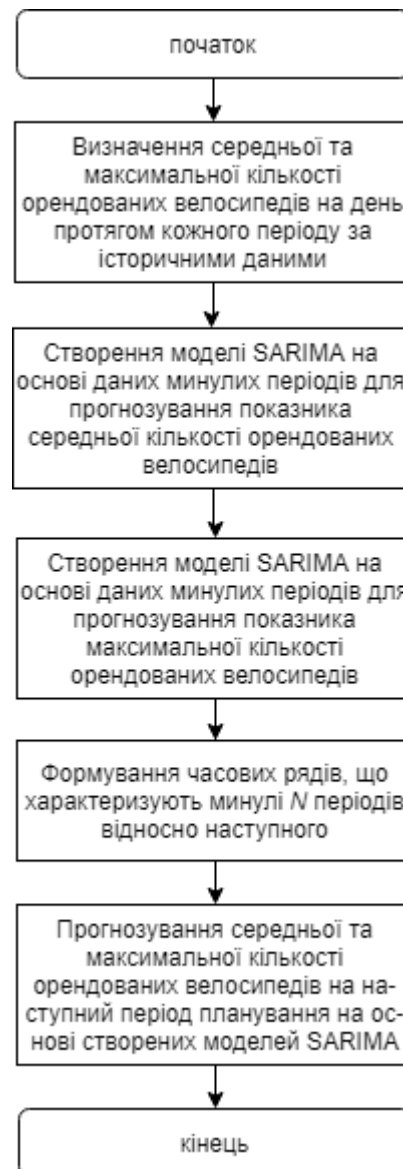


Рисунок 2.1 – Метод прогнозування попиту клієнтів прокату велосипедів

На основі отриманих даних виконується обчислення максимальної кількості велосипедів, які перебували в прокаті одночасно за день, за історичними даними. Для цього має бути проаналізовано кожне дискретне значення часу і визначено, скільки саме велосипедів було у прокаті в дану мить. Максимальне значення за день роботи відповідає потрібному.

Після обчислення показника максимальної кількості велосипедів, які перебували в прокаті одночасно за день, визначається для кожного періоду (місяць або сезон) з історичних даних середня та максимальна відповідні кількості шляхом аналізу всіх днів з періоду.

Етап 2. Створення моделі SARIMA на основі даних минулих періодів для прогнозування показника середньої кількості орендованих велосипедів.

Періодом, на який здійснюється прогнозування, може бути місяць або сезон. Якщо прогнозування має здійснюватися на рік, то має використовуватися модель ARIMA, адже сезонного фактору тоді немає.

При прогнозуванні на місяць параметр S моделі дорівнює 12, відповідно при прогнозуванні на сезон $S = 4$.

Етап 3. Створення моделі SARIMA на основі даних минулих періодів для прогнозування показника максимальної кількості орендованих велосипедів.

Етап 4. Формування часових рядів, що характеризують минулі N періодів відносно наступного.

Для обчислення показників у наступному періоді потрібно сформулювати два часові ряди, кожен з яких складається з послідовності значень показників (середньої та максимальної кількості орендованих велосипедів на день протягом періоду) за минулі N періодів.

Етап 5. Прогнозування значень показників середньої та максимальної кількості орендованих велосипедів на наступний період планування на основі створених моделей SARIMA.

Отримані в результаті роботи методу значення показників попиту клієнтів можуть бути використані для прийняття рішень про планування роботи прокату велосипедів. Наприклад, щодо зміни кількості велосипедів на майбутній період: придбання додаткових велосипедів або навпаки зменшення поточної використовуваної кількості (зайві велосипеди можуть бути законсервовані, що буде сприяти меншому їх зношуванню, вони також можуть бути продані або віддані в оренду іншим організаціям).

Метод розпочинає свою роботу на початку кожного періоду, тобто на початку відповідного місяця, сезону або року, виконуючи кожного разу всі етапи 1-5 послідовно. Рекомендовано зберігати результати обчислень першого етапу в файл для уникнення повторного виконання операцій та зберігати прогнози для уникнення повторного створення моделей.

2.4 Висновки за розділом 2

Проведений аналіз методів і моделей прогнозування під час планування роботи прокату велосипедів дозволив виділити основні підходи, які використовуються для прогнозування попиту, що включають: лінійну регресію, випадкові ліси, регресію екстремального градієнтного бустингу, рекурентні нейронні мережі. Сформульована задача прогнозування попиту клієнтів прокату велосипедів у вигляді задачі прогнозування часового ряду дозволила виділити набір відповідних моделей, що включає ARIMA, SARIMA, LSTM, DeepAR, N-BEATS, Temporal fusion transformer, серед яких було обрано SARIMA.

Представлено метод прогнозування попиту клієнтів прокату велосипедів, який дозволяє визначити середню, мінімальну та максимальну кількості орендованих велосипедів на наступний період планування (місяць, сезон) на основі використання моделі SARIMA для прогнозування відповідних показників на основі історичних даних. Це дозволяє отримувати основні показники попиту клієнтів для прийняття рішень про планування роботи прокату велосипедів.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

3.1 Вибір структурних рішень

Для початку проектування програмного забезпечення необхідно було обрати мову програмування. Вибір виконувався з множини мов, що включає Java, PHP, Python (табл. 3.1). За результатами порівняння було обрано для реалізації програмної системи мови PHP і Python. Мова PHP повинна забезпечувати реалізацію вебзастосунку, оскільки вона найбільш ефективна і зручна у використанні для програмування вебзастосунків, а мова Python необхідна для реалізації запропонованого методу, оскільки PHP не має готових бібліотек для реалізації базових моделей, але ефективна у веброзробці, а в той же час Python зручна для реалізації базових моделей, значно спрощує цей процес, проте не є на стільки ефективною у веброзробці.

Таблиця 3.1 – Вибір мови програмування

Критерій	Java	PHP	Python
Основне призначення	Загальне	Веброзробка	Загальне
Оптимізованість для веброзробки	+—	+	+—
Швидкість виконання коду	+	+—	—
Поетапна стабільність	—	+	+
Простота роботи з часовими рядами	+—	—	+
Невеликі витрати	+—	+	+

Для реалізації бази даних має бути використано систему керування базами даних MySQL, оскільки вона ефективно працює з вебзастосунками, створеними мовою програмування PHP, є безкоштовною, що важливо для

даного рішення, оскільки однією з його відмінностей порівняно з розглянутими аналогами є і безкоштовність створеного рішення.

Використання мови програмування PHP з врахуванням виконання розробки у вигляді вебзастосунку дозволять використовувати для побудови вебзастосунку архітектурний шаблон Model-View-Controller, що робить створене рішення гнучким і придатним для подальшої модифікації, адже це важливо в умовах роботи велопрокату.

3.2 Функціональні вимоги до програмної системи

Функціональними вимогами до програмної системи, що розробляється, визначено наступні:

- внесення інформації про велосипеди, наявні в прокаті;
- внесення інформації про обладнання, яке надається прокатом;
- внесення тарифів на прокат велосипедів;
- внесення тарифів на прокат обладнання;
- внесення пунктів прокату велосипедів;
- замовлення прокату велосипеда;
- замовлення групових прокатів велосипедів;
- оформлення технічного обслуговування велосипедів;
- визначення вартості замовлення прокату;
- оформлення повернення велосипеда;
- керування замовленням прокату велосипеда;
- перегляд активних замовлень прокатів велосипедів;
- перегляд замовлень прокатів велосипедів за вказаною датою;
- перегляд вільних для прокату велосипедів;
- перегляд вільного для прокату обладнання;
- перегляд доступних для прокату моделей велосипедів;
- перегляд велосипедів, які потребують обслуговування;
- перегляд складу велосипедів за моделлю;

- пошук замовлень прокату велосипеду;
- перегляд замовлень прокатів за клієнтом;
- перегляд незавершених вчасно замовлень прокатів;
- перегляд замовлень прокатів, які мають завершуватися у поточний день;
- завантаження даних для створення моделі прогнозування;
- прогнозування попиту клієнтів прокату велосипедів на місяць.

На основі визначеного переліку функціональних вимог визначено сценарії роботи клієнта прокату велосипедів (рис. 3.1).

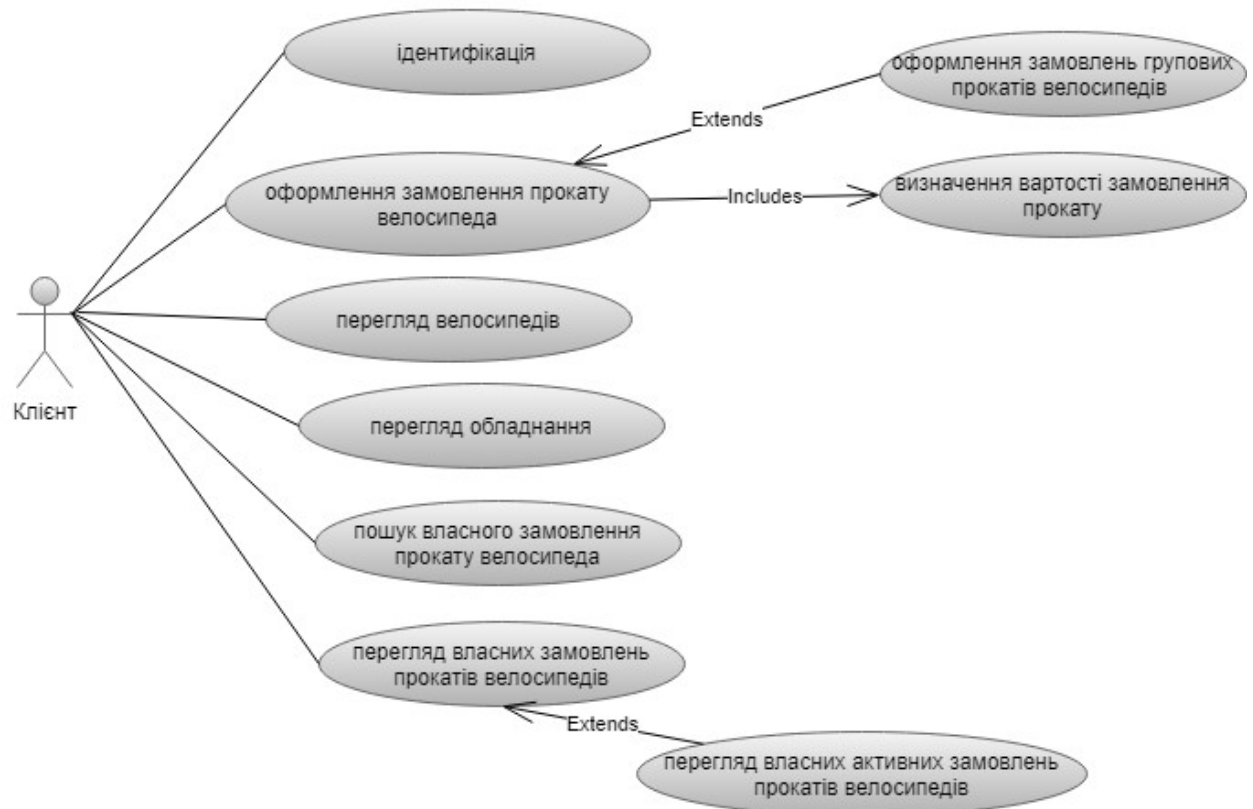


Рисунок 3.1 – Діаграма прецедентів клієнта

Сценарії включають:

- замовлення прокату велосипеда;
- замовлення групових прокатів велосипедів;
- визначення вартості прокату;
- пошук власного замовлення прокату велосипеду;

- перегляд власних активних замовлень прокатів велосипедів;
- перегляд власних замовлень прокатів велосипедів;
- перегляд доступних для прокату моделей велосипедів;
- ідентифікація клієнта.

Аналогічно було визначено сценарії роботи менеджера пункту прокату (рис. 3.2).

