

Інструментарій для спрощення розробки мобільних додатків

Виконав:

Беляєв Євгеній

Науковий керівник

Булах Б.В.



Постановка завдання

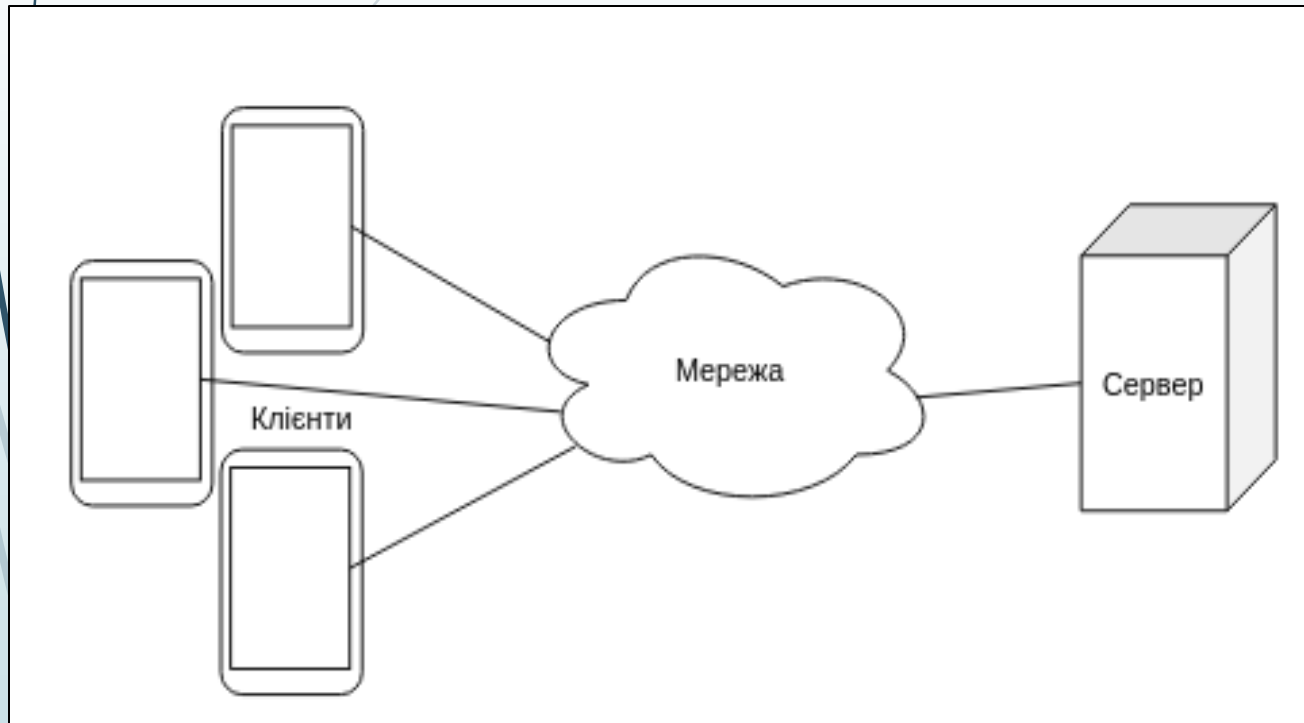
- Дослідити особливості роботи мобільних додатків і особливості сфери їх розробки.
- Проаналізувати модель клієнт-сервер і способи її застосування при розробці мобільних застосунків.
- Запропонувати підхід для спрощення розробки мобільних додатків.
- Створити інструментарій для спрощення розробки мобільних додатків, які використовують модель клієнт-сервер.



АКТУАЛЬНІСТЬ

- Розробка мобільних додатків - широка, в той же час молода сфера, яка охоплює велику кількість апаратних і програмних платформ, технологій, мов програмування, тощо. Через це, ця сфера до сих пір дуже швидко розвивається, часто зазнає значних змін і нововведень, які радикально впливають власне на розробку додатків для мобільних платформ.
- Мобільні пристрої обмежені апаратними можливостями, саме тому широкого використання при розробці мобільних додатків здобула модель клієнт-сервер.
- Існують рішення, які використовують різні підходи для розробки додатків з використанням моделі клієнт-сервер. Вони мають як переваги, так і недоліки.

