

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное
автономное образовательное
учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ

«UX\UI дизайн»

Направление подготовки 09.03.04 «Программная инженерия»
Направленность (профиль) «Инженерия сложных программных систем»
Квалификация выпускника: бакалавр

Ставрополь

2026

Введение

Цель изучения дисциплины «UX\UI дизайн» является подготовка будущих специалистов в области цифрового дизайна, способных разрабатывать интуитивно понятные интерфейсы и улучшать взаимодействие пользователей с цифровыми продуктами, а также компетенций будущего бакалавра по направлению подготовки 09.03.04 Программная инженерия.

Задачами обучения являются:

- изучение основ UX\UI-дизайна, его принципов, методологий и современных тенденций;
- изучение процесса исследования пользователей, выявления их потребностей и построения User Personas;
- изучение методов проектирования пользовательского пути (Customer Journey Map) для создания удобного взаимодействия с продуктом;
- изучение принципов информационной архитектуры и основ wireframing (создания каркасных макетов);
- обретение навыков работы с инструментарием Figma для проектирования интерфейсов, создания интерактивных прототипов и совместной работы над проектами;
- изучение основ типографики, цветовых решений, дизайн-систем и UI-китов для создания визуально целостных и удобных интерфейсов;
- обретение навыков оформления и презентации финального проекта, подготовки макетов к передаче разработчикам.

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
-------------------------------	------------------------------	---

ПК-8 – Способен адаптировать и применять методы искусственного интеллекта и машинного обучения для решения профессиональных задач, включая интеграцию готовых моделей в программные продукты	ИД-1_{ПК-8} Проектирует пользовательские интерфейсы с учетом возможности персонализации на основе ML-алгоритмов (рекомендательные системы, предсказание действий пользователя, адаптивный контент) ИД-2_{ПК-8} Разрабатывает сценарии взаимодействия (user flows), включающие обработку и визуализацию результатов работы AI-моделей (например, отображение распознанных объектов, confidence scores, альтернативных вариантов) ИД-3_{ПК-8} Учитывает в дизайне интерфейсов особенности представления данных, полученных от ML-моделей: индикаторы уверенности, пояснения результатов (explainable AI), обработку неопределенности ИД-4_{ПК-8} Проводит юзабилити-тестирование интерфейсов с AI-функциями, оценивая понятность их работы для пользователя, выявляя проблемы восприятия и доверия к рекомендациям системы	Результаты проектирования пользовательского интерфейса с учетом возможности персонализации на основе ML-алгоритмов. Не владеет необходимыми навыками и знанием проектирования пользовательского интерфейса для достижения результата. Результаты демонстрация умения разрабатывать сценарии взаимодействия (user flows), включающие обработку и визуализацию результатов работы AI-моделей Результаты демонстрация учета в дизайне интерфейсов особенности представления данных, полученных от ML-моделей: индикаторы уверенности, пояснения результатов (explainable AI), обработку неопределенности Результаты умений проводить юзабилити-тестирование интерфейсов с AI-функциями, оценивая понятность их работы для пользователя, выявляя проблемы восприятия и доверия к рекомендациям системы
--	--	--

Практическая работа № 1.

Введение в UI/UX-дизайн

Цель: познакомиться с основами UI/UX-дизайна, понять разницу между UI и UX, изучить основные принципы проектирования удобных интерфейсов, а также научиться анализировать удачные и неудачные примеры интерфейсов.

Формируемые компетенции: ПК-8

Теоретическое обоснование

UX (пользовательский опыт) – это общее впечатление пользователя от взаимодействия с продуктом. Он включает в себя удобство, логичность, интуитивность и удовлетворенность при использовании интерфейса.

Основные вопросы UX-дизайна:

- Легко ли пользователю понять, как работать с интерфейсом?
- Может ли он быстро выполнить свою задачу?
- Получает ли он обратную связь от системы?
- Насколько приятен процесс взаимодействия?

