

РОЗРОБКА ІОС-ЗАСТОСУНКУ
ДЛЯ УПРАВЛІННЯ СПРАВАМИ
З ВИКОРИСТАННЯМ АЛГОРИТМУ
АДАПТИВНОГО ПЛАНУВАННЯ НА
ОСНОВІ ЕМОЦІЙ КОРИСТУВАЧА



Національний університет
«Києво-Могилянська академія»

Доповідачка:
студентка 4 - го року навчання
спеціальності 121
Інженерія програмного забезпечення
Пізь Мар'яна Андріївна

Керівник:
ст. викл. Вознюк Я. І.

Київ - 2025

Актуальність

Потреба в балансі між продуктивністю та ментальним благополуччям

Активне використання екосистеми Apple для персональних цифрових асистентів

Зростання попиту на адаптивні та контекстно-чутливі системи

Розробити інтуїтивно зрозумілий і легкий у користуванні iOS-застосунок для планування справ, що підлаштовує графік до емоцій користувача, пропонує усвідомлені паузи, аналітику емоцій і виконаних завдань та підтримує українську мову.

Наявні рішення



Bearable



Знайдені функціональні обмеження, що потребують допрацювання:

- складність налаштування функціональності для планування;
- неможливість поєднання планів і самопочуття в єдиній системі;
- відсутність адаптивного планування з урахуванням емоцій;
- відсутня локалізація для українських користувачів.

Ідейна концепція застосунку



Логотип застосунку
Moodpace

(з англ. mood – настрій,
pace – темп)

Ідея: організувати день користувача в такому темпі, який відповідає його поточному емоційному стану

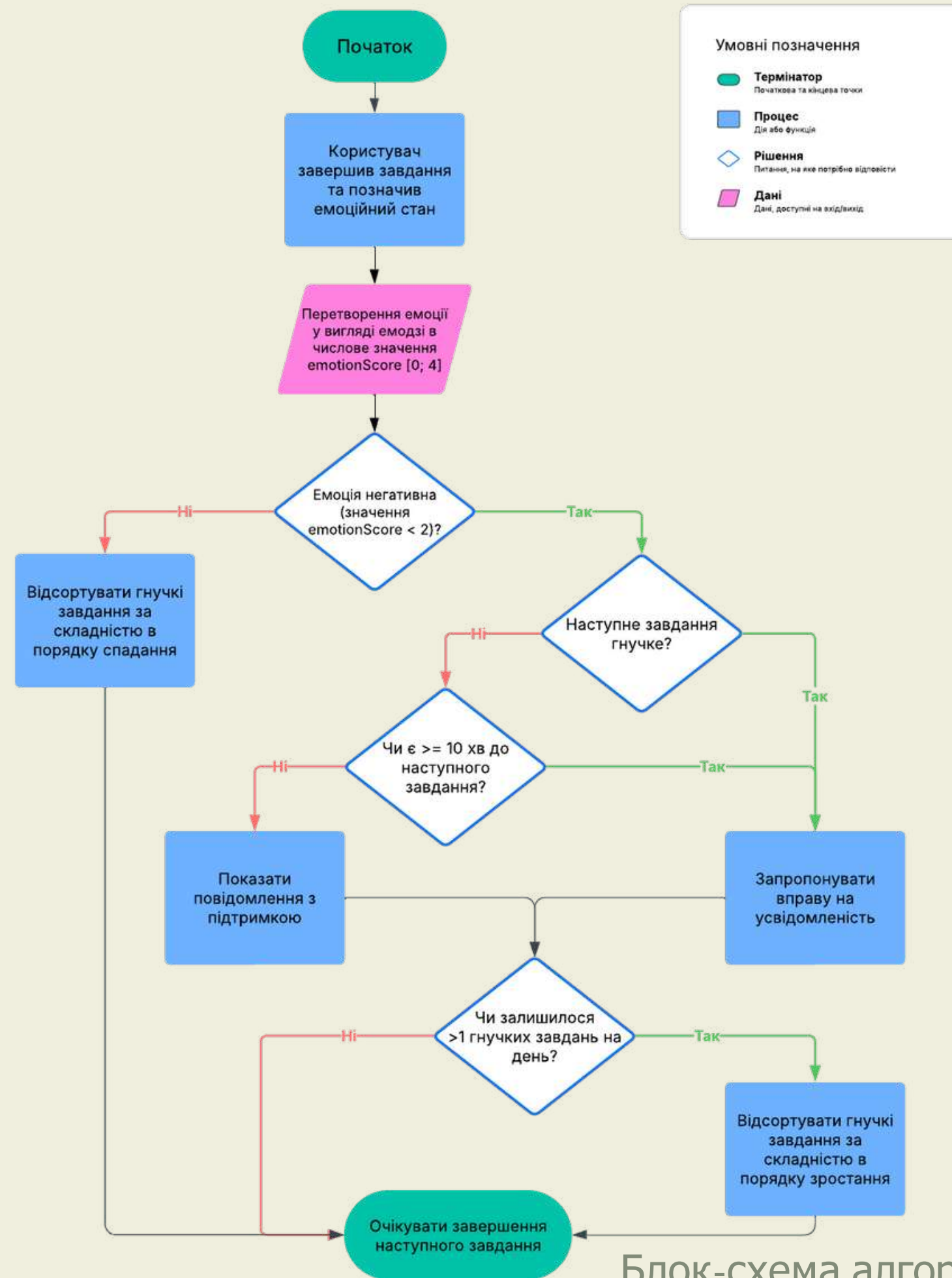
Реалізація:

- два типи завдань: фіксовані та гнучкі
- оцінка емоційного стану користувача після кожного виконаного завдання
- перепланування дня за рахунок гнучких задач та застосування вправ для усвідомленості й заспокоєння