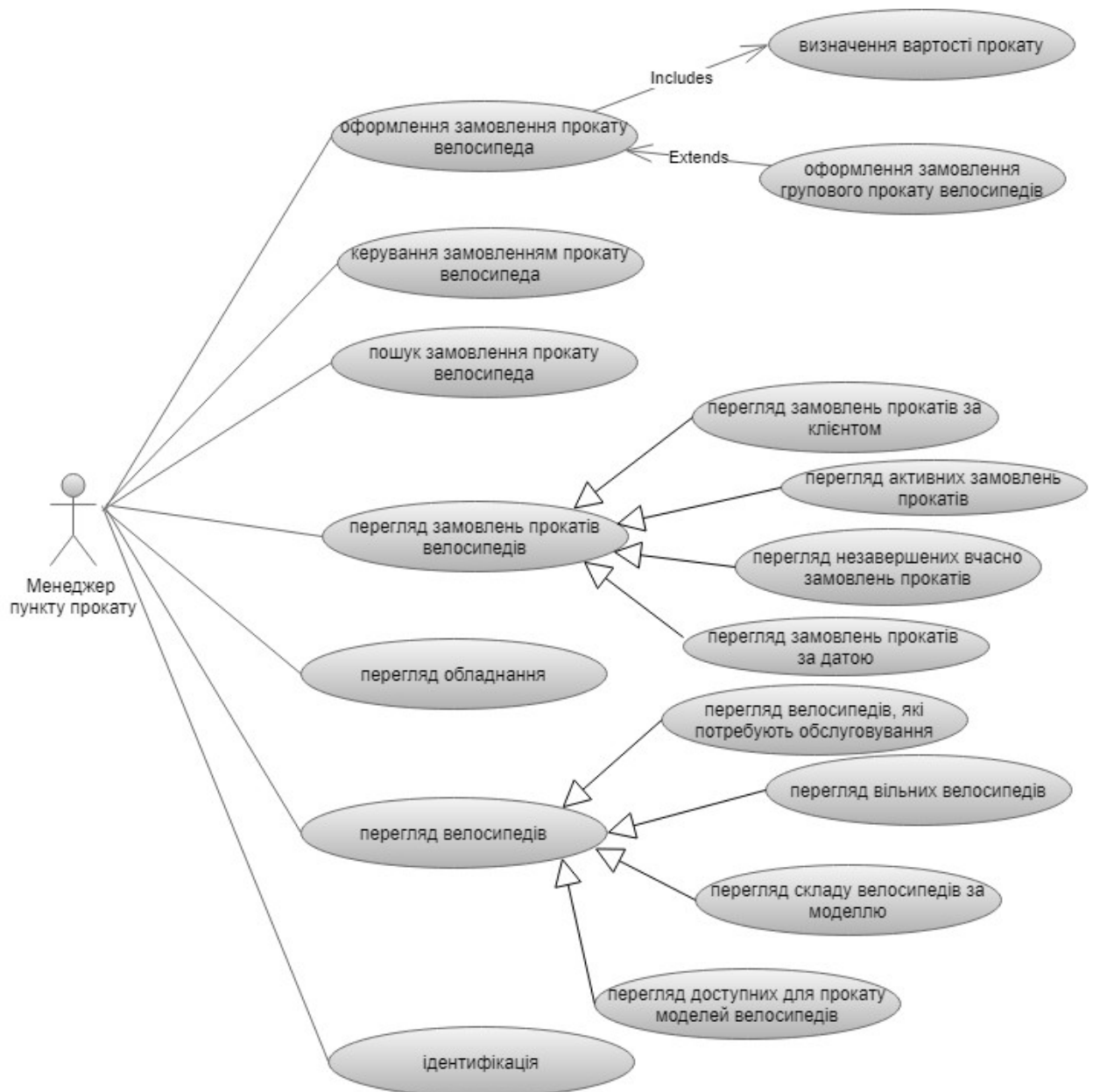


Рисунок 3.2 – Діаграма прецедентів менеджера пункту прокату

Сценарії включають:

- оформлення замовлення прокату велосипеда;
- оформлення замовлень групових прокатів велосипедів;
- визначення вартості замовлення прокату;
- оформлення повернення велосипеда;
- керування замовленням прокату велосипеда;
- перегляд замовлень прокатів за клієнтом;
- перегляд замовлень велосипедів, які потребують обслуговування;
- пошук замовлення прокату велосипеду;
- перегляд активних замовлень прокатів велосипедів;
- перегляд вільних для прокату велосипедів;
- перегляд вільного для прокату обладнання;
- перегляд складу велосипедів за моделлю;
- перегляд незавершених вчасно замовлень прокатів;
- перегляд замовлень прокатів велосипедів за вказаною датою;
- перегляд доступних для прокату моделей велосипедів;
- перегляд замовлень прокатів, які мають завершуватися у поточний день;
- ідентифікація.

Сценарії роботи менеджера з технічного обслуговування (рис. 3.3):

- внесення інформації про велосипеди, наявні в прокаті;
- внесення інформації про обладнання, яке надається прокатом;
- оформлення технічного обслуговування велосипедів;
- перегляд велосипедів, які потребують обслуговування.

Сценарії роботи адміністратора включають сценарії роботи менеджера пункту прокату і менеджера з технічного обслуговування та наступні сценарії (рис. 3.4):

- завантаження даних для створення моделі прогнозування;
- прогнозування попиту клієнтів прокату велосипедів на місяць;
- перегляд статистики прокату велосипедів;
- внесення тарифів на прокат велосипедів;

- внесення тарифів на прокат обладнання;
- внесення пунктів прокату велосипедів.



Рисунок 3.3 – Діаграма прецедентів менеджера з технічного обслуговування

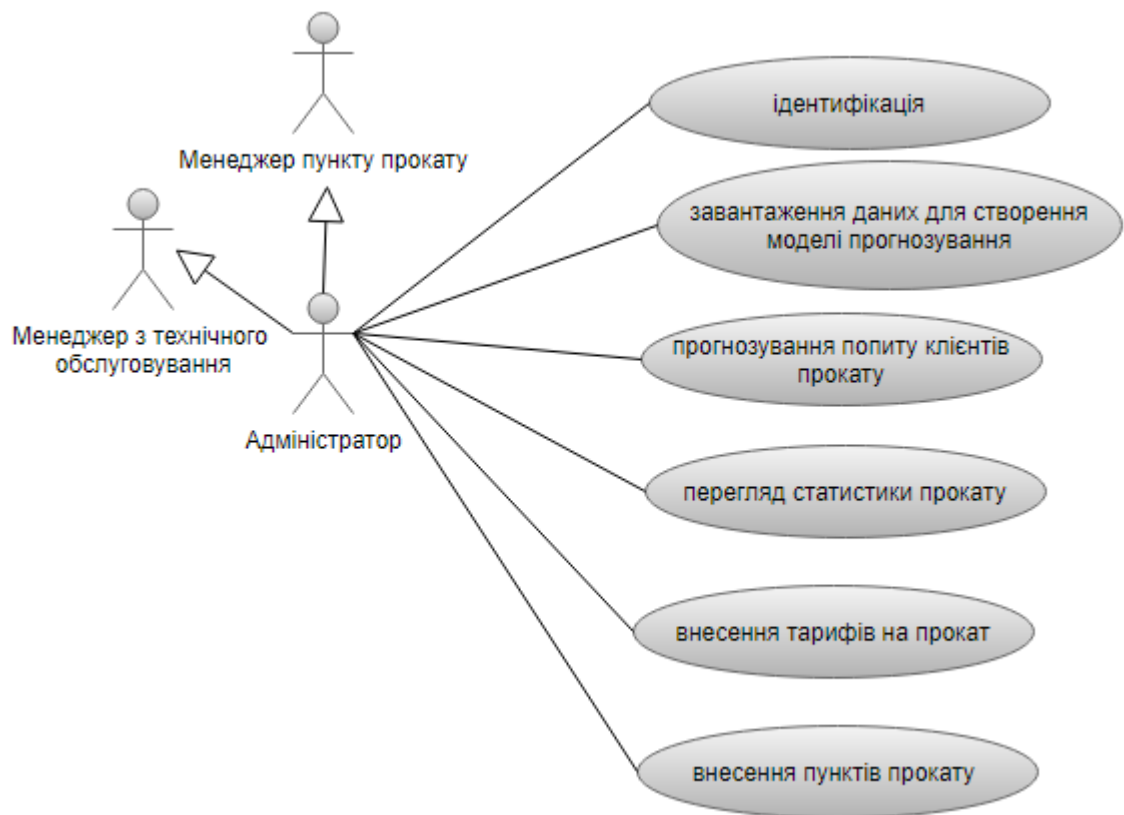


Рисунок 3.4 – Діаграма прецедентів адміністратора

3.3 Проектування бази даних

При проектуванні бази даних було враховано наступні принципи:

- замовлення може здійснюватися і велосипедів, і обладнання до нього;
- підтримка групових прокатів реалізується як у бронюванні, так і на рівні відповідних тарифів;
- окремі тарифи визначаються для велосипедів, а окремі для обладнання;
- усі роботи з обслуговування велосипедів мають бути зафіксовані в системі.

Таблиця Booking представляє інформацію про конкретне замовлення прокату велосипеда (або велосипедів) на основі наступних полів:

- id – номер замовлення прокату;
- client_id – клієнт, який бере велосипед та обладнання в прокат;
- begin – дата і час початку прокату;
- end_plan – заплановані дата і час завершення прокату;
- end_fact – фактична дата і час завершення прокату;
- creation – дата і час створення замовлення;
- price – сумарна вартість (фактична або запланована) прокату;
- payment – сума, сплачена на даний момент за прокат;
- notation – нотатка до прокату;
- bikerate_id – тариф на сплату за велосипеди;
- equipmenttrate_id – тариф на сплату за обладнання;
- service_begin – пункт прокату, з якого взято велосипед та обладнання;
- service_end – пункт прокату, на який повернуто велосипед та обладнання;
- manager1_id – менеджер прокату, який відповідальний за початок роботи з прокатом;

- `manager2_id` – менеджер прокату, який відповідальний за повернення з прокату;

- `bikes` – кількість велосипедів, що беруть участь в прокаті;

- `status` – поточний стан замовлення.

Таблиця `BikeBooking` представляє інформацію про велосипеди, які входять до складу замовлення прокату, на основі наступних полів:

- `id` – номер;

- `booking_id` – номер замовлення прокату;

- `bike_id` – номер велосипеда;

- `status` – поточний стан велосипеда у замовленні прокату;

- `notation` – нотатка до прокату конкретного велосипеда з замовлення.

Таблиця `BikeBooking` реалізує зв'язок багато-до-багатьох між таблицями `Bike` і `Booking`: одне замовлення може містити різну кількість велосипедів від одного і більше, а в свою чергу велосипед може входити до складу різних бронювань.

Таблиця `EquipmentBooking` представляє інформацію про обладнання, яке входить до складу замовлення прокату, на основі наступних полів:

- `id` – номер;

- `booking_id` – номер замовлення прокату;

- `equipment_id` – номер обладнання;

- `status` – поточний стан обладнання у замовленні прокату;

- `notation` – нотатка до прокату конкретного обладнання з замовлення.

Таблиця `EquipmentBooking` реалізує зв'язок багато-до-багатьох між таблицями `Equipment` і `Booking`: одне замовлення може містити різну кількість обладнання від нуля, одне і більше, а в свою чергу обладнання може входити до складу різних бронювань.

Таблиця `Operation` представляє інформацію про обслуговування велосипедів на основі наступних полів:

- `id` – номер обслуговування;

- `bike_id` – номер велосипеда;

- begin_time – дата і час початку обслуговування;
- end_time – дата і час завершення обслуговування;
- next_time – дата наступного обслуговування;
- notation – опис особливостей виконаного обслуговування.

Обслуговування здійснюється тільки для велосипедів, обслуговування обладнання не фіксується в системі.

Таблиця Mark представляє інформацію про марки велосипедів на основі наступних полів:

- id – номер марки велосипеда;
- name – назва марки.

Таблиця Model представляє інформацію про моделі велосипедів на основі наступних полів:

- id – номер моделі велосипеда;
- mark_id – номер марки велосипеда, якій належить модель;
- name – назва моделі велосипеда;
- photo – зображення моделі велосипеда;
- notation – опис моделі велосипеда;
- biketype – тип велосипеда (шосейний, міський тощо).

Таблиця Bike представляє інформацію про велосипеди на основі наступних полів:

- id – номер велосипеда;
- bikemodel_id – модель велосипеда;
- production – дата виробництва велосипеда;
- status – поточний стан велосипеда;
- notation – нотатка;
- last_operation – дата останнього обслуговування велосипеда;
- next_operation – дата наступного обслуговування велосипеда.

Тарифи окремо визначені для велосипедів і окремо для обладнання. Тарифи можуть бути визначені як для замовлення однієї одиниці, так і для декількох одночасно.

Таблиця BikeRate представляє інформацію про тарифи прокату велосипедів на основі наступних полів:

- id – номер тарифу на прокат велосипеда;
- bikemodel_id – номер моделі велосипеда, для якої встановлюється тариф;
- bikes – кількість велосипедів для застосування даного тарифу (для підтримки групових прокатів);
- hours – мінімальна кількість годин прокату, з якої починає діяти тариф;
- price – вартість прокату;
- begin – початкова сума, яку потрібно внести за прокат.

Таблиця EquipmentRate представляє інформацію про прокати обладнання на основі наступних полів:

- id – номер тарифу на прокат обладнання;
- equipmentmodel_id – модель обладнання, для якої встановлено тариф;
- equipments – кількість екземплярів обладнання даної моделі для застосування тарифу (для підтримки групових прокатів);
- hours – мінімальна кількість годин прокату, з якої починає діяти тариф;
- price – вартість прокату;
- begin – початкова сума, яку потрібно внести за прокат.

Таблиця EquipmentType представляє інформацію тип обладнання про на основі наступних полів:

- id – номер типу обладнання;
- name – назва типу обладнання.

Таблиця EquipmentModel представляє інформацію про моделі обладнання на основі наступних полів:

- id – номер моделі обладнання;
- name – назва моделі обладнання;

- equipmenttype_id – тип обладнання;
- photo – зображення даної моделі обладнання.

