

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Програмна система для закладів дошкільної освіти з  
керуванням режимом дня та харчуванням дітей. Mobile Application  
\_\_\_\_\_ (тема)

Виконав:  
студент 4 курсу, групи ПЗП-20-3  
\_\_\_\_\_ Нестеренко В.В. \_\_\_\_\_  
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного  
забезпечення \_\_\_\_\_  
(код і повна назва спеціальності)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
Освітня програма Програмна інженерія \_\_\_\_\_  
(повна назва освітньої програми)

Керівник доц. кафедри ПІ Побіженко І,О. \_\_\_\_\_  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри \_\_\_\_\_

\_\_\_\_\_ З.В.Дудар \_\_\_\_\_  
(підпис) (прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук  
 Кафедра \_\_\_\_\_ програмної інженерії  
 Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський)  
 Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення  
 Тип програми \_\_\_\_\_ Освітньо-професійна  
 Освітня програма \_\_\_\_\_ Програмна Інженерія  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Нестеренко Віталію Вячеславовичу \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Програмна система для закладів дошкільної освіти з керуванням режимом дня та харчуванням дітей. Mobile Application  
 Затверджена наказом по університету від 20.05.2024р. № 471 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 14.06.2024
3. Вихідні дані до роботи Розробити мобільну програмну систему, яка надаватиме функції для закладів дошкільної освіти, які різняться в залежності від ролі, використовуючи мову програмування TypeScript, бібліотеки React Native і Expo.
4. Перелік питань, що потрібно опрацювати в роботі  
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проєктування програмного забезпечення, опис прийнятих програмних рішень, тестування, впровадження розробленого програмного забезпечення, висновки, додатки.

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	08.04.2024	<i>виконано</i>
2	Створення специфікації ПЗ	12.04.2024	<i>виконано</i>
3	Проектування ПЗ	15.04.2024	<i>виконано</i>
4	Розробка ПЗ	15.05.2024	<i>виконано</i>
5	Тестування ПЗ	17.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	01.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2024	<i>виконано</i>
8	Попередній захист	16.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	16.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	16.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	17.06.2024	<i>виконано</i>

Дата видачі завдання 8 квітня 2024р.

Студент (ка) \_\_\_\_\_  
(підпис)

\_\_\_\_\_ Нестеренко В.В.

Керівник роботи \_\_\_\_\_  
(підпис)

доц. кафедри ПІ Побіженко І.О.  
(посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 142 стор., 109 рис., 1 табл., 10 джерел.

ДІТИ, МОБІЛЬНИЙ ЗАСТОСУНОК, РОЗКЛАД, ШТУЧНИЙ ІНТЕЛЕКТ, EXPO, GEMINI, LLM, REACT NATIVE

Об'єкт розробки – мобільний застосунок з керуванням режимом дня та харчуванням дітей.

Мета розробки – створення мобільного застосунку, який генеруватиме розклад дня для дітей і дозволить відмічати присутність дітей.

Метод рішення – мова програмування TypeScript, бібліотека React Native, фреймворк Expo, середовище розробки Visual Studio Code, емулятор Expo Go.

У результаті розробки було створено мобільний застосунок, котрий за допомогою штучного інтелекту генерує розклад дня і харчування дітей, враховуючи їх вік, а також дозволяє відмічати відвідування дітей за допомогою QR-кодів, відмічати присутність дітей і переглядати історію їх відвідувань.

CHILDREN, MOBILE APPLICATION, SCHEDULE, ARTIFICIAL INTELLIGENCE, EXPO, GEMINI, LLM, REACT NATIVE

The object of development is a mobile application for managing children's daily routine and nutrition.

The goal of the development is to create a mobile application that will generate a daily schedule for children and allow children to be noted.

Solution method – TypeScript programming language, React Native library, Expo framework, Visual Studio Code development environment, Expo Go emulator.

As a result of the development, a mobile application was created, which uses artificial intelligence to generate a schedule of children's day and meals, taking into

account their age, and also allows you to mark the visits of children using QR codes, mark the presence of children and view the history of their visits.

Я, Нестеренко Віталій Вячеславович, студент гр. ПЗПІ-20-3, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система для закладів дошкільної освіти з керуванням режимом дня та харчуванням дітей. Mobile частина», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	8
1.1 Аналіз предметної галузі.....	8
1.2 Виявлення та вирішення проблем.....	9
1.3 Постановка задачі.....	19
2 Формування вимог до програмної системи.....	21
2.1 Збір та аналіз вимог.....	21
2.2 Формування вимог.....	22
2.2.1 Функціональні вимоги.....	22
2.2.2 Нефункціональні вимоги.....	26
3 Архітектура та проектування програмного забезпечення.....	28
3.1 UML проектування ПЗ.....	28
3.2 Проектування архітектури ПЗ.....	36
3.3 Проектування структури зберігання даних.....	39
3.4 Створення UI/UX.....	40
4 Опис прийнятих рішень.....	43
4.1 Розробка клієнтської частини.....	43
4.2 Опис інтерфейсу програмного забезпечення.....	54
5 Тестування розробленого програмного забезпечення.....	80
6 Впровадження програмного забезпечення.....	85
Висновки.....	87
Перелік джерел посилання.....	88
Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	89
Додаток Б Слайди презентації.....	90
Додаток В Специфікація програмного забезпечення.....	98
Додаток Г ER-діаграма застосунку «ChildWell».....	136
Додаток Д Приклад програмного коду (компонент сканування QR-коду).....	137
Додаток Е Диплом за друге місце на конференції.....	140
Додаток Ж Подані на міжнародний форум тези.....	141

## ВСТУП

Умови в яких зростає дитина, якими справами вона займається, чому навчається, коли і як часто їсть і відпочиває – все це закладає фундамент для майбутнього поки що маленької людини. Ця сформована основа вплине на її звички і здоров'я, що в свою чергу вплине успішності в більшості сфер життя в майбутньому – від в кар'єри до особистого життя [1].

Визначення занять і справ, якими має займатися дитина для побудови правильного фундаменту її життя, а також створювати з них правильний розпорядок дня, враховуючи вікові потреби дитини (їжа, сон, ігри) є надзвичайно складною справою, що вимагає великих витрат часу. В такому разі застосунок для генерації розкладу для дітей є вирішенням цієї амбітної і відповідальної задачі. Також це надасть батькам актуальну інформацію про діяльність і благополуччя їхніх дітей.

Метою роботи є розробка мобільного застосунку, який на базі штучного інтелекту генерує розклад занять і харчування для груп дітей в закладах дошкільної освіти, враховуючи їх вікові потреби, такі як відпочинок та прийом їжі. Також застосунок дозволяє вести облік відвідувань дітей, підвищуючи їх безпеку, і автоматизує оплату за відвідування дитячих садочків.

Завдання роботи – створення мобільного застосунку для закладів дошкільної освіти, який генерує розклад занять та харчування для дітей, спрощує взаємодію між вихователями та батьками. Застосунок дозволить батькам зручно подавати заявки на зарахування дитини до дитячого садка, відстежувати присутність дитини за допомогою сканування QR-кодів та здійснювати оплату рахунків, виписаних закладом.

Для розробки програми використовувалась мова програмування TypeScript і бібліотека React Native у зв'язці з фреймворком Expo. Для емуляції мобільного застосунку було використано Expo Go. Visual Studio Code – середовище розробки.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз предметної галузі

Сьогодні неможливо уявити світ без таких революційних технологій, як штучний інтелект і машинне навчання, які вже укорінилися в більшість сфер нашого життя, особливо в освіту. Мобільні застосунки поступово інтегрують штучний інтелект для вирішення різноманітних завдань, зокрема в управлінні дошкільними навчальними закладами. Завдяки цим технологіям, процеси оптимізуються, покращуючи взаємодію між закладами дошкільної освіти, вихователями і батьками. Цей аналіз зосереджується на огляді сучасних трендів та інновацій у галузі дошкільної освіти як в Україні, так і за кордоном.

З випуском великих мовних моделей (ВММ) з'явилася низка нових трендів, що можуть покращити сферу дошкільної освіти [2].

По перше, ВММ використовуються для оптимізації розкладу дня і харчування, вони аналізують інформацію про особливості груп дітей, наприклад, їх вік, режим дня та інші параметри, щоб створювати більш точні та ефективні розклади.

По-друге, ВММ генерують адаптовані навчальні та розважальні заняття, знову ж таки враховуючи вікові особливості та інтереси груп дітей.

По-третє, штучний інтелект аналізує дані про успішність та поведінку дітей, що дозволяє виявляти проблеми та відповідно адаптувати навчальний процес для їх швидкого вирішення.

В Україні більшість задач, пов'язаних з управлінням дошкільними закладами, все ще виконуються вручну, наприклад, планування розкладу занять, облік відвідуваності дітей, а також процеси виставлення та оплати рахунків. Виконання цих задач вручну не тільки збільшує навантаження на працівників закладів, але й підвищує ризик помилок через людський фактор, що може призвести до помилок у веденні записів та фінансових розрахунках.

Наявність помилок може негативно позначитися як і на якості роботи закладів дошкільної освіти, так і на безпеці дітей, оскільки неправильне планування розкладу або помилки в обліку присутності можуть призвести до надзвичайних

ситуацій. Крім того, неефективне управління фінансами може завдати шкоди як репутації, так і економічному стану закладів.

Зростаючий попит на автоматизовані рішення, що мають в собі управління розкладом, облік відвідуваності дітей та автоматизацію платежів, спонукає розробників до створення застосунків, що вирішили б ці проблеми. Це не тільки підвищить ефективність управління освітніми закладами, але й значно зменшить ризики, пов'язані з людським фактором.

На міжнародному рівні ринок мобільних застосунків для дошкільних закладів демонструє високу ступінь інноваційності та інтеграції новітніх технологій, таких як штучний інтелект. Застосунки дозволяють батькам не тільки слідкувати за розкладом своїх дітей, а й виконувати платежі та відслідковувати присутність, стають стандартом у багатьох країнах. Це покращує процеси управління закладами дошкільної освіти, а також покращує рівень задоволеності та безпеки серед батьків та працівників закладів.

Аналіз ринку мобільних застосунків для дошкільних закладів в Україні та за кордоном підкреслює зростаючу потребу в комплексних і технологічно вдосконалених рішеннях, особливо в Україні. Розвиток таких застосунків, що включають широкий спектр функціональностей від генерації розкладу до системи платежів і обліку присутності, відкриває нові можливості для оптимізації роботи дошкільних закладів.

## 1.2 Виявлення та вирішення проблем

Провівши попередній аналіз, можна виділити наступні проблеми, з якими стикаються заклади дошкільної освіти і батьки дітей, які ходять в такі заклади:

- ручне складання розкладу часто робиться без урахування вікових потреб дітей;
- ручний облік присутності дітей допускає помилки та є витратним по часу;
- ручне управління платежами може вести до фінансових помилок і зловживань;
- складність подачі заявок в заклади дошкільної освіти;

- відсутність можливості для батьків переглянути розклад її дитини;
- відсутність уніфікованого місця управління всіма дітьми (актуально для батьків з більше ніж однією дитиною).

Ця робота має на меті створити мобільний додаток, який за допомогою штучного інтелекту автоматично складатиме розклади занять і харчування для дітей дошкільного віку, враховуючи їхні потреби в прийомі їжі та відпочинку. Крім того, додаток забезпечить ведення точного обліку присутності дітей, що сприятиме підвищенню їх безпеки, та автоматизує процес оплати за послуги дитячих садочків. Створення мобільного застосунку дозволить закладам дошкільної освіти зменшити кількість роботи, покращити якість створення розкладу занять, підвищити безпеку обліку відвідувань дітей і автоматизувати оплату рахунків.

Нова система вступить у конкуренцію з іншими застосунками для закладів дошкільної освіти. На ринку на даний момент присутні лише кілька програм, які можна розглядати як прямих конкурентів, проте багато з них мають недоліки, які вирішує нова мобільна система. Особлива увага приділена унікальним функціям розроблюваної мобільної системи «ChildWell».

Щоб добре проаналізувати конкурентів у сфері мобільних додатків для дошкільних закладів, треба звернути увагу на кілька важливих речей, таких як функції застосунків, їх поширення по світу, їх вигляд і зручність користування.

Перша система, яка конкурує з «ChildWell» є «Brightwheel» (див. рис. 1.1) - це мобільний застосунок, що дозволяє управляти всіма аспектами дошкільної освіти, включно з відвідуванням, комунікаціями з батьками, оплатою, і плануванням занять. «Brightwheel» забезпечує інтегроване середовище для вихователів та адміністрації закладів дошкільної освіти, що дозволяє ефективно керувати щоденними операціями. Завдяки функціям відстеження відвідуваності діти можуть реєструватися при вході та виході за допомогою QR-кодів, що спрощує процес для батьків і вихователів. Комунікації з батьками включають обмін повідомленнями, сповіщення про важливі події та надання звітів про розвиток дітей.

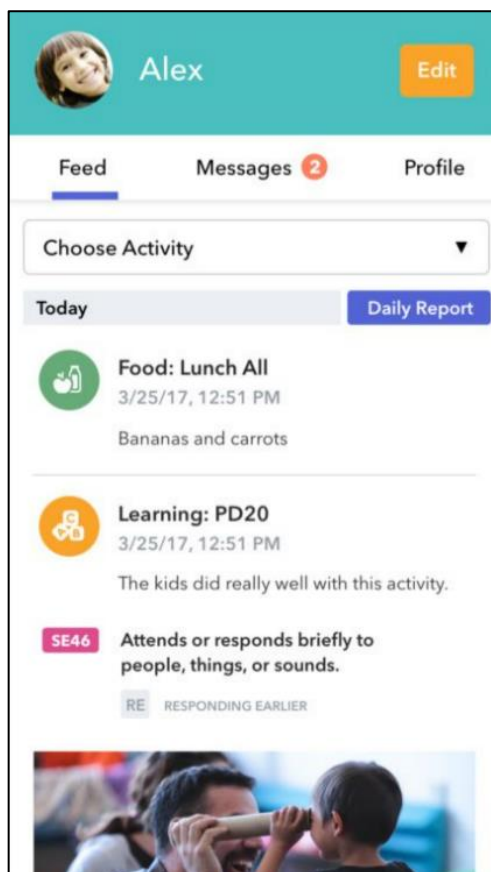


Рисунок 1.1 – Інтерфейс мобільного застосунку «Brightwheel» (за даними [3])

Мобільний застосунок «Brightwheel» пропонує наступний функціонал:

- перегляд розкладу дітей;
- управління документацією;
- реєстрація присутності дітей за допомогою QR-кодів;
- месенджер для спілкування вихователів з батьками;
- автоматизація оплати рахунків.

Проте, головним недоліком застосунку «Brightwheel» є відсутність можливості генерувати розклад занять за допомогою штучного інтелекту. Ця функція могла б значно підвищити ефективність планування освітнього процесу до потреб груп дітей, автоматизуючи та оптимізуючи процес складання розкладу дня і харчування, що є ключовою перевагою нашого застосунку. Хоча «Brightwheel» є потужним інструментом, його інтерфейс може здатися перевантаженим та складним для нових користувачів, особливо порівняно з більш інтуїтивно зрозумілим дизайном нашого застосунку. Також «ChildWell» в працюватиме по

всьому світу, а «Brightwheel» працює тільки в США і підтримує єдину мову – англійську, тим часом як «ChildWell» підтримує окрім англійської ще й українську.

Далі розглянемо наступного конкурента під назвою «Procare» (див. рис. 1.2).

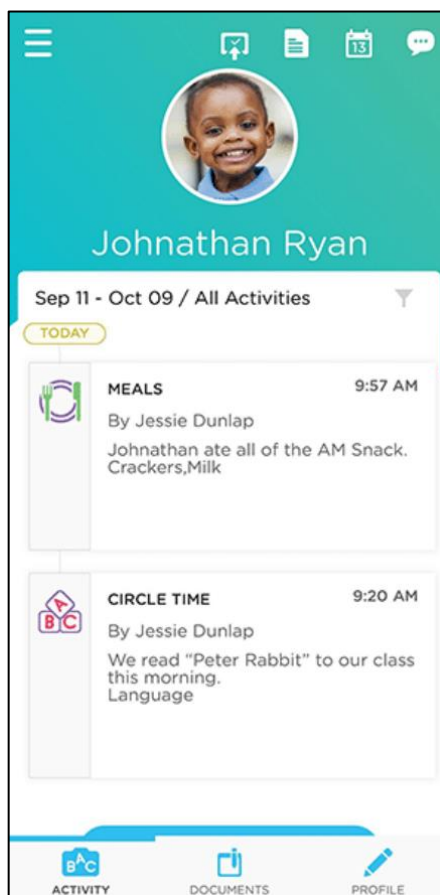


Рисунок 1.2 – Інтерфейс мобільного застосунку «Procure» (за даними [4])

Мобільний застосунок «Procure» є широко використовуваним застосунком у сфері дошкільної освіти, який надає ряд функцій для управління дитячими центрами, школами та таборами. До його переваг можна віднести наступний функціонал:

- управління харчуванням;
- перегляд освітнього матеріалу;
- реєстрація присутності дітей;
- детальний перегляд інформації про дитину, її вага, зріст тощо.

Цей застосунок має меншу кількість переваг, порівняно з «Brightwheel», поступаючись відсутністю месенджера і сканування QR-кодів. На відміну від нашого мобільного застосунку, «Procure» не використовує штучний інтелект для

генерування розкладу дня та харчування, що зменшує його здатність до адаптації під потреби груп дітей. Хоча «Procare» дозволяє вести облік відвідуваності дітей, він не використовує технологію QR-кодів, щоб пришвидшити і зробити цей процес більш безпечним, як це зроблено в нашому застосунку «ChildWell». Інтерфейс «Procare» має тільки англійську мову, що також можна віднести до недоліку.

Наступний конкурент «Remini» (див. рис. 1.3) є популярним застосунком для управління дошкільними закладами, який надає широкий спектр функцій для відстеження освітнього прогресу та зв'язку з батьками.

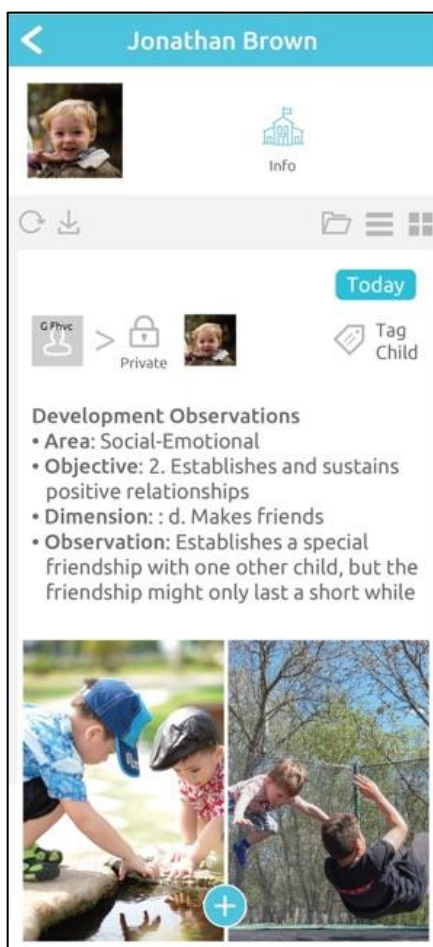


Рисунок 1.3 – Інтерфейс мобільного застосунку «Remini» (за даними [5])

До його переваг відносяться:

- можливість вихователям публікувати фото- і відео- звіти прогулянок і досягнень дітей;
- записки про поведінку дітей;
- перегляд похвал від вихователів;

- перегляд завдань дітей;
- реєстрація присутності дітей;
- перегляд портфоліо дитини;
- журнали здоров'я та харчування.

Проте до найбільшого недоліку необхідно віднести складність інтерфейсу, адже він є перевантаженим і менш інтуїтивним, особливо при спробі знайти специфічну інформацію або функцію. І порівнюючи із нашим застосунком, «Remini» також не використовує штучний інтелект для генерації розкладу занять.

Наступний конкурент «Nimata» (див. рис. 1.4) виділяється своїми деталізованими освітніми звітами. забезпечує високий рівень взаємодії з батьками, надаючи їм регулярні оновлення про активності, успіхи та повсякденне життя їхніх дітей, що допомагає створювати міцні зв'язки між сім'єю та закладом.

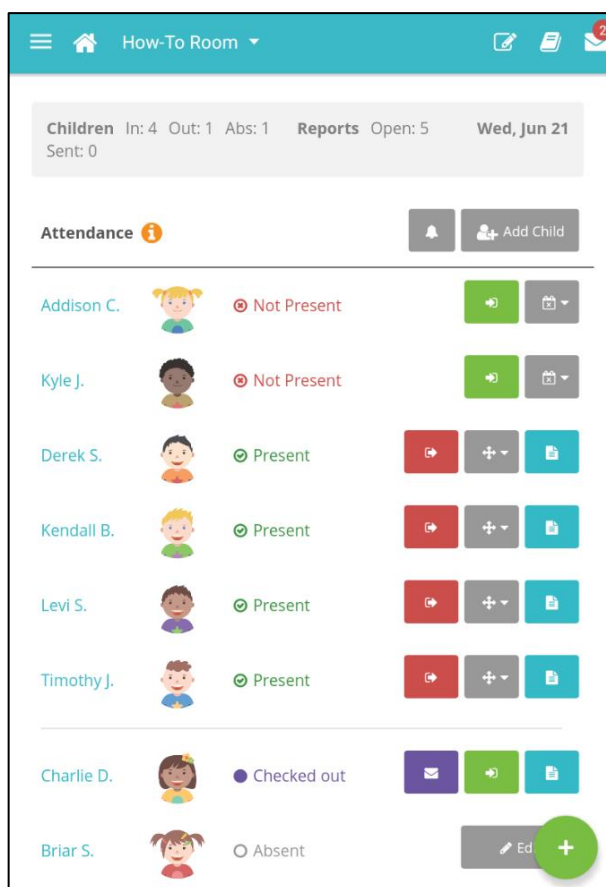


Рисунок 1.4 – Інтерфейс мобільного застосунку «Nimata» (за даними [6])

До інших переваг мобільного застосунку «Nimata» можна віднести:

- підтримка відгуків і коментарів;
- інтуїтивно зрозумілий інтерфейс;
- модуль планування заходів;
- персоналізовані нагадування.

Тим не менш, не дивлячись на численні переваги, застосунок «Nimata» також має деякі потенційні недоліки, які важливо врахувати, а саме висока вартість підписки, складність інтерфейсу і відсутність підтримки декількох мов.

І останній, тим не менш найважливіший, конкурент нашого застосунку є «Twigsee» (див. рис. 1.5).

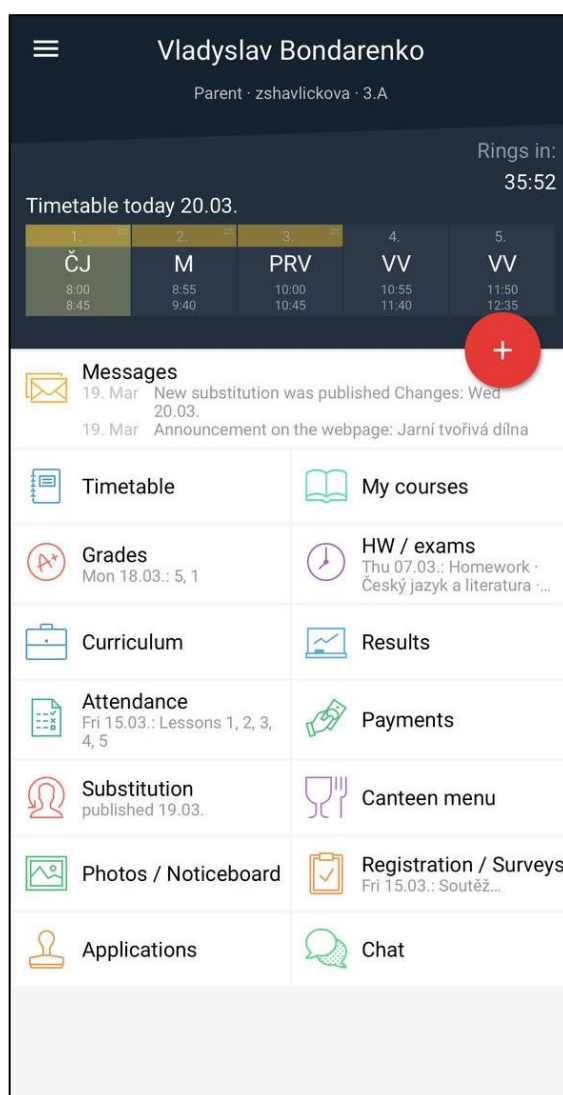


Рисунок 1.5 – Інтерфейс мобільного застосунку «Twigsee» (рисунок виконано самостійно)

«Twigsee» є сучасним мобільним застосунком для управління дошкільними закладами, який має кілька унікальних переваг, які можуть виділити його на тлі інших додатків, включаючи наш. До таких унікальних переваг відносяться:

- інтеграція з соціальними мережами;
- функція «Спільноти»;
- підтримка багатомовності;
- інтерактивні звіти;
- перегляд оцінок дітей;
- перегляд домашніх завдань.

«Twigsee» надає інструменти для інтеграції з соціальними мережами, що дозволяє вихователям легко ділитися новинами, фото та оновленнями безпосередньо через платформу, підвищуючи їхню видимість і залученість батьків. Також застосунок розвиває концепцію спільнот, створюючи окремі групи всередині застосунку для батьків, де вони можуть спілкуватися, ділитися порадами, планувати спільні заходи та організовувати спільні проекти.

Щодо інтерактивних звітів, то «Twigsee» генерує детальні звіти про розвиток та поведінку дітей, що включають візуалізації та аналітику. Батьки можуть легко переглядати ці звіти, отримуючи важливу інформацію про прогрес їхніх дітей, що значно підвищує рівень залученості та інформованості. Ці звіти дозволяють батькам бути в курсі ключових моментів розвитку дитини, що сприяє більш ефективному партнерству між родиною та освітнім закладом.

Хоча застосунок «Twigsee» пропонує багато інноваційних функцій, вони даються недешево – через широкий спектр можливостей плата за користування платформою є вищою у порівнянні з більш простими аналогами. Це може бути проблемою для менш фінансово забезпечених закладів, які можуть не мати достатніх ресурсів для покриття витрат на використання такого комплексного рішення. Незважаючи на це, інвестиції в «Twigsee» можуть бути виправдані за рахунок підвищення якості освіти та зручності управління, що він забезпечує.

Після ретельного вивчення конкурентів, визначення їхніх сильних та слабких сторін, можна представити аналітичний звіт про мобільний застосунок «ChildWell» (див. рис. 1.6).

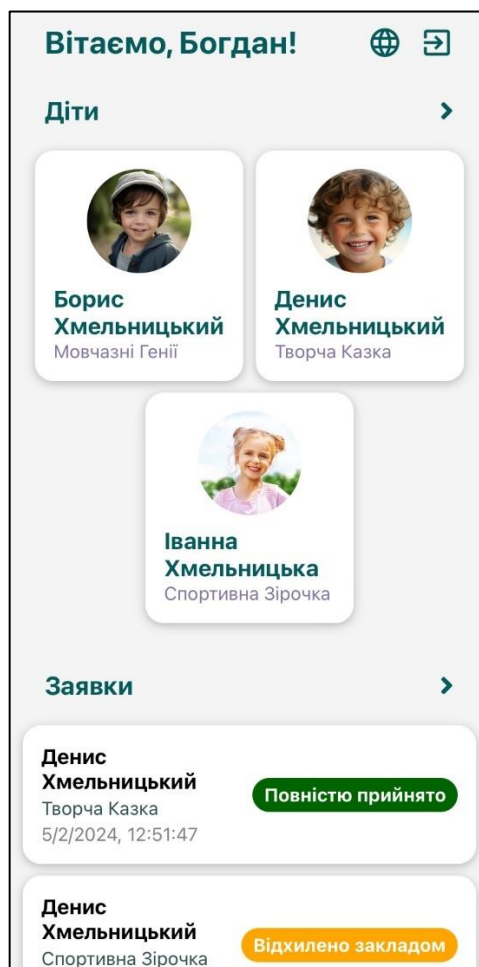


Рисунок 1.6 – Інтерфейс розроблюваного мобільного застосунку «ChildWell»  
(рисунок виконано самостійно)

Мобільний застосунок «ChildWell» має наступні функції в залежності від ролі користувача:

Адміністратор закладу дошкільної освіти має такі функції:

- перегляд закладів, до яких є доступ
- додавання нових активностей;
- налаштування існуючих активностей;
- прийом/відхилення заявок на вступ до дитячого садочку;
- перегляд вчителів дитячого садочку;

- перегляд дитячих груп садочку.

Батько або мати дитини:

- додавання і перегляд дітей;
- перегляд закладів для подачі заявки;
- перегляд і подача заявок у садки;
- прийом/відхилення заявок;
- перегляд розкладу дітей;
- генерація QR-коду для реєстрації присутності дитини;
- перегляд історії відвідувань дитини;
- нагадування про оплату;
- сторінка з квитанціями, які необхідно сплатити;
- можливість оплатити квитанції.

Вихователь групи:

- відображення груп, в яких працює вихователь;
- генерація розкладу за допомогою штучного інтелекту;
- ручне редагування розкладу;
- перегляд списку дітей групи;
- перегляд журналу відвідувань;
- сканування QR-коду вчителем для реєстрації відвідування дітей.

Отже, система мобільний застосунок «ChildWell» пропонує інтуїтивно зрозумілі та ефективні інструменти для адміністраторів закладів дошкільної освіти, батьків і вихователів. Адміністратори можуть легко керувати активностями садка, персоналом і дитячими групами, а також обробляти заявки на вступ. Батьки мають доступ до повної інформації про своїх дітей, заявки у садки, історію відвідувань, а також можливість здійснювати оплату автоматично сформованих виписок. Вихователі використовують штучний інтелект для генерації розкладу та відстежують відвідуваність за допомогою QR-кодів. Ця система не лише спрощує управління дошкільними закладами, але й підвищує залученість батьків та ефективність виховного процесу.

### 1.3 Постановка задачі

Після аналізу сфери дошкільної освіти, існуючих викликів та дослідження дійсних потреб ринку стає зрозумілою потреба у створенні мобільного застосунку, який зможе адресувати нагальні питання, що виникають у цій галузі, а саме – генерацію розкладу за допомогою штучного інтелекту і автоматизацію відвідувань за допомогою QR-кодів. Мобільний застосунок «ChildWell» надаватиме наступні можливості:

- перегляд закладів, доступ до яких має користувач;
- налаштування активностей;
- прийом або відхилення заявок на вступ;
- перегляд персоналу та дитячих груп;
- додавання та перегляд дітей;
- подача заявок у садки;
- перегляд закладів для подачі заявок;
- перегляд розкладу дітей;
- генерація QR-коду для реєстрації присутності дитини;
- ручне редагування розкладу;
- перегляд історії відвідувань;
- нагадування про оплату;
- сторінка з квитанціями для оплати;
- можливість оплати квитанцій;
- відображення груп, у яких працює вихователь;
- генерація розкладу за допомогою штучного інтелекту;
- перегляд списку дітей у групах;
- перегляд журналу відвідувань;
- сканування QR-коду вчителем для реєстрації відвідування дітей.

