

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Програмної інженерії  
(повна назва)

## **АТЕСТАЦІЙНА РОБОТА** **Пояснювальна записка**

рівень вищої освіти - другий (магістерський)

Дослідження методів прийняття рішень для підтримки організації  
мотивації в системі мотивації дітей  
(тема)

Виконала: студентка 2 курсу, групи ПЗСм-18-1

Федоренко А. М.  
(прізвище, ініціали)

спеціальності 121- Інженерія програмного забезпечення  
(код і повна назва спеціальності)

Освітньо-професійної програми  
(тип програми)

Програмне забезпечення систем  
(повна назва освітньої програми)

Керівник доцент, к.т.н. Мазурова О.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. \_\_\_\_\_

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти - другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-професійна програма

Освітня програма Програмне забезпечення систем

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

### ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентці Федоренко Анастасії Михайлівні

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів прийняття рішень для підтримки організації мотивації в системі мотивації дітей

затверджена наказом університету від " \_\_\_\_ " \_\_\_\_\_ 20 \_\_\_\_ р № \_\_\_\_\_

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

17 грудня 2019 р.

3. Вихідні дані до роботи електронні ресурси за обраною тематикою, мінімальні вимоги до функціональності програми, загальні вимоги до архітектури системи

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області і постановка задачі, аналіз вимог до програмної системи, UML-моделювання предметної області, дослідження методів теорії прийняття рішень, розробка математичної моделі для підтримки дерев цілей, розробка математичної моделі для вирішення оптимізаційної задачі про призначення, розробка бази даних, розробка алгоритмів побудови дерева цілей та вирішення оптимізаційної задачі про призначення, програмна реалізація системи, тестування.



## РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 97 с., 25 рис., 2 табл., 15 джер.

ANDROID, JSON, RETROFIT, SPRING, БАЗА ДАНИХ, ЗАВДАННЯ, МЕТОД ПРИЙНЯТТЯ РІШЕННЯ, МОТИВУВАННЯ, НАГОРОДА, РОЗКЛАД.

Об'єктом дослідження є методи теорії прийняття рішень для підтримки організації мотивації в системі мотивування дітей.

Метою роботи є підвищення підтримки мотивації дітей в програмній системі мотивування дітей.

Методи розробки базуються на таких технологіях, як Android SDK, Java, Spring, Retrofit, Maven, MySQL.

В результаті роботи було досліджено методи теорії прийняття рішень у напрямку їх використання для підтримки мотивації дітей і програмно реалізовано систему мотивування дітей, яка представляє собою мобільний додаток для операційної системи Android.

ANDROID, DATABASE, DECISION-MAKING METHOD, JSON, MOTIVATING, RETROFIT, REWARD, SCHEDULE, SPRING, TASK,

The object of the study is the methods of decision-making theory to support the organization of motivation in the system of children's motivation.

The purpose of the work is to create a software system for motivating children.

Development methods are based on such technologies as Android SDK, Java, Spring, Retrofit, Maven, Postgres.

As a result, the methods of decision-making theory in the direction of their use to support children's motivation were investigated, and the system for children's motivating, which is a mobile application for the Android operating system, was implemented.

## ЗМІСТ

Вступ.....	7
1 Аналіз проблемної області та постановка задачі.....	9
1.1 Аналіз проблемної області створення систем мотивування дітей.....	9
1.2 Аналіз аналогів.....	11
1.3 Постановка задачі.....	16
2 Перелік вимог до програмної системи.....	18
2.1 Призначення розробки.....	18
2.2 Вимоги до програмного продукту.....	18
3 Опис прийнятих проектних рішень.....	21
3.1 Аналіз та UML-моделювання предметної області мотивування дітей.....	21
3.2 Дослідження методів теорії прийняття рішень .....	24
3.3 Розробка математичної моделі для підтримки дерев цілей.....	28
3.4 Розробка математичної моделі для вирішення оптимізаційної задачі про призначення.....	30
3.5 Розробка бази даних для зберігання математичної моделі.....	32
3.6 Розробка алгоритму побудови дерева цілей.....	37
3.7 Розробка алгоритму вирішення оптимізаційної задачі про призначення...	39
4 Опис програмної реалізації.....	42
4.1 Вибір технологій для програмної реалізації.....	42
4.2 Опис фізичної моделі бази даних.....	43
4.3 Інтерфейс та функціонал системи.....	45
5 Аналіз дослідницької експлуатації та можливих застосувань.....	58
5.1 Аналіз можливих застосувань.....	58

5.2 Опис тестування системи.....	59
Висновки.....	61
Перелік джерел посилання .....	62
Додаток А Слайди презентації .....	63
Додаток Б Тези доповіді .....	80
Додаток В Стаття .....	82
Додаток Г Лістинг коду .....	89
Додаток Д Рецензії .....	92
Додаток Е Відгук керівника.....	96

## ВСТУП

Мотивація – це особливий стан людини, який спонукає її до дії. Вона є стимулом для виконання будь-якої роботи або для вирішення поставлених завдань. За допомогою мотивації забезпечується зацікавленість виконавця у реалізації чогось, прагнення виконати певну роботу максимально ретельно [1].

Мотивація дітей є невід’ємною частиною їх виховання, адже наскільки успішно батьки ще в дитинстві закладуть внутрішні мотиваційні механізми, які спочатку направлені на такі цілі як гарно навчатися, слухатися батьків, уміти добре поводитися в громадських місцях і так далі, настільки ці навички в подальшому допоможуть добиватися схожих успіхів, але уже у дорослому житті.

На сьогоднішній день багато батьків скаржаться на те, що їм складно змусити дитину виконувати її домашні обов’язки та навчатися, акцентуючи увагу на тому, що у дитини взагалі немає зацікавленості до цього. Батьки починають миритися з думкою про те, що у їхньої дитини просто не має здібностей до певної діяльності, хоча, насправді, дитині просто бракує мотивації.

Виділяють декілька основних способів мотивування дітей, найбільш ефективними серед яких є заохочення, гейміфікація процесу, планування та постановка цілей. Комбінація різних способів може принести найкращий результат, тому можна зробити висновок, що існує необхідність у створенні програмної системи, яка б давала можливість мотивувати дитину за допомогою всіх вищезгаданих способів.

В атестаційній роботі метод мотивації за допомогою заохочення та гейміфікації реалізований у вигляді системи нагород. Діти матимуть можливість отримувати бали за виконання завдань, за що отримуватимуть бали, а в подальшому зможуть обміняти їх на нагороди. Планування завдань, які необхідно виконати буде реалізовано за допомогою автоматичного складання розкладу завдань на тиждень. Для того, щоб діти були більш зацікавлені у виконанні завдань із розкладу, цей розклад має бути найбільш оптимальним для них, тобто виконання якого може принести їм найбільшу кількість балів.

Об'єктом дослідження є проблема мотивування дітей. Предметом дослідження є методи прийняття рішень для підтримки організації мотивації в системі мотивації дітей. Метою роботи є дослідження методів теорії прийняття рішень для створення програмної системи мотивування дітей.

В ході виконання атестаційної роботи було проведено аналіз та моделювання предметної області мотивування дітей, проаналізовано методи теорії прийняття рішень, розроблено математичну модель побудови дерев цілей та оптимізаційної задачі про призначення, розроблено базу даних для зберігання математичної моделі та розроблено алгоритми для побудови дерев цілей та рішення оптимізаційної задачі про призначення.

За результатами атестаційної роботи магістра було розроблено презентацію (див. додаток А).

Було зроблено доповідь на Міжнародному молодіжному форумі. Тези доповіді наведено в додатку Б. Написана та подана до опублікування стаття «Дослідження методів прийняття рішень для підтримки мотивації в системі мотивації дітей» в матеріалах IV міжнародної науково-практичної конференції в Ліверпулі «Scientific Achievements of Modern Society» (див. дод. В). Лістинг коду наведено в додатку Г.

## 1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Аналіз проблемної області створення систем мотивування дітей

Під мотивацією в психології розуміють сукупність психологічних процесів, які спрямовують поведінку людини. Мотиваційні процеси лежать в основі активності людини та її психічного функціонування, вони визначають той чи інший напрям людської поведінки, її траєкторію [2].

На сьогоднішній день батьки часто скаржаться на те, що їхні діти не мають жодної зацікавленості у навчанні та виконанні домашніх обов'язків, що свідчить про недостатню мотивацію дітей.

Мотивація може бути зовнішньою та внутрішньою. Внутрішня мотивація – це внутрішнє прагнення до досягнення цілей. Внутрішня мотивація може привести до успіху в кар'єрі, відносинах або просто до переживання приємних емоцій. Діти, які за своєю природою мотивовані, постійно прагнуть до розвитку і шукають нові шляхи самовдосконалення. Внутрішньо мотивованій дитині не потрібно розповідати, що робити і як себе поводити, щоб досягти мети.

Але якщо дитина не мотивована, їй потрібні постійні нагадування і зовнішня мотивація, щоб виконати роботу. Зовнішня мотивація викликана зовнішніми факторами. Для дитини зовнішньою мотивацією може бути дозвіл подивитися телевізор, обіцянка купити морозиво, іграшку і т.д.

Щоб дізнатися, що мотивує дитину, треба подивитися на неї як на особистість зі своїми смаками і перевагами та дізнатися, які сфери чи предмети її цікавлять. Знаючи, чого хоче дитина, батьки будуть знати, як її мотивувати.

Сьогодні психологами запропоновано безліч способів, які можуть сприяти мотивації дитини і кожен з них має як свої переваги, так і недоліки. Одним із таких способів є мотивування за допомогою маленьких «нагород», які дитина отримує за кожне виконане завдання. Даний метод сприяє формуванню зовнішньої мотивації, яка в свою чергу певним чином впливає на внутрішню. Маленькі «нагороди» є своєрідним стимулом для дитини, у дитини з'являється зацікавленість до роботи, яку вона виконує – розвивається зовнішня мотивація, що збільшує шанси того, що ця робота згодом стане звичкою.

Одним із важливих способів сприяння мотивації є планування задач, які необхідно виконати. Планування мотивує дитину досягати поставлених цілей, оскільки воно дає напрямок і керівництво на кожному етапі. Батькам необхідно допомогти дитині спланувати в деталях те, до чого вона проявляє інтерес.

Варто звернути увагу й на такий спосіб мотивування як постановка чітких цілей. Цілі дають дитині перспективу і напрямок, які спрощують виконання завдання. Цілі можуть бути простими: вчасно лягати спати, щоб вранці встигати у школу, вчасно вчити уроки, щоб залишався час пограти, і т.д. Це можуть бути і довгострокові цілі, які приведуть дитину до успіху в кар'єрі або відносинах (така стратегія особливо хороша для підлітків). Необхідно пам'ятати, що жодна мета не є кінцевою – це просто черговий крок до того, щоб досягти більш високого рівня або стати кращим.

Можна зробити висновок, що програмна система, яка розроблюється, має комбінувати в собі всі вищезгадані способи мотивування для досягнення найкращого результату.

Програмна система буде представляти собою мобільний додаток, оскільки це дає можливість користуватися ним у будь-який зручний для користувача час і в будь-якому місці. В даний час найбільш популярними мобільними ОС є Android від компанії Google, iOS від Apple і Windows Phone від компанії Microsoft. Лідируючу позицію вже не перший рік займає операційна система Android, і завдяки своїй поширеності вона була обрана платформою для розробки програмної системи мотивування дітей.

Як уже було згадано вище, програмна система, що розроблюється, буде включати в себе функціонал складання оптимального розкладу завдань на період часу, які дитині необхідно виконати. На сьогоднішній день для отримання ефективного рішення у багатьох сферах життєдіяльності існує необхідність у науковому підході, адже втрати, пов'язані з помилками, можуть бути дуже великими. Оптиміальні рішення дозволяють досягти мети при мінімальних трудових, матеріальних і сировинних витратах.

Таким чином, аналізу і методам прийняття оптимальних рішень (ефективних рішень) в даний час приділяється дуже велика увага. Методи пошуку оптимальних рішень розглядають в розділах класичної математики, пов'язаних з вивченням екстремумів функцій, в математичному програмуванні. Рішенням цих завдань є математичний об'єкт, основною властивістю якого є те, що він приносить екстремум заданої функції або функціоналу.

Як правило, оцінка рішення проводиться по одному аспекту або критерію. На практиці рішення потрібно оцінити з різних сторін, з огляду на фізичні (габарити, вага), економічні (вартість, ресурсомісткість), технічні (реалізовані функції) та інші критерії. Все це вимагає побудови моделі оптимізації рішень одночасно за кількома критеріями. Такі моделі розробляють в теорії вибору і прийняття рішень. Тут при постановці завдання вже недостатньо побудувати функціонали, що оптимізуються – потрібно ввести принцип оптимальності, який визначає поняття оптимального рішення [3].

Виходячи з усього вищезазначеного, для розробки програмної системи мотивування дітей впливає необхідність в математичному моделюванні системи, розробці схеми бази даних для зберігання створеної математичної моделі, аналізі методів, які можна використати для побудови оптимального розкладу завдань та підтримки цілей, та розробці алгоритмів для реалізації цих методів.

## 1.2 Аналіз аналогів

Для того щоб виділити основні функціональні можливості програмної системи, розставити пріоритети та уникнути недоліків, допущених компаніями, що розробляли схожі системи, необхідно проаналізувати дані системи. Сьогодні, коли технології стали важливою частиною нашого життя, існує декілька мобільних додатків, які сфокусовані на мотивування дитини за допомогою «нагород» і є аналогами додатку, що розроблюється.

Одним із таких додатків є IReward Chart (див. рис. 2.1), що є системою для батьків та дітей для обговорення, управління та заохочення гарної поведінки.

Додаток використовує систему нагород, що залежить від продуктивності та поведінки.

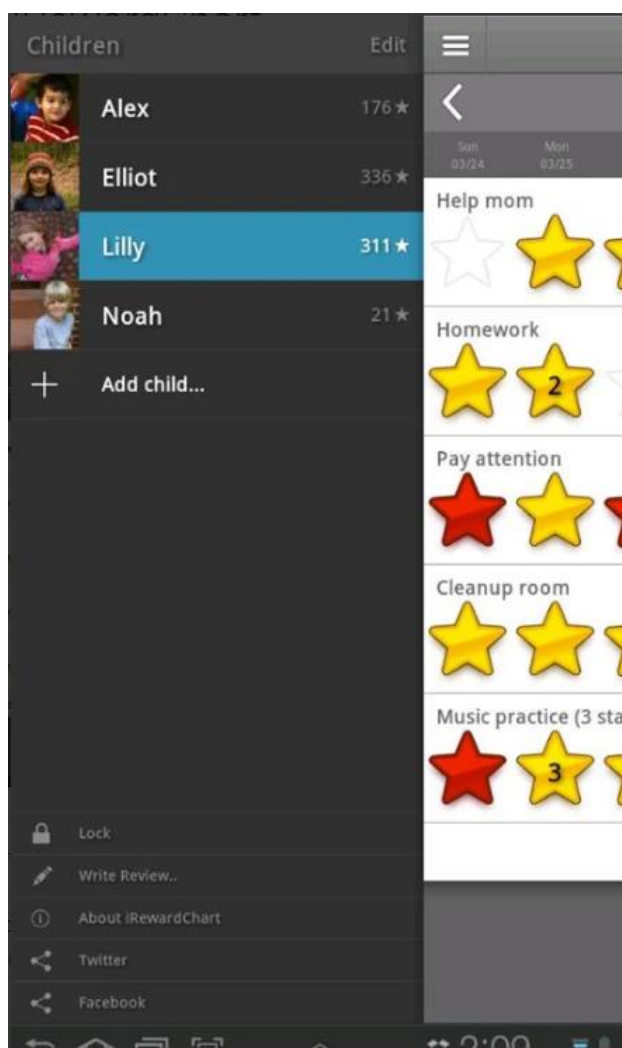


Рисунок 1.1 – Інтерфейс додатку iReward Chart

Він дає можливість батькам дарувати дітям зірки, а дітям обмінювати ці зірки на різні нагороди. iReward Chart допомагає дітям бути в курсі своєї поведінки, чітко розуміти, що від них очікується, усвідомлювати наслідки, що впливають з їхніх дій. Додаток орієнтований на дітей 4-10 років.