Таблиця Equipment представляє інформацію про конкретне обладнання на основі наступних полів:

- id – номер екземпляра обладнання;
- equipmentmodel_id – модель обладнання;
- production – дата виробництва екземпляра обладнання;
- status – стан екземпляра обладнання;
- notation – нотатка.

Таблиця User представляє інформацію про користувачів на основі наступних полів:

- id – номер користувача;
- login – логін користувача;
- password – пароль користувача;
- name – повне ім'я користувача;
- phone – телефон;
- status – рівень доступу.

Таблиця Manager представляє інформацію про менеджера пункту прокату на основі наступних полів:

- id – номер користувача;
- login – логін користувача;
- password – пароль користувача;
- name – повне ім'я користувача;
- phone – телефон;
- service_id – пункт прокату, на якому працює менеджер;
- status – рівень доступу.

Результуюча схема бази даних представлена на рис. 3.5.

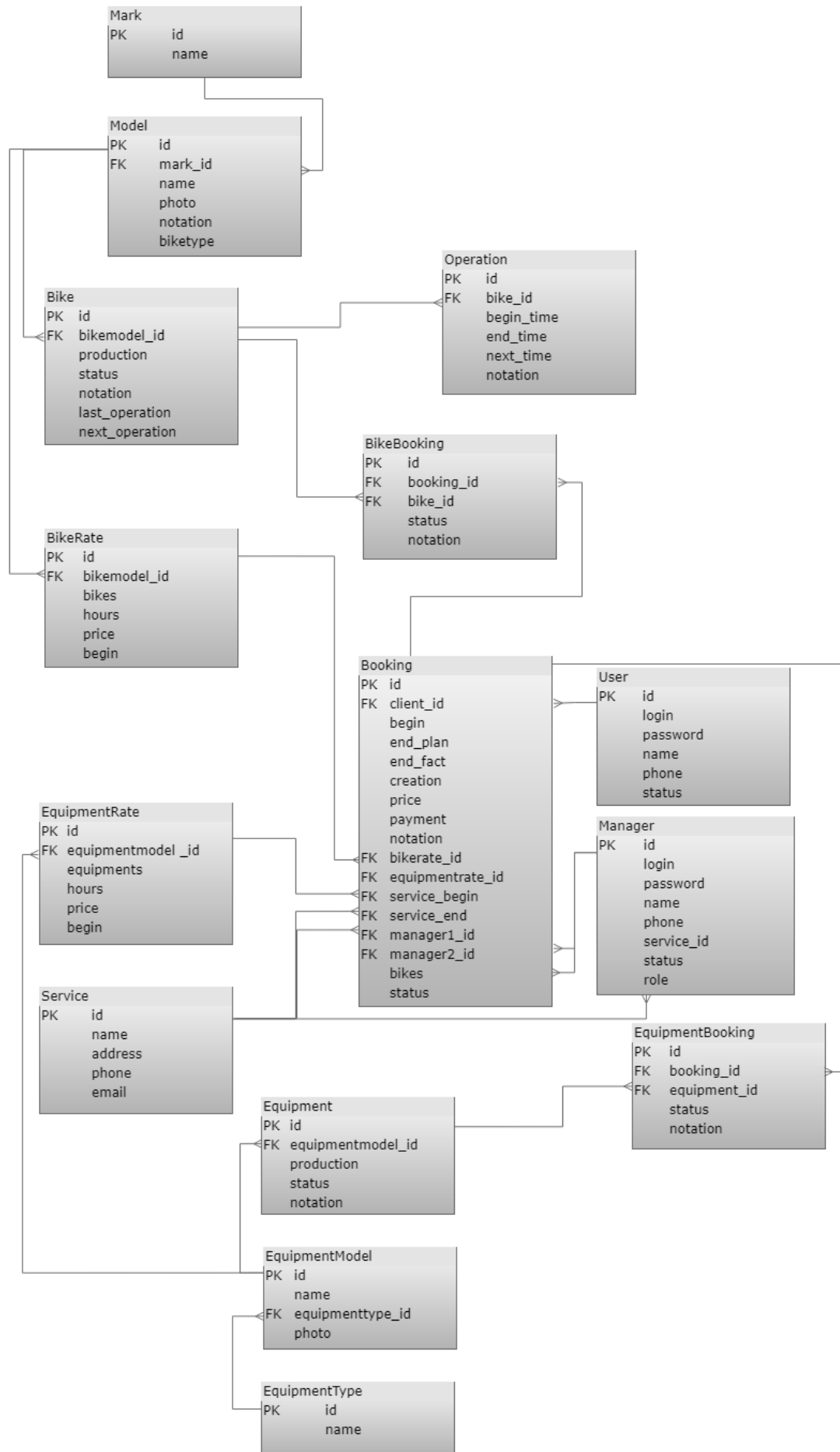


Рисунок 3.5 – Схема бази даних

Таблиця Service представляє інформацію про пункти прокату на основі наступних полів:

- id – номер пункту прокату;
- name – назва пункту прокату;
- address – адреса;
- phone – номер телефону;
- email – електронна адреса.

3.4 Висновки за розділом 3

Виконано вибір мови програмування PHP, що має забезпечувати безпосередньо розробку вебзастосунку, і мови програмування Python, що має забезпечувати роботу з часовими рядами під час реалізації запропонованого методу, для реалізації програмної системи за результатами проведеного порівняння. Обрано архітектурний шаблон Model-View-Controller для реалізації вебзастосунку, що робить створене рішення гнучким і придатним для подальшої модифікації, що важливо в умовах роботи велопрокату. Визначено функціональні вимоги до програмної системи зі створенням відповідних діаграм прецедентів. Визначено структуру бази даних.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Структура розробленої програмної системи

На рис. 4.1 представлено структурну схему розробленої програмної системи.



Рисунок 4.1 – Структурна схема програмної системи

Програмна система складається з підсистем, що підтримують роботу відповідних типів користувачів. В основі роботи цих підсистем підсистема реєстрації та аутентифікації користувачів, яка дозволяє визначити, з яким саме типом користувачів працює в поточний момент програмна система.

Підсистеми керування роботою менеджера з технічного обслуговування та менеджера пункту прокату є основою побудови підсистеми керування роботою адміністратора, оскільки він потребує всіх відповідних інструментів для своєї роботи та власних інструментів. Для реалізації частини власних інструментів призначена підсистема

прогнозування попиту клієнтів прокату велосипедів, яка реалізована за допомогою мови програмування Python.

4.2 Опис модулів реалізованої програмної системи

Клас BManagerModel реалізує містить методи:

- getActiveBookings(\$serviceId) – метод, який повертає набір активних замовлень прокату, які були розпочаті в пункті прокату serviceId;
- getAllActiveBookings() – метод, який повертає набір усіх активних замовлень прокату, наявних в програмній системі;
- getAllActivePreBookings() – метод, який повертає набір усіх активних передзамовлень, тобто замовлень прокату, які були оформлені клієнтами, але велосипеди ще не були за ними видані в пункті прокату;
- getPreBookingsByDate(\$pdate) – метод, який повертає перелік передзамовлень, які мають почати виконуватися в дату pdate;
- getPreBookingsByHour(\$pdate, \$phour) – метод, який повертає перелік передзамовлень, які мають почати виконуватися в дату pdate о phour годині;
- getUnTimeBookings(\$serviceId) – метод, який повертає перелік прострочених замовлень, за якими мали б повернути велосипеди в пункт прокату serviceId;
- getAllUnTimeBookings() – метод, який повертає перелік усіх прострочених замовлень;
- getAllReadyBookings() – метод, який повертає перелік усіх замовлень, які заплановані до завершення на поточну дату;
- getReadyBookings(\$serviceId) – метод, який повертає перелік усіх замовлень, які заплановані до завершення на поточну дату у пункті прокату serviceId;

- `getUserBookings($client_name)` – метод, який повертає перелік усіх замовлень клієнта з іменем `client_name`, відсортованих за зменшенням номеру замовлення;
- `getUserBookingsByPhone($phone)` – метод, який повертає перелік усіх замовлень клієнта з номером телефону `phone`, відсортованих за зменшенням номеру замовлення;
- `activatePreBooking($bookingId, $fullbooking, $bikes, $equipment)` – метод, який реалізує активацію передзамовлення (видачу велосипедів у прокат) з номером `bookingId` з внесенням до замовлення інформації `fullbooking` про нього, переліку велосипедів `bikes` та переліку обладнання `equipment`;
- `activateBooking($fullbooking, $bikes, $equipment, $client)` – метод, який реалізує створення і активацію замовлення (видачу велосипедів у прокат клієнту `client`) з внесенням до замовлення інформації `fullbooking` про нього, переліку велосипедів `bikes` та переліку обладнання `equipment`;
- `closeBooking($bookingId, $fullbooking, $bikes, $equipment, $manager, $mode)` – метод, який завершує замовлення з номером `bookingId` з внесенням інформації `fullbooking` про замовлення, переліком велосипедів замовлення `bikes` та обладнання `equipment` менеджером `manager`: параметр `mode` визначає істинним значенням успішне завершення замовлення, а хибним – відповідне неуспішне (наприклад, частину велосипедів не було повернуто);
- `updateBooking($bookingId, $bikes, $equipment)` – метод, який реалізує зміну стану велосипедів `bikes` та обладнання `equipment` за замовленням `bookingId`;
- `getBikesByModel($modelId)` – метод, який повертає перелік використовуваних (активних) і не орендованих у даний момент велосипедів, що належать до моделі `modelId`;
- `getFreeBikes($serviceId)` – метод, який повертає перелік вільних (неорендованих) у даний момент велосипедів, що знаходяться у пункті прокату `serviceId`;

- `getBikesByModelDate($modelId, $pdate)` – метод, який повертає перелік вільних (неорендованих) на дату `pdate` велосипедів моделі `modelId`;
- `getFreeEquipment($serviceId)` – метод, який повертає перелік вільного (неорендованого) у даний момент обладнання, що знаходяться у пункті прокату `serviceId`;
- `getEquipmentByModelDate($modelId, $pdate)` – метод, який повертає перелік вільного (неорендованого) на дату `pdate` обладнання типу `modelId`;
- `getBikesForTech($serviceId)` – метод, який повертає перелік велосипедів за пунктом прокату `serviceId`, що потребують технічного обслуговування на поточний момент.

Клас `TManagerModel` містить зокрема методи:

- `addBike($bike)` – метод, який дозволяє вносити дані нового велосипеда `bike`;
- `addEquipment($equipment)` – метод, який дозволяє вносити дані нового обладнання `equipment`;
- `addOperation($operation, $bikeid)` – метод, який дозволяє оформляти технічне обслуговування, інформація про яке міститься в `operation`, велосипеда `bikeid`;
- `getBikesForTech()` – метод, який повертає перелік велосипедів, що потребують технічного обслуговування на поточний момент;
- `getBikesForTechByDate($pdate)` – метод, який повертає перелік велосипедів, що потребують технічного обслуговування на дату `pdate`;
- `updateOperation($operationid, $operation)` – метод, який вносить дані `operation` про завершення технічного обслуговування з номером `operationid`.

Клас `AdminModel` наслідує класи `TManagerModel` і `BManagerModel` та зокрема містить методи:

- `addBikeModel($model)` – метод, який дозволяє вносити дані про нову модель велосипеда `model`;
- `addBikeMark($mark)` – метод, який дозволяє вносити дані про нову марку велосипедів `mark`;

- `addEquipmentModel($model)` – метод, який дозволяє вносити дані про нову модель обладнання `model`;
- `addEquipmentType($type)` – метод, який дозволяє вносити дані про новий тип обладнання `type`;
- `addService($service)` – метод, який дозволяє вносити дані про новий пункт прокату `service`;
- `getDemand($year)` – метод, який визначає попит на прокат за заданий рік `year`.

Для виконання методу `getDemand` реалізується звернення до файлу `demand.pickle`, в якому зберігаються місячні дані щодо попиту клієнтів прокату велосипедів та прогнозовані значення. Якщо виявляється, що даних за поточний місяць немає в файлі, то реалізується звернення до скрипту `predictdemand.py`, який на основі витягання з файлу `sarima.pickle` параметрів моделей виконує прогнозування попиту клієнтів на наступний період і зберігає отримані значення в файл `demand.pickle`.

Якщо під час звернення виявляється, що потрібно витягнути дані за минулий місяць, а для нього міститься тільки прогнозоване значення, то запускається скрипт `calculatedemand.py`, який витягає дані з бази даних про замовлення в цьому місяці та агрегує значення середньої та максимальної кількості велосипедів, які одночасно знаходились в прокаті.

4.3 Висновки за розділом 4

Описано реалізовану програмну систему підтримки роботи прокату велосипедів, для чого представлено структурну схему цієї системи, що включає підсистеми роботи відповідних типів користувачів, реалізовані мовою програмування PHP, та підсистему прогнозування попиту клієнтів прокату велосипедів, що реалізована мовою програмування Python для забезпечення роботи з запропонованим в роботі методом, розглянуто структуру відповідних класів.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Призначення програми

Програмна система призначена для організації роботи з відслідковування і керування станом велосипедів і обладнання, наявного в прокаті, та відслідковування і керування станом замовлень прокату на всіх їх етапах з подальшою підтримкою прийняття рішень.