Для розробки мобільного застосунку використовується мова програмування TypeScript, яка забезпечує типізацію даних і підвищує надійність коду. Бібліотека React Native дозволяє створювати кросплатформенні мобільні додатки, що

працюють на iOS та Android, з використанням єдиного кодової бази, що значно скорочує час та витрати на розробку. Використання фреймворку Expo спрощує налаштування та запуск проекту, надаючи готові інструменти для роботи з нативними функціями пристрою. Середовище розробки Visual Studio Code забезпечує зручний інтерфейс та потужні можливості для налагодження та редагування коду. Емулятор Expo Go дозволяє швидко тестувати додаток на реальних пристроях без необхідності проходити складні процедури налаштування.

Застосунок взаємодіє із сервером за допомогою HTTP протоколу, що забезпечує надійну та безпечну передачу даних між клієнтом та сервером.

Керування процесом розробки відбувається через інструмент для управління проектами Azure DevOps, який надає можливості для планування, відстеження прогресу та співпраці між членами команди. Використання методології Agile дозволяє гнучко реагувати на зміну вимог та швидко впроваджувати нові функції. Регулярні спринти та ретроспективи допомагають підтримувати високу якість коду та своєчасне виконання задач, забезпечуючи ефективну комунікацію та продуктивність команди.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Збір та аналіз вимог

Визначення вимог відіграє ключову роль у процесі створення будь-якої системи, адже це вимагає детального розуміння поточних ринкових умов та індивідуальних потреб користувачів. Ефективність системи і її успіх безпосередньо залежать від аналітичного оцінювання інформації про користувачів, ринкові тенденції та діяльність конкурентів [8]. Ця інформація допомагає сформулювати основні вимоги, які відповідають головним цілям та забезпечують довгострокову працездатність системи.

Глибоке розуміння вимог і очікувань цільової аудиторії є вирішальним у формуванні функціональних характеристик системи, що відповідають потребам ринку і сприяють збільшенню попиту. В рамках цього процесу була розроблена таблиця для детального аналізу цільової аудиторії, що дозволяє ясно визначити ключові параметри для розробки (див. табл. 2.1).

Таблиця 2.1 – Аналіз цільової аудиторії

Цільова аудиторія	Больова точка	Мета	Як система допоможе
Адміністратори дошкільних закладів	<ul style="list-style-type: none"> <li>– відсутність ефективного інструменту для управління закладом;</li> <li>– труднощі з організацією персоналу;</li> <li>– складність у відстеженні відвідуваності та заявок</li> </ul>	<ul style="list-style-type: none"> <li>– максимізація ефективності управління закладом;</li> <li>– покращення управління персоналом;</li> <li>– автоматизація процесів прийому заявок</li> </ul>	<ul style="list-style-type: none"> <li>– автоматизація процесів управління, налаштування активностей;</li> <li>– керування персоналом і заявками через централізовану систему;</li> <li>– цифрове відстеження відвідуваності</li> </ul>

Кінець таблиці 2.1

Цільова аудиторія	Больова точка	Мета	Як система допоможе
Батьки	<ul style="list-style-type: none"> <li>– складність у відстеженні розвитку і відвідування дітей;</li> <li>– обмежений доступ до інформації про заняття;</li> <li>– труднощі з оплатою послуг</li> </ul>	<ul style="list-style-type: none"> <li>– покращення комунікації між закладом і батьками;</li> <li>– прозорий доступ до освітньої інформації;</li> <li>– спрощення процесу оплати</li> </ul>	<ul style="list-style-type: none"> <li>– перегляд розвитку та відвідуваності дитини через мобільний застосунок;</li> <li>– інформація про заняття та події у доступній формі;</li> <li>– інтеграція платіжної системи</li> </ul>
Вихователі	<ul style="list-style-type: none"> <li>– складність планування та адаптації занять;</li> <li>– відсутність ефективних засобів для відстеження відвідуваності</li> </ul>	<ul style="list-style-type: none"> <li>– оптимізація робочого часу;</li> <li>– покращення взаємодії з дітьми;</li> <li>– залучення батьків до освітнього процесу</li> </ul>	<ul style="list-style-type: none"> <li>– генерація розкладу за допомогою штучного інтелекту;</li> <li>– моніторинг відвідуваності через QR-коди</li> </ul>

Створення таблиці здійснювалось за допомогою колективного обговорення, де члени команди зустрілись для обміну думками про ключові характеристики цільової аудиторії. Такий підхід до збору ідей і аналізу важливих параметрів цільової аудиторії важливий для подальшої ефективності розробки проекту.

## 2.2 Формування вимог

### 2.2.1 Функціональні вимоги

Функціональні вимоги визначають необхідні функції та процедури системи, які мають бути реалізовані для того, щоб користувачі могли повноцінно використовувати її можливості. Документація цих вимог була виконана таким чином, щоб вона була доступною всім членам команди, використовуючи Azure DevOps, що є ключовим елементом в процесі створення ефективної системи.

Для формулювання функціональних вимог використовувались User Story, кожне з яких було доповнене критеріями прийняття для детальності та ясності. Ці User Story були інтегровані в спринти, що дозволяло ефективно планувати та управляти робочим процесом, забезпечувало послідовність розвитку проекту та спрощувало взаємодію між членами команди. Крім того, кожен спринт передбачав регулярні зустрічі для перегляду прогресу, що стимулювало швидке вирішення виникаючих проблем та адаптацію до змін у проектних вимогах, забезпечуючи гнучкість.

Процес інтеграції User Story в спринти не лише структурував роботу, але й дозволив команді дотримуватися чіткого графіку виконання задач, що позитивно вплинуло на якість кінцевого продукту. Кожна User Story, перед включенням до спринту, була ретельно обговорена та оцінена, що забезпечувало її відповідність поточним потребам проекту та користувачів.

Перелічимо User Story для кожної ітерації системи «ChildWell», які відображають основні функції та можливості мобільної системи. Наприклад, перший спринт пов'язаний з основами управління закладом, тобто мобільним функціоналом для адміністраторів закладів дошкільної освіти. Він включає в себе наступні User Story:

- «Управління закладами»;
- «Управління активностями закладу»;
- «Управління персоналом»;
- «Управління заявками в закладах для адміністратора»;
- «Управління дитячими групами».

Кожна User Story має свій детальний опис. Наприклад, детальний опис із Azure DevOps до User Story про «Управління заявками в закладах для адміністратора» містить опис процесу подачі заявок для кращого розуміння цієї сторі (див. рис. 2.1). Цей підхід дозволяє чітко розуміти всі етапи розробки та забезпечує прозорість у процесі створення вказаної функціональності, необхідної для успішної розробки мобільного застосунку «ChildWell».

**USER STORY 11**

11 Управління заявками в закладах для адміністратора

VN Vitalii Nesterenko 0 Comments Add Tag

State	● Resolved	Area	ChildWell
Reason	🔒 Code complete and unit t	Iteration	ChildWell\Iteration 1

**Description** ↗ ^

Як адміністратор закладу, я хочу мати можливість переглядати, відхиляти та підтверджувати заявки на вступ, щоб ефективно управляти процесом прийому дітей до закладу та їх розподілу по групах відповідно до їхніх потреб та вимог.

1. Подача заявки батьками: Батько обирає садочок і подає заявку на вступ дитини до закладу через систему.
2. Перегляд заявки адміністратором: Адміністратор переглядає подану заявку, вивчає внесені дані та вирішує, чи відповідає дитина вимогам закладу.
3. Рішення адміністратора: Адміністратор може відхилити заявку, якщо дитина не відповідає критеріям, або обрати відповідну групу та підтвердити заявку.
4. Фінальне підтвердження від батьків: Після підтвердження заявки адміністратором, батьки отримують сповіщення та можуть фінально підтвердити вступ дитини до обраної групи, або відхилити, якщо їм щось не підходить.

**Acceptance Criteria**

- Заявки від батьків приходять до системи без збоїв.
- Адміністратор може легко переглядати всі деталі заявок у своєму інтерфейсі.
- Система надає адміністратору інструменти для прийняття або відхилення заявок.
- Батьки отримують повідомлення про статус заявки (підтверджено/відхилено).
- Після підтвердження заявки адміністратором, батьки можуть завершити процес підтвердження або відхилення.
- Всі взаємодії мають бути зафіксовані в системі для звітності та аудиту.

Рисунок 2.1 – Опис User Story «Управління заявками в закладах для адміністратора» (рисунок виконано самостійно)

Мобільна система має задовольняти низку затверджених функціональних вимог, які були чітко визначені та структуровані. Ці вимоги викладено у вигляді User Story, дозволяючи наочно представити основні функції та можливості, які система повинна надати своїм користувачам.

US-1: Як користувач, я хочу авторизуватися в системі, щоб мати доступ до персоналізованого інтерфейсу і функцій.

US-2: Як адміністратор закладу, я хочу переглядати список закладів до яких маю доступ, щоб переглядати і редагувати їх інформацію.

US-3: Як адміністратор закладу, я хочу створювати активності для розкладу дітей, щоб збагачувати їх розпорядок дня.

US-4: Як адміністратор закладу, я хочу редагувати активності для дітей, щоб можна було генерувати більш адаптований розклад.

US-5: Як адміністратор закладу, я хочу переглядати профілі персоналу, щоб легше управляти ресурсами.

US-6: Як адміністратор закладу, я хочу приймати заявки від батьків, щоб оптимізувати процес запису.

US-7: Як адміністратор закладу, я хочу відхиляти заявки від батьків, щоб відбирати лише підходящих кандидатів.

US-8: Як адміністратор закладу, я хочу переглядати дитячі групи закладів, щоб ефективно організувати навчальний процес.

US-9: Як батько, я хочу додавати інформацію про мою дитину, щоб усі необхідні дані були доступні у системі.

US-10: Як батько, я хочу переглядати інформацію про садки, щоб знати, куди подати свою дитину.

US-11: Як батько, я хочу подавати заявки в садки, щоб забезпечити навчання моєї дитини.

US-12: Як батько, я хочу переглядати статус заявок, щоб бути в курсі процесу вступу.

US-13: Як батько, я хочу переглядати розклад занять своєї дитини, щоб знати, як розвивається моя дитина.

US-14: Як батько, я хочу отримувати нагадування про оплату, щоб не пропустити терміни платежів.

US-15: Як батько, я хочу подавати платежі через систему, щоб спростити процес оплати.

US-16: Як батько, я хочу переглядати історію платежів, щоб контролювати мої фінансові витрати.

US-17: Як батько, я хочу генерувати QR-код для входу/виходу, щоб легко реєструвати присутність моєї дитини.

US-18: Як батько, я хочу переглядати журнал відвідувань моєї дитини, щоб стежити за її відвідуваністю.

US-19: Як вихователь, я хочу переглядати призначені мені групи, щоб організувати свою роботу.

US-20: Як вихователь, я хочу генерувати розклад занять за допомогою штучного інтелекту, щоб забезпечити ефективне планування.

US-21: Як вихователь, я хочу вручну редагувати розклад занять, враховувати поточні потреби.

US-22: Як вихователь, я хочу регенерувати розклад занять за допомогою штучного інтелекту, щоб забезпечити більш точне і адаптоване планування.

US-23: Як вихователь, я хочу переглядати список дітей у групі, щоб краще підготуватися до занять.

US-24: Як вихователь, я хочу вести журнал відвідуваності, щоб відстежувати присутність.

US-25: Як вихователь, я хочу сканувати QR-коди для відмітки відвідувань, щоб автоматизувати процес реєстрації.

Цей формат представлення вимог допомагає команді розробників більш ефективно організувати процес розробки та забезпечити високий рівень відповідності кінцевого продукту очікуванням користувачів.

### 2.2.2 Нефункціональні вимоги

Для мобільного застосунку управління дошкільними закладами «ChildWell» нефункціональні вимоги відіграють ключову роль у забезпеченні якості, безпеки та зручності використання. Варто зазначити, що нефункціональних вимог може бути дуже багато, тому було зосереджено увагу тільки на тих вимогах, які відіграють ключову роль в функціонуванні нашого застосунку. Такими вимогами є:

- продуктивність та швидкість відгуку;
- надійність;
- масштабованість;
- безпека;
- сумісність;
- локалізація;

- інтернаціоналізація;
- підтримка та обслуговування.

Тепер, після того як ми визначили необхідні нефункціональні вимоги, можна описати кожну вимогу більш детально.

**Продуктивність та швидкість відгуку.** Програма повинна забезпечувати швидку відповідь на запити користувачів, з часом відгуку не більше 3 секунд. Обробка та завантаження великого обсягу даних (наприклад, список всіх дітей та персоналу) не повинні перевищувати 5 секунд.

**Надійність.** Мобільний застосунок повинен мати високу доступність, працюючи 24/7 без значних перерв. А також має мати регулярне резервне копіювання даних для запобігання втраті інформації

**Масштабованість.** Система повинна бути здатною підтримувати до 10,000 активних користувачів одночасно та зберігати до 1 мільйона записів без зниження продуктивності або швидкості відгуку системи.

**Безпека.** Захист даних має забезпечуватися за допомогою сучасних методів шифрування, наприклад, SHA-256.

**Сумісність.** Система має підтримувати останні версії iOS, від iOS 12 і вище, а також версії Android починаючи з Android 6.0 (Marshmallow) і вище, щоб забезпечити підтримку останніх функцій та безпеку.

**Локалізація.** Всі елементи інтерфейсу користувача, такі як кнопки, меню, форми вводу і помилки, повинні бути доступні англійською та українською мовами.

**Інтернаціоналізація.** Програма має використовувати Unicode для підтримки різних мовних алфавітів. Мобільний застосунок має підтримувати динамічну зміну мови на льоту, без необхідності перезавантаження застосунку.

## 3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 UML проєктування ПЗ

У процесі створення мобільного застосунку «ChildWell», команда використовувала різноманітні UML діаграми для глибшого розуміння і документування архітектури системи. Серед ключових діаграм були діаграми використання (Use Case Diagrams), які відображають основні функції, які система має виконувати для різних типів користувачів, діаграми активності (Activity Diagrams) для зображення процесів та дій, що відбуваються в системі, та діаграми послідовності (Sequence Diagrams), що показують взаємодію між об'єктами в часі для виконання конкретних завдань.

Особливу увагу приділено діаграмам використання, які допомагають виділити основні сценарії взаємодії користувачів із мобільним застосунком. Створено декілька варіантів таких діаграм, а саме для адміністраторів садочків, вихователів і батьків, кожна група із яких має свої унікальні функції системи і права доступу.

Інтеграція вказаних діаграм в процес розробки дозволяє систематично перевіряти відповідність функціоналу реальним потребам користувачів, а також оптимізувати процеси, усуваючи потенційні недоліки в інтерфейсі та взаємодіях ще до релізу продукту. Це важливо для створення зрозумілого та надійного продукту, який відповідатиме очікуванням користувачів та забезпечить їхнє задоволення.

На рисунку 3.1 зображено Use Case діаграму для адміністраторів закладів дошкільної освіти, яка демонструє основні взаємодії з системою. Адміністратор може виконувати авторизацію, переглядати або редагувати списки дітей та груп, а також обробляти заявки від батьків. Основні функції включають перегляд дітей та їхніх груп, управління активностями та вибір групи для нових заявок. Діаграма також показує можливість виходу з системи, що дає змогу адміністраторам зберігати безпеку доступу до своїх адміністративних функцій.

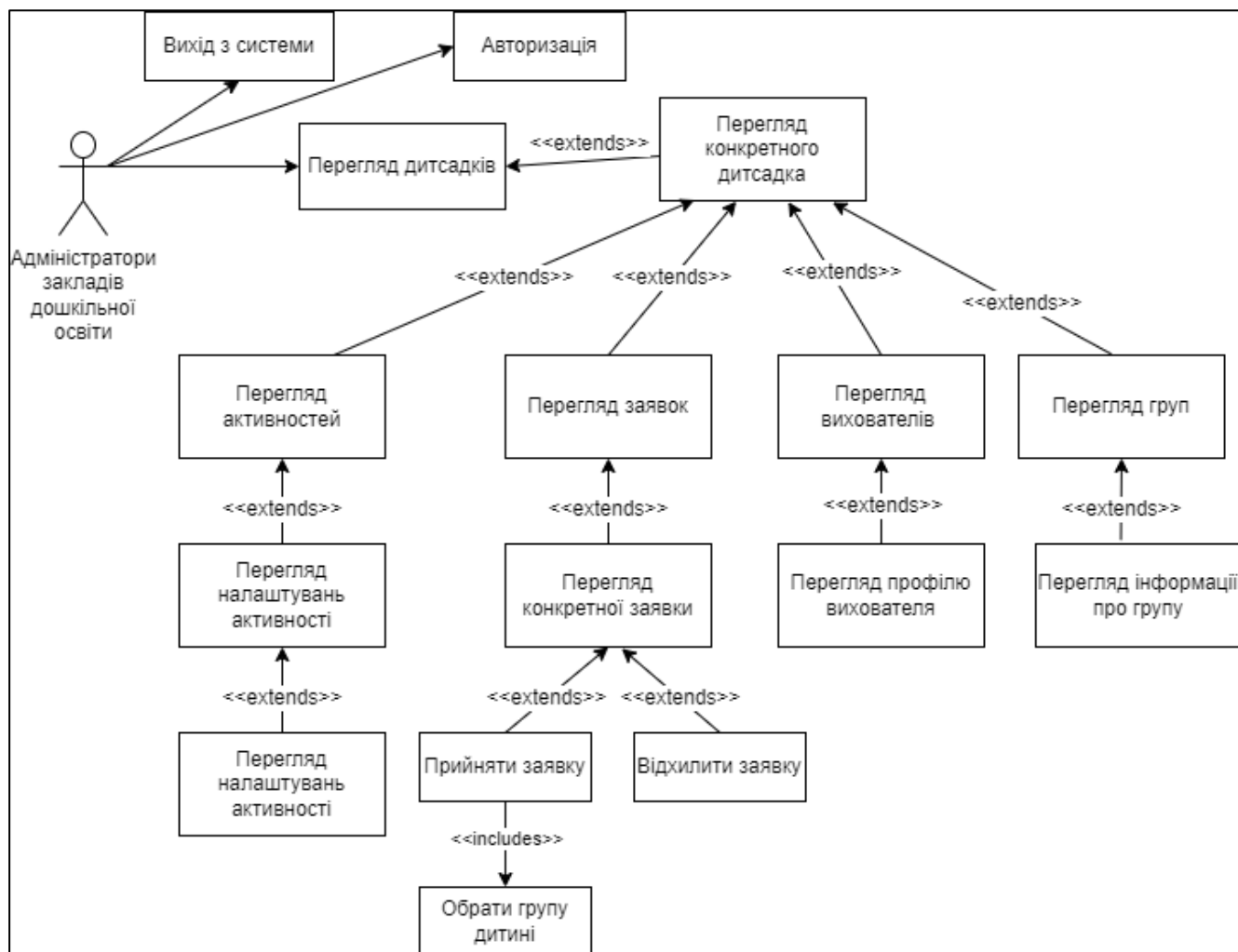


Рисунок 3.1 – Use Case діаграма адміністратора садочка (рисунок виконано самостійно)

На рисунку 3.2 зображено діаграму використання для користувачів, що є батьками. Від моменту авторизації, батьки мають доступ до широкого спектру функцій, що полегшують управління освітніми та адміністративними аспектами догляду за їхніми дітьми. Вони можуть переглядати і оплачувати рахунки, керувати поданням заявок у конкретні навчальні заклади, а також слідкувати за діяльністю та розкладами своїх дітей. Крім того, батьки мають змогу переглядати інформаційні панелі, які забезпечують інтуїтивне керування відвідуваннями та активностями дітей. Також батьки можуть створювати профілі для своїх дітей, керувати списком поданих заявок та переглядати історію відвідувань, що забезпечує комплексний підхід до управління процесом виховання та навчання.

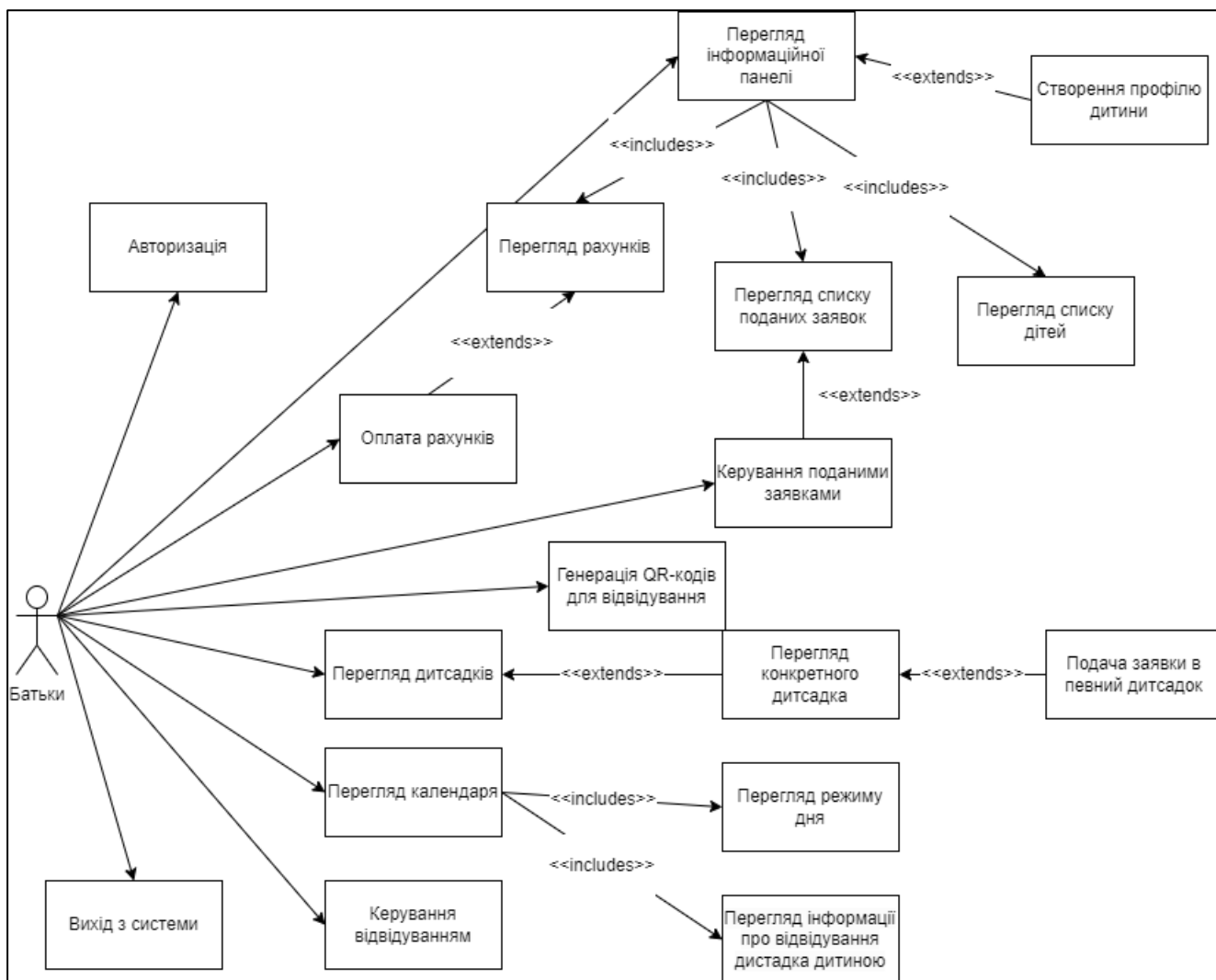


Рисунок 3.2 – Use Case діаграма батьків (рисунок виконано самостійно)

Однією з ключових функцій мобільного застосунку «ChildWell» є можливість генерації QR-кодів для відмічання відвідувань дітей, що дозволяє батькам швидко реєструвати прибуття та відхід своїх дітей з закладу. Це не тільки спрощує процес відмічення присутності, але й забезпечує додатковий рівень безпеки, дозволяючи закладу точно відстежувати хто і коли відвідує заклад.

Діаграма використання для батьків також вказує на можливість керування особистими профілями дітей, що включає додавання інформації про дитину. Важливою функцією є також перегляд календаря, який дозволяє батькам бути в курсі усіх майбутніх подій та організаційних моментів, пов'язаних із життям і навчанням їхньої дитини. Вихід з системи, як завершальний крок, забезпечує

додаткову безпеку, гарантуючи, що особиста інформація залишається доступною лише під час активної сесії батька-користувача.

На рисунку 3.3 Use Case діаграма описує варіанти взаємодії вихователів з мобільним застосунком «ChildWell». Діаграма відображає функції, що дозволяють вихователям керувати розкладом дня, відвідуваннями дітей та отримувати інформацію про групи і садочки.



Рисунок 3.3 – Use Case діаграма вихователів (рисунок виконано самостійно)

Кейс «Перегляд груп» дає змогу вихователям переглядати список груп, до яких вони мають доступ. Далі кейс «Перегляд конкретної групи» надає деталізовану інформацію про вибрану групу, включаючи список дітей, які в неї входять. «Перегляд розкладу дня» відображає розклад активностей для групи, яку обрав вихователь. Кейс «Генерація та регенерація розкладу дня за допомогою штучного інтелекту» дозволяє автоматизувати створення та коригування розкладу на основі вказівок вихователя.

«Сканування QR-коду для реєстрації відвідувань» уможливорює фіксацію часу приходу та відходу дітей, що забезпечує точний і безпечний облік присутності. Тим часом кейс «Перегляд історії відвідувань дітей» допомагає вихователям переглядати історію відвідувань дітей в групі протягом певного періоду.

Щоб глибше зрозуміти, як функціонує система, було створено декілька діаграм станів та переходів (State-transition diagram), по одній для кожної головної функції застосунку. Ця діаграма ілюструє зміни станів об'єктів протягом їх взаємодій з користувачами. Вони відтворюють весь процес взаємодії користувача з системою, наглядно показуючи кожен етап взаємодії.

На рисунку 3.4 зображено діаграму станів та переходів генерації календаря за допомогою штучного інтелекту. Процес розпочинається зі стану «Розкладу не існує», де ще не створено жодного розкладу. З цього стану можна перейти до стану «Розклад згенеровано», що відбувається після того, як система генерує перший варіант розкладу на основі введених даних. Після генерації, маючи стан «Розклад згенеровано», користувач може переглянути розклад. У цей момент розклад може бути або прийнятий (перехід до стану «Розклад сформовано»), або відхилений (перехід до стану «Розклад відхилено»). Після відхилення розкладу користувач може знову згенерувати розклад, що призводить до стану «Розклад згенеровано». Після прийняття розкладу користувач може завершити генерацію розкладу, що призводить до стану «Розклад сформовано». Після завершення генерації розкладу користувач може зупинити генерацію розкладу, що призводить до стану «Розклад відхилено».

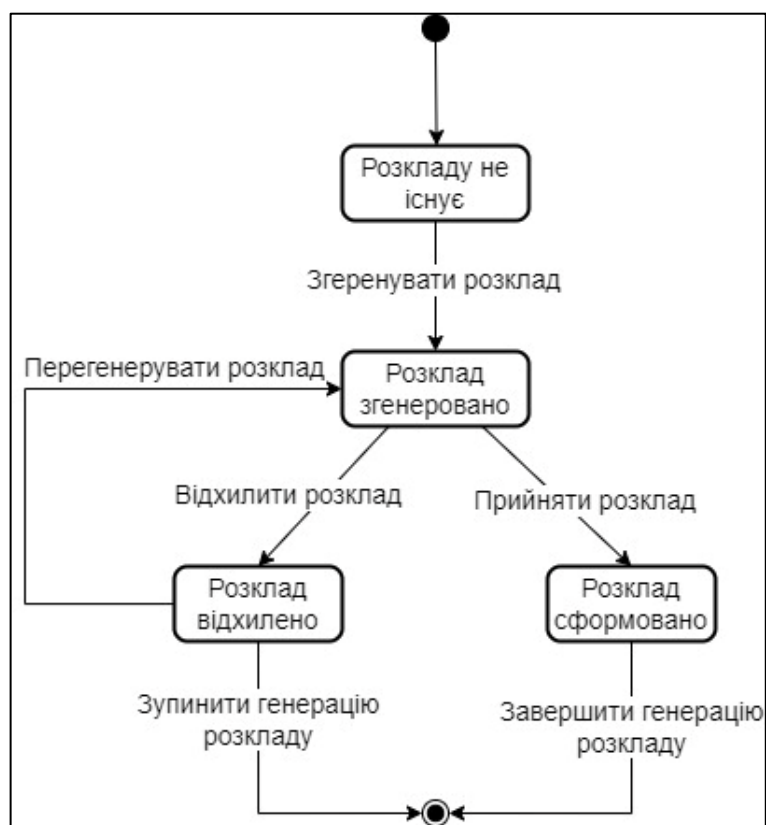


Рисунок 3.4 – Діаграма станів та переходів генерації розкладу (рисунок виконано самостійно)

Якщо розклад не підійшов під вимоги вихователя, він має можливість внести зауваження і запустити процес регенерації розкладу. Це знову приводить до стану «Розклад згенеровано», де розклад обробляється з урахуванням нових умов або вимог. А якщо ж після регенерації розклад влаштовує користувача, він переходить до стану «Розклад сформовано», що означає завершення процесу генерації і стабілізацію розкладу для подальшого використання.

В кінці кінців, незалежно від того, чи був розклад прийнятий або відхилений, користувач може завершити процес генерації, що веде до кінцевого стану і виходу з меню генерації розкладу.

Наступна головна функція, яку було описано за допомогою діаграми станів та переходів є процес відмічання присутності дитини у дитячому садку за допомогою QR-кодів (див. рис. 3.5).



Рисунок 3.5 – Діаграма станів та переходів присутності дитини (рисунок виконано самостійно)

«Дитина відсутня» - це початковий стан, коли дитина не присутня в закладі. Цей стан застосовується, коли дитина ще не прибула до садка. Після сканування вихователем наданого батьками QR-коду стан присутності дитини змінюється на «Дитину приведено батьками». Це вказує на те, що дитина була успішно приведена до закладу та відмічена як присутня.

В кінці дня батьки приходять забирати дитину, а отже їм знову необхідно надати QR-код для його сканування. Таким чином стан переходить в остаточний «Дитину забрано батьками», який вказує, що дитина була забрана з закладу. Це підтверджує завершення візиту дитини до садка на той день.

