|  |  |
| --- | --- |
| ДИСЦИПЛИНА | Программирование корпоративных систем |
| ИНСТИТУТ | Институт перспективных технологий и индустриального программирования |
| КАФЕДРА | Кафедра индустриального программирования |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Практические задание |
| ПРЕПОДАВАТЕЛЬ | Адышкин Сергей Сергеевич |
| СЕМЕСТР | 5 семестр, 2025-2026 гг. |

**Практическое занятие № 6**

**UI/UX дизайн. Верстка приложения по готовому дизайну (Figma → Flutter).**

**Цели занятия**

* Освоить процесс переноса готового дизайна из Figma в Flutter.
* Изучить принципы адаптивной верстки мобильного интерфейса.
* Научиться использовать стили, отступы и компоненты в соответствии с UI-гайдом.
* Сверстать основные экраны мобильного приложения на основе предоставленного макета **MAD Shopp Mobile App Design**.
* Получить навыки настройки навигации между экранами.

**Теоретическая часть**

1. **Связь между UI/UX и версткой.**

UI (User Interface) — визуальная часть, UX (User Experience) — взаимодействие. Хороший дизайн — это не только эстетика, но и логика. Задача верстальщика — перенести визуальные решения в код, сохранив композицию, цветовую палитру и юзабилити.

1. **Экспорт параметров из Figma.**

В Figma можно получить значения:

– цвета (HEX),

– отступы (Padding/Margin),

– размеры и радиусы,

– шрифты (название, вес, размер).

Для ускорения верстки удобно использовать плагины: **Figma → Flutter**, **Figma Tokens**, **Design to Code**.

1. **Примерная структура проекта Flutter для UI-разметки.**

/lib

/screens // экраны приложения

/widgets // переиспользуемые компоненты

/theme // стили, цвета, шрифты

main.dart

Такой подход обеспечивает масштабируемость и чистую архитектуру.

1. **Основные виджеты для верстки:**
   * Container, Padding, Align, Center — компоновка;
   * Column, Row, Stack — размещение элементов;
   * Image.asset — изображения;
   * Text, TextStyle — стилизация текста;
   * ListView, SingleChildScrollView — прокрутка;
   * Navigator.push() — переход между экранами.
2. **Адаптивность и плотность пикселей.**

Flutter автоматически масштабирует UI, но важно использовать относительные размеры (MediaQuery, Expanded, Flexible) вместо жёстких чисел.

1. **Цветовая палитра и шрифты из Figma.**

В Flutter их удобно вынести в отдельный файл:

class AppColors {

static const primary = Color(0xFF1E88E5);

static const accent = Color(0xFFFFC107);

static const background = Color(0xFFF5F5F5);

}

class AppTextStyles {

static const title = TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

);

}

1. **Навигация между экранами.**

Для упрощения можно использовать Navigator.push / pop, или библиотеку go\_router.

**Практическая часть**

**Задание:** сверстать приложение по готовому макету

[Ссылка на дизайн](https://www.figma.com/design/6wiKJ3IQ7GAop6BI6jGrA8/MAD-Shopp-Mobile-App-Design?node-id=0-1&p=f&t=VzuAPCFs4qxxu9Fl-0)

**Алгоритм выполнения**

1. **Создание проекта.**

flutter create madshop\_ui\_<Фамилия>

Открыть в VS Code / Android Studio.

1. **Подготовка структуры (шаблон).**

lib/

main.dart

screens/

home\_screen.dart

product\_screen.dart

cart\_screen.dart

widgets/

product\_card.dart

theme/

colors.dart

text\_styles.dart

1. **Импорт ресурсов.**

– Добавить изображения из Figma (иконки, фото товаров) в assets/images/.  
– Зарегистрировать их в pubspec.yaml.

1. **Верстка экранов.**
   * **Главный экран**: список карточек товаров (GridView или ListView.builder).
   * **Экран товара**: изображение, описание, цена, кнопка «Добавить в корзину».
   * **Корзина**: список выбранных товаров, итоговая сумма.
2. **Навигация.**  
   Использовать Navigator.push(context, MaterialPageRoute(builder: …)).
3. **Работа со стилями.**
   * Вынести цвета и шрифты в отдельные файлы.
   * Использовать ThemeData для единого стиля приложения.
4. **Проверка соответствия дизайну.**
   * Расстояния и цвета — по Figma;
   * Проверить отступы (EdgeInsets.symmetric, EdgeInsets.all);
   * Протестировать на разных размерах экранов.

**Контрольные точки**

1. Проект запускается без ошибок.
2. Главный экран сверстан по макету.
3. Работает переход на экран товара и в корзину.
4. Используются собственные цвета и шрифты.
5. Интерфейс визуально соответствует Figma.

**Контрольные задания**

* Скриншот всех экранов
* Переход между экранами.
* Краткий отчёт (2–3 стр.):
  + какие экраны сверстаны;
  + какие элементы использовались;
  + какие отличия от макета остались и почему.

**Требования к отчёту**

* Код приложения должен быть залит на Git.
* В ReadMe cкриншоты всех ключевых экранов и показ переходов между экранами.
* Краткий анализ: какие решения по UI/UX оказались удачными, какие — нуждаются в доработке.