

Технічні науки

УДК.004.9-004.58

Носовець Олена Костянтинівна

кандидат технічних наук,

доцент кафедри біомедичної кібернетики

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Nosovets Olena

Candidate of Technical Sciences,

Associate Professor of Biomedical Cybernetics

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

Волосожар Дар'я Владиславівна

студентка

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Volosozhar Daria

Student of the

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

ANDROID ДОДАТОК ДЛЯ ПРИВАТНОГО ЛІКАРЯ

ANDROID ПРИЛОЖЕНИЕ ДЛЯ ЧАСТНОГО ВРАЧА

ANDROID APPLICATION FOR A PRIVATE DOCTOR

***Анотація.** Актуальність розробки додатку для приватного лікаря полягає в тому, що існує велика частина приватних лікарів, які працюють з*

виїздом додому та мають заповнювати картки пацієнтів. Для того, аби полегшити їм роботу, було розроблено додаток, який допомагає їм це робити.

Ключові слова: приватний лікар, додаток, платформа Android, дані пацієнта, зберігання даних пацієнта.

Аннотація. Актуальность разработки приложения для частного врача заключается в том, что существует большая часть частных врачей, которые работают с выездом на дом и должны заполнять карточки пациентов. Для того чтобы облегчить им работу, было разработано приложение, которое помогает им это делать.

Ключевые слова: приватный врач, приложение, платформа Android, данные пациента, хранение данных пациента.

Summary. The relevance of developing a supplement for a private doctor is that there are a large number of private doctors who work on the go and have to fill out patient cards. To make their job easier, an app has been developed to help them do so.

Key words: private doctor, application, Android platform, patient data, patient data storage.

Вступ. Впровадження ІТ у сферу охорони здоров'я дозволяє покращити якість обслуговування, помітно прискорити роботу персоналу та знизити витрати на обслуговування для пацієнтів. Інформаційні технології у медицині дають можливість проводити якісне спостереження за станом пацієнтів. Ведення електронних медичних карток дозволяє скоротити час співробітників клінік, витрачений на оформлення різних бланків. Уся інформація про пацієнта представлена в одному документі, доступному медичному персоналу закладу. Усі дані про обстеження та результати процедур також вводяться

безпосередньо до електронної медичної картки. Це дає можливість іншим фахівцям оцінити якість призначеного лікування, виявити неточність діагностики.

Інформаційні технології в медицині та охороні здоров'я допомагають вирішити такі завдання:

- вести облік пацієнтів;
- спостерігати дистанційно за станом;
- зберігати та передавати результати діагностичних обстежень;
- контролювати правильність призначеного лікування;
- проводити віддалене навчання;
- давати консультації малодосвідченим працівникам.

Застосування ІТ у медицині дозволяє лікарям проводити онлайн-консультації у будь-який зручний час. У цьому підвищується доступність медичних послуг. Люди можуть отримати кваліфіковану допомогу від досвідчених лікарів віддалено. Таким чином, пацієнтам або лікарям не потрібно долати великі відстані, щоб отримати консультацію. Лікар може за допомогою сучасних інформаційних технологій оцінити стан пацієнта, провести його огляд та ознайомитись з усіма результатами його обстежень [1, с. 16].

Огляд обраних технологій. Java — це мова програмування розроблена Sun Microsystems. Вона схожа на C і C++, але є набагато простішою. Java була спочатку розроблена Джеймсом Гослінгом разом зі своїми колегами з Sun Microsystems на початку 1990-х років [2, с. 42].

Переваги мови Java:

1. Багатопотокова

Можливість багатопотокової роботи вбудовано безпосередньо в мову Java. Це означає, що можна створювати дуже інтерактивні додатки з низкою одночасних потоків діяльності.

2. Об'єктно-орієнтована мова програмування

На основі C++, який є напівоб'єктно-орієнтованим, Java розширює функціональні можливості, щоб стати повністю об'єктно-орієнтованою мовою програмування. Деякі з найважливіших особливостей, які роблять його об'єктно-орієнтованим: абстракція, інкапсуляція, успадкування, поліморфізм.

3. Полегшене управління пам'яттю

На відміну від C і C++, Java спирається на просту модель управління пам'яттю.

4. Безпечна

Функції безпеки вбудовані в систему мови та виконання. Вони включають перевірку часу виконання та статичну перевірку типів під час компіляції.

5. Проста для вивчення

Легкість читання та письма робить будь-яку мову простою. Це справедливо для Java, оскільки вона має менш неоднозначну синтаксичну термінологію. Будь-хто може почати з Java з розумінням основних принципів програмування.