Кольорова гама застосунку

Алгоритм адаптивного планування

- Враховує емоційний стан після кожного завдання
- Пропонує вправу на усвідомленість у разі негативних емоцій та наявності часу
- Динамічно змінює порядок завдань за складністю залежно від поточної емоції



```
enum Emotion: String, Codable, CaseIterable {
    case verySad
    case sad
    case neutral
    case happy
    case veryHappy
}

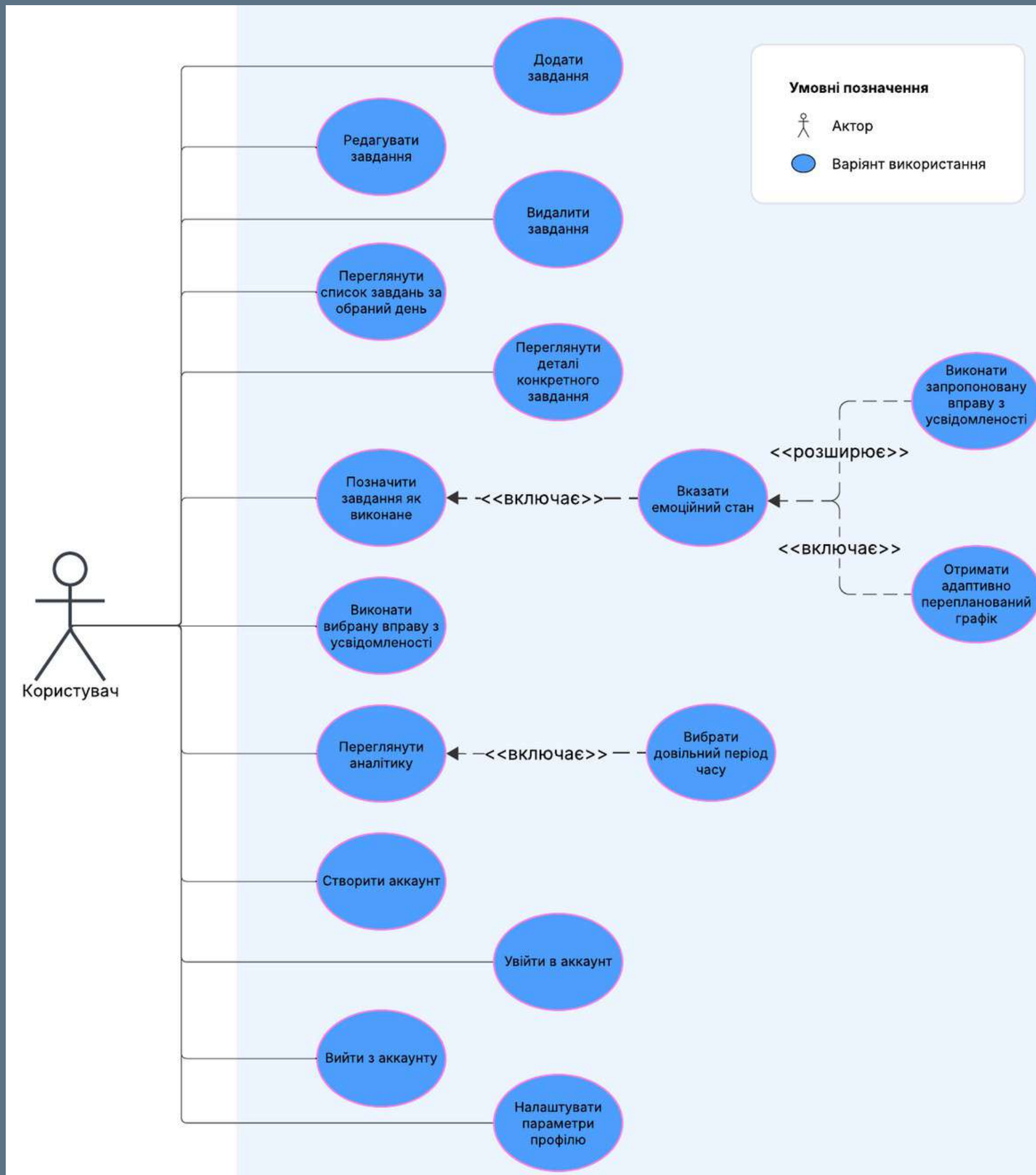
var emotionScore: Int {
    switch self {
        case .verySad:
            0
        case .sad:
            1
        case .neutral:
            2
        case .happy:
            3
        case .veryHappy:
            4
    }
}
```

