

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Інформаційна система обліку медичної інформації для прийому ліків
Назва теми


КВРІСТ 200178 20.01.01 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва


Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент IV курсу, група ICT-20-1  Г. О. Станіславська
Підпис Ініціали, прізвище

Керівник  С.М. Лисенко
Підпис, дата Ініціали, прізвище

Нормоконтролер  І.О. Засорнова
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем  Т.О. Говорущенко
Підпис Ініціали, прізвище

«31» травня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Станіславській Галині Олегівні

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система обліку медичної інформації для прийому ліків

Керівник проекту (роботи) Лисенко С.М., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2024 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Актуальність використання інформаційних систем в медичній сфері

Проектування інформаційної системи обліку медичної інформації для прийому ліків

Програмна реалізація інформаційної системи обліку медичної інформації для прийому ліків


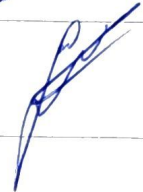
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Проектування архітектури програмного забезпечення інформаційної системи

UML діаграми

Програмна реалізація інформаційної системи

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.О., доцент кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	01.04.2024	виконано
5	Робота над розділом 3 – проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	26.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент



Г.О. Станіславська

Підпис

Ініціали, прізвище

Керівник роботи



Підпис

С.М. Лисенко

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система обліку медичної інформації для прийому ліків».

Автор роботи: Станіславська Галина Олегівна.

Керівник роботи: Лисенко Сергій Миколайович.

Пояснювальна записка: 62 с., 26 рис., 0 табл., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІКУ МЕДИЧНОЇ ІНФОРМАЦІЇ, UML,
АРХІТЕКТУРА ПЗ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Метою дипломної роботи є розроблення інформаційної системи обліку медичної інформації для прийому ліків.

Об'єктом дослідження є облік медичної інформації для прийому ліків за допомогою інформаційних систем.

Предметом дослідження є інформаційної системи обліку медичної інформації для прийому ліків.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.







30.05.2024

Підпис студента Дата

ЗМІСТ

ВСТУП	4
1 АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В МЕДИЧНІЙ СФЕРІ	6
1.1 Актуальність застосування інформаційних систем в медичній сфері для обліку прийому ліків пацієнтами.....	6
1.2 Аналіз наявного програмного забезпечення предметної області	8
1.4 Висновки	20
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ МЕДИЧНОЇ ІНФОРМАЦІЇ ДЛЯ ПРИЙОМУ ЛІКІВ	22
2.1 Розробка користувальницьких вимог. Функційні вимоги	22
2.2.1 Реєстрація пацієнтів	23
2.1.2 Ведення медичної інформації	24
2.1.3 Призначення лікування.....	25
2.1.4 Сповіщення про прийом ліків.....	26
2.1.5 Ведення реєстру ліків	27
2.1.6 Перевірка взаємодії ліків.....	28
2.1.7 Стеження за вживанням ліків	29
2.1.8 Генерація звітів і аналітика	29
2.2 Застосування Android SDK для розроблення програмного забезпечення інформаційної системи	30
2.3 Компоненти Android для програмного забезпечення інформаційної системи	32
2.4 Застосування шаблон проектування MVC (Model View Controller) в Android.....	38
2.5 Макети в дизайні інтерфейсу Android.....	39
2.6 Використання елементів інтерфейсу в XML-файлі.....	42
2.6 Висновок	43

КВРІСТ 200178 20.01.01 ПЗ								
Зм.	Лрк.	Нодокум.	Підпис	Дата	Інформаційна система обліку медичної інформації для прийому ліків Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Станіславська Г.				у		
Перевір.		Лисенко С.М.					2	62
Н.контр.		Засорнов І. О.				ХІІУ ІСТ-20-1		
Затвер.		Говорушченко Г.О.						

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ МЕДИЧНОЇ ІНФОРМАЦІЇ ДЛЯ ПРИЙОМУ ЛІКІВ.....	44
3.1 Проєктування архітектури програмного забезпечення інформаційної системи	44
3.2 UML діаграми	47
3.1.1 Створення діаграми варіантів	50
3.1.2 Створення діаграми класів	51
3.1.3 Створення діаграми послідовності.....	55
3.1.4 Створення діаграми активності	56
3.1.5 Створення діаграми станів	58
3.6. Особливості програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків.....	59
3.6. Висновки	64
ВИСНОВКИ	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	66
ДОДАТОК А Копія креслення «Проєктування архітектури програмного забезпечення інформаційної системи»	73
ДОДАТОК Б Копія креслення «UML діаграми».....	74
ДОДАТОК В Копія креслення «Програмна реалізація інформаційної системи»	75

ВСТУП

Актуальність застосування інформаційних систем в медичній сфері є незаперечною і постійно зростає з кожним роком. Це пов'язано з рядом ключових факторів та переваг, які вони пропонують як медичним працівникам, так і пацієнтам.

Інформаційні системи дозволяють лікарям отримувати швидкий доступ до медичних записів пацієнтів, історії хвороб, результатів аналізів тощо, що сприяє прийняттю обґрунтованих клінічних рішень.

Завдяки автоматизації обробки та зберігання медичних даних зменшується ризик помилок, пов'язаних з ручним введенням даних, та поліпшується загальна ефективність управління інформацією.

Інформаційні системи допомагають медичним установам оптимізувати робочі процеси, зокрема, через автоматизацію рутинних задач, таких як запис на прийом, ведення електронних медичних карток, управління запасами медикаментів.

Цифрові технології, такі як телемедицина та онлайн-консультації, розширюють доступ до медичних послуг, особливо для людей у віддалених або сільських районах.

Великі дані та штучний інтелект відкривають нові можливості для досліджень у медицині, дозволяючи аналізувати обсяги інформації для виявлення тенденцій, паттернів і нових методів лікування.

Сучасні інформаційні системи включають передові технології шифрування та захисту даних, що забезпечує високий рівень конфіденційності пацієнтської інформації.

Хоча початкові інвестиції в інформаційні системи можуть бути значними, довгостроково вони сприяють зниженню витрат завдяки зменшенню кількості помилок, оптимізації використання ресурсів та підвищенню загальної ефективності.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, актуальність інформаційних систем у медичній сфері обумовлена їх спроможністю вирішувати ряд викликів сучасної медицини, від підвищення якості лікування до забезпечення безпеки пацієнтських даних.

					КвРІСТ 200178 20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В МЕДИЧНІЙ СФЕРІ

1.1 Актуальність застосування інформаційних систем в медичній сфері для обліку прийому ліків пацієнтами

Актуальність використання інформаційних систем в медичній сфері для обліку прийому ліків пацієнтами є високою, особливо у контексті забезпечення ефективності лікування та безпеки пацієнтів.

Розглянемо основні ключові аспекти актуальності застосування інформаційних систем в медичній сфері для обліку прийому ліків пацієнтами:

1. Точність дотримання призначень лікаря.
2. Профілактика помилок у медикації.
3. Підвищення дотримання медичних рекомендацій безпосередньо пацієнтами.
4. Аналіз ефективності лікування.
5. Інтеграція з іншими медичними системами.
6. Оптимізація співпраці між лікарями та пацієнтами.

Інформаційні системи дозволяють точно фіксувати призначення лікаря та відстежувати, чи дотримується пацієнт режиму прийому ліків. Це критично важливо для забезпечення ефективності лікування, особливо при хронічних захворюваннях та комплексній терапії.

Автоматизовані системи можуть попереджати про можливі взаємодії між ліками, неправильну дозування або інші помилки, які можуть стати причиною серйозних наслідків для здоров'я пацієнта.

Інформаційні системи можуть включати нагадування для пацієнтів про необхідність прийому ліків, що сприяє підвищенню рівня дотримання медичних рекомендацій.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 6
Зм.	Арк.	№ докум.	Підпис	Дата		

Збір даних про прийом ліків дозволяє лікарям аналізувати ефективність лікування, коригувати терапію в реальному часі та приймати більш обґрунтовані рішення щодо подальшого лікування.

Інформаційні системи для обліку прийому ліків можуть інтегруватися з електронними медичними записами, системами управління запасами ліків в аптеках та іншими платформами, забезпечуючи комплексний підхід до лікування пацієнта.

Використання захищених інформаційних систем дозволяє забезпечити конфіденційність медичної інформації пацієнтів та відповідність нормам захисту персональних даних.

Цифрові платформи можуть слугувати засобом для підтримки безперервного діалогу між лікарем та пацієнтом, сприяючи кращому розумінню та виконанню медичних рекомендацій.

У сучасному світі, де технології набувають все більшого значення в усіх аспектах життя, інтеграція інформаційних систем в медичну практику для обліку прийому ліків є важливим кроком до підвищення ефективності та безпеки медичного обслуговування.

Точність дотримання призначень лікаря засобами програмного забезпечення є ключовим фактором, який може значно підвищити ефективність лікування та безпеку пацієнтів. Програмне забезпечення та мобільні додатки для управління медичними призначеннями надають пацієнтам інструменти для кращого контролю над їхнім лікуванням, дозволяючи точно дотримуватися рекомендацій лікарів.

Програмне забезпечення може автоматично нагадувати пацієнтам про необхідність прийняти ліки у певний час, що значно знижує ризик пропуску дози.

Пацієнти мають доступ до детальної інформації про свої ліки, включно з дозуванням та часом прийому, що допомагає уникнути помилок.

Пацієнти можуть вести записи про прийняті дози, що дозволяє лікарям аналізувати дотримання режиму лікування та його ефективність.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

Деяке програмне забезпечення аналізують призначення на предмет потенційної небезпечної взаємодії між різними ліками, що може підвищити безпеку пацієнта.

Програмне забезпечення може адаптувати нагадування та інструкції з урахуванням індивідуальних потреб пацієнта, включно з дієтами, алергіями та іншими особливостями.

Деяке програмне забезпечення дозволяють лікарям моніторити дотримання пацієнтом призначень в режимі реального часу, що поліпшує комунікацію між лікарем та пацієнтом та дозволяє швидко коригувати лікування при необхідності.

Багато програмного забезпечення включає освітні матеріали про здоров'я та лікування, допомагаючи пацієнтам краще розуміти важливість дотримання медичних призначень.

Застосування програмного забезпечення для дотримання призначень лікаря є важливим елементом сучасної медичної практики, що сприяє підвищенню якості та безпеки лікування, а також забезпечує краще взаєморозуміння і співпрацю між лікарями та пацієнтами.

1.2 Аналіз наявного програмного забезпечення предметної області

Управління прийомами ліками, включаючи запам'ятовування, коли і як їх приймати, може бути складним завданням, особливо для тих, хто жонглює багатьма обов'язками та діяльністю. Однак якщо забути прийняти ліки згідно з інструкцією, це може мати негативні наслідки, навіть небезпечні для життя.

Є багато доступних додатків, які надійно нагадуватимуть користувачам вчасно прийняти ліки. Тому, щоб потрапити до нашого списку найкращих програм для нагадування про прийом ліків, ми шукали програмне забезпечення з високим рейтингом, широко доступні та багатофункціональні.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

Було проаналізовано додатки у магазинах Apple і Google Play, щоб переконатися, яке ПЗ для нагадування про прийом ліків є зручним, надійним та не має помилок.

MediSafe – це проста, зручна мобільна програма, яка пропонує низку функцій, які спрощують керування рецептами. Додаток описується як «віртуальний дот», який надсилає користувачам персоналізовані нагадування приймати ліки згідно з інструкціями. Сповіщення також містять попередження про взаємодію з ліками, щоб користувачі знали, чи не слід їм приймати ліки з певними ліками, продуктами харчування чи напоями (рис. 1.1).



Рисунок 2.1 – Інтерфейс MediSafe

Додаток також ідеально підходить для опікунів. Існує функція утриманців, яка дозволяє користувачам додавати рецепти для кількох людей, наприклад дітей або пацієнтів похилого віку. Користувачі також можуть використовувати додаток, щоб надіслати сповіщення опікуну, якщо вони забули прийняти ліки.

MediSafe також допомагає користувачам бути впевненими, що ліки у них не закінчаться, завдяки нагадуванням про поповнення. Користувачі можуть імпортувати свої рецепти з великих аптечних мереж, таких як CVS, Rite–Aid або Walgreens, або додавати свої ліки (і безрецептурні вітаміни та добавки) вручну.

Переваги MediSafe:

					КВРІСТ 200178 20.01.01 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

1. MediSafe надає користувачам доступ до купонів і знижок у тисячах аптек по всій країні завдяки партнерству з GoodRx, допомагаючи користувачам знаходити найнижчі ціни на свої рецепти.

2. Надаючи попередження про взаємодію з ліками, MediSafe допомагає убезпечити користувачів, гарантуючи, що вони випадково не комбінують ліки з іншими ліками чи речовинами, які негативно взаємодіють або знижують ефективність ліків.

3. Обліковий запис MediSafe автоматично зберігає дані користувачів на безпечному сервері. Тому, якщо користувач отримує новий пристрій, все, що йому потрібно зробити, це завантажити програму та увійти у свій обліковий запис, і вся його інформація та нагадування автоматично перемістяться на новий пристрій.

4. Користувачі можуть бути впевнені, що їхня інформація зберігається конфіденційною за допомогою MediSafe. Додаток відповідає HIPAA та використовує 256-бітне шифрування для захисту конфіденційності користувачів.

Недоліки MediSafe:

- Одним із недоліків програмне забезпечення MediSafe є те, що розробники надають медичним компаніям анонімні зведені дані користувачів. Хоча ці дані є анонімними та можуть бути корисними для розробки кращих методів лікування та систем охорони здоров'я, деяким користувачам може не сподобатися той факт, що їхні дані збираються та поширюються.

- На деяких пристроях нагадування про сповіщення MediSafe вимикаються, коли пристрій перебуває в беззвучному режимі, тому користувачі повинні залишити звук увімкненим, щоб отримати нагадування.

MangoHealth, який доступний як для пристроїв iOS, так і для пристроїв Android, – це безкоштовна програма, розроблена, щоб зробити лікування ліками та охороною здоров'я цікавим. Додаток допомагає користувачам створювати щоденний розклад, який включає нагадування про прийом ліків, вимірювання артеріального тиску та рівня глюкози в крові, засоби відстеження ваги та активності, а також персоналізовані нагадування про інші здорові звички.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

Користувачі можуть за потреби встановлювати нагадування про прийом ліків, а також за потреби відкладати або пропускати сповіщення. MangoHealth також зберігає всю інформацію про ліки в одному місці, тому користувачі можуть легко побачити, як їхні ліки взаємодітимуть з іншими ліками, що відпускаються за рецептом або без рецепта, а також їжею та напоями.

MangoHealth також має функцію створення нотаток, щоб користувачі могли відстежувати, як діють їхні ліки, і ділитися цією інформацією зі своїм лікарем або постачальником медичних послуг. Додаток також стимулює користувачів дотримуватися графіка прийому ліків за допомогою програмне забезпечення винагород, яка включає розіграші подарункових карток і благодійні пожертви (рис. 1.2).



Рисунок 1.2 – MangoHealth

Переваги MangoHealth:

- Одним із значних плюсів MangoHealth є те, що програма пропонує програму винагород, яка стимулює користувачів дотримуватися розкладу прийому ліків. Користувачі заробляють бали, які відкривають шанс виграти подарункові картки для роздрібних магазинів, таких як Target або The Gap, у щотижневих розіграшах.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

- Під час налаштування нагадувань про прийом ліків користувачі можуть налаштувати свій розклад. Наприклад, користувачі можуть встановлювати нагадування про ранкові ліки у вихідні дні пізніше, ніж у будні.
- MangoHealth також дозволяє користувачам створювати нагадування про здорові звички, зокрема зволоження, відстеження артеріального тиску та глюкози в крові, рухи та розтяжку, стеження за своєю вагою та запис настрою.
- Додаток також містить функцію Care Recipient, за допомогою якої користувачі MangoHealth можуть керувати ліками для тих, хто перебуває на їхньому лікуванні.