Отже, ця діаграма відображає як систематичний так і безпечний підхід до відмічання присутності та виходу дітей, що дозволяє точно контролювати їхнє прибуття та відхід з закладу дошкільної освіти.

На рисунку 3.6 відображено діаграму станів і переходів процесу подачі та обробки заявок батьками на вступ їхніх дітей до садочка в мобільному застосунку «ChildWell».

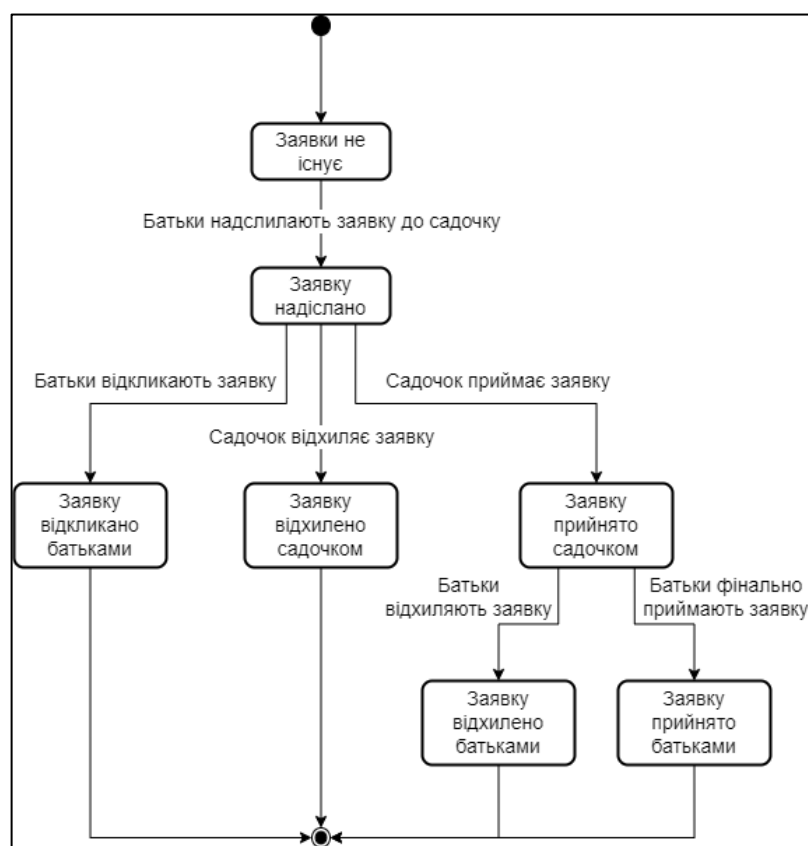


Рисунок 3.6 – Діаграма станів та переходів заявки до садочку (рисунок виконано самостійно)

Процес подачі заявки в заклад дошкільної освіти розпочинається з відсутності заявки, після чого батьки можуть подати заявку до садочка. Коли заявка подана, вона перебуває в стані «Заявку надіслано», де чекає на розгляд адміністрацією садочка.

Після розгляду, адміністратор дитсадочку може відхилити заявку або прийняти її. Якщо садочок відхиляє заявку, вона переходить у стан «Заявку відхилено садочком», і батьки мають можливість її відхилити остаточно або відновити процес, знову надсилаючи заявку. У випадку, коли заявка прийнята садочком, вона переходить у стан «Заявку прийнято садочком», звідки батьки можуть остаточно прийняти заявку, переводячи її у стан «Заявку прийнято батьками», або ж відхилити, переводячи у стан «Заявку відхилено батьками».

Показаний процес дає можливість батькам контролювати стан заявки в реальному часі та забезпечує відповідну гнучкість у прийнятті рішень щодо вибору освітнього закладу для їхньої дитини. Щойно наведена діаграма наглядно показує, що батьки та садочок взаємодіють протягом усього процесу, забезпечуючи чітке розуміння кожного кроку від подачі заявки до її остаточного прийняття або відхилення.

Діаграма розгортання, яка представлена на рисунку 3.7, відображає архітектуру системи мобільного застосунку «ChildWell», яка включає мобільні додатки під Android і iOS, серверну частину і систему ідентифікації.

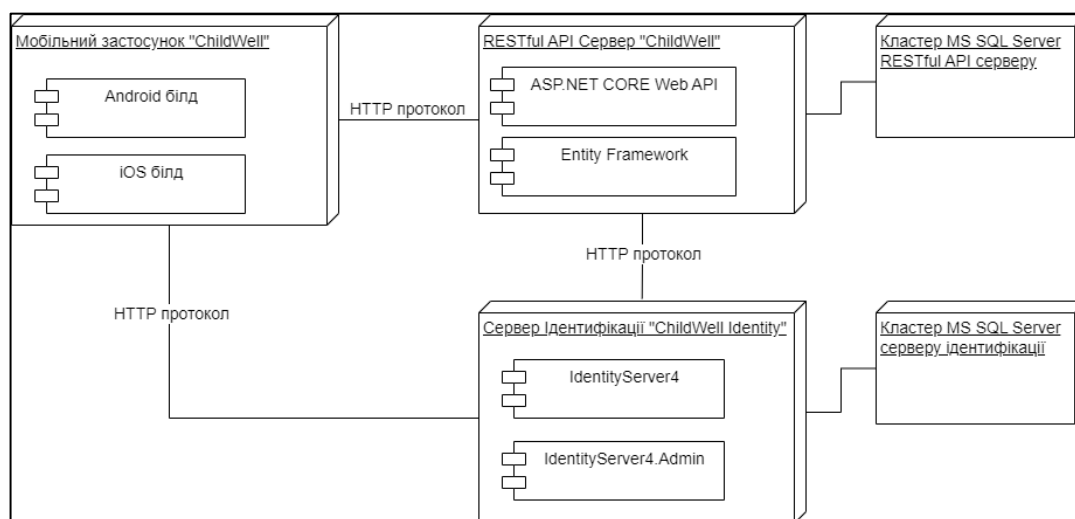


Рисунок 3.7 – Діаграма розгортання мобільного застосунку «ChildWell» (рисунок виконано самостійно)

Мобільний клієнт «ChildWell» складається з версій для Android та iOS, з яких кожна взаємодіє з серверними компонентами через HTTP протокол.

Зображений на малюнку RESTful API сервер «ChildWell» використовує фреймворк ASP.NET Core Web API, що обробляє запити від мобільних клієнтів, усі, окрім запитів авторизації, адже їх обробляє інший сервер. Сервер використовує Entity Framework для взаємодії з реляційною базою даних. Сервер взаємодіє з кластером MS SQL Server, який зберігає дані застосунку, окрім даних авторизації.

Для обробки запитів авторизації і забезпечення безпекою мобільного застосунку використовується сервер ідентифікації «ChildWell Identity». Він використовує IdentityServer4, стандартне рішення для імплементації OpenID Connect та OAuth 2.0 протоколів. IdentityServer4.Admin дозволяє адмініструвати політики доступу та управління ідентифікаційними даними користувачів. Сервер ідентифікації також з'єднаний з власним кластером MS SQL Server для зберігання даних авторизації та ідентифікації (не містить даних застосунку, тільки зашифровані дані для авторизації).

Діаграма розгортання надає зрозуміле уявлення про те, як компоненти системи взаємопов'язані між собою для забезпечення правильної роботи мобільного застосунку «ChildWell». Діаграма підкреслює модульність та розподіленість системи, що є ключовими для сучасних мобільних додатків, гарантуючи користувачам швидкість і безпеку даних.

### 3.2 Проектування архітектури ПЗ

Архітектура мобільного застосунку «ChildWell» базується на клієнт-серверному підході, де мобільний додаток виступає як клієнт, а серверна частина обробляє авторизацію, бізнес-логіку і управління даними. Цей підхід дозволяє розділити відповідальності між клієнтськими пристроями та сервером.

Взагалі, клієнт-серверна архітектура є найбільш поширеним видом архітектури, адже вона має цілий ряд переваг. По-перше, клієнт-серверна архітектура дозволяє централізовано управляти даними та ресурсами. Сервер є

відповідальним за обробку запитів, управлінням базою даних, ресурсами та бізнес-логікою, що значно спрощує взаємодію між різними клієнтами.

По-друге, ця архітектура дозволяє легко масштабувати застосунок, збільшуючи серверні ресурси, щоб мати змогу обслуговувати зростаючу кількість клієнтів без необхідності модифікації клієнтської частини.

І, по-третє, клієнти, які звертаються до серверу, вимагають менше обчислювальних ресурсів, адже обробка даних відбувається на сервері. Це знижує вимоги до апаратного забезпечення клієнтських пристроїв і пришвидшує відгук системи, що особливо актуально на застарілих моделях смартфонів.

Проте, незважаючи на свої численні переваги, клієнт-серверна архітектура має головний недолік – уся система залежить від стабільності та доступності сервера. Якщо сервер вийде з ладу, це може призвести до повної непрацездатності всіх клієнтів, що впливає на всіх користувачів.

На рисунку 3.8 зображено діаграму послідовності, яка висвітлює клієнт-серверну архітектуру і відображає як користувач взаємодіє з мобільним застосунком і рештою системи.

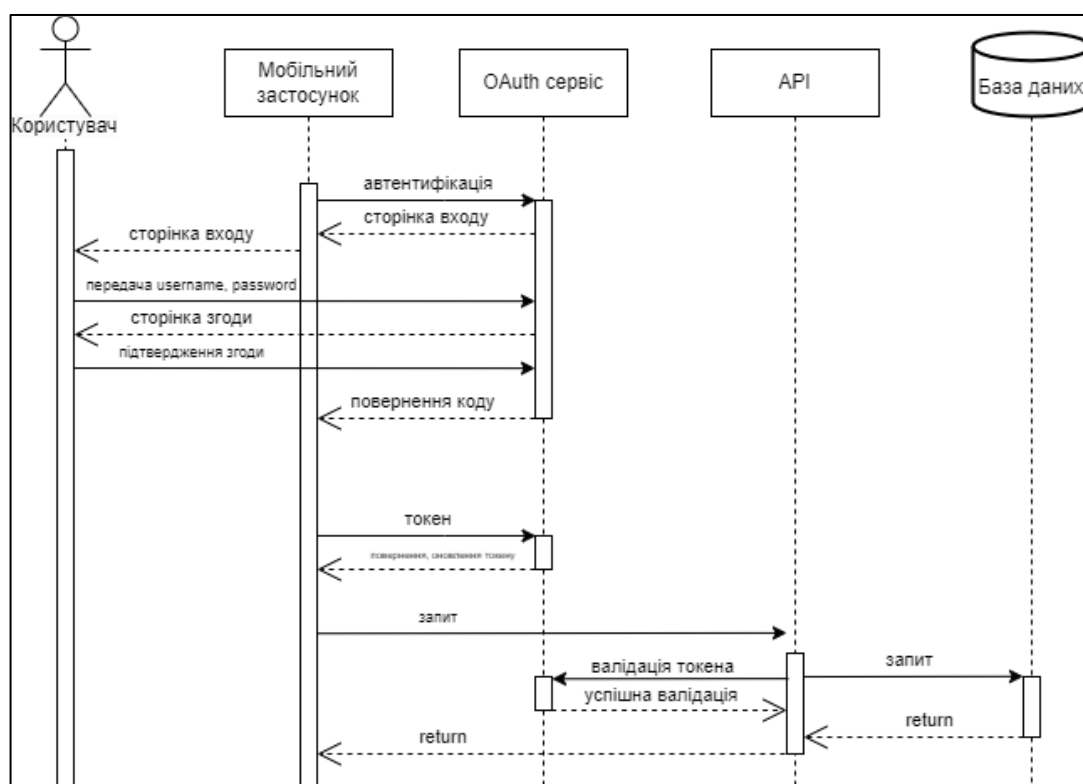


Рисунок 3.8 – Діаграма послідовності мобільного застосунку «ChildWell»  
(рисунок виконано самостійно)

Коли користувач відкриває сторінку входу та вводить свої логін та пароль, ці дані надсилаються до OAuth сервісу для аутентифікації. Далі сервіс аутентифікації обробляє ці дані і, якщо логін і пароль введено правильно, надсилає назад авторизаційний код, який мобільний застосунок використовує для отримання токена доступу. Токен потім використовується для запитів до API-серверу, де його валідність перевіряється перед наданням доступу до даних у базі даних.

Коли мобільний застосунок робить запит до API-сервера, цей сервер перевіряє токен, і у разі успішної валідації дозволяє доступ до запитаної інформації, надаючи відповідь клієнту. Усі дані зберігаються та обробляються в базі даних, забезпечуючи централізоване управління і захист інформації.

Наведений на діаграмі послідовності процес підвищує безпеку мобільного застосунку завдяки використанню стандартизованих протоколів аутентифікації, таких як OAuth2 і OpenID Connect [9].

Основою мобільного додатка «ChildWell» стала мова програмування TypeScript, яка є надбудовою для JavaScript із додаванням типізації, що дозволяє зменшити кількість помилок на етапі компіляції. Розробка клієнтської частини застосунку велася на основі популярної бібліотеки React Native яка дозволяє створювати нативні мобільні додатки за допомогою TypeScript та React.js. Це забезпечує можливість одночасної розробки під Android та iOS, значно зменшуючи витрати часу.

Фреймворк Expo, інтегрований із React Native, використовувався для спрощення процесу розробки та тестування. Expo надає широкий набір готових до використання API і компонентів, які спрощують інтеграцію з сервісами мобільних пристроїв, наприклад, доступ до камери і галереї.

Для емуляції та тестування мобільного додатку застосовувався Expo Go, який дозволяє запускати застосунки на реальних пристроях без необхідності компіляції нативних бінарних файлів, що пришвидшує розробку.

Розробка здійснювалась в середовищі Visual Studio Code, яке надає можливість встановлювати плагіни та інструменти для пришвидшення якості і

швидкості розробки. Git було використано як систему контролю версій через свою простоту і поширеність.

### 3.3 Проектування структури зберігання даних

Мобільний застосунок «ChildWell» покладається на сервер для зберігання і управління даними. Сервер взаємодіє із SQL базою даних, а якщо бути точнішим то з MS SQL Server.

Перш за все, SQL база даних була обрана через потужні можливості управління транзакціями, що забезпечує консистентність даних навіть при високих навантаженнях, що є особливо необхідним для нашого застосунку, адже штучний інтелект генерує багато даних для розкладу, і не можна якусь частину даних загубити або пошкодити.

SQL бази даних також підтримують складні запити з операціями JOIN, що є необхідним для ефективної роботи з великими та зв'язаними даними, які є типовими для системи управління дошкільними закладами. Наприклад, разом із дитиною дуже часто потрібно мати інформацію про її групу і відвідування. Щоб не робити декілька окремих запитів, використовується JOIN операція, що значно пришвидшує роботу застосунку.

Проте, використання SQL бази даних має великий недолік. Цей вид баз даних вимагає попереднє планування схеми бази даних, що ускладнює розробку і майже унеможливорює швидкі зміни в структурі даних.

В додатку А зображено ER-діаграму застосунку «ChildWell». На цій ER-діаграмі відображені сутності та зв'язки між ними для мобільного додатку «ChildWell».

Розпочнемо з головної сутності «Preschool» (заклад дошкільної освіти), вона необхідна для зберігання інформації про дитячі садки, включаючи назву, контактні дані, адресу, розклад роботи та інші характеристики. Ця сутність має зв'язки «один до багатьох» з сутностями «Group» (група), «PreschoolAdmin» (адміністратор садочки), «PreschoolPhoto» (фото садочки) і «Child» (дитина), що дозволяє кожному закладу мати кілька груп, адміністраторів, фотографій та зареєстрованих дітей.

Сутність «Group» (група) відображає різні групи в межах дитячих садків, де кожна група має свої вікові обмеження та опис. Кожна група пов'язана з сутностями «Activity» (активність для розкладу) та «PreschoolTeacher» (вихователь), що дозволяє генерувати розклад із наданих активностей та мати групі вчителів відповідно.

Сутність «Child» (дитина) включає всю необхідну інформацію про дітей, зареєстрованих у системі, включно з даними про батьків через зв'язок з сутністю «User» (користувач). Вона також зв'язана з «Preschool» (садочок), що дозволяє ідентифікувати, у якому саме закладі навчається дитина.

Сутність «User» (користувач) зберігає дані користувачів системи, які можуть бути адміністраторами, вчителями або батьками. Ця сутність пов'язана з «UserRole» (роль користувача, наприклад «Батько» або «Вихователь»), що визначає рівень доступу кожного користувача. Кожен користувач може мати різні ролі в системі, залежно від своїх обов'язків та прав доступу, але не більше однієї ролі на одного користувача.

Всі ці сутності взаємопов'язані таким чином, щоб забезпечити ефективне управління даними та процесами в межах застосунку, дозволяючи різним користувачам взаємодіяти з системою згідно з їхніми ролями та потребами.

### 3.4 Створення UI/UX

User Experience (UX) і User Interface (UI) — це дві важливі складові, які грають ключову роль у розробці будь-якої клієнтської частини, з якою взаємодіє людина, особливо важливо це в мобільних застосунках. UX стосується загального досвіду користувача з використанням застосунку, включаючи те, наскільки зручно і інтуїтивно зрозуміло користувачам взаємодіяти з ним. Це охоплює всі частини застосунку: від навігації і швидкості роботи до зручності виконання потрібних дій та отримання необхідної інформації.

UI, з іншого боку, зосереджений на візуальних аспектах застосунку — це дизайн інтерфейсу, включаючи розташування кнопок, кольорову гаму, шрифти і інші візуальні елементи, які впливають на те, як користувач сприймає застосунок.

Гарний UI робить застосунок не тільки привабливим, але й забезпечує чіткість і простоту користування. Більше того, дуже гарний і швидкий сервер не відчується користувачем так сильно, як гарний дизайн.

При розробці мобільних застосунків особлива увага до UX і UI є критично важливою, навіть більшою ніж при розробці веб-застосунків, адже смартфони мають невеликий екран, що ускладнює взаємодію з будь-яким застосунком. Друга проблема – це те, що екран мобільних телефонів сенсорний, і взаємодіяти з ним потрібно безпосередньо пальцями, а не мишею, як на веб-застосунках, що в свою чергу потребує великої уваги при розробці дизайну, адже замалі деталі ускладнять або взагалі унеможливають взаємодію із застосунком.

Перед розробкою дизайну мобільного застосунку «ChildWell» було обрано кольорову палітру (див. рис. 3.9), яка символізує спокій, адже це особливо важливо під час піклування дітьми.

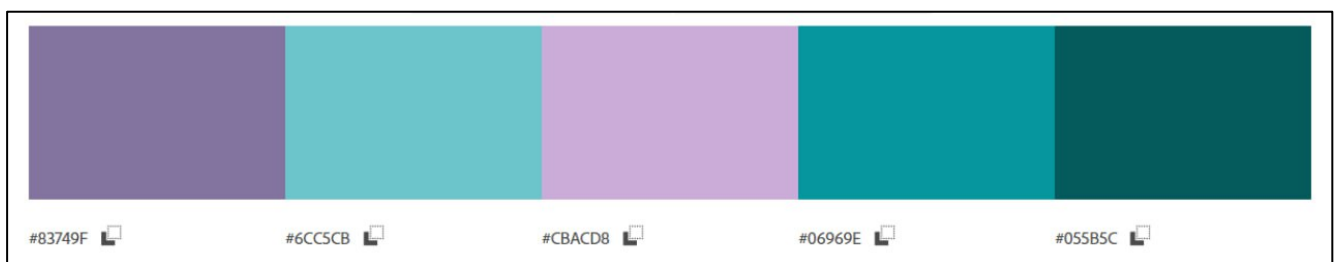


Рисунок 3.9 – Кольорова палітра інтерфейсу застосунку (рисунок виконано самостійно)

Обрані кольори включають відтінки від пастельних до насичених тонів: м'який фіолетовий (#83749F), ніжно бірюзовий (#6CC5CB), блідо-рожевий (#CBACDB), глибокий морський зелений (#06969E) та темно-бірюзовий (#055B5C). Ця палітра не лише сприяє естетичному візуальному сприйняттю, але й покращує навігацію та орієнтацію користувачів у програмі, виокремлюючи важливі елементи управління та розділи.

Інтерфейс мобільного застосунку «ChildWell» використовує шрифт «San Francisco», який є стандартним системним шрифтом для операційних систем iOS та macOS від Apple. Використання цього шрифту сприяє покращенню читабельності та забезпечує сучасний та елегантний вигляд додатку. Шрифт «San

Francisco» відомий своєю чіткістю та оптимізацією для різних розмірів екранів, забезпечуючи комфортні умови для користувачів під час щоденного використання застосунку.

На рисунку 3.10 зображено скріншот із дизайн-системи «Figma», де представлено декілька екранів мобільного застосунку «ChildWell».

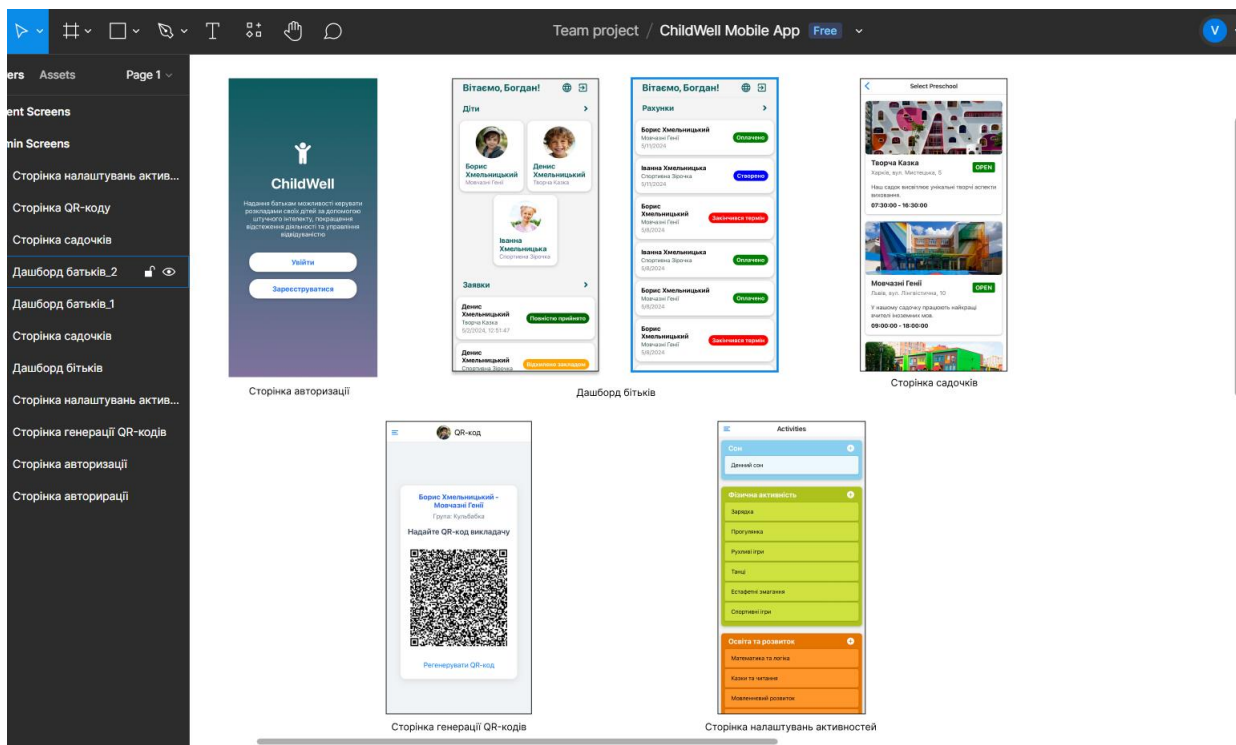


Рисунок 3.10 – Скріншот із дизайн-системи Figma (рисунок виконано самостійно)

На знімках екрана видно деякі екрани застосунку: екран авторизації, домашня сторінка, перегляд детальної інформації про садочки, управління активностями та QR-коди для відмічання відвідування дітей. Ці екрани відображають як візуальні, так і функціональні аспекти застосунку, надаючи користувачам зручний та інтуїтивно зрозумілий досвід користування. Це демонструє важливість добре продуманого дизайну інтерфейсу та його роль у загальній архітектурі мобільного застосунку.

## 4 ОПИС ПРИЙНЯТИХ РІШЕНЬ

### 4.1 Розробка клієнтської частини

Мобільний застосунок «ChildWell» розроблено використовуючи технології React Native та Expo, що дозволило значно спростити процес розробки і тестування застосунку одразу на обидві платформи: iOS та Android. Використання React Native дозволяє використовувати єдину кодову базу для платформ Android і iOS платформ. Основою архітектури застосунку став підхід до створення односторінкового застосунку SPA (Single Page Application), що забезпечує більш швидку і плавну взаємодію користувачів із системою без видимих для користувача пауз під час навігації від сторінки до сторінки.

Для реалізації маршрутизації в застосунку «ChildWell» використано Expo Router, який дозволяє ефективно управляти навігацією між різними екранами застосунку. Використання Expo Router сприяє оптимізації процесу розробки та забезпечує зручний і зрозумілий підхід до побудови навігаційних шляхів, що покращує загальний користувацький досвід.

Структура застосунку «ChildWell» організована у відповідності до найкращих практик розробки на React Native з використанням Expo (див .рис. 4.1).

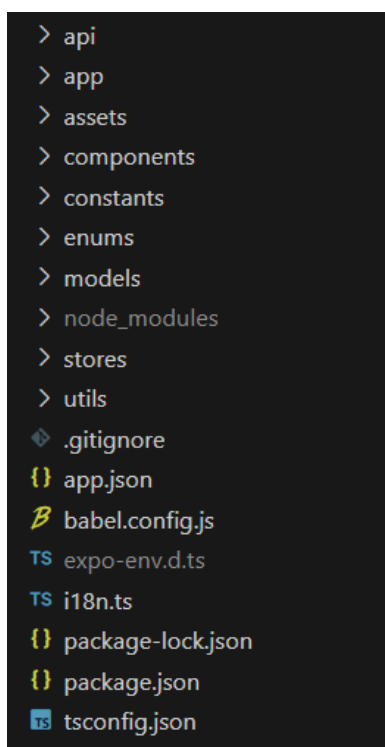


Рисунок 4.1 – Структура папок застосунку (рисунок виконано самостійно)

В папці «ari» містяться файли, які відповідають за взаємодію з серверною частиною застосунку. Вони включають функції для здійснення HTTP-запитів до API серверу.

Папка «app» є основною директорією застосунку, що містить початкові налаштування та головні компоненти додатку.

У папці «assets» зберігаються всі медіафайли, такі як зображення, іконки та інші статичні ресурси, які використовуються в застосунку.

Папка «components» призначена для зберігання багаторазово використовуваних React компонентів, таких як кнопки, поля введення, модальні вікна та інші елементи інтерфейсу.

В папці «constants» містяться файли з константами, які використовуються у додатку. Це константи для налаштування теми, які використовуються в різних частинах застосунку.

Папка «enums» зберігає перерахування (enum), які використовуються у додатку для визначення фіксованих наборів значень.

У папці «models» зберігаються моделі даних, які представляють структуру даних, що використовуються в додатку. Це інтерфейси або класи, які визначають дані для API запитів або внутрішніх операцій застосунку.

Папка «node\_modules» є стандартною директорією для всіх залежностей проекту, які встановлені через npm.

В папці «stores» зберігаються файли, які відповідають за управління станом додатку за допомогою бібліотеки Zustand, наприклад інформація про авторизованого користувача.

Папка «utils» містить допоміжні функції та утиліти, які використовуються в різних частинах застосунку, такі як функції для обробки дат, форматування тексту або інші загальні утиліти.

Файл «.gitignore» є конфігураційним файлом, який визначає, які файли та директорії мають бути проігноровані системою контролю версій Git.

Файл «app.json» містить конфігурацію для Expo, включаючи основні налаштування додатку, такі як ім'я, ідентифікатор пакету, іконки та інші метадані.

Файл «babel.config.js» є конфігураційним файлом для Babel, який використовується для трансформації коду JavaScript у формат, сумісний з різними середовищами виконання.

Файл «expo-env.d.ts» містить декларації типів для TypeScript, визначаючи специфічні для Expo глобальні змінні та інші типи.

Файл «i18n.ts» є конфігураційним файлом для інтернаціоналізації (i18n), який визначає налаштування для підтримки багатомовності у додатку.

Файл «package-lock.json» є автоматично згенерованим файлом, що зберігає точну інформацію про версії всіх встановлених залежностей проекту, забезпечуючи детерміновану установку залежностей.

Файл «package.json» є конфігураційним файлом для npm, який містить інформацію про проект, його залежності, скрипти для виконання та інші метадані.

Файл «tsconfig.json» є конфігураційним файлом для TypeScript, який визначає налаштування компілятора TypeScript для проекту.

Мобільний застосунок взаємодіє із сервером за допомогою HTTP запитів. Фрагмент коду, що зображено на рисунку 4.2, показує визначення об'єкта «Preschools», який містить методи для взаємодії з даними дитячих садків через API запити.

```
const Preschools = {
  list: () => requests.get<Preschool[]>("/preschools"),
  activities: (preschoolId: string) =>
    requests.get<Activity[]>(`/preschools/${preschoolId}/activities`),
  enroll: (preschoolId: string, childrenIds: string[]) =>
    requests.post<Preschool>(`/preschools/${preschoolId}/enroll`, childrenIds),
  editActivitySetting: (preschoolId: string, setting: EditActivitySetting) =>
    requests.patch<void>(`/preschools/${preschoolId}/activities/settings`, [
      setting,
    ]),
  teachers: (preschoolId: string) => requests.get<Teacher[]>(`/preschools/${preschoolId}/teachers`),
  apply: (preschoolId: string, application: ApplyToPreschool) =>
    requests.post<void>(`/preschools/${preschoolId}/apply`, [application]),
  applications: (preschoolId: string) =>
    requests.get<Application[]>(`/preschools/${preschoolId}/applications`),
};
```

Рисунок 4.2 – Фрагмент коду, що визначає методи для взаємодії із сервером  
(рисунок виконано самостійно)

Метод «list» виконує GET запит для отримання списку всіх дитячих садків. Метод «activities» отримує всі активності певного садка на основі його ідентифікатора. Метод «enroll» виконує POST запит для зарахування дітей у дитячий садок, приймаючи ідентифікатор садка та масив ідентифікаторів дітей. Метод «editActivitySetting» виконує PATCH запит для редагування налаштувань активностей садочку. Метод «teachers» повертає список вчителів певного садка, а метод «apply» виконує POST запит для подання заявки до дитячого садка. Метод «applications» отримує список заявок, поданих до певного садка.

Цей об'єкт «Preschools» є частиною загального об'єкта «agent», який містить всі методи для взаємодії з різними аспектами системи «ChildWell», включаючи розклад, групи дитячих садків, публічні дитячі садки, дітей, заявки, групи вчителів та рахунки.

