

замовчуванням: прийняти пакети або відхилити пакети.

Приклади пакетів для перевірки були отримані випадковим чином, використовуючи дані параметрів мережі.

З огляду на результати тестів, проведених з різними параметрами було встановлено, що НМ дійсно може бути корисним в справі поліпшення можливостей фільтрації. Результати показують, що НМ здатна класифікувати мережеві пакети майже так само як брандмауери з великим числом правил. Це може бути використано для розширення можливостей фільтрації брандмауера, особливо за рахунок поліпшення правил по замовчуванню, з неправильною фільтрації пакетів, які не входять в правила брандмауера.

Ці поліпшення не тільки в кінцевому рахунку не тільки розширяють можливості застосування фільтрації пакетів брандмауерів, але можуть також включати в себе механізми безпеки, які запобігають атакам.

Перелік посилань

1. M. Jordan and T. Mitchell, "Machine learning: Trends, perspectives, and prospects", Science, vol. 349, no. 6245, pp. 255-260, 2015.

2. K. Valentin and M. Maly, "Network firewall using artificial neural networks," Computing and Informatics, vol. 32, pp. 1312–1327, 2013.

Програмна реалізація методу автоматичного визначення оптимального режиму тренування із застосуванням мобільних пристроїв

Кисельов В.О., Пирч О.В.

Науковий керівник – к.т.н., доц. Чешун В.М.

Хмельницький національний університет

Для практичної реалізації методу автоматичного визначення оптимального режиму тренування із застосуванням мобільних пристроїв застосовано мову програмування Swift, яка є сучасною мовою програмування і розроблена Apple для створення своїх пристроїв на базі iOS та всіх наступних екосистем. Програми можуть бути розроблені також для роботи на macOS (для комп'ютерів Apple), watchOS (AppleWatch), tvOS (цифровий медіаплеєр Apple TV) і, що може бути дивно, z/OS, тобто живлення комп'ютерів IBM Mainframe.

Наразі мова поширюється на ліцензії Apache, що робить її доступною для всіх бажаючих. Це порівняно новий проект, запущений у червні 2014 року, через сім років після першого запуску iPhone.



Рисунок 1 – Розвиток Swift

Оскільки більшість компаній світу притримуються думки, що найбільш вдалим ходом є розробляти додатки на нативних мовах програмування (для Android це Java та Kotlin, а для Apple це Objective-C), цей додаток було створено під iOS на мові програмування Swift. 21 століття це вік комп'ютеризації, програмування та інтернетизації автоматизованого управління того процесу, що люди робили власноруч. Даний метод полягає у роботизації процесу та поєднання роботи людей, яким наразі займаються тренера у спортзалах та дієтологи. Всі послуги якими користуються ті хто хоче тримати себе у тонусі та спортсмени, які всерйоз займаються і планують отримати результат. Розробка такого додатку є актуальним питанням на сьогоднішній день, оскільки майже кожна людина займається спортом та користується телефоном.

Перш за все було створено [1] StoryBoards для проекту, це так звана візуалізація всіх екранів iOS та розробка зв'язків між ними.

Це є важливим аспектом так як стає одразу видно як буде виглядати додаток візуально і також це дозволить розділити весь код на 3 частини: патерн проєктування [2]MVC: Model View Controller, який складається з трьох компонентів: View (відображення, користувальницький інтерфейс), Model (модель, ваша бізнес логіка) і Controller (контролер, містить логіку на зміну моделі при певних діях користувача, реалізує Use Case). Основна ідея цього патерну в тому, що і контролер і відображення залежать від моделі, але модель ніяк не залежить від цих двох компонент. Це як раз і дозволяє розробляти і тестувати модель, нічого не знаючи про відображення та контролери.

Даний метод в XCode спрощує управління кодом і за допомогою таких інструментів, як Interface Builder ми можемо побудувати всі екрани нашого додатку а також провести зв'язки між ними та присвоїти кожному з них по Class якому вони відповідають.

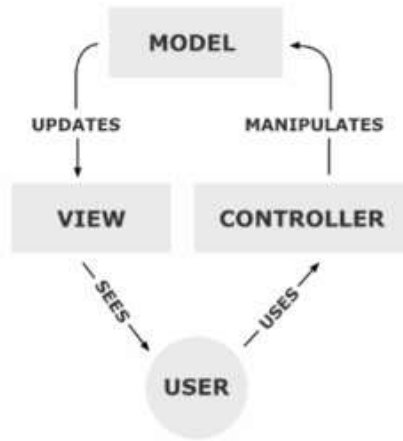


Рисунок 2 – Патерн MVC

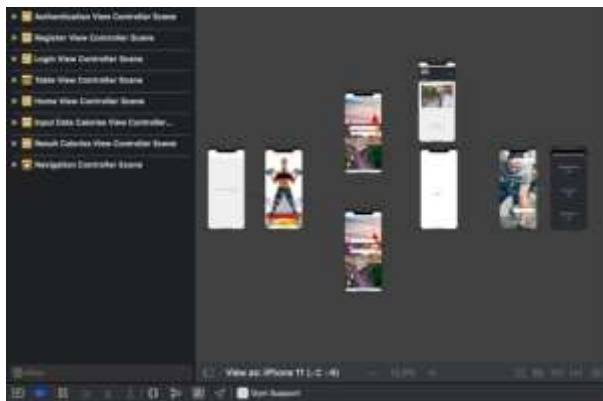


Рисунок 3 – Interface Builder iOS додатку

Найскладніша частина це розробка[3] Controllera, логіки проекту, а саме в додатку необхідно реалізувати такі функції:

