УДК 004.51

Швець Г.К.[1], Тягунова М.Ю.[2]
[1] студ. гр. КНТз-512м НУ «Запорізька політехніка»
[2] канд. техн. наук, доц. НУ «Запорізька політехніка»

РОЗРОБКА ІНТЕРФЕЙСУ ТЕРМІНАЛУ САМООБСЛУГОВУВАННЯ ДЛЯ АПТЕКИ

Аптеки знаходяться в постійному розвитку і потребують інноваційних рішень для поліпшення сервісу та ефективності роботи. Використання терміналу самообслуговування може зменшити черги та покращити якість обслуговування в аптеці. Розробка інтерфейсу терміналу самообслуговування для аптеки є актуальною задачею в галузі фармації та ІТ.

Метою даної роботи є розробка ефективного та зручного інтерфейсу терміналу самообслуговування для аптеки з метою покращення сервісу та забезпечення більшої ефективності.

На основі проведеного дослідження було виявлено, що термінали самообслуговування для аптек є досить популярними в середньому та великому бізнесі, оскільки вони дозволяють зменшити черги та витрати на персонал. Вони зручні для покупців, оскільки дозволяють їм шукати та купувати ліки без допомоги фармацевта.

Однак термінали мають технічні та дизайнерські проблеми, такі як низька зручність використання та недостатня функціональність. Ці технічні проблеми можуть призвести до незадоволення користувачів та зменшення кількості апаратів, що використовуються в аптеках.

Перевага наявних інтерфейсів полягає в їхній простоті та доступності для користувача. Багато інтерфейсів мають прості, інтуїтивно зрозумілі значки і текст, що допомагають користувачам зрозуміти, які опції доступні. Деякі інтерфейси також мають функції персоналізованого налаштування для користувача, що важливо для підвищення задоволеності користувачів терміналом.

З недоліків виявлено, що деякі інтерфейси можуть бути занадто складними або важкими для розуміння користувачами, що може призвести до погіршення зручності використання терміналу та збільшення кількості помилок. Вони не пристосовані до різних типів користувачів, наприклад, до людей з різними здібностями або іншими потребами.

У роботі розроблено детальну структурно-функціональну схему для аптечного терміналу самообслуговування. Розглянуті основні блоки, з яких складається термінал, такі як дисплей, клавіатура, сканер штрих-коду та оплата. Для кожного блоку розроблені функції та їхні можливі варіанти. Розглянуто можливість інтеграції терміналу з аптечною базою даних для точної та швидкої роботи системи. Для кожної функції терміналу визначені їх призначення та взаємодію з іншими блоками. Для візуалізації роботи терміналу та взаємодії між блоками створено функціональну діаграму. Вивчено наявність технологій для швидкої та надійної роботи терміналу. Визначено архітектуру програмного забезпечення терміналу та технології, які будуть використані під час його розробки.

У роботі розглянуто основні принципи побудови архітектури терміналу та технології реалізації інтерфейсу. Обрано модульну архітектуру з використанням патерну Model-View-Controller (MVC), який дає змогу відокремити логіку бізнес-процесів від логіки інтерфейсу терміналу. Також вирішено використовувати для реалізації інтерфейсу терміналу сучасні технології веброзробки, такі як HTML, CSS і JavaScript. Щоб термінал працював швидше, планується використовувати AJAX-запити, які оновлюють тільки частини сторінки, а не перезавантажують всю сторінку. Крім того, для зберігання даних терміналу буде використовуватися база даних MySQL.

Прототип інтерфейсу терміналу розроблено на чистому JavaScript. Розробка заснована на модульній архітектурі з використанням патерну Model-View-Controller (MVC), визначеного в попередньому розділі.

У процесі розроблення інтерфейсу використовувалися сучасні технології веброзробки, як HTML5, CSS3 і чистий JavaScript ES6; для кожного компонента патерну MVC було розроблено три незалежні модулі - модель, подання та контролер. Модуль Model зберігає й обробляє дані.

Таким чином, використання терміналу самообслуговування дозволяє зменшити черги та підвищити якість обслуговування в аптеці . В основі розробки лежить модульна архітектура з використанням патерну Model-View-Controller (MVC). Для візуалізації роботи терміналу та взаємодії між блоками була створена функціональна діаграма. З метою покращення сервісу та забезпечення більшої ефективності для реалізації інтерфейсу терміналу були використані сучасні технології веброзробки, такі як HTML, CSS та JavaScript.