Інтерфейс користувача цього додатка для дитячого садка розроблений для забезпечення зручності та простоти використання для всіх учасників. Меню батьків забезпечує легкий доступ до важливих функцій, таких як перегляд розкладу, сканування QR-кодів для відмічання присутності, історія відвідувань та зміна мови. Наприклад, меню батьків містить зрозумілу інформацію про обрану дитину та ключові опції для взаємодії.

Так як застосунок має авторизацію, важливою задачею стало забезпечення безпеки даних в застосунку. Тобто, якщо користувач не авторизований, він не зможе побачити якісь сторінки. Фрагмент коду, який забезпечує цей функціонал зображено на рисунку 4.3. Відображено налаштування маршрутизації в застосунку, використовуючи компоненти з бібліотеки «Expo Router». Компонент «RootSiblingParent» використовується для надання контексту ToastManager, який обробляє сповіщення у застосунку.

Всі екрани маршрутизації вкладаються в компонент «Stack», який створює стек навігації. Два основні екрани, «sign-in» та «protected», визначені всередині «Stack». Екран «sign-in» не показує заголовок та редиректить користувача на основі значення змінної «isAuthorized».

```

<PaperProvider theme={PaperCustomDefaultTheme}>
  <RootSiblingParent>
    <ThemeProvider value={CustomDefaultTheme}>
      <ToastManager />
      <Stack>
        <Stack.Screen
          name="sign-in"
          options={{ headerShown: false }}
          redirect={isAuthorized}
        />
        <Stack.Screen
          name="(protected)"
          options={{ headerShown: false }}
          redirect={!isAuthorized}
        />
      </Stack>
    </ThemeProvider>
  </RootSiblingParent>
</PaperProvider>

```

Рисунок 4.3 – Фрагмент коду, що забезпечує захист даних від неавторизованих користувачів (рисунок виконано самостійно)

Якщо користувач не авторизований, він залишається на екрані входу. Екран «protected» використовує протилежну логіку редиректу, відправляючи користувача на екран входу, якщо він не авторизований. Ця структура забезпечує захист доступу до захищених екранів, забезпечуючи безпеку та конфіденційність даних користувачів на основі стану їх авторизації.

Також необхідно розділити сторінки за ролями. Тобто, батько може генерувати QR-коди, але не може їх сканувати, а от вихователь навпаки – може сканувати QR-код, але не генерувати. Фрагмент коду, що відповідає за розділ сторінок за ролями, зображено на рисунку 4.4.

Для кожного екрану задані специфічні ролі користувачів, які мають доступ до цих екранів. Наприклад, екран «parent» призначений для батьків. Якщо користувач не належить до ролі «Parent» (батько) або не вибрано дитину, то відбудеться редирект на інший екран.

```

<Stack>

  <Stack.Screen
    name="(parent)"
    options={{ headerShown: false }}
    redirect={!inRole(user, UserRole.Parent) || !parentSelectedChild}
  />

  <Stack.Screen
    name="(teacher)"
    options={{ headerShown: false }}
    redirect={!inRole(user, UserRole.Teacher)}
  />

  <Stack.Screen
    name="(preschoolAdmin)"
    options={{ headerShown: false }}
    redirect={!inRole(user, UserRole.PreschoolAdmin) || !adminSelectedPreschool}
  />

</Stack>

```

Рисунок 4.4 – Фрагмент коду, що відповідає за розділ сторінок за ролями  
(рисунок виконано самостійно)

Аналогічно, екран «teacher» доступний лише користувачам з роллю «Teacher» (вихователь). Якщо користувач не має цієї ролі, він буде перенаправлений.

Для екрану «preschoolAdmin» визначені умови, що користувач має бути в ролі «PreschoolAdmin» (адміністратор садочку) і повинна бути вибрана відповідна дошкільна установа. У випадку невідповідності цим умовам, користувач буде перенаправлений.

Цей підхід забезпечує точний контроль доступу до різних частин застосунку, відповідно до ролі користувача, тим самим підвищуючи безпеку та персоналізацію інтерфейсу.

Перейдемо до частини застосунку, яка забезпечує зберігання даних про авторизованого користувача, тим самим дозволяючи закривати і відкривати застосунок без необхідності повторної авторизації (див. рис. 4.5). Наведений код створює сховище для керування станом автентифікації в мобільному застосунку за допомогою Zustand. Використовується функція «create<State>()» для створення

сховища під назвою «useAuthStore». Вона включає механізм збереження стану «persist», що дозволяє зберігати об'єкт «tokenResponse» та дві функції для його оновлення або очищення.

Об'єкт «Storage» відповідає за збереження даних у сховищі «SecureStore», забезпечуючи зчитування, запис та видалення даних. Таким чином, цей код забезпечує безпечне зберігання токенів автентифікації, які потрібні для доступу до захищених ресурсів у застосунку. Це також гарантує, що користувачі залишаються авторизованими навіть після перезапуску застосунку, що покращує загальний користувацький досвід. Крім того, використання «SecureStore» забезпечує високий рівень безпеки для збереження конфіденційної інформації.

```
export const useAuthStore = create<State>()(
  persist(
    (set, get) => ({
      tokenResponse: undefined,
      setTokenResponse: (tokenResponse: AuthSession.TokenResponse) =>
        set({ tokenResponse }),
      clearTokenResponse: () => set({ tokenResponse: undefined }),
    }),
    {
      name: "auth-store",
      storage: createJSONStorage(() => Storage),
    }
  )
);

const Storage: StateStorage = {
  getItem: async (name: string): Promise<string | null> => {
    return await SecureStore.getItemAsync(name);
  },
  setItem: async (name: string, value: string): Promise<void> => {
    return await SecureStore.setItemAsync(name, value);
  },
  removeItem: async (name: string): Promise<void> => {
    return await SecureStore.deleteItemAsync(name);
  },
};
```

Рисунок 4.5 – Фрагмент коду захищеного сховища даних авторизованого користувача (рисунок виконано самостійно)

Фрагмент коду, що зображено на рисунку 4.6, відповідає за сканування QR-кодів для відмітки відвідування в мобільному застосунку. Основний компонент «CameraView» використовується для доступу до камери пристрою. Коли

сканується QR-код, викликається асинхронна функція «handleBarcodeScannedAsync». Камера налаштована на зйомку з задньої сторони («facing="back"»).

Якщо «isProcessing» встановлено в «true», відображається «ActivityIndicator» для індикації завантаження, який займає все доступне місце завдяки стилю «styles.loadingOverlay».

Після сканування QR-коду, компонент «AttendanceMarkedModal» показує інформацію про відмітку відвідування дитини. Він отримує дані через «attendanceData», а видимість модального вікна керується властивістю «visible». Коли модальне вікно закривається, функція «setScannedAttendance(undefined)» очищає дані про сканування.

```

<View style={styles.container}>
  <CameraView
    style={styles.camera}
    onBarcodeScanned={handleBarcodeScannedAsync}
    facing="back"
  >
    {scannedAttendance && (
      <BlurView intensity={100} style={StyleSheet.absoluteFill} />
    )}
    {!scannedAttendance && <CornerOverlay />}
    {isProcessing && (
      <View style={styles.loadingOverlay}>
        <ActivityIndicator size="large" color="white" />
      </View>
    )}
  </CameraView>
  <AttendanceMarkedModal
    attendanceData={scannedAttendance}
    visible={!scannedAttendance}
    onClose={() => {
      setScannedAttendance(undefined);
    }}
  />
</View>

```

Рисунок 4.6 –Фрагмент коду компоненту для сканування QR-кодів (рисунок виконано самостійно)

На рисунку 4.7 зображено фрагмент коду, який містить асинхронну функцію «handleBarcodeScannedAsync», яка обробляє результат сканування QR-коду.

Спочатку функція перевіряє, чи процес сканування вже йде, чи є помилка, або чи вже є відскановані дані, щоб уникнути повторної обробки. Якщо жодна з цих умов не виконана, процес сканування починається.

```
const handleBarcodeScannedAsync = async (
  scanningResult: BarcodeScanningResult
) => {
  if (isProcessing || showingError || scannedAttendance) return;

  setIsProcessing(true);
  try {
    const res = await agent.Children.markAttendance(scanningResult.data);
    setScannedAttendance(res);
  } catch {
    setShowingError(true);
    Alert.alert(t("error"), t("failedToMarkAttendance"), [
      { text: t("ok"), onPress: () => setShowingError(false) },
    ]);
  } finally {
    setIsProcessing(false);
  }
};
```

Рисунок 4.7 – Фрагмент коду обробки даних сканованого QR-коду (рисунок виконано самостійно)

Фрагмент коду, що зображено на рисунку 4.8, є компонентом календаря у мобільному застосунку. Основний компонент «CalendarProvider» забезпечує контекст для роботи з календарем, де передаються налаштування мови, поточна дата та функції обробки змін дати та місяця. Параметр «disabledOpacity» встановлює рівень прозорості для неактивних елементів.

Компонент «ExpandableCalendar» відображає календар, починаючи з першого дня тижня. Нижче йде компонент «TimelineList», який відображає події у вигляді списку, організованого за датами. Події передаються через властивість «eventsByDate», а налаштування таймлайна — через «timelineProps».

Умовний рендеринг перевіряє, чи є вибрана активність («selectedEvent»). Якщо так, відображається модальне вікно «EventEditModal», де можна редагувати вибрану активність.

```

<CalendarProvider
  key={i18n.language}
  date={currentDate}
  onChange={onDateChanged}
  onMonthChange={onMonthChange}
  showTodayButton
  disabledOpacity={0.6}
>
  <ExpandableCalendar firstDay={1} />
  <TimelineList events={eventsByDate} timelineProps={timelineProps} />
  {selectedEvent && (
    <EventEditModal
      event={selectedEvent}
      onClose={() => setSelectedEvent(null)}
      onSave={handleSaveEvent}
    />
  )}
</CalendarProvider>

```

Рисунок 4.8 – Фрагмент коду календаря з розкладом (рисунок виконано самостійно)

Для забезпечення багатомовності застосунку використовується бібліотека «i18n-next». На рисунку 4.9 зображено фрагмент коду, що відповідає за ініціалізацію об'єкта, який забезпечує багатомовність додатку. Спершу отримується збережена мова користувача за допомогою функції «getUserLanguage()». Якщо збережена мова не знайдена, використовується мова за замовчуванням, що встановлена як «uk» (українська). Далі, за допомогою функції «i18n.use(initReactI18next).init()», ініціалізується конфігурація «i18next».

```

const savedLanguage = getUserLanguage();
const defaultLanguage = "uk";
const language = savedLanguage ?? defaultLanguage;

i18n.use(initReactI18next).init({
  resources,
  lng: language,
  fallbackLng: defaultLanguage,
  fallbackNS: "translation",
  interpolation: {
    escapeValue: false,
  },
});

```

Рисунок 4.9 – Фрагмент коду ініціалізації об'єкта управління багатомовності (рисунок виконано самостійно)

У параметрах конфігурації визначаються ресурси перекладу, мова, яка буде використовуватися (отримана або збережена, або мова за замовчуванням), мова за замовчуванням, яка буде використовуватися у випадку, якщо обрана мова не доступна, простір імен за замовчуванням для перекладів та налаштування інтерполяції, де значення не будуть екрановані.

Так як бідь-яка інтернаціоналізація застосунка вимагає перекладів на відповідні мови, було створено 2 файли ресурсів. На рисунку 4.10 наведено приклад ресурсів, що представляє локалізовані рядки для української версії мобільного застосунку «ChildWell». Вони зберігаються в об'єкті, який містить різні сторінки та компоненти застосунку.

```

scheduleCalendarPage: {
  daily: "День",
  monthly: "Місяць",
  selectMode: 'Оберіть режим календаря'
},
generateQrCode: {
  provideCodeToTeacher: "Надайте QR-код викладачу",
  codeNotAvailable: "QR код не доступний",
  regenerateQrCode: "Регенерувати QR-код",
  childGroup: "Група: {{group}}"
},
childAttendanceHistory: {
  loaderTitle: "Завантаження історії відвідувань",
  attendanceDetailsTitle: "Деталі відвідування",
  attendanceSummaryTitle: "Підсумок відвідувань за місяць",
  presentDays: "Присутній: {{count}} днів",
  absentDays: "Відсутній: {{count}} днів",
  attendanceRate: "Рівень відвідуваності: {{rate}}%",
  date: "Дата",
  arrivedAt: "Прибув [d]",
  leftAt: "Забрали [d]",
  close: "Закрити",
},

```

Рисунок 4.10 – Фрагмент ресурсів для перекладу застосунку на українську мову  
(рисунок виконано самостійно)

Наприклад, сторінка із історією відвідувань дитини («childAttendanceHistory») включає заголовки для завантаження історії («loaderTitle»), деталі відвідувань («attendanceDetailsTitle») та підсумок за місяць («attendanceSummaryTitle»). Є рядки для відображення присутніх днів («presentDays»), відсутніх днів («absentDays»), рівня відвідуваності

(«attendanceRate»), дати («date»), часу прибуття («arrivedAt») та часу відбуття («leftAt»). Також передбачено рядок для закриття модального вікна («close»).

Ці ресурси забезпечують перемикання мови «на льоту», без необхідності перезавантажувати застосунок.

## 4.2 Опис інтерфейсу програмного забезпечення

Авторизація є ключовим елементом безпеки та персоналізації в будь-якому сучасному мобільному застосунку, і «ChildWell» не є винятком. Важливість авторизації полягає в забезпеченні захищеного доступу до персональних даних користувачів, а також в наданні індивідуалізованих функцій для різних ролей, таких як батьки, адміністратори та вихователі. Завдяки цьому користувачі можуть бути впевнені, що їхні особисті дані та інформація про дітей зберігаються в безпеці та доступні лише авторизованим особам. На рисунку 4.11 зображено екран входу в мобільний застосунок «ChildWell», який демонструє простий та інтуїтивно зрозумілий інтерфейс для авторизації користувачів.



Рисунок 4.11 – Екран авторизації (рисунок виконано самостійно)

Після успішної авторизації користувач потрапляє на головний екран, який забезпечує швидкий доступ до основних функцій застосунку (див. рис. 4.12). На головному екрані користувач бачить привітання з його іменем, список своїх дітей та їх груп, статус заявок, поданих до навчальних закладів, а також виписані рахунки. Інтерфейс розроблений таким чином, щоб забезпечити зручність і простоту у використанні, дозволяючи батькам швидко отримувати необхідну інформацію та приймати відповідні рішення. На рисунку зображено екран з привітанням користувача, переліком дітей та статусом їхніх заявок.

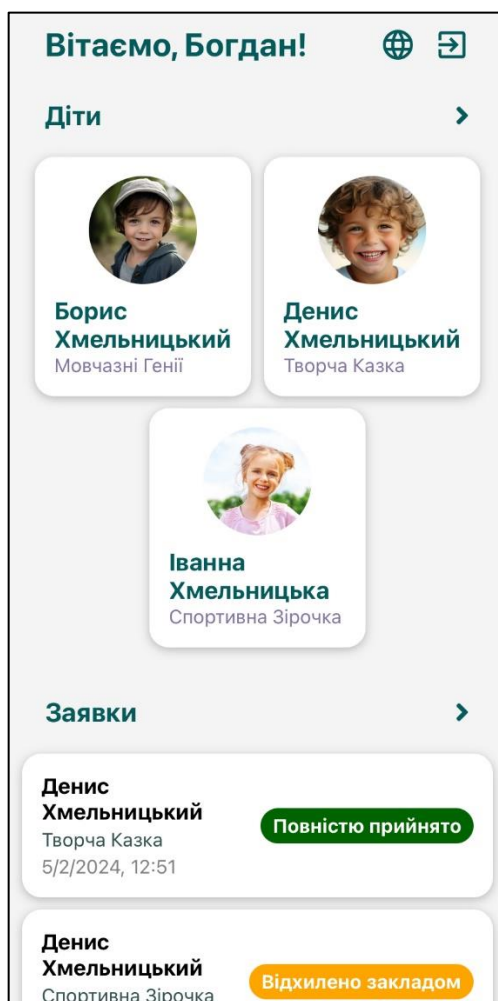
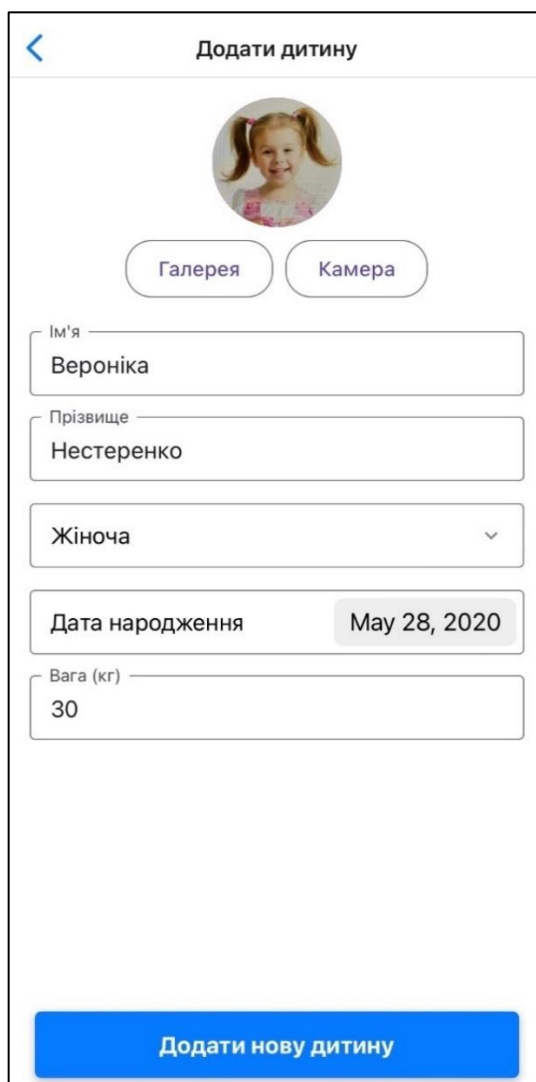


Рисунок 4.12 – Головний екран батьків (рисунок виконано самостійно)

Наступним важливим кроком у процесі використання застосунку є додавання інформації про дитину (див. рис. 4.13). На цьому екрані батьки можуть вводити основні дані про свою дитину, такі як ім'я, прізвище, стать, дата народження та вага. Додатково є можливість завантажити фотографію дитини з галереї або зробити

нову фотографію за допомогою камери. Цей процес забезпечує створення повного профілю дитини, що полегшує управління її відвідуваннями та активностями у майбутньому. На рисунку зображено екран, де батьки можуть додати нову дитину до системи.



Додати дитину

Галерея Камера

Ім'я  
Вероніка

Прізвище  
Нестеренко

Жіноча

Дата народження  
May 28, 2020

Вага (кг)  
30

Додати нову дитину

Рисунок 4.13 – Екран додавання батьками дитини (рисунок виконано самостійно)

Для того, щоб подати заявку на вступ дитини до дитячого садка, батькам спочатку необхідно вибрати відповідний садок. На рисунку 4.14 представлено екран, де можна переглянути доступні дитячі садки. Кожен садок має короткий опис, адресу, години роботи та статус реєстрації, що дозволяє батькам швидко знайти потрібний варіант.

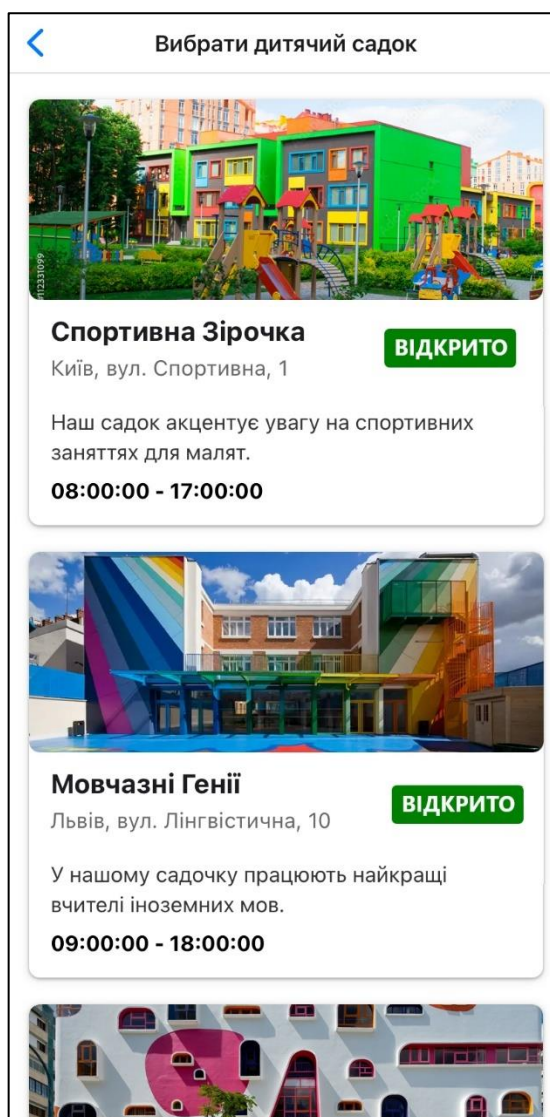


Рисунок 4.14 – Вибір дитячого садка (рисунок виконано самостійно)

Після вибору садка батьки можуть перейти до детальної інформації про обраний заклад, як це зображено на рисунку 4.15. Тут представлена більш детальна інформація про садок, включаючи контактні дані, унікальні особливості закладу та інформацію про наявність відкритої реєстрації. Це допомагає батькам краще ознайомитися з можливостями та умовами закладу перед подачею заявки. Важливими аспектами є зручність доступу до контактної інформації та можливість дізнатися про особливі програми садка, що допомагає зробити обґрунтований вибір. Такий підхід також сприяє прозорості процесу та підвищує довіру батьків до закладу.

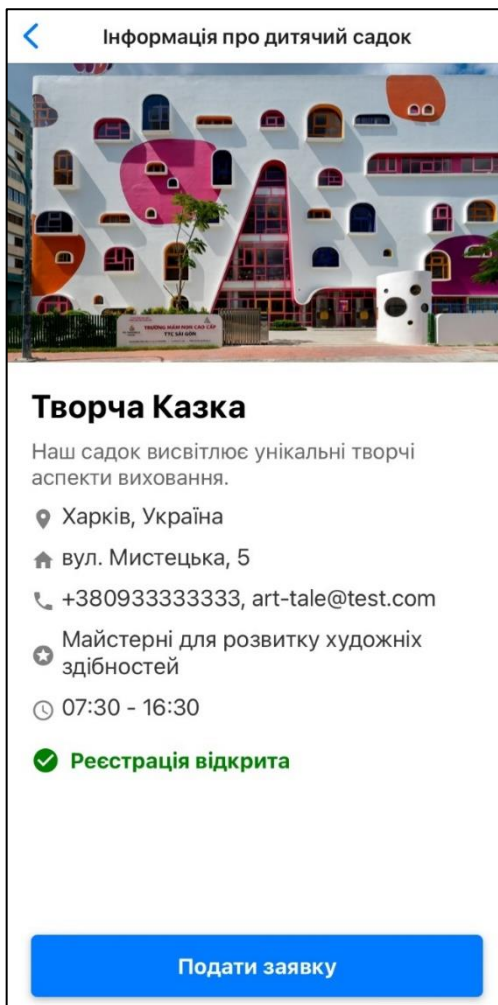


Рисунок 4.15 – Деталі дитячого садка (рисунок виконано самостійно)

Останнім кроком є подача заявки, що показано на рисунку 4.16. У цьому розділі батьки обирають дитину, яку бажають зареєструвати, додають будь-які додаткові коментарі та підтверджують вибір відповідної групи. На екрані також відображається статус заявки, що дозволяє батькам стежити за процесом її розгляду. Такий підхід забезпечує повну прозорість процесу та дозволяє батькам бути в курсі всіх етапів подачі та розгляду заявки. Завдяки зрозумілому та зручному інтерфейсу, користувачі можуть легко взаємодіяти із застосунком, що підвищує їхню довіру та задоволення від користування додатком. Додаток забезпечує ефективну комунікацію між батьками та адміністрацією садочка, надаючи можливість швидко і без зайвих зусиль зареєструвати дитину. Це значно спрощує процес реєстрації та робить його доступним для кожного користувача.

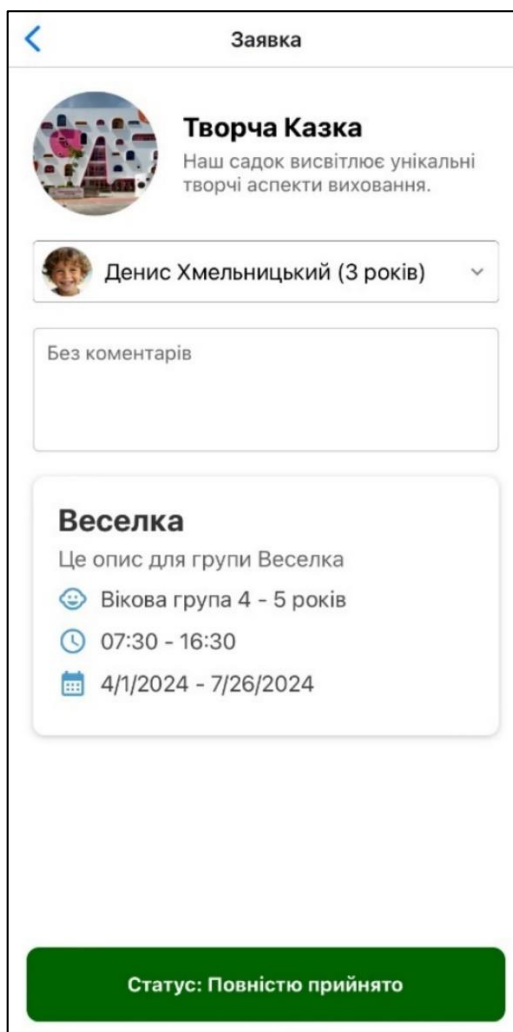


Рисунок 4.16 – Подача заявки (рисунок виконано самостійно)

Скріншот, що зображено на рисунку 4.17, демонструє головне меню для батьків у застосунку «ChildWell». Меню містить всі необхідні функції для управління інформацією про дитину. У верхній частині меню відображається фото та ім'я обраної дитини, що допомагає батькам швидко ідентифікувати дитину, дані якої вони наразі переглядають. Під фото дитини батьки можуть знайти кілька важливих опцій: розклад, QR-код, відвідування. Кожна з цих опцій надає доступ до відповідних розділів інформації та дозволяє батькам ефективно керувати навчальним процесом дитини. Також меню включає опції для зміни мови інтерфейсу та виходу з облікового запису, забезпечуючи користувачам можливість налаштувати застосунок відповідно до своїх потреб і підтримувати безпеку особистих даних.

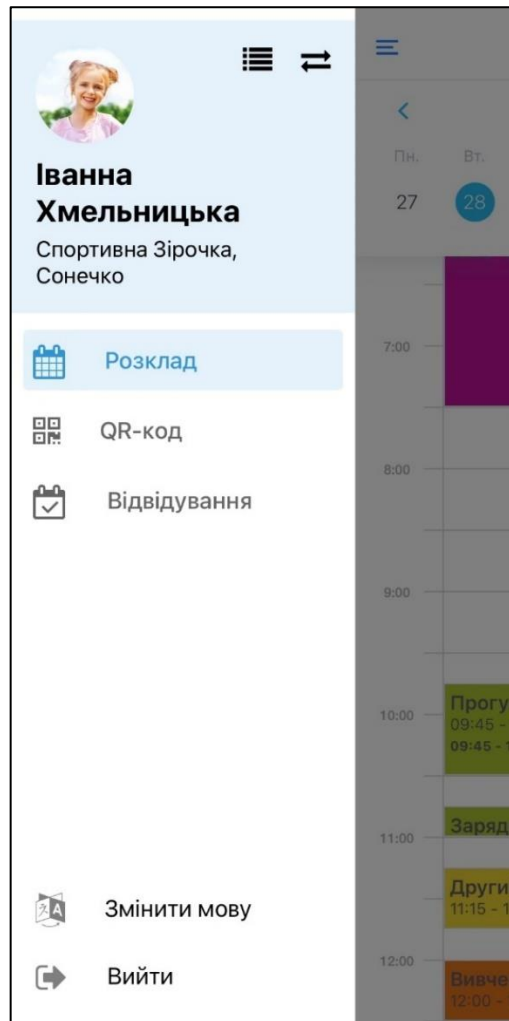


Рисунок 4.17 – Меню батьків (рисунок виконано самостійно)

Завдяки такому інтуїтивно зрозумілому інтерфейсу, батьки можуть легко і швидко отримувати доступ до всієї необхідної інформації та інструментів, що підвищує зручність використання застосунку і загальне задоволення від користування ним.

Далі в застосунку «ChildWell» (див. рис. 4.18) батьки можуть переглядати детальний розклад дня своєї дитини. Інтерфейс дозволяє обирати конкретну дату та переглядати розклад на вибраний день, що забезпечує гнучкість та зручність у плануванні. Кожна подія у розкладі має свій колір, що допомагає легко ідентифікувати різні типи активностей, такі як заняття, прогулянки, прийоми їжі та інші важливі заходи. Це не лише полегшує орієнтацію в щоденному графіку, але й дозволяє батькам бути впевненими, що їхні діти займаються різноманітними та корисними активностями.



Рисунок 4.18 – Розклад дитини (рисунок виконано самостійно)

Розклад відображає точний час початку і завершення кожної активності, що дає змогу батькам точно знати, чим зайнята їхня дитина у будь-який момент дня. Крім того, функціонал додатку дозволяє батькам синхронізувати розклад з їхніми власними календарями, що сприяє кращій організації сімейного часу. Це особливо корисно для батьків, які хочуть планувати свої зустрічі та інші справи, враховуючи розклад своїх дітей.

Таким чином, можливість перегляду розкладу дитини є ключовою функцією «ChildWell», яка сприяє більш ефективному управлінню часом та забезпечує комфорт для батьків у повсякденному житті.

Для забезпечення безпечного та зручного способу відмітки відвідуваності дитини, застосунок «ChildWell» використовує систему QR-кодів (див. рис. 4.19). Ця функція дозволяє батькам генерувати унікальний QR-код для кожної дитини, який можна легко відсканувати вихователю під час прийому або видачі дитини. QR-код містить всю необхідну інформацію про дитину та її групу, що мінімізує ймовірність помилок та забезпечує швидкий і зручний процес реєстрації відвідуваності.



Рисунок 4.19 – QR-код для реєстрації відвідування дитини (рисунок виконано самостійно)

На рисунку показано, як виглядає сторінка з QR-кодом для дитини. Батьки можуть переглядати QR-код, дізнатися інформацію про групу дитини, а також

мають можливість регенерувати QR-код у випадку необхідності. Такий підхід забезпечує високий рівень зручності та безпеки, роблячи процес відвідування дитячого садка більш прозорим і контрольованим.