Числове представлення емоційного стану

Блок-схема алгоритму дій системи після завершення завдання

Варіанти використання застосунку

Діграма варіантів використання



Використані технології



Swift
мова програмування



SwiftUI
фреймворк для побудови
користувацького інтерфейсу



Xcode
середовище розробки

SwiftData
фреймворк для роботи з
локальними даними



SwiftFormat
інструмент для
форматування Swift-коду

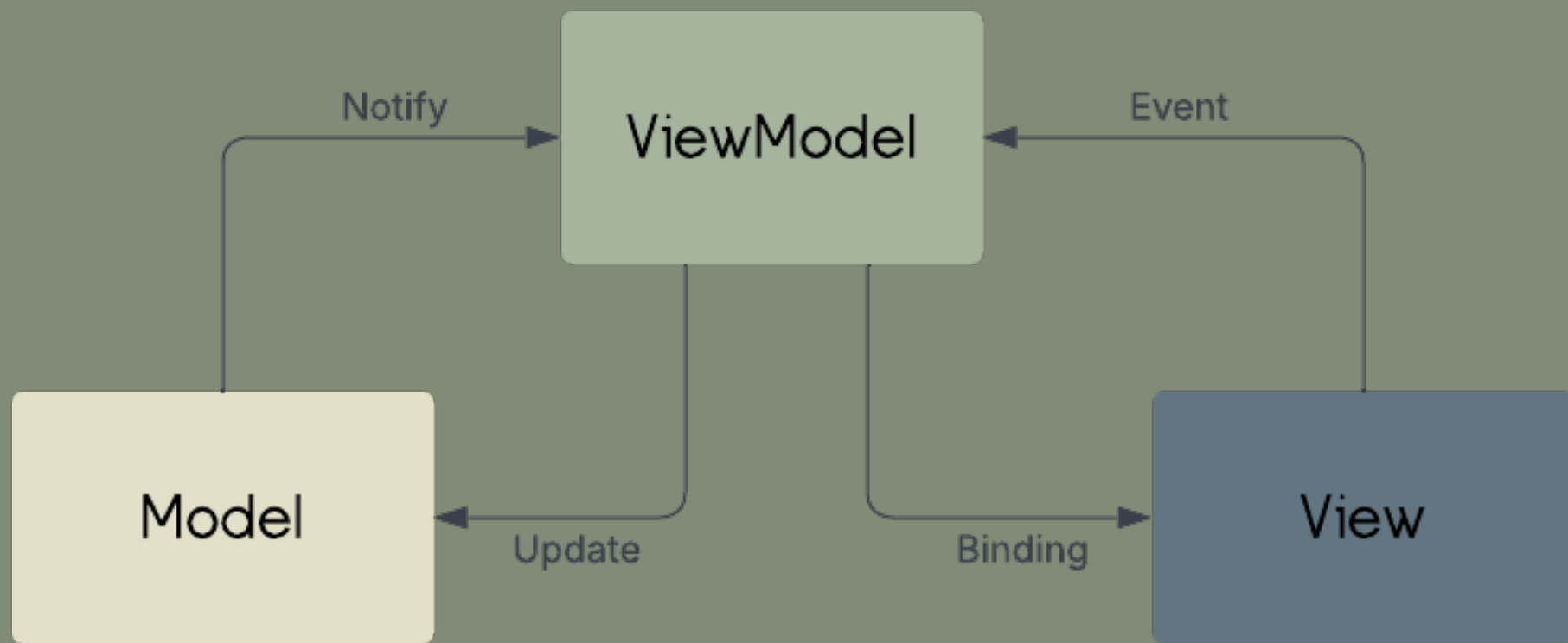


SwiftLint
інструмент для статичного
аналізу Swift-коду



Архітектура застосунку

M V V M



Структура проєкту



Навігаційна панель

На основі 4 основних категорій варіантів використання розроблено навігаційну панель застосунку:

```

TabView {
  CalendarView(manager: taskManager)
  .tabItem {
    Label("Planning", systemImage: "calendar")
  }

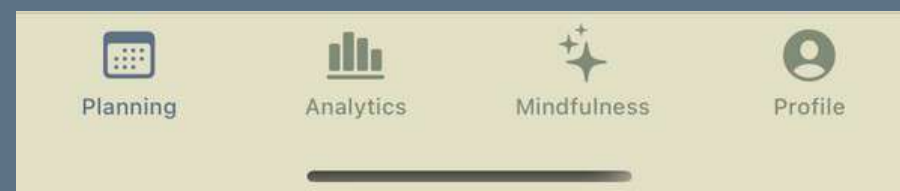
  AnalyticsView(taskManager: taskManager)
  .tabItem {
    Label("Analytics", systemImage: "chart.bar.xaxis")
  }

  MindfulnessView()
  .tabItem {
    Label("Mindfulness", systemImage: "sparkles")
  }

  ProfileView()
  .environmentObject(taskManager)
  .tabItem {
    Label("Profile", systemImage: "person.crop.circle")
  }
}

```

Фрагмент коду для створення навігаційної панелі



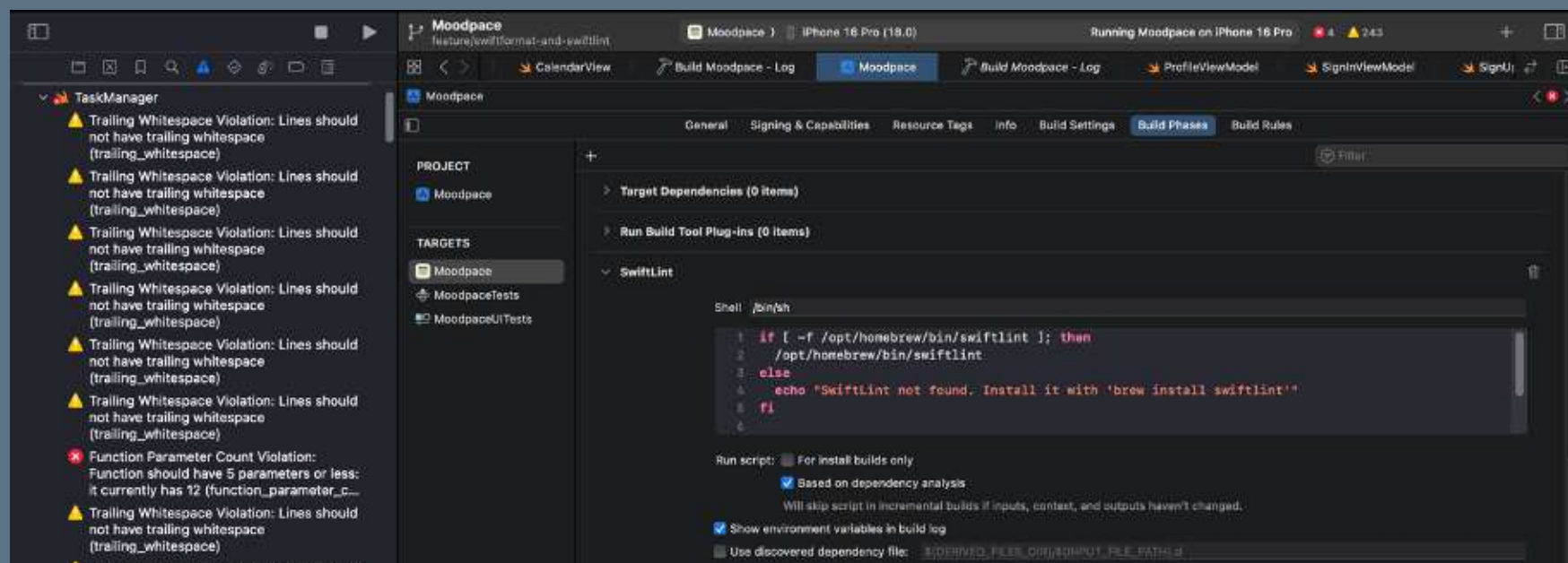
Навігаційна панель

Усередині вкладок використовуються різні види навігації між екранами та вікнами:

- NavigationStack + navigationDestination(item:)
- .sheet(isPresented:)
- .overlay + свій View

Автоматизована перевірка стилю й форматування коду

SwiftLint інтегровано до проєкту як Run Script Build Phase в Xcode.