Java надзвичайно портативна. Одна й та сама програма Java працюватиме однаково на будь-якому комп'ютері, незалежно від апаратних можливостей чи операційної системи, якщо в ній є інтерпретатор Java [3, с. 396-397].

SQLite — це програмна бібліотека, яка забезпечує систему управління реляційною базою даних. SQLite має легкий з точки зору налаштування, адміністрування бази даних та необхідних ресурсів. SQLite має такі помітні особливості: автономність, нульова конфігурація, транзакційна.

Програми, що використовують SQLite, отримують такі переваги як підвищеної стійкості та скорочення часу розробки. Використання бази даних SQLite замість текстових файлів для конфігурації та зберігання об’єднує доступ до даних на всіх пристроях і допомагає підтримувати стабільну продуктивність [4, с. 256].

SQLite добре працює в середовищах, де кінцеві користувачі не повинні знати про існування бази даних. SQLite не потребує обслуговування або адміністрування, що робить його ідеальним вибором для мобільних додатків.

SQLite також добре працює у ролі бекенду для серверних програм і веб-сайтів. SQLite може збігатися або навіть перевищувати продуктивність інших механізмів баз даних. Він запускає один процес без мережевого обміну, усуваючи традиційні накладні витрати на базу даних [5, с. 89].

Android – популярна комп’ютерна платформа, заснована на операційній системі Linux. Початкова комерційна версія Android вийшла на ринок у 2008 році у вигляді платформи для мобільних телефонів. Дивлячись на широту можливостей Android, його легко сплутати з операційною системою для настільних комп’ютерів. Android — це багат шарове середовище, яке побудовано на основі ядра Linux і включає багаті функціональні можливості. Android має величезний набір варіантів підключення, включаючи Wi-Fi, Bluetooth, NFC і, звичайно, стільникові з’єднання в кожній мережі, яку ви можете собі уявити. Важливим аспектом середовища програм Android є те, що програми Android історично писалися мовою програмування Java [6, с. 256].

Опис розробки додатку. З метою якісної розробки додатку потрібно розробити модель життєвого циклу. Модель життєвого циклу проекту (Software life cycle model, SLCM) - це структура, що визначає послідовність виконання та взаємозв’язку процесів, дій та задач на протязі виконання проекту.

1. Ініціювання.

На цьому етапі визначається бізнес-проблема, яку потрібно буде вирішити. Бізнес – проблема: приватні лікарі, які виїжджають до пацієнта на дім, не мають системи, яка б обслуговувала данні, що вони отримують та використовують у подальшому. Для цього їм потрібне певне «сховище» для цієї інформації. Також потрібно мати функцію додавати інформацію до цього «сховища». Таким чином, потрібно розробити портативний програмний продукт, що буде дозволяти лікарям створювати записи про пацієнтів (додавати інформацію) та зберігати цю інформацію на електронному носії. Під час цього етапу також визначається сфера обслуговування, для ДПЛ це буде медична сфера.

2. Планування.

Під час цієї частини розробка проекту розбивається на менші завдання та розробляється графік виконання завдань. Під час цього етапу потрібно визначити часові рамки виконання кожного етапу проекту, завдання, які необхідно виконати, та можливі обмеження.

3. Виконання.

На цьому етапі виконується сукупність дій, що призводять до втілення задуму замовника та отримання запланованого продукту. Етап «виконання» при розробці додатку виглядав наступним чином: розробка бд, проектування та реалізація користувацького інтерфейсу, розробка архітектури додатку та подальше тестування готового продукту.

4. Фаза впровадження.

На цьому етапі надаються остаточні результати, аналізується пророблена робота. Також аналізується ефективність проекту, а саме визначається чи були досягнуті всі поставлені цілі (завдання виконані вчасно) і чи вирішено початкову проблему за допомогою розробленого продукту. Також, аналізуючи

період розробки проекту, можна винести корисну інформацію, наприклад, як краще планувати час [7, с. 345-347].

Останні три фази, а саме фазу планування, виконання та впровадження можна зобразити за допомогою діаграми «дерево цілей» (рис.1).