Клієнт-серверна архітектура



- Архітектура клієнт-сервер - одне із архітектурних шаблонів програмного забезпечення.
- сервер - програма, або набір програм, що працюють на віддаленому комп'ютері і надають інформацію або інші послуги програмам, що звертаються до НИХ
- клієнт - програма на мобільному пристрої, яка вміє формувати зрозумілий серверу запит і читати отриману відповідь. Клієнти звертаються до серверів для використання їх послуг.
- Мережа забезпечує зв'язок між клієнтами і серверами.
- інтерфейс взаємодії - це формат і спосіб передачі, отримання запитів, відповідей обома сторонами.



Особливості використання моделі клієнт-сервер в розробці мобільних додатків

- Якщо не використовувати додаткове ПЗ, а створювати все з нуля то розробка ведеться і на стороні клієнта, і на стороні серверу.
- Основний функціонал найчастіше розташовується на стороні клієнта.
- Частина даних, що не потрібна на стороні серверу може зберігатися на пристрої. Дані, що потрібні не лише на локальному пристрої переважно зберігаються у віддаленому сховищі.
- Можливе делегування ресурсоємних задач на сторону серверу.

Модель MBaaS



Firebase

- MBaaS - це модель для надання розробникам можливості просто і швидко створити і пов'язати мобільні додатки з хмарним сховищем даних за допомогою спеціального API.
- Основною задачею моделі є забезпечення простоти збереження інформації в зовнішньому сховищі без розробки окремого бекенду.
- Постачальники послуг іноді надають додатковий функціонал, такий як управління користувачами, відправлення сповіщень, авторизація за допомогою соціальних мереж.
- Ці послуги надаються за допомогою користувацьких наборів для розробки програмного забезпечення (SDK) та інтерфейсів програмування прикладних програм (API).

 AWS Amplify

 **realm**

 Parse



Причини пошуку іншого підходу

- Створення бекенду з нуля - типова та трудомістка задача, яка потребує додаткових ресурсів та навичок для розробника мобільних додатків.
- Створення мобільних додатків з використанням моделі MBaaS дещо обмежене у використанні. Основною проблемою MBaaS є і його перевагою. MBaaS максимально відсторонює розробника від втручання власне в бекенд.
- Іноді постачальники MBaaS включають деякі типові рішення в склад моделі, але створення власних рішень на бекенді переускладнене або така можливість просто відсутня.
- Деякі студії розробники створюють власні інструменти в тому числі і бекенд, які використовують в своїх проектах, але вони вимагають певних усталених звичок їх розробників для їх використання, тому залишаються виключно корпоративними розробками.



Інструментарій для спрощення розробки мобільних додатків

- Інструментарій орієнтований на спрощення розробки мобільних додатків, що використовують модель клієнт-сервер.
- Інструментарій включає бекенд і програмну бібліотеку, для взаємодії з бекендом.
- Перевагою інструментарію є простота використання і налаштування бекенду, можливість створення і користування запропонованими модулями, розширення функціоналу і сфери використання.



Модулі

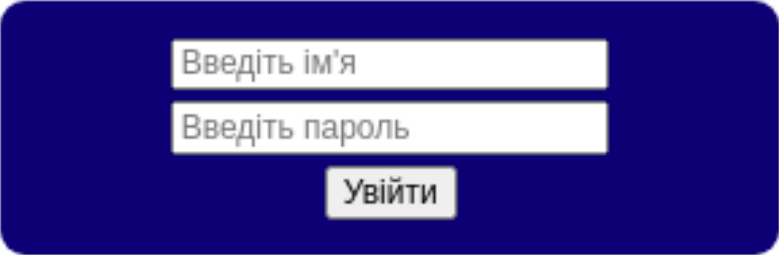
- Для спрощення розробки, був створений концепт модулів. Модулі пов'язані на беку і в бібліотеці на клієнтській стороні задля реалізації функціоналу, що вимагає одночасної участі і бекенду, і клієнта.
- Модулі - це функціональна одиниця пов'язана даними і функціями, що реалізує певну виділену поведінку, пов'язану як з клієнтом, так і з сервером. Частина модуля реалізується на бекенді (бекенд модуль) і в складі бібліотек для клієнтів.
- В даній реалізації, модулі на стороні серверу покликані в першу чергу для додавання налаштування таблиць для роботи, а також для визначення зв'язків з іншими модулями. Це все робиться задля того, щоб спростити підготовку бекенду для простого доступу клієнту.
- Основними задачами модулів на стороні клієнта є визначення функцій і можливостей модуля. Тут визначаються основні методи модуля.