Скриншот Xcode з інтегрованим SwiftLint

SwiftFormat інтегровано до процесу розробки як Git pre-commit hook.

```
(base) mapthree@Marianas-Laptop Moodpace % git status
On branch feature/swiftformat-and-swiftlint
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
       modified:   Moodpace/UserStory/Planning/Calendar/CalendarView.swift

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
       modified:   Moodpace/UserStory/Planning/Calendar/TaskCard.swift

(base) mapthree@Marianas-Laptop Moodpace % git commit -m "Make animations smoother"
Running SwiftFormat...
SwiftFormat completed successfully.
Reformatted Moodpace/UserStory/Planning/Calendar/CalendarView.swift with swiftformat --swiftversion 6 stdin --stdinpath '{}'
[feature/swiftformat-and-swiftlint 2441659] Make animations smoother
```

Приклад роботи Git pre-commit hook із SwiftFormat у терміналі

ДЕМОНСТРАЦІЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ



Взаємодія із завданнями

- Створення нового завдання
- Перегляд усіх завдань на календарі
- Перегляд деталей конкретного завдання з можливістю редагувати та видалити його

гнучке завдання

Title	
Description	
Flexible	Fixed
Category	Other ▾
Difficulty	Medium ▾
Estimated Duration, min	30 ▾

фіксоване завдання

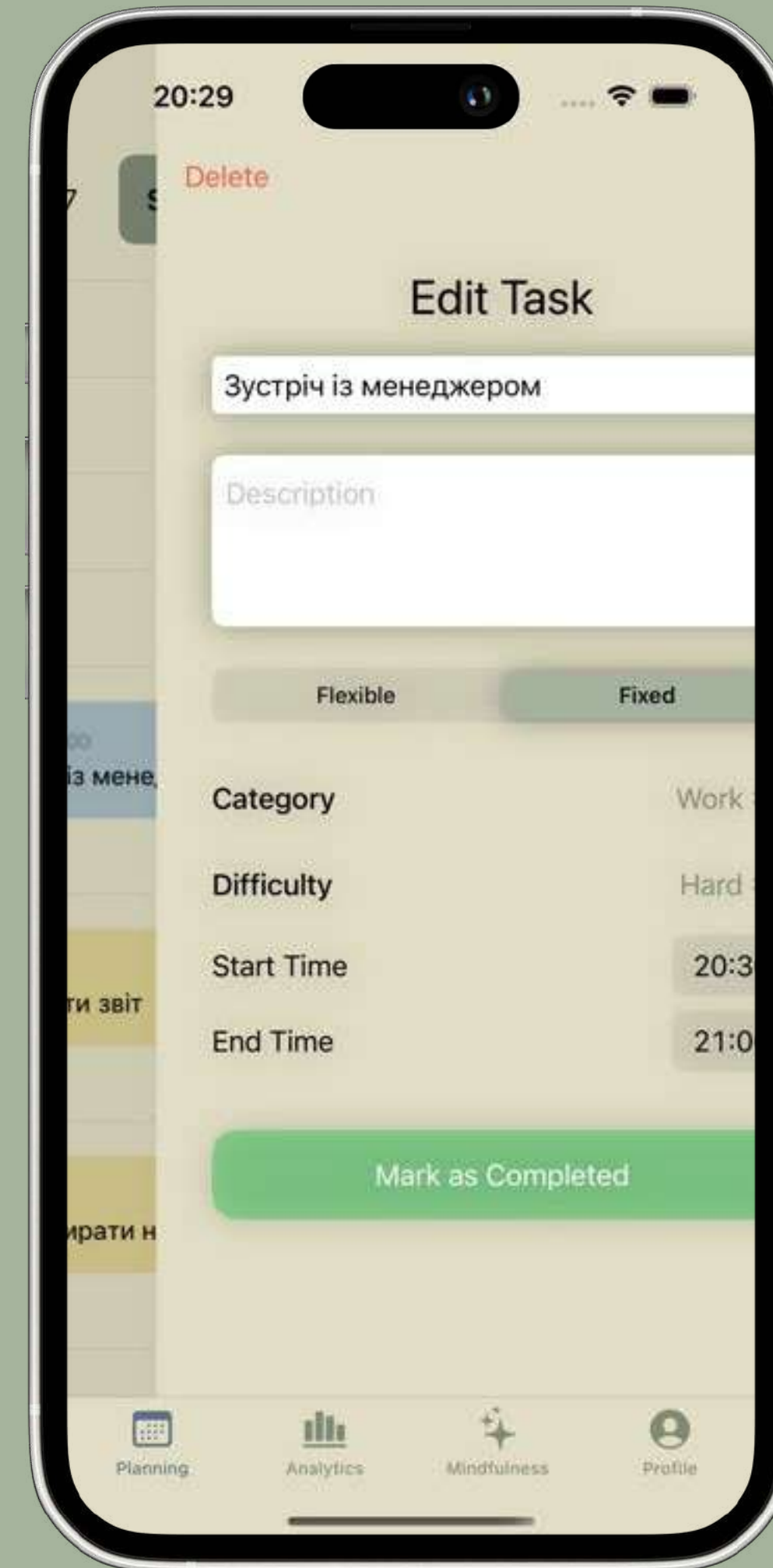
Title	
Description	
Flexible	Fixed
Category	Other ▾
Difficulty	Medium ▾
Start Time	08:00
End Time	09:00



Адаптивне планування завдань

Після позначення завдання виконаним у дію вступає алгоритм адаптивного планування:

- користувачеві пропонується виконати вправу на усвідомленість;
- наявні гнучкі завдання пересортовуються за складністю в порядку зростання.