5.2 Умови виконання програми

Технічні засоби, за допомогою яких організовується підтримка роботи серверної частини програмної системи, повинні відповідати наступним вимогам:

- процесор: щонайменше два ядра, мінімальна частота – 2 ГГц;
- оперативний запам'ятовуючий пристрій – не менше 4 Гб;
- жорсткий диск: вільний простір не менше 1 Гб.

5.3 Виконання програми

Усі типи користувачів взаємодіють з програмною системою як з вебзастосунком. Стандартний інтерфейс вебзастосунка (рис. 5.1) передбачає режим роботи для клієнта. В ньому користувач отримує домашню сторінку, на якій він може обрати пункт прокату та дати прокату. Задати час прокату з цієї сторінки неможливо, тобто використовується пошук велосипедів для подорожей тривалістю від одного дня. За потреби виконати замовлення на менш тривалий час потрібно натиснути на кнопку «Замовити Велосипед» в верхньому правому куті.

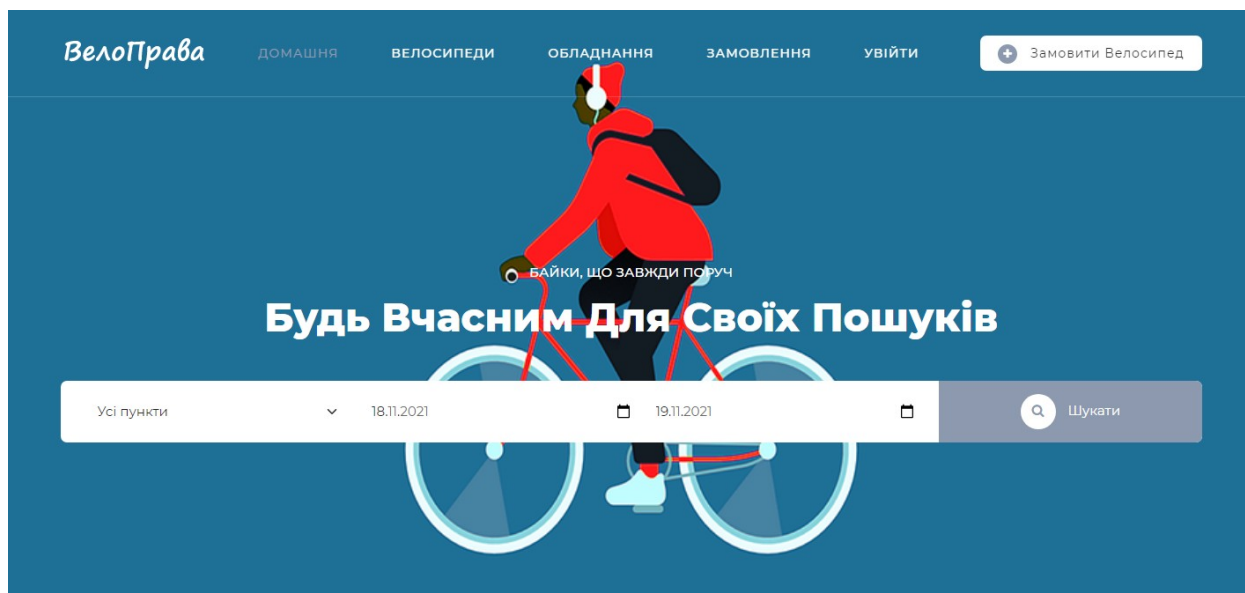


Рисунок 5.1 – Домашня сторінка вебзастосунка

Користувачу надається меню, що складається з наступних пунктів:

- «Велосипеди» (рис. 5.2), що дозволяє переглянути усі наявні в прокаті велосипеди;
- «Обладнання», що дозволяє переглянути усе наявне в прокаті обладнання;
- «Замовлення», що дозволяє переглянути усі замовлення прокату клієнта як авторизованого користувача: якщо користувач ще не авторизувався, то його буде перенаправлено на сторінку авторизації;
- «Профіль», що дозволяє переглянути і відредагувати за потреби дані власного облікового запису (якщо користувач ще не авторизувався, то буде відображено пункт «Увійти»);
- «Домашня», що відкриває розглянуту вище сторінку.

На сторінці перегляду велосипедів (рис. 5.2) потрібно обрати тип велосипеду (усі типи, міські, шосейні, гірські, дитячі). Тоді буде відображено усі моделі відповідного типу. За кожною моделлю відображається її назва, посилання на перелік тарифів на прокат цієї моделі, опис, мінімальна вартість прокату, зображення велосипеда. Якщо навести курсор на зображення велосипеда, то з'явиться кнопка «Замовити», що дозволить

перейти до сторінки оформлення замовлення, на якій потрібно буде задати дати і час прокату.

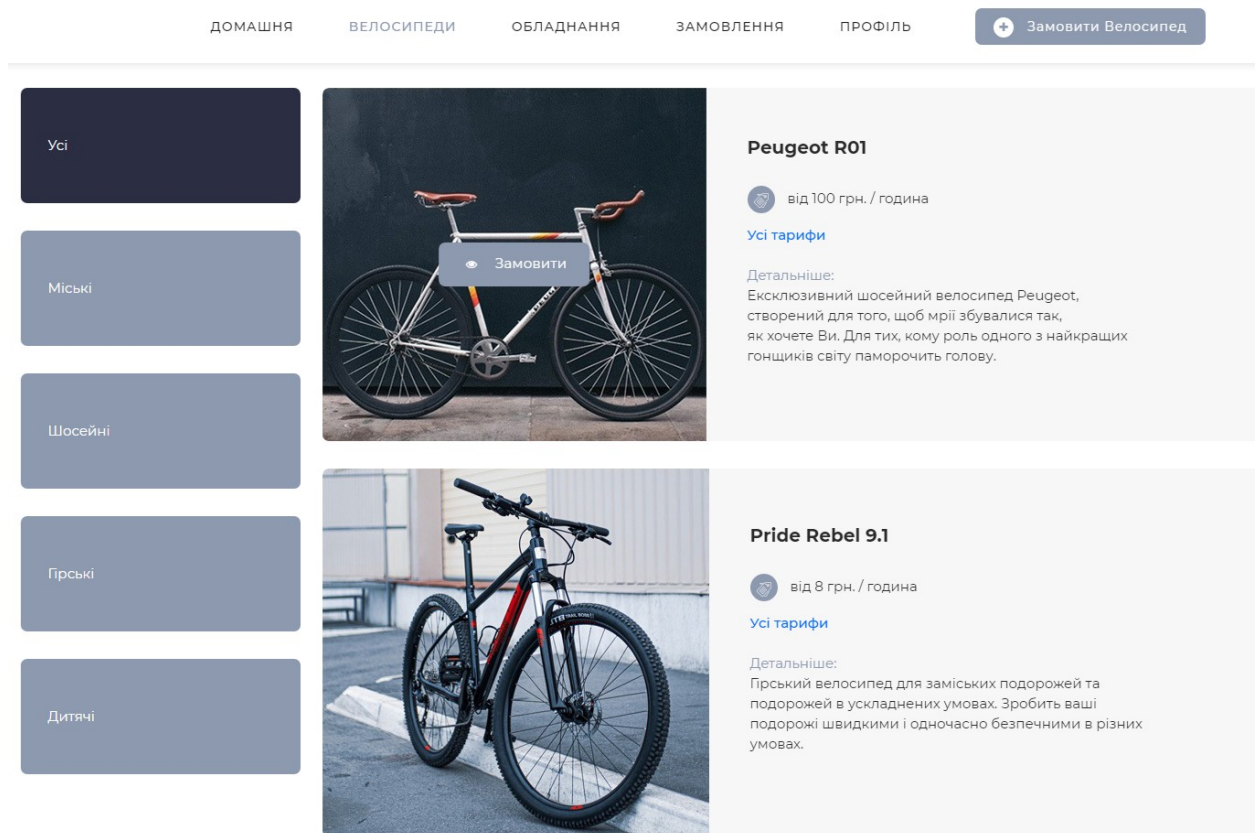


Рисунок 5.2 – Перегляд велосипедів клієнтом

На сторінці перегляду замовлень (рис. 5.3) у верхній частині можна оформити замовлення. У такому випадку необхідно визначити пункт прокату, де буде взято велосипеди, і пункт прокату, де його буде повернено, задати дати і час початку і завершення прокату. Після цього необхідно визначити перелік велосипедів та обладнання, що орендується. Для цього потрібно обрати модель велосипеда і натиснути на кнопку «Додати». Додана модель з'явиться вище в переліку велосипедів, де для неї можна буде обрати кількість велосипедів, натисканням на кнопки збільшення та зменшення. Аналогічно додається обладнання. В залежності від заданої тривалості замовлення, визначеної кількості складових замовлення, обчислюється сумарна вартість замовлення. Після цього для збереження замовлення необхідно натиснути на кнопка «Підтвердити замовлення». Тоді буде

створено передзамовлення, з яким потім має працювати менеджер для видачі велосипедів. Менеджеру в повній інформації про замовлення відображається також розмір депозита.

ДОМАШНЯ

ВЕЛОСИПЕДИ

ОБЛАДНАННЯ

ЗАМОВЛЕННЯ

ПРОФІЛЬ

Очистити Замовлення

Ваше нове замовлення

Пункт прокату:

Бульвар Шевченко

Пункт повернення:

Бульвар Шевченко

Дата і час початку прокату:

19.11.2021

10:15

Дата і час повернення:

19.11.2021

15:00

Велосипеди

Marin Bobcat Trail

-

1

+

90 грн.

Orbea MX

-

1

+

110 грн.

Додати модель:

Giant Talon

Додати

Обладнання

Додати:

Дитяче крісло

Vergo

Всього: 200 грн.

Підтвердити Замовлення

Історія замовлень

Активні

Номер

Введіть номер

Шукати

Номер	Початок	Повернення	Стан	Вартість	Дія
6	18.11.2021 15:03 Дубовий гай	18.11.2021 16:55 Дубовий гай	Передзамовлення	200 грн.	Скасувати
3	18.11.2021 11:12 Бульвар Шевченко	18.11.2021 12:47 Бульвар Шевченко	Завершений	200 грн.	
2	18.11.2021 09:27 Дубовий гай	18.11.2021 11:26 Дубовий гай	Завершений	110 грн.	

Рисунок 5.3 – Робота клієнта з власними замовленнями

На сторінці перегляду власних замовлень відображаються стандартно всі замовлення користувача, включаючи їх номери, пункт і час початку, пункт і час завершення, стан та вартість. Для активних замовлень відображається також кнопка «Скасувати», що дозволяє завершити передзамовлення (не підтверджене менеджером) передчасно. Для перегляду детальної інформації про замовлення потрібно натиснути на посилання з номером замовлення.

Клієнт також може виконати пошук замовлення за його номером або відобразити тільки активні замовлення, для цього йому потрібно натиснути на кнопку «Активні».

Якщо користувач авторизувався як менеджер з технічного обслуговування, то вебзастосунок для нього буде мати відповідний вигляд (рис. 5.4).

ВЕЛОСИПЕДИ ОБЛАДНАННЯ ОБСЛУГОВУВАННЯ ПРОФІЛЬ + Внести Обслуговування					
Велосипеди, що очікують обслуговування					
Номер	Велосипед	Вироблено	Останнє обслуговування	Наступне обслуговування	Дія
22	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
21	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
20	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
17	Marin Bobcat Trail	2020	-	19.11.2021	Керувати
11	Orbea MX	2019	20.11.2020	20.11.2021	Керувати
10	Orbea MX	2019	20.11.2020	20.11.2021	Керувати

Рисунок 5.4 – Перегляд менеджером з технічного обслуговування велосипедів, що очікують обслуговування

Для менеджера з технічного обслуговування доступні пункти меню «Велосипеди», звідки можна вносити інформацію про нові велосипеди,

«Обладнання», звідки можна вносити інформацію про нове обладнання, «Обслуговування», звідки можна переглянути перелік велосипедів, які потребують технічного обслуговування, тобто таких, для яких дата наступного технічного обслуговування настає в поточний день або наступила в попередні дні.

Кнопка «Внести Обслуговування» в верхньому правому куті дозволяє внести інформацію про виконане обслуговування.

Менеджер пункту прокату має меню (рис. 5.5), що складається з пунктів:

- «Велосипеди», який дозволяє переглядати велосипеди за допомогою одного з варіантів, представленого кнопками «Вільні», що дозволяє переглянути вільні для прокату на вказану дату велосипеди, «Моделі», що дозволяє переглянути всі моделі, які на задану дату доступні в прокаті, «Усі», що дозволяє переглянути всі наявні моделі, «Сьогодні», що дозволяє переглянути перелік велосипедів за обраною моделлю в наявності на поточний день;

- «Обладнання», який дозволяє переглянути все наявне обладнання (кнопка «Усе»), доступне на задану дату («Дата») і на поточний момент («Зараз»);

- «Обслуговування», який дозволяє переглянути перелік велосипедів, які мають проходити технічне обслуговування у даний момент;

- «Замовлення», який дозволяє обрати потрібний стан замовлення з випадającego переліку, задати пункт прокату, для якого потрібно вивести замовлення (або той, де працює менеджер, або за усіма одразу), окрім того для передзамовлень зокрема можна задати дату і час, після чого за допомогою кнопки «Відобразити» можна у таблиці нижче відобразити всі замовлення, які відповідають вказаним параметрам, окрім того за передзамовленнями можна підтвердити або скасувати замовлення, а за іншими станами відповідно завершити успішно, неуспішно, зафіксувати повернуті велосипеди, якщо були повернуті не всі.