Використання QR-кодів у додатку «ChildWell» дозволяє значно спростити процеси, пов'язані з відміткою відвідуваності, та забезпечити більш точний і ефективний облік дітей у закладах дошкільної освіти.

Для зручного моніторингу відвідуваності дитини, застосунок «ChildWell» пропонує функцію перегляду статистики відвідувань за місяць (див. рис. 4.20). Цей інструмент дозволяє батькам отримати загальну картину присутності та відсутності дитини у дошкільному закладі. На екрані відображається підсумок за місяць, включаючи кількість присутніх і відсутніх днів, а також загальний рівень відвідуваності у відсотках.

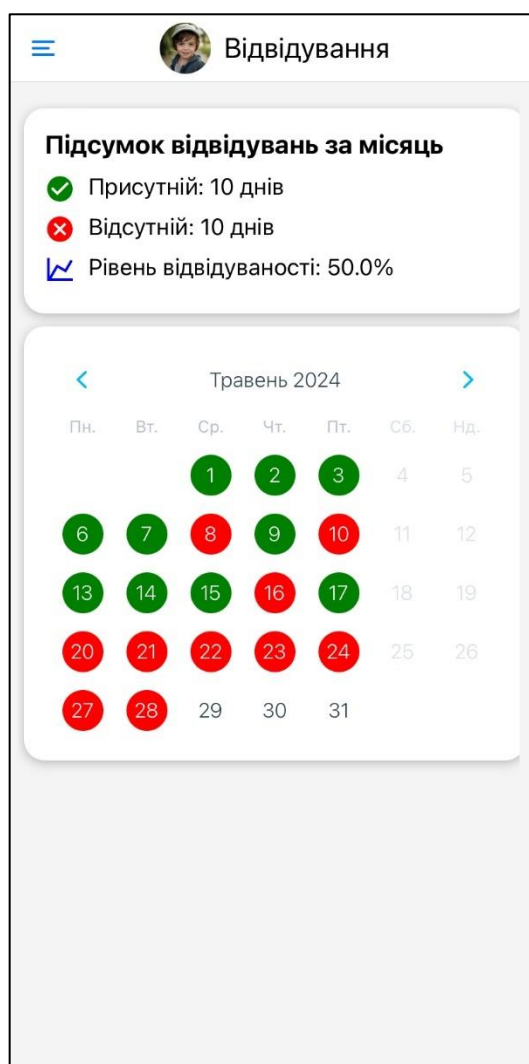


Рисунок 4.20 – Історія відвідувань дитини (рисунок виконано самостійно)

У нижній частині екрану показано календар з розподілом днів на присутні (позначені зеленим кольором) та відсутні (позначені червоним кольором). Це дозволяє батькам швидко і легко оцінити, як часто їх дитина відвідувала заклад протягом місяця. Такий підхід забезпечує простоту та зручність у користуванні, дозволяючи батькам залишатися в курсі відвідуваності дитини.

Важливим аспектом мобільного додатку «ChildWell» є забезпечення зручності та ефективності як для батьків, так і для вихователів. Після розгляду функціоналу, доступного для батьків, перейдемо до можливостей, що надаються вихователям.

На рисунку 4.21 відображено меню для вихователів у мобільному додатку «ChildWell». У меню вказано ім'я вихователя, Валерій Крутко, та назву і опис групи, до якої він належить — група «Сонечко». Вихователь має доступ до розкладу групи (генерація та налаштування), QR-коду для сканування та інформації про відвідування. Додатково, є можливість змінити мову інтерфейсу та вийти із системи. Цей інтерфейс забезпечує вихователям швидкий доступ до необхідних функцій для керування розкладом та відвідуванням дітей.

Вихователі можуть використовувати ці функції для покращення планування та організації навчального процесу. Доступ до розкладу дозволяє їм створювати та редагувати плани занять, враховуючи індивідуальні потреби та інтереси дітей. Функція QR-коду спрощує процес реєстрації та відвідування, забезпечуючи точний облік присутності кожної дитини.

Інтеграція можливості зміни мови інтерфейсу є важливою для забезпечення комфортного використання додатку вихователями з різних мовних середовищ. Також функція виходу із системи гарантує безпеку та конфіденційність даних, що є особливо важливим у роботі з дітьми.

Таким чином, додаток створює комфортні умови для роботи вихователів, дозволяючи їм ефективно планувати та контролювати навчальний процес у своїй групі.

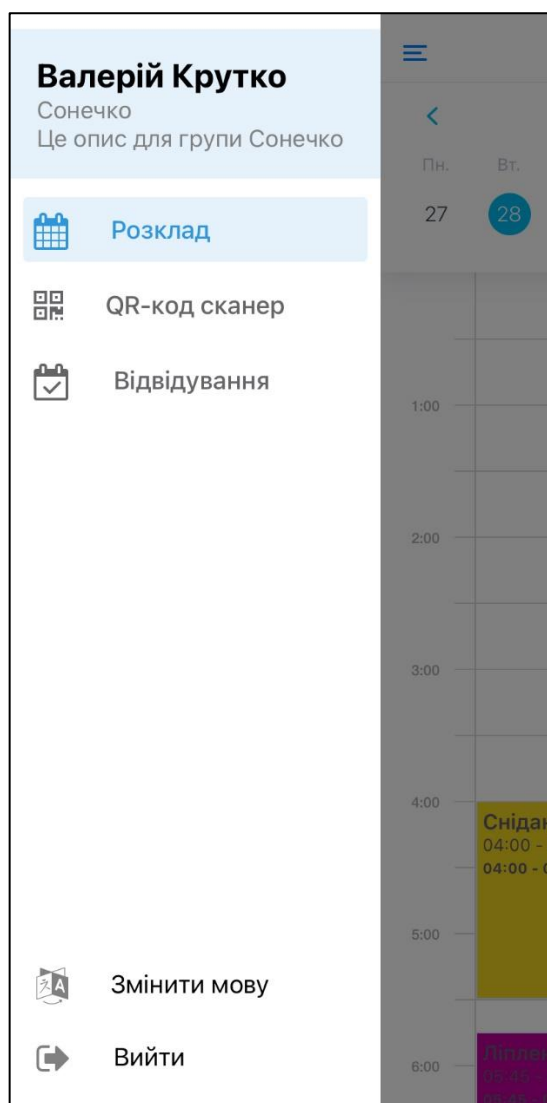


Рисунок 4.21 – Меню вихователя (рисунок виконано самостійно)

Вихователь має можливість генерувати розклад занять для групи. На першому етапі генерації (див. рис. 4.22) вихователь обирає різні типи активностей, які будуть включені до розкладу. Ці активності поділені на категорії, такі як «Сон», «Фізична активність», «Освіта та розвиток» тощо. Всі активності згруповані в своїх категоріях, і кожна категорія має свій колір, що допомагає вихователям швидко орієнтуватися у виборі. Наприклад, активності фізичної активності мають зелений колір, а освітні заходи – оранжевий.

Вихователь може легко вмикати або вимикати потрібні активності за допомогою перемикачів. Деякі активності є обов'язковими і їх неможна вимкнути, їх перемикачі трохи прозорі на зображенні, що робить їх виділеними серед інших. Цей підхід дозволяє вихователям гнучко налаштовувати розклад відповідно до

потреб дітей та забезпечувати збалансований розпорядок дня. Кольорове кодування категорій додає візуальну чіткість і полегшує процес планування, роблячи його інтуїтивно зрозумілим та ефективним.

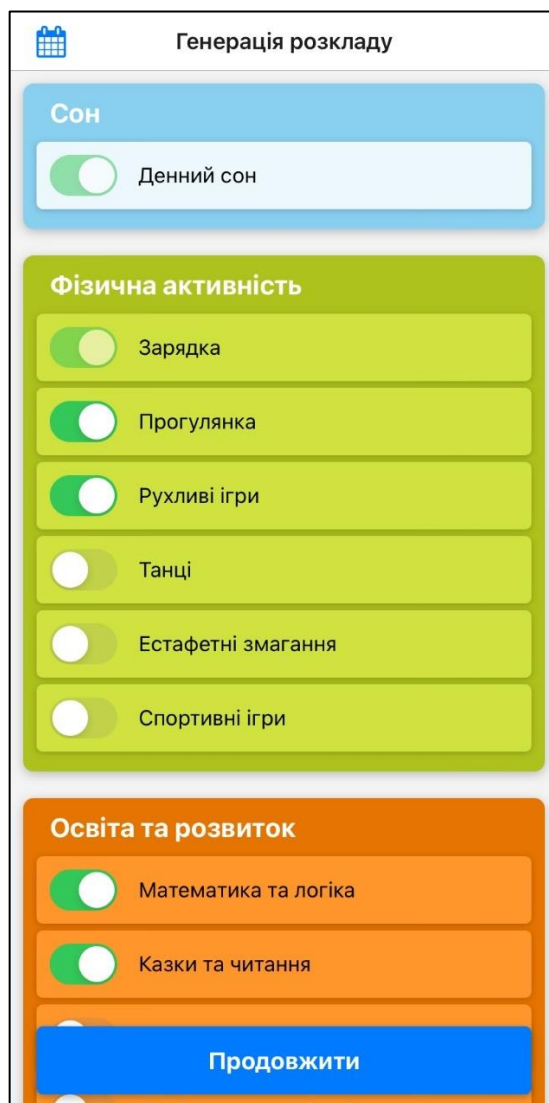


Рисунок 4.22 – Вибір активностей для розкладу (рисунок виконано самостійно)

На другому етапі (див. рис. 4.23) вихователь бачить попередній вигляд розкладу, згенерованого на основі вибраних активностей. Розклад відображається у форматі тижня з зазначенням часу кожної активності. Вихователь має можливість внести примітки до розкладу, регенерувати його або завершити процес створення, натиснувши відповідні кнопки. Цей процес забезпечує гнучкість та зручність у плануванні освітнього процесу, дозволяючи враховувати індивідуальні потреби кожної групи дітей.



Рисунок 4.23 – Попередній вигляд згенерованого розкладу (рисунок виконано самостійно)

На рисунку 4.24 зображено процес ручного редагування календаря. У цьому інтерфейсі вихователь може змінювати час запланованих подій, наприклад, прогулянок, шляхом налаштування точного часу початку та закінчення активності. Це дозволяє персоналізувати розклад для кожної групи дітей, враховуючи їхні індивідуальні потреби та графіки. Після внесення змін користувач має можливість зберегти новий час або скасувати зміни, що забезпечує гнучкість в управлінні розкладом.

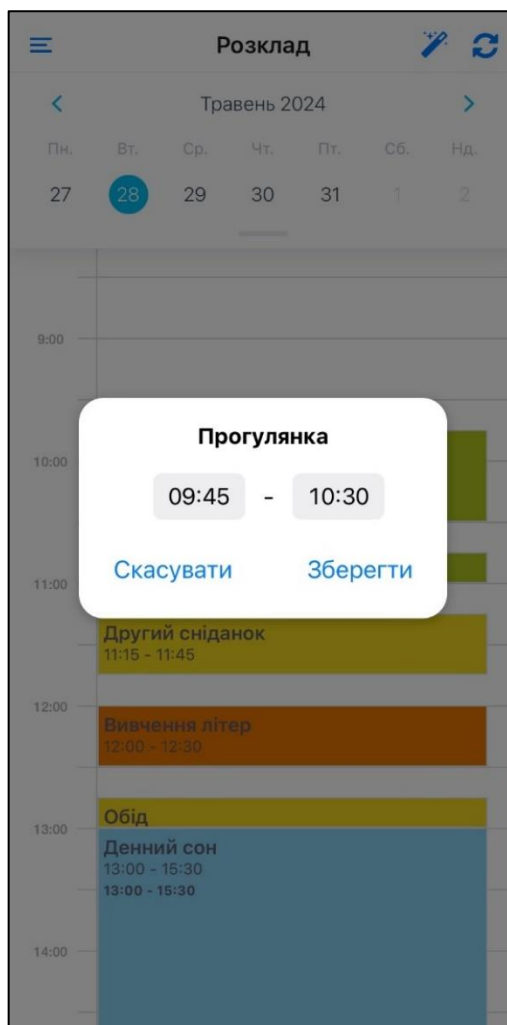


Рисунок 4.24 – Ручне редагування розкладу (рисунок виконано самостійно)

Функція сканування QR-коду є важливим елементом для забезпечення зручності та безпеки у мобільному додатку «ChildWell». На рисунку 4.25 відображено екран сканування QR-коду. Тут можна побачити, що QR-код генерується для кожної дитини окремо. У цьому випадку QR-код належить Іванні Хмельницькій, яка відвідує групу «Сонечко» в садку «Спортивна Зірочка». QR-код містить інформацію про дитину, яку вихователі можуть швидко сканувати для підтвердження її присутності.

Цей QR-код можна легко регенерувати за потреби, що забезпечує додатковий рівень безпеки. Наприклад, у разі втрати або компрометації коду, батьки чи вихователі можуть згенерувати новий код, зберігаючи при цьому безпеку та конфіденційність даних дитини. Такий підхід робить процес відвідування більш автоматизованим та зручним як для вихователів, так і для батьків.

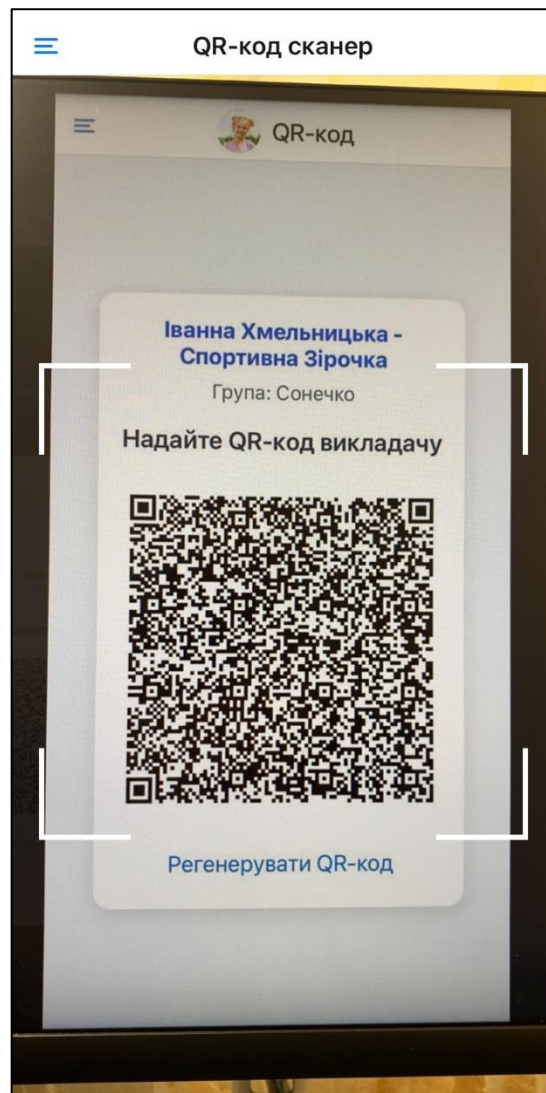


Рисунок 4.25 – Сканування QR-коду (рисунок виконано самостійно)

Після успішного сканування, як показано на рисунку 4.26, з'являється підтвердження про успішне додавання запису про відвідування. Це повідомлення містить ім'я дитини, час реєстрації, а також можливість закрити повідомлення для продовження роботи з додатком. Такий функціонал дозволяє вихователям швидко і ефективно відстежувати відвідуваність дітей, зберігаючи при цьому всі необхідні дані в електронному вигляді. Автоматичне підтвердження та збереження інформації забезпечують точність і зручність у використанні, що є важливим для щоденної роботи вихователів та адміністрації дитячого садка. Таким чином, система надає можливість оперативно і безпечно обробляти дані про присутність дітей, що сприяє покращенню організації навчального процесу та забезпечує комфорт для батьків.

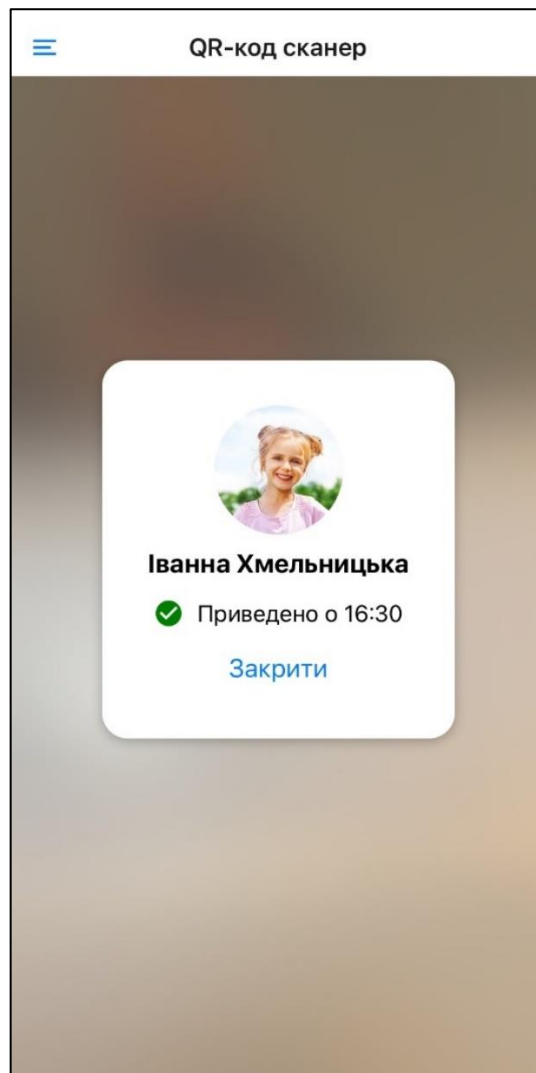


Рисунок 4.26 – Підтвердження реєстрування відвідування дитини (рисунок виконано самостійно)

Сторінка з історією відвідувань дітей, як показано на рисунку 4.27, дозволяє вихователям відстежувати поточний статус відвідувань кожної дитини. На цьому прикладі видно одну дитину, Іванну Хмельницьку, з позначкою «Ще не забрано», що вказує на те, що дитина ще знаходиться в садочку і не була забрана батьками. Це дає вихователям можливість інформувати батьків про місцезнаходження їхніх дітей. Сторінка також містить календар для вибору дати, що дозволяє переглядати історію відвідувань за різні дні. Такий функціонал забезпечує ефективно та зручно управління інформацією про присутність дітей, що сприяє покращенню організації роботи дитячого садка.

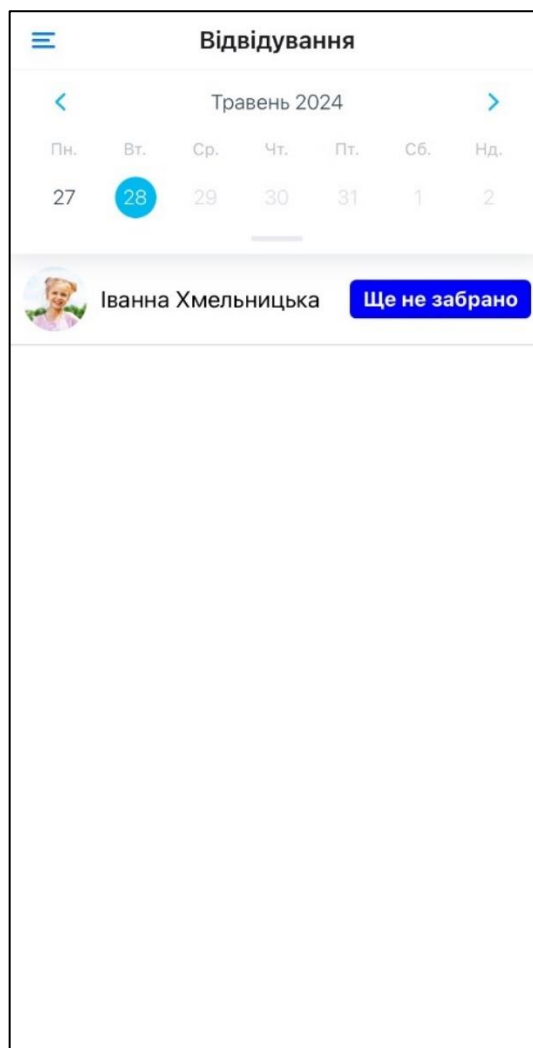


Рисунок 4.27 – Історія відвідувань дітей групи (рисунок виконано самостійно)

Після розгляду функціоналу для вихователів, варто звернути увагу на можливості адміністраторів дитячих садків. Вони мають доступ до головної сторінки, що надає можливість керувати інформацією про дошкільні заклади.

Головний екран для адміністраторів містить перелік дошкільних закладів, які адміністратор може контролювати (див. рис. 4.28). Наприклад, у списку відображаються такі заклади як «Спортивна Зірочка» в Києві, «Мовчазні Генії» у Львові та «Творча Казка» у Харкові. Кожен заклад має інформацію про місцезнаходження, години роботи та контактні дані. Такий інтерфейс дозволяє адміністраторам зручно та ефективно керувати декількома закладами одночасно, надаючи їм усю необхідну інформацію в одному місці.



Рисунок 4.28 – Головний екран адміністраторів садочків (рисунок виконано самостійно)

Меню адміністратора дошкільного закладу надає широкий спектр можливостей для ефективного управління (див. рис. 4.29). У верхній частині меню відображається назва та адреса закладу, що допомагає швидко ідентифікувати його серед інших.

У меню адміністратор може вибрати серед кількох основних розділів. Розділ «Заявки» дозволяє переглядати та обробляти заявки на вступ до закладу. Наступний розділ «Активності» надає можливість додавати, редагувати та видаляти різні активності, що проводяться в закладі. Розділ «Групи» забезпечує управління групами дітей, їхнім розподілом та складом. Останній розділ «Вихователі» призначений для управління персоналом, включаючи додавання нових вихователів, редагування інформації про них та видалення з системи.

Такий підхід до структуризації меню дозволяє адміністраторам швидко і ефективно здійснювати всі необхідні операції, що сприяє покращенню організації роботи дошкільного закладу.

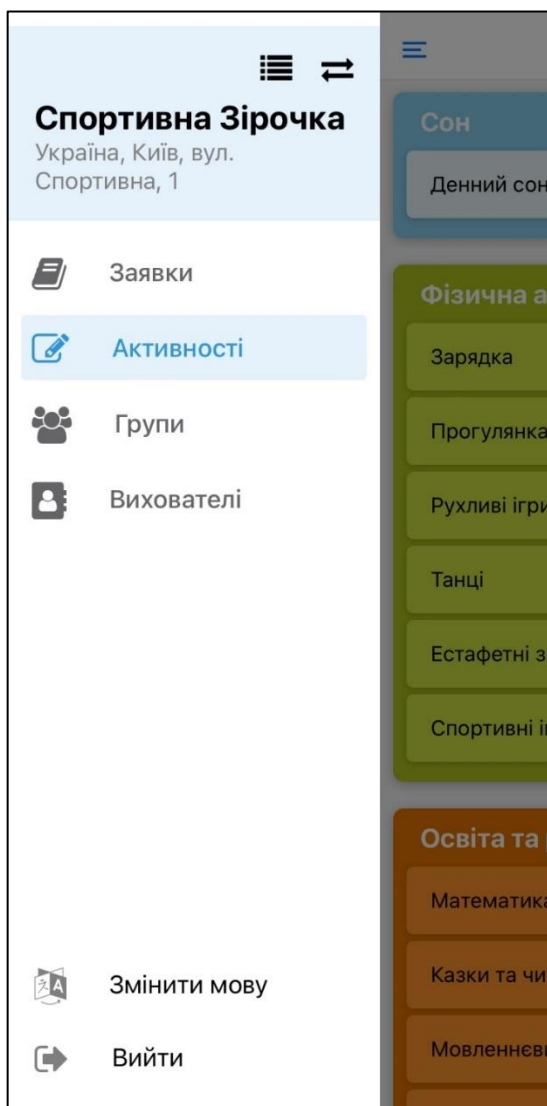


Рисунок 4.29 – Меню адміністраторів садочків (рисунок виконано самостійно)

Адміністратори мають змогу переглядати та керувати заявками на вступ до закладу (див. рис. 4.30). Відображається список заявок з іменами дітей, датами та часом подання. Кожна заявка має статус, що дозволяє швидко визначити її стан. Наприклад, заявка Дениса Хмельницького зі статусом «Відхилено закладом» та заявка Бориса Хмельницького зі статусом «Скасовано». Така візуалізація допомагає адміністраторам ефективно відслідковувати процес прийому нових дітей та своєчасно реагувати на кожну заявку.

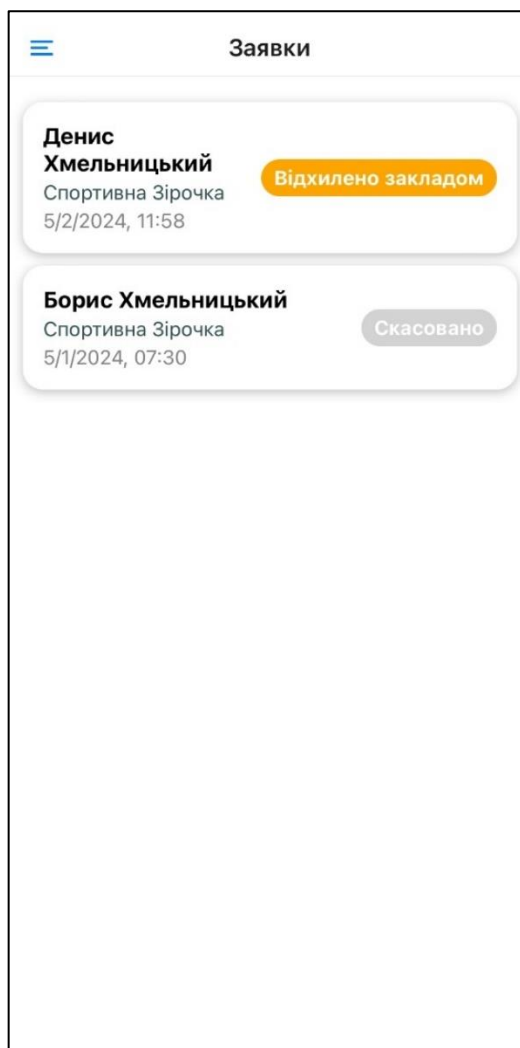


Рисунок 4.30 – Список усіх заявок до садочку (рисунок виконано самостійно)

Адміністратори можуть додавати нові активності та редагувати існуючі для покращення розкладу занять (див. рис. 4.31). У розділі «Активності» відображаються різні типи активностей, такі як сон, фізична активність та освітні заходи. Кожна активність згрупована за своїм типом, а кожна група активностей має свій унікальний колір для кращого візуального сприйняття. Це дозволяє швидко і легко знаходити потрібну активність і бачити структуру дня.

Наприклад, у розділі «Фізична активність» є опції для додавання зарядки, прогулянки, рухливих ігор, танців, естафетних змагань та спортивних ігор. У розділі «Освіта та розвиток» можна додавати такі активності, як математика та логіка, казки та читання, мовленнєвий розвиток. Адміністратор також може додавати нові категорії активностей або змінювати існуючі, забезпечуючи

гнучкість та індивідуальний підхід до організації занять. Цей інтерфейс надає адміністраторам можливість забезпечити комплексний розвиток дітей, комбінуючи різні типи активностей.

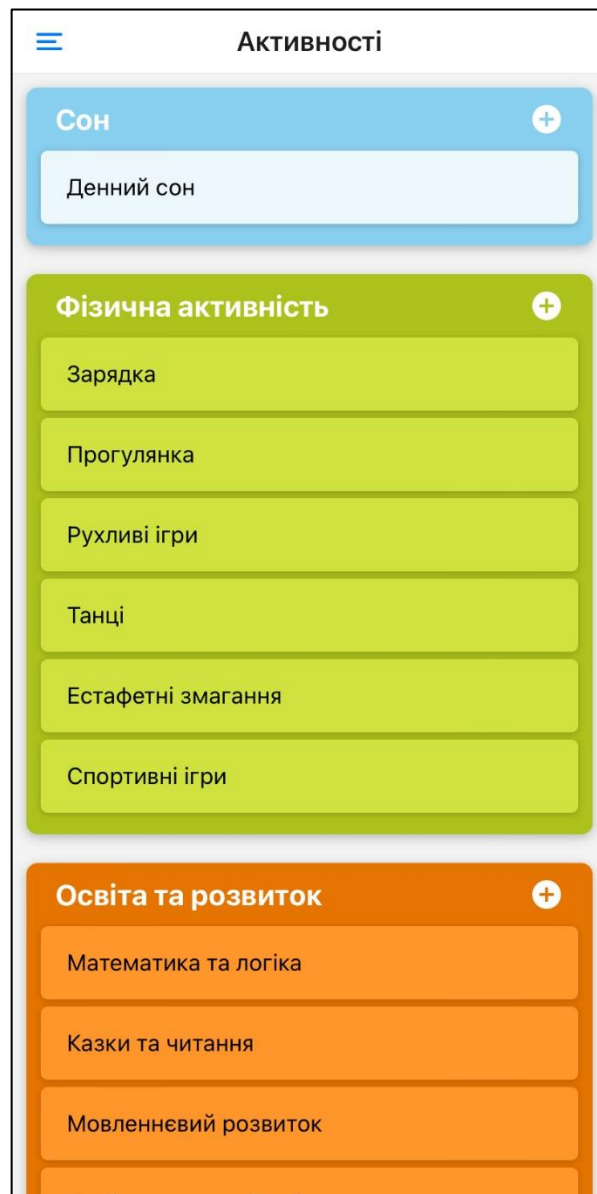
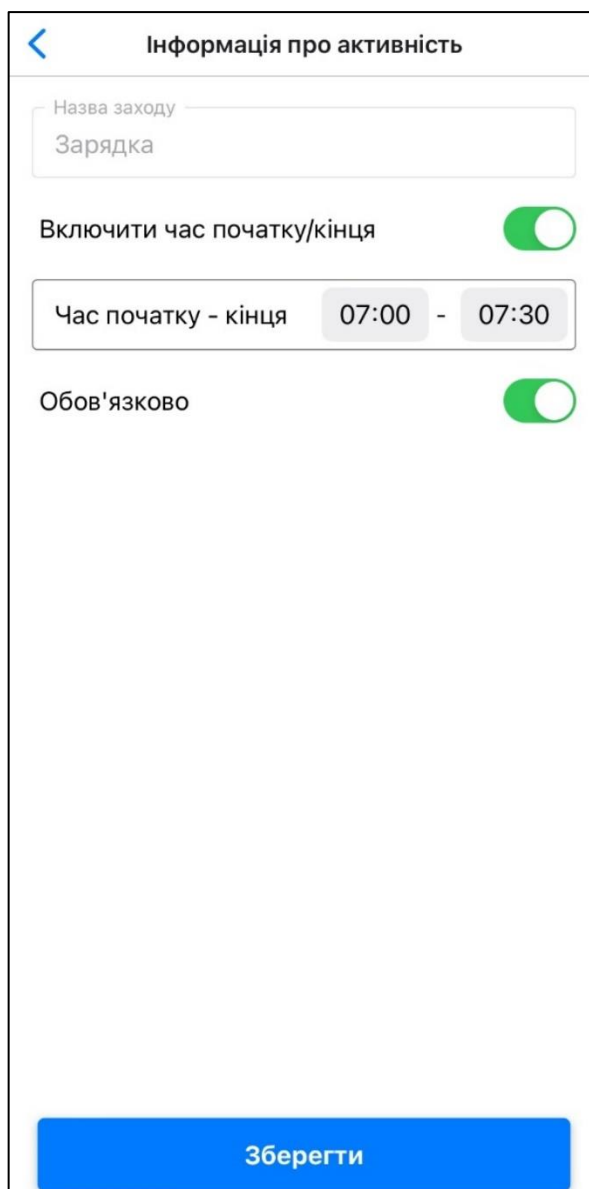


Рисунок 4.31 – Список активностей садочку (рисунок виконано самостійно)

При натисканні на кнопку додавання нової активності відкривається форма (див. рис. 4.32), де можна ввести назву заходу, встановити час початку та кінця, а також вказати, чи є ця активність обов'язковою. Це дозволяє адміністраторам налаштовувати розклад відповідно до потреб дітей та програм навчання. Після введення необхідних даних потрібно натиснути кнопку «Зберегти», щоб активність була додана до списку і стала доступною для використання у розкладі. Форма

забезпечує простий та інтуїтивно зрозумілий спосіб управління активностями, що дозволяє ефективно організовувати навчальний процес.