Аналітика

За довільний період часу можна переглянути:

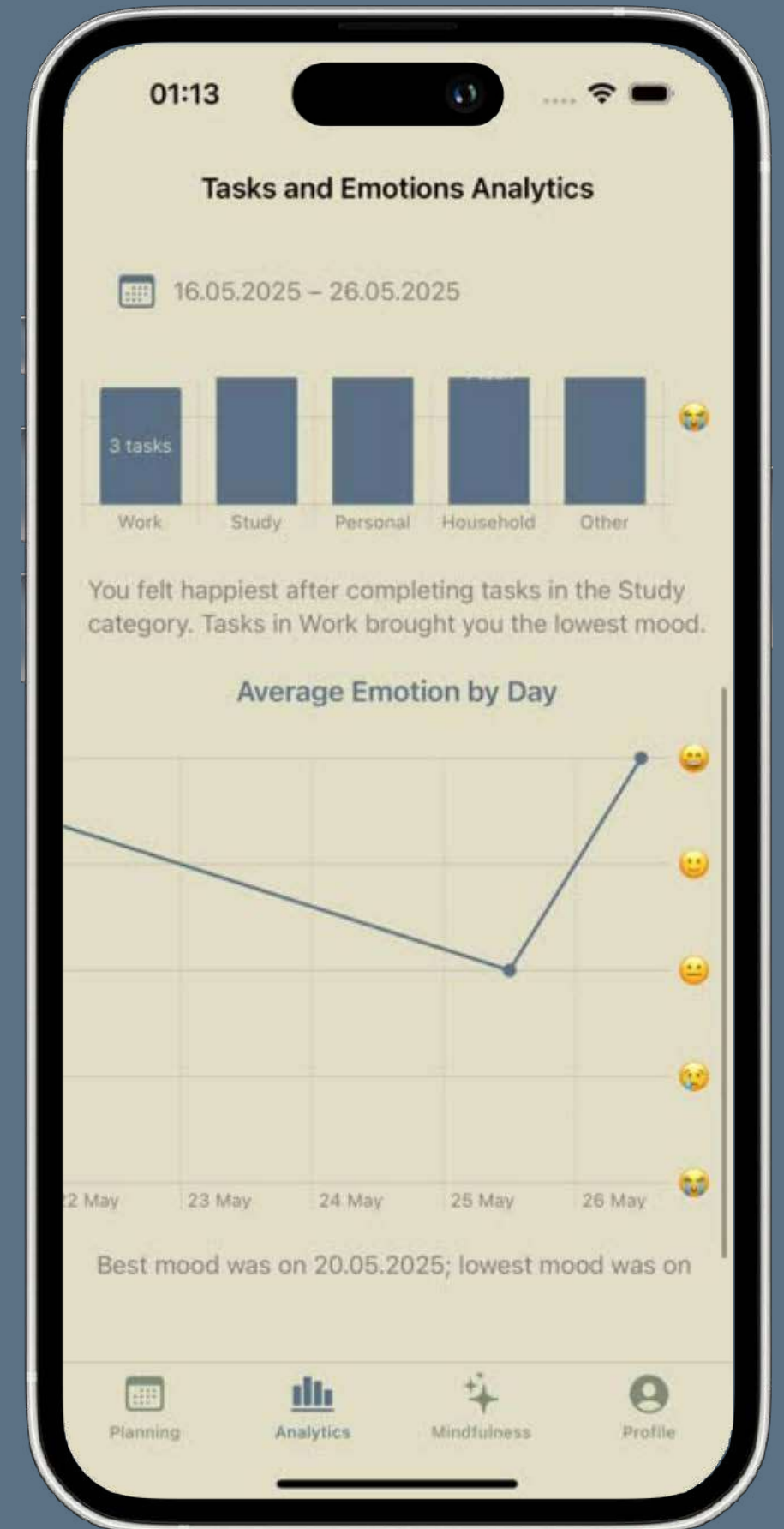
- кількість виконаних та невиконаних завдань;
- стовпчикову діаграму середнього настрою за категоріями завдань;
- лінійну діаграму середнього настрою після завдань за день.

Побудова графіків реалізована за допомогою системного фреймворку Swift Charts:

```
AnalyticsView.swift

Chart {
  ForEach(stats.filter(\.hasData)) { stat in
    LineMark(
      x: .value("Date", stat.date, unit: .day),
      y: .value("Average Emotion", stat.averageEmotion)
    )
    .foregroundColor(AppColors.additional)
  }
  ForEach(stats.filter(\.hasData)) { stat in
    PointMark(
      x: .value("Date", stat.date, unit: .day),
      y: .value("Average Emotion", stat.averageEmotion)
    )
    .symbol(Circle())
    .foregroundColor(AppColors.additional)
  }
}
```

