# ВИМОГИ ДО МОБІЛЬНОГО ДОДАТКУ «РОЗКЛАД»

    Сьогодення диктує певні вимоги для всіх додатків. Останні віяння в ІТ, показують, що все більше додатків переносять у WEB та на мобільні платформи, а інформацію все частіше зберігають у хмарних сховищах. Безперечно, у такого підходу є дуже багато плюсів: можливість працювати, майже з будь-якого пристрою, доступ до сервісу з будь-якої частини світу, такі додатки не потребують наявності комп’ютера та вимагають лише підключення до мережі Інтернет.

Але саме підключення до всесвітньої мережі деколи неможливе, хоча останнім часом інтернет стає все доступнішим і з’являється на кожному кроці, проте існують і такі місця, де підключення неможливе з певних причин. Ярким прикладом такого місця є літаки чи метро, а в сучасному ритмі життя, люди працюють майже весь час, і попрацювати в літаку або метро це нормальна справа. Але як бути, якщо потрібний додаток функціонує лише з підключення до інтернет-мережі? Тому багато сучасних програм мають можливість встановити спеціальний клієнт, який буде дозволяти працювати в режимі офлайн, а при підключені синхронізувати напрацювання. Також однією з проблема є зручність роботи. Зараз у світі смартфонів більше ніж людей, і багато веб-ресурсів переносяться у нішу мобільних додатків. Такий підхід зумовлений надзвичайною зручністю та гнучкістю.

Тому, було вирішено, що підсистема «Розклад» повинна бути реалізована для мобільних платформ та мати схожу технологію, що дозволить її використовувати без підключення до інтернет. Таким чином, якщо користувач має доступ до інтернету, то він може працювати або в онлайн-режимі, або в офлайн. В онлайн-режимі вся робота буде зберігатись у реальному часі на серверах, в офлайн-режимі всі напрацювання синхронізувати при першому підключенні.

## Вимоги до функціоналу

Система для роботи з поточним навчальним розкладом повинна забезпечувати систему авторизації, що забезпечить певний рівень безпеки при використанні додатка. Уся персональна інформація та профілі користувачів уже містяться у системі «Електронного кампусу», тому з нашого боку достатньо обмежити доступ до цієї інформації шляхом введення авторизації.

Зважаючи на те, що підсистема створюється як для студентів, так і для викладачів, то внутрішня логіка додатку підтримує роботу з різними правами доступу до даних. Такий підхід дозволяє надати різний функціонал для користувачів, що в свою чергу забезпечує ще один рівень захисту даних.

Дана підсистема розрахована на надання можливості опрацювання даних в широкому діапазоні дій. Таким чином, функціонал додатку включає в себе не тільки стандартні можливості перегляду та видалення елементів розкладу, а й більш складні, такі як: редагування та додавання нових даних на рівні свого профілю, редагування та додавання на рівні груп, оповіщення інших користувачів про зміни та інше.

Використовуючи додаток, користувачі матимуть змогу також вносити певні зміни у свій профіль, згідно з реалізованим функціоналом додатку. Ця функція не є цільовою та виконує суто допоміжну роль.

Більш детальні вимоги по функціоналу наведено в табл. 1.

*Таблиця 1. Вимоги до функціоналу мобільного додатку*

|  |  |  |
| --- | --- | --- |
| Функція | Вимоги | |
| Авторизація | | Потрібно надати користувачу можливість авторизуватися, використовуючи вже існуючий профіль в системі «Електроний кампус». Авторизація – це процес валідації логіна і пароля. Якщо така пара є в БД системи, то користувачу надається доступ до внутрішніх даних. |
| Редагування профілю | | Потрібно забезпечити функціонал, для редагування персональної інформації користувача. Змінити можна лише певні дані профілю, відповідно наданій користувачу інформації. |
| Перегляд поточного навчального розкладу | | Потрібно забезпечити можливість перегляду поточного навчального розкладу для всіх користувачів «Електронного кампусу». Реалізована функція розширеного перегляду з виведенням номеру корпуса, аудиторії, П.І.Б. викладача, годинами початку та закінчення, скороченою та повною назвою предмета. |
| Видалення елементів навчального розкладу | | Потрібно реалізувати можливість видалення предметів розкладу. Видалення може відбуватись на двох рівнях: студента та викладача.  Рівень студента дозволяє видаляти елементи розкладу лише для себе, ці зміни не відносяться більше ні для кого в системі.  Рівень викладача видаляє елементи розкладу на рівні груп, тобто змінює розклад для всіх користувачів-студентів, що мають відношення до цього елементу. |
| Редагування існуючих елементів навчального розкладу | | Потрібно надати змогу привносити зміни до уже існуючих елементів навчального плану. Редагування також може виконуватись на двох рівнях.  Рівень студента дозволяє редагувати елементи розкладу виключно для себе.  Рівень викладача вносить зміни на рівні груп, тобто змінює розклад для всіх користувачів-студентів, що мають відношення до цього елементу. |
| Створення нових елементів навчального розкладу | | Потрібно забезпечити функцію створення нових елементів розкладу, таких як консультації, додаткові практики, тощо. Створення нових елементів також може виконуватись на двох рівнях.  Рівень студента дозволяє створювати події лише для себе, залишаючи їх невидимими для всіх інших користувачів системи.  Рівень викладача створює нові елементи на рівні груп, тобто вносить зміни для всіх користувачів-студентів відповідної групи. |
| Забезпечення коректності вводу всіх даних | | Потрібно перевіряти коректність всіх введених даних, та, при помилках під час введення даних, повідомляти про це користувача. |
| Функція сповіщення про зміни | | Потрібно реалізувати функцію сповіщення про зміни усіх користувачів, які «зв’язані» з елементом що підлягає редагуванню. |
| Кешування інформації | | Необхідно забезпечити можливість роботи в офлайн режимі шляхом кешування змін з синхронізацією в майбутньому. |

## Вимоги до інтерфейсу

З розвитком технологій вимоги до інтерфейсу невпинно зростають. Останні віяння мобільних технологій схиляють розробку додатків до впровадження здатності пристосовувати під різні розміри дисплеїв пристроїв. А також бути естетичним, лаконічним, гарним. Отже, додаток для роботи з поточним навчальним розкладом не є виключенням.