Він орієнтований тільки на IOS пристрої і, безперечно, його недоліком є неможливість додавання більш, ніж однієї дитини у безкоштовній версії додатку.

Іншим аналогом додатку, що розроблюється, може слугувати Vubu – застосунок, що дає можливість батькам мотивувати своїх дітей у ігровій формі

(див. рис. 2.2). Додаток є безкоштовним і його перевагою є зручний інтерфейс користувача.

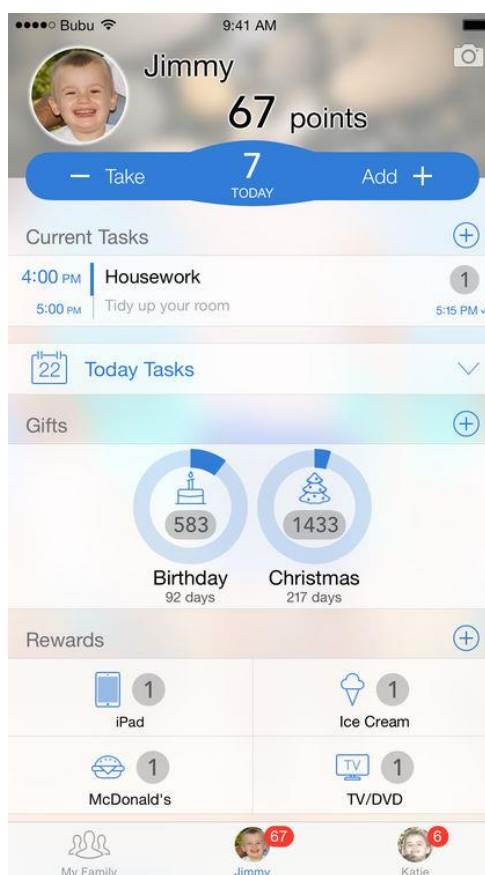


Рисунок 1.2 – Інтерфейс додатку Bubu

Серед особливостей додатку необхідно зазначити наступне:

- відсутність паролів – для входу необхідно ввести тільки нікнейм;
- двоє батьків – можливість запрошувати свого партнера, щоб мати спільних дітей у додатку;
- автоматичний менеджмент регулярних завдань;
- можливість обирати завдання із стандартного списку;
- зняття трьох балів опівночі кожного дня;
- друк розпорядку на тиждень (платна версія);
- нотифікації (платна версія);
- додавання більш, ніж однієї дитини (платна версія);
- взаємодія з друзями і обмін балами (платна версія);

- додавання необмеженої кількості балів вручну (платна версія);
- можливість вибору кольорової схеми додатку (платна версія).

Також аналогом Kidstim є мобільний додаток DooApp (див. рис. 2.3), створений харківською командою розробників з компанії Screen Interactive. DooApp має версії додатку для IOS та Android. Перша версія додатку, що розроблюється, буде найбільш схожа на нього за функціоналом. Даний додаток розроблений з використанням технологій HTML5, CSS, Ajax, JavaScript: jQuery і jQuery Mobile для фронтенд частини та PHP і MySQL для бекенду.

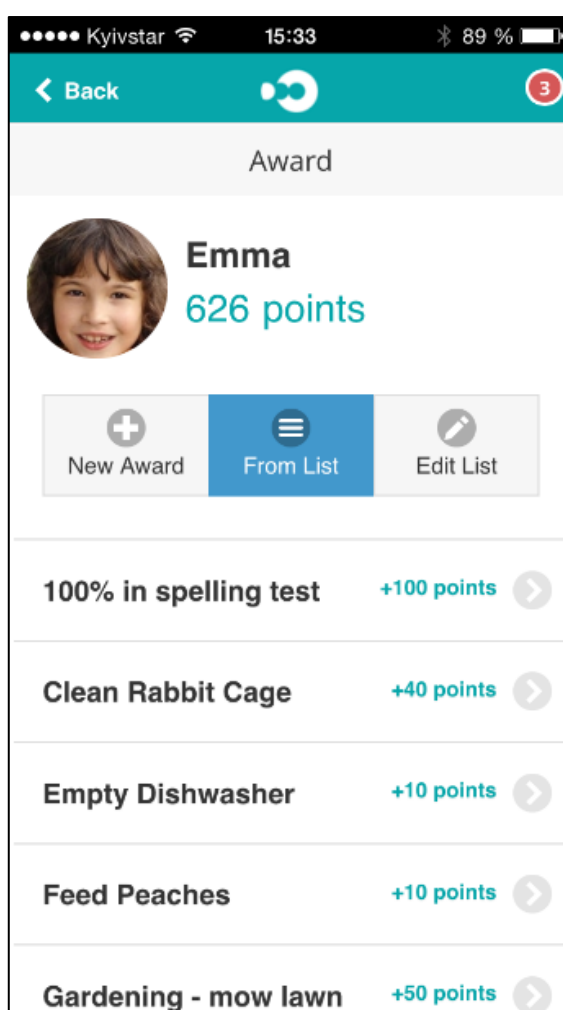


Рисунок 1.3 – Інтерфейс додатку DooApp

Разом із вищезгаданими додатками, аналогами проекту, що розроблюється, є ChorePal та HomeyApp – на відміну від інших додатків, що передбачають

маленькі нагороди за виконані завдання або спільне дозвілля батьків та дітей, ці додатки базуються на грошових винагородах.

Такий підхід критикується психологами, особливо у випадках з дуже малими дітьми, адже це погіршує стосунки між батьками та дітьми. Грошові нагороди, наприклад, за виконання домашніх завдань, зміщують справжні мотиви дитини навчатися і формують неправильні поведінкові моделі.

Також варто звернути увагу на те, що розмір грошової нагороди має рости з часом, тому що потреби мають властивість змінюватися. Батьки можуть прийти до того, що за виконання навіть елементарних справ діти будуть вимагати від них непомірно високої плати. Інтерфейс згаданих додатків спрямований на зацікавлення дитини у грошовій нагороді (див. рис. 1.4).

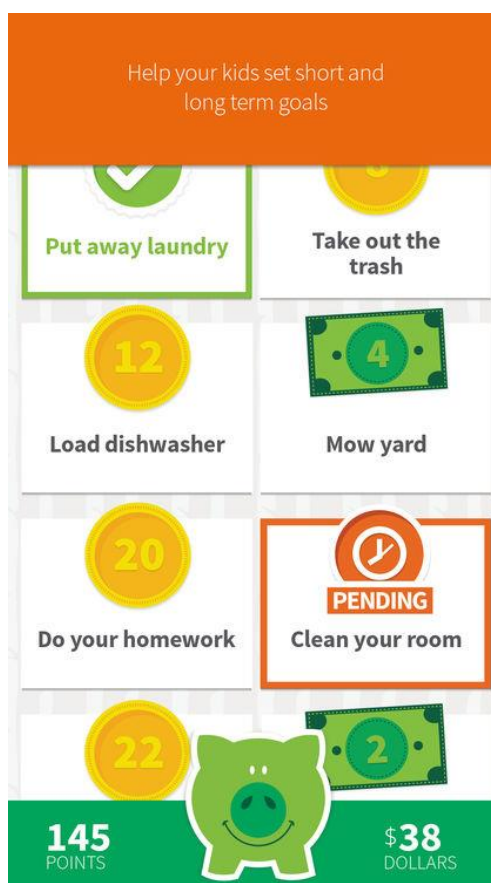


Рисунок 1.4 – Інтерфейс додатку ChorePal

Майже всі вище описані додатки працюють тільки під iPhone та iPad, тобто власники Android, наприклад, взагалі не мають можливості ними скористатися.

Разом з тим вони не забезпечують батьків можливістю автоматичного складання розкладу та управління цілями. Саме тому існує необхідність у створенні програмної системи, яка б давала змогу мотивувати дітей за допомогою нагород, створювати оптимальний розклад виконання завдань на період часу та прив'язувати конкретні завдання до більш високорівневих цілей.

### 1.3 Постановка задачі

В атестаційній роботі магістра потрібно провести дослідження методів теорії прийняття рішень для підтримки мотивації в системі мотивації дітей.

На основі моделей та методів теорії прийняття рішень необхідно розробити математичну модель системи підтримки мотивації, яка допоможе вирішити задачі побудови дерева мотиваційних цілей та призначення завдань в розклад дитини з урахуванням важливості та інших характеристик завдань, а також мотиваційних цілей.

Таким чином, в ході виконання роботи необхідно виконати наступні завдання:

- провести аналіз та моделювання (UML та інше) предметної області мотивування дітей;
- дослідити методи теорії прийняття рішень для підтримки мотивування дітей;
- розробити математичну модель системи підтримки мотивації та базу даних для зберігання інформації з неї;
- розробити алгоритми підтримки мотивації;
- програмно реалізувати систему мотивації дітей та провести її тестування.

В роботі потрібно спроектувати та розробити мобільний додаток для мотивування дітей на платформі Android, який даватиме можливість скласти оптимальний розклад завдань на тиждень, підтримувати функціонал управління цілями та нагородами та функціонал встановлення пріоритетів цілей. Додаток має надавати можливість дітям надсилати запити на отримання нагород.

Для розробки програмної системи було обрано середовище розробки IntelliJ IDEA, оскільки воно надає значно більше можливостей, ніж його конкуренти, і є більш зручним для використання. Було прийнято рішення розробляти мобільний додаток під платформу Android, оскільки вона є найбільш поширеною серед мобільних платформ. Те, що додаток є мобільним, дасть змогу користувачам використовувати його у будь-якому місці у зручний для них час.

Для написання серверної частини обрано мову програмування Java [4] з використанням технологій Spring MVC та Hibernate. Spring дозволяє створювати додатки, код яких слабопов'язаний між собою, завдяки вбудовуванню залежностей [5]. Також ця технологія дозволяє відділяти налаштування компонентів від їх програмного коду та зручно керувати транзакціями. Hibernate дає можливість позбутися від дублювання коду при реалізації шару доступу до даних та не залежати від бази даних, що використовується.

Для написання клієнтської частини буде використовуватися Android SDK з використанням додаткових бібліотек. В якості бази даних було обрано MySQL, що дає можливість створювати складні залежності між сутностями.

## 2 ПЕРЕЛІК ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Кожна програмна система – це перетворювач, функцією якого є визначене оброблення даних і вивід отриманих результатів. З метою побудови програмної системи до неї, насамперед, формуються вимоги до умов виконання функції і обробки даних. У загальному випадку під вимогами до програмної системи розуміють властивості, які повинна мати система для виконання запропонованих замовником функцій. В дипломному проекті треба реалізувати систему мотивування дітей, яка б давала можливість автоматичного складання розкладу завдань для дітей з урахуванням їх цілей. Для створення якісного сервісу необхідно чітко визначитися з вимогами до нього.

### 2.1 Призначення розробки

Програмна система мотивування дітей та складання оптимального розкладу виконання завдань призначена в першу чергу для батьків, які хочуть мотивувати своїх дітей до виконання домашніх завдань та обов'язків по дому, а також для самих дітей. Необхідно розробити мобільний додаток, який давав би змогу створювати завдання, які діти мають виконати, а також нагороди, які вони можуть отримати за виконання цих завдань. Для того, щоб батьки не забували про те, що настав час для отримання нагороди дитиною, додаток має відправляти їм нагадування на мобільний пристрій. Зазвичай, батьки мають деякий досвід складання розкладу виконання завдань для своїх дітей, але він може виявитися досить суб'єктивним, тому складений розклад може бути неоптимальним або не враховуватиме багато важливих факторів.

Саме тому необхідно розробити програмну систему, що допоможе батькам автоматично створювати оптимальний розклад завдань для дітей, який буде враховувати дедлайни завдань, їх важливість, а також актуальні цілі дітей.

### 2.2 Вимоги до програмного продукту

Програмна система повинна бути побудована на клієнт-серверній архітектурі і реалізована за допомогою мови програмування Java і Kotlin. Сервер

має бути абсолютно незалежним від клієнта, для того, щоб він міг функціонувати на іншій машині. Серверна частина повинна бути реалізована з використанням фреймворку Spring MVC для розмежування шарів системи та забезпечення спільного використання компонентів. Для реалізації шару доступу до даних було вирішено використовувати ORM Hibernate, оскільки він дозволяє значно спростити представлення даних та зменшити кількість коду, що дублюється. У якості СУБД має використовуватися MySQL.

Клієнтська частина повинна бути реалізована за допомогою Android SDK та деяких зовнішніх бібліотек для спрощення реалізації окремих можливостей системи (наприклад, Butterknife – для більш зручного способу встановлення зв'язку між UI елементами і програмним кодом, Glide – для завантаження зображення на сервер/з серверу та кешування зображень. Для посилення запитів до сервера має бути використаний Retrofit2, що дає змогу реалізувати звернення до сервера за допомогою анотацій і не дублюючи код.

Дані, що передаються з клієнта на сервер, мають бути захищені від зовнішнього втручання та сторонньої модифікації. Всі зображення батьків та дітей мають зберігатися на сервері у вигляді звичайних файлів, а в базі даних мають зберігатися лише шляхи до цих файлів. Для більш швидкого відгуку системи, має бути передбачене кешування зображень.

Основна функціональність програмної системи полягає у наступному:

- створення цілей, підцілей та конкретних завдань для їх досягнення;
- установка пріоритетів для цілей та підцілей;
- створення списків нагород для дітей;
- отримання нагадувань про необхідність надання нагороди дитині;
- редагування та видалення завдань і нагород;
- можливість відмітити завдання як виконані;
- можливість автоматичного складання розкладу завдань на тиждень;
- фільтрація завдань та нагород.

Користувачам має бути доступна функція реєстрації у системі. Всі користувачі, які реєструються у системі, повинні автоматично належати до

категорії батьки. Користувачі категорії діти мають додаватися в систему самими батьками. Батьки повинні мати можливість створювати завдання, призначаючи декілька дітей для його виконання. Після того, як батьки помічають завдання як виконане, кількість балів дитини повинна збільшуватися на відповідну кількість. Діти не повинні мати змоги робити запит на отримання нагороди, якщо вони не мають достатньої кількості балів.

Оскільки додаток орієнтований на дітей, то він повинен мати максимально простий та зрозумілий інтерфейс користувача, не перевантажений графічними елементами. До інтерфейсу користувача висуваються наступні вимоги:

- уніфікація розміщення керуючих елементів;
- адаптивність розмітки для пристроїв з різними розмірами екранів;
- підтвердження виконання критичних дій у системі;
- мінімізація кількості полів, що заповнюються;
- сповіщення користувача про виконання ним некоректних дій;
- мінімально можливий час відгуку системи [6].

При проектуванні системи повинна бути передбачена можливість перекладу інтерфейсу на різні мови.

Безпека програмної системи має забезпечуватися такими засобами, як формування різних рівнів доступу із переліком допустимих прав та авторизацією користувачів. Вразливі дані, необхідні для функціонування системи (наприклад, паролі), мають зберігатися на сервері у зашифрованому вигляді.

Стійке функціонування додатку має бути забезпечене шляхом контролю коректності та повноти вхідних даних – всі дані, що вводяться користувачем, перевіряються на формальну коректність.