Недоліки MangoHealth:

- Недоліком платформи MangoHealth є те, що користувачі повинні вводити свої ліки вручну. На відміну від деяких інших програм, вони не мають можливості сканувати ліки чи імпортувати їх з аптеки. Це може зробити цю програму менш привабливою для користувачів, які приймають кілька ліків.
- Функція Healthy Habits обмежена попередньо запрограмованими звичками, які включені в додаток. Користувачі не можуть додавати власні звички для відстеження.

Для тих, кому не потрібно багато наворотів, RoundHealth – це простий у використанні додаток для нагадування про прийом ліків, доступний для користувачів пристроїв iOS.

RoundHealth дозволяє користувачам програмувати сповіщення про нагадування про рецептурні та безрецептурні ліки, а також про вітаміни та добавки, вибираючи їх із великого розкривного списку поширених ліків і добавок. Після вибору препарату користувачі можуть вибрати дозування зі списку варіантів або ввести власну дозу, якщо це необхідно (рис .1.3).

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

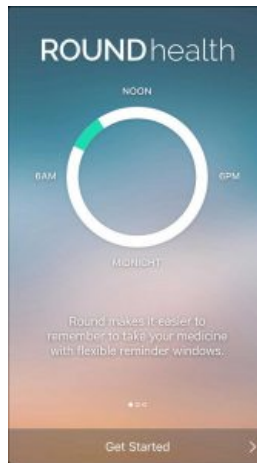


Рисунок 1.3 – RoundHealth

Користувачі встановлюють «вікно» сповіщень, протягом якого RoundHealth надсилатиме три сповіщення, по одному на початку, у середині та в кінці вікна, гарантуючи, що якщо користувачі пропустять сповіщення, вони все одно матимуть можливість прийняти ліки, коли це буде потрібно. Просте проведення пальцем або дотик фіксує, що сповіщення було підтверджено, і користувач прийняв ліки. Користувачі розумних годинників можуть синхронізувати програму зі своїм годинником, щоб отримувати сповіщення, навіть якщо вони знаходяться далеко від свого телефону.

Переваги RoundHealth:

- Ключова перевага програмне забезпечення нагадування про ліки RoundHealth полягає в тому, що клієнти можуть використовувати її для встановлення нагадувань про вітаміни та добавки, а також про призначення. Додаток попередньо завантажено з тисячами поширених вітамінів і добавок, а також ліків, що відпускаються за рецептом і без рецепта.
- RoundHealth також відстежуватиме, скільки доз ліків залишилося у користувача, що полегшить користувачам інформацію про те, коли їм потрібно замовити доповнення.
- Ще одна приємна особливість RoundHealth полягає в тому, що він синхронізується з Apple Watches, тож користувачам не обов'язково бути біля своїх телефонів, щоб отримувати нагадування про сповіщення.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

- Хоча всі ліки постачаються з попередньо встановленими дозами, RoundHealth також дає користувачам можливість додати індивідуальну дозу.
- Від користувачів не вимагається створювати обліковий запис у RoundHealth, тому це хороший варіант для користувачів, які хочуть захистити свою конфіденційність.

Недоліки RoundHealth:

- Найважливішим недоліком програмне забезпечення нагадування про ліки RoundHealth є те, що вона наразі доступна лише для пристроїв iOS. Користувачам Android доведеться вибрати інший додаток для нагадування про прийом ліків.
- Для ліків, які приймаються кілька разів на день, користувачі повинні створити окреме сповіщення для кожної дози. RoundHealth наразі не має опції для встановлення кількох сповіщень для того самого ліки чи добавки.

Існує безкоштовна версія DoseCast, але для користувачів, яким потрібна додаткова функціональність, також є платна версія Pro. Безкоштовні функції включають можливість налаштувати сповіщення про прийом ліків, вводити індивідуальні дози, робити нотатки про окремі ліки та інтелектуальне глушіння, тому сповіщення не надсилаються, коли користувачі сплять (рис. 1.4).

Версія Pro, яка коштує 2,99 доларів на місяць або 27,99 доларів на рік, ще надійніша. Користувачі з підпискою Pro можуть синхронізувати нагадування на необмеженій кількості пристроїв Apple, Android і Amazon.

Він також веде журнал прихильності до прийому ліків, щоб користувачі могли легко побачити, наскільки добре вони дотримувалися свого режиму прийому ліків.

Версія Pro також відстежує поповнення та дозволяє користувачам підключати свій обліковий запис до лікаря чи аптеки, що полегшує користувачам зв'язок із медичним працівником, якщо у них виникнуть запитання або потрібно замовити поповнення.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.4 – DoseCast

Переваги DoseCast:

– Основною перевагою DoseCast є те, що він сумісний майже з усіма розумними пристроями, включаючи пристрої Amazon. Користувачі з платною підпискою можуть синхронізувати свою програму на кількох пристроях, тож вони можуть отримувати нагадування, навіть якщо вони далеко від свого телефону.

– DoseCast відстежує, у якому часовому поясі перебуває користувач, і автоматично налаштовує нагадування, що робить це ідеальним варіантом для користувачів, які часто подорожують і не хочуть турбуватися про повторне налаштування нагадувань по ходу.

– DoseCast не збирає жодну особисту ідентифікаційну інформацію від користувачів і шифрує всю інформацію, надіслану через додаток, щоб користувачі були спокійні, знаючи, що їхні дані та конфіденційність захищені.

– Програма доступна французькою, німецькою та англійською мовами, що робить її доступною для більшої кількості користувачів.

Недоліки DoseCast:

– Найбільшим недоліком DoseCast є те, що деякі функції, як-от підрахунок таблеток і нагадування про поповнення, які включені безкоштовно в інші програмне забезпечення, доступні лише в платній версії Pro цієї програмне забезпечення.

– Якщо користувачі приймають різні дози того самого препарату протягом дня, вони повинні створити окреме нагадування для кожної дози.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток MyTherapy доступний майже 30 мовами, включаючи іспанську, арабську, корейську, японську, італійську та спрощену китайську, що робить його оптимальним вибором для людей, які не розмовляють англійською.

За допомогою MyTherapy користувачі можуть створювати нагадування про прийом ліків, скануючи штрих-коди своїх рецептів або вводячи їх власноруч.

Існує також функція команди, яка дозволяє користувачам додавати довірену особу, яка отримає сповіщення, якщо пацієнт не прийме ліки за розкладом (рис.1.5).

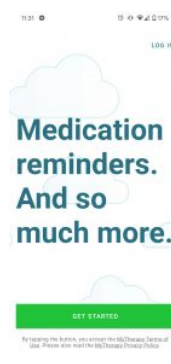


Рисунок 1.5 – MyTherapy

MyTherapy – це комплексна програма для охорони здоров'я, тобто користувачі також можуть встановлювати нагадування та відстежувати інші види здоров'я та вимірювання, зокрема вагу, артеріальний тиск, рівень глюкози в крові, фізичні вправи тощо. Усі ці вимірювання, включно з прихильністю до прийому лікарських засобів, консолідуються в простому щоденному звіті, щоб користувачі могли бачити, де вони на правильному шляху, а де потребують покращення. Потім уся ця інформація додається до ширшого огляду історії, що дозволяє користувачам бачити довгострокові моделі свого здоров'я та благополуччя.

Переваги MyTherapy:

- Ключовою перевагою MyTherapy є те, що він доступний для широкої міжнародної аудиторії. Програма доступна майже 30 мовами та пов'язана з базами даних про ліки в багатьох країнах, що полегшує користувачам у всьому світі введення інформації про ліки.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

- Особи, яким потрібна додаткова допомога, щоб дотримуватися розкладу прийому ліків, можуть скористатися командною функцією MyTherapy, яка дозволяє користувачам додавати довірених опікунів. Ці особи будуть повідомлені, якщо пацієнт не прийме ліки за розкладом.

- Для користувачів, яким потрібна комплексна програма для здоров'я та здоров'я, MyTherapy є гарним вибором. Окрім того, що користувачі допомагають відстежувати споживання ліків, існують також інструменти для моніторингу загального стану здоров'я та самопочуття, як-от нагадування про фізичні вправи та засоби відстеження вимірювань.

- Уся інформація про користувача записується в щоденні та щомісячні звіти, що дає їм коротко- та довгостроковий огляд свого здоров'я.

Недоліки MyTherapy:

- На відміну від деяких інших програм для нагадування про ліки, MyTherapy не пропонує багато нагадувань про поповнення. Користувачі, які хочуть покладатися на свій додаток, щоб не забувати замовляти поповнення, можуть захотіти звернути увагу на іншу платформу.

- MyTherapy також не надає попереджень про взаємодію з ліками або місця, де користувачі можуть додавати примітки про ліки, що є загальним для багатьох інших програм для нагадування про ліки.

CareZone – це цілісна програма для управління охороною здоров'я, яка полегшує пацієнтам запам'ятовувати прийом ліків, замовляти повторні рецепти, записувати важливі життєво важливі показники та відстежувати призначення лікаря (рис. 1.6).

У додатку CareZone користувачі можуть налаштовувати нагадування про прийом ліків у групах для кількох ліків, які приймаються одночасно щодня, або індивідуальні нагадування про прийом ліків за унікальним графіком. CareZone має функцію сканування, яка дозволяє користувачам легко додавати нові ліки до свого облікового запису.

Через аптеку CareZone користувачі можуть отримати безкоштовну доставку додому для поточних рецептів. CareZone надсилатиме користувачам їхні ліки в попередньо відсортованих упаковках таблеток, упорядкованих за дозами, датою та часом, допомагаючи гарантувати, що користувачі прийматимуть ліки за розкладом і в правильних дозах. Додаток CareZone також дозволяє користувачам відстежувати іншу інформацію про здоров'я, як-от артеріальний тиск і рівень глюкози в крові, а також надає відповіді на основні запитання щодо здоров'я через персонального радника з догляду.



Рисунок 1.6 – CareZone

Переваги CareZone:

- Однією з головних переваг CareZone є те, що вона включає аптечні послуги. Користувачі можуть замовляти дозаправки безпосередньо через додаток і безкоштовно доставляти їх у попередньо відсортованих упаковках за дозами, датою та часом прямо додому. Користувачі платять лише регулярні доплати за ліки, а CareZone приймає всі основні страхові плани, включаючи Medicare і Medicaid.
- Для користувачів, які мають право на Medicare, CareZone пропонує ресурси, які допоможуть людям купувати плани, заощаджувати гроші на премії та доплати, а також відповідати на запитання щодо програмне забезпечення.
- CareZone дозволяє користувачам створювати додаткові профілі для утриманців або близьких людей, ліками яких вони також можуть керувати.

					КВРІСТ 200178 20.01.01 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

Користувачі можуть легко перемикатися між профілями, щоб відстежувати всю відповідну інформацію про ліки для тих, про кого вони піклуються.

- CareZone також має функцію досягнення, яка допомагає мотивувати користувачів досягати цілей щодо здоров'я та активності.

Для користувачів, які хочуть поєднати свої нагадування про ліки з іншими нагадуваннями про здоров'я та самопочуття та відстеженням, варто розглянути додаток Care4Today.

Care4Today полегшує керування ліками. Користувачі можуть додавати свої ліки з бази даних Care4Today, розробленої за допомогою провідних аптек, і за потреби налаштовувати свої нагадування. Додаток також відстежуватиме запаси ліків і сповіщатиме користувачів, коли настане час замовити поповнення. Існує також інструмент, який дозволяє користувачам додавати довірених осіб або утриманців до свого облікового запису, що полегшує опікунам керування ліками для своїх близьких.

Окрім нагадувань про прийом ліків, Care4Today також може допомогти користувачам не відставати від своїх цілей щодо здоров'я та благополуччя за допомогою інструментів для розробки планів фізичних вправ, відстеження активності та дієти та вимірювання життєво важливих показників, таких як артеріальний тиск і вага. Користувачі також можуть покладатися на Care4Today, щоб допомогти їм відслідковувати прийом до лікаря за допомогою інструменту планування нагадувань у програмі.

Переваги Care4Today:

- Важливою перевагою Care4Today є те, що він пропонує надійні функції, окрім нагадувань про ліки. Користувачі також можуть покладатися на цю програму для нагадувань про запис до лікаря, засобів відстеження вправ і активності, вимірювань життєво важливих статистичних даних і звітів для відстеження свого прогресу.

- Оскільки Care4Today дозволяє користувачам зберігати всю важливу інформацію про здоров'я в одному місці, пацієнтам стає легше спілкуватися зі

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

своїми лікарями або постачальниками медичних послуг щодо схем лікування, симптомів і загального стану здоров'я.

Недоліки Care4Today:

- Програма Care4Today була вперше представлена в 2014 році, але багато відгуків клієнтів відзначають, що нещодавнє оновлення залишило програму з деякими помилками, включаючи численні небажані сповіщення та проблеми з позначенням ліків як прийнятих.
- Детальна інформація про те, як ліки додаються до програми, обмежена без фактичного завантаження програми, що може бути стримуючим фактором для користувачів, які хочуть заздалегідь знати, чи потрібно їм вводити інформацію про ліки вручну чи існує автоматична опція.

1.4 Висновки

У межах розділу 1 проведено було теоретичний аналіз та виявлено недоліки існуючих рішень, що привело до формулювання завдання щодо дослідження шляхів їх вирішення.

Також проведено аналіз відомих інформаційних систем обліку медичної інформації для прийому ліків мають свої недоліки.

Деякі інформаційні системи можуть мати складний або незручний інтерфейс, що може стати перешкодою для користувачів, особливо тих, хто не має великого досвіду в роботі з комп'ютерами.

У багатьох випадках інформаційні системи можуть бути не відомими або погано інтегрованими з іншими медичними системами або базами даних, що може ускладнювати обмін інформацією.

Деякі системи можуть має обмежені можливості нагадувань або не надають можливості персоналізації для різних типів призначень чи пацієнтів.

Недостатня безпека може стати серйозною проблемою, особливо враховуючи чутливість медичної інформації. Якщо система не забезпечує

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

ефективний рівень шифрування та захисту, це може вести до порушення конфіденційності пацієнтів.

Інформаційні системи обліку ліків можуть виявитися залежними від технологій, і в разі відмови системи або технічної проблеми може виникнути тимчасова недоступність або втрата даних.

Вартість розробки та впровадження інформаційних систем може бути значною, особливо для малих медичних практик або лікарень.

Впровадження нових інформаційних систем може вимагати додаткового навчання медичного персоналу, що може бути трудомістким та витратним за часом.

Таким чином, для подолання цих недоліків важливо розробити нову інформаційну систему з урахуванням конкретних потреб медичних закладів та забезпечити їхню високу якість, безпеку та користувацьку зручність.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ МЕДИЧНОЇ ІНФОРМАЦІЇ ДЛЯ ПРИЙОМУ ЛІКІВ

2.1 Розробка користувальницьких вимог. Функційні вимоги

Розглянемо основні практичні задачі та вимоги щодо проєктування інформаційної системи обліку медичної інформації для прийому ліків.

Метою проєкту є створення системи обліку медичної інформації для прийому ліків, яке буде використовуватися для ефективного управління прийомом ліків.