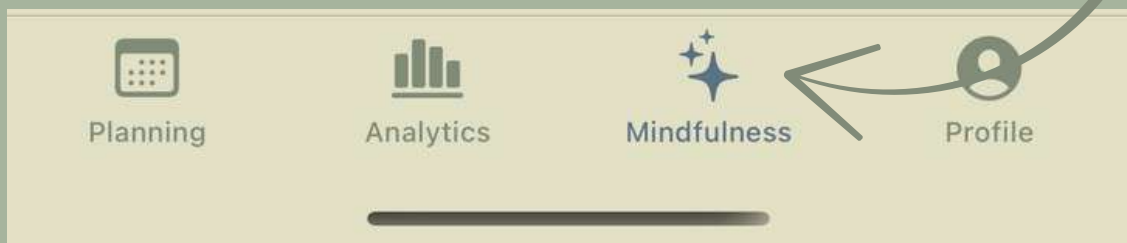
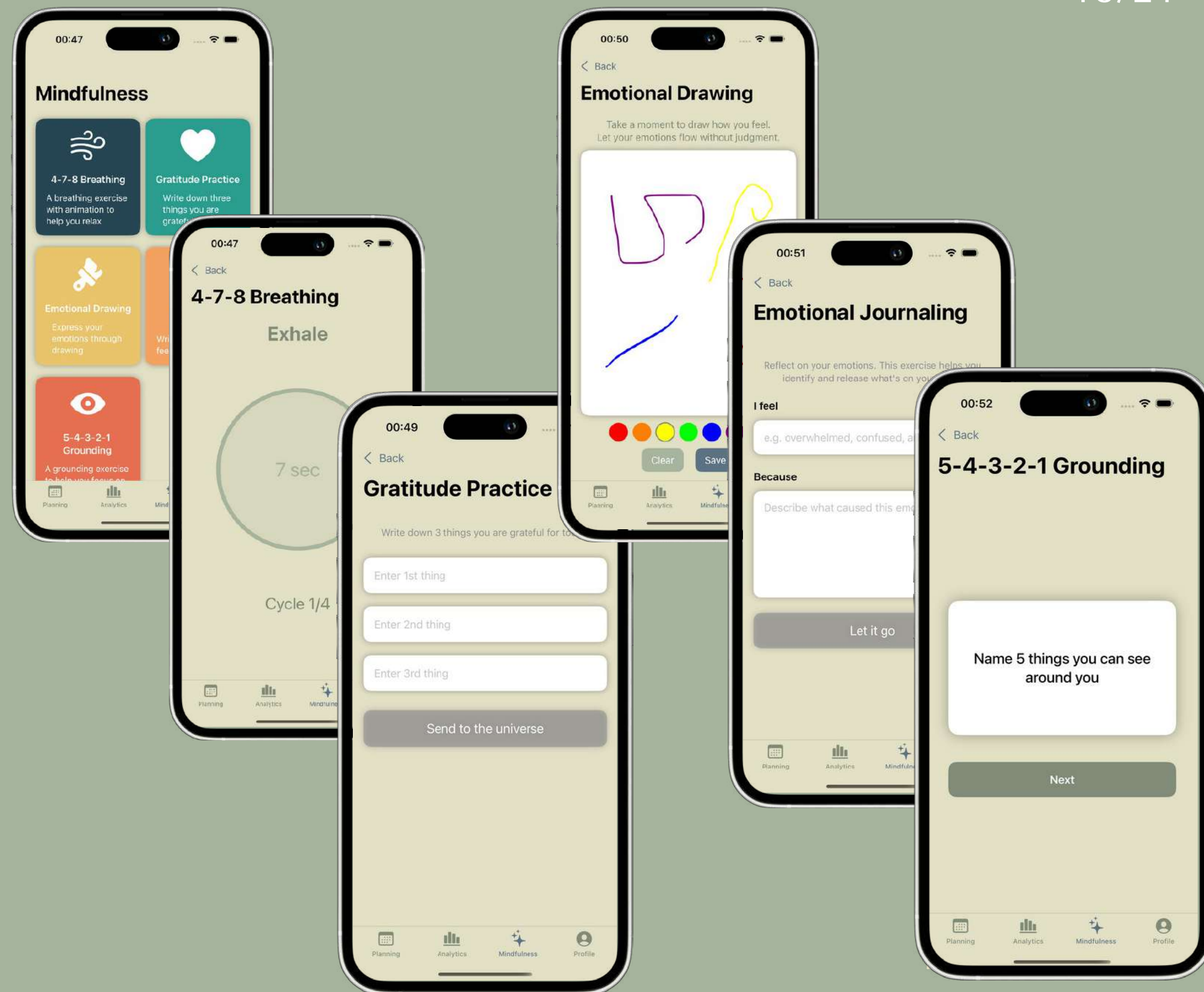
Фрагмент коду побудови графіку середнього настрою за день



Вправи з усвідомленості

Виконання вправ може відбуватися:

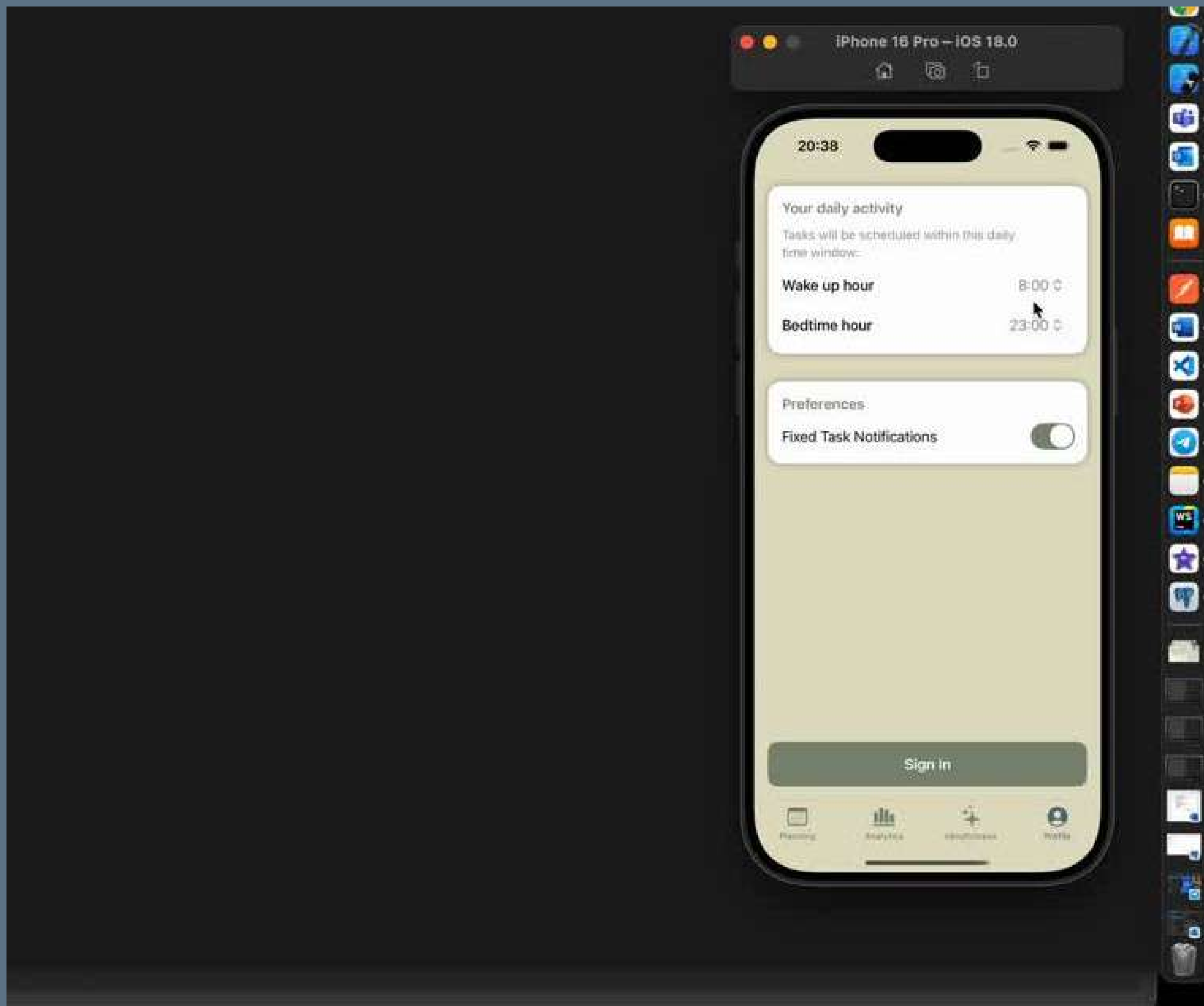
- автоматично як відповідь застосунку на негативний емоційний стан користувача;
- вручну за допомогою відповідної вкладки навігаційної панелі.



Взаємодія з профілем

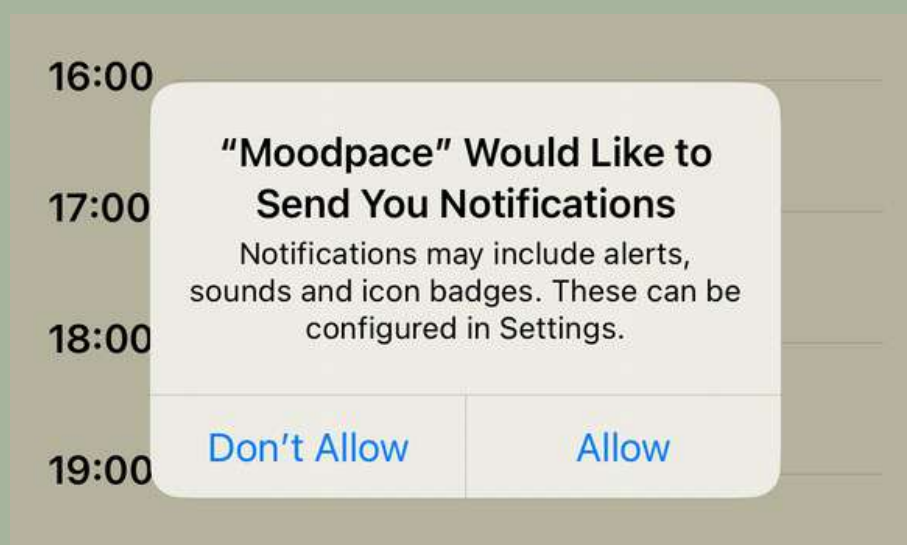
Можливості для користувача:

- налаштування годин активності, що впливають на розміщення гнучких завдань у календарі;
- вибір, чи отримувати нагадування про початок завдань;
- експорт завдань на пошту у форматі CSV (для авторизованого користувача)



Сповіщення

Механізм надсилання локальних сповіщень реалізовано за допомогою системного фреймворку UserNotifications.



На кожне фіксоване завдання користувач отримує 2 сповіщення:

- за 15 хв до початку завдання
- в момент початку завдання



Локалізація

Мова користувацького інтерфейсу визначається системною мовою пристрою.