UI (пользовательский интерфейс) – это внешний вид продукта. Он включает в себя цвета, шрифты, кнопки, иконки, анимации, макеты страниц и другие визуальные элементы.

Основные вопросы UI-дизайна:

- Удобно ли расположены элементы на экране?
- Читаем ли текст?
- Соответствует ли цветовая схема целевой аудитории?
- Логично ли выделены основные элементы?

Пример: если вы заходите на сайт интернет-магазина и быстро находите нужный товар, без труда добавляете его в корзину и оформляете заказ – это хороший UX. Если при этом кнопки красивые, анимации плавные, а цвета приятные глазу – это хороший UI.

Принцип User-Centered Design (UCD) – это проектирование, ориентированное на пользователя. Оно включает несколько этапов:

- 1) Исследование аудитории (проведение интервью, анализ поведения пользователя);
- 2) Создание прототипов (первые наброски интерфейсов);
- 3) Внедрение изменений (исправление ошибок, найденных в ходе тестирования).

Основные UX-ошибки:

- 1) Слишком сложный интерфейс;
- 2) Нет четкого пути выполнения задачи;
- 3) Отсутствие обратной связи от системы;
- 4) Недостаточная адаптивность (плохо работает на мобильных);
- 5) Непонятные иконки и кнопки.

Датский специалист по юзабилити Якоб Нильсен сформулировал 10 эвристик – базовые правила удобных интерфейсов. Они применяются при проектировании и оценке дизайна.

1. Видимость состояния системы

Принцип: система должна всегда информировать пользователя о том, что происходит.

Пример хорошего UX: индикатор загрузки показывает, сколько времени осталось до завершения.

Плохой UX: при нажатии на кнопку ничего не происходит, пользователь не понимает, работает ли система.

2. Соответствие реальному миру

Принцип: интерфейс должен говорить на понятном пользователю языке и использовать знакомые ему образы.

Пример хорошего UX: значок корзины в интернет-магазине напоминает реальную корзину.

Плохой UX: использование непонятных символов без подписей.

3. Контроль и свобода пользователя

Принцип: пользователь должен иметь возможность отменить или повторить действие.

Пример хорошего UX: кнопка "Отменить" в формах ввода.

Плохой UX: пользователь случайно удалил данные, и их невозможно восстановить.

4. Единообразие и стандарты

Принцип: одинаковые элементы должны работать одинаково.

Пример хорошего UX: все кнопки на сайте выглядят одинаково.

Плохой UX: одна и та же кнопка на разных страницах имеет разный дизайн и поведение.

5. Предотвращение ошибок

Принцип: дизайн должен предотвращать ошибки до их возникновения.

Пример хорошего UX: в форме регистрации показываются требования к паролю до его ввода.

Плохой UX: пользователь заполняет форму, но после отправки появляется ошибка, и он не понимает, в чем проблема.

6. Узнаваемость вместо запоминания

Принцип: важно показывать пользователю нужную информацию вместо того, чтобы заставлять его запоминать её.

Пример хорошего UX: автозаполнение формы или подсказки.

Плохой UX: поле для ввода пароля без возможности посмотреть введенный текст.

7. Гибкость и эффективность использования

Принцип: интерфейс должен быть удобен как для новичков, так и для опытных пользователей.

Пример хорошего UX: горячие клавиши и настройки для продвинутых пользователей.

Плохой UX: интерфейс, где нельзя ускорить рутинные действия.

8. Эстетика и минимализм

Принцип: дизайн должен быть лаконичным, без лишних элементов.

Пример хорошего UX: четкая структура страницы, минимум текста.

Плохой UX: загроможденные страницы с многочисленными кнопками, текстами и рекламой.

9. Помощь пользователю в случае ошибки

Принцип: ошибки должны сопровождаться четкими инструкциями по их устранению.