Розглянемо основний перелік функційних вимог до інформаційної системи обліку медичної інформації для прийому ліків:

- Реєстрація пацієнтів. Система повинна мати можливість реєстрації нових пацієнтів, зберігання їх особистої і медичної інформації.
- Ведення медичної інформації. Система повинна забезпечувати зберігання і обробку медичних даних пацієнтів, включаючи історію захворювань, рецепти, результати аналізів тощо.
- Призначення лікування. ІС повинна дозволяти лікарям призначати ліки для пацієнтів, враховуючи їх медичну історію, діагнози та інші фактори.
- Сповіщення про прийом ліків.
- Ведення реєстру ліків. Система повинна мати базу даних лікарських засобів, включаючи їх склад, дозування, побічні ефекти та іншу інформацію.
- Перевірка взаємодії ліків. Система повинна забезпечувати можливість перевірки взаємодії між різними ліками, щоб уникнути небажаних побічних ефектів.
- Стеження за вживанням ліків. Система може вести облік виписаних рецептів та вживаних доз ліків для кожного пацієнта.
- Генерація звітів і аналітика. Інформаційна система може надавати звіти та аналітичні дані щодо вживання ліків, ефективності лікування, запасів ліків тощо.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

- Безпека та конфіденційність. Система повинна гарантувати безпеку та конфіденційність медичної інформації пацієнтів від несанкціонованого доступу.

- Інтеграція з іншими системами: Інформаційна система може бути інтегрована з іншими медичними системами, такими як системи лабораторних досліджень або системи електронних медичних записів.

Розглянемо детальніше кожен пункт вимог.

2.2.1 Реєстрація пацієнтів

Функціональні вимоги до модуля реєстрації пацієнтів в інформаційній системі можуть включати наступні елементи:

- Створення нового запису пацієнта. Система повинна дозволяти операторам створювати нові записи для кожного нового пацієнта, зберігаючи основні особисті дані, такі як ім'я, прізвище, дата народження, адреса тощо.

- Редагування інформації пацієнта. Оператори повинні мати можливість змінювати та оновлювати інформацію про пацієнтів в разі зміни даних або виникнення помилок.

- Додавання медичної інформації. Покрокове заповнення медичної інформації, такої як медична історія, список алергій, хронічні захворювання, рецепти, аналізи та інші медичні дані.

- Пошук та фільтрація. Можливість швидкого пошуку пацієнтів за різними критеріями, такими як ім'я, прізвище, номер медичної картки тощо.

- Відображення історії візитів. Система повинна вміти відображати історію візитів пацієнта до медичного закладу, включаючи дати, призначення, результати обстежень тощо.

- Керування правами доступу. Забезпечення різних рівнів доступу до інформації в залежності від ролі користувача (лікар, медична сестра, адміністратор тощо).

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

- Інтеграція з іншими системами. Можливість інтегрувати модуль реєстрації пацієнтів з іншими системами, такими як система електронної медичної історії або система електронного запису візитів.

- Забезпечення безпеки даних. Захист особистих та медичних даних пацієнтів шляхом застосування відповідних методів шифрування та безпеки даних.

2.1.2 Ведення медичної інформації

Функціональні вимоги до модуля ведення медичної інформації в інформаційній системі можуть включати наступні елементи:

- Створення історії медичного обстеження. Модуль повинен дозволяти лікарям створювати та оновлювати історію медичного обстеження для кожного пацієнта, включаючи інформацію про симптоми, діагнози, процедури, лікування та інше.

- Зберігання результатів обстежень та аналізів. Система повинна мати можливість зберігати результати різних обстежень, лабораторних аналізів, зображень з медичних обстежень та іншу важливу медичну інформацію.

- Управління лікарськими рецептами. Модуль повинен дозволяти лікарям призначати лікування та виписувати рецепти для пацієнтів, забезпечуючи детальну інформацію про препарати, дозування та тривалість лікування.

- Історія прийому ліків. Система повинна вести історію вживання ліків для кожного пацієнта, включаючи дати та часи прийому, дозування та звіти про будь-які побічні ефекти або реакції на ліки.

- Доступ до медичної інформації в реальному часі. Модуль повинен надавати можливість лікарям та іншим медичним працівникам доступ до актуальної медичної інформації пацієнтів у режимі реального часу.

- Графіки та нагадування. Система може надавати функціональність для створення графіків лікування та нагадувань про прийом ліків для пацієнтів.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

- Інтеграція з іншими системами. Можливість інтеграції модуля ведення медичної інформації з іншими медичними системами, такими як система електронної медичної історії або системи лабораторних аналізів.

- Безпека даних. Забезпечення безпеки та конфіденційності медичної інформації пацієнтів шляхом застосування відповідних методів шифрування та захисту даних.

2.1.3 Призначення лікування

Функціональні вимоги до модуля призначення лікування в інформаційній системі можуть включати наступні елементи:

- Призначення ліків. Модуль повинен дозволяти лікарям призначати лікування для пацієнтів, враховуючи їх медичну історію, діагнози, алергії та інші фактори.

- Вибір ліків з бази даних. Система повинна мати базу даних лікарських засобів, з якої можна вибирати потрібні препарати для призначення.

- Дозування та режим прийому. Модуль повинен дозволяти вказувати дозування ліків та режим прийому для кожного призначення.

- Контроль за взаємодією ліків. Система може надавати можливість перевірки взаємодії між різними ліками, щоб уникнути небажаних побічних ефектів.

- Моніторинг результатів лікування. Модуль може включати функціонал для моніторингу ефективності лікування та змін у стані пацієнта після призначення ліків.

- Запис в електронну медичну картку. Призначення лікування повинно автоматично записуватись в електронну медичну картку пацієнта для подальшого контролю та аналізу.

- Генерація рецептів. Система може надавати можливість генерації рецептів для призначених ліків для пацієнтів.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

- Нагадування про лікування. Модуль може надавати можливість створення нагадувань для пацієнтів щодо прийому ліків.
- Аналітика та звіти. Система може надавати аналітичні звіти про призначене лікування, включаючи кількість призначень, популярність ліків та інше.

2.1.4 Сповіщення про прийом ліків

Функціональні вимоги до модуля сповіщення про прийом ліків в інформаційній системі можуть включати наступні елементи:

- Налаштування нагадувань. Модуль повинен дозволяти користувачам встановлювати належні нагадування для прийому ліків, включаючи частоту, час та тип сповіщень (звукові, візуальні, текстові тощо).
- Індивідуальні налаштування. Користувачі повинні мати можливість налаштовувати нагадування залежно від своїх потреб і графіка прийому ліків.
- Підтримка кількох лікарських рецептів. Модуль повинен здати надавати нагадування про прийом різних лікарських засобів, які були призначені різними лікарями.
- Можливість затвердження нагадувань. Користувачі можуть підтверджувати, що вони прийняли відповідну дозу ліків після отримання нагадування.
- Нагадування про попередні пропущені дози. Система може надавати нагадування про попередні пропущені дози ліків для підтримки відповідності графіку прийому.
- Нагадування для альтернативних методів прийому. Модуль може надавати можливість налаштування нагадувань для альтернативних методів прийому ліків, таких як застосування пластинок, ін'єкцій або зовнішніх застосувань.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

- Інтерактивність та гнучкість. Система може надавати можливість користувачам взаємодіяти з нагадуваннями, змінюючи час або відкладаючи їх, якщо потрібно.

Інтеграція з іншими системами. Модуль може бути інтегрований з іншими системами, такими як система електронної медичної історії, щоб автоматично оновлювати дані про прийом ліків.

2.1.5 Ведення реєстру ліків

Функціональні вимоги до модуля ведення реєстру ліків в інформаційній системі можуть включати наступні елементи:

- Додавання нових лікарських засобів. Модуль повинен дозволяти адміністраторам або іншим уповноваженим користувачам додавати нові лікарські засоби до бази даних, включаючи назву, дозування, форму випуску, інструкції з використання, побічні ефекти тощо.

- Редагування і видалення існуючих лікарських засобів. Користувачі повинні мати можливість редагувати інформацію про існуючі лікарські засоби або видаляти їх з бази даних при необхідності.

- Пошук і фільтрація лікарських засобів. Система повинна надавати можливість швидкого пошуку та фільтрації лікарських засобів за різними критеріями, такими як назва, дозування, категорія тощо.

- Ведення історії виписаних рецептів. Модуль може вести історію виписаних рецептів для кожного пацієнта, включаючи дату виписування, назву лікарського засобу, дозування та тривалість прийому.

- Контроль за залишками ліків. Система може надавати можливість ведення обліку залишків ліків на складі або в аптеці, оновлюючи інформацію про кількість доступних одиниць після кожного виписування рецептів.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

2.1.6 Перевірка взаємодії ліків

Функціональні вимоги до модуля перевірки взаємодії ліків в інформаційній системі можуть включати наступні елементи:

- Перевірка взаємодії між ліками. Модуль повинен забезпечувати можливість перевірки взаємодії між різними лікарськими засобами, включаючи виявлення можливих негативних взаємодій, алергій або побічних ефектів при одночасному прийомі декількох ліків.

- База даних взаємодій. Система повинна мати базу даних, яка містить інформацію про відомі взаємодії між різними лікарськими засобами, алергічні реакції та інші побічні ефекти.

- Автоматичне сповіщення. Модуль може надавати автоматичні сповіщення про виявлені взаємодії ліків, щоб лікарі могли прийняти відповідні заходи, такі як зміна дози, вибір альтернативного лікування або нагляд за пацієнтом.

- Інтерфейс для користувача. Система повинна мати зручний інтерфейс, який дозволяє лікарям швидко та легко перевіряти взаємодію між ліками для конкретного пацієнта.

- Інтерактивні можливості. Модуль може надавати можливість взаємодії з користувачем, наприклад, шляхом запитання додаткової інформації про призначені ліки або надання рекомендацій щодо виявлених взаємодій.

- Аналіз історії прийому ліків. Система може враховувати історію прийому ліків пацієнтом для виявлення попередніх випадків негативних взаємодій або алергічних реакцій.

- Інтеграція з іншими модулями. Модуль може бути інтегрований з іншими частинами інформаційної системи, такими як модуль ведення медичної інформації або модуль призначення лікування, для отримання додаткової контекстної інформації.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

2.1.7 Стеження за вживанням ліків

Функціональні вимоги до модуля стеження за вживанням ліків в інформаційній системі можуть включати наступні елементи:

- Реєстрація прийому ліків. Модуль повинен дозволяти пацієнтам або медичному персоналу реєструвати прийом кожної дози лікування.
- Нагадування про прийом ліків. Система повинна надавати можливість встановлення нагадувань про прийом ліків для пацієнтів або їх опікунів за заданим графіком.
- Підтвердження прийому ліків. Пацієнти можуть підтверджувати прийом кожної дози лікування за допомогою системи, що підтверджує дотримання рекомендацій щодо лікування.
- Аналіз режиму прийому. Система може аналізувати дані про прийом ліків для виявлення відхилень від рекомендованого режиму та надсилати сповіщення про це.
- Спостереження за дозуванням. Модуль може відслідковувати дозування кожного лікування та надавати сповіщення у випадку виявлення помилок або відхилень від рекомендованої дози.
- Графіки та звіти. Система може генерувати графіки та звіти про прийом ліків для пацієнтів та їх лікарів, що допомагає в оцінці дотримання режиму прийому.

2.1.8 Генерація звітів і аналітика

Функціональні вимоги до модуля генерації звітів і аналітики в інформаційній системі можуть включати наступні елементи:

- Створення звітів за різними критеріями. Модуль повинен надавати можливість створення звітів за різними параметрами, такими як період часу, пацієнти, лікарські засоби, медичні процедури тощо.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

- Аналіз даних та показників. Система повинна забезпечувати можливість аналізу даних та показників, таких як кількість виписаних рецептів, популярність ліків, дотримання режиму прийому ліків тощо.
- Генерація графіків і діаграм. Модуль повинен мати можливість створювати графіки, діаграми та інші візуалізації даних для зручного аналізу та сприяння прийняттю рішень.
- Експорт звітів. Система може надавати можливість експорту звітів у різні формати, такі як PDF, Excel, CSV, що дозволяє додатково опрацьовувати інформацію за межами системи.
- Підтримка розрахунків та статистичного аналізу. Модуль може забезпечувати можливість розрахунків різних статистичних показників, таких як середнє значення, медіана, стандартне відхилення тощо.
- Сповіщення про важливі події. Система може надавати можливість налаштовувати сповіщення про важливі події або тенденції, які виявлені під час аналізу даних.
- Збереження звітів і історії аналітики. Модуль повинен забезпечувати можливість зберігання звітів і історії аналітики для подальшого використання або аудиту.

2.2 Застосування Android SDK для розроблення програмного забезпечення інформаційної системи

Android SDK - це набір бібліотек та інструментів для розробки програмного забезпечення, необхідних для розробки додатків для Android [!!!!!!!!].

Android SDK складається з деяких інструментів, які є дуже важливими для розробки додатків для Android. Ці інструменти забезпечують безперебійний потік процесу розробки, починаючи від розробки і закінчуючи налагодженням. Android SDK сумісний з усіма операційними системами, такими як Windows, Linux, macOS тощо.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

Компоненти Android SDK відіграють важливу роль у розробці додатків для Android. Нижче наведені найважливіші компоненти:

1. Інструменти Android SDK. Інструменти Android SDK є важливим компонентом Android SDK. Він складається з повного набору інструментів для розробки та налагодження. Нижче наведено інструменти для розробників SDK:

1. Android SDK Build tool.
2. Емулятор Android.
3. Android SDK Platform-tools.
4. Інструменти Android SDK.

Інструменти збірки Android SDK використовуються для створення власне двійкових файлів Android-додатків. Основними функціями інструментів Android SDK Build є створення, налагодження, запуск і тестування Android-додатків.

Емулятор Android - це пристрій, який імітує пристрій Android у вашій системі. Припустимо, необхідно запустити застосунок для Android, один з варіантів - запустити його на мобільному пристрої Android, увімкнувши на ньому функцію налагодження через USB. Інший варіант - використання емулятора Android. В емуляторі Android в системі відображається віртуальний пристрій Android, на якому запускається ПЗ для Android, яку ми кодуємо.

Таким чином, це просто означає, що без необхідності фізичного пристрою Android SDK компонент «Емулятор Android» надає віртуальний пристрій в системі, на якому ми запускаємо наш додаток.

Android SDK Platform-tools корисний, коли ми працюємо над проектом, і він одночасно показує повідомлення про помилки. Він використовується спеціально для тестування. Він включає в себе:

Android Debug Bridge (ADB) – це інструмент командного рядка, який допомагає спілкуватися з пристроєм. Він дозволяє нам виконувати такі дії, як встановлення програми, налагодження програми тощо.

Fastboot дозволяє прошивати пристрій новим образом системи.

Інструменти Systrace допомагають збирати та перевіряти часову інформацію. Це дуже важливо для налагодження програми.

Android SDK tool - це компонент SDK tool. Він складається з набору інструментів та інших утиліт, які мають вирішальне значення для розробки додатків для Android. Він містить повний набір інструментів для налагодження та розробки для Android.

2.3 Компоненти Android для програмного забезпечення інформаційної системи

Для реалізації проєкту було застосовано Android Jetpack - це набір програмних компонентів, бібліотек, інструментів та інструкцій для розробки надійних додатків для Android.

Jetpack включає в себе існуючі бібліотеки підтримки Android, компоненти архітектури Android, а також бібліотеку Android KTX як єдиний модульний об'єкт.

Область інтерфейсу включає віджети, анімацію, палітри тощо для покращення користувацького досвіду. Вона також надає сучасні шрифти емодзі, які можна використовувати в програмах.

Jetpack складається з широкої колекції бібліотек, які створені таким чином, щоб працювати разом і створювати надійні мобільні додатки. Його програмні компоненти поділено на 4 категорії:

1. Базові компоненти.
2. Компоненти архітектури.
3. Компоненти поведінки.
4. Компоненти інтерфейсу користувача.

В проєкті було використано такі компоненти інтерфейсу користувача:

1. Анімація та перехід.
2. Авто.
3. Емодзі.

4. Фрагмент.

5. Макет.

6. Палітра.

7. Телевізор.