Особливості Бекенду

- Використання реляційної бази даних на відміну від найпопулярніших постачальників MBaaS (вони використовують переважно документальні бази даних)
- Типовий стек для веб - PHP + MySQL, що забезпечує сумісність з більшістю хостингів на даний час
- Швидке розгортання бекенду, схоже на "п'ятихвилинне встановлення WordPress"
- Присутність авторизації
- Самодостатній юзер інтерфейс, що включає інтерфейс для взаємодії з модулями і інтерфейс для роботи з таблицями бази даних

Авторизація в інтерфейсі користувача на бекенді



Введіть ім'я

Введіть пароль

Увійти

Сторінка модулі на бекенді

Таблиці

Модулі

Вийти

User Module

A module that implements user data basics. By default, used for modules that need a relationships with users.

Модуль вже інтегровано:

```
TABLE users (  
  user_id INT NOT NULL AUTO INCREMENT ,  
  username VARCHAR(100) NOT NULL UNIQUE,  
  user_password VARCHAR(32) NOT NULL,  
  PRIMARY KEY (user_id))
```

Author's Article Module

A module that implements data basics for author's articles.

```
TABLE authorsArticles (  
  authors_id INT NOT NULL,  
  heading TEXT,  
  content TEXT,
```

```
FOREIGN KEY (authors_id) REFERENCES users(user_id))
```

Інтегрувати

Сторінка таблиці на бекенді

Таблиці Модулі Вийти

users ▾

| user_id | username | user_password | |
|---------|-----------|----------------------------------|---|
| 1 | somebody | 51243bad817be5b297fb0d7e21db5a3d | X |
| 5 | yesterday | 85cf01fad2e358d6c2ce6af76499254b | X |
| 7 | sumbarine | 1af862bda3a14bd36a35d768d02e59a9 | X |
| 9 | incognito | 827ccb0eea8a706c4c34a16891f84e7b | X |
| 10 | iamuser | 7967c80ea656e97b191135e12315e3cc | X |



Програмна бібліотека на стороні клієнту

- Програмна бібліотека легко включається в склад проекту. Потрібно лише її імпортувати і додати як залежність у проект. А також визначити дані для доступу при створенні головної активності.

```
QueryMgr queryMgr = QueryMgr.getInstance();  
queryMgr.setAccessData("example.com/mi/", "admin", "qwerty");
```

- У складі бібліотеки на стороні клієнта присутні:
 - QueryMgr - інтерфейс, що включається до складу для звернень до серверу
 - Module - кореневий абстрактний клас реалізації модуля
 - Модулі, що мають свої методи і наслідуються від класу Module



Використання модулів на стороні клієнта

- Модулі використовують в своїй реалізації екземпляр класу *QueryMgr* для надсилання запитів і отримання відповідей від бекенду.

- Приклад використання:

```
UserModule userModule = new UserModule();  
userModule.authentication("somebody", "password");
```

```
if(userModule.isAuthenticated()) {  
    startActivity(someActivity);  
}
```

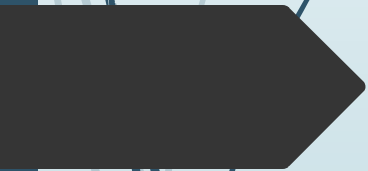


Висновки



- У дипломній роботі було розглянуто особливості роботи мобільних додатків і особливості сфери їх розробки.
- Тут було проаналізовано принципи роботи і причини переважного використання деяких інструментів і підходів.
- Було досліджено модель клієнт-сервер, причини її широкого застосування у цій сфері і способи застосування
- Було запропоновано альтернативний підхід і створено інструментарій для спрощення розробки мобільних додатків, що використовують клієнт-серверну модель.

Дякую за увагу



Створення модулю на бекенді

```
<?php
include_once "../modules/module.php";
class UserModule extends Module {
    protected $moduleName = "User Module";
    protected $moduleDescription = "A module that implements user data basics. By default, used for modules that
    need a relationships with users.";
    protected $tables = array(
        "users"=>["user_id INT NOT NULL AUTO_INCREMENT ",
        "username VARCHAR(100) NOT NULL UNIQUE", "user_password VARCHAR(32) NOT NULL"]
    );
    protected $structuralProperties = array(
        "users"=>["PRIMARY KEY (user_id)"]
    );
}
$modulesList["User Module"] = new UserModule;
?>
```