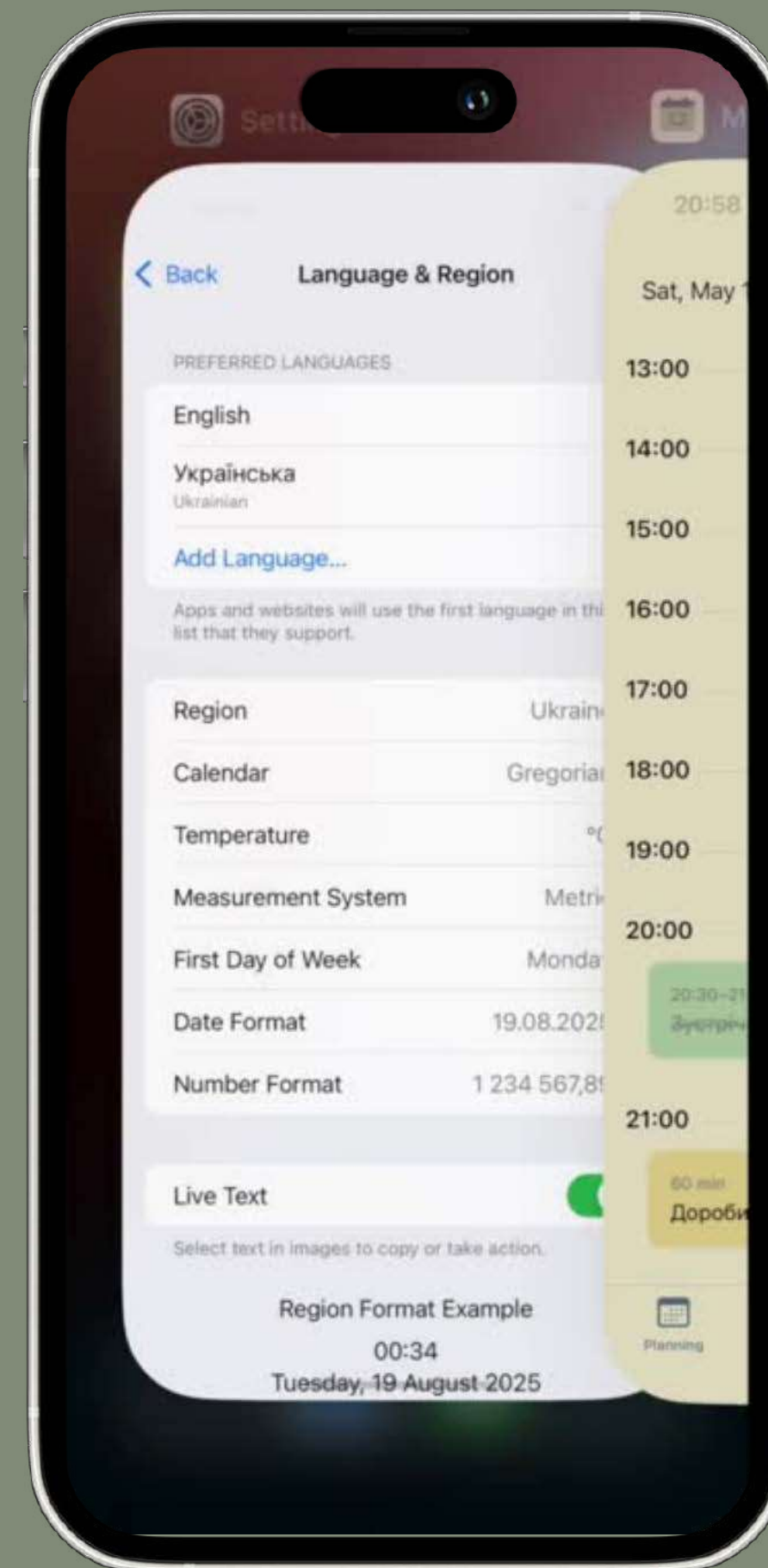
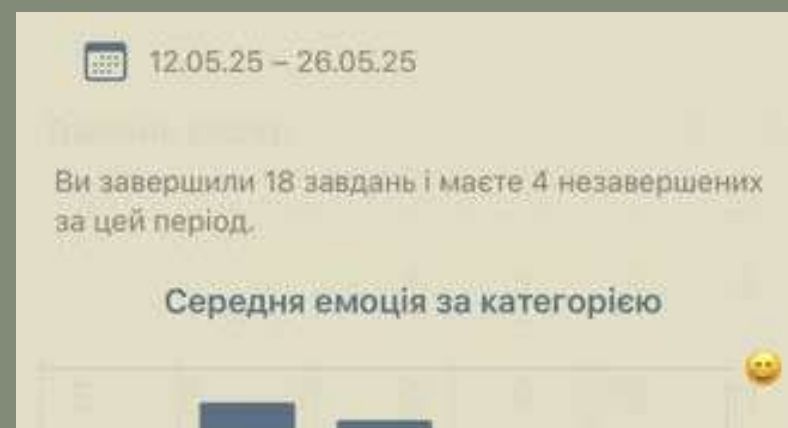
Наразі доступна підтримка двох мов: англійської та української 🇺

Реалізовано з використанням `StringCatalogXcode`:

Key	Default Localization (en)	Ukrainian (uk)	Comment	State
Add Task	Add Task	Додати завдання		✓
Already have an account? Sign In	Already have an account? Sign In	Уже маєте аккаунт? Увійти		✓
Analytics	Analytics	Аналітика		✓
Are you sure you want to delete this task?	Are you sure you want to delete this task?	Ви впевнені, що хочете видалити це завдання?		✓

Зокрема забезпечено підтримку однини та множини:

One	Few	Many
You completed 1 task and have 1 unfinished during this period.	You completed 18 tasks and have 4 unfinished during this period.	You completed 18 tasks and have 4 unfinished during this period.
Ви завершили 1 завдання і маєте 1 незавершених за цей період.	Ви завершили 18 завдань і маєте 4 незавершених за цей період.	Ви завершили 18 завдань і маєте 4 незавершених за цей період.



Поставленої мети досягнуто - реалізовано iOS-застосунок для планування завдань із урахуванням емоційного стану користувача. Унікальною рисою застосунку є використання розробленого алгоритму адаптивного планування.

У процесі дослідження проаналізовано наявні застосунки в нішах продуктивності й ментального здоров'я. Враховано функціональні й нефункціональні вимоги до розробки власного рішення:

- архітектура M V V M забезпечила чіткий розподіл зон відповідальності між компонентами системи;
- SwiftUI дозволив побудувати користувацький інтерфейс із високою продуктивністю рендерингу;
- SwiftData забезпечив роботу з локальними даними зі зручною інтеграцією зі SwiftUI;
- SwiftLint і SwiftFormat - якість та єдиний стиль коду;
- UserNotifications - локальні сповіщення - нагадування про початок завдання;
- String Catalog - українську локалізацію.

Перспективи для подальшого розвитку:

- використання методів машинного навчання для покращення алгоритму адаптивного планування;
- інтеграція з інструментами збору даних показників здоров'я для врахування їх при плануванні

**ДЯКУЮ
ЗА УВАГУ!**