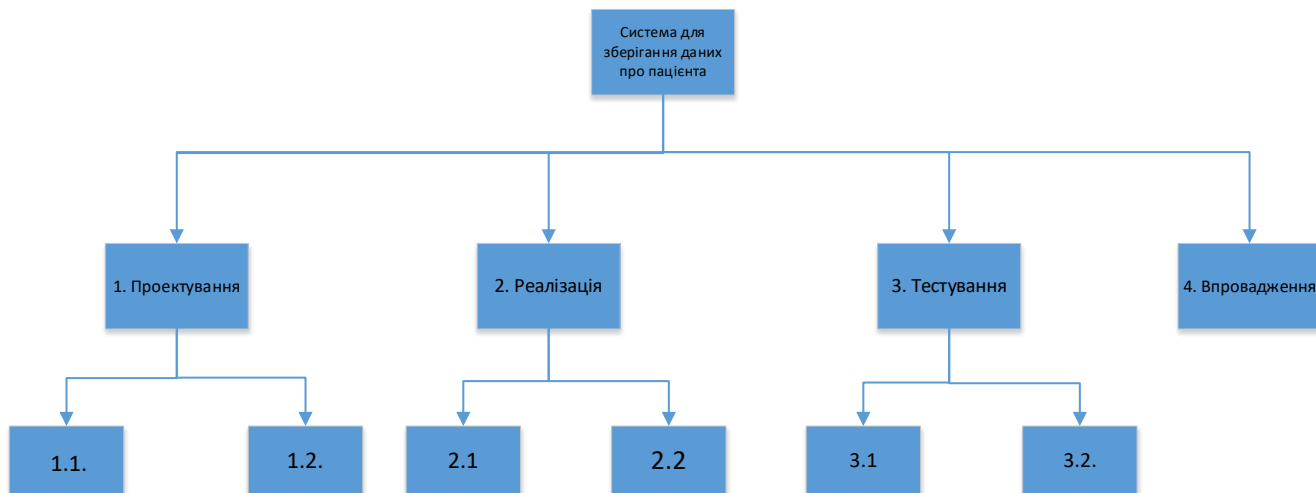


Рис. 1. Діаграма «Дерево цілей»

Де 1.1 - Проектування модуля для додавання вхідних даних у БД, 1.2 - Проектування користувацького інтерфейсу; 2.1 - Розробка модуля для додавання вхідних даних у БД, 2.2 - Розробка користувацького інтерфейсу; 3.1 - Тестування модуля для додавання вхідних даних у БД, 3.2 - Тестування користувацького інтерфейсу.

Функціонал розробленого додатку. Додаток, що було розроблено, має допомагати лікарям зберігати данні пацієнта.

Додаток має такий функціонал:

1. Вхід в систему (авторизація).
2. Створення нової картки пацієнта
3. Перегляд списку зі створеними картками пацієнта.

Вхід в систему (авторизація). Для того, аби захистити данні пацієнтів, потрібно зробити процес авторизації (рис. 2).

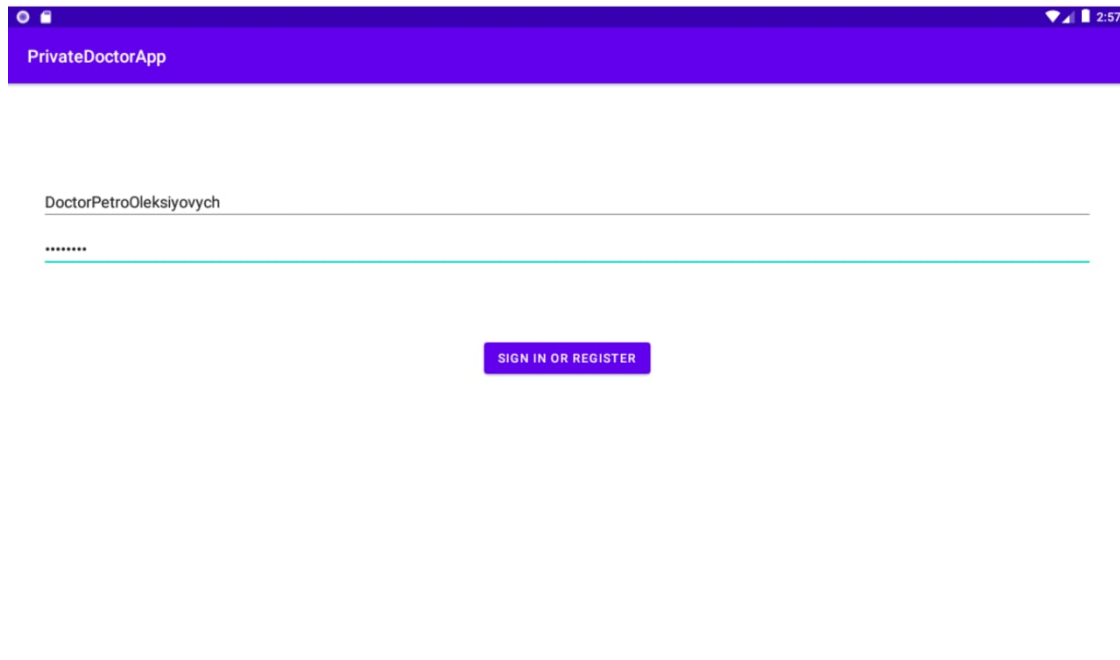


Рис. 2. Авторизація

При авторизації, користувач вводить свій логін і пароль. За допомогою цього система ідентифікує користувача, та надає йому доступ лише до тих даних, які вони може змінювати. Наприклад, за логіном та паролем доктора, йому надається доступ лише до тих карток пацієнта, які він створював, тобто до його клієнтів.