ВЕЛОСИПЕДИ
ОБЛАДНАННЯ
ОБСЛУГОВУВАННЯ
ЗАМОВЛЕННЯ
ПРОФІЛЬ
Оформити Замовлення

Замовлення

Передзамовлення

Власний пункт

21.11.2021

11:00

Відобразити

Користувач

Номер телефону

Введіть номер

Номер	Створення	Клієнт	Вартість	Завершення		
57	21.11.2021 10:17	Іванов Борис	220 грн.	22.11.2021 11:00	Підтвердити	Скасувати
56	21.11.2021 09:52	Рябчун Дмитро	350 грн.	21.11.2021 18:00	Підтвердити	Скасувати
55	20.11.2021 21:23	Лесик Катерина	450 грн.	21.11.2021 14:00	Підтвердити	Скасувати
50	20.11.2021 19:14	Петров Андрій	220 грн.	22.11.2021 11:00	Підтвердити	Скасувати
48	20.11.2021 18:38	Музейко Іван	100 грн.	21.11.2021 16:00	Підтвердити	Скасувати

Рисунок 5.5 – Керування замовленнями менеджером пункту прокату

Якщо авторизувався адміністратор, то йому буде доступне широке меню, що включає пункти «Попит» (рис. 5.6), «Тарифи», що дозволяє додавати, редагувати, вилучати поточні використовувані тарифи, «Пункти», що дозволяє додавати, вилучати, редагувати пункти прокату, «Велосипеди», що дозволяє керувати наявним набором велосипедів за потреби, «Обладнання», що дозволяє керувати наявним набором обладнання в прокаті, «Обслуговування», що дозволяє керувати даними про технічне обслуговування велосипедів, «Замовлення», що дозволяє керувати всіма наявними замовленнями. Таким чином усі пункти дозволяють адміністратору за потреби виконувати будь-які дії з наявними даними, а пункт «Попит» доступний тільки йому для роботи. Він дозволяє переглянути дані про попит на прокат велосипедів помісячно. Для цього потрібно спочатку обрати рік, за який потрібно переглянути дані, а далі натиснути на кнопку «Відобразити». Дані відображаються помісячно. За кожним місяцем відображається максимальний та середній попит. Якщо попит було спрогнозовано, то в

стовпці «Примітка» відображається повідомлення «прогноз». Для обчислених, а не прогнозованих значень дана примітка не відображається.

ПОПИТ	ТАРИФИ	ПУНКТИ	ВЕЛОСИПЕДИ	ОБЛАДНАННЯ	ОБСЛУГОВУВАННЯ	ЗАМОВЛЕННЯ	Профіль
Попит на прокат за рік							
2021		Відобразити					
Місяць	Максимальний попит	Середній попит	Примітка				
Січень	443	49					
Лютий	288	36					
Березень	918	102					
Квітень	890	140					
Травень	632	141					
Червень	541	153					
Липень	1019	181					
Серпень	463	133					
Вересень	611	151					
Жовтень	603	134					
Листопад	787	99	прогноз				
Грудень	412	48	прогноз				

Рисунок 5.6 – Перегляд попиту на прокат адміністратором

5.4 Експериментальне дослідження методу прогнозування попиту клієнтів прокату велосипедів

Для експериментального дослідження запропонованого методу було використано вибірку даних [20]. Вибірка включає дані про подорожі, виконані з січня 2010 року по жовтень 2021 року.

Кожен екземпляр вибірки даних складається з наступних параметрів:

- тривалість подорожі;
- дата і час початку подорожі;
- дата і час завершення подорожі;
- пункт прокату, звідки було видано велосипед;
- пункт прокату, куди було повернуто велосипед;
- номер велосипеда, прокат якого виконувався;
- тип користувача (має реєстрацію в прокаті чи ні та яким тарифом користується).

На основі цих даних було узагальнено значення максимального та середнього попиту за кожен місяць. Ґрунтуючись на запропонованому методі з використанням в основі моделей SARIMA (як було запропоновано в самому методі) та ARIMA, якою було замінено відповідні моделі SARIMA, було виконано прогнозування попиту клієнтів. Створення моделей було виконано на основі даних за період з січня 2010 року по грудень 2020 року. Дані за період з січня 2021 року по жовтень 2021 року були використані для оцінювання результатів прогнозування шляхом обчислення середньоквадратичної похибки.

Результати експериментального дослідження представлено в табл. 5.1.

Таблиця 5.1 – Результати експериментального дослідження методу прогнозування попиту клієнтів прокату велосипедів

Модель	Середньоквадратична похибка для максимального попиту	Середньоквадратична похибка для середнього попиту
ARIMA	62,83	23,19
SARIMA	47,95	14,72

Отримані результати продемонстрували практичну придатність запропонованого методу для розв’язання проблеми. Для прогнозування середнього попиту виявилось, що модель SARIMA дозволяє зменшити

середньоквадратичну похибку на 36,5 % порівняно з моделлю ARIMA, а для прогнозування максимального попиту – на 23,68 %.

5.5 Висновки за розділом 5

Описано взаємодію клієнтів, менеджерів пунктів прокату, менеджерів з технічного обслуговування та адміністраторів з розробленою програмною системою.

Проведене експериментальне дослідження підтвердило практичну придатність запропонованого методу для прогнозування попиту клієнтів прокату велосипедів.

ВИСНОВКИ

Виконано огляд основних принципів організації сучасних прокатів велосипедів, створення програмного забезпечення підтримки роботи прокату велосипедів, проаналізовано сучасні програмні системи підтримки роботи прокатів велосипедів.

Проведений аналіз методів і моделей прогнозування під час планування роботи прокату велосипедів дозволив виділити основні підходи, які використовуються для прогнозування попиту, що включають: лінійну регресію, випадкові ліси, регресію екстремального градієнтного бустингу, рекурентні нейронні мережі. Сформульована задача прогнозування попиту клієнтів прокату велосипедів у вигляді задачі прогнозування часового ряду дозволила виділити набір відповідних моделей, що включає ARIMA, SARIMA, LSTM, DeepAR, N-BEATS, Temporal fusion transformer.

Наукова новизна роботи полягає в запропонованому методі прогнозування попиту клієнтів прокату велосипедів, який характеризується використанням моделі SARIMA для забезпечення прогнозування, що дозволяє отримувати основні показники попиту клієнтів для прийняття рішень про планування роботи прокату велосипедів.

Виконано вибір мови програмування PHP, що забезпечувала безпосередньо розробку вебзастосунку, і мови програмування Python, що забезпечувала роботу з часовими рядами під час реалізації запропонованого методу, для реалізації програмної системи. Визначено функціональні вимоги до програмної системи зі створенням відповідних діаграм прецедентів. Визначено структуру бази даних. Описано основні розроблені класи.

Розроблена програмна система може використовуватися для підтримки роботи прокату велосипедів щодо фіксації виконуваних робіт та планування майбутньої діяльності, практичну придатність щодо чого підтвердило проведене експериментальне дослідження.

Усі поставлені завдання на роботу виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Global Bike Rental Market to Reach USD 4.00 Billion by 2025 : Hexa Research [Electronic resource]. – Access mode : <https://www.prnewswire.com/news-releases/global-bike-rental-market-to-reach-usd-4-00-billion-by-2025-hexa-research-300816295.html>.
2. Ayadi, H. An exploratory study of the bike rental system [Electronic resource] / Hana Ayadi, Alaeddine Zouari, Nadia Hamani // 7th IEEE International Conference on Advanced Logistics & Transport (ICALT 2019). – Access mode : https://www.researchgate.net/publication/334401657_An_exploratory_study_of_the_bike_rental_system.
3. Bike hire and what European cycling tourists want ? – Bike Rental Manager [Electronic resource]. – Access mode : <https://bikerentalmanager.com/2018/07/17/bike-hire-and-what-european-cycling-tourists-want/>.
4. Our Top Tips for Running a Bike Rental Shop [Electronic resource]. – Access mode : <https://www.dialmycalls.com/blog/top-7-bike-rental-shop-tips>.
5. 3 Common Bike Rental Problems and How yoyo Can Help [Electronic resource]. – Access mode : <https://bookingyoyo.com/3-common-bike-rental-problems-and-how-yoyo-can-help/>.
6. Homepage | Easy To Use Bike Rental Management Software | yoyo [Electronic resource]. – Access mode : <https://bookingyoyo.com/>.
7. Software Feature List | Bike Rental Manager [Electronic resource]. – Access mode : <https://bikerentalmanager.com/feature-list/>.
8. Feng, Y. L. A Forecast for Bicycle Rental Demand Based on Random Forests and Multiple Linear Regression [Text] / You Li Feng, Shan Shan Wang // ICIS 2017. – Wuhan, China, 2017. – Pp. 101-105.
9. Clowe, B. Predicting Bike Rental Demand [Electronic resource] /

B. Clowe. – Access mode : <https://medium.com/@benclowe/predicting-bike-rental-demand-in-seoul-7f1e61d087bc>.

10. Lu, E. H.-C. Rental Prediction in Bicycle-Sharing System Using Recurrent Neural Network [Text] / Eric Hsueh-Chan Lu, Zhan-Qing Lin // IEEE Access. – 2020. – Volume 8. – Pp. 92262-92274.

11. Liu, X. Excess demand prediction for bike sharing systems [Electronic resource] / Xin Liu, Konstantinos Pelechrinis // PLOS One. – 2021. – 16 (6). – Access mode : <https://doi.org/10.1371/journal.pone.0252894>.

12. DeepAR Forecasting Algorithm [Electronic resource]. – Access mode : <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html>.

13. N-BEATS : Neural basis expansion analysis for interpretable time series forecasting [Electronic resource] / Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, Yoshua Bengio. – Access mode : <https://arxiv.org/abs/1905.10437>.

14. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting [Electronic resource]. – Access mode : <https://arxiv.org/abs/1912.09363>.

15. Regunath, G. 10 Incredibly Useful Time Series Forecasting Algorithms [Electronic resource] / G. Regunath and D. Lantos. – Access mode : <https://www.advancinganalytics.co.uk/blog/2021/06/22/10-incredibly-useful-time-series-forecasting-algorithms>.

16. Hyndman, R. J. Forecasting : Principles and Practice [Text] / Rob J. Hyndman and George Athanasopoulos. – Heathmont : OTexts, 2013. – 292 p.

17. Sarkar, S. Time Series Forecasting and Analysis : ARIMA and Seasonal-ARIMA [Electronic resource] / S. Sarkar. – Access mode : <https://medium.com/analytics-vidhya/time-series-forecasting-and-analysis-arima-and-seasonal-arima-cacaf61ae863>.

18. Seasonal ARIMA models [Electronic resource]. – Access mode : <https://online.stat.psu.edu/stat510/lesson/4/4.1>.

19. Short, Medium and Long-Term Forecasting – Marketing

Management [Electronic resource]. – Access mode :
[https://www.wisdomjobs.com/e-university/marketing-management-tutorial-294/
short-medium-and-long-term-forecasting-9586.html](https://www.wisdomjobs.com/e-university/marketing-management-tutorial-294/short-medium-and-long-term-forecasting-9586.html).

20. System Data | Capital Bikeshare [Electronic resource]. – Access
mode : <https://www.capitalbikeshare.com/system-data>.

ДОДАТОК А
Технічне завдання

Вступ

Прокат велосипедів, підтримка діяльності якого передбачається в роботі, повинен належати до стаціонарних прокатів з потенційною наявністю декількох пунктів прокату.

A.1 Підстави для розробки

Завдання на кваліфікаційну роботу магістра визначене за темою “Дослідження та програмна реалізація методів прогнозування під час планування роботи прокату велосипедів” і затверджене наказом № 435 від 5 листопада 2021 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програмна система призначена для організації роботи з відслідковування і керування станом велосипедів і обладнання, наявного в прокаті, та відслідковування і керування станом замовлень прокату на всіх їх етапах з подальшою підтримкою прийняття рішень.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Функціональні характеристики, що повинна забезпечувати програма, включають:

- внесення інформації про велосипеди, наявні в прокаті;
- внесення інформації про обладнання, яке надається прокатом;
- внесення тарифів на прокат велосипедів;
- внесення тарифів на прокат обладнання;
- внесення пунктів прокату велосипедів;

- оформлення замовлення прокату велосипеда;
- оформлення замовлень групових прокатів велосипедів;
- оформлення технічного обслуговування велосипедів;
- визначення вартості замовлення прокату;
- оформлення повернення велосипеда;
- керування замовленням прокатом велосипеда;
- перегляд активних замовлень прокатів велосипедів;
- перегляд замовлень прокатів велосипедів за вказаною датою;
- перегляд вільних для прокату велосипедів;
- перегляд вільного для прокату обладнання;
- перегляд доступних для прокату моделей велосипедів;
- перегляд велосипедів, які потребують обслуговування;
- перегляд складу велосипедів за моделлю;
- пошук замовлення прокату велосипеду;
- перегляд замовлень прокатів за клієнтом;
- перегляд незавершених вчасно замовлень прокатів;
- перегляд замовлень прокатів, які мають завершуватися у поточний день;
- завантаження даних для створення моделі прогнозування;
- прогнозування попиту клієнтів прокату велосипедів на місяць.