Jetpack пропонує API для налаштування різних видів анімації, доступних для додатків Android. Цей фреймворк надає можливість переміщувати віджети, а також перемикатися між екранами за допомогою анімації та переходів у додатку.

Щоб покращити користувацький досвід, анімація було використано для анімації змін, що відбуваються в додатку, а фреймворк переходів дає можливість налаштувати зовнішній вигляд цих змін.

Для здійснення керування тим, як перехід змінює зовнішній вигляд програми під час переходу з одного екрану на інший.

Jetpack Compose - це інструментарій для створення нативного інтерфейсу Android. Він пропонує більш модульний підхід до розробки додатків, організовуючи код у менші та багаторазові компоненти. Ці компоненти легко підтримувати, а код, написаний у них, описує зовнішній вигляд інтерфейсу декларативно, тобто на основі наявного стану компоненти.

Для створення анімації в Android було використано композит під назвою Transition. Його гнучка природа дозволяє розробникам легко передавати інформацію користувачеві за допомогою анімації властивостей компонента.

Нижче наведено елементи, що входять до складу композиту Transition, який керує загальною анімацією компонента:

1. TransitionDefinition включає визначення всіх анімацій, а також різних станів анімації, необхідних під час переходу.

2. initState описує початкову стадію переходу. Якщо він не визначений, то приймає значення першого toState, доступного в переході.

3. toState описує наступний стан переходу.

4. clock керує анімацією зі зміною часу. Це необов'язковий параметр.

5. `onStateChangeFinished` - необов'язковий слухач, який повідомляє про завершення анімації зміни стану.

6. `children` - дочірні елементи, які будуть анімовані

Android Jetpack надає кілька попередньо визначених конструкторів анімації, які розробники можуть використовувати безпосередньо у своєму додатку. `TransitionDefinition` містить код для всіх цих анімацій:

1. `Tween` - анімація геометричних змін об'єкта, таких як рух, обертання, розтягування тощо.

2. `Physics` визначає анімацію пружини для об'єкта, задаючи коефіцієнт демпфування та жорсткість.

3. `Keyframe` - створення анімації, в якій значення цільового об'єкта змінюється з плином часу.

4. `Snap` - анімація миттєвого переходу з одного стану в інший.

5. `Repeatable` використовується для повторення анімації стільки разів, скільки потрібно розробнику.

Клас підтримки фрагментів Android дозволив розділити інтерфейс користувача на окремі частини, що забезпечує модульність і можливість повторного використання інтерфейсу користувача в активності.

Частина користувацького інтерфейсу визначається фрагментом, який потім вбудовується в активність. Фрагменти не існують без активності.

Android Jetpack надав значні покращення та розширені можливості у використанні фрагментів. Бібліотеки `Navigation`, `BottomNavigationView` і `ViewPager2` в Jetpack призначені для набагато ефективнішої роботи з фрагментами ПЗ.

Крім того, гарантується належна інтеграція класу фрагмента з класом життєвого циклу компонента архітектури jetpack. Нижче наведено список нещодавно функцій для Android розробників.

Спільне використання та комунікація між фрагментами. Для того, щоб підтримувати незалежність фрагментів, розробники пишуть код таким чином, щоб

не дозволяти фрагментам безпосередньо взаємодіяти з іншими фрагментами або з активністю їхнього хоста.

Бібліотека фрагментів Jetpack надала два варіанти для встановлення зв'язку, а саме Fragment Result API та спільна ViewModel. Fragment Rest API підходить для одноразових результатів з даними, які можна розмістити у пакеті. Крім того, якщо є потреба у спільному доступі до постійних даних разом з будь-якими користувацькими API, ViewModel є кращим. Вона також здатна зберігати та керувати даними інтерфейсу користувача. Розробники можуть вибирати між цими двома підходами відповідно до вимог програми.

Конструктор може утримувати ідентифікатор ресурсу Layout. AndroidX AppCompatActivity 1.1.0 та Fragment 1.1.0 дозволяють конструктору приймати ідентифікатор макета як параметр.

Завдяки цьому у фрагментах спостерігається значне зменшення кількості перевизначень методів.

FragmentManager та бібліотека Navigation. Всі найважливіші завдання фрагмента, такі як додавання, видалення, заміна, а також відправка їх назад до стеку, виконуються класом FragmentManager. Для виконання всіх цих завдань, пов'язаних з навігацією, Jetpack рекомендує використовувати бібліотеку Navigation. Структура цієї бібліотеки надає розробникам деякі найкращі практики, щоб вони могли ефективно працювати з фрагментами, менеджером фрагментів та зворотним стеком.

Кожен андроїд застосунок, який містить фрагменти у своєму інтерфейсі, повинен використовувати FragmentManager на певному рівні. Однак можна не взаємодіяти з FragmentManager безпосередньо під час використання бібліотеки Jetpack Navigation.

FragmentFactory. Бібліотека AndroidX, представлена разом з Jetpack, пропонує клас FragmentFactory. Структура цього API є простою та узагальненою, що дозволяє розробникам створювати екземпляри фрагментів у власний спосіб. Для того, щоб перевизначити спосіб екземплярів фрагмента за

замовчуванням, необхідно зареєструвати `FragmentManager` у `FragmentManager` додатку.

Тестування фрагментів. `AndroidX Test` був запущений Google, щоб зробити тестування необхідною частиною `Jetpack`. Існуючі бібліотеки разом з деякими новими API та повною підтримкою `Kotlin` у `AndroidX Test` надають можливість писати відповідні та стислі тести.

Клас `FragmentScenario` бібліотеки `AndroidX` створює середовище для виконання тестів на фрагментах. Він складається з двох основних методів для запуску фрагментів у тесті, перший - `launchInContainer()`, який використовується для тестування користувацького інтерфейсу фрагмента. Інший метод - `launch()`, який використовується для тестування без користувацького інтерфейсу фрагмента. У деяких випадках фрагменти мають певні залежності.

Щоб згенерувати тестові версії цих залежностей, необхідно надати власну `FragmentManager` методам `launchInContainer()` або `launch()`.

Можна вибрати один з методів і використовувати `Espresso UI` тести для перевірки інформації про елементи інтерфейсу фрагмента. Для використання класу `FragmentScenario` необхідно визначити артефакт тестування фрагментів у файлі `build.gradle` на рівні додатку.

`FragmentManager`. У `AndroidX Fragment 1.2.0` є `FragmentManager`, який розширює `FrameLayout` і надає можливість кастомізувати компонування фрагментів у додатках для `Android`. Він використовується як батько фрагмента, щоб узгодити його поведінку з поведінкою фрагмента і забезпечити гнучкість у таких завданнях, як транзакції фрагментів. Він також підтримує атрибути `<fragment>` і вирішує проблеми диспетчеризації віконних вставок. Крім того, цей контейнер вирішує деякі проблеми анімації, пов'язані з z-впорядкуванням фрагментів, наприклад, фрагменти, що завершують роботу, більше не з'являються у верхній частині подання.

Структура інтерфейсу користувача, як і діяльність програми, визначається макетом (`Layout`). Він визначає об'єкти `View` і `ViewGroup`. `View` і `ViewGroup`

можуть бути створені двома способами: оголошенням елементів інтерфейсу у XML або написанням коду, тобто програмно. Ця частина Jetpack охоплює деякі з найпоширеніших макетів, таких як `LinearLayout`, `RelativeLayout` і абсолютно новий `ConstraintLayout`. Крім того, офіційна документація Jetpack Layout надає деякі вказівки щодо створення списку елементів за допомогою `RecyclerView` і макета картки за допомогою `CardView`. Подання є видимим для користувача. `EditView`, `TextView` та `Button` є прикладами `View`.

`ViewGroup` є об'єктом-контейнером, який визначає структуру макета для `View(iv)` і тому є невидимим. Прикладами `ViewGroup` є `LinearLayout`, `RelativeLayout` і `ConstraintLayout`.

Правильне поєднання кольорів відіграє важливу роль у покращенні користувацького досвіду. Тому це важливий аспект у процесі розробки додатків. Розробники часто створюють додатки, в яких елементи інтерфейсу змінюють свій колір відповідно до часу (день і ніч). Цей тип практики дає користувачеві гарне відчуття і забезпечує ефект занурення під час використання програми. Для виконання цих завдань Android Jetpack надає нову бібліотеку підтримки `Palette`. Вона здатна витягувати невеликий набір кольорів із зображення.

Колір стилізує елементи керування інтерфейсу програми та оновлює іконки на основі кольору фонового зображення. `Material Design` андроїд-застосунків є причиною популярності динамічного використання кольорів.

Бібліотека `Palette` містить яскраві та приглушені тони зображення. Вона також включає кольори тексту на передньому плані, щоб забезпечити максимальну читабельність.

Бібліотека `Palette` також надає розробникам можливість вибрати кількість кольорів, які вони хочуть згенерувати з певного джерела зображення.

За замовчуванням значення `numberOfColors` у результуючій палітрі встановлено рівним 16. Однак, кількість кольорів може бути збільшена до 24-32. Час, необхідний для створення повної палітри, прямо пропорційний кількості кольорів. Після того, як `Palette` згенерована, для доступу до цих кольорів

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

використовується зразок (своєрідний метод). Кожному колірному профілю відповідає відповідний зразок, який повертає колір з цієї палітри. Нижче наведено колірні профілі, згенеровані API бібліотеки Palette.

2.4 Застосування шаблон проектування MVC (Model View Controller) в Android

Для реалізації інформаційної системи обліку медичної інформації для прийому ліків було застосовано шаблон проектування MVC.

Шаблон проектування надає модульність файлам проекту і гарантує, що всі коди будуть охоплені модульним тестуванням. Це полегшує розробникам завдання підтримки програмного забезпечення та розширення можливостей програмного забезпечення в майбутньому.

Патерн MVC передбачає поділ коду на 3 компоненти. Створюючи клас/файл додатку, розробник повинен віднести його до одного з наступних трьох рівнів:

1. Модель. Цей компонент зберігає дані програми. Він не має ніяких знань про інтерфейс. Модель відповідає за обробку логіки предметної області (реальних бізнес-правил) і зв'язок з базою даних і мережевим рівнем.

2. Представлення. Це рівень інтерфейсу користувача (UI), який містить компоненти, видимі на екрані. Крім того, він забезпечує візуалізацію даних, що зберігаються в моделі, і пропонує взаємодію з користувачем.

3. Контролер: Цей компонент встановлює взаємозв'язок між представленням і моделлю. Він містить основну логіку програми, отримує інформацію про поведінку користувача та оновлює модель відповідно до потреб.

Незважаючи на те, що схема MVC використовується для створення модульного дизайну додатку, шари коду залежать один від одного.

У цьому патерні View і Controller залежать від моделі. Існує декілька підходів до застосування патерну MVC у проекті.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

Підхід 1. Діяльність і фрагменти можуть виконувати роль контролера і відповідати за оновлення Представлення.

Підхід 2. Існує можливість використовувати активності або фрагменти як представлення і контролер, а модель буде окремим класом, який не розширює жодного класу Android.

В архітектурі MVC дані додатку оновлюються контролером, а View отримує ці дані. Оскільки компонент Model відокремлений, його можна тестувати незалежно від інтерфейсу користувача.

Крім того, якщо рівень View дотримується принципу єдиної відповідальності, то його роль полягає лише в оновленні контролера для кожної події користувача і простому відображенні даних з моделі, без реалізації будь-якої бізнес-логіки.

У цьому випадку тестів інтерфейсу користувача повинно бути достатньо, щоб покрити функціональність представлення.

2.5 Макети в дизайні інтерфейсу Android

В проєкті було звстосовано Layout Managers, який є розширенням класу ViewGroup. Вони використовуються для встановлення положення дочірніх Видів в інтерфейсі, який будується.

Можна вкладати компонування, а отже, створювати як завгодно складні інтерфейси, використовуючи комбінацію компонувань.

В Android SDK є декілька класів макетів. Їх можна використовувати, змінювати або створювати власні для створення інтерфейсу для ваших Поглядів, Фрагментів і Дій. Ви можете ефективно відображати ваш вміст, використовуючи правильну комбінацію макетів.

Найпоширенішими класами макетів, які можна знайти в Android SDK, є:

1. `FrameLayout` - це найпростіший з менеджерів компонування, який закріплює кожне дочірнє подання у своєму фреймі. За замовчуванням позиція -

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

верхній лівий кут, хоча атрибут гравітації можна використовувати для зміни розташування. Ви можете додавати декілька дочірніх елементів, кожен новий елемент накладається на попередній, при цьому кожен новий елемент може затуляти попередні.

2. Лінійне компоування вирівнює кожне з дочірніх подань по вертикальній або горизонтальній лінії. Вертикальна компоновка має стовпчик Поглядів, тоді як горизонтальна компоновка має рядок Поглядів. Він підтримує атрибут ваги для кожного дочірнього подання, який може контролювати відносний розмір кожного дочірнього подання в межах доступного простору.

3. `RelativeLayout` - макет, який дозволяє визначати положення кожного дочірнього подання відносно інших подань і розмірів екрана.

4. `GridLayout` використовує прямокутну сітку з нескінченно тонких ліній для компоування подань у рядки та стовпці. Сітчастий макет неймовірно гнучкий і може бути використаний для значного спрощення макетів і зменшення або усунення складної вкладеності, часто необхідної для побудови інтерфейсів з використанням макетів, описаних раніше.

Кожен з цих макетів розроблений для масштабування відповідно до розміру екрану пристрою, уникаючи використання абсолютних координат позицій або заздалегідь визначених значень пікселів. Це робить додаток придатним для різноманітного набору пристроїв Android.

Макет Android використовується для визначення інтерфейсу користувача, який містить елементи керування інтерфейсом або віджети, що з'являтимуться на екрані андроїд-додатку або на екрані активності.

Як правило, кожен додаток є комбінацією `View` і `ViewGroup`. Як ми знаємо, андроїд-додаток містить велику кількість активностей, і можна сказати, що кожна активність - це одна сторінка додатку. Отже, кожна активність містить декілька компонентів користувацького інтерфейсу, і ці компоненти є екземплярами `View` і `ViewGroup`. Всі елементи в макеті будуються за допомогою ієрархії об'єктів `View` і `ViewGroup`.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

Вигляд (View) - це інтерфейс користувача, який використовується для створення інтерактивних компонентів інтерфейсу, таких як TextView, ImageView, EditText, RadioButton тощо, і відповідає за обробку подій та малювання. Зазвичай їх називають віджетами.

Група ViewGroup є базовим класом для макетів і параметрів макетів, які містять інші View або ViewGroups, а також для визначення властивостей макета. Вони зазвичай називаються компоунваннями.

Фреймворк Android дозволяє використовувати елементи інтерфейсу або віджети двома способами:

1. Використовувати елементи інтерфейсу в XML-файлі.
2. Динамічно створювати елементи у файлі Kotlin.

В проєкті було застосовано такі типи макетів:

1. Лінійна компоновка Android. LinearLayout - це підклас ViewGroup, який використовується для розміщення дочірніх елементів View один за одним у певному напрямку - по горизонталі або по вертикалі - на основі властивості орієнтації.

2. Відносне розташування Android. RelativeLayout - це підклас ViewGroup, який використовується для визначення положення дочірніх елементів View відносно один одного, наприклад (A праворуч від B) або відносно батьківського (зафіксувати у верхній частині батьківського).

3. Android Constraint Layout. ConstraintLayout - це підклас ViewGroup, який використовується для визначення положення обмежень компоунвання для кожного дочірнього подання відносно інших присутніх подань. ConstraintLayout подібний до RelativeLayout, але має більше можливостей.

4. Android Frame Layout: FrameLayout - це підклас ViewGroup, який використовується для визначення положення елементів View, що містяться в ньому, один над одним для відображення лише одного View всередині FrameLayout.