Отже у ході розробки програмної системи необхідно проаналізувати предметну область мотивування дітей, спроектувати та розробити за допомогою обраних технологій та мови програмування інформаційну систему для мотивування дітей і складання оптимального розкладу завдань. Розроблена програма повинна відповідати вимогам зручності використання та надавати можливість перегляду, редагування та обробки усієї необхідної інформації.

### 3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

#### 3.1 Аналіз та UML-моделювання предметної області мотивування дітей

Для ефективної розробки системи мотивування дітей необхідно проаналізувати, які способи мотивування існують і який функціонал допоможе втілити дані способи в програмній реалізації. Одним із таких способів є заохочення, або ж мотивування за допомогою маленьких «нагород», які дитина отримує за кожне виконане завдання. Маленькі «нагороди» є своєрідним стимулом для дитини, тому у неї з'являється зацікавленість до роботи, яку вона виконує. Заохочення чудово поєднується з гейміфікацією – підходом, коли елементи гри використовуються у неігрових процесах. У системі мотивування дітей, що розроблюється, заохочення і гейміфікація будуть реалізовані у вигляді своєрідної системи нагород. За кожне виконане завдання діти будуть отримувати певну кількість балів. Батьки матимуть можливість створювати списки нагород і визначати кількість балів, яку необхідно здобути дитині для їх отримання. Коли у дитини буде достатня кількість балів, вона може обміняти їх на нагороду.

Одним із важливих способів сприяння мотивації є планування задач, які дитині необхідно виконати. Планування мотивує дитину досягати поставлених цілей, оскільки воно дає напрямок і керівництво на кожному етапі. Дуже часто батьки не можуть ефективно спланувати проведення часу своїх дітей, призначають їм занадто багато завдань одночасно або просто часто змінюють свою думку про те, чим повинна займатися дитина в конкретний момент часу. Саме для вирішення цієї проблеми і для сприяння мотивації за допомогою планування, програмна система буде мати функціонал складання автоматичного розкладу завдань на тиждень, які дитина має виконати. Було обрано тиждень як період, на який необхідно скласти розклад, оскільки пріоритети завдань, які потрібно виконати, змінюються дуже часто. Розклад буде будуватися з урахуванням деяких обмежень на завдання та самі дні. Кожний день, на який необхідно скласти розклад, має певний визначений ліміт часу, який дитина може витратити на виконання завдань. Батьки матимуть можливість задавати ці ліміти перед складанням розкладу. Кожне завдання включає в себе параметри, на основі

яких будується оптимальний розклад завдань на тиждень: кількість балів, які дитина може отримати за виконання цього завдання (рахується на основі важливості загальних цілей та підцілей, а також часу, необхідного для виконання завдання), мінімальна та максимальна частота виконання завдання, дедлайн та час виконання.

Варто звернути увагу й на такий спосіб мотивування як постановка чітких цілей. Цілі дають дитині перспективу і напрямок, які спрощують виконання завдань. Цілі можуть бути простими: вчасно лягати спати, щоб вранці встигати у школу, вчасно вчити уроки, щоб залишався час пограти, і т.д. Це можуть бути і довгострокові цілі, які приведуть дитину до успіху в кар'єрі або відносинах (така стратегія добре підходить для підлітків). У програмній системі для мотивування дітей постановка цілей буде реалізована за допомогою функціоналу, що дає можливість батькам створювати загальні цілі верхнього рівня, підцілі та конкретні завдання. Батьки матимуть змогу встановлювати пріоритети цілям, попарно порівнюючи їх важливість між собою. Бачивши перед собою не просто список завдань, а також і мету, для якої це завдання виконується, діти будуть більш зацікавлені у їх виконанні.

Взявши до уваги всю вищезгадану інформацію, було побудовано діаграму прецедентів [7] (див. рис. 3.1), яка дозволяє візуалізувати основний функціонал програмної системи для підтримки мотивації дітей. З даної діаграми видно, що система має дві категорії користувачів: батьки та діти. Користувачі мають можливість увійти в систему, після чого обом категоріям користувачів доступний функціонал перегляду інформації свого профілю, перегляду списків нагород (для батьків – створених ними, для дітей – потенційних), перегляду списків цілей та завдань, фільтрації завдань та нагород.

Користувачі категорії батьки мають змогу реєструвати дітей в системі, додавати, видаляти та редагувати цілі, завдання, і нагороди, відмічати завдання як виконані, встановлювати часові ліміти на кожний день перед генерацією розкладу, встановлювати пріоритети створеним цілям. Користувачі категорії діти

мають можливість надіслати батькам сповіщення про бажання отримати нагороду.



Рисунок 3.1 – Діаграма прецедентів

Побудована діаграма прецедентів дає можливість проаналізувати основні вимоги до системи та візуалізувати її з точки зору користувача.

### 3.2 Дослідження методів теорії прийняття рішень

Програмна система мотивування дітей повинна включати в себе функціонал управління цілями та розробки оптимального розкладу завдань, які необхідно виконати дитині в певний період часу. Оскільки проблема розробки розкладу для дітей направлена на визначення найкращого (оптимального) способу дій для досягнення певної мети, то її можна вважати задачею прийняття рішень, розв'язанням яких займається теорія прийняття рішень (ТПР). ТПР є набором понять і систематичних методів, які дозволяють всебічно аналізувати проблеми прийняття рішень в різних умовах (визначеності, невизначеності, ризику, конфліктних ситуацій або нечіткості) [8]. Дана теорія сформувалась на базі теорії дослідження операцій і по аналогії з нею вона займається розробкою і застосуванням методів обґрунтування рішень. Класифікація методів прийняття рішень необхідна для визначення загальних і конкретно-специфічних підходів до їх розробки, використання і оцінки, що дозволяє підвищити якість прийнятих рішень. Різноманіття ситуацій, цілей та умов, в рамках яких виникають завдання прийняття рішень, ускладнює проведення цієї класифікації, однак можна виділити наступні параметри:

- сфера вирішуваних завдань прийняття рішень і застосування методів (економічні, технічні, технологічні і т.д.);
- цільовий характер задач прийняття рішень (оперативні, тактичні, стратегічні);
- облік якості і визначеності вихідної інформації (завдання і методи прийняття рішення в умовах визначеності – з детермінованими характеристиками; завдання і методи прийняття рішень для умов невизначеності, в тому числі для умов ймовірнісної невизначеності, умов ризику і умов повної невизначеності; завдання та методи прийняття рішень в конфліктних ситуаціях, коли в процесі управління економічними системами виникають ситуації, при яких стикаються інтереси декількох конкуруючих сторін, що переслідують протилежні цілі);
- шкали значень обраних критеріїв. При цьому мова йде не тільки про те, що вони можуть бути як числовими, так і нечисловими, але й про ті методи, які

використовуються для вимірювання цих критеріїв з метою їх оцінки. Дана класифікаційна ознака є істотним важливою для багатокритеріальних задач, так як для вимірювання різних критеріїв можуть використовуватися різні шкали значень цих критеріїв;

– кількість критеріїв для оцінки рішення, що приймається. За цим параметром можна виділити однокритеріальні і багатокритеріальні задачі і методи прийняття рішень. У свою чергу багатокритеріальні методи на рівні алгоритмічної реалізації можуть включати цілий перелік інших методів, наприклад, метод головного критерію, метод лінійної згортки, метод максимінної згортки, метод лексикографічної оптимізації, метод Нелдера-Міда, адаптивний метод, метод Сааті, метод обмежень і т.п.;

– спосіб ранжування критеріїв прийняття рішень. При використанні багатокритеріальних методів прийняття рішень може використовуватися лексикографічне ранжування критеріїв, ранжування критеріїв способом попарних порівнянь, ранжування на основі виділення недомінуючих критеріїв (множина Парето) [9];

– вихідні методи розробки рішень. До їх числа можна віднести ситуаційні і кількісні методи прийняття рішень. Ситуаційні методи, пов'язані з характером виникаючих обставин. Особливість ситуаційних методів полягає в тому, що вони застосовуються для формалізації певного виду людської діяльності, орієнтованої на встановлення найкращого варіанту дій в даній прикладній ситуації. В основі кількісних методів лежить науково-практичний підхід, що передбачає вибір оптимальних рішень з використанням інструментальних засобів і автоматизованої обробки великих об'ємів інформації. Залежно від типу математичних функцій, покладених в основу моделей, розрізняють декілька видів програмування. Лінійне програмування – коли використовуються лінійні залежності в цільовій функції і обмеженнях на значення шуканих змінних. Нелінійне, цілочисельне і комбінаторне програмування – коли функціональні залежності в цільовій функції і обмеженнях носять дискретний, цілочисельний або комбінаторний характер. Динамічне програмування розглядає в динаміці процес поетапного

багатокрокового рішення задачі і вводить додаткові змінні на окремих кроках цього процесу. Ймовірнісне і статистичне програмування використовується для прийняття рішень в системах, процес функціонування яких пов'язаний з випадковими чинниками, і реалізується на основі чисельних методів і комп'ютерного моделювання [10].

Оскільки складання оптимального розкладу для дітей – це процес призначення завдань на конкретні дні з виконанням різних обмежень (дедлайн, мінімальна і максимальна частота виконання завдань і т.д.), і ці обмеження можуть бути представлені у вигляді лінійних рівнянь або нерівностей, то можна припустити, що дана задача може бути вирішена за допомогою алгоритмів лінійного програмування.

В загальному виді задача лінійного програмування розподілу ресурсів може бути представлено у вигляді таблиці (див. табл. 3.1).

Таблиця 3.1 – Загальний вигляд задачі лінійного програмування

I	Ресурси	Роботи, які необхідно виконати						Об'єм
		J <sub>1</sub>	J <sub>2</sub>	...	J <sub>j</sub>	...	J <sub>n</sub>	наявних R
1	R <sub>1</sub>	C <sub>11</sub>	C <sub>12</sub>	...	C <sub>1j</sub>	...	C <sub>1n</sub>	b <sub>1</sub>
2	R <sub>2</sub>	C <sub>21</sub>	C <sub>22</sub>	...	C <sub>2j</sub>	...	C <sub>2n</sub>	b <sub>2</sub>
.	...	...	...	...	...	...	...	...
I	R <sub>i</sub>	C <sub>i1</sub>	C <sub>i2</sub>		C <sub>ij</sub>	...	C <sub>in</sub>	b <sub>i</sub>
.	...	...	...	...	...	...	...	...
M	R <sub>m</sub>	C <sub>m1</sub>	C <sub>m2</sub>	...	C <sub>mj</sub>	...	C <sub>mn</sub>	b <sub>m</sub>
	Об'єм необхідних R	a <sub>1</sub>	a <sub>2</sub>	...	a <sub>j</sub>	...	a <sub>n</sub>	

У таблиці C<sub>ij</sub> – витрати або прибуток, щодо виділення однієї одиниці ресурсу R<sub>i</sub> до роботи J<sub>j</sub>. За шукану невідому величину X<sub>ij</sub> приймається об'єм і-го

ресурсу на виконання  $j$ -ї роботи. Витрати (або прибуток) на використання  $i$ -го ресурсу для виконання  $j$ -ї роботи дорівнюють  $X_{ij}C_{ij}$ .

Задача лінійного програмування може бути описана наступною математичною моделлю:

$$Z = \sum_i \sum_j C_{ij} X_{ij} \rightarrow \min, \quad (1)$$

де  $C_{ij}$  – витрати або прибуток, якщо виділено ресурс  $R_i$  для роботи  $J_j$ ;

$X_{ij}$  – об'єм  $i$ -го ресурсу на виконання  $j$ -ї роботи;

Система обмежень даної функції представлена нерівностями:

$$\begin{aligned} \sum_j X_{ij} &\leq b_i, \forall i = \overline{1, m}, \\ \sum_j X_{ij} &\geq a_j, \forall j = \overline{1, n}, \\ X_{ij} &\geq 0, \end{aligned} \quad (2)$$

де  $b_i$  – об'єм наявних ресурсів;

$a_j$  – об'єм необхідних ресурсів.

Найбільш відомим і широко застосовуваним на практиці для вирішення загальної задачі лінійного програмування є симплекс-метод. Незважаючи на те, що він є досить ефективним алгоритмом, який часто демонструє хороші результати при вирішенні прикладних задач, він має експоненційну складність. Причина цього полягає в комбінаторному характері симплекс-методу, який послідовно перебирає вершини багатогранника допустимих рішень при пошуку оптимального рішення [11].

Одним із підходів до вирішення задач лінійного програмування є застосування жадібних алгоритмів. При застосуванні цього методу основне завдання знову розбивається на підзадачі (на етапи), але на відміну від динамічного програмування, кожна підзадача вирішується як окрема, не пов'язана з іншими підзадачами, і спільне рішення основної задачі являє собою суму рішень

підзадач. Вважається, що до оптимізаційної задачі можна застосувати принцип жадібного вибору, якщо послідовність локально оптимальних виборів дає глобально оптимальне рішення [12].

Виходячи з усього вищеназваного для системи мотивування дітей було обрано жадібний алгоритм для вирішення оптимізаційної задачі складання розкладу для дитини.

### 3.3 Розробка математичної моделі для підтримки дерев цілей

В багатьох сферах життя необхідний порядок і вибудована структура, і саме тому для ранжування та ясності часто використовується поняття дерев цілей. Дерево цілей – це структурована, побудована за ієрархічним принципом (розподілена по рівнях) сукупність цілей певної системи, програми, плану, в якій виділені головна мета («вершина дерева»), підлеглі їй підцілі першого, другого і подальшого рівнів («гілки дерева»). Назва «дерево цілей» пов'язана з тим, що схематично представлена сукупність розподілених за рівнями цілей нагадує за виглядом перевернуте дерево. За допомогою даного методу можна відобразити як особисті плани, так і професійні, причому вони можуть бути місячні, річні, а також орієнтовані на більш тривалий термін. Дерево цілей дозволяє наочно відобразити цілісну систему конкретних рішень, орієнтованих на досягнення основного, глобального орієнтиру. Особливістю цього методу є те, що він відображає не тільки окремі цілі, але і їх взаємодію і зв'язок. Іншими словами, другорядні цілі є своєрідним інструментом для досягнення інших, які в сукупності призводять до успіху – досягнення головного цілі.

Будемо вважати системою підтримки мотивації  $M = \{GT, Z^{opt}\}$  таку систему, яка на базі дерева мотиваційних цілей  $GT$  дозволяє побудувати оптимізаційну задачу  $Z^{opt}$  про призначення завдань в розклад дитини.

Дерево цілей може бути представлене як  $GT = \langle G, R \rangle$ , де  $G$  – множина всіх цілей, а  $R$  – множина зв'язків між ними. Множина цілей  $G$  може бути представлена як  $G = \{G_i^d\}$ ,  $d = \overline{1, n}$ , де  $n$  – кількість рівнів в дереві цілей,  $G_i^d$  – конкретна мотиваційна ціль. Множина зв'язків між цілями може бути

представлена у вигляді множини дуг  $R = \{R(G_{id_1}^d, G_{id_2}^{d+1})\}$ . Цілі, що не мають підцілей, тобто такі  $G_{id_1}^d$ , для яких не існує зв'язків  $R(G_{id_1}^d, G_{id_2}^{d+1})$ , є конкретними задачами, виконання яких призводить до досягнення цілі. Позначимо множину задач як  $T = \{T_j\}_{j=1}^m$ , де  $j$  – ідентифікатор задачі,  $m$  – кількість завдань дитини.

Приклад дерева цілей з конкретними задачами наведено на рисунку 3.2.