A.3.2 Вимоги до надійності

Для забезпечення надійного функціонування програмного забезпечення необхідна постійна підтримка віддалених запитів користувачів як через локальну мережу, так і через мережу Інтернет. Для цього програма має розташовуватися на виділеному для цього сервері, який повинен функціонувати постійно і обробляти запити до вебсервера та сервера бази даних.

А.3.3 Вимоги до складу та параметрів технічних засобів

Технічні засоби, за допомогою яких організовується підтримка роботи серверної частини програмної системи, повинні відповідати наступним вимогам:

- процесор: щонайменше два ядра, мінімальна частота – 2 ГГц;
- оперативний запам'ятовуючий пристрій – не менше 4 Гб;
- жорсткий диск: вільний простір не менше 1 Гб.

А.4 Порядок контролю та приймання

Контроль виконання роботи здійснюється за календарним планом роботи. Під час приймання роботи здійснюється перевірка відповідності визначеним функціональним характеристикам програми.

ДОДАТОК Б
Текст програми

```
<?php
```

```
class BManagerModel extends UserModel
{
    public function getActiveBookings($serviceId)
    {
        $conn = DBConn::getDB();
        $bookings = $conn->prepare('Select Booking.id as b_id,
            Booking.begin as b_begin,
            Booking.end_plan as b_endplan,
            Booking.price as b_price,
            Booking.payment as b_payment,
            Booking.bikes as b_bikes,
            Booking.status as b_status,
            U.name as u_name,
            M.name as m_name,
            Booking.service_begin as serv_id
        from Booking
        inner join User U on Booking.client_id = U.id
        inner join Manager M on Booking.manager1_id = M.id
        Where serv_id = :service_id AND b_status = 2
        order by b_id desc');
        $bookings->bindParam('service_id', $serviceId);
        $bookings->execute();
        $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
        return $bookings;
    }

    public function getAllActiveBookings()
    {
        $conn = DBConn::getDB();
        $bookings = $conn->prepare('Select Booking.id as b_id,
            Booking.begin as b_begin,
            Booking.end_plan as b_endplan,
            Booking.price as b_price,
            Booking.payment as b_payment,
            Booking.bikes as b_bikes,
            Booking.status as b_status,
            U.name as u_name,
            M.name as m_name
        from Booking
        inner join User U on Booking.client_id = U.id
```

```

        inner join Manager M on Booking.manager1_id = M.id
        Where b_status = 2
        order by b_id desc');
$bookings->execute();
$bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
return $bookings;
}

```

```

public function getAllActivePreBookings()
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status
        U.name as u_name
        from Booking
        inner join User U on Booking.client_id = U.id
        Where b_status = 1
        order by b_begin');
    $bookings->execute();
    $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
    return $bookings;
}

```

```

public function getPreBookingsByDate($pdate)
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status
        U.name as u_name
        from Booking
        inner join User U on Booking.client_id = U.id
        Where date(b_begin) = :begin_date AND b_status = 1
        order by b_id desc');
}

```

```

        $bookings->bindParam('begin_date', $pdate);
        $bookings->execute();
        $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
        return $bookings;
    }

    public function getPreBookingsByHour($pdate, $phour)
    {
        $conn = DBConn::getDB();
        $bookings = $conn->prepare('Select Booking.id as b_id,
            Booking.begin as b_begin,
            Booking.end_plan as b_endplan,
            Booking.price as b_price,
            Booking.payment as b_payment,
            Booking.bikes as b_bikes,
            Booking.status as b_status
            U.name as u_name
            from Booking
            inner join User U on Booking.client_id = U.id
            Where date(b_begin) = :begin_date AND
DATE_FORMAT(b_begin) = :begin_hour AND b_status = 1
            order by b_id desc');
        $bookings->bindParam('begin_date', $pdate);
        $bookings->bindParam('begin_hour', $phour);
        $bookings->execute();
        $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
        return $bookings;
    }

    public function getServPreBookingsByDate($serviceId, $pdate)
    {
        $conn = DBConn::getDB();
        $bookings = $conn->prepare('Select Booking.id as b_id,
            Booking.begin as b_begin,
            Booking.end_plan as b_endplan,
            Booking.price as b_price,
            Booking.payment as b_payment,
            Booking.bikes as b_bikes,
            Booking.status as b_status,
            Booking.service_begin as b_sb
            U.name as u_name
            from Booking
            inner join User U on Booking.client_id = U.id

```

```

        Where date(b_begin) = :begin_date AND b_status = 1 AND
b_sb = :service
        order by b_id desc');
$bookings->bindParam('begin_date', $pdate);
$bookings->bindParam('service', $serviceId);
$bookings->execute();
$bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
return $bookings;
}

public function getServPreBookingsByHour($serviceId, $pdate,
$phour)
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        Booking.service_begin as b_sb
        U.name as u_name
        from Booking
        inner join User U on Booking.client_id = U.id
        Where date(b_begin) = :begin_date AND
DATE_FORMAT(b_begin) = :begin_hour AND b_status = 1 AND b_sb = :service
        order by b_id desc');
    $bookings->bindParam('begin_date', $pdate);
    $bookings->bindParam('begin_hour', $phour);
    $bookings->bindParam('service', $serviceId);
    $bookings->execute();
    $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
    return $bookings;
}

public function getUnTimeBookings($serviceId)
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,

```

```

        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.name as u_name,
        M.name as m_name,
        Booking.service_end as serv_id,
        Booking.service_begin as serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M on Booking.manager1_id = M.id
    Where serv_id = :service_id AND b_status = 2 AND
date(b_endplan)>date(now())
        order by b_id desc');
$bookings->bindParam('service_id', $serviceId);
$bookings->execute();
$bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
return $bookings;
}

```

```

public function getAllUnTimeBookings()
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.name as u_name,
        M.name as m_name,
        Booking.service_end as serv_id,
        Booking.service_begin as serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M on Booking.manager1_id = M.id
    Where b_status = 2 AND date(b_endplan)>date(now())
        order by b_id desc');
    $bookings->execute();
    $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
    return $bookings;
}

```

```

public function getAllReadyBookings()
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.name as u_name,
        M.name as m_name,
        Booking.service_end as serv_id,
        Booking.service_begin as serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M on Booking.manager1_id = M.id
    Where b_status = 2 AND date(b_endplan)=date(now())
    order by b_id desc');
    $bookings->execute();
    $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
    return $bookings;
}

public function getReadyBookings($serviceId)
{
    $conn = DBConn::getDB();
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.name as u_name,
        M.name as m_name,
        Booking.service_end as serv_id,
        Booking.service_begin    serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M on Booking.manager1_id = M.id
    Where serv_id = :service_id AND b_status = 2 AND
date(b_endplan)=date(now())

```

```

        order by b_id desc');
$bookings->bindParam('service_id', $serviceId);
$bookings->execute();
$bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
return $bookings;
}

```

```

public function getUserBookings($client_name)
{

```

```

    $conn = DBConn::getDB();
    $client_name2 = $client_name."%";
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,
        Booking.end_plan as b_endplan,
        Booking.end_fact as b_endfact,
        Booking.creation as b_creation,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.name as u_name,
        M1.name as m1_name,
        M2.name as m2_name,
        Booking.service_end as serv_id,
        Booking.service_begin as serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M1 on Booking.manager1_id = M.id
    inner join Manager M2 on Booking.manager2_id = M.id
    Where u_name LIKE :client
    order by b_id desc');
    $bookings->bindParam('client', $client_name2);
    $bookings->execute();
    $bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
    return $bookings;
}

```

```

public function getUserBookingsByPhone($phone)
{

```

```

    $conn = DBConn::getDB();
    $client_name2 = "%".$client_name."%";
    $bookings = $conn->prepare('Select Booking.id as b_id,
        Booking.begin as b_begin,

```

```

        Booking.end_plan as b_endplan,
        Booking.end_fact as b_endfact,
        Booking.creation as b_creation,
        Booking.price as b_price,
        Booking.payment as b_payment,
        Booking.bikes as b_bikes,
        Booking.status as b_status,
        U.phone as u_phone,
        M1.name as m1_name,
        M2.name as m2_name,
        Booking.service_end as serv_id,
        Booking.service_begin as serv0_id
    from Booking
    inner join User U on Booking.client_id = U.id
    inner join Manager M1 on Booking.manager1_id = M.id
    inner join Manager M2 on Booking.manager2_id = M.id
    Where u_phone = :client_phone
    order by b_id desc');
$bookings->bindParam('client_phone', $phone);
$bookings->execute();
$bookings = $bookings->fetchAll(PDO::FETCH_ASSOC);
return $bookings;
}

    public function activatePreBooking($bookingId, $fullbooking,
    $bikes, $equipment)
    {
        unset($booking);
        $booking ['id'] = $bookingId;
        $booking ['status'] = 2;
        $booking ['begin'] = date("Y-m-d H:i:s");
        $booking ['service_begin'] = $this->
>getManagerService($fullbooking['manager1_id']);
        $booking ['manager1_id'] = $fullbooking['manager1_id'];
        $booking ['client_id'] = $fullbooking['client_id'];
        $booking ['end_plan'] = $fullbooking['end_plan'];
        $booking ['payment'] = $fullbooking['payment'];
        $booking ['notation'] = $fullbooking['notation'];
        $booking ['service_end'] = $fullbooking['service_end'];
        $booking ['bikerate_id'] = $this->getBikeRate($bikes,
        $booking ['begin'], $booking ['end_plan'])
        $booking ['equipment_rate_id'] = $this->
>getEquipmentRate($equipment, $booking ['begin'], $booking ['end_plan'])

```

```

        $booking ['bikes'] = count($bikes);
        $booking ['price'] = $this->calcBookingPrice($booking
['equipmentrate_id'], $booking ['bikerate_id'], $booking ['begin'],
$booking ['end_plan']);
        $conn = DBConn::getDB();
        $book = $conn->prepare('UPDATE Booking
SET status = :status,
client_id = :client_id,
begin = :begin,
end_plan = :end_plan,
payment = :payment,
notation = :notation,
service_begin = :service_begin,
service_end = :service_end,
manager1_id = :manager1_id,
bikerate_id = :bikerate_id,
equipmentrate_id = :equipmentrate_id,
bikes = :bikes,
price = :price
WHERE id = :id');
        $result = $book->execute($booking);
        if ($result) {
            foreach ($b as $bikes) {
                $result = $result && this-
>updateBookingBike($bookingId, $b);
            }
            foreach ($e as $equipment) {
                $result = $result && this-
>updateBookingEquipment($bookingId, $e);
            }
        }
        return $result;
    }

    public function activateBooking($fullbooking, $bikes,
$equipment, $client)
    {
        unset($booking);
        $booking ['status'] = 2;
        $booking ['begin'] = date("Y-m-d H:i:s");
        $booking ['creation'] = date("Y-m-d H:i:s");
        $booking ['service_begin'] = $this-
>getManagerService($fullbooking['manager1_id']);

```

```

        $booking ['manager1_id'] = $fullbooking['manager1_id']);
        $booking ['client_id'] = $fullbooking['client_id']);
        $booking ['end_plan'] = $fullbooking['end_plan']);
        $booking ['payment'] = $fullbooking['payment']);
        $booking ['notation'] = $fullbooking['notation']);
        $booking ['service_end'] = $fullbooking['service_end']);
        $booking ['bikerate_id'] = this->getBikeRate($bikes,
$booking ['begin'], $booking ['end_plan'])
        $booking ['equipmentrate_id'] = this-
>getEquipmentRate($equipment, $booking ['begin'], $booking ['end_plan'])
        $booking ['bikes'] = count($bikes);
        $booking ['price'] = $this->calcBookingPrice($booking
['equipmentrate_id'], $booking ['bikerate_id'], $booking ['begin'],
$booking ['end_plan']);
        $conn = DBConn::getDB();
        $book = $conn->prepare('INSERT INTO Booking
        {status, client_id, begin, end_plan, payment, notation,
service_begin, service_end, manager1_id, bikerate_id, equipmentrate_id,
bikes, price, creation}
        VALUES
(:status, :client_id, :begin, :end_plan, :payment, :notation, :service_be
gin, :service_end, :manager1_id, :bikerate_id, :equipmentrate_id, :bikes,
:price, :creation)');
        $result = $book->execute($booking);
        $id = $result;
        if ($result) {
            foreach ($b as $bikes) {
                $result = $result && this->insertBookingBike($id,
$bikes);
            }
            foreach ($e as $equipment) {
                $result = $result && this-
>insertBookingEquipment($id, $equipment);
            }
        }
        return $result;
    }

    public function closeBooking($bookingId, $fullbooking, $bikes,
$equipment, $manager, $mode)
    {
        unset($booking);
        $booking ['id'] = $bookingId;