5. Android Table Layout. `TableLayout` - це підклас `ViewGroup`, який використовується для відображення дочірніх елементів `View` у рядках і стовпчиках.

6. Android Web View: `WebView` - це браузер, який використовується для відображення веб-сторінок у нашому макеті діяльності.

7. Android ListView: `ListView` - це `ViewGroup`, що використовується для відображення прокручуваних списків елементів в одному стовпчику.

8. Android Grid View: `GridView` - це група перегляду, яка використовується для відображення прокручуваного списку елементів у вигляді сітки з рядків і стовпців.

2.6 Використання елементів інтерфейсу в XML-файлі

Використання елементів інтерфейсу в XML-файлі дозволило створити макет, подібний до веб-сторінок. Файл макета XML містить принаймні один кореневий елемент, до якого можна додати додаткові елементи макета або віджети для побудови ієрархії подання.

Коли було створено макет, потрібно завантажити ресурс XML-макета з методу зворотного виклику `onCreate()` нашої активності і отримати доступ до елементів інтерфейсу з XML за допомогою `findViewById`.

`LinearLayout` - це найпростіша компоновка в android studio, яка вирівнює всі дочірні елементи послідовно по горизонталі або по вертикалі за допомогою атрибуту `android:orientation`. Якщо застосувати `android:orientation=«vertical»`, то елементи будуть розташовані один за одним по вертикалі, а якщо застосувати `android:orientation=«horizontal»`, то елементи будуть розташовані один за одним по горизонталі.

`Android RelativeLayout` - це підклас `ViewGroup`, який використовується для визначення положення дочірніх елементів `View` відносно один одного, наприклад (А праворуч від В) або відносно батьківського (зафіксувати у верхній частині батьківського).

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Замість використання `LinearLayout`, ми повинні використовувати `RelativeLayout`, щоб спроектувати інтерфейс користувача і зберегти нашу ієрархію пласкою, тому що це покращує продуктивність програми.

2.6 Висновок

У розділі представлено опис користувальницьких вимог до інформаційної системи обліку медичної інформації для прийому ліків.

Також у розділі подано функційні вимоги до інформаційної системи, зокрема процедури реєстрація пацієнтів, ведення медичної інформації, призначення лікування, сповіщення про прийом ліків, ведення реєстру ліків, перевірка взаємодії ліків, стеження за вживанням ліків, генерація звітів і аналітика.

					КвРІСТ 200178 20.01.01 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ МЕДИЧНОЇ ІНФОРМАЦІЇ ДЛЯ ПРИЙОМУ ЛІКІВ

3.1 Проектування архітектури програмного забезпечення інформаційної системи

Програмна реалізація інформаційної системи розроблена з використанням інтеграції мобільного зв'язку в режимі реального часу між кількома користувачами на одній платформі. Він розроблений для роботи на смартфонах з операційною системою Android. Фронт-енд розроблений за допомогою android studio, а бек-енд - за допомогою фреймворку Firebase.

Методологія розробки програмного забезпечення, що використовується для цієї роботи, - це agile-підхід. Agile методологія - це практика, яка сприяє безперервній ітерації розробки та тестування протягом усього життєвого циклу розробки програмного забезпечення проєкту (рис. 3.1).



Рисунок 3.1 - Agile-процес

Фреймворк Firebase надає базу даних у режимі реального часу та бекенд як послугу. Сервіс надає розробникам додатків API, який дозволяє синхронізувати дані додатків між клієнтами та зберігати їх у хмарі Firebase Farr (2013), Marshall (2013). Компанія надає клієнтські бібліотеки, які забезпечують інтеграцію з додатками на Android, iOS, JavaScript, Java, Objective-C, Swift та Node.js. База даних

також доступна через REST API та прив'язки для декількох JavaScript фреймворків, таких як Angular JS, React, Ember.js та Backbone.js REST API використовує протокол Server-Sent Events, який є API для створення HTTP-з'єднань для отримання push-повідомлень від сервера.

Розробники, які використовують базу даних в режимі реального часу, можуть захистити свої дані, використовуючи правила безпеки на стороні сервера визначеної компанії.

Застосунок розроблений як клієнт-серверний застосунок з двома клієнтськими інтерфейсами: інтерфейс спеціаліста та інтерфейс пацієнта, який називається "консультація".

Інтерфейс користувача застосунку для пацієнтів дещо відрізняється від інтерфейсу застосунку для спеціалістів.

Наприклад, пацієнтам потрібно спочатку зареєструватися, а потім увійти в систему, щоб отримати повний доступ до застосунку, і немає обмежень на кількість пацієнтів, які можуть зареєструватися, тоді як чотири спеціалісти були додані до внутрішньої частини системи, і їхні паролі для входу в систему будуть надані їм адміністрацією.

Пацієнти можуть відновити свій пароль, якщо вони його забули, за допомогою діючої адреси електронної пошти.

Клієнт-сервер firebase допомагає з'єднати застосунок пацієнта з додатком спеціаліста і слугує репозиторієм для зберігання трансляцій та розмовних повідомлень між пацієнтом та спеціалістом.

Основна перевага полягає в тому, що користувачі можуть легко отримати доступ до життєво важливої інформації після успішного входу в систему, вона не використовує внутрішню пам'ять смартфона і запобігає втраті інформації, оскільки все зберігається в хмарі firebase (рис.3.2).

Блок-схеми алгоритмів роботи застосунку подано на рисунку 3.3.



Рисунок 3.2 - Інтеграція з Firebase

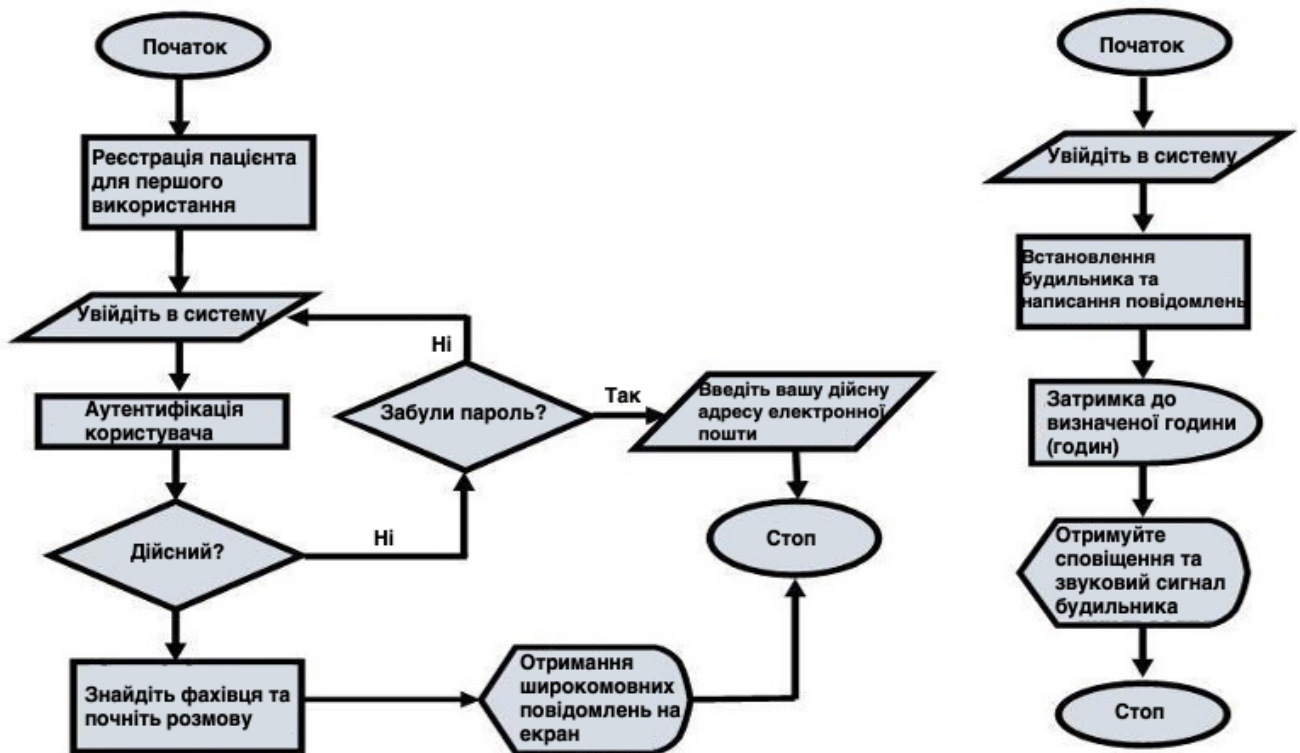


Рисунок 3.3 - Блок-схеми алгоритму роботи застосунку

Таким чином, користувачі:

- можуть запускати застосунок;

- повинні вибрати між входом (після попередньої реєстрації) або реєстрацією (для нового користувача);
- Якщо було обрано опцію входу в систему;
- автентифікація користувачів буде перевірятися на справжню ідентичність;
- могли спілкуватися з медичним персоналом і встановлювати одне або кілька нагадувань;
- можуть отримувати повідомлення;
- налаштовувати сповіщення.

Лікарі можуть виконувати наступні дії:

- лікарі запустили б застосунок;
- Лікарі повинні будуть обирати між входом (після попередньої реєстрації) та реєстрацією (для нового користувача);
- обрати варіант входу в систему;
- автентифікація користувачів буде перевірятися на предмет справжньої ідентичності;
- після цього медичний персонал може спілкуватися з пацієнтами в приватному порядку і надсилати широкомовні повідомлення всім своїм пацієнтам.

На рисунку 3.4 показано послідовність дій на згаданих вище етапах.

На рисунку 3.5 подано блок-схему алгоритму руху даних в системі.

3.2 UML діаграми

Діаграми UML - це стандартний спосіб візуалізації дизайну та архітектури програмних систем. Існує кілька типів UML-діаграм, кожен з яких слугує різним цілям у розробці програмного забезпечення. Нижче наведено кілька найпоширеніших UML-діаграм, які використовуються при розробці програмного забезпечення:

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Діаграма класів ілюструє структуру системи, показуючи класи, атрибути, операції та зв'язки між ними.

2. Діаграма варіантів використання представляє взаємодію між акторами (користувачами або іншими системами) і системою, показуючи різні варіанти використання і залучених акторів.

3. Діаграма послідовності показує, як об'єкти взаємодіють у певному сценарії варіанту використання, показуючи послідовність повідомлень, якими обмінюються об'єкти в часі.

4. Діаграма діяльності представляє потік діяльності або дій в системі, включаючи рішення, цикли і паралельні дії.

5. Діаграма станів ілюструє стани об'єкта та переходи між цими станами у відповідь на події.

6. Діаграма компонентів показує компоненти системи та зв'язки між ними, включаючи залежності, інтерфейси та взаємодію.

7. Діаграма розгортання зображує фізичне розгортання програмних компонентів на апаратних вузлах, показуючи, як програмні артефакти розподіляються по мережі.

8. Діаграма пакетів організовує елементи системи в пакети, показуючи залежності між пакетами і структурою системи на більш високому рівні.

Це найбільш часто використовуваних діаграм UML при розробці програмного забезпечення, кожна з яких служить різним цілям у моделюванні та передачі різних аспектів дизайну та архітектури програмної системи.

Залежно від вимог вашого проекту, можна використовувати одну або декілька з цих діаграм для ефективної комунікації та документування дизайну вашої системи.

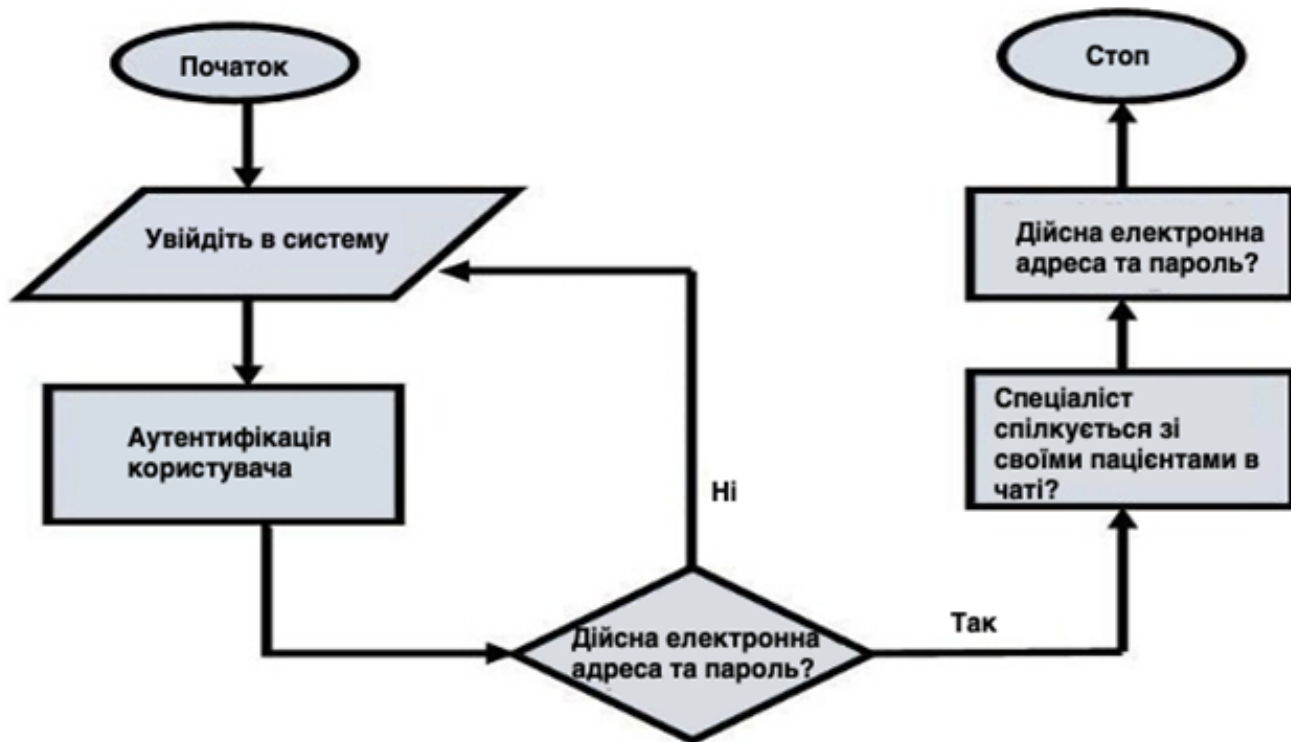


Рисунок 3.4 - Блок-схеми алгоритму роботи застосунку, що демонструє діяльність лікаря за допомогою мобільного застосунку

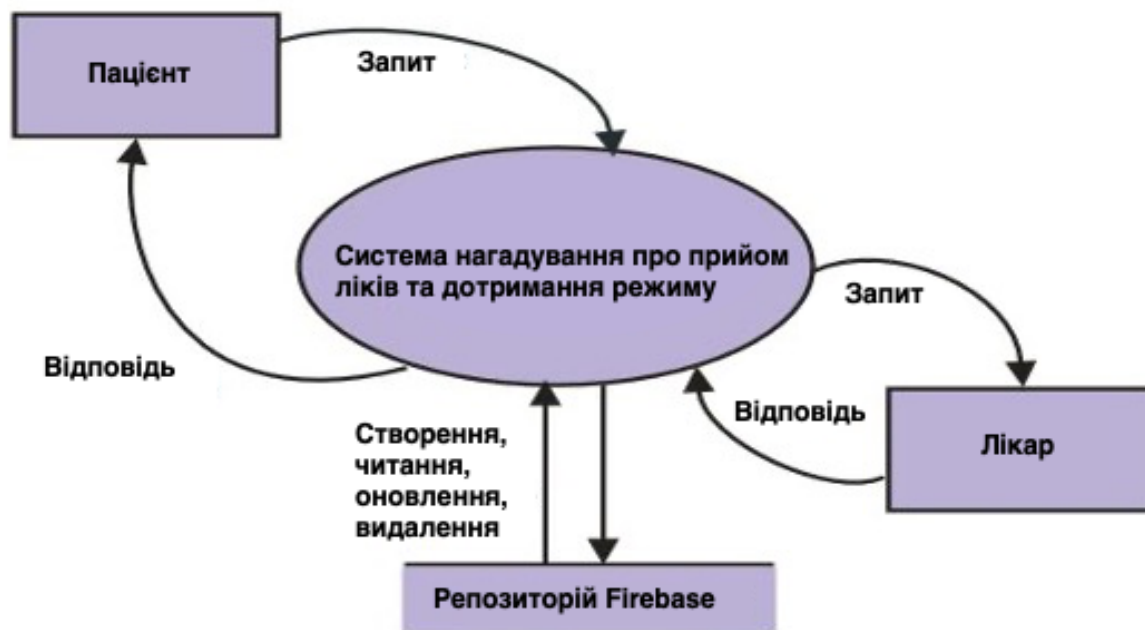


Рисунок 3.5 - Блок-схеми алгоритму руху даних в системі

3.1.1 Створення діаграми варіантів

Створення діаграми варіантів використання для розробки програмного забезпечення передбачає визначення різних суб'єктів, що взаємодіють з системою, і різних варіантів використання (функціональних вимог), пов'язаних з цією взаємодією.