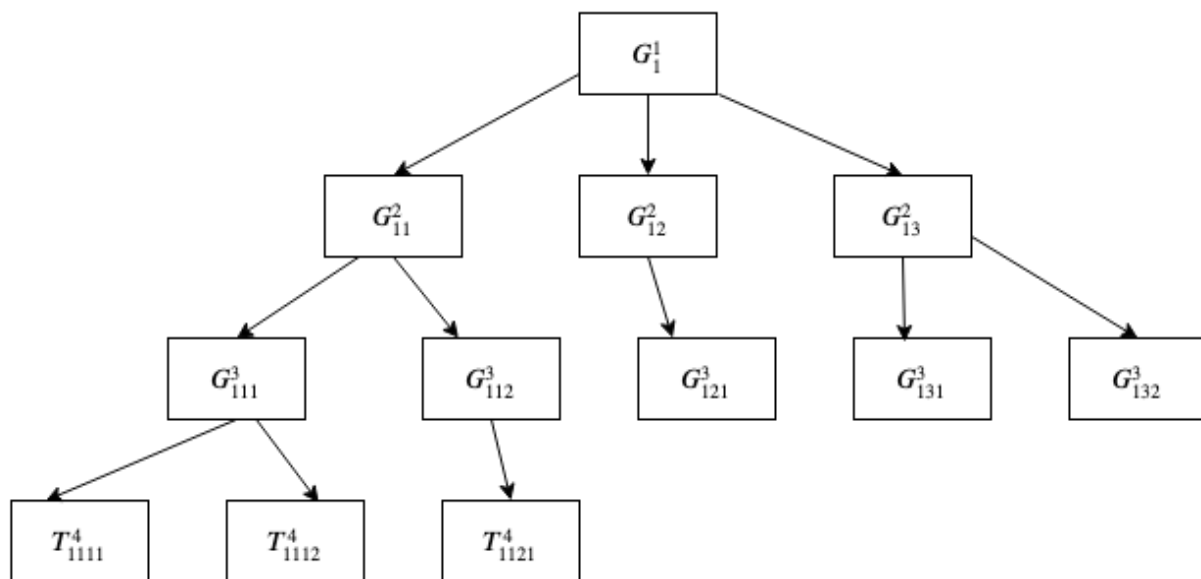


Рисунок 3.2 – Приклад дерева цілей

Кожна ціль характеризується важливістю  $W(G_{id}^d)$ . Будемо вважати, що важливість кореневої цілі  $W(G_1^1) = 1$ . Між підцілями будь-якої цілі  $G_{id}^d$  існує відношення не строгого порядку  $P(G_{id_1}^{d+1}, G_{id_2}^{d+1})$ , яке означає, що  $W(G_{id_1}^{d+1}) \geq W(G_{id_2}^{d+1})$ , тобто користувач може надати певній цілі більший пріоритет.

Сумарна важливість усіх підцілей  $\{G_{id}^{d+1}\}$  будь-якої цілі  $G_{id}^d$  може бути представлена наступною формулою:

$$W(G_{id}^d) = \sum_1^c W(G_{id_k}^{d+1}), \quad (3)$$

де  $c$  – кількість підцілей цілі  $G_{id}^d$ .

Кожній цілі  $G_{id_k}^{d+1}$ , яка є підціллю  $G_{id}^d$ , скористаємося відношенням порядку  $P$  і присвоїмо кожній цілі ранг  $r(G_{id_k}^{d+1}) \in [1, c]$ . Звідси важливість цілі  $G_{id_k}^{d+1}$  можна розрахувати наступною формулою:

$$W(G_{id_k}^{d+1}) = \frac{r(G_{id_k}^{d+1})}{\sum_{k=1}^c r(G_{id_k}^{d+1})}, \quad (4)$$

де  $c$  – кількість підцілей цілі  $G_{id}^d$ .

Таким чином, було побудовано математичну модель дерев цілей та визначено параметри, на основі яких буде обчислюватися важливість кожного завдання.

### 3.4 Розробка математичної моделі для вирішення оптимізаційної задачі про призначення

Система мотивування дітей повинна підтримувати функціонал складання оптимального розкладу завдань, які дитині необхідно виконати у певний період часу. Розклад повинен давати можливість дитині заробити якомога більшу кількість балів, тобто він має складатися на основі важливості завдання, розрахунок якої був наведений у попередньому розділі, а також враховувати обмеження, які накладаються на виконання цих завдань і на дні.

Задача  $Z^{opt}$  являє собою оптимізаційну задачу розподілу сукупності завдань на визначені дні, з урахуванням лімітів вільного часу у дитини по дням, важливості завдань з урахуванням їх належності до певної цілі з дерева цілей, дедлайнів завдань, необхідної періодичності їх виконання і т.п.

Нехай  $T = \{T_i\}_{i=1}^n$  – множина завдань;  $D = \{D_j\}_{j=1}^m$  – множина днів, за якими може бути розподілено завдання;  $c_i$  – важливість виконання завдання  $T_i$ , яка залежить від важливості цілі, до якої належить завдання, тобто  $c_i = W(G_{id}^d)$ , де  $G_{id}^d$  – кінцева ціль, яка розглядається як завдання  $T_i$ ;  $a_j$  – максимальне часове навантаження дитини в день  $D_j$  (ліміт вільного часу);  $b_i$  – часове обмеження на

виконання завдання  $T_i$ ;  $s(T_i)$  – час початку виконання завдання;  $\varphi_i^{min}$  – мінімальна періодичність виконання завдання  $T_i$ ;  $\varphi_i^{max}$  – максимальна періодичність виконання завдання  $T_i$ ;  $d(T_i)$  - дедлайн завдання.

Задача складання оптимального розкладу може бути промодельована як задача про призначення [13] (в таблиці 3.2 показано її табличне надання).

Таблиця 3.2 – Оптимізаційна модель задачі про призначення

Завдання та їх важливості	Дні для розкладу						Час виконання	Min частота	Max частота	Дедлайн
	D <sub>1</sub>	D <sub>2</sub>	...	D <sub>j</sub>	...	D <sub>m</sub>				
T <sub>1</sub> , C <sub>1</sub>	X <sub>11</sub>	X <sub>12</sub>	...	X <sub>1j</sub>	...	X <sub>1n</sub>	b <sub>1</sub>	$\varphi_1^{min}$	$\varphi_1^{max}$	d <sub>1</sub>
T <sub>2</sub> , C <sub>2</sub>	X <sub>21</sub>	X <sub>22</sub>	...	X <sub>2j</sub>	...	X <sub>2n</sub>	b <sub>2</sub>	$\varphi_2^{min}$	$\varphi_2^{max}$	d <sub>2</sub>
...	...	...	...	...	...	...	...	...	...	...
T <sub>i</sub> , C <sub>i</sub>	X <sub>i1</sub>	X <sub>i2</sub>	...	X <sub>ij</sub>	...	X <sub>in</sub>	b <sub>i</sub>	$\varphi_i^{min}$	$\varphi_i^{max}$	d <sub>3</sub>
...	...	...	...	...	...	...	...	...	...	...
T <sub>k</sub> , C <sub>n</sub>	X <sub>m1</sub>	X <sub>m2</sub>	...	X <sub>mj</sub>	...	X <sub>mn</sub>	b <sub>k</sub>	$\varphi_k^{min}$	$\varphi_k^{max}$	d <sub>k</sub>
Часові ліміти на виконання	a <sub>1</sub>	a <sub>2</sub>	...	a <sub>j</sub>	...	a <sub>m</sub>				

Введемо змінну  $X_{ij} = 1|0$  як ознаку того, чи буде призначено завдання  $T_i$  на день  $D_j$ . Тоді функція оптимізації повинна максимізувати важливість всього розкладу:

$$F = \sum_{i=1}^n \sum_{j=1}^m C_i * X_{ij} \rightarrow \max, \quad (5)$$

де  $C_{ij}$  – важливість виконання завдання  $T_i$ ;

$X_{ij}$  – ознака того, чи буде призначено завдання  $T_i$  на  $j$ -й день.

Складений розклад повинен дотримуватися часових лімітів на дні розкладу, враховувати дедлайн завдання, задовольняти вимогам мінімальної та максимальної періодичності.

Тобто під час складання розкладу мають виконуватися наступні умови:

$$\begin{aligned} \sum_{i=1}^n X_{ij} b_i &\leq a_j, \forall j = \overline{1, m}, \\ s(T_i) + b_i &\leq d(T_i), \\ \sum_{j=1}^m X_{ij} &\geq \varphi_i^{min}, \forall i = \overline{1, k}, \\ \sum_{j=1}^m X_{ij} &\leq \varphi_i^{max}, \forall i = \overline{1, k}, \end{aligned} \quad (6)$$

де  $X_{ij}$  – ознака того, чи буде призначено завдання на  $j$ -й день;

$b_i$  – тривалість виконання завдання;

$a_j$  – максимальне часове навантаження дитини в  $j$ -й день;

$s(T_i)$  – час початку виконання завдання;

$\varphi_i^{min}$  – мінімальна періодичність виконання завдання;

$\varphi_i^{max}$  – максимальна періодичність виконання завдання;

$d(T_i)$  – дедлайн завдання.

Розроблена оптимізаційна задача про призначення завдань в розклад дитини дозволить оптимальним чином сформулювати розклад дитини відповідно до мотиваційних цілей та врахувати важливі аспекти в навантаженні дитини. Для зберігання інформації з розробленої моделі необхідно буде спроектувати базу даних та розробити алгоритм з використанням жадібного підходу, який буде будувати оптимальний розклад завдань для виконання.

### 3.5 Розробка бази даних для зберігання математичної моделі

Для розробки бази даних, яка буде зберігати всю необхідну інформацію з математичної моделі необхідно визначити основні сутності системи, які описують концептуальну область програмної системи мотивування дітей (див. рис. 3.3).

Основною сутністю системи є дитина, яка виконує призначені їй завдання. Виконання завдань приносять дитині бали, які в свою чергу приносять нагороди. Кожне завдання зумовлює певна мета, і сукупність завдань для виконання формують розклад.

Для зберігання всієї необхідної інформації з математичної моделі була розроблена реляційна база даних [14], основними сутностями якої є батько, дитина, завдання, нагорода, розпорядок, призначення та день.

Розглянемо зв'язки, які існують між цими сутностями:

– у одного батька може бути декілька дітей, і в одній дитини може бути декілька батьків, тому між сутностями батько і дитина існує зв'язок багато до багатьох;

– одній дитині може бути призначено декілька завдань, і одне і те ж завдання може бути призначене декільком дітям, тому між завданням і дитиною існує зв'язок багато до багатьох;

– одній дитині може бути призначено декілька потенційних нагород, і одна і та ж сама нагорода може бути призначена декільком дітям, тому між нагородою і дитиною існує зв'язок багато до багатьох.

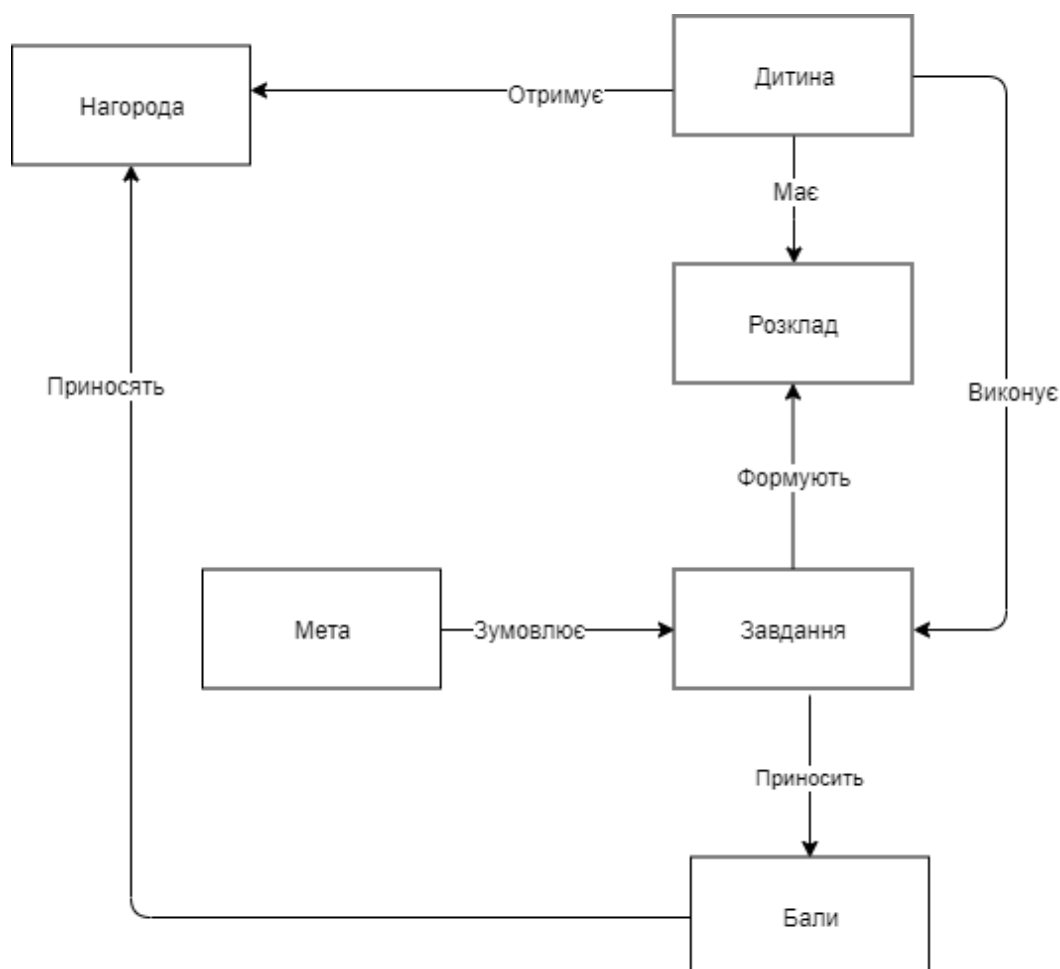


Рисунок 3.3 – Концептуальна модель системи

Розглянемо зв'язки, які існують між цими сутностями:

– у одного батька може бути декілька дітей, і в одній дитині може бути декілька батьків, тому між сутностями батько і дитина існує зв'язок багато до багатьох;

– одній дитині може бути призначено декілька завдань, і одне і те ж завдання може бути призначене декільком дітям, тому між завданням і дитиною існує зв'язок багато до багатьох;

– одній дитині може бути призначено декілька потенційних нагород, і одна і та ж сама нагорода може бути призначена декільком дітям, тому між нагородою і дитиною існує зв'язок багато до багатьох;

– один батько може створювати декілька завдань, але кожне завдання може бути створене лише одним батьком, тому між батьком і завданням існує зв'язок один до багатьох;

– одна дитина може мати декілька розпорядків, але один розпорядок може бути призначеним тільки одній дитині, тому між дитиною і розпорядком існує зв'язок один до багатьох;

– один розпорядок може включати в себе декілька призначень (пар днів і завдань), але одне призначення може належати тільки одному розкладу, тому між розпорядком і призначенням існує зв'язок один до багатьох;

– один день може бути пов'язаним із декількома призначеннями, але одне призначення може мати тільки один день, тому між днем і призначенням існує зв'язок один до багатьох.

З урахуванням даних зв'язків між сутностями була розроблена схема бази даних, яка зображена на рисунку 3.4.

Розглянемо більш детально основні таблиці бази даних. Таблиця `parent` призначена для збереження даних про батьків. Вона має наступні поля:

- `id` – ідентифікатор користувача, первинний ключ;
- `email` – адреса електронної пошти (логін);
- `password` – пароль;
- `name` – ім'я;

- surname – прізвище;
- device\_token – токен мобільного застосування (для отримання сповіщень);
- photo – ім'я файлу фотографії профілю (або зображення за замовчуванням).