```

```

        if ($mode) {
            $booking ['status'] = 3;
        }
        else {
            $booking ['status'] = 4;
        }
        $booking ['end_fact'] = date("Y-m-d H:i:s");
        $booking ['service_end'] = $this->getManagerService($manager);
        $booking ['manager2_id'] = $manager;
        $booking ['payment'] = $fullbooking['payment'];
        $booking ['notation'] = $fullbooking['notation'];
        $booking ['bikerate_id'] = this->getBikeRate($bikes,
$booking ['begin'], $booking ['end_fact'])
        $booking ['equipmentrate_id'] = this->getEquipmentRate($equipment, $booking ['begin'], $booking ['end_fact'])
        $booking ['price'] = $this->calcBookingPrice($booking
['equipmentrate_id'], $booking ['bikerate_id'], $booking ['begin'],
$booking ['end_fact']);
        $conn = DBConn::getDB();
        $book = $conn->prepare('UPDATE Booking
SET status = :status,
payment = :payment,
notation = :notation,
service_end = :service_end,
bikerate_id = : bikerate_id,
equipmentrate_id = :equipmentrate_id,
price = :price
WHERE id = :id');
        $result = $book->execute($booking);
        if ($result) {
            foreach ($b as $bikes) {
                $result = $result && this->updateBookingBike($bookingId, $b);
            }
            foreach ($e as $equipment) {
                $result = $result && this->updateBookingEquipment($bookingId, $e);
            }
        }
        return $result;
    }
}

```

```

public function updateBooking($bookingId, $bikes, $equipment)
{
    $result = TRUE;
    foreach ($b as $bikes) {
        $result = $result && this-
>updateBookingBikeStatus($bookingId, $b);
    }
    foreach ($e as $equipment) {
        $result = $result && this-
>updateBookingEquipmentStatus($bookingId, $e);
    }
    return $result;
}

public function getBikesByModel($modelId)
{
    $conn = DBConn::getDB();
    $bikes = $conn->prepare('Select Bike.id as b_id,
        Bike.production as b_begin,
        Bikenotation as b_endplan,
        Bike.last_operation as b_loper,
        Bike.next_operation as b_noper,
        Bike.bikemodel_id,
        Bike.status
        from Bike
        Where bikemodel_id = :model_id AND status = 1
        order by b_id desc');
    $bikes ->bindParam('model_id', $modelId);
    $bikes ->execute();
    $bikes = $bikes ->fetchAll(PDO::FETCH_ASSOC);
    $today = date("Y-m-d H:i:s");
    foreach ($bikes as $k => $val)
    {
        $bs = $conn->prepare('Select Booking.id as b_id,
            B.bike_id as b_bike,
            Booking.begin as b_begin,
            Booking.end_plan as b_eplan,
            Booking.end_fact as b_efact
            from Booking
            inner join BikeBooking B on b_id = B.booking_id
            Where b_bike = :b_id AND ((b_begin <= :tod AND b_eplan
            >= :tod) OR (b_begin <= :tod AND b_efact >= :tod))');
    }
}

```

```

        $bs->bindParam('b_id', $val['id']);
        $bs->bindParam('tod', $today);
        $bs->execute();
        if ($bs->rowCount()>0) {
            unset($bikes[$k]);
        }
    }
    return $bikes;
}

public function getFreeBikes($serviceId)
{
    $conn = DBConn::getDB();
    $bikes = $conn->prepare('Select Bike.id as b_id,
        Bike.production as b_begin,
        Bikenotation as b_endplan,
        Bike.last_operation as b_loper,
        Bike.next_operation as b_noper,
        Bike.bikemodel_id as bm,
        Bike.status,
        M.name,
        M.mark_id,
        Ma.name
    from Bike
    inner join Model on bm = M.id
    inner join Mark on Mark.id = maid
    order by b_id desc');
    $bikes ->execute();
    $bikes = $bikes ->fetchAll(PDO::FETCH_ASSOC);
    foreach ($bikes as $k => $val)
    {
        $bs = $conn->prepare('Select Booking.id as b_id,
            B.bike_id as b_bike,
            B.status as b_status,
            Booking.begin as b_begin,
            Booking.end_plan as b_eplan,
            Booking.end_fact as b_efact,
            Booking.service_end as b_eserv,
            Booking.status as b_s
        from Booking
        inner join BikeBooking B on b_id = B.booking_id
        Where b_bike = :b_id
        order by b_id desc');
    }
}

```

```

        $bs->bindParam('b_id', $val['id']);
        $bs->execute();
        $result = $bs->get_result();
        $row = $result->fetch_assoc();
        if (!($row ['b_eserv']==serviceId && $row
['b_status']==2 && !$row ['b_efact'] && $row ['b_s']==3)) {
            unset($bikes[$k]);
        }
    }
    return $bikes;
}

public function getBikesByModelDate($modelId, $pdate)
{
    $conn = DBConn::getDB();
    $bikes = $conn->prepare('Select Bike.id as b_id,
        Bike.production as b_begin,
        Bikenotation as b_endplan,
        Bike.last_operation as b_loper,
        Bike.next_operation as b_noper,
        Bike.bikemodel_id,
        Bike.status
    from Bike
    Where bikemodel_id = :model_id AND status = 1
    order by b_id desc');
    $bikes ->bindParam('model_id', $modelId);
    $bikes ->execute();
    $bikes = $bikes ->fetchAll(PDO::FETCH_ASSOC);
    foreach ($bikes as $k => $val)
    {
        $bs = $conn->prepare('Select Booking.id as b_id,
        B.bike_id as b_bike,
        Booking.begin as b_begin,
        Booking.end_plan as b_eplan,
        Booking.end_fact as b_efact
    from Booking
    inner join BikeBooking B on b_id = B.booking_id
    Where b_bike = :b_id AND ((b_begin <= :pdate AND
b_eplan >= :pdate) OR (b_begin <= :pdate AND b_efact >= :pdate))');
        $bs->bindParam('b_id', $val['id']);
        $bs->bindParam('pdate', $pdate);
        $bs->execute();
        if ($bs->rowCount()>0) {

```

```

        unset($bikes[$k]);
    }
}
return $bikes;
}

public function getFreeEquipment($serviceId)
{
    $conn = DBConn::getDB();
    $equipment = $conn->prepare('Select Equipment.id as e_id,
        Equipment.production as e_prod,
        Equipment.status as e_st,
        Equipmentnotation as e_not,
        Equipment.last_operation as e_lo,
        Equipment.next_operation as e_no
        from Equipment
        order by e_id desc');
    $equipment->execute();
    $equipment = $equipment->fetchAll(PDO::FETCH_ASSOC);
    foreach ($equipment as $k => $val)
    {
        $bs = $conn->prepare('Select Booking.id as b_id,
            E.equipment_id as e_eq,
            E.status as e_st,
            Booking.begin as b_begin,
            Booking.end_plan as b_eplan,
            Booking.end_fact as b_efact,
            Booking.service_end as b_eserv
            from Booking
            inner join EquipmentBooking E on b_id = E.equipment_id
            Where e_eq = :e_id
            order by e_id desc');
        $bs->bindParam('e_id', $val['id']);
        $bs->execute();
        $result = $bs->get_result();
        $row = $result->fetch_assoc();
        if (!($row ['b_eserv']==serviceId && $row ['e_st']==2
        && !$row ['b_efact'] && $row ['status']==3)) {
            unset($equipment[$k]);
        }
    }
    return $equipment;
}

```

```

public function getEquipmentByModelDate($modelId, $pdate)
{
    $conn = DBConn::getDB();
    $equip = $conn->prepare('Select Equipment.id as e_id,
        Equipment.production as e_prod,
        Equipment.status as e_st,
        Equipment.notation as e_not,
        Equipment.equipmentmodel_id as e_em
        from Equipment
        Where e_em = :model_id AND status = 1
        order by b_id desc');
    $equip ->bindParam('model_id', $modelId);
    $equip ->execute();
    $equip = $equip->fetchAll(PDO::FETCH_ASSOC);
    foreach ($equip as $k => $val)
    {
        $bs = $conn->prepare('Select Booking.id as b_id,
            E.equipment_id_id as e_eq,
            Booking.begin as b_begin,
            Booking.end_plan as b_eplan,
            Booking.end_fact as b_efact
            from Booking
            inner join EquipmentBooking E on e_eq = E.booking_id
            Where e_eq = :b_id AND ((b_begin <= :pdate AND b_eplan
            >= :pdate) OR (b_begin <= :pdate AND b_efact >= :pdate))');
        $bs->bindParam('b_id', $val['id']);
        $bs->bindParam('pdate', $pdate);
        $bs->execute();
        if ($bs->rowCount()>0) {
            unset($equip[$k]);
        }
    }
    return $equip;
}

public function getBikesForTech($serviceId)
{
    $conn = DBConn::getDB();
    $bikes = $conn->prepare('Select Bike.id as b_id,
        Bike.production as b_begin,
        Bike.notation as b_endplan,
        Bike.last_operation as b_lo,