Інформація про активність

Назва заходу  
Зарядка

Включити час початку/кінця

Час початку - кінця 07:00 - 07:30

Обов'язково

Зберегти

Рисунок 4.32 – Редагування активності (рисунок виконано самостійно)

На сторінці «Список вихователів» адміністратори мають змогу переглядати інформацію про вихователів дитячого садка (див. рис. 4.33). Кожен вихователь відображається в окремому блоці з іменем, прізвищем, групою, в якій він працює, та гендерною ознакою. У прикладі зображені два вихователі: Валерій Ігорович Крутко, який працює у групі «Сонечко» та має чоловічу гендерну ознаку, та Ганна Степанівна Даниленко, яка також працює у групі «Сонечко» та має жіночу стать.

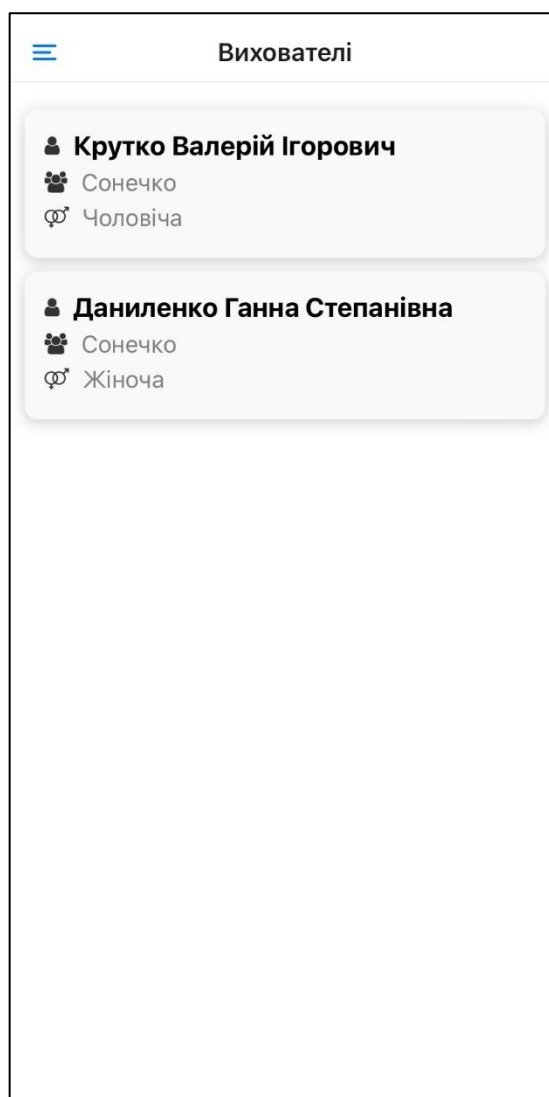


Рисунок 4.33 – Список вихователів групи (рисунок виконано самостійно)

На сторінці «Групи» адміністратори можуть переглядати деталі про кожну групу дитячого садка (див. рис. 4.34). Відображена група «Сонечко» містить всю необхідну інформацію: опис, терміни навчального періоду, графік роботи, віковий діапазон дітей у групі, основного вихователя та підлеглого вихователя. Основний вихователь Ганна Степанівна Даниленко та підлеглий вихователь Валерій Ігорович Крутко відповідальні за цю групу. Також вказано терміни роботи групи з 1 квітня 2024 року до 26 липня 2024 року, а графік роботи з 08:00 до 17:00. Віковий діапазон дітей у групі становить від 3 до 5 років. Ця інформація допомагає адміністраторам мати чітке уявлення про структуру груп і їх склад, що сприяє ефективному управлінню навчальним процесом

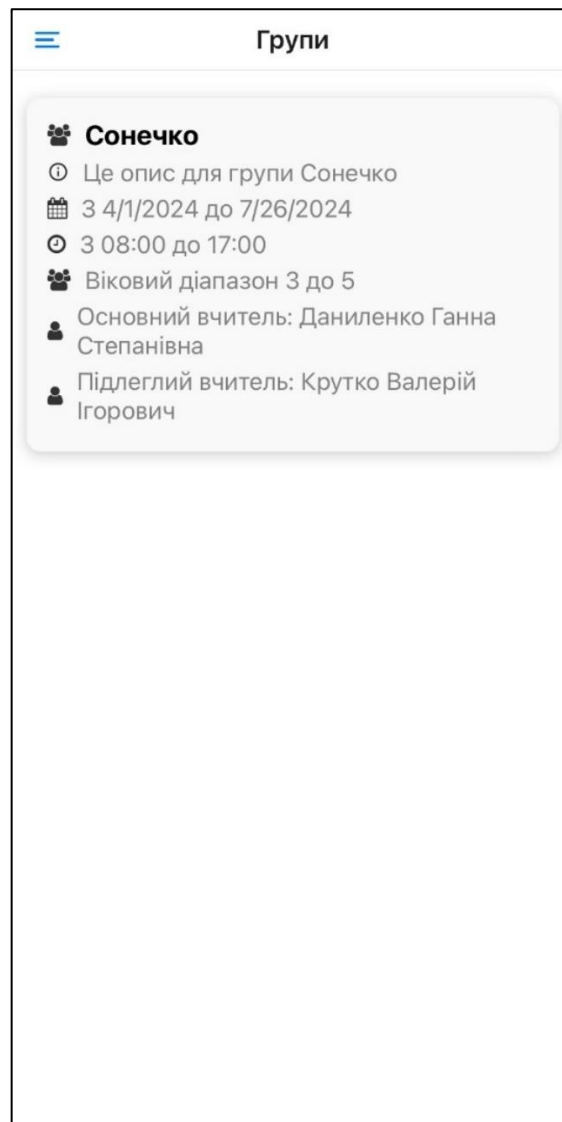


Рисунок 4.34 – Список вихователів групи (рисунок виконано самостійно)

Інтерфейс користувача цього додатка для дитячого садка розроблений для забезпечення зручності та простоти використання для всіх учасників. Меню батьків забезпечує легкий доступ до важливих функцій, таких як перегляд розкладу, сканування QR-кодів для відмічання присутності, історія відвідувань та зміна мови. Наприклад, меню батьків містить зрозумілу інформацію про обрану дитину та ключові опції для взаємодії.

Для вихователів інтерфейс передбачає можливість зручно генерувати та редагувати розклад за допомогою простих у використанні інструментів. Функція сканування QR-кодів для відмічання часу приходу дітей інтегрована в інтерфейс

для швидкого доступу. Процес генерації розкладу включає вибір активностей та їхнього часу проведення.

Адміністратори мають розширені можливості управління через інтуїтивно зрозуміле меню, яке містить опції для роботи із заявками, активностями, групами та вихователями. Додавання та редагування активностей здійснюється через чіткі та структуровані форми, що дозволяє легко налаштувати необхідні параметри.

Загалом, інтерфейс користувача цього додатка спрямований на максимальну простоту та зручність використання, забезпечуючи ефективну комунікацію та управління процесами в дитячому садку для батьків, вихователів та адміністраторів.

## 5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування програмного забезпечення є надзвичайно важливим етапом розробки, який дозволяє забезпечити високу якість кінцевого продукту. Воно спрямоване на виявлення та усунення помилок, перевірку функціональності та відповідності програмного забезпечення вимогам користувачів. Основна мета тестування - гарантувати, що програмний продукт працює стабільно та ефективно в різних умовах експлуатації.

Застосування тестування у розробці програмного забезпечення має декілька ключових аспектів. По-перше, воно допомагає виявити помилки на ранніх етапах розробки, що дозволяє значно знизити витрати на їх виправлення. По-друге, тестування забезпечує впевненість у тому, що всі функції програми працюють правильно, а взаємодія з користувачем є зручною та інтуїтивно зрозумілою. По-третє, воно дозволяє перевірити, як програмний продукт поводить себе в реальних умовах використання, включаючи різні сценарії взаємодії користувача з системою.

Мета тестування програмного забезпечення «ChildWell» полягає у забезпеченні його стабільності, функціональності та відповідності вимогам користувачів. Це включає перевірку всіх основних функцій додатку, таких як авторизація та реєстрація користувачів, управління профілем, генерація розкладу занять, сканування QR-кодів, перегляд історії відвідувань та адміністрування закладу. Особлива увага приділяється перевірці коректності відображення даних, зручності користувацького інтерфейсу та адаптивності додатку на різних пристроях і платформах.

Тестування програмного забезпечення «ChildWell» проводилося в реальних умовах, використовуючи різні пристрої та операційні системи для забезпечення максимальної точності та достовірності результатів. Застосунок тестувався з використанням середовища Expo Go на iPhone 11, що працює під управлінням операційної системи iOS версії 17.5.1. Це дозволило перевірити коректність роботи програмного забезпечення на одному з популярних мобільних пристроїв, забезпечуючи високу ймовірність виявлення потенційних проблем та помилок.

iPhone 11 є однією з найбільш розповсюджених моделей смартфонів в світі, тому тестування саме на цій моделі покриває велику кількість пристроїв користувачів. iPhone 11 має встановлену iOS 17.5.1. Expo Go є зручним інструментом для тестування, що дозволяє швидко та ефективно запускати додаток на мобільному пристрої, забезпечуючи високий рівень деталізації та точності перевірки функціональності.

Тестування розкладу занять здійснювалося для забезпечення правильності та зручності роботи з розкладом як для вихователів, так і для адміністраторів.

Генерація розкладу перевірялася шляхом вибору різних типів активностей, які повинні бути включені до щоденного розкладу. Особлива увага приділялася коректності роботи перемикачів для вибору активностей, їх поділ на категорії та збереження налаштувань. Під час тестування була перевірена можливість додавання нових активностей та їх відображення в загальному списку. Важливо було забезпечити, щоб всі обрані активності правильно зберігалися і відображались у згенерованому розкладі. Перевірка включала в себе також тестування функціоналу обов'язкових активностей, які не можна було вимкнути, що було відображено прозорими перемикачами.

Редагування розкладу також проходило детальне тестування. Перевірялося, як додаток обробляє зміни у вже згенерованому розкладі. Це включало зміну часу активностей, додавання або видалення активностей, а також зміни у категоріях активностей. Всі зміни повинні правильно відображатися та зберігатися, щоб забезпечити безперебійну роботу розкладу та відповідність потребам вихователів та дітей.

Тестування сканування QR-кодів відіграло ключову роль у забезпеченні функціональності додатка «ChildWell». Сканування QR-коду було перевірено за допомогою різних сценаріїв, включаючи сканування при вході до дитячого садка та при виході. Під час тестування була оцінена швидкість та точність сканування, а також стабільність роботи додатка під час використання камери пристрою. Особлива увага приділялася ситуаціям з використанням невалідних та прострочених QR-кодів. У таких випадках система повинна була коректно

розпізнавати помилки та висвітлювати відповідні повідомлення про помилку, що дозволяло запобігти некоректному додаванню записів про відвідування.

Перевірка успішного додавання запису про відвідування здійснювалася для впевненості в тому, що кожне сканування QR-коду правильно реєструє дитину у системі. Тестувалося відображення повідомлень про успішну реєстрацію, включаючи час і дату сканування, ім'я дитини та статус відвідування. Це дозволяло батькам та вихователям отримувати точну і актуальну інформацію про присутність дитини у садку. Важливою складовою тестування було також забезпечення безперебійної роботи системи під час використання камери для сканування QR-кодів, що гарантувало стабільну та надійну роботу додатка.

Під час тестування були створені різноманітні сценарії для перевірки всіх можливих варіантів використання QR-кодів. Перевірялися як коректні коди, так і невалідні або прострочені коди, щоб упевнитися в надійності та точності системи. У випадку сканування невалідних або прострочених QR-кодів, додаток правильно обробляв ці ситуації та висвітлював повідомлення про помилку, вказуючи на необхідність перевірки коду. Це дозволяло уникати помилок у реєстрації відвідувань та забезпечувало високу точність даних.

Таким чином, тестування QR-кодів у додатку «ChildWell» охоплювало широкий спектр сценаріїв і забезпечувало високу надійність та точність роботи системи, що є критично важливим для ефективного управління відвідуваннями у дитячих садках.

Було проведено тестування функціональності історії відвідувань, щоб переконатися у правильному оновленні даних після сканування QR-кодів. Після того як дитину приводили до садочка і сканували QR-код, історія відвідувань миттєво оновлювалася, відображаючи інформацію про прибуття. Аналогічно, після того як дитину забирали і сканували QR-код, дані в історії відвідувань також оновлювалися, фіксуючи час виходу. Це забезпечувало точне та своєчасне відображення статусу присутності кожної дитини у системі. Усі перевірки показали, що додаток правильно відстежує та оновлює інформацію про відвідування, гарантуючи надійну роботу цієї функції.

Було проведено ретельне тестування функціональності заявок до дитячих садочків, щоб переконатися у правильній роботі системи на кожному етапі процесу. Відповідно до діаграми станів і переходів заявки, тестування включало кілька ключових сценаріїв.

Перевірялося надсилання заявок до садочків, щоб впевнитися, що батьки можуть успішно надіслати заявку до обраного садочка. Після натискання кнопки «Надіслати заявку» заявка мала з'явитися в системі з правильним статусом «Заявку надіслано». Тестувалося відхилення заявки з боку садочка. У випадках, коли дитячий садочок відхилив надіслану заявку, система повинна була змінити статус на «Заявку відхилено садочком» і сповістити про це батьків. Також перевірялося відкриття заявки батьками. Після того, як батьки надіслали заявку, вони мали можливість її відкрити, і система змінювала статус на «Заявку відкрито батьками», щоб заявка більше не відображалася як активна.

Тестування прийняття заявки садочком було важливим для впевненості, що при прийнятті заявки садочком її статус змінювався на «Заявку прийнято садочком». Цей статус мав відображатися коректно, і батьки отримували відповідне повідомлення. Перевірялося відхилення заявки батьками після того, як садочок її прийняв, і система повинна була змінювати статус на «Заявку відхилено батьками».

Під час тестування було враховано всі можливі стани заявки, щоб переконатися в коректності їх відображення та переходів між ними. Було важливо, щоб усі зміни статусів заявок були належним чином відображені у системі, і користувачі отримували відповідні повідомлення про зміну статусу. Це забезпечувало прозорість процесу подачі та обробки заявок, дозволяючи батькам та адміністраторам дитячих садочків ефективно взаємодіяти з системою.

Підсумовуючи виконані роботи з тестування, можна зробити висновок, що додаток «ChildWell» успішно пройшов всі етапи мануального тестування. Були перевірені генерація та редагування розкладу занять, сканування QR-кодів, реєстрація відвідувань, а також подача заявок до дитячих садочків. Тестування QR-кодів включало перевірку валідних, невалідних та прострочених кодів. Система

правильно реагувала на кожен сценарій, виводячи повідомлення про помилку при скануванні невалідних або прострочених кодів. Перевірка історії відвідувань показала, що після сканування QR-кодів історія правильно оновлюється, забезпечуючи точне відображення часу, дати сканування, імені дитини та статусу відвідування. Тестування процесу подачі заявок охоплювало всі можливі стани заявки, включаючи надсилання, відкликання, відхилення та прийняття заявок садочком і батьками. Система коректно обробляла кожен етап процесу. Додаток продемонстрував стабільну роботу, високу швидкість обробки запитів та точність у відображенні інформації.

Перспективи автоматизації тестування в майбутньому є дуже важливими для забезпечення постійної якості програмного забезпечення. Автоматизація дозволить швидше та ефективніше виявляти помилки, знижуючи ризики, пов'язані з людським фактором. Розробка автоматизованих тестів для ключових функцій додатка, таких як генерація розкладу та сканування QR-кодів, може значно скоротити час на тестування та підвищити надійність додатка. Впровадження інструментів для безперервної інтеграції та доставки дозволить забезпечити стабільну роботу додатка при кожному оновленні, що є важливим аспектом для підтримки високої якості програмного забезпечення у довгостроковій перспективі.

## 6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Участь у міжнародній конференції стала важливим етапом у розвитку дослідницької роботи, присвяченої програмному інструменту для управління розкладом та харчуванням дітей у дошкільних закладах. Було написано тези для XXVIII Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». Форум було проведено 16-18 квітня 2024 року. На цій конференції було отримано диплом за 2-ге місце в секції «Програмна Інженерія. Інформаційні технології в освіті» (див. додаток Б) на конференції «Інформаційні інтелектуальні системи».

Під час конференції було представлено тези за темою «Аналіз використання штучного інтелекту для генерації розкладу занять і харчування дітей в закладах дошкільної освіти». Ці тези стосувалися використання штучного інтелекту, зокрема технології GPT (Generative Pre-trained Transformer), для автоматизації створення розкладу, який враховує потреби різних вікових груп дітей, а також їх харчові звички. Завдяки використанню штучного інтелекту система дозволяє вихователям створювати адаптовані розклади, які оптимально підходять для конкретних груп дітей, забезпечуючи їм збалансовані активності та харчування. Це не лише сприяє покращенню освітнього процесу, але й допомагає забезпечити здоровий та гармонійний розвиток дітей. Батьки, в свою чергу, отримують зручний інструмент для відстеження щоденних активностей та харчування своїх дітей, що дозволяє їм бути більш поінформованими про їхній добробут.

Дослідження підкреслило актуальність оптимізації освітніх процесів та планування харчування в дошкільних закладах з використанням сучасних технологічних рішень. Застосування GPT моделей дозволяє реалізувати індивідуальний підхід до кожної вікової групи дитячого садка, сприяючи гармонійному розвитку дітей. Використання штучного інтелекту для генерації розкладів та харчування враховує не лише освітні, але й фізіологічні потреби дітей, забезпечуючи ефективно та здорове навчальне середовище.

Наукова новизна роботи полягає в інтеграції штучного інтелекту для розробки комплексної системи, що враховує освітні та фізіологічні потреби дітей у

дошкільних закладах. Ця система оптимізує розклад занять та режим харчування, що є значним кроком вперед у використанні штучного інтелекту в сфері раннього розвитку дітей. Результати дослідження демонструють, що впровадження штучного інтелекту може суттєво підвищити якість освітніх і виховних процесів.

Дослідження вказує на значні переваги використання описаних технологічних рішень. Система спрямована на врахування вікових потреб, забезпечуючи ефективне планування та керування розкладом. З подальшим розвитком технології очікується, що система стане ще більш гнучкою та орієнтованою на забезпечення конкретних потреб вихованців, педагогів та батьків, вносячи істотний вклад у розвиток сфери раннього розвитку дітей.

Участь у конференції дала можливість продемонструвати досягнення та обговорити подальші кроки в напрямку удосконалення технологічних рішень для дошкільних навчальних закладів. Представлені тези були високо оцінені експертами, що підтверджує актуальність і важливість проведеного дослідження. Використання штучного інтелекту у дошкільній освіті відзначається значним потенціалом для підвищення ефективності освітнього процесу та забезпечення всебічного розвитку дітей. Застосування GPT-моделей дозволяє створювати індивідуалізовані розклади, що відповідають освітнім та фізіологічним потребам дітей. Батьки, отримуючи детальну інформацію про щоденні активності та харчування своїх дітей, можуть краще контролювати та підтримувати їхній розвиток.

З подальшим розвитком технологій очікується, що системи на базі штучного інтелекту стануть ще більш досконалими, забезпечуючи адаптивне та персоналізоване навчання для кожної дитини. Таким чином, участь у міжнародній конференції підтвердила значимість дослідження та дала імпульс для подальшого розвитку і впровадження інновацій у сфері дошкільної освіти..

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено дослідження в області автоматизації процесів у дошкільних закладах. Визначено основні вимоги та сформульовано задачу для мобільного застосунку «ChildWell». В рамках дизайну застосунку було проведено аналіз користувачів та розроблено інтерактивний прототип, що забезпечує ефективне взаємодію вихователів, адміністраторів та батьків із системою.

Для кращого розуміння функціональності і пришвидшення розробки застосунку було створено UML-діаграми (Use Case, Deployment, Sequence, State & Transition діаграми), які детально описують процеси в системі, від планування розкладу дня до відстеження відвідувань. Застосунок розроблено використовуючи сучасні технології, такі як TypeScript, React Native і фреймворк Expo.

При розробці мобільного застосунку особлива увага приділялася до UX і UI частині, щоб забезпечити інтуїтивно зрозуміле та привабливе використання системи. Було обрано кольорову палітру, яка символізує спокій, що є важливим для батьків. Також було створено макет в дизайн-системі «Figma».

Весь проект було розроблено з використанням гнучкої методології Agile, що забезпечило ефективне управління процесами та адаптивність до змін під час розробки. Процес керування проектом здійснювався через систему Azure DevOps, яка дозволила забезпечити високий рівень співпраці в команді та ефективне відстеження прогресу робіт.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Важливість розпорядку дня в житті дитини – Розвиток дитини. URL: <https://childdevelop.com.ua/articles/upbring/4696/> (дата звернення: 13.05.2024).
2. Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. URL: [https://www.researchgate.net/publication/360132717\\_Artificial\\_Intelligence\\_AI\\_in\\_early\\_childhood\\_education\\_Curriculum\\_design\\_and\\_future\\_directions](https://www.researchgate.net/publication/360132717_Artificial_Intelligence_AI_in_early_childhood_education_Curriculum_design_and_future_directions) (дата звернення 13.05.2024).
3. Preschool and childcare mobile app - brightwheel. URL: <https://mybrightwheel.com/childcare-mobile-app/> (дата звернення 13.05.2024).
4. Procare - the Leading App for Child Care and Daycare. URL: <https://www.procaresoftware.com/capabilities/child-care-mobile-app/> (дата звернення 13.05.2024).
5. Remini App - Telling Children Life Stories. URL: <https://remini.me/site/> (дата звернення 13.05.2024).
6. HiMama - Child Care Apps with Daycare Daily Sheets. URL: [https://www.himama.com/parents/request\\_invite](https://www.himama.com/parents/request_invite) (дата звернення 13.05.2024).
7. Azure DevOps Services. URL: <https://azure.microsoft.com/en-us/products/devops> (дата звернення 13.05.2024).
8. Research Directions in Requirements Engineering. URL: [https://www.researchgate.net/publication/4250874\\_Research\\_Directions\\_in\\_Requirements\\_Engineering](https://www.researchgate.net/publication/4250874_Research_Directions_in_Requirements_Engineering) (дата звернення 13.05.2024).
9. User Access Privacy in OAuth 2.0 and OpenID Connect. URL: [https://www.researchgate.net/publication/346374429\\_User\\_Access\\_Privacy\\_in\\_OAuth\\_20\\_and\\_OpenID\\_Connect](https://www.researchgate.net/publication/346374429_User_Access_Privacy_in_OAuth_20_and_OpenID_Connect) (дата звернення 13.05.2024).
10. Радіоелектроніка та молодь у XXI столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи" : матеріали 28-го Міжнар. молодіж. форуму, 16–18 квіт. 2024 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2024. – 958 с.

## ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

**UNICHECK**  
by Turnitin

Ім'я користувача: Олійник Олена Володимирівна каф. ПІ ID перевірки: 1016310927

Дата перевірки: 02.06.2024 15:16:42 EEST Тип перевірки: Doc vs Library

Дата звіту: 02.06.2024 15:20:45 EEST ID користувача: 100012353

---

Назва документа: 2024\_Б\_ПІ\_ПЗПІ-20-3\_Нестеренко\_В\_В\_скорочений

Кількість сторінок: 85 Кількість слів: 13434 Кількість символів: 107127 Розмір файлу: 2.70 MB ID файлу: 1016107711

---

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**1.88%**  
**Схожість**

Найбільша схожість: 0.7% з джерелом з Бібліотеки (ID файлу: 1008270568)

Пошук збігів з Інтернетом не проводився

1.88% Джерела з Бібліотеки 234 ..... Сторінка 87

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 18 сторінок

Рисунок А.1 – Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

ДОДАТОК Б  
Слайди презентації

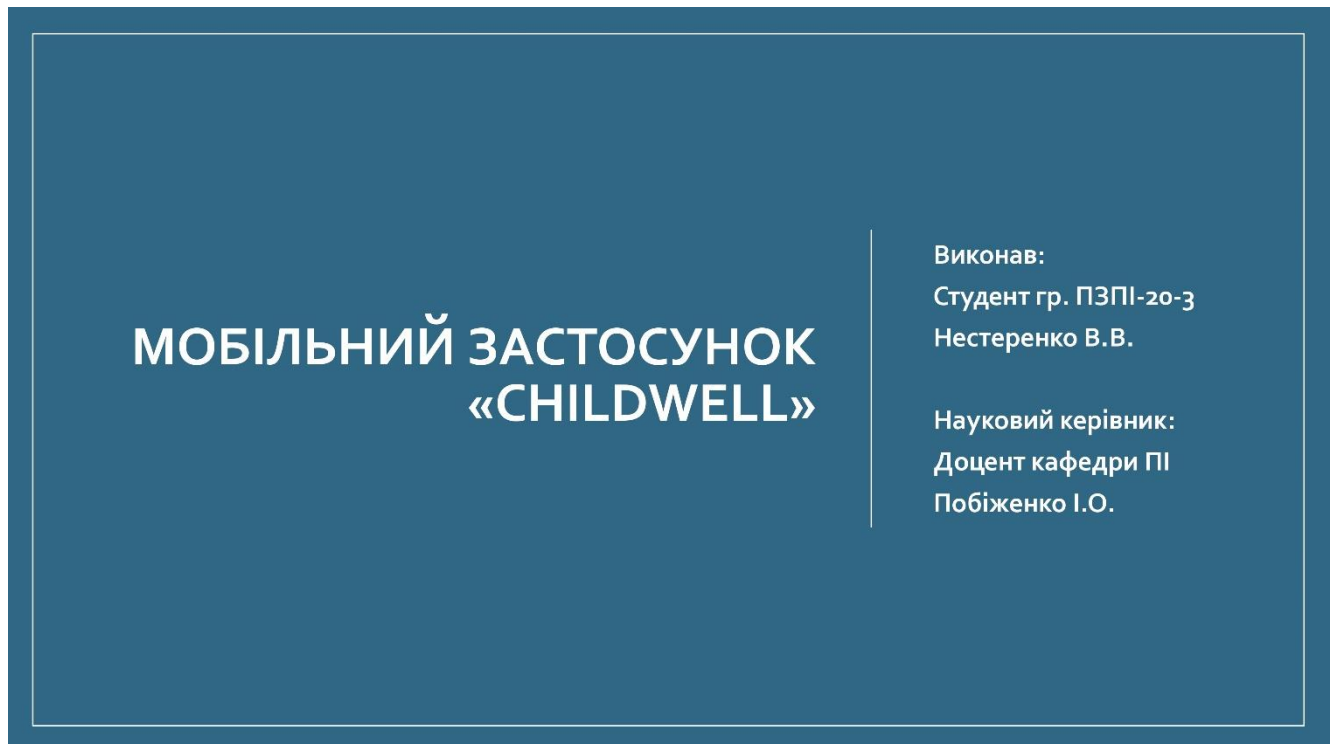


Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

## Вступ

Мобільний застосунок «ChildWell» оптимізує управління дошкільними закладами, надаючи зручні інструменти для створення розкладу занять, управління закладами та відвідуваністю дітей за допомогою QR-кодів



3

Рисунок Б.3 – Слайд 3

## Унікальні функції мобільного застосунку

### 01

Сканування і генерація QR-кодів для безпечної реєстрації відвідуваності дітей

### 02

Оплата рахунків одразу в застосунку monobank

4

Рисунок Б.4 – Слайд 4

## Основний функціонал



5

Рисунок Б.5 – Слайд 5

## Генерація розкладу

The screenshot shows the 'Генерація розкладу' screen with the following sections:

- Сон:** Денний сон (checked)
- Фізична активність:** Зарядка (checked), Прогулянка (checked), Рухливі ігри (checked), Танці (unchecked), Естафетні змагання (unchecked), Спортивні ігри (unchecked)
- Освіта та розвиток:** Математика та логіка (checked), Казки та читання (checked)
- Button: Продовжити

Вибір активностей

The screenshot shows the 'Генерація розкладу' screen displaying a generated schedule for Monday through Friday. The schedule is organized into a grid with columns for days (Пн, Вт, Ср, Чт, Пт) and rows for time slots (08:00-09:00, 09:00, 10:00). Activities are color-coded: green for 'Ігра', orange for 'Математика та логіка', yellow for 'Рухливі ігри', and purple for 'Малювання'.