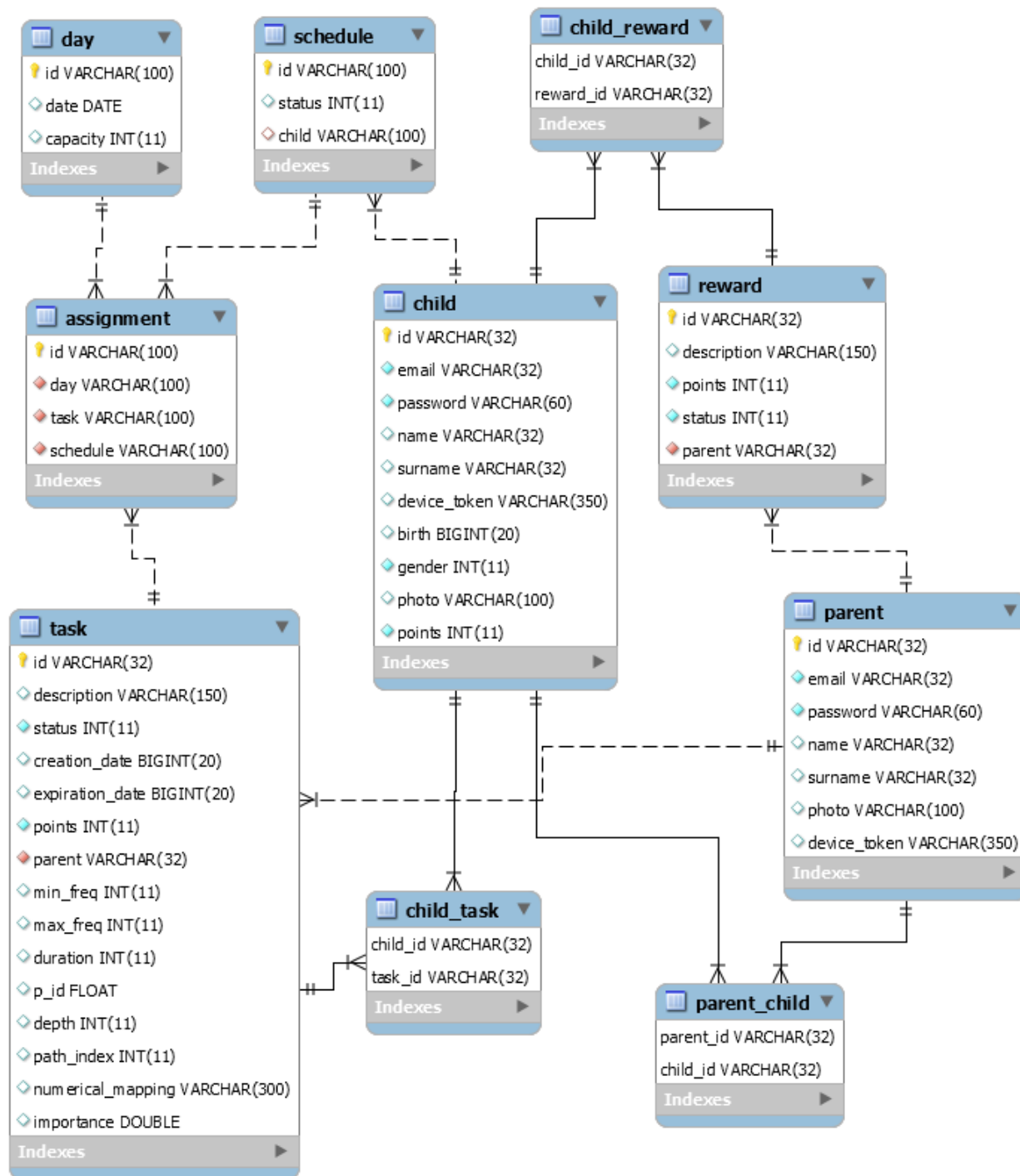


Рисунок 3.4 – Схема бази даних

Таблиця child зберігає дані про дітей і має наступні поля:

– id, email, name, surname, device\_token, photo – такі ж поля, як і в таблиці батьків;

- data\_of\_birth – дата народження;
- gender – стать;
- points – кількість отриманих балів за виконання завдань.

Таблиця task зберігає завдання і має наступні поля:

- id – ідентифікатор завдання, первинний ключ;
- description – опис завдання;
- status – статус завдання;
- creationDate – дата створення;
- expirationDate – дата, до якої необхідно виконати завдання;
- points – бали, які дитина отримує за його виконання;
- min\_freq – мінімальна частота виконання завдання на період часу;
- max\_freq – максимальна частота виконання завдання на період часу;
- parent – користувач, що створив завдання;
- duration – час, необхідний для виконання завдання;
- importance – важливість завдання;
- p\_id – ідентифікатор цілі, підціллю якої є завдання (ціль);
- depth – рівень завдання (цілі) в дереві цілей;
- path\_id – ідентифікатор шляху, до якого належить завдання (ціль);
- numerical\_mapping – представлення номеру завдання (цілі) в дереві цілей.

Таблиця reward зберігає нагороди і має наступні поля:

- id – ідентифікатор нагороди, первинний ключ;
- description – опис нагороди;
- status – статус нагороди;
- points – бали, необхідні для отримання нагороди;
- parent – користувач, що створив нагороду.

Таблиця складається із наступних полів:

- id – ідентифікатор дня, первинний ключ;

- date – дата дня;
- capacity – кількість годин, яку дитина може витратити на виконання завдань у цей день.

Таблиця schedule зберігає розпорядки і складається із таких полів:

- id – ідентифікатор розпорядку, первинний ключ;
- child – дитина, з якою пов'язаний даний розпорядок;
- status – статус розпорядка.

Таблиця assignment складається із наступних полів:

- id – ідентифікатор призначення, первинний ключ;
- day – день, на який призначено виконання завдання;
- task – завдання, яке повинне бути виконане;
- schedule – розпорядок, пов'язаний із даним днем і завданням.

### 3.6 Розробка алгоритму побудови дерева цілей

Система мотивування дітей повинна підтримувати механізм побудови дерев цілей, тобто користувач повинен мати змогу додавати в систему нові цілі та завдання, необхідні для їх досягнення.

На рисунку 3.5 наведено блок-схему алгоритму побудови дерева цілей.

Першим кроком при створенні дерева цілей є визначення головної цілі і встановлення її важливості. Після цього головну ціль необхідно розбивати на підцілі, до тих пір, поки для них не можна буде визначити конкретні завдання, необхідні для їх досягнення.

Після того як визначаються всі цілі певного рівня, розраховується їх важливість на основі пріоритетів, які їм надає користувач. Цілі можуть мати однаковий пріоритет.

Після створення всіх підцілей, можна створити конкретну задачу, вказавши всю необхідну інформацію (опис, час виконання, дедлайн або мінімальна та максимальна періодичність виконання завдання за період часу). На базі важливості цілі розраховуються важливості завдань.

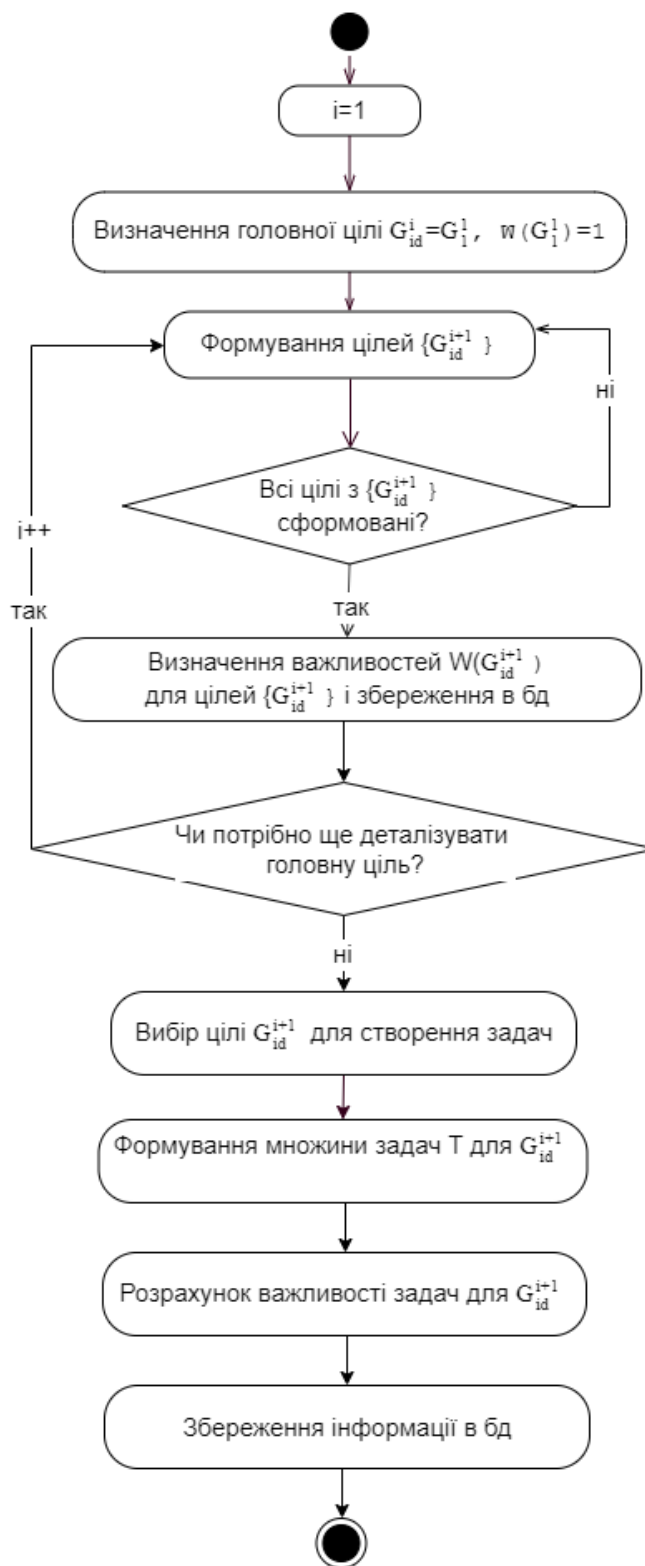


Рисунок 3.5 – Блок-схема алгоритму побудови дерева цілей

Таким чином було розроблено алгоритм побудови дерева цілей і визначення важливостей цілей та задач, які використовуються при генерації оптимального розкладу для дитини.

### 3.7 Розробка алгоритму вирішення оптимізаційної задачі про призначення

Система мотивування дітей повинна підтримувати можливість автоматичного складання розкладу завдань на тиждень для дитини.

Після проведення аналізу різних методів, за допомогою яких можна реалізувати цей функціонал, було обрано жадібний алгоритм, який забезпечує швидкість виконання, відносну простоту реалізації та непогану точність отриманих результатів. У якості критерія жадібності було обрано важливість завдання, яка розраховується на основі важливості цілей, до яких належить завдання, та часу, необхідного для його виконання.

Алгоритм побудови розкладу складається із наступних етапів:

- визначення множини днів для складання розкладу (батьки обирають початок розкладу і кількість днів, на які необхідно скласти розклад. За замовчуванням – тиждень);
- задання часових лімітів кожному дню – кількості годин, які дитина може витратити на виконання завдань;
- формування множини завдань (беруться всі завдання дитини з невиконаним статусом і дедлайном, який більший, ніж перший день розкладу);
- сортування завдань по важливості і дедлайну (якщо присутній);
- проходження по кожному із днів і призначення завдань для виконання у цей день.

Для призначення завдання на певний день розкладу мають виконуватися наступні умови:

- не прострочення дедлайну;
- не перевищення ліміту часу дня при призначенні даного завдання на день;
- виконання умов мінімальної та максимальної періодичності (завдання має виконуватися не рідше мінімальної періодичності і не частіше максимальної).

Детальна інформація про роботу алгоритму та логіку призначення завдань для виконання у певний день наведено на рисунку 3.6.



Рисунок 3.6 – Блок-схема алгоритму побудови розкладу завдань

Таким чином в атестаційній роботі були досліджені методи теорії прийняття рішень, які можуть застосуватися для підтримки мотивації в системі мотивації дітей. Були розроблені математичні моделі підтримки побудови дерев цілей та вирішення оптимізаційної задачі про призначення. Для вирішення оптимізаційної задачі було проведено дослідження алгоритмів її вирішення та обрано жадібний алгоритм, що забезпечують задовільну швидкість вирішення задачі та прийнятний розклад. Було розроблено алгоритм підтримки побудови дерева цілей. Для зберігання інформації з математичної моделі було спроектовано реляційну базу даних та реалізовано її за допомогою СУБД MySQL. Розроблені модель та алгоритми програмно реалізовані в складі мобільного додатку мотивування дітей на базі платформи Android.

## 4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Реалізована система являє собою мобільний додаток, який дозволяє батькам мотивувати своїх дітей за допомогою нагород, надає можливість управління цілями і автоматичного складання розкладу завдань на період часу. Ця система складається із серверної частини, яка взаємодіє з базою даних та мобільним клієнтом.

### 4.1 Вибір технологій для програмної реалізації

Сервер займається обробкою запитів від користувачів та взаємодіє з базою даних. Обмін даними між мобільним додатком і сервером відбувається по протоколам стеку TCP/IP, зокрема з використанням HTTP-запитів, при чому дані передаються у форматі JSON. Як засіб автоматизації роботи і керування залежностями проекту було використано Apache Maven.

Сервер був написаний на мові програмування Java. Шар доступу до даних реалізований за допомогою технології Hibernate, що дає можливість не турбуватися про те, яка база даних використовується у проекті.

Також для написання серверу була використана технологія Spring MVC: існують спеціальні контролери, які оброблюють запити від користувачів, сервіси, які виконують основну бізнес-логіку системи і шар доступу до даних.

Авторизація та автентифікація користувачів реалізована за допомогою Spring Security. Кожного разу, коли користувач успішно входить в додаток, генерується токен, який однозначно його ідентифікує. Даний токен передається у заголовок кожного запита, який робить користувач до сервера. Така реалізація дає можливість авторизації за допомогою клієнтів, що не підтримують механізму cookies і сесій, наприклад, клієнт під Android.

Мобільний клієнт був написаний з використанням технології Android SDK та мов програмування Java, Kotlin. Kotlin розроблений JetBrains і дозволяє подолати ряд недоліків, пов'язаних з Java. Kotlin більш зрозумілий, лаконічний і поставляється з набором інструментів і середовищ, які працюють з Java. Це значно зменшує кількість рядків коду і мінімізує кількість помилок. Kotlin

оснащений просунутим компілятором, який може перевіряти помилки під час компіляції і під час виконання. Kotlin є повністю сумісним з Java, забезпечує Null safety, розширення та інші можливості.

Для посилення запитів до сервера був використаний Retrofit2. Також при розробці мобільного клієнта були використані технології Butterknife для більш зручного способу встановлення зв'язку між UI елементами і програмним кодом, і Glide – для завантаження зображення на сервер/з серверу та кешування зображень.

#### 4.2 Опис фізичної моделі бази даних

Для збереження всієї інформації системи мотивання дітей була використана СКБД MySQL, яка є однією із найбільш поширених систем керування базами даних. Дана СКБД має наступні переваги:

- швидкість роботи;
- простота використання;
- підтримка мови запитів;
- підтримка тригерів, курсорів, збережених процедур
- можливості обробки. Кількість рядків в таблицях може досягати 50 млн;
- можливість одночасного підключення до бази багатьох користувачів.

MySQL призначена для роботи в мережі і може бути доступна через Internet. Таким чином, до сервера MySQL можна підключатися з будь-якої точки земної кулі через програму-клієнт, встановлену на комп'ютері користувача, але при цьому СУБД MySQL забезпечена розвиненою системою захисту від несанкціонованого доступу.

Для забезпечення додаткового захисту СКБД MySQL підтримує захищені з'єднання з використанням протоколу SSL. MySQL підтримує реляційні бази даних, тобто дані яких представлені у вигляді таблиць, розбитих на рядки і стовпці, на перетині яких знаходяться дані.