Пример хорошего UX: ошибка с объяснением, например: "Пароль должен содержать не менее 8 символов".

Плохой UX: сообщение "Ошибка 404" без дополнительной информации.

10. Справка и документация

Принцип: пользователь должен легко находить информацию о том, как работать с системой.

Пример хорошего UX: раздел "Помощь" с поиском по вопросам.

Плохой UX: полное отсутствие справки.

Эти принципы помогают анализировать интерфейсы и создавать удобные пользовательские продукты.

Методика и порядок выполнения работы

Задание 1. Выберите два примера интерфейсов (один удачный, один неудачный). Это могут быть мобильные приложения, веб-сайты, ПО и т. д.

Задание 2. Заполните таблицу, сравнивая их по ключевым параметрам UX/UI-дизайна.

Задание 3. Сделайте выводы. Какие ошибки есть у неудачного интерфейса, какие принципы UX в нем нарушены?

Критерий	Удачный интерфейс (пример)	Неудачный интерфейс (пример)
Простота и удобство	Интуитивно понятная навигация, минималистичный дизайн, важные элементы выделены.	Сложная структура, слишком много кнопок, перегруженный экран.
Логика расположения элементов	Логичное расположение кнопок, пользователь понимает, куда нажимать.	Кнопки расположены хаотично, важные элементы трудно найти.
Контраст и читаемость	Хорошая цветовая контрастность, удобные шрифты, текст легко читаем.	Плохая контрастность, мелкий текст, сложно различимые цвета.
Обратная связь	Кнопки меняют	Нет обратной связи,

системы	состояние при наведении, есть анимации загрузки, ошибки понятны.	при нажатии ничего не происходит, ошибки не объясняются.
Адаптивность	Отлично работает на разных экранах (мобильный, планшет, десктоп).	Интерфейс не адаптирован под мобильные устройства.
Скорость выполнения задачи	Пользователь может быстро выполнить нужное действие.	Долгая загрузка, слишком много шагов для выполнения одной задачи.

Таблица 1 – пример заполнения анализа интерфейса

Содержание отчета и его форма

Представить отчёт в формате .doc о анализе удачного и неудачного интерфейса.

Вопросы для защиты работы

1. В чем разница между UI и UX?
2. Что такое дизайн, ориентированный на пользователя (User-Centered Design, UCD)? Какие этапы он включает?
3. Какие ошибки UX чаще всего встречаются в интерфейсах? Приведите примеры.
4. Опишите принцип "Видимость состояния системы" из 10 эвристик Нильсена. Почему он важен?
5. Какой эвристике Нильсена соответствует правило "интерфейс должен быть понятен без обучения и запоминания" и как это реализуется в удобных продуктах?
6. Какими способами можно предотвратить ошибки пользователей при взаимодействии с интерфейсом?
7. Что такое обратная связь в интерфейсе и почему она важна?
8. Какие проблемы могут возникнуть, если интерфейс не соответствует принципу "Единообразие и стандарты"?
9. Как влияет эстетика и минимализм на восприятие интерфейса?

Практическая работа № 2.

Исследование пользователей (User Personas)

Цель: Научиться исследовать целевую аудиторию, выявлять потребности и поведение пользователей, а также создавать персоны пользователей (User Personas) для проектирования удобных интерфейсов.

Формируемые компетенции: ПК-8

Основные теоретические сведения

User Persona (персона пользователя) – это собирательный образ реального пользователя, основанный на исследованиях. Персона помогает дизайнерам и разработчикам понять потребности, цели и проблемы аудитории.

Зачем нужны User Personas?

- Они помогают сфокусироваться на реальных потребностях пользователей, а не на субъективных догадках;
- Позволяют делать продукт удобнее за счет учета более и потребностей аудитории;
- Упрощают принятие дизайнерских решений – команда понимает, для кого делает интерфейс;
- Используются для создания пользовательских сценариев (User Flows) и тестирования UX.