UML діаграма варіантів використання системи подано на рисунку 3.4.

Діаграма включає в себе:

1. Актор (Actor). Актор представляє користувача або іншу систему, що взаємодіє з програмним забезпеченням.
2. Варіант використання. Варіант використання представляє конкретну функціональність або функцію, що надається системою.
3. Асоціація. Лінії асоціації з'єднують акторів з варіантами використання, щоб вказати, які актори беруть участь у кожному варіанті використання.
4. Зв'язок включення. Зв'язок включення означає, що один варіант використання включає функціональність іншого варіанта використання. Наприклад, «Створити звіт» включає «Переглянути звіт».
5. Розширювальний зв'язок. Зв'язок «розширювати» означає, що один варіант використання може розширювати інший варіант використання за певних умов. Наприклад, «Скасувати замовлення» може розширити «Розмістити замовлення» за певних умов.

Розглянемо акторів системи.

Актор. Основним актором є «Пацієнт».

Лікар: Вторинний актор, який може взаємодіяти з системою для призначення або оновлення інформації про ліки для пацієнта.

Випадки використання:

1. Запис ліків. Дозволяє пацієнту записувати інформацію про призначені йому ліки.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Переглянути історію прийому ліків. Дозволяє пацієнту переглянути історію прийому ліків.
3. Встановити нагадування про прийом ліків. Дозволяє пацієнту встановити нагадування про прийом ліків.
4. Переглянути інформацію про ліки. Дозволяє пацієнту переглянути детальну інформацію про кожен препарат.
5. Оновити інформацію про ліки. Дозволяє пацієнту оновити інформацію про ліки відповідно до призначень лікаря.
6. Створити звіт про ліки. Дозволяє пацієнту створити звіт про історію прийому ліків.



Рисунок 3.6 – UML діаграма варіантів використання системи

3.1.2 Створення діаграми класів

Створення діаграми класів (Class Diagram) полягає в моделюванні структури системи шляхом ідентифікації класів, їх атрибутів та методів, а також визначення

взаємозв'язків між цими класами. Ця діаграма допомагає візуалізувати структуру програми або системи та з'ясувати, як вони взаємодіють між собою.

Основні кроки створення діаграми класів:

1. Ідентифікація класів. Визначення основних класів, які представляють сутності або об'єкти у вашій системі. Це можуть бути класи, що відображають реальні об'єкти (наприклад, Клієнт, Замовлення, Продукт) або абстрактні концепції (наприклад, Сервіс, Менеджер).

2. Визначення атрибутів. Для кожного класу визначаються його атрибути - характеристики або властивості цього класу. Наприклад, для класу "Клієнт" атрибутами можуть бути ім'я, адреса, номер телефону.

3. Визначення методів. Для кожного класу визначаються його методи - операції або функції, які можна виконувати над об'єктами цього класу. Наприклад, для класу "Клієнт" можуть бути методи для додавання, видалення або оновлення інформації про клієнта.

4. Визначення взаємозв'язків. Встановлення зв'язків між класами, які показують, як об'єкти цих класів взаємодіють між собою. Зв'язки можуть бути асоціаціями, агрегаціями, композиціями, успадкуванням тощо.

5. Додавання додаткової інформації. Додавання додаткової інформації до класів, такої як модифікатори доступу, типи даних, параметри тощо.

6. Уточнення та перевірка. Перегляд та уточнення діаграми з урахуванням вимог та цілей моделювання. виправлення помилок та уточнення діаграми за необхідності.

Діаграма класів представляє різні сутності та їхні атрибути в системі нагадування про ліки.

Розглянемо основні класи системи.

Пацієнт. Представляє пацієнта з такими атрибутами, як patientID, ім'я, вік, стать та contactInfo.

Лікар. Представляє лікаря з такими атрибутами, як doctorID, ім'я та спеціальність.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

Препарат. Представляє ліки з такими атрибутами, як medicationID, назва, дозування, частота та інструкція.

MedicationRecord. Представляє запис про ліки, призначені пацієнту, з такими атрибутами, як recordID, patientID, medicationID, дати та примітки.

Нагадування. Представляє набір нагадувань для пацієнта про прийом ліків з такими атрибутами, як reminderID, patientID, medicationID та reminderTime.

Ці класи відображають зв'язки між пацієнтами, лікарями, ліками, записами про ліки та нагадуваннями в системі.

Діаграма класів представляє різні сутності та їхні атрибути в системі нагадування про ліки.

Клас Пацієнт представляє пацієнта з такими атрибутами, як patientID, ім'я, вік, стать та contactInfo.

Клас Лікар представляє лікаря з такими атрибутами, як doctorID, ім'я та спеціальність.

Клас Препарат представляє ліки з такими атрибутами, як medicationID, назва, дозування, частота та інструкція.

Клас MedicationRecord представляє запис про ліки, призначені пацієнту, з такими атрибутами, як recordID, patientID, medicationID, дати та примітки.

Клас Нагадування представляє набір нагадувань для пацієнта про прийом ліків з такими атрибутами, як reminderID, patientID, medicationID та reminderTime.

Клас нагадування про прийом ліків представляє систему управління нагадуваннями про прийом ліків. Він містить методи для встановлення нагадувань, скасування нагадувань та перегляду існуючих нагадувань. Клас представляє індивідуальні нагадування про ліки. Він містить такі атрибути, як ідентифікатор нагадування, назва ліків, дозування, частота та час.

Ці класи відображають зв'язки між пацієнтами, лікарями, ліками, записами про ліки та нагадуваннями в системі.

Діаграму класів подано на рисунку 3.7.

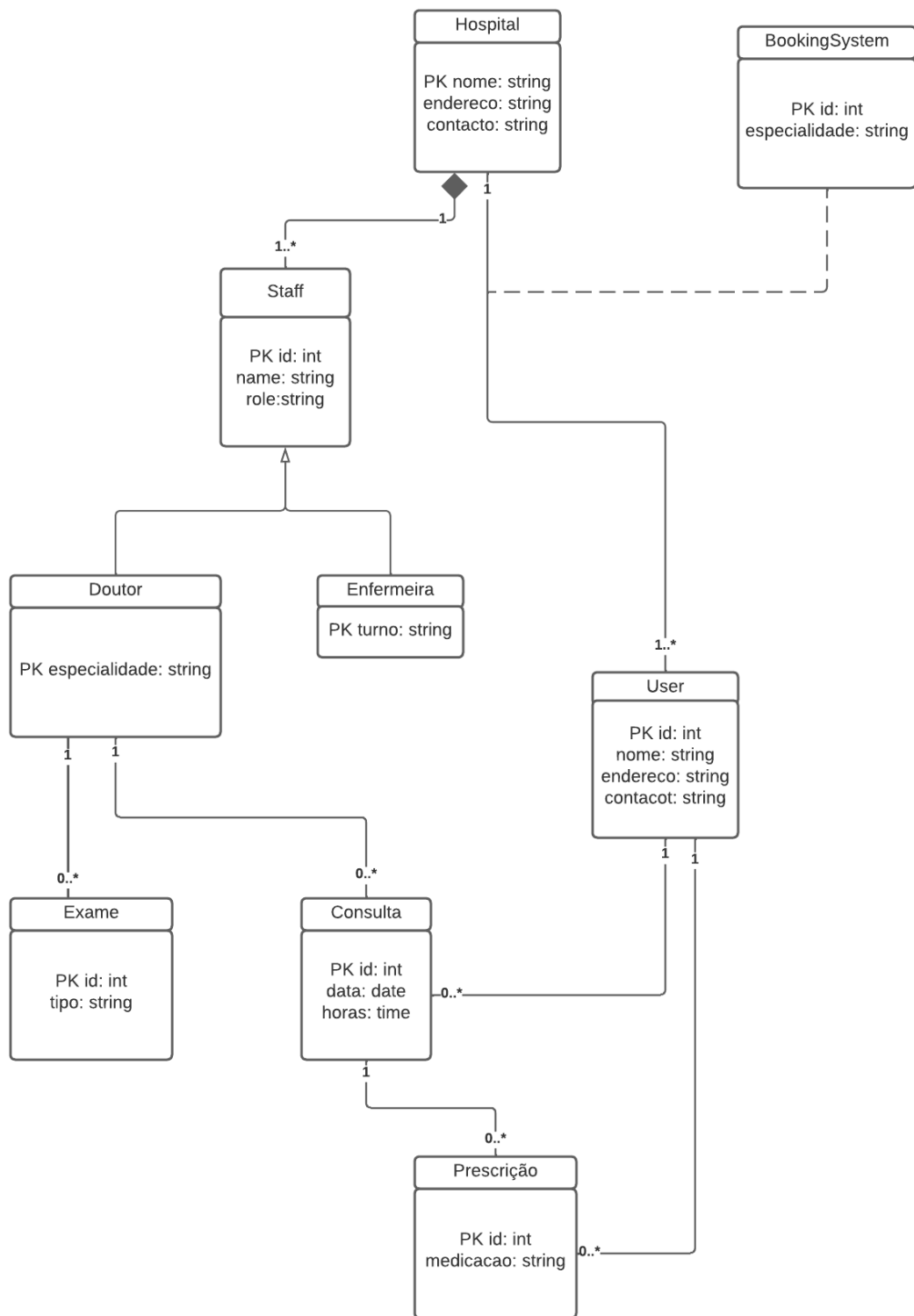


Рисунок 3.7 – Діаграма класів

3.1.3 Створення діаграми послідовності

Створення діаграми варіантів полягає у візуалізації функціональних вимог до системи шляхом ідентифікації основних акторів, що взаємодіють з системою, та визначення сценаріїв використання (варіантів використання), які описують взаємодію між акторами та системою.

Основні кроки створення діаграми варіантів використання:

Визначення акторів: Ідентифікація всіх акторів (користувачів, зовнішніх систем, тощо), які взаємодіють з системою.

Визначення варіантів використання. Визначення основних функціональних можливостей системи або основних сценаріїв використання. Кожен варіант використання описує конкретний функціонал системи або сценарій взаємодії з акторами.

Визначення взаємозв'язків. Встановлення зв'язків між акторами та варіантами використання. Кожен актор може бути пов'язаний з одним або декількома варіантами використання, що відображає їхню взаємодію з системою.

Додавання додаткових відносин. Визначення додаткових відносин між варіантами використання, таких як включення, розширення та узагальнення, якщо вони є необхідними для опису функціональності системи.

Перевірка та уточнення. Перегляд та перевірка діаграми з урахуванням вимог та цілей моделювання. виправлення помилок та уточнення діаграми за необхідності.

Документування. Додавання необхідних коментарів та пояснень до діаграми для забезпечення зрозумілості її змісту.

Таким чином, Пацієнт надсилає запит до MedicineReminderSystem з проханням встановити нагадування для певного препарату, надаючи такі дані, як назва препарату, дозування, частота та час прийому.

Отримавши запит, MedicineReminderSystem викликає метод createReminder об'єкта Reminder, передаючи надані дані.

Об'єкт `Reminder` створює новий об'єкт нагадування і повертає його до `MedicineReminderSystem`.

Потім `MedicineReminderSystem` викликає метод `saveReminder` об'єкта `Reminder`, передаючи об'єкт нагадування як параметр, щоб зберегти його в системі.

Після успішного збереження нагадування об'єкт `Reminder` надсилає повідомлення про успіх назад до `MedicineReminderSystem`.

Нарешті, система `MedicineReminderSystem` повідомляє пацієнта про те, що нагадування було успішно встановлено.

```
Participant: Patient
Participant: MedicineReminderSystem
Participant: Reminder
Patient -> MedicineReminderSystem:
setReminder(medicationName, dosage, frequency, time)
MedicineReminderSystem -> Reminder:
createReminder(medicationName, dosage, frequency, time)
Reminder --> MedicineReminderSystem: reminderObject
MedicineReminderSystem -> Reminder:
saveReminder(reminderObject)
Reminder --> MedicineReminderSystem: success
MedicineReminderSystem --> Patient: Reminder set
successfully
```

3.1.4 Створення діаграми активності

Створення діаграми активності полягає в моделюванні послідовності дій або процесів у системі чи програмі. Ця діаграма використовується для візуалізації послідовності кроків, які виконуються у системі або компоненті. Основні елементи діаграми активності включають активності (дії або кроки), рішення (розгалуження), вибір альтернативних шляхів, паралельні процеси та з'єднання між активностями.

Процес створення діаграми активності зазвичай включає наступні кроки:

1. Визначення цілей: Визначення цілей моделювання і того, що саме потрібно відобразити на діаграмі.

2. Ідентифікація активностей: Визначення основних дій або кроків, які виконуються в процесі. Це можуть бути операції, функції, події тощо.

3. Побудова послідовності. Установлення послідовності кроків, що відбуваються в процесі. Це включає визначення того, яка активність починається першою і яка слідує за нею.

4. Додавання рішень та розгалужень. Вставлення в діаграму вузлів рішень, щоб відобразити вибір альтернативних шляхів або умови, які впливають на подальший хід процесу.

5. Додавання паралельних процесів. Встановлення паралельних активностей, які відбуваються одночасно у системі.

6. Встановлення зв'язків між активностями. Додавання з'єднань між активностями, щоб відобразити послідовність та умови переходу між ними.

7. Перевірка та уточнення. Перегляд та перевірка діаграми з урахуванням вимог та цілей моделювання. виправлення помилок та уточнення діаграми за необхідності.

8. Документування. Додавання необхідних коментарів та пояснень до діаграми для забезпечення зрозумілості її змісту.

Діаграма активності, що ілюструє процес налаштування нагадування про прийом ліків у системі нагадувань про ліки:

Start: Patient initiates setting a reminder

Patient -> Enter medication details: Enter medication name, dosage, frequency, and time for the reminder.

Enter medication details -> Validate details: Ensure all required fields are filled and data is valid.

Validate details -> Reminder set?: If details are valid, proceed to set the reminder.

Reminder set? -> Reminder set: If the reminder is successfully set, end the process.

Reminder set? -> Adjust details: If there is an issue with the details, go back to entering medication details.

Adjust details -> Enter medication details: Allow the patient to correct the entered medication details.

Reminder set -> End: End the process.

На діаграмі активності процес починається, коли пацієнт ініціює встановлення нагадування.

Пацієнт вводить дані про ліки, такі як назва, дозування, частота та час нагадування.

Система перевіряє введені дані, щоб переконатися, що вони є повними та дійсними.

Якщо дані дійсні, система переходить до налаштування нагадування.