База даних для системи мотивування дітей складається з 10 таблиць. На рисунку 5.1 наведений фрагмент скрипту створення бази даних.

```

CREATE TABLE `parent` (
  `id` varchar(32) NOT NULL,
  `email` varchar(32) NOT NULL,
  `password` varchar(60) NOT NULL,
  `name` varchar(32) DEFAULT NULL,
  `surname` varchar(32) DEFAULT NULL,
  `photo` varchar(100) DEFAULT NULL,
  `device_token` varchar(350) DEFAULT NULL,
  PRIMARY KEY (`id`));
CREATE TABLE `task` (
  `id` varchar(32) NOT NULL,
  `description` varchar(150) DEFAULT NULL,
  `status` int(11) NOT NULL,
  `creation_date` bigint(20) DEFAULT NULL,
  `expiration_date` bigint(20) DEFAULT NULL,
  `points` int(11) NOT NULL DEFAULT '0',
  `parent` varchar(32) NOT NULL,
  `min_freq` int(11) DEFAULT '1',
  `max_freq` int(11) DEFAULT '1',
  `duration` int(11) DEFAULT '0',
  `p_id` float DEFAULT NULL,
  `depth` int(11) DEFAULT NULL,
  `path_index` int(11) DEFAULT NULL,
  `numerical_mapping` varchar(300) DEFAULT NULL,
  `importance` double DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `parId` (`parent`),
  CONSTRAINT `parId` FOREIGN KEY (`parent`) REFERENCES `parent` (`id`) ON DELETE
  CASCADE ON UPDATE CASCADE);

```

Рисунок 4.1 – Фрагменту скрипту створення бази даних

Таким чином, для системи мотивування дітей було розроблено базу даних з використанням MySQL, яка забезпечує компроміс між широкими функціональними можливостями та простотою використання.

### 4.3 Інтерфейс та функціонал системи

Коли користувач відкриває мобільний додаток на своєму пристрої, то перед ним з'являється вікно входу (див. рис. 4.2).

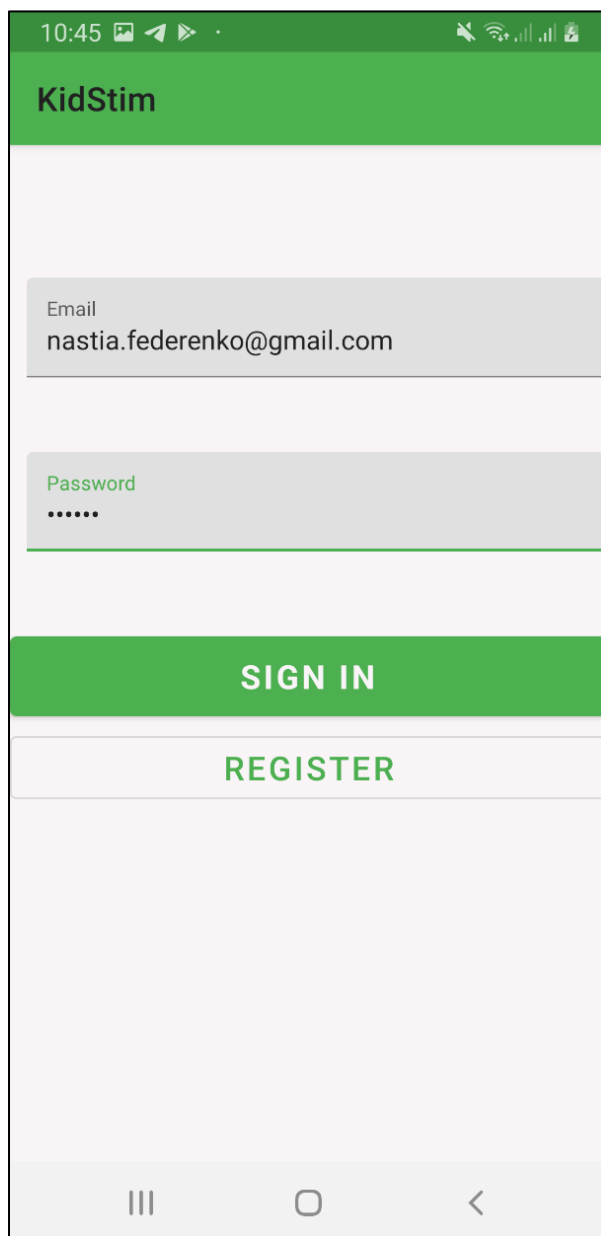


Рисунок 4.2 – Вікно входу в додаток

Щоб увійти в свій акаунт, користувачу необхідно ввести адресу своєї електронної пошти та пароль. Якщо введені значення електронної пошти не валідні, то відповідні повідомлення про помилки будуть показані під полями

введення. Ввівши коректні дані та натиснувши на кнопку Sign in користувач має змогу увійти до свого акаунту.

Якщо користувач, що входить в систему, належить до категорії “Батьки”, то після входу перед ним відкривається вікно, де він може побачити список своїх дітей (див. рис. 4.3). Відображаються фото дітей, імена та кількість балів, які вони отримали за виконання завдань.

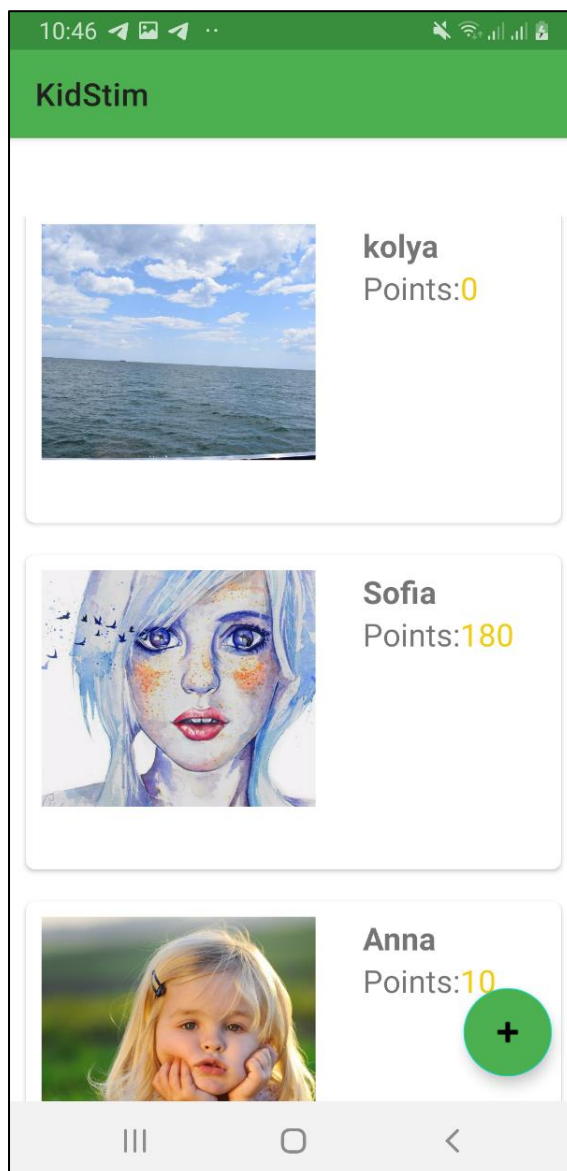
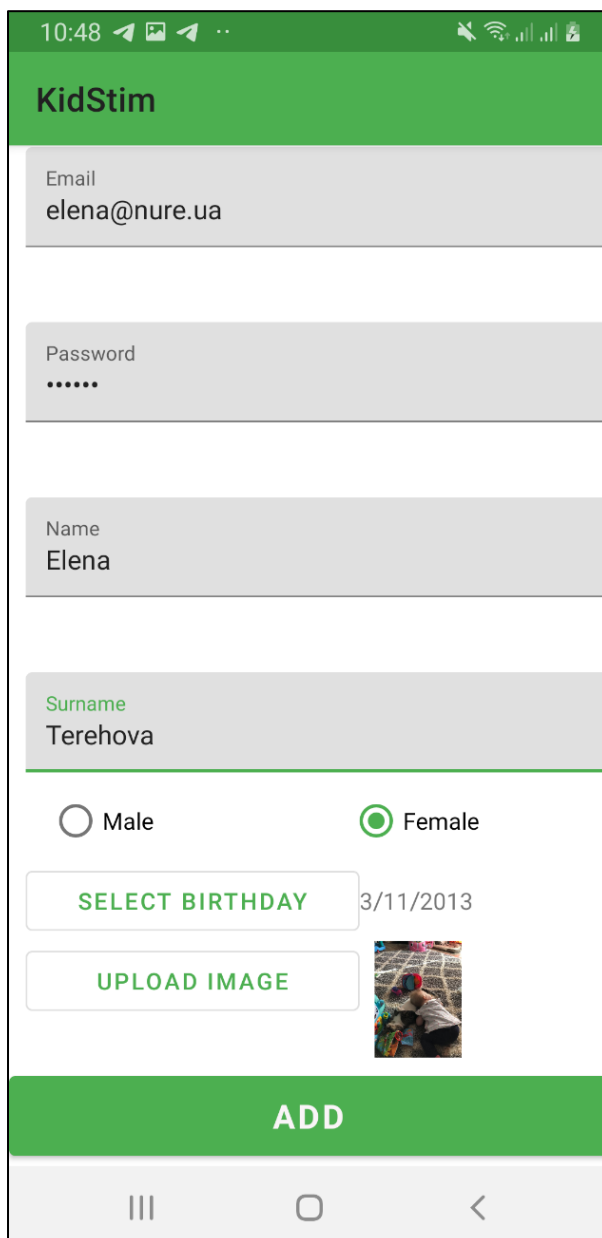


Рисунок 4.3 – Вікно зі списком дітей

Користувач має змогу додавати акаунт дитини в систему, натиснувши на плаваючу кнопку у вікні зі списком дітей. Після цього перед ним з'являється наступне вікно (див. рис. 4.4). Користувачу необхідно увести валідне значення

електронної пошти, пароль (не менше 6 символів), ім'я та прізвище дитини (не можуть бути порожніми). Користувач може обрати стать дитини, дату народження, натиснувши на **Select Birthday**, та завантажити фото дитини, натиснувши на **Upload Image**. Після того, як користувач натисне на кнопку **Add**, буде створений акаунт дитини і дитина матиме можливість авторизуватися у додатку.



10:48

**KidStim**

Email  
elena@nure.ua


Password  
.....

Name  
Elena

Surname  
Terehova

Male  Female

**SELECT BIRTHDAY** 3/11/2013

**UPLOAD IMAGE** 

**ADD**

Рисунок 4.4 – Вікно додавання дитини

Також користувач системи може користуватись навігаційним меню (див. рис. 4.5). Зверху даного меню відображається електронна адреса

користувача, його ім'я та фото профілю. Натиснувши на Log out, користувач має змогу вийти зі свого акаунту, повернувшись на вікно входу. Якщо користувач натискає на Children, перед ним відкриється вікно зі списком його дітей, яке уже було згадане вище. Якщо він натискає на Rewards, відкривається вікно зі списком усіх створених ним нагород.

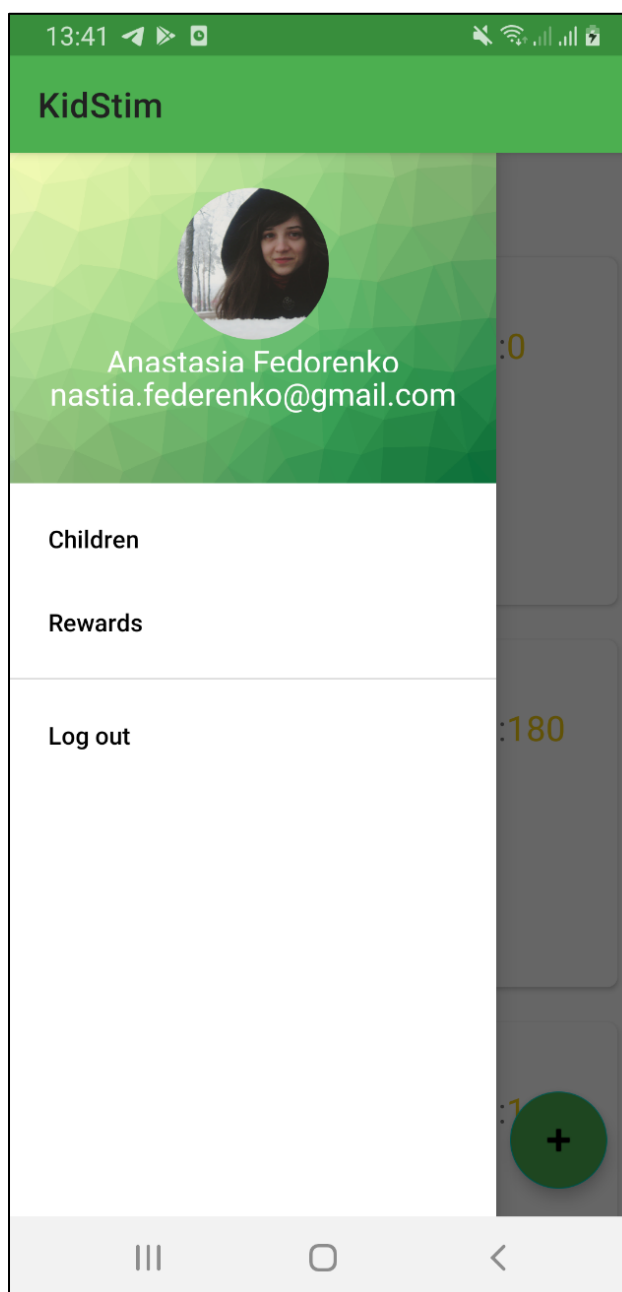


Рисунок 4.5 – Навігаційне меню

Якщо користувач натискає на будь-яку дитину із списку, відкривається вікно з детальною інформацією про дитину (див. рис. 4.6).

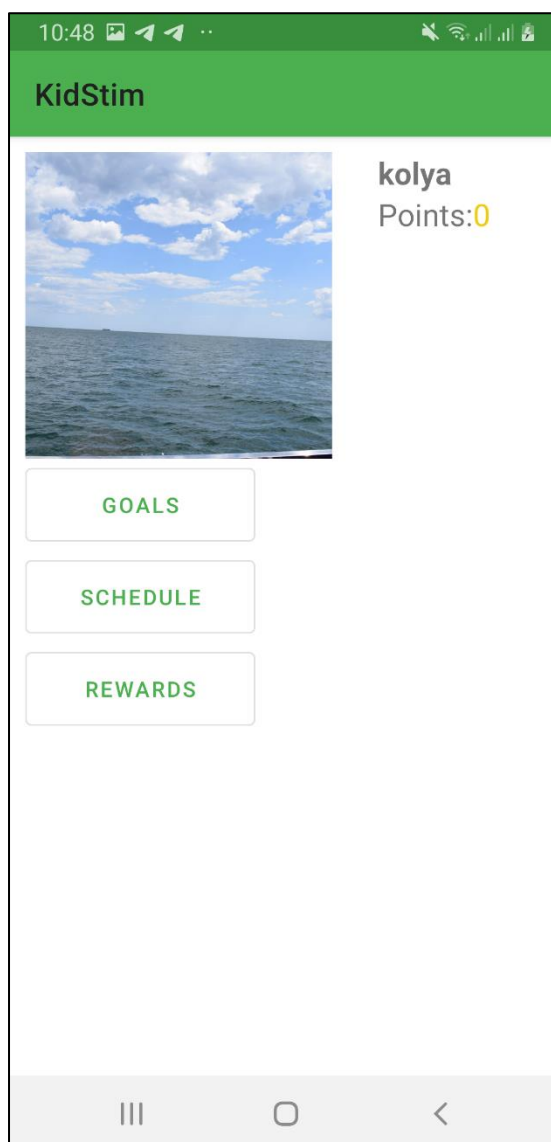


Рисунок 4.6 – Вікно з детальною інформацією про дитину

Коли користувач натискає на Rewards, відкривається вікно із нагородами конкретної дитини (див. рис. 4.7). Відображаються опис нагороди, фото дітей, які можуть її отримати, її статус та кількість балів, яка необхідна для її отримання.