Створення нової картки пацієнта. Після входу в систему можна побачити порожній список викликів, адже жодного запису не було створено. Для того, аби створити картку пацієнта, потрібно натиснути на кнопку «Додати картку» (рис. 3)



Список викликів порожній

ДОДАТИ КАРТКУ ПАЦІЄНТА

Рис. 3. Інтерфейс програми, коли не створено жодної створеної картки пацієнта

Для того, щоб додати картку пацієнта, потрібно натиснути на кнопку «Додати картку пацієнта». Після цього відкривається форма, яка містить всі необхідні поля для заповнювання (рис. 4).

На формі представлені всі необхідні поля: прізвище пацієнта, його ім'я та по-батькові, вік, його адреса, номер картки, страховий поліс, страхова компанія (якщо такі є) та місце роботи. Можна заповнити не всі поля, наприклад, якщо невідомий вік пацієнта, його місце роботи, тощо. Обов'язковими є поля: прізвище пацієнта, його ім'я, по-батькові, вік та номер медичної картки.

Медична карта амбулаторного хворого № _____

Адреса пацієнта _____ Район _____

Привід для виклику _____

Прізвище _____ Ім'я _____

По-батькові _____ Стать _____

Вік, дні _____ міс _____ роки _____ Мешканець _____

Страховий поліс _____ Страхова компанія _____

Страхова компанія, коментар _____

Місце роботи _____

ЗБЕРЕГТИ КАРТКУ ПАЦІЄНТА ВІДМІНИТИ

Рис. 4. Порожня картка пацієнта

Далі користувач, в даному випадку лікар, має дві опції: заповнити картку, як це показано (рис.5) та зберегти її, або відмінити додавання запису, наприклад, якщо кнопку було натиснуто випадково або немає сенсу у додаванні запису у базу даних.

Медична карта амбулаторного хворого № 233445

Адреса пацієнта вул. Шевченко, д/м4, кв.34 Район Оболонь

Привід для виклику Температура

Прізвище Павлов Ім'я Василь

По-батькові Олександрович Стать М

Вік, дні _____ міс _____ роки 21 Мешканець Міста

Страховий поліс Відсутній Страхова компанія Відсутній

Страхова компанія, коментар _____

Місце роботи вчитель, школа 24

ЗБЕРЕГТИ КАРТКУ ПАЦІЄНТА ВІДМІНИТИ

Рис. 5. Заповнена картка пацієнта

Висновок. Було розроблено додаток, який буде використовуватися приватними лікарями при наданні медичної допомоги пацієнту. А саме: зберігання інформації про пацієнта на планшеті або іншому електронному носії. Для розробки було обрано мову програмування Java, систему керування базами даних SQLite, та платформа Android. Для полегшення розробки застосунки було розроблено модель життєвого циклу додатку. В результаті було розроблено мобільний застосунок, який дозволяє створювати картки пацієнта з необхідною інформацією, зберігати їх та використовувати у разі проблеми. Також була розроблена функція авторизації, що забезпечує захист інформації.

Література

1. *Methods of Information in Medicine* // Видавництво Schattauer. 2021. №21. С. 14.
2. Кей С. Хорстманн, Гари Корнелл *Java*. Библиотека профессионала, том 1. Основы. 9-е издание. «Вільямс». 2013. С. 56-58.
3. Джошуа Блох. *Java. Эффективное программирование = Effective Java*. 3-е. М.: Диалектика, 2019. 464 с. ISBN 978-5-6041394-4-8.
4. Siahaan V. *SQLite with JDBC for Beginners: Learn Fundamentals of Queries and Implement NetBeans-Based Projects Easily* / Vivian Siahaan., 2019. 307 с.
5. Rungta K. *Вивчіть SQLite за 1 день: повний посібник із вивчення SQLite для початківців* / Krishna Rungta., 2017. 102 с.
6. Хортон Д. *Програмування на Android для початківців: створіть поглиблені повнофункціональні програми Android 9 Pie, починаючи з нульового досвіду програмування*, 2-е видання / Джон Хортон., 2018. 766 с.
7. Роберт М. *Чиста архітектура: мистецтво розробки програмного забезпечення* / Мартін Роберт., 2019. 416 с. (#PROSystem).