Для создания персонажей важно провести исследование пользователей.

Основные методы исследования:

- Анкетирование. Опрос пользователей о предпочтениях, сложностях, опыте работы с продуктами;
- Интервью. Глубокие беседы с пользователями для понимания их целей и более;
- Анализ конкурентов. Изучение отзывов и поведения пользователей на аналогичных платформах.
- Аналитика пользовательского поведения. Использование сервисов

типа Google Analytics, чтобы понять, как люди взаимодействуют с продуктом.

User Persona должна быть конкретной и реалистичной.

Структура User Persona:

1. Имя и фото – делает персонажа более «живым»;
2. Возраст, профессия – помогает понять образ жизни пользователя;
3. Цели – чего хочет пользователь достичь, используя продукт;
4. Трудности – какие проблемы возникают у него при использовании интерфейсов;
5. Поведение – как он принимает решения, какие устройства использует;
6. Цитата – выражает отношение к продукту (например, «Хочу заказывать еду за 2 клика»).

Ошибки при создании User Personas:

1. Фиктивные персонажи – если они основаны на догадках, а не на реальных данных;
2. Слишком общее описание – например, «Анастасия, 25 лет, любит удобные интерфейсы»;
3. Создание только одной персоны – в реальности у продукта всегда несколько типов пользователей;
4. Игнорирование трудностей и мотивации – важно понимать не только, кто пользователь, но и почему он так действует.

После создания персонажей их используют на всех этапах проектирования интерфейсов.

Применение в UI/UX-дизайне:

- Разработка навигации – интерфейс должен соответствовать привычкам пользователя;
- Создание контента – шрифты, кнопки, инструкции адаптируются под целевую аудиторию;
- Тестирование интерфейса – проверяется, насколько удобен продукт

для каждой персоны.

Пример использования персонажа в UX: Персона пожилого пользователя? значит, в интерфейсе нужны крупные кнопки, простая навигация, понятные инструкции.

Задания для выполнения

1. Создать две User Persona для цифрового продукта по выбору (например, мобильного приложения для заказа еды, интернет-магазина, образовательной платформы и т. д.).

Выберите продукт или сервис, для которого будете разрабатывать персон. Это может быть:

1. Мобильное приложение для заказа еды.
2. Онлайн-кинотеатр.
3. Интернет-магазин одежды.
4. Образовательная платформа.
5. Приложение для поиска работы.

Важно: продукт должен решать реальные пользовательские задачи.

Пример: Сервис доставки еды. Его аудитория может включать студентов, офисных работников, семейных людей и пожилых пользователей.

Перед созданием персонажей нужно исследовать целевую аудиторию. Для этого ответьте на вопросы:

1. Кто ваши пользователи? (возраст, профессия, уровень технических навыков).
2. Как они используют продукт? (какие задачи решают, на каких устройствах работают).
3. Какие у них боли и потребности? (что мешает им комфортно пользоваться сервисом).

Способы сбора данных:

- Анкетирование (например, задать вопросы друзьям или провести мини-опрос).
- Интервью (пообщаться с потенциальными пользователями).
- Анализ конкурентов (изучить отзывы на похожие продукты).

Пример исследования аудитории (для сервиса доставки еды):

Студенты: хотят заказывать еду быстро и дешево, ищут промокоды.

Офисные работники: делают заказы заранее, ценят точность доставки.