Поряд з кожною нагородою у списку розташовані кнопки Delete та Edit, які дають змогу видаляти та редагувати нагороду відповідно. Після натискання на Delete, перед користувачем з'являється діалогове вікно для підтвердження його дії.

Користувач має змогу фільтрувати нагороди по їх статусу, обравши необхідний статус у випадаючому списку зверху екрана. За замовчуванням показуються всі нагороди.

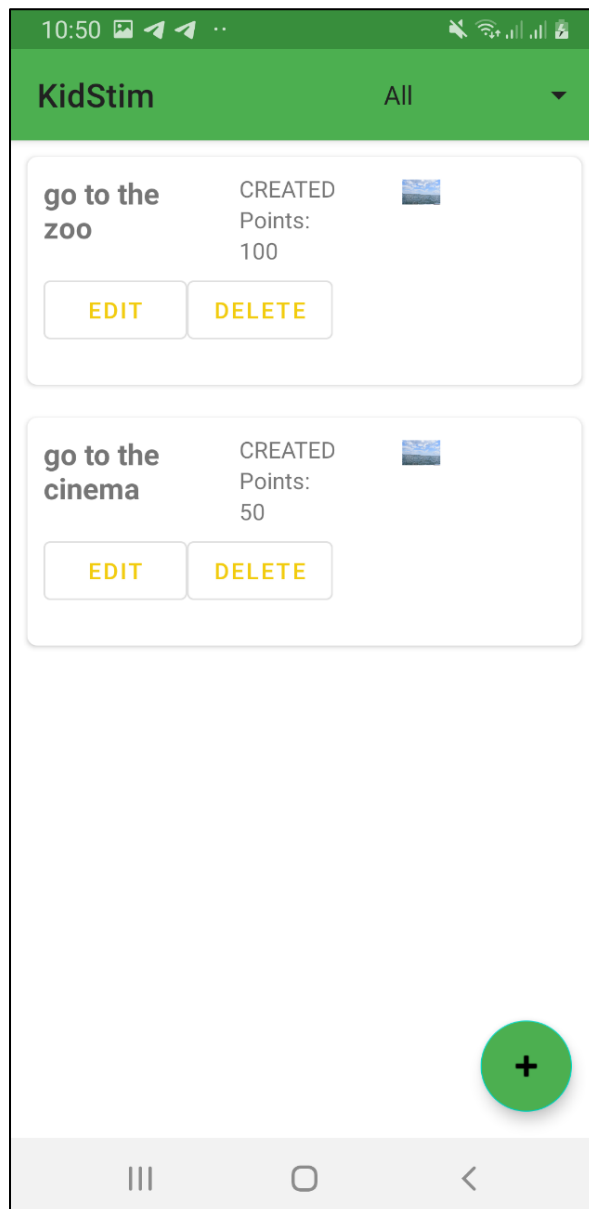


Рисунок 4.7 – Вікно нагород

Натиснувши на плаваючу кнопку у вікні нагород, користувачі категорії батьки мають можливість додати нову нагороду до списку. Вікно додавання нагороди має наступний вигляд (див. рис. 4.8).

Користувачу необхідно заповнити поля опису нагороди і кількості балів, які необхідні для її отримання. Користувач не може ввести не числове значення у поле кількості балів і залишити порожнім поле з описом нагороди.

Після натискання на кнопку Save, перед користувачем з'явиться діалогове вікно, у якому він може обрати дітей, які можуть отримати дану нагороду. Після

того як користувач обере дітей та натисне на ОК, нагороду буде успішно додано до списку нагород.

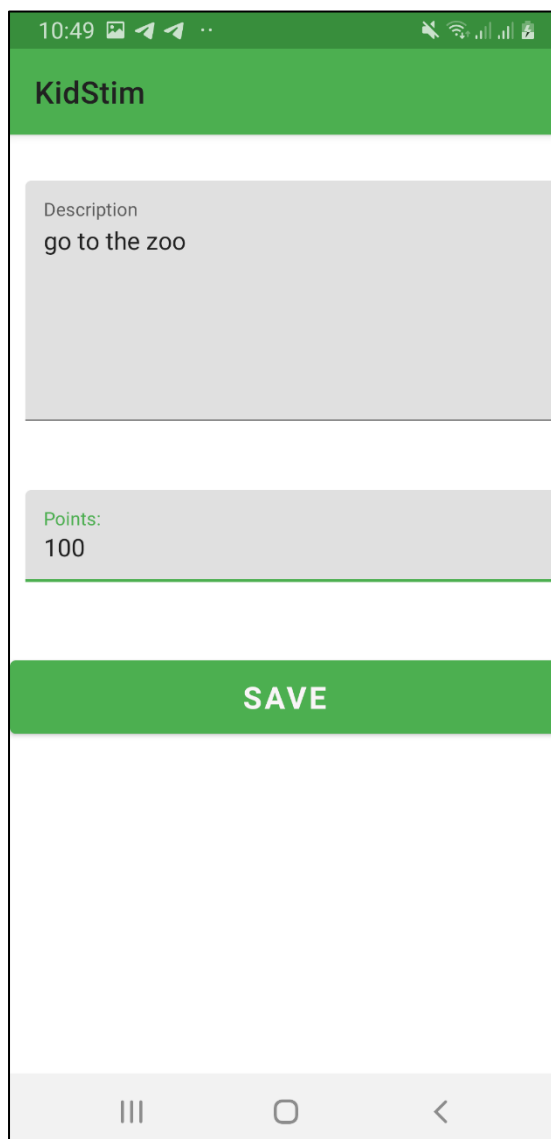


Рисунок 4.8 – Вікно додавання нагороди

Коли користувач натискає на Goals у вікні з детальною інформацією про дитину, відкривається вікно, у якому ієрархічно відображаються всі цілі, підцілі та завдання дитини (див. рис. 4.9).

Навпроти кожного завдання є чекбокс Completed, який показує чи виконане завдання. Якщо користувач відмічає завдання як виконане, дитині нараховуються бали за нього.

Користувач має змогу додати нову ціль натиснувши на Add goal, додати підціль свореній цілі натиснувши на Add subgoal та додати завдання натиснувши на Add task.

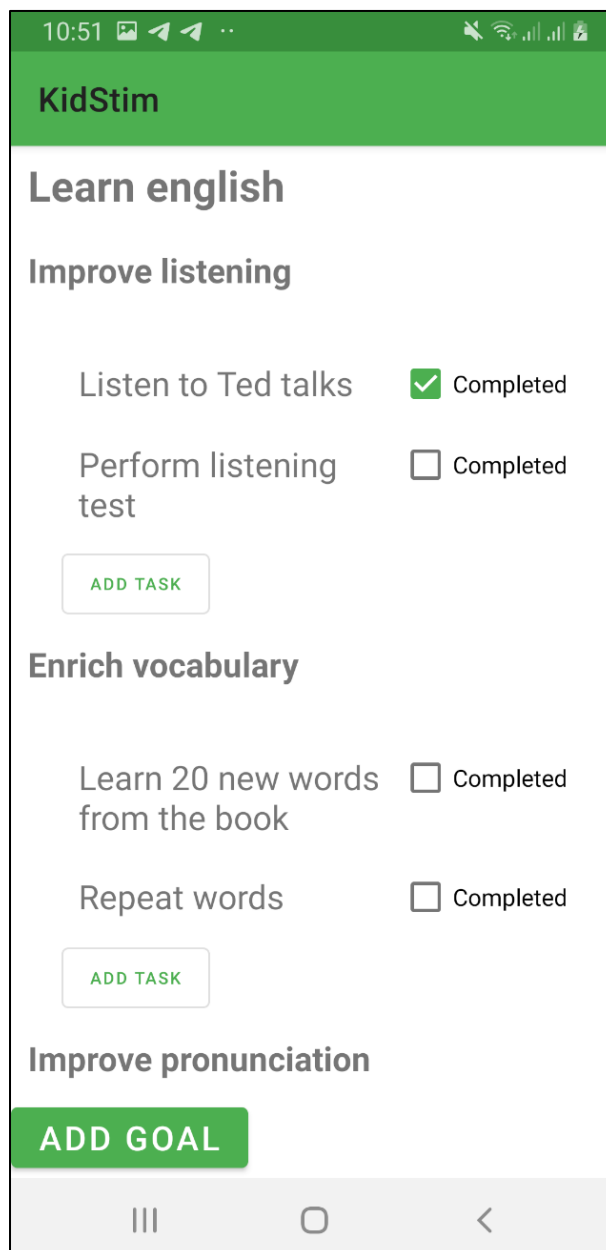


Рисунок 4.9 – Вікно цілей

Після додавання нової цілі відкривається вікно встановлення пріоритетів цілей (див. рис. 4.10), де відображаються всі цілі верхнього рівня. Користувач може змінювати порядок, у якому розташовані цілі, таким чином порівнюючи їх важливості між собою. Коли користувач натискає на Save, на основі визначених

користувачем пріорітетів обчислюються важливості цілей, їх підцілей та завдань. Коли користувач додає нову підціль до певної цілі, то у вікні встановлення пріорітетів відображаються всі підцілі батьківської цілі.

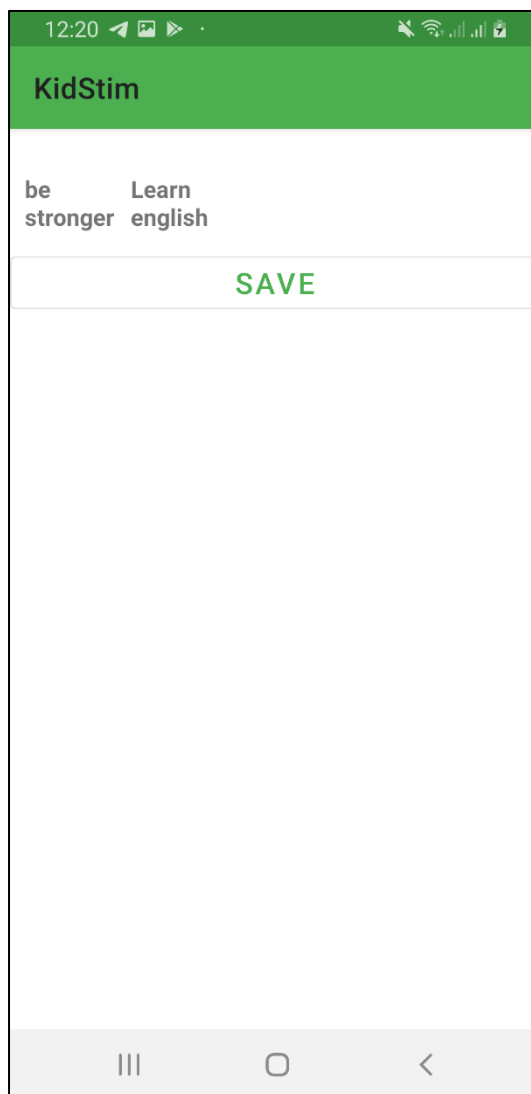
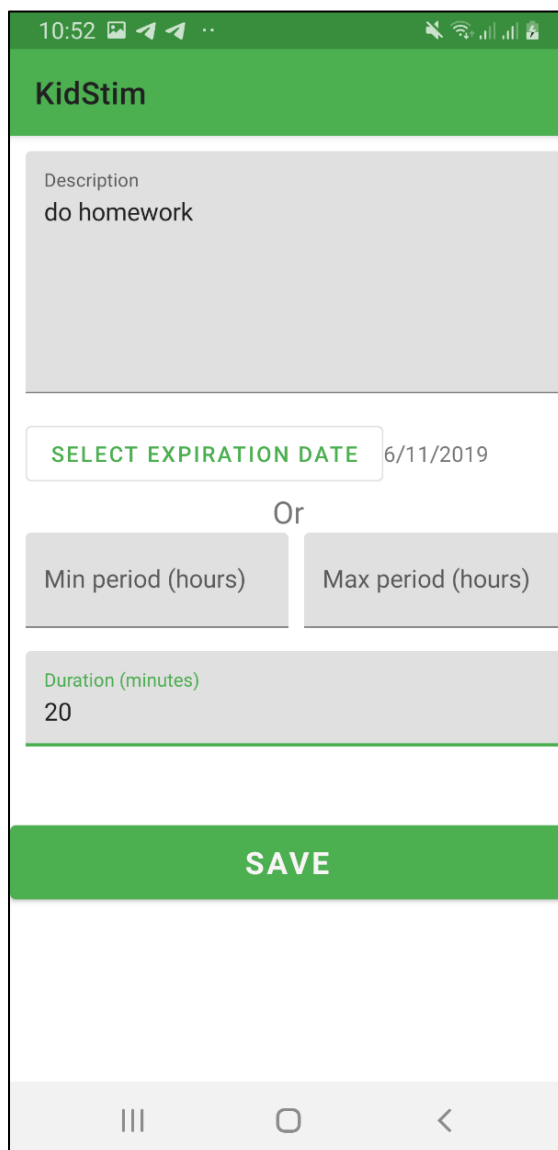


Рисунок 4.10 – Вікно встановлення пріорітетів цілей

Коли користувач натискає на Add task у вікні цілей, відкривається вікно додавання нового завдання, яке виглядає наступним чином (див. рис. 4.11). Користувачу необхідно заповнити такі поля, як опис завдання, дедлайн або мінімальна та максимальна періодичність виконання та час виконання завдання. Після натискання на кнопку Save, перед користувачем з'явиться діалогове вікно, у якому він може обрати дітей, яким необхідно виконати дане завдання. Якщо у

когось із дітей ще немає цілей, до яких відноситься створене завдання, то вони будуть створені автоматично.



The screenshot displays the 'KidStim' app interface for creating a new task. At the top, the status bar shows the time 10:52 and various system icons. The app title 'KidStim' is in a green header. Below it, a grey box contains the description 'do homework'. A date selection field is labeled 'SELECT EXPIRATION DATE' with the value '6/11/2019'. Below this, the word 'Or' is centered. Two grey boxes are provided for 'Min period (hours)' and 'Max period (hours)'. A 'Duration (minutes)' field shows the value '20'. A prominent green 'SAVE' button is at the bottom. The bottom of the screen shows the Android navigation bar with back, home, and recent apps icons.

Рисунок 4.11 – Вікно додавання нового завдання

Коли користувач натискає на кнопку Schedule у вікні детальної інформації про дитину, то перед ним відкривається вікно генерації нового розкладу, якщо у дитини немає активних розкладів (див. рис. 4.12).

У цьому вікні відображаються дати розкладу починаючи з наступного дня і поля для введення, у які необхідно ввести ліміт часу у годинах, який дитина може витратити на виконання завдань із розкладу.

Після натискання на кнопку Generate для дитини буде згенерований новий розклад з урахуванням зазначених часових обмежень на дні, дедлайнів завдань, їх мінімальної і максимальної періодичності та часу на їх виконання.