```

```

        Bike.next_operation as b_no,
        Bike.bikemodel_id as bm,
        Bike.status,
        M.name,
        M.mark_id,
        Ma.name
    from Bike
    inner join Model on bm = M.id
    inner join Mark on Mark.id = maid
    Where b_no <= :tday
    order by b_id desc');
$today = date("Y-m-d H:i:s");
$bs->bindParam('tday', $today);
$bikes ->execute();
$bikes = $bikes ->fetchAll(PDO::FETCH_ASSOC);
return $bikes;
    }

}

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Кваліфікаційна робота магістра

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
МЕТОДІВ ПРОГНОЗУВАННЯ ПІД ЧАС
ПЛАНУВАННЯ РОБОТИ ПРОКАТУ
ВЕЛОСИПЕДІВ**

Виконав

студент групи КНТ-110м

Д. О. Рябчун

Керівник

к.т.н., доцент

В. М. Льовкін

2021

Рисунок В.1 – Слайд 1

Об'єкт дослідження – процес планування роботи прокату велосипедів.

Предмет дослідження – методи прогнозування під час планування роботи прокату велосипедів.

Мета роботи – розроблення програмного забезпечення підтримки роботи прокату велосипедів для прийняття рішень щодо планування його роботи на основі прогнозування попиту клієнтів.

Рисунок В.2 – Слайд 2

Завдання

- ✓ дослідження методів прогнозування під час планування роботи прокату велосипедів;
- ✓ розробка технічного завдання і вибір засобів розробки;
- ✓ проектування структури програмної системи;
- ✓ розробка програмної системи підтримки роботи прокату велосипедів.

3

Рисунок В.3 – Слайд 3

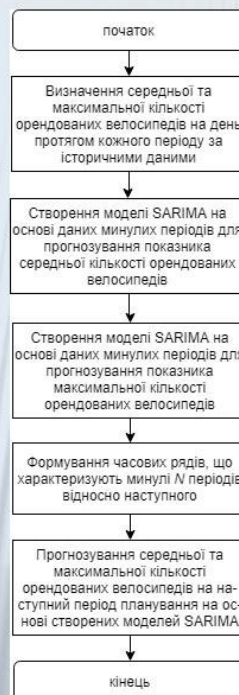
Аналіз методів прогнозування часових рядів для прогнозування попиту прокату велосипедів

Модель	Прогнозування часових рядів	Потреба великого обсягу даних	Період прогнозу	Характеристика
ARIMA	+	-	Зазвичай короткостроковий	Не враховує сезонність
SARIMA	+	-	Зазвичай короткостроковий	Враховує сезонність
LSTM	+	+	Придатна для довгострокового	Дозволяє вносити додаткові показники
DeepAR	+	+	Придатна для довгострокового	Розроблена Amazon, створює спільну модель для різних показників
N-BEATS	+	+	Придатна для довгострокового	Глибока нейронна архітектура
Temporal fusion transformer	+	+	Придатна для довгострокового	Розроблена Google архітектура на основі механізму уваги

4

Рисунок В.4 – Слайд 4

Метод прогнозування попиту клієнтів прокату велосипедів



5

Рисунок В.5 – Слайд 5

Вибір мови програмування

Критерій	Java	PHP	Python
Основне призначення	Загальне	Веброзробка	Загальне
Оптимізованість для веброзробки	+-	+	+-
Швидкість виконання коду	+	+-	-
Поетапна стабільність	-	+	+
Простота роботи з часовими рядами	+-	-	+
Невеликі витрати	+-	+	+

6

Рисунок В.6 – Слайд 6

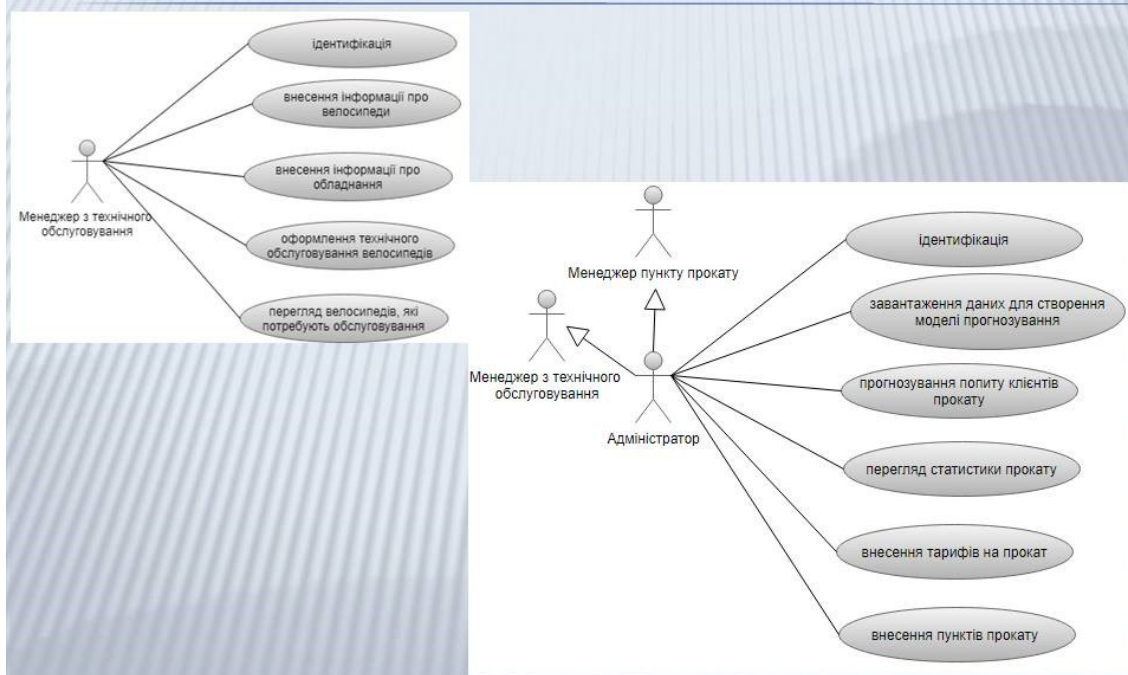
Діаграма прецедентів менеджера пункту прокату



7

Рисунок В.7 – Слайд 7

Діаграми прецедентів менеджера з технічного обслуговування та адміністратора



8

Рисунок В.8 – Слайд 8



Рисунок В.9 – Слайд 9



Рисунок В.10 – Слайд 10

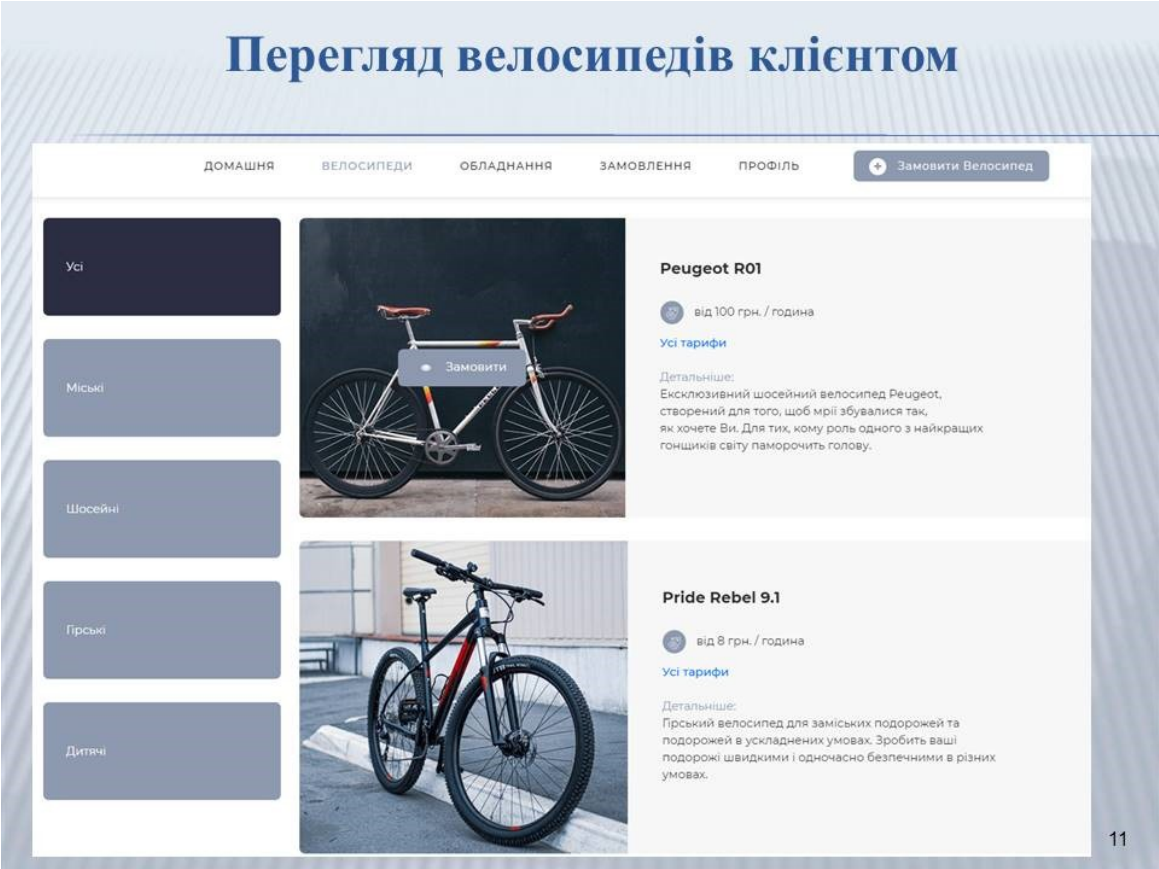


Рисунок В.11 – Слайд 11

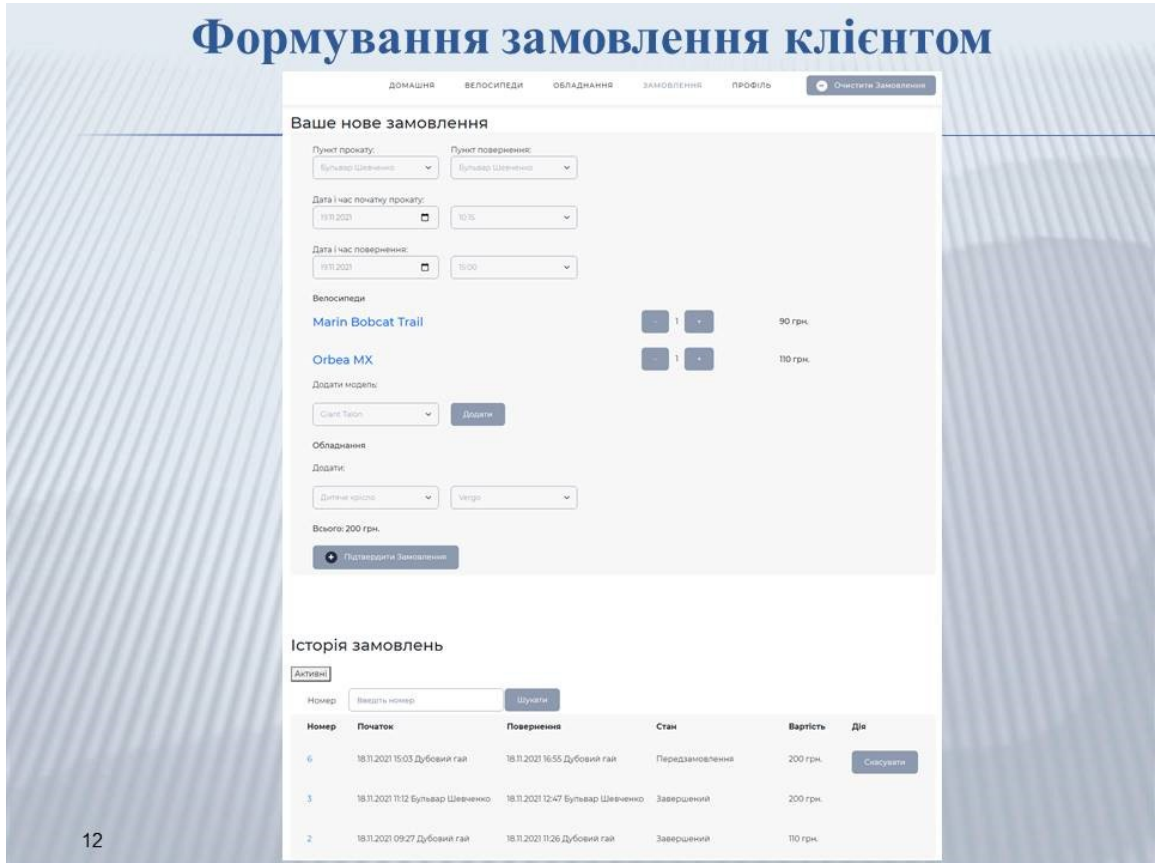


Рисунок В.12 – Слайд 12

Взаємодія менеджерів з вебзастосунком

ВЕЛОСИПЕДИ

ОБЛАДНАННЯ

ОБСЛУГОВУВАННЯ

ЗАМОВЛЕННЯ

ПРОФІЛЬ

Оформити Замовлення

Замовлення

Перезамовлення:

Власний пункт

21.11.2021

11:00

Відобразити

Користувач

Номер телефону

Введіть номер

Номер	Створення	Клієнт	Вартість	Завершення		
57	21.11.2021 10:17	Іванов Борис	220 грн.	22.11.2021 11:00	Підтвердити	Скасувати
56	21.11.2021 09:52	Рябун Дмитро	350 грн.	21.11.2021 18:00	Підтвердити	Скасувати
55	20.11.2021 21:23	Лесик Катерина	450 грн.	21.11.2021 14:00	Підтвердити	Скасувати
50	20.11.2021 19:14	Петров Андрій	220 грн.	22.11.2021 11:00	Підтвердити	Скасувати
48	20.11.2021 18:38	Музейко Іван	100 грн.			

ВЕЛОСИПЕДИ

ОБЛАДНАННЯ

ОБСЛУГОВУВАННЯ

ПРОФІЛЬ

Внести Обслуговування

Велосипеди, що очікують обслуговування

Номер	Велосипед	Вироблено	Останнє обслуговування	Наступне обслуговування	Дія
22	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
21	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
20	Pride Rebel 9.1	2020	-	20.11.2021	Керувати
17	Marin Bobcat Trail	2020	-	19.11.2021	Керувати
11	Orbea MX	2019	20.11.2020	20.11.2021	Керувати
10	Orbea MX	2019	20.11.2020	20.11.2021	Керувати

Рисунок В.13 – Слайд 13

Визначення попиту на прокат велосипедів

ПОПИТ

ТАРИФИ

ПУНКТИ

ВЕЛОСИПЕДИ

ОБЛАДНАННЯ

ОБСЛУГОВУВАННЯ

ЗАМОВЛЕННЯ

ПРОФІЛЬ

Попит на прокат за рік

2021

Відобразити

Місяць	Максимальний попит	Середній попит	Примітка
Січень	443	49	
Лютий	288	36	
Березень	918	102	
Квітень	890	140	
Травень	632	141	
Червень	541	153	
Липень	1019	181	
Серпень	463	133	
Вересень	611	151	
Жовтень	603	134	
Листопад	787	99	прогноз
Грудень	412	48	прогноз

Рисунок В.14 – Слайд 14

Результати експериментального дослідження методу прогнозування попиту клієнтів прокату велосипедів

Модель	Середньоквадратична похибка для максимального попиту	Середньоквадратична похибка для середнього попиту
ARIMA	62,83	23,19
SARIMA	47,95	14,72

15

Рисунок В.15 – Слайд 15

Висновки

Виконано огляд основних принципів організації сучасних прокатів велосипедів, створення програмного забезпечення підтримки роботи прокату велосипедів, проаналізовано сучасні програмні системи підтримки роботи прокатів велосипедів.

Проведений аналіз методів і моделей прогнозування під час планування роботи прокату велосипедів дозволив виділити основні підходи, які використовуються для прогнозування попиту, що включають: лінійну регресію, випадкові ліси, регресію екстремального градієнтного бустингу, рекурентні нейронні мережі.

Наукова новизна роботи полягає в запропонованому методі прогнозування попиту клієнтів прокату велосипедів, який характеризується використанням моделі SARIMA для забезпечення прогнозування, що дозволяє отримувати основні показники попиту клієнтів для прийняття рішень про планування роботи прокату велосипедів.

Виконано вибір мови програмування PHP, що забезпечувала безпосередньо розробку вебзастосунку, і мови програмування Python, що забезпечувала роботу з часовими рядами під час реалізації запропонованого методу, для реалізації програмної системи. Визначено функціональні вимоги до програмної системи зі створенням відповідних діаграм прецедентів. Визначено структуру бази даних. Описано основні розроблені класи.

Розроблена програмна система може використовуватися для підтримки роботи прокату велосипедів щодо фіксації виконуваних робіт та планування майбутньої діяльності, практичну придатність щодо чого підтвердило проведене експериментальне дослідження.

Усі поставлені завдання на роботу виконано.

16

Рисунок В.16 – Слайд 16