	Пн	Вт	Ср	Чт	Пт
08:00-09:00	Ігра	Ігра	Ігра	Ігра	Ігра
09:00	Математика та логіка	Математика та логіка	Математика та логіка	Математика та логіка	Математика та логіка
10:00	Казки та читання	Казки та читання	Казки та читання	Казки та читання	Казки та читання
	Малювання	Малювання	Малювання	Малювання	Малювання

Buttons: Перегенерувати, Завершити

Згенерований розклад

6

Рисунок Б.6 – Слайд 6

## QR-коди відвідувань

- Підвищення безпеки передачі дітей від батьків до вихователів і навпаки
- Зменшують ризик помилок через людський фактор



7

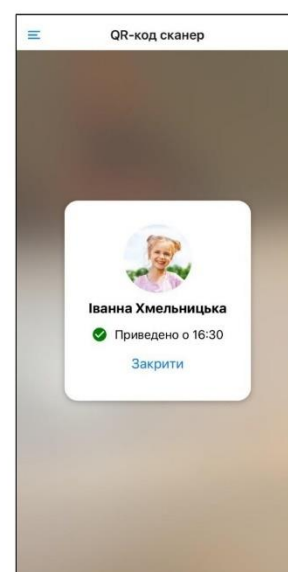
Рисунок Б.7 – Слайд 7



QR-код у батька



Сканер QR-кодів у викладача



Результат сканування

8

Рисунок Б.8 – Слайд 8

## Історія відвідувань

**Історія відвідувань групи (викладач)**

Ім'я	Статус
Борис Хмельницький	Відсутній
Вікторія Мазепа	Відсутній
Олександр Мазепа	Присутній
Катерина Мазепа	Присутній
Андрій Мазепа	Присутній
Олексій Сагайдачний	Відсутній
Марія Сагайдачна	Присутній
Іван Сагайдачний	Присутній
Анна Довженко	Присутній

**Історія відвідувань дитини (батько)**

Підсумок відвідувань за місяць

- Присутній: 10 днів
- Відсутній: 10 днів
- Рівень відвідуваності: 50.0%

9

Рисунок Б.9 – Слайд 9

## Безпека QR-кодів

Дані QR-коду зашифровані і не можуть бути підробленими

Строк дії QR-коду 2 хвилини

10

Рисунок Б.10 – Слайд 10



## Тестування

- Застосунок тестувався з використанням середовища Expo Go на iPhone 11, що працює під управлінням операційної системи iOS версії 17.5.1
- Система успішно відображає згенерований розклад
- Система успішно сканує QR-коди і надає відповідне повідомлення в разі невалідності коду
- Система успішно відображає історію відвідувань
- Було протестовано і всі інші основні функції, тести пройдено успішно

13

Рисунок Б.13 – Слайд 13

## Використані технології



14

Рисунок Б.14 – Слайд 14

## Висновки



Мобільний застосунок «ChildWell» забезпечує зручний доступ до розкладу занять і управлінням садочками



QR-коди у підвищили ефективність реєстрації та управління відвідуваннями дітей, забезпечуючи швидке та точне оновлення даних



Оплата рахунків одразу в застосунку monobank підвищує безпеку і швидкість оплати, зменшуючи ризик помилки через людський фактор

Рисунок Б.15 – Слайд 15

ДОДАТОК В  
Специфікація програмного забезпечення

ChildWell

Software Requirements Specification

4.0

15.02.2024

Богдан Смейко  
Віталій Нестеренко  
Анна Бондаренко

## Історія версій

Дата	Опис	Автор	Коментарі
25.01.2024	Версія 1.0		Перша редакція
05.02.2024	Версія 2.0	Анна Бондаренко	Перелік функцій продукту розміщено в порядку пріоритетності Проекту надано назву
10.02.2024	Версія 3.0	Богдан Смейко	Оновлено класи
15.02.2024	Версія 4.0	Віталій Нестеренко	Оновлено діаграми

## Затвердження документів

Наступну Специфікацію вимог до програмного забезпечення було прийнято та схвалено:

Підпис	Ім'я	Роль	Дата
	Богдан Смейко	Back-end частина	15.02.2024
	Віталій Нестеренко	Mobile частина	15.02.2024
	Анна Бондаренко	Front-end частина	15.02.2024

## Зміст

1. Вступ.....	6
1.1 Мета .....	6
1.2 Межі.....	7
1.3 Означення та аббревіатури .....	7
1.4 Посилання .....	8
1.5 Огляд.....	9
2. Загальний опис .....	10
2.1 Перспективи продукту .....	10
2.2 Функції продукту.....	10
2.3 Характеристики користувачів .....	11
2.4 Загальні обмеження.....	11
2.5 Припущення й залежності .....	12
3. Конкретні вимоги .....	13
3.1 Вимоги до зовнішніх інтерфейсів .....	13
3.1.1 Інтерфейс користувача .....	13
3.1.2 Апаратний інтерфейс .....	13
3.1.3 Програмний інтерфейс .....	13
3.1.4 Комунікаційні інтерфейси .....	13
3.2 Функціональні вимоги .....	14
3.2.1 Реєстрація та авторизація користувачів .....	14
3.2.2 Генерація розкладу для дітей, з урахуванням фізіологічних особливостей.....	15
3.2.3 Зміна викладачами режимів дня та активностей.....	15
3.2.4 Перегляд журналу відвідування групи .....	16
3.2.5 Контроль батьків над процедурою передачі дітей в довірені руки ...	16
3.2.6 Автоматичне сповіщення вихователів про відсутність дитини .....	17
3.2.7 Додавання дитини в групу .....	17
3.2.8 Перегляд інформації про режим дня та активності дитини .....	18
3.2.9 Оплата рахунків для батьків.....	19

3.2.10	Подача заявки до закладу освіти .....	19
3.2.11	Керування заявкою до закладу освіти .....	20
3.2.12	Оновлення налаштувань активностей закладу освіти .....	20
3.3	Варіанти використання (Use Cases) .....	22
3.3.1	Неавтентифікований варіант використання .....	22
3.3.2	Варіант використання батьків .....	23
3.3.2	Варіант використання вихователів .....	24
3.3.3	Варіант використання адміністраторів садочків .....	25
3.4	Класи / Об'єкти .....	26
3.4.1	Group .....	26
3.4.2	Child .....	26
3.4.3	Preschool .....	26
3.4.4	Attendance .....	27
3.4.5	Application .....	27
3.4.6	Invoice .....	28
3.4.7	ScheduleActivity .....	28
3.5	Нефункціональні вимоги .....	29
3.5.1	Вимоги до продуктивності .....	29
3.5.2	Вимоги до надійності .....	29
3.5.3	Вимоги до доступності .....	29
3.5.4	Вимоги до безпеки .....	30
3.5.5	Вимоги до ремонтпридатності .....	30
3.5.6	Вимоги до портативності .....	30
3.6	Обмеження та винятки (Inverse Requirements) .....	31
3.7	Архітектурні обмеження .....	31
3.8	Вимоги до збереження даних (Logical Database Requirements) .....	32
3.9	Інші вимоги .....	32
4.	Моделі аналізу .....	33
4.1	Діаграма послідовності .....	33
4.2	Діаграма переходу станів (STD) .....	35
4.3	Діаграма потоку даних (DFD) .....	37

5. Процес управління змінами.....37

Рисунок В.5 – Сторінка 5

## 1. Вступ

У сучасному світі розвиток дитини визначається не лише академічними досягненнями, проте й фізичним та ментальним здоров'ям, емоційним станом, обсягом уваги. Тож виникає виклик у створенні та впровадженні ефективного розпорядку дня в закладах дошкільної освіти, де закладається фундамент для розвитку подальших здібностей дитини та формування їхніх звичок, що безумовно має вплив на все подальше життя вихованців. Важко сумістити бажання вихователів забезпечити оптимальний розвиток дітей, уникнути виснаження та гарантування засвоєння необхідних навичок і знань та потребу батьків у своєчасному доступі до інформацію про активності дітей в дитячих садках.

Виходячи з цього, постає актуальна потреба в розробці системи, яка не лише допоможе в розвитку навичок, а дозволить планувати розпорядок дня так, щоб діти не виснажувалися й забезпечить батькам надійний та миттєвий доступ до інформації щодо режиму дня та харчування їхніх дітей. Наша програма має на меті оптимізацію описаних вище процесів, що відповідає як вимогам педагогів, так і батьків. Детальніше про проєкт розповідається в даному документі.

### 1.1 Мета

Мета документу — надати детальний опис розроблюваної програмної системи для закладів дошкільної освіти з керуванням режимом дня та харчуванням дітей. Специфікація призначена для опису функціональних та нефункціональних вимог першої версії програмного продукту. SRS використовуватиметься членами команди, які будуть реалізовувати та забезпечувати коректну роботу системи та іншими стейкхолдерами.

## 1.2 Межі

Система, яку ми розробляємо отримала назву «ChildWell». Вона надаватиме можливість для викладачів створювати ефективний розклад для вихованців з урахуванням їх фізіологічних особливостей, який гарантує здобуття потрібних навичок. Для батьків — можливість отримувати інформацію про активності дитини в закладі дошкільної освіти та її харчування в режимі реального часу.

Корисним функціоналом для вихователів стане планувальних розпорядку дня дітей, щоб вони не виснажувались та здобували потрібні навички. Батьки ж в свою чергу зможуть швидко отримувати достовірну інформацію щодо режиму дня дітей та їхнього харчування. Більш того корисним функціоналом стане можливість відмічання присутності дітей за допомогою QR-кодів, що буде доступним як для батьків так і вихователів. Крім цього батьки зможуть управляти заявками на подання дітей до закладів дошкільної освіти, а після матимуть змогу зручно сплачувати рахунки за послуги закладу.

Таким чином, система сприятиме ефективній організації розвитку дітей, забезпечуючи комфорт та контроль як для вихователів й адміністраторів закладу, так і для батьків.

## 1.3 Означення та аббревіатури

Front-end — частина програмного забезпечення, яка відповідає за взаємодію користувача з системою або веб-додатком (дизайн, інтерфейс користувача та усі елементи, які користувач може бачити та з якими він може взаємодіяти).

int — дані цілочисельного типу.

text — дані текстового типу.

`bool` — це тип даних, який має одне з двох можливих значень (істинне і хибне), які призначені для представлення двох істинних значень логіки та булевої алгебри.

`date` — тип даних для зберігання інформації про конкретний день у форматі рік-місяць-день.

`datetime` — тип даних, що включає в себе інформацію про конкретний момент у часі, включаючи як дату, так і час.

`Git` — це розподілена система контролю версій, яка відстежує зміни в будь-якому наборі комп'ютерних файлів.

`ISO/IEC 27001` — це міжнародний стандарт для управління інформаційною безпекою.

`WCAG` (Інструкції щодо доступності веб-вмісту) — частина серії рекомендацій щодо доступності веб-сайтів, опублікованих Ініціативою веб-доступності (WAI) Консорціуму Всесвітньої павутини (W3C), головною міжнародною організацією стандартів для Інтернету. Вони являють собою набір рекомендацій щодо того, як зробити веб-вміст більш доступним.

`GDPR` (Загальний регламент захисту даних) — це нормативний акт Європейського Союзу щодо конфіденційності інформації в Європейському Союзі та Європейській економічній зоні.

#### **1.4 Посилання**

- ГОСТ 34.602-89: стандарти для автоматизованих систем, які регулюють вимоги до проектування та розробки.
- Методичні рекомендації МОН України: вказівки Міністерства освіти і науки України щодо організації дошкільної освіти дітей.
- ISO/IEC 27001: стандарт, що встановлює вимоги до систем управління інформаційною безпекою, який допоможе забезпечити безпеку даних користувачів.

- WCAG (Web Content Accessibility Guidelines): рекомендації щодо доступності веб-контенту, що допоможуть зробити додаток доступним для всіх користувачів, включаючи людей з обмеженими можливостями.
- GDPR (General Data Protection Regulation): регламент ЄС щодо захисту персональних даних.

## 1.5 Огляд

Цей документ містить повну інформацію про проєкт: його перспективи, функції, інтерфейс, обмеження, відомості про варіанти використання та діаграми для візуалізації різних аспектів системи. Кожна секція призначена для різних стейкхолдерів, від чого залежить зміст секцій.

Розділ «Загальний опис» націлений на опис загальних факторів, що впливають на вимоги до продукту і продукт в цілому. Цільовою аудиторією цього розділу перш за все є власник проєкту.

Розділ «Конкретні вимоги», на відміну від попереднього, містить подробиці щодо вимог до проєкту, його інтерфейси та функціональність, у ньому деталі реалізації описані технічною мовою для сприйняття розробниками.

У розділі «Моделі аналізу» розміщено діаграми та моделі, що використовуватимуться в процесі розробки відповідно до вимог, визначених в цьому документі SRS. Вони допоможуть у розумінні системи, шляхом візуалізації її роботи та взаємодії, що відбувається всередині системи.

Останній розділ, «Процес управління змінами», передбачує та описує яким чином відбуватиметься оновлення SRS, якщо виникне така необхідність при внесенні змін до вимог чи обсягів проєкту. Також він має вказувати на те, ким вноситимуться та ухвалюватимуться зміни.

## 2. Загальний опис

### 2.1 Перспективи продукту

Система планується як самостійний продукт, що не залежить від схожих застосунків. Для повноцінного використання користувачі повинні зареєструватися та використовувати браузерну або мобільну версію додатку.

### 2.2 Функції продукту

Основною метою продукту є оптимізація планування розпорядку дня дітей у закладах дошкільної освіти з управлінням харчуванням. Викладачам треба мати змогу керувати режимом дня дітей та отримувати інформацію про те, чи є він відповідним до вікових особливостей вихованців. Батьки ж у свою чергу отримають інструмент для відстеження активностей дитини та комунікації з працівниками закладу дошкільної освіти. До основних функцій продукту належать наступні:

- MF-1: Реєстрація та авторизація користувачів;
- MF-2: Створення оптимального розкладу для дітей, з урахуванням фізіологічних особливостей;
- MF-3: Зміна вихователями розкладу;
- MF-4: Контроль батьків над процедурою передачі дітей в довірені руки;
- MF-5: Автоматичне сповіщення вихователів про відсутність дитини;
- MF-6: Електронний журнал відвідуваності з часовими рамками;
- MF-7: Перегляд докладної інформації про режим дня та активності дитини;
- MF-8: Оплата рахунків (батьками);
- MF-9: Подача заявок до закладу освіти;
- MF-10: Керування заявками до закладу освіти;

- MF-11: Налаштування завідувачем (адміністратором) певних аспектів закладу освіти.

Також система повинна надавати інформацію про розробників та контакти.

### **2.3 Характеристики користувачів**

Передбачено 3 види користувачів, що матимуть доступ до системи:

- батьки дітей дошкільного віку (подання та управління заявками до дитячих садків, перегляд режиму дня, сплата рахунків);
- вихователі закладів дошкільної освіти (створення режиму дня, відмічання присутності дітей);
- адміністративний персонал закладу (управління заявками від батьків, управління та налаштування дитячого садка, створення занять, реєстрація вихователів).

### **2.4 Загальні обмеження**

Система повинна працювати на пристроях з наступними характеристиками:

- мінімум 4 ГБ оперативної пам'яті;
- процесор не нижче як чотирьохядерний, з тактовою частотою не менше 2.0 ГГц.

Система має забезпечувати сумісність з операційними системами:

- Android 6.0 або новіше;
- iOS 13.4 або новіше.

Сучасні браузері підтримують виконання JavaScript, тому front-end частина буде виконана з використанням фреймворку React.js. Система

повинна бути готовою до впровадження не пізніше 8 травня 2024 року. Система повинна відповідати стандартам безпеки даних, включаючи шифрування особистої інформації та заходи забезпечення конфіденційності.

## 2.5 Припущення й залежності

- Припущення: Користувачі мають доступ до Інтернету і він достатньо якісний;
  - Залежність: За відсутності доступу до Інтернету, користувачі не можуть використовувати застосунок у повній мірі;
- Припущення: Користувачі зареєстровані в системі;
  - Залежність: Незареєстровані користувачі не можуть користуватися системою в повному обсязі;
- Припущення: Всі користувачі мають апаратні та програмні засоби, необхідні для коректної роботи системи, зокрема, підтримують необхідні версії браузерів та операційних систем;
  - Залежність: Використання браузерів або пристроїв, що не підтримуються системою, може призвести до нестабільної роботи або відмови в доступі до певних функцій;
- Припущення: Всі користувачі володіють основами мови, в якій реалізовано інтерфейси користувача;
  - Залежність: Коректне користування системою передбачає знання користувачами мови, в якій вона реалізована.

### **3. Конкретні вимоги**

#### **3.1 Вимоги до зовнішніх інтерфейсів**

##### **3.1.1 Інтерфейс користувача**

Зареєстровані користувачі поділятимуться на вихователів та батьків й матимуть трохи різні інтерфейси.

Ці обидва типи користувачів матимуть доступ до режимів дня та планів харчування дітей, але лише вихователі зможуть їх редагувати. Також батьки матимуть окрему сторінку з дошкою оголошень, на якій бачитимуть повідомлення від дитсадків та вихователів.

##### **3.1.2 Апаратний інтерфейс**

Для відображення інтерфейсу користувача до пристрою має бути підключений дисплей.

##### **3.1.3 Програмний інтерфейс**

Для доступу до веб-системи користувач повинен використовувати веб-браузер, який повинен підтримувати Javascript, HTML5, CSS3. Для використання мобільного додатку, користувачеві необхідно мати пристрій з ОС Android 6.0+ або iOS 13.4+.

##### **3.1.4 Комунікаційні інтерфейси**

Щоб користувач мав доступ до веб-системи, необхідна наявність Інтернету.

## **3.2 Функціональні вимоги**

### **3.2.1 Реєстрація та авторизація користувачів**

#### **3.2.1.1 Опис**

Нові користувачі можуть створювати акаунти в системі реєструватися за допомогою Google акаунта та використовувати їх для входу.

#### **3.2.1.2 Вхідні умови**

Заповнення усіх обов'язкових полів для реєстрації або входу в систему.

#### **3.2.1.3 Виконання**

Сервер створює нового користувача при реєстрації. При вході в систему сервер перевіряє чи існує такий користувач, якщо ні, то виводить відповідне повідомлення.

#### **3.2.1.4 Вихідні умови**

Створений новий користувач.

#### **3.2.1.5 Обробка помилок**

Якщо введено неправильний пароль або логін, буде показане відповідне повідомлення.

### **3.2.2 Генерація розкладу для дітей, з урахуванням фізіологічних особливостей**

#### 3.2.2.1 Опис

Система в автоматичному режимі може створити розклад для дітей, враховуючи розподіл часу між іграми, навчанням, прогулянками та прийомами їжі.

#### 3.2.2.2 Вхідні умови

Задано початкові значення та пріоритети (обрано активності з наданих категорій).

#### 3.2.2.3 Виконання

Сервер виконує обробку, розташовуючи активності в потрібному та правильному порядку.

#### 3.2.2.4 Вихідні умови

Створений розклад дня для дітей.

#### 3.2.2.5 Обробка помилок

В разі помилки система запропонує змінити пріоритети в розкладі, так як з наявними неможливо створити оптимальний розклад.

### **3.2.3 Зміна викладачами режимів дня та активностей**

#### 3.2.3.1 Опис

Викладачі можуть корегувати розклад дня дітей, що був згенерований системою.

#### 3.2.3.2 Вхідні умови

Користувач повинен мати роль викладача.

#### 3.2.3.3 Виконання

Сервер оновлює існуючий розклад.

#### 3.2.3.4 Вихідні умови

Змінений розклад дня.

#### 3.2.3.5 Обробка помилок

Якщо користувач не викладач – заборонити зміну розкладу.

### **3.2.4 Перегляд журналу відвідування групи**

#### 3.2.4.1 Опис

Вихователі повинні мати можливість переглядати журнал відвідуваності групи дітьми за обраний день.

#### 3.2.4.2 Вхідні умови

Обрано групу та день, за який присутня інформація про відвідуваність групи дітьми.

#### 3.2.4.3 Виконання

Сервер обробляє запит на надає відповідні дані у відповідь.

#### 3.2.4.4 Вихідні умови

Журнал відвідуваності групи за обраний день успішно відображено на екрані вихователя.

#### 3.2.4.5 Обробка помилок

В разі помилки система виводить повідомлення про відсутність запитуваних даних.

### **3.2.5 Контроль батьків над процедурою передачі дітей в довірені руки**

#### 3.2.5.1 Опис

Можливість батьків контролювати процедури передачі своєї дитини в довірені руки.

#### 3.2.5.2 Вхідні умови

Батьки обирають дитину, для якої треба згенерувати check-in/check-out QR-код.

#### 3.2.5.3 Виконання

Сервер генерує QR-код, що містить всю необхідну інформацію для передачі дитини в довірені руки.

#### 3.2.5.4 Вихідні умови

Надано дані, які система буде використовувати при здійсненні цієї процедури.

#### 3.2.5.5 Обробка помилок

Система надає повідомлення про будь-які помилки.

### **3.2.6 Автоматичне сповіщення вихователів про відсутність дитини**

#### 3.2.6.1 Опис

Система автоматично повідомляє вихователів про відсутність дитини, відображаючи відповідну інформацію в журналі присутності.

#### 3.2.6.2 Вхідні умови

Система фіксує інформацію про відсутність дитини, якщо її присутність не зареєстровано за допомогою QR-код.

#### 3.2.6.3 Виконання

Сервер автоматично збирає необхідну інформацію та надає її вихователям.

#### 3.2.6.4 Вихідні умови

Вихователі отримують інформацію про відсутність дитини в закладі дошкільної освіти.

#### 3.2.6.5 Обробка помилок

Система виявляє можливі конфлікти чи невідповідності в інформації і повідомляє вихователів для вирішення ситуації.

### **3.2.7 Додавання дитини в групу**

#### 3.2.7.1 Опис

Керівники ЗДО матимуть можливість додати дитину в групу, обравши одну з-поміж переліку груп садочка, відповідних до віку дитини.

#### 3.2.7.2 Вхідні умови

Для додавання дитини в групу користувач повинен мати роль керівника ЗДО. Дитина як і група повинні бути обрані для коректного здійснення операції.

#### 3.2.7.3 Виконання

Сервер додає нову дитину до групи та зберігає зміни.

#### 3.2.7.4 Вихідні умови

Батьки та керівник бачать оновлену інформацію .

#### 3.2.7.5 Обробка помилок

При відсутності необхідної ролі – повідомити користувача про це.

### **3.2.8 Перегляд інформації про режим дня та активності дитини**

#### 3.2.8.1 Опис

Батьки та вихователі зможуть переглядати детальну інформацію про режим дня та активності дітей.

#### 3.2.8.2 Вхідні умови

На відповідній сторінці календар дозволить переглядати розклад і одному з режимів (місяць, тиждень, день, табличний вигляд).

#### 3.2.8.3 Виконання

Сервер збирає та надсилає інформацію про дитину.

#### 3.2.8.4 Вихідні умови

Батьки або вихователі бачать інформацію про режим дня та активності дитини.

#### 3.2.8.5 Обробка помилок

При помилці сервер спробує зібрати інформацію кілька разів, після чого відобразить повідомлення про неуспішність операції.

### **3.2.9 Оплата рахунків для батьків**

#### 3.2.9.1 Опис

Функція дозволяє батькам здійснювати оплату рахунків за утримання та надані послуги в дошкільному закладі.

#### 3.2.9.2 Вхідні умови

Батькам доступно отримання деталізації рахунків.

#### 3.2.9.3 Виконання

Система формує рахунки для батьків, відображаючи вартість та послуги, надані дитині.

#### 3.2.9.4 Вихідні умови

Батьки мають можливість вчасно та зручно оплачувати рахунки.

#### 3.2.9.5 Обробка помилок

Система надає повідомлення про будь-які помилки та рекомендації для виправлення ситуації.

### **3.2.10 Подача заявки до закладу освіти**

#### 3.2.10.1 Опис

Батьки матимуть можливість подати заявку на вступ до закладу дошкільної освіти.

#### 3.2.10.2 Вхідні умови

Обрано дитину, на ім'я якої й буде створено заявку, набір у освітній заклад відкрито.

#### 3.2.10.3 Виконання

Сервер обробляє подану заявку, зберігає її та надає інформацію про неї керівнику ЗДО.

#### 3.2.10.4 Вихідні умови

Можливість переглянути заявки на відповідній сторінці.

#### 3.2.10.5 Обробка помилок

При помилці сервер спробує відправити заявку кілька разів, після чого відобразить повідомлення про неуспішність відправлення.

### 3.2.11 Керування заявкою до закладу освіти

#### 3.2.11.1 Опис

Батьки матимуть можливість скасувати, прийняти або відхилити заявку. Керівники ЗДО матимуть можливість для перегляду деталей заявки, прийому або відхилення чи коментування заявки.

#### 3.2.11.2 Вхідні умови

Обрано заявку, керування якою треба здійснити.

#### 3.2.11.3 Виконання

Сервер обробляє оновлену заявку, зберігає її.

#### 3.2.11.4 Вихідні умови

Можливість переглянути заявки на відповідній сторінці та оновити за допомогою відповідних кнопок.

#### 3.2.11.5 Обробка помилок

При помилці сервер спробує оновити заявку кілька разів, після чого відобразить повідомлення про неуспішність відправлення.

### 3.2.12 Оновлення налаштувань активностей закладу освіти

#### 3.2.12.1 Опис

Керівники ЗДО можливість оновити налаштування активностей ЗДО.

#### 3.2.12.2 Вхідні умови

Обрано ЗДО та активності.

#### 3.2.12.3 Виконання

Сервер обробляє оновлені налаштування, зберігає їх.

ChildWell

#### 3.2.12.4 Вихідні умови

Можливість переглянути налаштування на відповідній сторінці та оновити за допомогою відповідних кнопок.

#### 3.2.12.5 Обробка помилок

При помилці сервер спробує оновити налаштування кілька разів, після чого відобразить повідомлення про неуспішність відправлення.

### 3.3 Варіанти використання (Use Cases)

#### 3.3.1 Неавтентифікований варіант використання

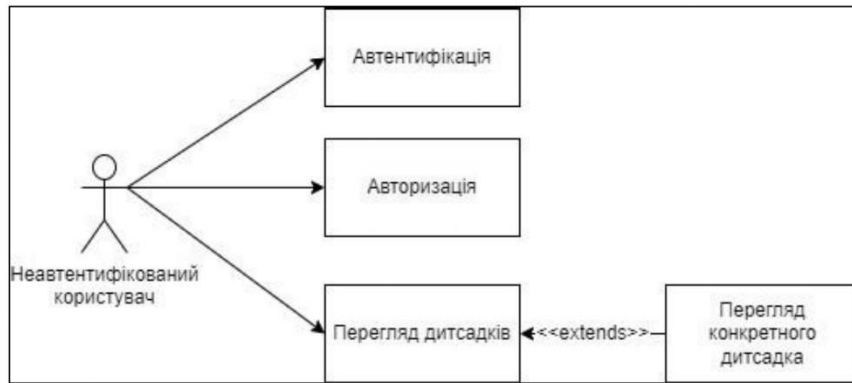


Рисунок 3.1 – Неавтентифікований варіант використання

### 3.3.2 Варіант використання батьків

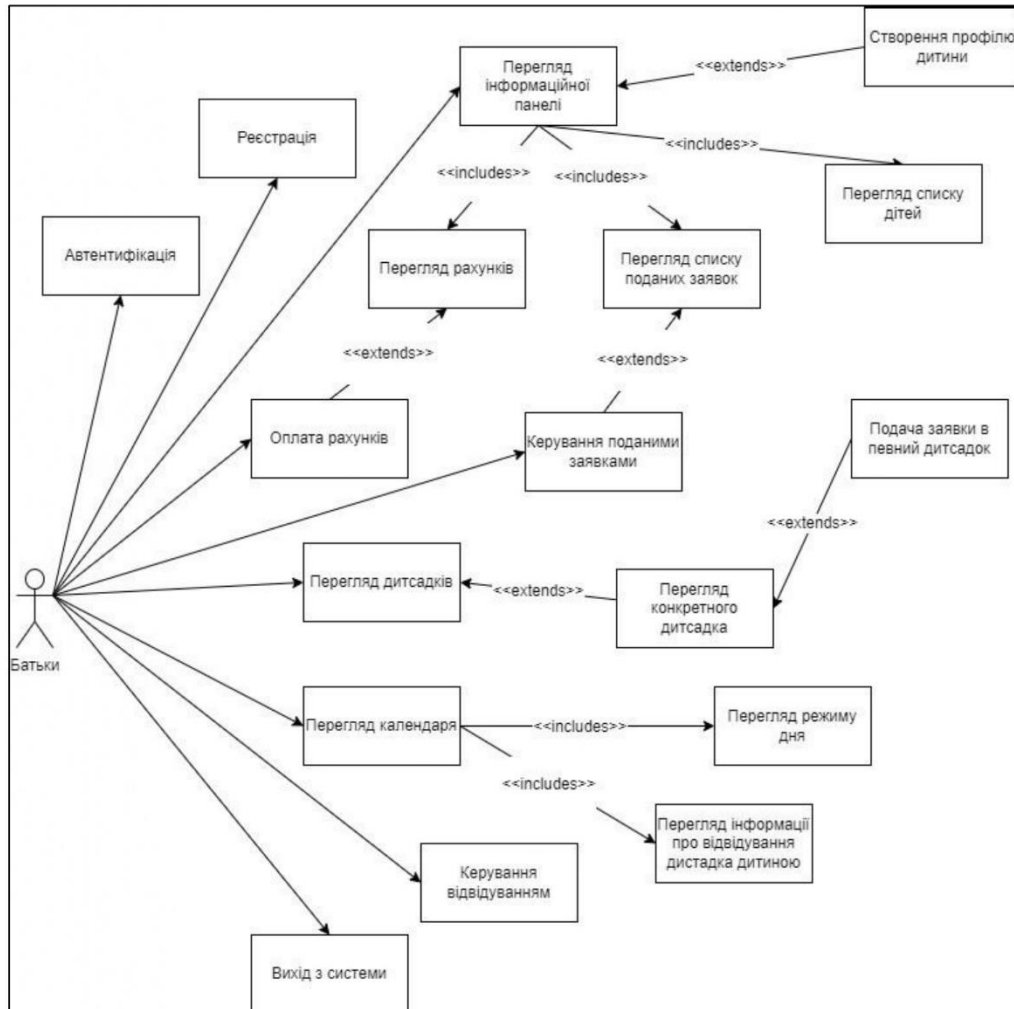


Рисунок 3.2 – Варіант використання батьків

### 3.3.2 Варіант використання вихователів



Рисунок 3.3 – Варіант використання вихователів

### 3.3.3 Варіант використання адміністраторів садочків

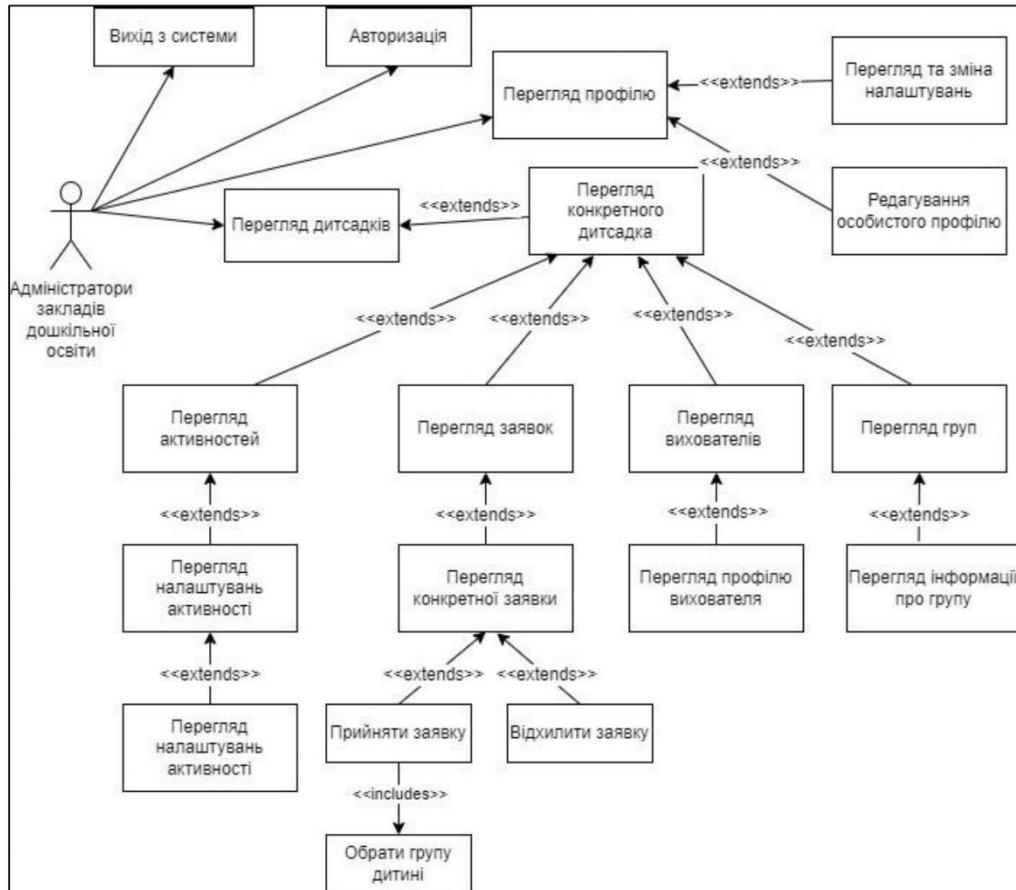


Рисунок 3.4 – Варіант використання адміністраторів садочків

### **3.4 Класи / Об'єкти**

#### **3.4.1 Group**

##### 3.4.1.1 Атрибути

Id – text

PreschoolId – text

Name – text

Description – text

MinAge – int

MaxAge – int

StartDateTime – datetime

EndDateTime – datetime

#### **3.4.2 Child**

##### 3.4.2.1 Атрибути

Id – text

ParentId – text

GroupId – text

FirstName – text

LastName – text

BirthDate - date

Gender – text

Weight – float

#### **3.4.3 Preschool**

##### 3.4.3.1 Атрибути

Id – text  
Name – text  
Description – text  
ContactNumber – text  
Email – text  
City – text  
Country – text  
Address – text  
Website – text  
Facilities – text  
OpeningHours – datetime  
ClosingHours – datetime  
OpenRegistration – bool  
PricePerDayInUah – decimal

#### **3.4.4 Attendance**

##### 3.4.4.1 Атрибути

Id – text  
ChildId – text  
Date – date  
DropOffTime – datetime  
PickUpTime – datetime

#### **3.4.5 Application**

##### 3.4.5.1 Атрибути

Id – text  
ChildId – text  
PreschoolId – text

ChildWell

GroupId – text

Status – int

Comment – text

### **3.4.6 Invoice**

#### 3.4.6.1 Атрибути

Id – text

ChildId – text

ExternalId – text

Status – int

DateCreated – datetime

DueDate – datetime

Period – date

PayUrl – text

### **3.4.7 ScheduleActivity**

#### 3.4.7.1 Атрибути

Id – text

ActivityId – text

GroupId – text

StartDateTime – datetime

End DateTime – datetime

### **3.5 Нефункціональні вимоги**

#### **3.5.1 Вимоги до продуктивності**

1. Веб-додаток і мобільний клієнт мають відображати вміст у відповідний спосіб, забезпечуючи оптимальну взаємодію з користувачем на різних пристроях і розмірах екрана.

