

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра ЕОМ

## КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Мобільний застосунок для ведення бюджету»

виконав:  
ст. гр. КІУКІ-21-4  
Бояков І.С.

керівник:  
доц. каф. ЕОМ  
Колтун Ю.М.

Харків 2025

## Мета роботи

2

**Метою роботи** є розробка інтуїтивного та функціонального мобільного застосунку для слідування за особистим бюджетом з функцією "скарбничка", що дозволить користувачам слідувати за доходами та витратами та ефективно досягати фінансових цілей.



### Контроль над власним бюджетом

Слідування за власними витратами є надійним рішенням для запобігання здійснення необдуманих витрат та забезпечення більш стабільного рівня життя.



### Накопичення коштів

Це допоможе уникнути фінансово невіддних та необдуманих витрат сучасному споживачу.



## Наявні рішення

3



## Поставлені задачі

4

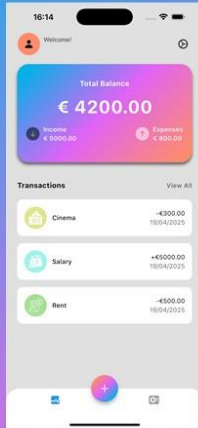
В ході проведення аналізу предметної області була поставлена задача: розробити кросплатформний мобільний застосунок, який матиме певний набір можливостей. А саме:

- зрозумілий та логічний інтерфейс користувача із інтуїтивною структурою для покращення досвіду користувача;
- облік доходів та витрат (транзакцій) з урахуванням необхідної інформації: сума, тип, дата та інше;
- формування звітності з візуальним оформленням у вигляді кільцевих діаграм, гістограм, для легкої оцінки фінансового стану;
- проста у використанні та функціональна "Скарбничка";
- контроль над даними: можливість їх видалення, редагування чи повного очищення.

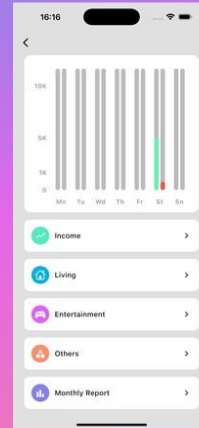


## Інтерфейс застосунку

5



Головний екран

Головний екран  
в "темному" оформленніЕкран статистики  
та звітів

Екран звіту



## Оновлення даних | Плагін Provider

6

Даний плагін дозволяє автоматично відображати зміни даних у відповідних елементах інтерфейсу без необхідності вручну оновлювати екран

provider 6.1.4

Published 23 days ago • @dash-overflow.net ( Dart 3 compatible )

SDK | FLUTTER | PLATFORM | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

10.6K

Readme Changelog Example Installing Versions Scores

A wrapper around `InheritedWidget` to make them easier to use and more reusable.

By using `provider` instead of manually writing `InheritedWidget`, you get:

- simplified allocation/disposal of resources
- lazy-loading
- a vastly reduced boilerplate over making a new class every time
- devtool friendly – using `Provider`, the state of your application will be visible in the Flutter devtool
- a common way to consume these `InheritedWidgets` (See `Provider.of<ConsumerSelector>`)
- increased scalability for classes with a listening mechanism that grows exponentially in complexity (such as `ChangeNotifier`, which is O(N) for dispatching notifications).

```
63 void updateIncome(double newIncome) {
64   weekIncome = newIncome;
65   notifyListeners();
66 }
67
68 void updateExpenses(double newExpenses) {
69   weekExpenses = newExpenses;
70   notifyListeners();
71 }
```

```
73 Future<void> loadGoals() async {
74   goals = await AppDatabase.instance.getGoals();
75   notifyListeners();
76 }
77
78 Future<void> loadProfileName() async {
79   profileName = await AppDatabase.instance.getProfileName();
80   notifyListeners();
81 }
```



## Збереження даних

7

В застосунку використовуються два механізми, які зберігають дані локально в пам'яті пристрою, що дозволяє застосунку працювати без підключення до Інтернету.



Механізм для збереження простих даних у вигляді пар "ключ-значення"



Легка вбудована реляційна база даних, яка не потребує окремого серверу для роботи

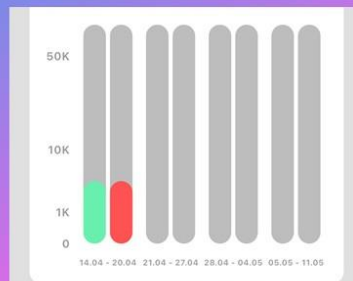


## Звіти та діаграми

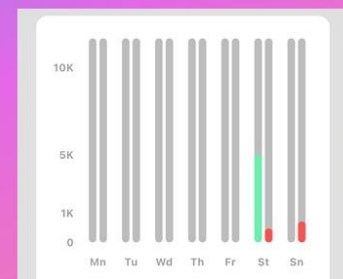
8



Щомісячний звіт з кільцевою діаграмою для легшої оцінки



Гістограма щотижневої активності. Оновлюється кожні 4 тижні (~1 місяць)

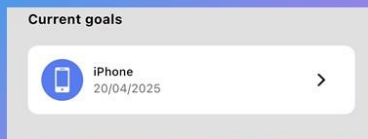


Гістограма щоденної активності. Оновлюється кожен тиждень

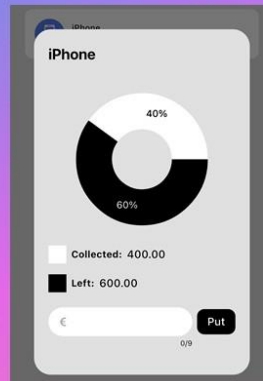


## Скарбничка

9



```
ListView.builder(
  itemCount: dataProvider.goals.length,
  itemBuilder: (context, int i) {
    final filteredTransactions =
      dataProvider.goals.toList();
    final goal = filteredTransactions[i];
    final amount = "${goal['amount']}.toStringAsFixed(2)";
    String iconName = goal['icon'];
    String iconPath = 'assetsg/${iconName}.png';
    final color = Color(int.parse(goal['color']));
    return Padding(
```



```
TextButton(
  onPressed: () {
    double enteredAmount = double.tryParse(amountController.text
      .trim()
      .replaceAll('.', '')) ??
      0.0;
    if (enteredAmount > collected +
      double.parse(widget.amount.replaceAll('.', ''))) {
      showMessage(context, 'The amount is too big');
      return;
    }
    setState(() {
      collected += enteredAmount;
      _saveCollected();
      amountController.clear();
    });
  },
  style: TextButton.styleFrom(
    backgroundColor: Theme.of(context).colorScheme.onSurface,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12)), // RoundedR
  child: CustomTextWidget(
    text: 'Put',
    styleType: TextStyleType.defaultStyle,
    color: Theme.of(context).colorScheme.onPrimary,
    size: 18,
  ), // CustomTextWidget
), // TextButton
```



## Висновки

10

В ході виконання кваліфікаційної роботи була виконана задача розробки мобільного застосунку для слідкування за власним бюджетом із функцією скарбнички.

Застосунок успішно виконує основну мету – надає користувачам простий засіб для управління фінансами.

Результат роботи має деякі недоліки: архітектура, яка використана у проєкті, є дещо примітивною, однак через невеликий масштаб проєкту застосування flat architecture є виправданим та спрощує розробку.

Наведена розробка може поступатися популярним аналогам за функціональністю, але у перспективі можна впровадити вдосконалення:

- інтеграція з банківськими рахунками;
- реалізація хмарного збереження даних;
- можливість синхронізації з різних пристроїв для сімейного/групового використання.