Сучасні вимоги до мобільних додатків диктують досить жорсткі правила до їх інтерфейсної частини. Потрібно розуміти що додаток створюється для користувача, і задоволення його потреб чи не найперший крок для будь-якого мобільного сервісу незалежно від платформи. Клієнт має справу тільки з інтерфейсом і не бере безпосередньої участі в роботі з внутрішніми модулями. Як результат, головний критерій для користувача – це дизайн.

Так як додаток розробляється для AndroidOS платформ, то даний інтерфейс побудований за «Принципами дизайну для Android», що були запропоновані представниками Google.

При розробці додатку була поставлена ціль «мінімум дій - максимум потрібної інформації». Для реалізації цієї мети був використаний набір інструментів наданий Google для розробки інтерфейсів на Android-платформу.

Як говорилось вище, додаток розрахований на підтримку різних пристроїв, як планшетів, так і смартфонів. Такий підхід накладає певні вимоги на проектування функціонального інтерфейсу. В результаті дизайн повинен бути «гумовим», що забезпечить коректне відображення графічних елементів на будь-якому з дисплеїв.

Підсистема «Розклад» створювалась з метою надання можливості перегляду поточного учбового плану, що, в свою чергу, накладає певні обмеження. Спосіб подання інформації залежить від значимості відповідних даних та виділяється акцентами. Так, назва предмету повинна займати головне положення в елементі поточного розкладу та виділятись більшим розміром шрифту. Дані про викладача та аудиторії розміщуються безпосередньо нижче назви, виводяться меншим шрифтом та курсивом.

Також велику роль відіграють кольори. Набір кольорів повинен відповідати вимогам «Принципів дизайну для Android», таким чином в додатку переважають нейтрально синій та білий кольори, з чорним текстом. Для акцентування уваги використовуються сірий, зелений та червоний. Так, для відображення неактивної інформації блоки з нею фарбуються в сірий.

## Вимоги до архітектури

Для реалізації правильної і швидкої роботи в додатку «Розклад» було реалізовано патерн проектування MVC, а точніше його нащадка MVP. Реалізація цього шаблону зумовлює розбиття програми на блоки «model», «view» та «controller». Така структура накладає певні вимоги на елементи архітектури. Таким чином, «controller» виконує логіку та внутрішній функціонал додатку, використовуючи при цьому елемент даних «model», та відображає результат у «view». «Model» повинен виступати набором елементів додатку які служать джерелом даних. Користувач отримує інформацію на пряму з «view» та працює з додатком використовуючи «controller». Такий підхід використовується в дуже широкому спектрі розробки додатків, не тільки мобільних, а й десктопних. Використовуючи подібний підхід додаток набуває додаткової гнучкості, можливості масштабування та удосконалення.

Також, функціонал додатку «Розклад» накладає деякі вимоги на архітектуру, а саме створення внутрішнього кешу. В нашому випадку кешем буде слугувати локальна база даних. Такий підхід дозволить як пришвидшити роботу всього додатку, так і забезпечити офлайн режим. Внутрішня БД дублює відповідні таблиці серверної БД та зберігає там дані для відображення та роботи з додатком.

## ВИСНОВОК ПО РОЗДІЛУ

В даному розділі були розглянуті основні вимоги для мобільних додатків, в тому числі і підсистеми «Розклад». Ці вимоги є обов’язковими, тому що наявність таких властивостей у системи, забезпечує її зручність та практичність.

Також, була вирішено задовольнити вимогу можливості працювати з системою при відсутності інтернету, що є надзвичайно важливо, адже інколи отримати доступ до мережі просто неможливо. Особливо, зважаючи на те, що використання цієї підсистеми-додатка планується в Україні, де покриття мережею інтернет, залишає бажати кращого. Ця вимога є комплексною для підсистеми і була вирішена шляхом реалізації такого підходу – за присутності зв’язку з сервером, усі данні завантажуються до додатку і одразу дублюються у нашу локальну БД. Таким чином створюється кеш, який оновлюється і дає можливість додатку користуватися інформацією навіть в оффлайн режимі.

В розділі був розглянутий основний функціонал додатку з детальним описом кожної з можливостей підсистеми. Більшість функцій розділені правами доступу, та можуть виконувати дії у двох режимах. В результаті отримуємо широкий спектр можливостей підсистеми з мінімалізацією дій, що виконує одну з основних вимог дизайну інтерфейсу. Детальніше функціонал описаний в табл. 1, з якої можна зробити висновок про задачі всієї підсистеми в цілому.

Розділ містить в собі опис реалізації вимог до архітектури додатку відповідно з патерном проектування MVC(MVP). Подібний підхід спрощує не тільки розробку, а й пришвидшує та оптимізує роботу внутрішньої логіки додатку «Розклад» в цілому. Як результат, підсистема-додаток буде гнучкою та масштабуємою.

Щодо дизайну, виконуються вимоги надані компанією Google, а саме «Принципи дизайну для Android». Реалізовано акцентування інформації шляхом розмірів шрифту, стилів та виділенням елементів певними кольорами. Також реалізований «гумовий» дизайн для забезпечення підтримки різних дисплеїв.