2. Мобільні сторінки мають завантажуватися протягом 5 секунд, зберігаючи стабільну продуктивність веб-версії.

#### **3.5.2 Вимоги до надійності**

1. Впроваджувати надійні механізми обробки помилок, щоб ефективно обробляти несподівані помилки та запобігати збоям системи.

2. Вести докладні журнали помилок, включаючи відповідну інформацію для налагодження та аналізу.

3. Забезпечити цілісність транзакцій, особливо в критичних операціях, шляхом впровадження таких механізмів, як транзакції бази даних.

4. Спроектувати систему для плавного відновлення після збою, мінімізуючи вплив на користувачів.

#### **3.5.3 Вимоги до доступності**

1. Система має прагнути до мінімум 99,9% часу безвідмовної роботи, враховуючи вікна планового технічного обслуговування.

2. Програма має бути розроблена з механізмами відмовостійкості для обробки несподіваних збоїв у апаратному забезпеченні, програмному забезпеченні або мережевих компонентах.

3. Регулярно виконувати автоматичне резервне копіювання даних програми, включаючи бази даних і конфігураційні файли.

### **3.5.4 Вимоги до безпеки**

1. Система повинна реалізовувати безпечні механізми автентифікації як для веб-програми, так і для мобільного клієнта.
2. Облікові дані користувача мають надійно зберігатися за допомогою галузевих стандартних методів шифрування.
3. Необхідно запровадити контроль доступу на основі ролей, щоб гарантувати, що користувачі мають доступ лише до ресурсів і функцій, які відповідають їхнім ролям.
4. Усі конфіденційні дані, включно з обліковими даними користувача, мають передаватися захищеними каналами з використанням протоколів шифрування, таких як HTTPS.
5. База даних програми повинна реалізовувати шифрування для конфіденційної інформації в стані спокою.

### **3.5.5 Вимоги до ремонтпридатності**

1. Розробити систему з модульною архітектурою, що дозволяє незалежну розробку та підтримку компонентів.
2. Чітко визначені інтерфейси та залежності між модулями для полегшення майбутніх оновлень.
3. Використовуйте систему контролю версій (наприклад, Git) для відстеження змін у вихідному коді, забезпечуючи співпрацю між командами розробників і забезпечуючи історію змін коду.

### **3.5.6 Вимоги до портативності**

1. Веб-програма сумісна з основними веб-браузерами, включаючи, але не обмежуючись ними, Google Chrome, Mozilla Firefox, Microsoft Edge і Safari.
2. Програма працює узгоджено в різних браузерах і версіях.

3. Мобільний клієнт сумісний з основними мобільними операційними системами, такими як iOS і Android.

### **3.6 Обмеження та винятки (Inverse Requirements)**

1. Веб-програма може не працювати оптимально в застарілих або непідтримуваних веб-браузерах.

2. Сумісність не гарантується для операційних систем, відмінних від Windows, macOS і Linux.

3. Мобільний клієнт може не підтримуватися на пристроях із застарілою операційною системою або обмеженими апаратними можливостями.

4. Сумісність обмежена платформами iOS і Android, інші мобільні операційні системи не підтримуються.

5. Хоча мобільний клієнт підтримує основні функції в автономному режимі, певні функції можуть бути обмежені або взагалі не працювати без підключення до Інтернету.

### **3.7 Архітектурні обмеження**

Система має відповідати відповідним галузевим стандартам і вказівкам, таким як стандарти безпеки (наприклад, ISO/IEC 27001) і стандарти веб-доступності (наприклад, WCAG), щоб забезпечити відповідність юридичним і нормативним вимогам.

Система має відповідати положенням про захист даних, таким як Загальний регламент захисту даних (GDPR), забезпечуючи безпечну обробку та обробку даних користувачів.

### 3.8 Вимоги до збереження даних (Logical Database Requirements)

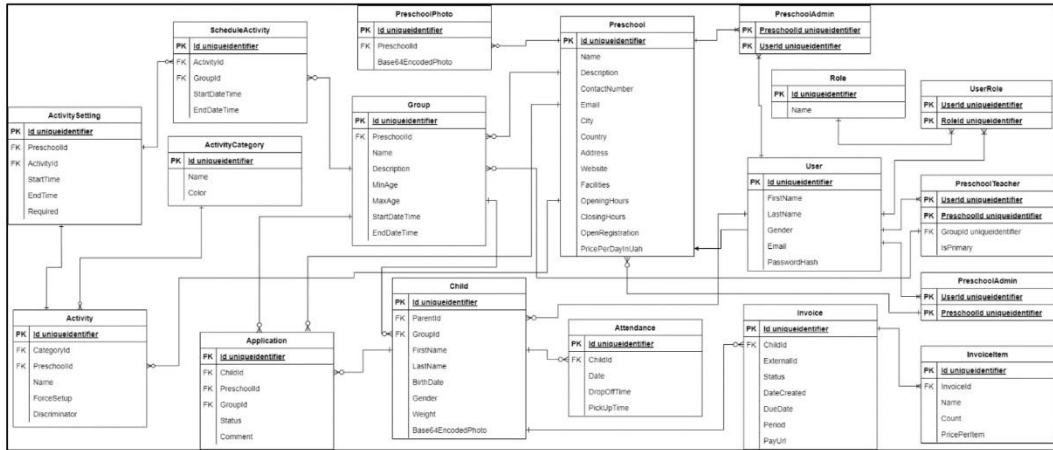


Рисунок 3.5 – ER-діаграма

### 3.9 Інші вимоги

Інших вимог до продукту немає.

## 4. Моделі аналізу

### 4.1 Діаграма послідовності

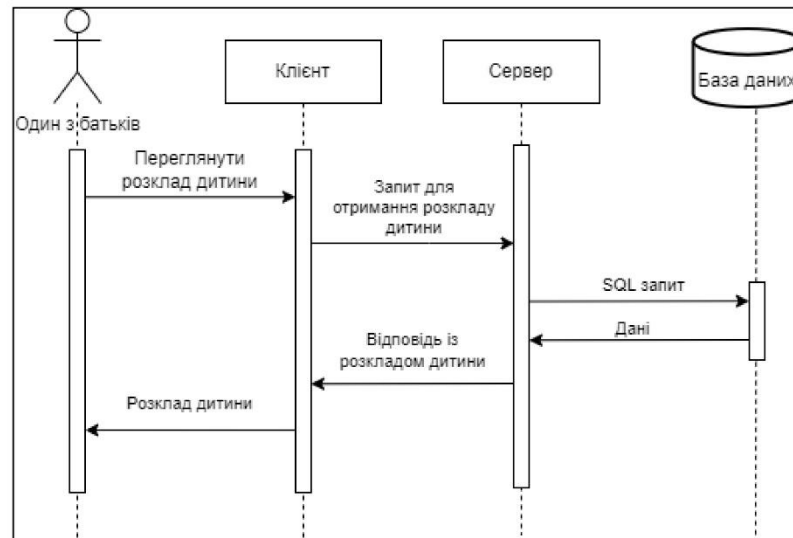


Рисунок 4.1 – Діаграма послідовності перегляду розкладу дитини

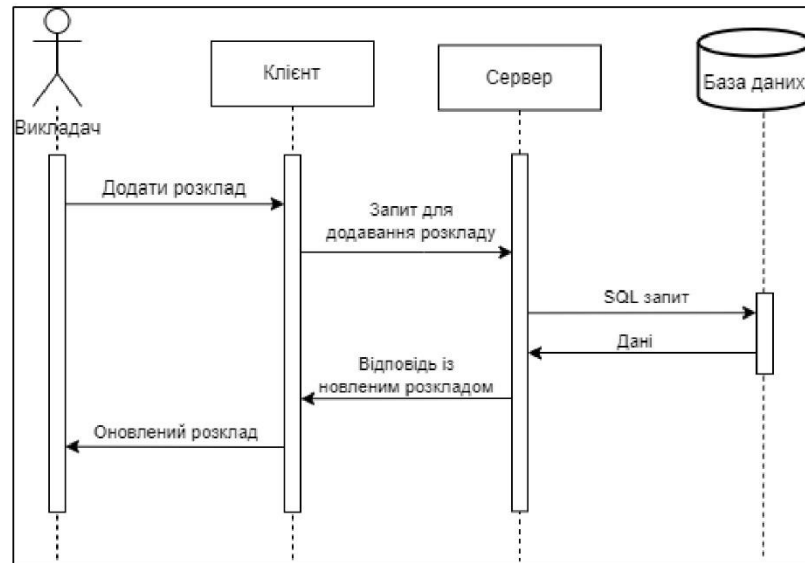


Рисунок 4.2 – Діаграма послідовності додавання розкладу викладачем

## 4.2 Діаграма переходу станів (STD)

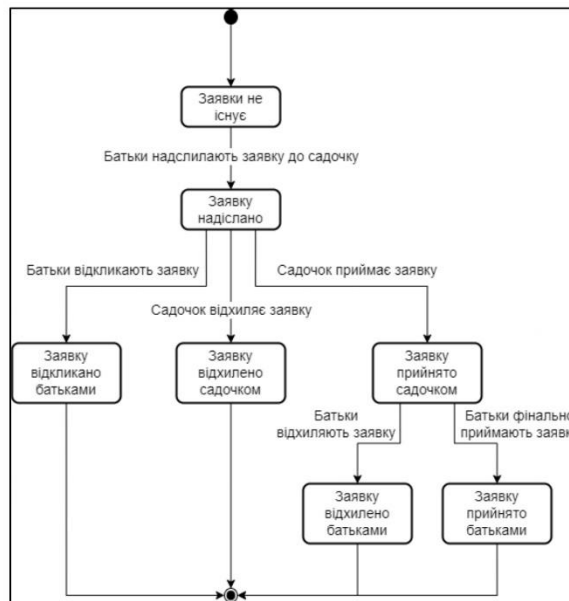


Рисунок 4.3 – Діаграма переходу станів заявки до садочку



Рисунок 4.4 – Діаграма переходу станів відвідуваності дитини

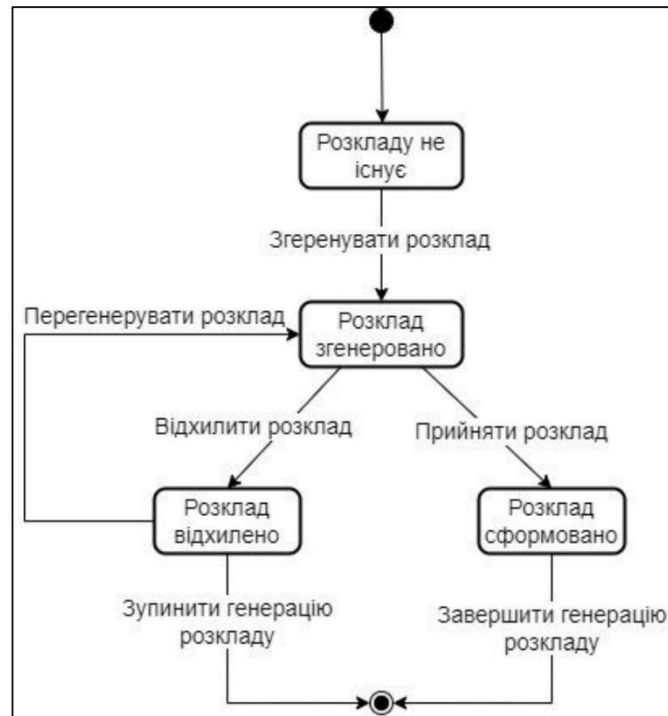


Рисунок 4.5 – Діаграма переходу станів генерації розкладу

### 4.3 Діаграма потоку даних (DFD)

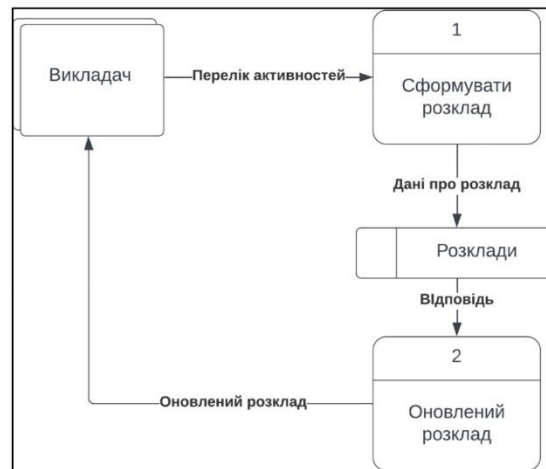


Рисунок 4.6 – Діаграма потоку даних формування розкладу викладачем

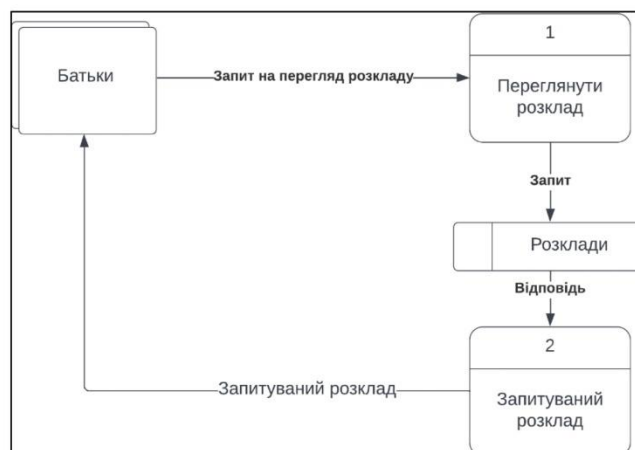


Рисунок 4.7 – Діаграма потоку даних перегляду розкладу батьками

## 5. Процес управління змінами

Зміни до SRS можуть бути запропоновані будь-яким членом команди, це може бути викликане змінами вимог до системи або обсягу робіт. Наступним

ChildWell

кроком повинно бути проведено обговорення щодо змін, після чого визначений член команди вносить схвалені зміни в документ. Оновлена версія має бути внесена в таблицю та підписана кожним членом команди.

## ДОДАТОК Г

## ER-діаграма застосунку «ChildWell»

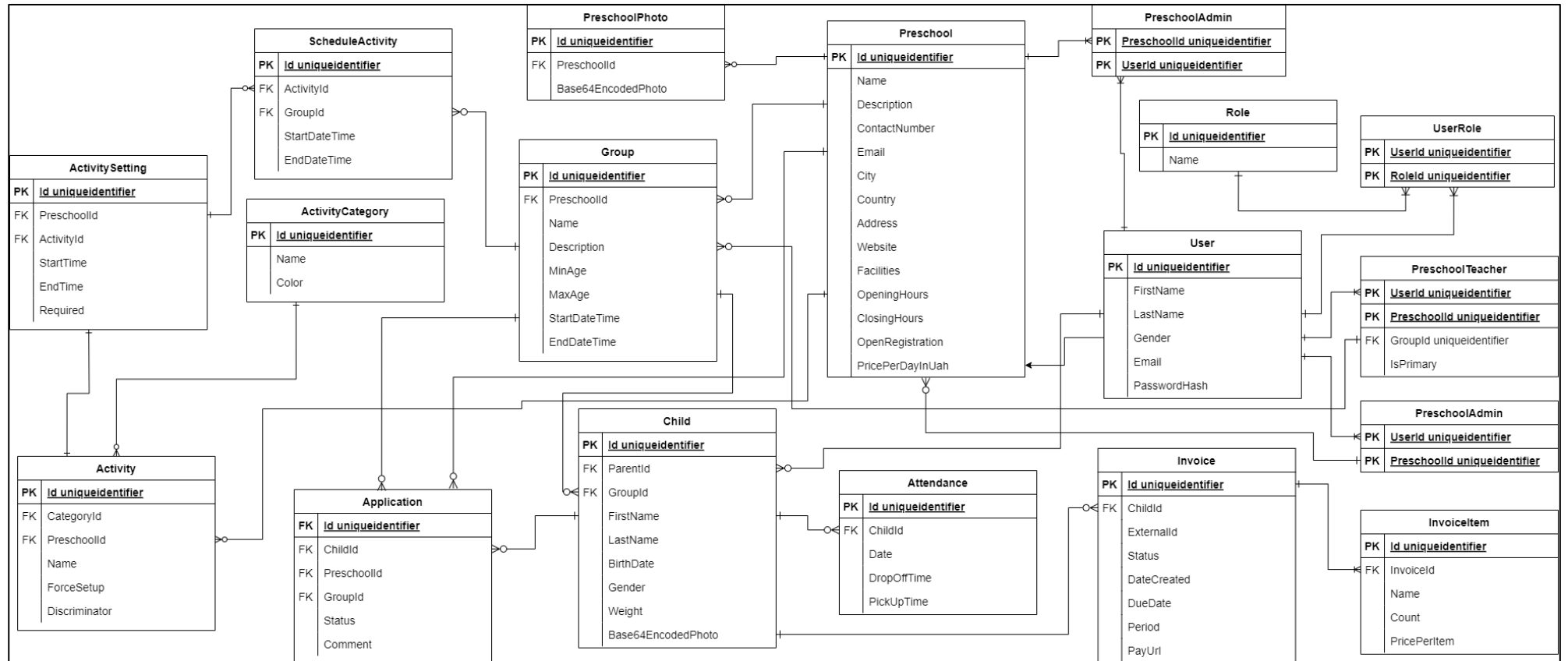


Рисунок Г.1 – ER-діаграма застосунку «ChildWell»

## ДОДАТОК Д

## Приклад програмного коду (компонент сканування QR-коду)

```

import React from "react";
import {
  Button,
  Text,
  View,
  StyleSheet,
  Alert,
  ActivityIndicator,
} from "react-native";
import {
  CameraView,
  useCameraPermissions,
  BarcodeScanningResult,
} from "expo-camera";
import agent from "@api/agent";
import { AttendanceMarked } from "@models/attendance";
import Loader from "@components/Loader";
import AttendanceMarkedModal from "@components/AttendanceMarkedModal";
import { BlurView } from "expo-blur";
import { useTranslation } from "react-i18next";
import { useTeacherGroup } from "@stores/teacherGroupsStore";
import { useGroupAttendanceStore } from "@stores/groupAttendanceStore";

const ScanQrCodeScreen = () => {
  const { t } = useTranslation("scanQrCodeScreen");
  const [permission, requestPermission] = useCameraPermissions();
  const [scannedAttendance, setScannedAttendance] =
    React.useState<AttendanceMarked>();
  const [isProcessing, setIsProcessing] = React.useState(false);
  const [showingError, setShowingError] = React.useState(false);
  const loadAttendanceHistory = useGroupAttendanceStore(s =>
    s.loadAttendanceHistory);
  const group = useTeacherGroup()!;

  if (!permission) {
    return <Loader />;
  }

  if (!permission.granted) {
    return (
      <View style={styles.container}>
        <Text style={{ textAlign: "center"
}}>{t("cameraAccessRequired")}</Text>
        <Button onPress={requestPermission} title={t("grantPermission")} />
      </View>
    );
  }

  const handleBarcodeScannedAsync = async (
    scanningResult: BarcodeScanningResult
  ) => {
    if (isProcessing || showingError || scannedAttendance) return;

    setIsProcessing(true);
  }

```

```

try {
  const res = await agent.Children.markAttendance(scanningResult.data);
  loadAttendanceHistory(group.preschool.id, group.id)
  setScannedAttendance(res);
} catch {
  setShowingError(true);
  Alert.alert(t("error"), t("failedToMarkAttendance"), [
    { text: t("ok"), onPress: () => setShowingError(false) },
  ]);
} finally {
  setIsProcessing(false);
}
};

return (
  <View style={styles.container}>
    <CameraView
      style={styles.camera}
      onBarcodeScanned={handleBarcodeScannedAsync}
      facing="back"
    >
      {scannedAttendance && (
        <BlurView intensity={100} style={StyleSheet.absoluteFill} />
      )}
      {!scannedAttendance && <CornerOverlay />}
      {isProcessing && (
        <View style={styles.loadingOverlay}>
          <ActivityIndicator size="large" color="white" />
        </View>
      )}
    </CameraView>
    <AttendanceMarkedModal
      attendanceData={scannedAttendance}
      visible={!scannedAttendance}
      onClose={() => {
        setScannedAttendance(undefined);
      }}
    />
  </View>
);
};

const CornerOverlay = () => {
  return (
    <View style={styles.overlay}>
      <View style={[styles.corner, styles.cornerTopLeft]} />
      <View style={[styles.corner, styles.cornerTopRight]} />
      <View style={[styles.corner, styles.cornerBottomLeft]} />
      <View style={[styles.corner, styles.cornerBottomRight]} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "black",
  },
});

```

```

camera: {
  flex: 1,
  justifyContent: "center",
  alignItems: "center",
},
overlay: {
  position: "absolute",
  left: 0,
  right: 0,
  top: 200,
  bottom: 200,
  justifyContent: "space-between",
  alignItems: "center",
},
corner: {
  width: 70,
  height: 70,
  position: "absolute",
  borderColor: "white",
},
cornerTopLeft: {
  borderTopWidth: 4,
  borderLeftWidth: 4,
  left: "5%",
  top: "5%",
},
cornerTopRight: {
  borderTopWidth: 4,
  borderRightWidth: 4,
  right: "5%",
  top: "5%",
},
cornerBottomLeft: {
  borderBottomWidth: 4,
  borderLeftWidth: 4,
  left: "5%",
  bottom: "5%",
},
cornerBottomRight: {
  borderBottomWidth: 4,
  borderRightWidth: 4,
  right: "5%",
  bottom: "5%",
},
loadingOverlay: {
  ...StyleSheet.absoluteFillObject,
  backgroundColor: "rgba(0,0,0,0.5)",
  justifyContent: "center",
  alignItems: "center",
},
});

```

```
export default ScanQrCodeScreen;
```

## ДОДАТОК Е

Диплом за друге місце на конференції



Рисунок Е.1 – Диплом за друге місце на конференції «Інформаційні інтелектуальні системи»

## ДОДАТОК Ж

Подані на міжнародний форум тези

УДК 004.89:373.2

**АНАЛІЗ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ  
ГЕНЕРАЦІЇ РОЗКЛАДУ ЗАНЯТЬ І ХАРЧУВАННЯ ДІТЕЙ В  
ЗАКЛАДАХ ДОШКІЛЬНОЇ ОСВІТИ**

Бондаренко А. А., Нестеренко В. В., Смейко Б. М.

Науковий керівник – к.т.н., доц. Побіженко І. О.

Харківський національний університет радіоелектроніки, каф. ПІ,  
м. Харків, Українаe-mail: [anna.bondarenko1@nure.ua](mailto:anna.bondarenko1@nure.ua), [vitalii.nesterenko@nure.ua](mailto:vitalii.nesterenko@nure.ua)e-mail: [bohdan.smieiko@nure.ua](mailto:bohdan.smieiko@nure.ua)

This research examines a software tool designed for preschools to manage children's daily schedule and nutrition. Through artificial intelligence, specifically GPT which is a generative AI technology, this system enables teachers to create schedules adjusted to the needs of preschool groups and eating patterns, ensuring they get the right schedule and eating habits. Parents can easily see their children's daily activities and meals, making it easier to stay informed about their well-being. The main purpose is to use this AI software to make the management of daily activities and nutrition in preschools more efficient and tailored to physical needs, enhancing both educational and health outcomes.

Актуальність дослідження впливає з необхідності оптимізації освітніх процесів та планування харчування в дошкільних закладах з використанням сучасних технологічних рішень [1]. Застосування штучного інтелекту, зокрема GPT (Generative Pre-trained Transformer) моделей, дозволяє реалізувати індивідуальний підхід до кожної вікової групи дитячого садочка, сприяючи гармонійному розвитку дітей. Використання ШІ для генерації розкладів та харчування дозволяє врахувати не тільки освітні, але й фізіологічні потреби дітей, забезпечуючи тим самим ефективне та здорове навчальне середовище.

Метою даного дослідження є створення системи на базі штучного інтелекту для автоматизації процесів планування розкладу освітньої діяльності і харчування [2] у дошкільних закладах. Такий підхід має на меті підвищення ефективності освітнього процесу та забезпечення урахування вікових особливостей дошкільнят. Для досягнення поставленої мети було використано як теоретичні, так і практичні методи дослідження, аналіз потреб користувачів системи та технології штучного інтелекту [1].

Штучний інтелект застосовується в процесі генерації розкладу дня для дошкільних закладів, автоматизуючи розподіл активностей, харчування та відпочинку у вигляді таймслотів на електронному календарі. Система аналізує вікові особливості та потреби груп, оптимізуючи розподіл занять для забезпечення гармонійного розвитку дітей.

Наукова новизна полягає в інтеграції штучного інтелекту для розробки комплексної системи, що враховує освітні та фізіологічні потреби дітей у дошкільних закладах. Система оптимізує розклад занять та режим харчування, що є значним кроком уперед у використанні ШІ в сфері раннього розвитку дітей. Результати дослідження демонструють, що впровадження штучного інтелекту може суттєво підвищити якість освітніх і виховних процесів.

Крім того, системи штучного інтелекту, які підтримують багатомовність, надають можливість планування розкладу для дітей, з використанням бажаної мови, що сприяє більш ефективній взаємодії з системою. Використання «сильної» багатомовності [3] в системах штучного інтелекту дозволяє зменшити складність і підвищити загальну ефективність, яка не потребує знань певної мови.

В оцінці аналогів слід зазначити, що багато рішень фокусуються на автоматизації та ефективності управлінських задач закладу, а не на плануванні розкладу освітньої діяльності та харчування. Система генерації розкладу занять і харчування дітей в закладах дошкільної освіти, яка опирається на потенціал штучного інтелекту, забезпечує детальний розгляд вікових особливостей дітей, що дозволяє значно покращити не лише освітній процес, але й враховувати фізіологічні потреби різних вікових груп.

Отже, дослідження вказує на значні переваги використання описаних технологічних рішень. Система спрямована на врахування вікових потреб, забезпечуючи ефективне планування та керування розкладом. З подальшим розвитком технології очікується, що система стане ще більш гнучкою та орієнтованою на забезпечення конкретних потреб вихованців, педагогів та батьків, вносячи істотний вклад у розвиток сфери раннього розвитку дітей.

Список використаних джерел:

1. On the Prospects of Incorporating Large Language Models (LLMs) in Automated Planning and Scheduling (APS) / V. Pallagani та ін. arXiv.org. URL: <https://arxiv.org/abs/2401.02500> (дата звернення: 20.02.2024).
2. Garcia M. B. ChatGPT as a Virtual Dietitian: Exploring Its Potential as a Tool for Improving Nutrition Knowledge. Applied System Innovation. 2023. Т. 6, С. 96. URL: <https://doi.org/10.3390/asi6050096> (дата звернення: 28.02.2024).
3. Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning / Erdem E., Babii A., Turuta O. та ін. Journal of Artificial Intelligence Research. 2022. Т. 73. С. 1131–1207. URL: <https://doi.org/10.1613/jair.1.12918> (дата звернення: 20.03.2024).