Якщо нагадування налаштовано успішно, процес завершується.

Якщо є проблема з даними (наприклад, відсутня інформація), система дозволяє пацієнту відкоригувати дані і повторно вводить дані про ліки.

Цей цикл повторюється доти, доки нагадування не буде успішно встановлено, після чого процес завершується.

3.1.5 Створення діаграми станів

Створення діаграми станів (State Diagram) полягає в моделюванні різних станів, у яких може перебувати об'єкт або система, а також переходів між цими станами в залежності від подій або вхідних сигналів. Ця діаграма допомагає візуалізувати поведінку системи та умови, за яких ця поведінка змінюється.

Основні кроки створення діаграми станів:

1. Ідентифікація станів: Визначення всіх можливих станів, у яких може перебувати об'єкт або система. Наприклад, якщо моделюється стан роботи світлофора, можливі стани можуть бути "червоний", "жовтий" та "зелений".

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Визначення подій: Визначення подій або вхідних сигналів, які можуть спричинити перехід між станами. Наприклад, для світлофора це можуть бути події "натискання на кнопку" або "зміна часу".

3. Визначення переходів: Встановлення переходів між станами в залежності від виникнення певних подій або умов. Наприклад, при виникненні події "час для зміни" світлофор може переходити зі стану "червоний" у стан "жовтий".

4. Визначення дій в станах: Визначення дій або операцій, які виконуються у кожному стані. Наприклад, у стані "жовтий" світлофор може виконувати дію "мигання".

5. Уточнення та перевірка: Перегляд та уточнення діаграми з урахуванням вимог та цілей моделювання. виправлення помилок та уточнення діаграми за необхідності.

На діаграмі станів необхідно:

- Встановити нагадування. Це початковий стан, коли пацієнт встановлює нагадування.

- Нагадування активне. Коли нагадування встановлено, воно переходить у цей стан, де чекає на запланований час для спрацювання.

- Нагадування спрацювало. Коли настає запланований час, нагадування переходить у цей стан, де запускає оповіщення, щоб нагадати пацієнтові про необхідність прийняти ліки.

- Нагадування відхилено. Після того, як пацієнт ігнорує нагадування, воно повертається до стану «Нагадування активне», щоб чекати на наступний запланований час.

3.2 Особливості програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків

Застосунок реалізовано на мові програмування Java, а користувацький інтерфейс - на розширюваній мові розмітки (XML) за допомогою android studio.

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

Розроблена система була розгорнута на мобільних телефонах з ОС Android і протестована. Також було проведено модульне тестування з використанням додатків пацієнта та лікаря в якості модулів. Спочатку було протестовано застосунок пацієнта, а згодом і застосунок лікаря, який також був протестований на валідність та ефективність. Після цього було проведено інтеграційне тестування, після того, як клієнт-сервер Firebase був інтегрований в систему. Також було проведено системне тестування.

На цьому етапі мобільна система нагадування про ліки та охорони здоров'я тестується як єдине ціле. Застосунок запускається на телефоні з ОС Android, відображає календар для пацієнта (рисунок 3.7). На рисунку 3.8 подано деталізовану інформацію щодо ліків. Рисунок 3.9 представляє сторінку налаштування додатку. Рисунок 3.10 подає сторінку сповіщень. Рисунок 3.11 представляє сторінку меню застосунку. Рисунок 3.12 показує сторінку ліків. Вікно старту застосунку подано на рисунку 3.13.

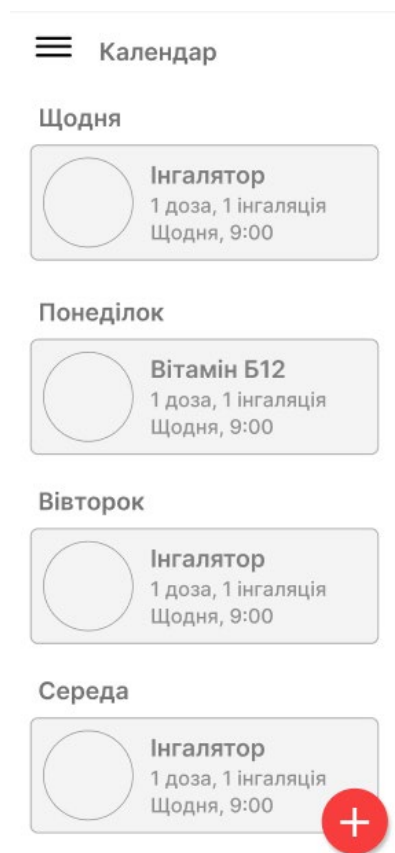


Рисунок 3.7 – Сторінка реєстрації пацієнта

					КВРІСТ 200178 20.01.01 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

< Пеніцилін

Пеніцилін
1 доза
15 мг

Графік
15:45
Щодня

Рисунок 3.8 – Сторінка деталізованої інформації щодо ліків пацієнта

☰ Налаштування

Доступність

Сповіщення

Рисунок 3.8 – Сторінка налаштувань застосунку

< Додати ліки

Назва ліків
Кількість доз
Кількість на дозу

Графік
Час
День тижня

Рисунок 3.9 – Додавання ліків

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ 200178 20.01.01 ПЗ

Арк.
61

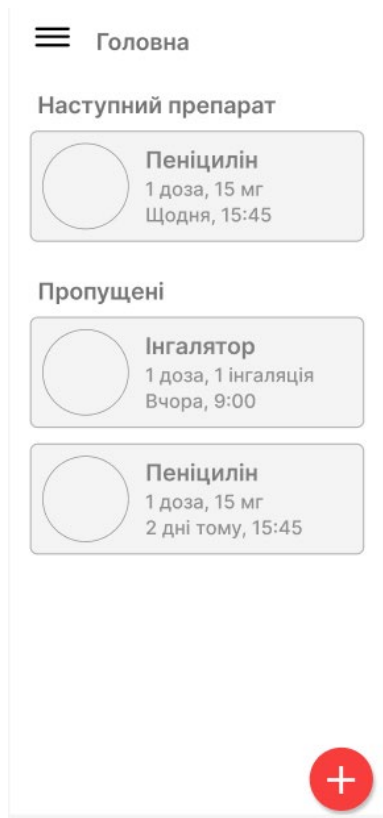


Рисунок 3.10 – Сторінка сповіщень



Рисунок 3.11 – Сторінка меню застосунку

					КВРІСТ 200178 20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

☰ Мої ліки




-  **Інгалятор**
1 доза, 1 інгаляція
Щодня, 9:00
-  **Пеніцилін**
1 доза, 15 мг
Щодня, 15:45
-  **Ксанакс**
1 доза, 30 мг
В міру необхідності



Рисунок 3.12 – Сторінка ліків

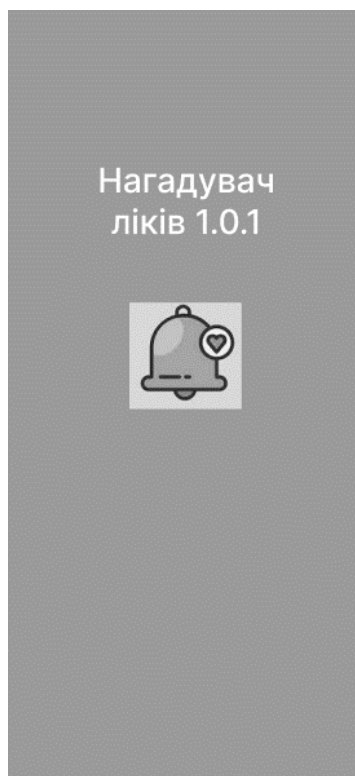


Рисунок 3.13 – Вікно старту застосунку

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ 200178 20.01.01 ПЗ

Арк.
63

3.3 Висновки

В розділі подано основні аспекти програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків.

Зокрема, наведено процес проектування архітектури програмного забезпечення інформаційної системи.

Подано процес розроблення UML діаграм, а саме : створення діаграми варіантів, діаграми класів, діаграми послідовності, діаграми активності діаграми станів.

В розділі також представлено особливості програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків.

					КвРІСТ 200178 20.01.01 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У першому розділі проведено аналіз проведено теоретичний аналіз та виявлено недоліки існуючих рішень, що привело до формулювання завдання щодо дослідження шляхів їх вирішення. Також проведено аналіз відомих інформаційних систем обліку медичної інформації для прийому ліків мають свої недоліки.

Зроблено висновок, що подолання недоліків важливо розробити нову інформаційну систему з урахуванням конкретних потреб медичних закладів та забезпечити їхню високу якість, безпеку та користувацьку зручність.

У другому розділі представлено опис користувальницьких вимог до інформаційної системи обліку медичної інформації для прийому ліків. Такж у розділі подано функційні вимоги до інформаційної системи, зокрема процедури реєстрація пацієнтів, ведення медичної інформації, призначення лікування, сповіщення про прийом ліків, ведення реєстру ліків, перевірка взаємодії ліків, стеження за вживанням ліків, генерація звітів і аналітика.

У третьому розділі подано основні аспекти програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків. Зокрема, наведено процес проєктування архітектури програмного забезпечення інформаційної системи. Подано процес розроблення UML діаграм, а саме : створення діаграми варіантів, діаграми класів, діаграми послідовності, діаграми активності діаграми станів.

В розділі також представлено особливості програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків.

					КвРІСТ 200178 20.01.01 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Wu X., Li X. Hack Android application and defence. 2014. doi:10.1109/ICCSNT.2013.6967202.
2. Karthick S., Binu S. Android security issues and solutions. 2017. doi:10.1109/ICIMIA.2017.7975551.
3. Android Open Source Project. <https://source.android.com/> (дата звернення 12 березня 2024).
4. Google's Android OS: Past, Present, and Future - PhoneArena. URL: https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273 (дата звернення 12 березня 2024).
5. Alabaster J. Android founder: We aimed to make a camera OS, PC World. 2020. URL: <https://www.pcworld.com/article/2034723/android-founder-we-aimed-to-make-a-camera-os.html> (дата звернення 12 березня 2024).
6. A Murky Road Ahead for Android, Despite Market Dominance - The New York Times. URL: <https://www.nytimes.com/2015/05/28/technology/personaltech/a-murky-road-ahead-for-android-despite-market-dominance.html> (дата звернення 12 березня 2024).
7. Android founder: We aimed to make a camera OS | PCWorld. <https://www.pcworld.com/article/2034723/android-founder-we-aimed-to-make-a-camera-os.html> (дата звернення 12 березня 2024).
8. R. Block 2007 Google is working on a mobile OS, and it's due out shortly, Engadget, 2020. URL: <https://www.engadget.com/2007/08/28/google-is-working-on-a-mobile-os-and-its-due-out-shortly/> (дата звернення 12 березня 2024).
9. McKay M. Can iPhone become your phone? Linksys introduces versatile line for cordless service, Rec. (Bergen County), p. L9, Dec, Accessed: Jul. 17, 2020. URL: <http://record-bergen.vlex.com/vid/iphone-phone-linksys-versatile-cordless-62885923> (дата звернення 12 березня 2024).
10. Ionescu D. Original Android Prototype Revealed During Google, Oracle Trial," PC World, Apr. 2022. URL:

					КвРІСТ 200178 20.01.01 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

https://www.pcworld.com/article/254539/original_android_prototype_revealed_during_google_oracle_trial.html (дата звернення 12 березня 2024).

11. Gao R. Android and its first purchasable product, the T-Mobile G1, celebrate their 8th birthdays today. *Android Police*. 2020. URL: <http://www.androidpolice.com/2016/09/23/android-first-purchasable-product-t-mobile-g1-celebrate-8th-birthdays-today/> (дата звернення 12 березня 2024).

12. Menon M. K. Android Nougat: Here's why Google names the OS after sweets. *The Indian Express*, 2020. URL: <http://indianexpress.com/article/lifestyle/food-wine/from-donut-to-nougat-why-are-android-versions-named-after-sweets-2887237/> (дата звернення 12 березня 2024).

13. Golhar R. V., Vyawahare P. A., Borghare P. H. Manusmare A. Design and implementation of android base mobile app for an institute. *Int. Conf. Electr. Electron. Optim. Tech.* 2022. pp. 3660–3663, doi: 10.1109/ICEEOT.2016.7755391.

14. Dalvik Virtual Machine - javatpoint. URL: <https://www.javatpoint.com/dalvik-virtual-machine> (accessed Jul. 17, 2020).

15. Android - Architecture - Tutorialspoint. URL: https://www.tutorialspoint.com/android/android_architecture.htm (дата звернення 12 березня 2024).

16. The Java™ Tutorials. URL: <https://docs.oracle.com/javase/tutorial/index.html> (дата звернення 12 березня 2024).

17. Kalogiannakis M., Stamatios P., Alkinoos-Ioannis Z. Gamification in science education. A systematic review of the literature. *Education Sciences*. 2021. Vol. 11, no. 1. <https://doi.org/10.3390/educsci11010022> .

18. Morin É. On the importance of the relationship to knowledge in science education. *Cultural Studies of Science Education*. 2019. Vol. 14, no. 3. Pp. 621-625. <https://doi.org/10.1007/s11422-018-9881-1> .

19. Saenpuk N., Chaiyapong R. Development of 8 students' scientific concept in cause of moon phase by using metacognitive strategy. *AIP Conference Proceedings*, 2019. vol. 2081, no. 1. <https://doi.org/10.1063/1.5094007> .

					КВРІСТ 200178 20.01.01 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

20. Amadi C. S. The integration of 21st-century skills in science: A Case Study of Canada and the USA. *Education and Urban Society*. 2023. Vol. 55, no. 1. Pp. 56-87. <https://doi.org/10.1177/00131245211062531> .

21. Chastenay P. Martin R. Development and validation of the moon phases concept inventory for middle school. *Physical Review Physics Education Research*. 2020. Vol. 16, no. 2 . pp. 020107. <https://doi.org/10.1103/PhysRevPhysEducRes.16.020107>.

22. , Nur Izyan Shafinaz A., Haron Z. Penggunaan Model 3 Dimensi Fasa-Fasa Bulan Untuk Meningkatkan Penguasaan Murid Bagi Konsep Fasa-Fasa Bulan. *International Journal*. 2019. Vol. 2, no. 8 . pp. 64-72. <https://doi.org/10.35631/IJMTSS.28007>.

23. Jack H., Stevenson W. A., Jonathon P., Byungdoo K., Swati P., Yilu S., T. J. Virtual reality as a teaching tool for moon phases and beyond. arXiv preprint arXiv:1807.11179. 2018.

24. Sugrah N. Implementasi teori belajar konstruktivisme dalam pembelajaran sains., *Kajian Ilmiah Mata Kuliah Umum*. 2019. Vol. 19, no. 2. Pp. 121-138. <https://doi.org/10.21831/hum.v19i2.29274> .

25. Galperin D., Andrés R. Reference frames and astronomy teaching: the development of a topocentric approach to the lunar phases. *Science Education International*. 2019. Vol. 30, no. 1. Pp. 28-37. <https://doi.org/10.33828/sei.v30.i1.4> .

26. Hamilton D, McKechnie J., Edgerton E. Immersive virtual reality as a pedagogical tool in education: a systematic literature review of quantitative learning outcomes and experimental design. *Journal of Computers in Education*. 2021. Vol. 8, no. 1 (0): 1-32. <https://doi.org/10.1007/s40692-020-00169-2>.

27. Yehuda P., Pundak D., Weiser-Biton R. From a passive information consumer to a critically thinking learner. *Technology, Pedagogy and Education*. 2020. Vol. 29, no. 1. Pp. 73-88. <https://doi.org/10.1080/1475939X.2019.1699853> .

28. Nicolette H., Cailee E. Developing cognitive skills through active learning: a systematic review of health care professions. *Athletic Training Education Journal*. 14, no. 2. Pp. 135-148. <https://doi.org/10.4085/1402135/>.