- 1) Підключення до бази даних
- 2) Реєстрація користувачів у системі та захист користувача від некоректної реєстрації
- 3) Авторизація користувачів та вихід
- 4) Внесення даних користувача про його вік, вагу, здоров'я та спосіб життя
- 5) Розрахунок кількості калорій по формулі для підтримання ваги,

схуднення або набору маси

6) Розробка кнопки "Back" на всіх Storyboards

Для підключення до баз даних використовується Firebase, мобільний cloud storage, який надає гарну можливість для побудування потужних мобільних додатків для телефонів. Він має три основних сервіса, таких як побудова баз даних в реальному часі, аутентифікація користувача та хостинг. Основне направлення це cloud storage NoSQL БД для RealTime додатків. API Firebase дозволяє розробникам зберігати дані та синхронізувати їх між декількома клієнтами. В додатку "GymSupport" Користувач зберігає дані для аутентифікації та реєстрації в Firebase.



Рисунок 4 – Інтерфейсна частина програми

Після успішної реєстрації відкривається доступ в якому користувач буде мати можливість зберігати свої дані та отримати розрахунки індивідуальні для його ваги, росту та способу життя який він веде(Сидячий образ життя, "не активний", "середня активність", "Дуже активний", "Надзвичайно активний").

Харріс і Бенедикт доказали[4], що число калорій, необхідних людині щодня, змінюється від таких показників: базального метаболізму (BMR) і активного метаболізму (AMR). Щоб розрахувати калорії для конкретної людини за її параметрами будемо використовувати цю формулу яка звучить так:

для чоловіків: $BMR = 88.36 + (13.4 \times \text{вага, кг}) + (4.8 \times \text{ріст, см}) - (5.7 \times \text{вік,})$

для жінок: $BMR = 447.6 + (9.2 \times \text{вага, кг}) + (3.1 \times \text{ріст, см}) - (4.3 \times \text{вік})$



Рисунок 5 – Головне меню та калькулятор розрахунків калорій для людини

Таблиця 1. Розрахування AMR в залежності від способу життя

Сидячий спосіб життя	1,2
Помірна активність (легкі фізичні навантаження або заняття 1-3 разів на тиждень)	1.375
Середня активність (заняття 3-5 разів на тиждень)	1.55
Активні люди (інтенсивні навантаження, заняття 6-7 разів на тиждень)	1.725
Спортсмени та люди, що виконують схожі навантаження	1.9

Для визначення повної потреби організму в енергії і калоріях потрібно помножити BMR, що відповідає вашій статі, віку і вазі, на коефіцієнт денної активності. У свою чергу, цей коефіцієнт визначається в залежності від рівня фізичної активності в рамках робочого тижня (він варюється від 1.2 до 1.9 в залежності від того яку активність буде вести користувач) Тим самим ми отримуємо потрібні нам дані, кількість калорій, яку нам потрібно втрачати за день, щоб тримати ту вагу яку маємо або схуднути чи набрати вагу (Для спортсменів які набирають на масу). Також додаток дозволяє нам обрати одну з програм тренування з розрахованими даними про спалені калорії. На Скріншоті вказана людина яка вважить 80 кг у віці 24 роки при зрості 83 кг, чоловічого статі, яка веде сидячий спосіб життя. Щоб схуднути їй потрібно в день вживати менше ніж 2300 калорій.

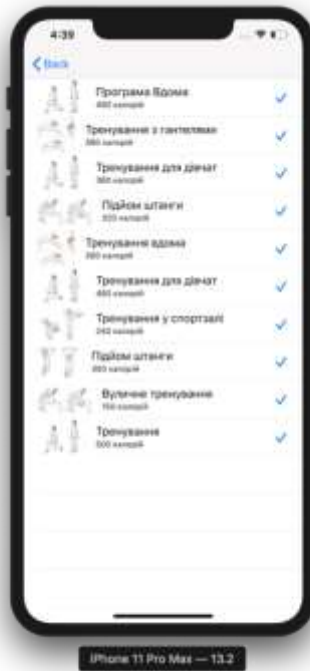


Рисунок 5 – Екран з тренуваннями

Висновок. Додаток зміг поєднати і автоматизувати професії дієтолога та тренера у собі, тому враховуючи всі переваги і можливості методу визначення оптимального режиму тренування із застосуванням мобільного додатку, сучасну технологію мови програмування Swift та БД Firebase, було розроблено програму — GymSupport. Найбільш узгодженою IDE для розробки виявився рідний для Apple Xcode 11 версії та мова програмування Swift версії 5

Перелік посилань

- 1.Swift. Разработка приложений под iOS и MacOS, Василий Усов 2014
2. Хрестоматія iOS паттернов. На всякий... Дима Малеев. 2016
- 3.Swift 3. Разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK. Молли Маскри, Ким Топли, Дэвид Марк. 2017
4. Традиционная диетология. Дворянчиков А.Ю.2019