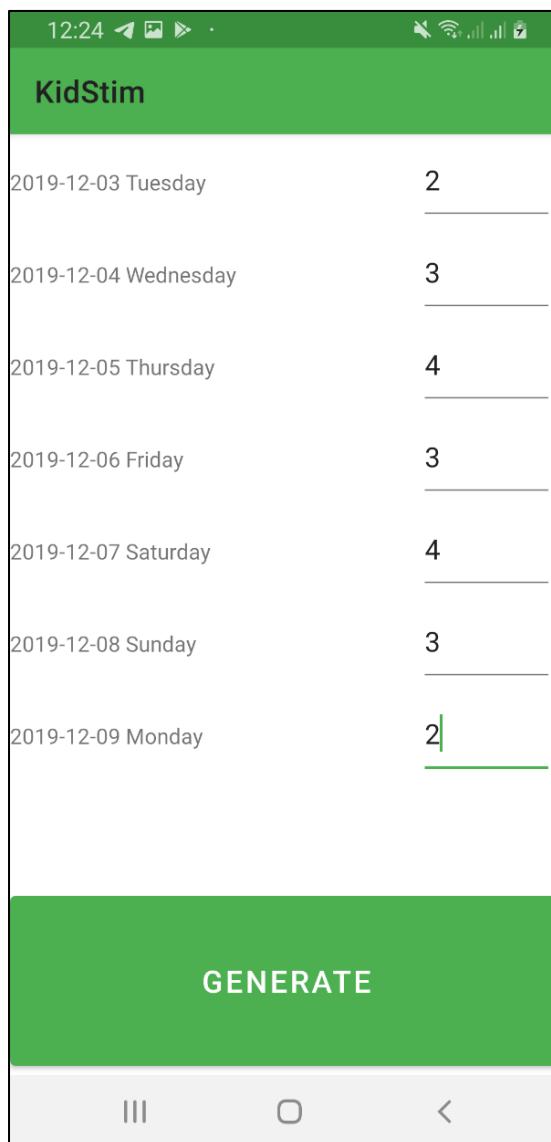


Рисунок 4.12 – Вікно генерації нового розкладу

Якщо у дитини існує активний розклад, то після натискання на кнопку Schedule у вікні детальної інформації про дитину, відкривається вікно із цим розкладом (див. рис. 4.13). Дане вікно відкривається також після генерації нового розкладу. У цьому вікні відображаються дні тижня з призначеними завданнями для виконання з урахуванням їх тривалості.

Коли користувач авторизується у системі під акаунтом дитини, то його навігаційне меню складається з таких елементів як Goals, Schedule, Rewards. Після натискання на Goals відображаються всі цілі та завдання дитини. Після натискання на Schedule відображається активний розклад дитини. Якщо розклад ще не створений батьками, то показується відповідне повідомлення.

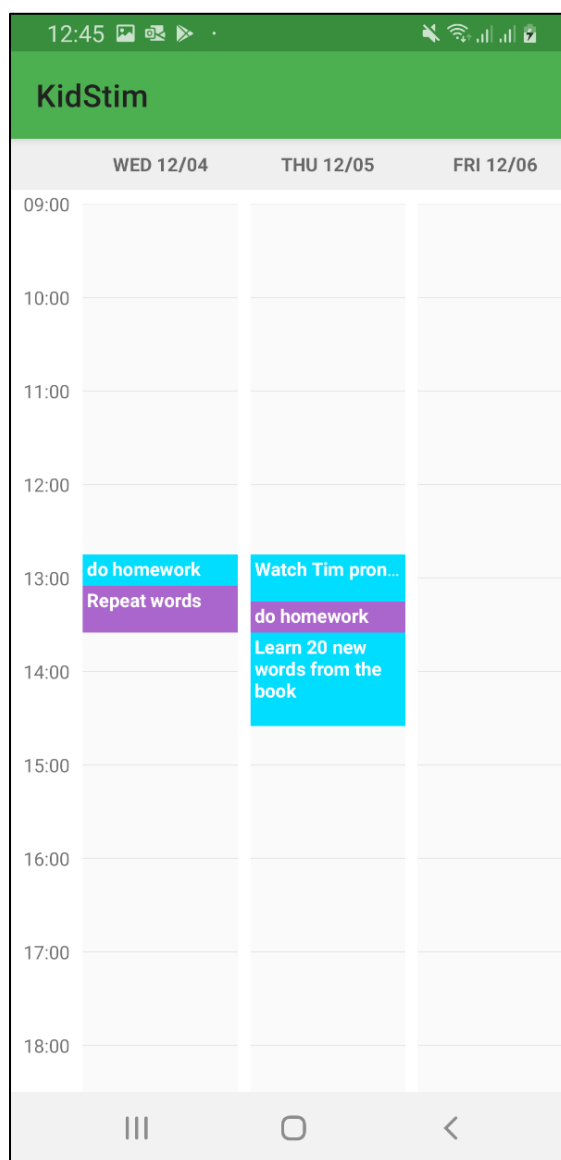


Рисунок 4.13 – Вікно розкладу дитини

Після натискання на Rewards, показується вікно з нагородами, які дитина може отримати (див. рис. 4.14). Вікно нагород має наступний вигляд: поряд з інформацією про нагороду є кнопка Request, що дозволяє зробити запит на отримання нагороди. Якщо дитина не має достатньої кількості балів для

отримання даної нагороди, то перед нею з'явиться спливаюче вікно з відповідним повідомленням як показано на рисунку.

Якщо дитина має достатньо балів, перед нею з'явиться діалогове вікно підтвердження того, що він/вона дійсно хоче зробити цей запит. Якщо дитина підтверджує, то батьки отримують нотифікацію про це, яка нагадує їм про те, що настав час надати обіцяну нагороду дитині.

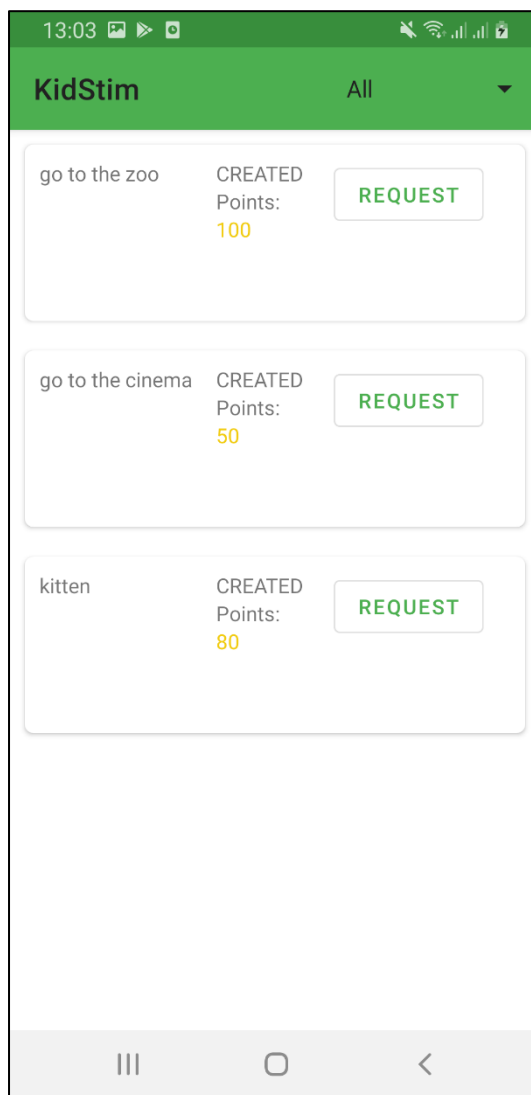


Рисунок 4.14 – Вікно нагород дитини

Користуватися додатком можуть лише зареєстровані користувачі. Щоб зареєструватися у системі, у вікні реєстрації необхідно ввести валідну адресу електронної пошти, пароль (не менше, ніж 6 символів), ім'я та прізвище (не мають бути порожніми).

## 5 АНАЛІЗ ДОСЛІДНИЦЬКОЇ ЕКСПЛУАТАЦІЇ ТА МОЖЛИВИХ ЗАСТОСУВАНЬ

### 5.1 Аналіз можливих застосувань

В ході виконання магістерської атестаційної роботи було спроектовано та розроблено програмну систему, яка дає можливість батькам мотивувати своїх дітей за допомогою таких методів як заохочення, гейміфікація, планування та встановлення цілей. Розроблена система дозволяє батькам складати списки завдань, за виконання яких вони можуть отримувати бали, а в подальшому обмінювати їх на нагороди.

Кожне створене батьками завдання належить до певної цілі, що сприяє мотивації дітей, адже вони бачать, що виконання завдання – це маленький крок до досягнення мети. Планування реалізовано у системі за допомогою функціоналу автоматичного складання розкладу завдань на певний період часу. Складений розклад враховує такі фактори як часові ліміти на кожний день розкладу, важливість завдань, дедлайни, мінімальну та максимальну періодичність виконання завдань та їх тривалість. Інтерфейс створеної програмної системи дозволяє зручно переглядати створені цілі, завдання та нагороди, а також редагувати їх, видаляти та фільтрувати.

Програмна система мотивування дітей перш за все призначена для батьків, які хочуть, щоб їхні діти були більш мотивованими та зацікавленими у виконанні домашніх обов'язків та навчанні, а також для дітей 6-13 років.

В подальшому можна розширити функціональні можливості системи, а саме додати можливість декільком батькам мати спільних дітей у додатку. Це сприяє зручності використання системи та залученню обох батьків до виховання та розвитку дитини, кращому розумінню її потреб.

Також корисним буде вдосконалення гейміфікації системи, тобто не просто додавання балів за виконане завдання, а віднімання їх при простроченні дедлайну чи додавання подвоєної кількості, якщо дитина виконує певні завдання частіше заданої мінімальної періодичності.

## 5.2 Опис тестування системи

Тестування програмного забезпечення – це процес технічного дослідження, який ставить за мету виявлення дефектів, підвищення рівня впевненості в якості продукту, надання інформації для прийняття рішення та запобігання майбутніх дефектів.

Метою тестування розробленого додатку для мотивування дітей є виявлення помилок в бізнес-логіці додатку, перевірка коректності роботи алгоритму генерації розкладу, та забезпечення коректного відображення функціоналу.

Під час процесу реалізації додатку для мотивування дітей було здійснено модульне та функціональне тестування системи [15]. Модульне тестування, яке також називають юніт-тестуванням, дозволяє перевірити на коректність роботи окремі модулі коду програми. Ідея модульного тестування полягає в написанні тестів для кожного нетривіального методу в коді, що дозволяє швидко перевірити чи не призвели зміни в коді до появи нових дефектів, полегшує виявлення та усунення таких дефектів. За допомогою модульного тестування був протестований алгоритм складання оптимального розкладу для дітей. На рисунку 5.1 наведено фрагмент коду, який перевіряє даний алгоритм на коректність. Перший тест перевіряє, чи не перевищує загальний час завдань, призначених на кожний день розкладу, часового ліміта цього дня. Другий тест перевіряє, що всі призначені на конкретний день завдання, які мають дедлайн, не будуть простроченими.

Функціональне тестування передбачає аналіз функціональних характеристик додатка та перевірку на невідповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і бізнес-вимог. Функціональне тестування фактично імітує фактичне використання системи. Функціональне тестування програмної системи мотивування дітей було здійснено за допомогою мануального тестування мобільного додатку, тобто перевірка роботи системи шляхом моделювання дій користувача.

```

@Test
void shouldNotExceedMaxCapacityOfDayWhenScheduleIsGenerated() {
    Map<Day, List<Task>> result = scheduleService.generateSchedule(CHILD_ID,
getMockCapacities());
    for (Map.Entry<Day, List<Task>> pair : result.entrySet()) {
        Day day = pair.getKey();
        int maxCapacityInMinutes = day.getMaxCapacity() * 60;
        assertTrue(day.getRealCapacity() <= maxCapacityInMinutes);
    }
}

@Test
void shouldNotSetTasksIntoDaysWhenTheyNotMeetDeadline() {
    Map<Day, List<Task>> result = scheduleService.generateSchedule(CHILD_ID,
getMockCapacities());
    for (Map.Entry<Day, List<Task>> pair : result.entrySet()) {
        Day day = pair.getKey();
        long date = day.getDate();
        List<Task> assignedTasks = pair.getValue();
        assignedTasks.stream().map(Task::getExpirationDate)
            .filter(Objects::nonNull)
            .forEach(x -> assertTrue(x >= date));
    }
}

```

Рисунок 5.1 – Фрагмент модульних тестів

В ході тестування було виявлено ряд незначних помилок, які успішно були виправлені в подальшому. Завдяки проведенню тестування було доведено функціонування розробленої системи до стабільного рівня, при якому можна не боятися появи неочікуваних помилок.

## ВИСНОВКИ

В атестаційній роботі магістра були досліджені основні методи теорії прийняття рішень і можливість їх застосування при розробці системи мотивування дітей. На основі цього аналізу були обрані методи, які будуть використовуватися під час програмної реалізації системи.

Було створено математичну модель побудови дерев цілей та оптимізаційної задачі про призначення, яка лежить в основі складання розкладу завдань для дітей. Для зберігання інформації з розробленої моделі було спроектовано реляційну базу даних та реалізовано її за допомогою СУБД MySQL. Для вирішення оптимізаційної задачі про призначення було реалізовано жадібний алгоритм, що забезпечують задовільну швидкість вирішення задачі та прийнятний розклад. Було розроблено алгоритм підтримки побудови дерева цілей. Розроблені модель та алгоритми програмно реалізовані в складі мобільного додатку мотивування дітей на базі платформи Android.

Була розроблена UML-модель для візуалізації функціоналу системи з точки зору різних категорій користувачів. В ході розробки було проведено аналіз вимог до продукту та проектування архітектури системи в цілому. Серед технологій для розробки системи було обрано Spring MVC, Hibernate, Retrofit2, JavaRx, Kotlin, Android SDK. Мобільний додаток дозволяє підтримувати мотивування дітей за допомогою нагород, підтримує функціонал формування цілей та складання оптимального розкладу для дітей. Додаток дає можливість батькам визначати пріоритети цілей, отримувати сповіщення про бажання дітей отримати нагороду та підтримує функціонал управління списками завдань та нагород.

За результатами роботи опубліковані тези на Міжнародному молодіжному форумі та зроблена презентація. Написана та опублікована стаття «Дослідження методів прийняття рішень для підтримки мотивації в системі мотивації дітей» в матеріалах IV міжнародної науково-практичної конференції в Ліверпулі «Scientific Achievements of Modern Society».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Трейсі Б., Стратцел Д. Наука мотивації. Мінськ: Попуррі, 2018. 240 с.
2. Ахмадуллін Ш. Мотивація дітей. Як мотивувати дитину вчитися. Москва: Білінгва, 2016. 80 с.
3. Бородачов С. Теорія прийняття рішень. Москва: Флінта, 2018. 125 с.
4. Шилдт Г. Java. Повне керівництво. Київ: Діалектика, 2016. 1488 с.
5. Хемраджані А. Гнучка розробка застосувань на Java з використанням Spring, Hibernate і Eclipse. Москва: Вільямс, 2008. 319 с.
6. Material design guidelines. URL: <https://material.io/guidelines/> (дата звернення: 13.10.2019).
7. Буч Г. Мова UML. Керівництво користувача. Москва: ДМК Прес, 2006. 248 с.
8. Ларичев О. Теорія і методи прийняття рішень. Москва: Логос, 2002. 392 с.
9. Мазурова О.О. Метод автоматизированной поддержки формирования множества критериев для задач принятия решений.// Вісн. Національного Технічного університету «ХПІ». Тематичний збірник наукових праць «Нові рішення у сучасних технологіях». 2001. №15. С.36-42.
10. Таха Хемді. Дослідження операцій. Москва: Вільямс, 2016. 912 с.
11. Симплекс-метод. Wikipedia, the free encyclopedia. URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4> (дата звернення: 21.10.2019).
12. Окулов С. Динамічне програмування. Москва: Біном, 2014. 296 с.
13. Кормен Т., Лейзерсон Ч., Рівест Р., Штайн К. Алгоритми. Побудова і аналіз. Москва: Вільямс, 2016. 1328 с.
14. Пирогов В. Інформаційні системи і БД: організація і проектування. Санкт-Петербург: БХВ-Петербург, 2009. 528 с.
15. Лі Коупленд. A Practitioner's Guide to Software Test Design. Бостон: Artech House Publishers, 2010. 294 с.