					КВПІСТ 200178 20.01.01 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

29. Goldschmidt K. The COVID-19 pandemic: Technology use to support the wellbeing of children. *Journal of Pediatric Nursing*. Vol. 53. Pp. 88-90. <https://doi.org/10.1016/j.pedn.2020.04.013>.

30. Lin M.H., Chen H.C., Liu K.S. A study of the effects of digital learning on learning motivation and learning outcome. *Eurasia Journal of Mathematics, Science and Technology Education*. Vol. 13, No. 7. Pp. 3553-3564. <https://doi.org/10.12973/eurasia.2017.00744a>.

31. Goundar M.S., Kumar B.A. The use of mobile learning applications in higher education institutes. *Education and Information Technologies*. Pp. 1-24. <https://doi.org/10.1007/s10639-021-10611-2>.

32. Abd Rahman N., Mohd Rafhan Syamil F.I., Rodzman S.B. Development of mobile application for Malay translated hadith search engine. *Indonesian Journal of Electrical Engineering and Computer Science*. Vol. 20, No. 2. Pp. 932-938. <https://doi.org/10.11591/ijeecs.v20.i2.pp932-938>.

33. Yahaya H., Sardi J., Radzi M., Youssef I., Abdelhamid F.P. Development of a mobile application in Arabic language learning in Malaysia: An overview. *International Journal of Academic Research in Business and Social Sciences*. Vol. 9, No. 7. <https://doi.org/10.6007/IJARBSS/v9-i7/6403>.

34. Shawai Y.G., Almaiah M.A. Malay language mobile learning system (MLMLS) using NFC technology. *International Journal of Education and Management Engineering*. Vol. 8, No. 2. P. 1. <https://doi.org/10.5815/ijeme.2018.02.01>.

35. Razalli A.R., Mamat N., Razali N., Yasin M.H.M, Lakulu M., Hashim A.T.M., Ariffin A. Development of prayer mobile application software for the hearing impaired (Deaf) based on Malaysian sign language. *International Journal of Academic Research in Business and Social Sciences*. Vol. 11, No. 6. Pp. 1108-1122. <https://doi.org/10.6007/IJARBSS/v11-i6/10243>.

36. Ariffin M.M., Halim F.A.A., Abd N. Mobile application for dyscalculia children in Malaysia. *In Proceedings of the 6th International Conference on Computing & Informatics*. 2017. Pp. 467-472.

					КВПІСТ 200178 20.01.01 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

37. Muslimin MS, Nordin NM, Mansor AZ, Yunus MM. The design and development of MobiEko: A mobile educational app for microeconomics module. *Malaysian Journal of Learning and Instruction*. Pp. 221-255. <https://doi.org/10.32890/mjli.2017.7804>.

38. Ismail A., Syed Abdul Kadir S.A., Abdul Aziz A., Mokshin M., Mohd Lokman A. ITourism travel buddy mobile application. In *2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST)*. 2016. Pp. 82-87. <https://doi.org/10.1109/NGMAST>.

39. Chen Y.S., Wong J.E., Ayob A.F., Othman NE, Poh BK. Can Malaysian young adults report dietary intake using a food diary mobile application? A pilot study on acceptability and compliance. *Nutrients*. Vol. 9, No. 1. P. 62. <https://doi.org/10.3390/nu9010062>.

40. Rousseau R, Zhang L, Hu X. Knowledge integration: Its meaning and measurement. *Springer Handbook of Science and Technology Indicators*. 2019. Pp. 69-94. https://doi.org/10.1007/978-3-030-02511-3_3.

41. Yang R, Wong CWY, Miao X. Analysis of the trend in the knowledge of environmental responsibility research. *Journal of Cleaner Production*. Vol. 278. P. 123402. <https://doi.org/10.1016/j.jclepro.2020.123402>.

42. Harlow I, Fischer L, Wang L. Beyond the foundations: Improving higher-order understanding through foundational knowledge. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Pp. 628-633. Association for the Advancement of Computing in Education (AACE), 2019.

43. Mishra P, Mehta R. What educators get wrong about 21st century learning. In *Society for Information Technology & Teacher Education International Conference*. 2016. Pp. 2968-2975. Association for the Advancement of Computing in Education (AACE),. <https://doi.org/10.1080/21532974.2016.1242392>.

44. Eva R. Pengaruh aplikasi model ASSURE terhadap motivasi dan hasil belajar peserta didik dalam pembelajaran Geografi. *Jurnal Geografi Gea*. Vol. 15, No. 2. <https://doi.org/10.17509/gea.v15i2.3543>.

					КВПІСТ 200178 20.01.01 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

45. Hansen D.M., Holland M.M., Munn J. Teaching note-A call for social work education modification: Moving toward a model of interprofessional education. *Journal of Social Work Education*. Vol. 56, No. 3. Pp. 595-601. <https://doi.org/10.1080/10437797.2019.1661902>.

46. Suhaizal H, Abdul Rahman KA, Khamis N, Shukor UH, Che Lah NH, Zulkifli NN. The design and development of augmented reality (AR) application for Internet evolution learning topics. *International Journal of Interactive Mobile Technologies*. Vol. 17, No. 5. <https://doi.org/10.3991/ijim.v17i05.36483>.

47. Criollo-C S, Guerrero-Arias A, Jaramillo-Alcázar Á, Luján-Mora S. Mobile learning technologies for education: Benefits and pending issues. *Applied Sciences*. Vol. 11, No. 9. P. 4111. <https://doi.org/10.3390/app11094111>.

48. Hao C, Zheng A, Wang Y, Jiang B. Experiment information system based on an online virtual laboratory. *Future Internet*. Vol. 13, No. 2. P. 27. <https://doi.org/10.3390/fi13020027>.

49. Jalinus N. Developing blended learning model in vocational education based on 21st century integrated learning and industrial revolution 4.0. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*. Vol. 12, No. 8. Pp. 1239-1254.

50. Vlachopoulos D, Makri A. Quality teaching in online higher education: The perspectives of 250 online tutors on technology and pedagogy. *International Journal of Emerging Technologies in Learning (IJET)*. Vol. 16, No. 6. Pp. 40-56. <https://doi.org/10.3991/ijet.v16i06.20173>.

51. Mat R.C., ed. *Inovasi dan Kreativiti Berteraskan Teknologi* (UUM Press). UUM Press, 2020.

52. Chi MTH. Translating a theory of active learning: An attempt to close the research-practice gap in education. *Topics in Cognitive Science*. Vol. 13, No. 3. Pp. 441-463. <https://doi.org/10.1111/tops.12539>.

53. Marion T.J., Fixson S.K. The transformation of the innovation process: How digital tools are changing work, collaboration, and organizations in new product

					КВПІСТ 200178 20.01.01 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

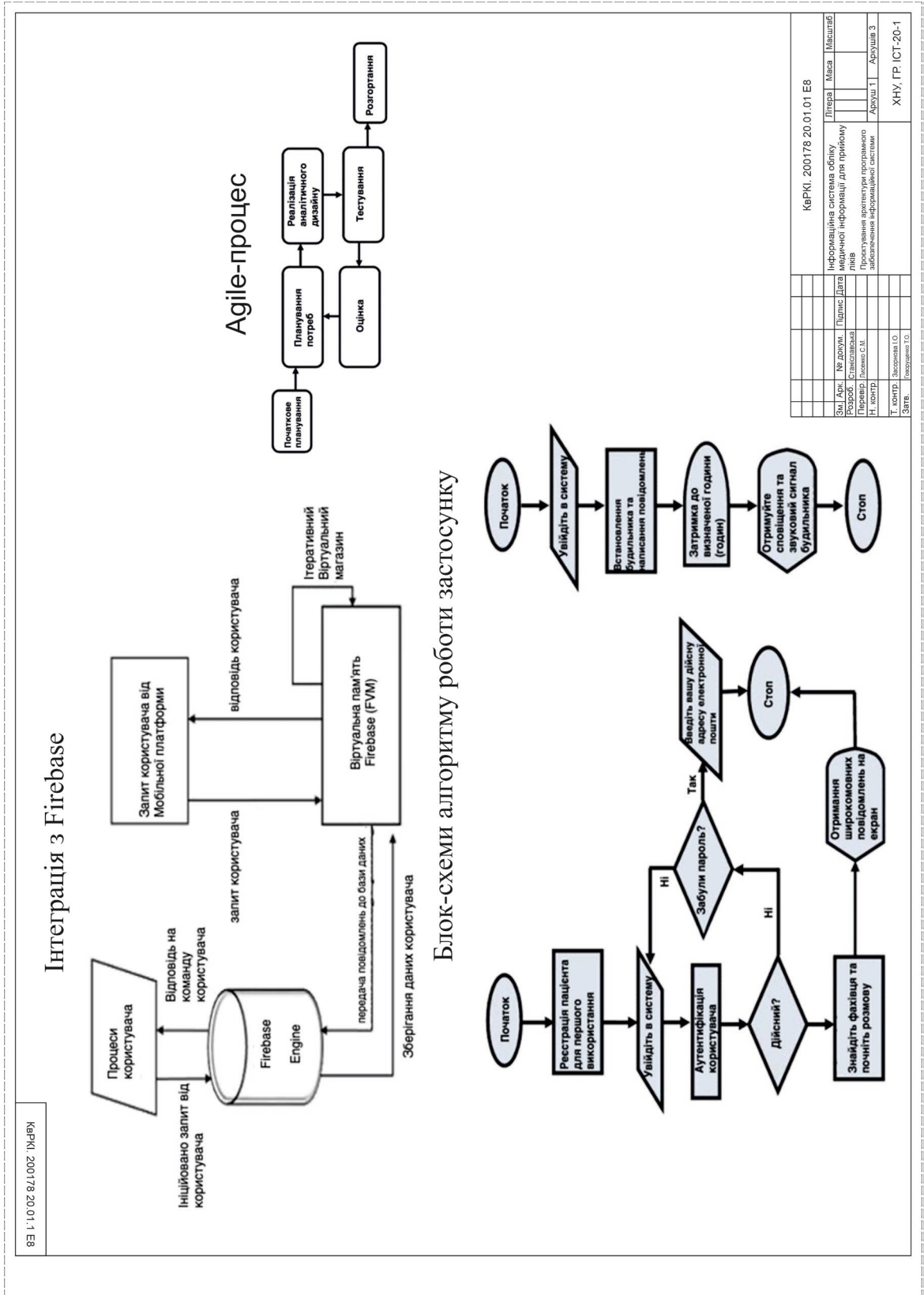
development. *Journal of Product Innovation Management*. Vol. 38, No. 1. Pp. 192-215.

<https://doi.org/10.1111/jpim.12547>.

					КвРІСТ 200178 20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

Додаток А (обов'язковий)

Копія креслення «Проектування архітектури програмного забезпечення інформаційної системи»



Додаток Б (обов'язковий)

Копія креслення «UML діаграми»

Блок-схеми алгоритму роботи застосунку, що демонструє діяльність лікаря за допомогою мобільного застосунку

Блок-схеми алгоритму руху даних в системі

Діаграма класів

UML діаграма варіантів використання системи

КерРКІ. 200178.20.01.1.Е8

Зна. Адр.	№ докум.	Підпис	Дата	Літера	Масштаб
Розроб.	Статус/авт.	Ліценз. С.П.			
Н. контр.	Ліценз. С.П.	УML-діаграми. Схеми		Аркуш 2	Аркуш 3
Т. констр.	Закорчука І.О.				
Затв.	Покорчука І.О.				


ХНУ, ГР. ІСТ-20-1

Додаток В (обов'язковий)


Копія креслення «Програмна реалізація інформаційної системи»

КвРКІ. 200778.20.01.1.Е8


Сторінка реєстрації пацієнта



Сторінка деталізованої інформації щодо ліків пацієнта




Додавання ліків




КвРКІ. 200778.20.01.1.Е8


Сторінка сповіщень



Сторінка ліків



Вікно старту застосування



КвРКІ. 200778.20.01.1.Е8										
Зм. Акт.	№ докум.	Підпис	Дата	Після	Місяць	Місяць	Місяць	Місяць	Місяць	Місяць
Розроб.	Статус/авт.	Ініціал	Дата	Лінійка	Лінійка	Лінійка	Лінійка	Лінійка	Лінійка	Лінійка
Н. контр.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.	Лінійка С.М.
Т. контр.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.	Зображення І.О.
Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак
Інформаційна система «БІСІ» медичної інформації для прибіччя ліків										
Програма реалізації інформаційної системи										
Архив 3										
Архив 3										
ХНУ, ГР. ІСТ-20-1										

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 10%

ID: 128036 Назва: БКР Інформаційна система обліку медичної інформації для прийому ліків Додано в БД: 2024-06-02 Автора: Г. О. Станіславська Керівники: С.М. Лісенко Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	86745	742	1523 (2%)	21 (3%)

Джерело плагиату

ID	Опис	Наявність плагиату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КІ

Дата перевірки:
02.06.2024 16:26:46 EEST

Дата звіту:
02.06.2024 16:29:36 EEST

ID перевірки:
1016311023

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Станіславська Інформаційна система обліку медичної інформації для прийому ліків

Кількість сторінок: 73 Кількість слів: 12757 Кількість символів: 103275 Розмір файлу: 1.19 MB ID файлу: 1016107794

2.59% Схожість

Найбільша схожість: 0.63% з джерелом з Бібліотеки (ID файлу: 1014982266)

2.25% Джерела з Інтернету 61 Сторінка 75

1.46% Джерела з Бібліотеки 60 Сторінка 75

0.31% Цитат

Цитати 3 Сторінка 76

Посилання 1 Сторінка 76

0% Вилучень

Немає вилучених джерел

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Станіславська Галина Олегівна

Тема: Інформаційна система обліку медичної інформації для прийому ліків

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 62

1. Короткий зміст роботи та прийнятих рішень: Метою дипломної роботи є розроблення інформаційної системи обліку медичної інформації для прийому ліків.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У межах розділу 1 проведено було проведено теоретичний аналіз та виявлено недоліки існуючих рішень, що привело до формулювання завдання щодо дослідження шляхів їх вирішення. Також проведено аналіз відомих інформаційних систем обліку медичної інформації для прийому ліків мають свої недоліки. В другому розділі кваліфікаційної роботи представлено опис користувальницьких вимог до інформаційної системи обліку медичної інформації для прийому ліків. Також у розділі подано функційні вимоги до інформаційної системи, зокрема процедури реєстрація пацієнтів, ведення медичної інформації, призначення лікування, сповіщення про прийом ліків, ведення реєстру ліків, перевірка взаємодії ліків, стеження за вживанням ліків, генерація звітів і аналітика. В розділі подано основні аспекти програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків. Зокрема, наведено процес проектування архітектури програмного забезпечення інформаційної системи. Подано процес розроблення UML діаграм, а саме: створення діаграми варіантів, діаграми класів, діаграми послідовності, діаграми активності діаграми станів. В розділі також представлено особливості програмної реалізації інформаційної системи обліку медичної інформації для прийому ліків.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага аналізу аналогам інформаційних систем.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному технічному рівні.

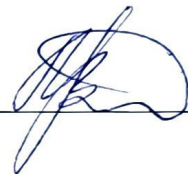
8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та
робототехніки, професор, д.т.н. Мартинюк В.В.

“5” 06 2024 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говоруненко Т. О.

Станіславська Г.О.

ІІІ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-20-1

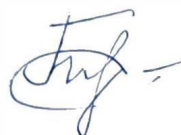
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система обліку медичної інформації для прийому ліків

Автор: Станіславська Галина Олегівна

Спеціальність: 126– Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Лисенко Сергій Миколайович, д.т.н., професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноповживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.59% і адресується до 61 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



С.М. Лисенко

Є.Г. Гнатчук

Т. О. Говорущенко