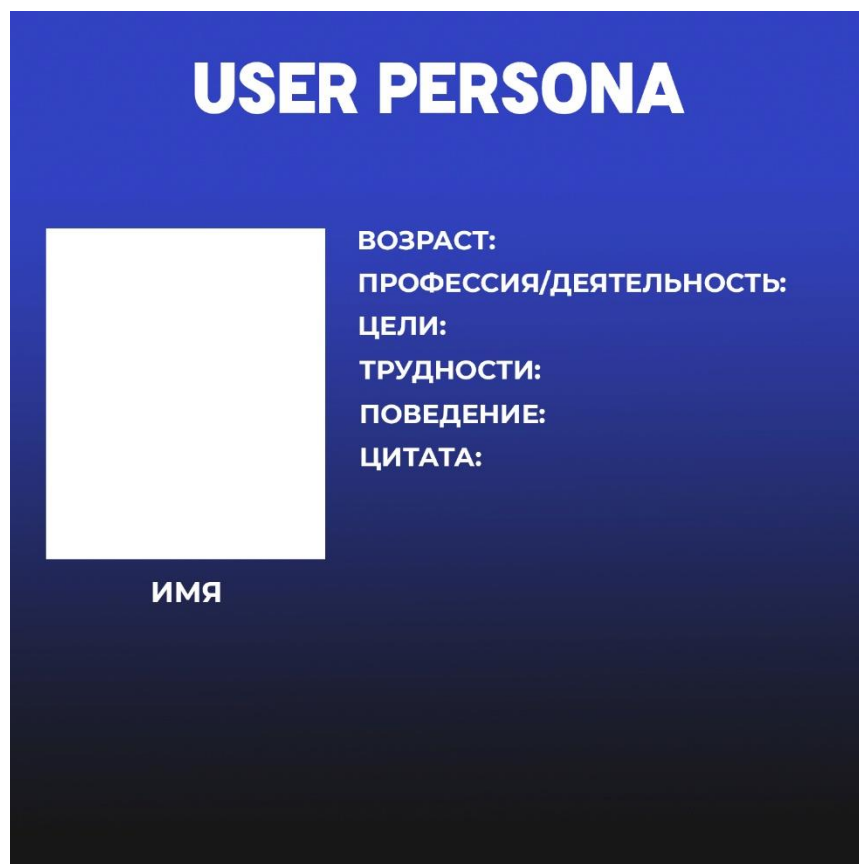
Семейные люди: заказывают на всю семью, выбирают здоровое питание.

Пожилые пользователи: не разбираются в технологиях, им нужен простой интерфейс.

Теперь, на основе исследования, создайте две персонализированные User Persona.

Первая персона – типичный пользователь с одной группой потребностей.

Вторая персона – пользователь с отличающимися потребностями, чтобы показать разнообразие аудитории.

A blue rectangular card with a gradient from dark blue at the bottom to a lighter blue at the top. At the top center, the text "USER PERSONA" is written in bold white capital letters. On the left side, there is a white square placeholder for a profile picture. Below this square, the word "ИМЯ" (Name) is written in white capital letters. To the right of the white square, there is a list of labels in white capital letters: "ВОЗРАСТ:", "ПРОФЕССИЯ/ДЕЯТЕЛЬНОСТЬ:", "ЦЕЛИ:", "ТРУДНОСТИ:", "ПОВЕДЕНИЕ:", and "ЦИТАТА:". Each label is followed by a blank space for text entry.

USER PERSONA

ИМЯ

ВОЗРАСТ:
ПРОФЕССИЯ/ДЕЯТЕЛЬНОСТЬ:
ЦЕЛИ:
ТРУДНОСТИ:
ПОВЕДЕНИЕ:
ЦИТАТА:

Рисунок 1 – Шаблон User Persona

Пример 1: Студент, который часто заказывает еду

USER PERSONA



ВОЗРАСТ: 21 год

ПРОФЕССИЯ/ДЕЯТЕЛЬНОСТЬ:
Студент

ЦЕЛИ: Найти дешевую и быструю доставку еды

ТРУДНОСТИ: Часто не хватает денег, неудобно заполнять длинные формы заказа

ПОВЕДЕНИЕ: Пользуется телефоном, часто заказывает ночью, ищет промокоды

ЦИТАТА: "Хочу заказать бургер за минуту и без сложностей!"

МАКСИМ ИВАНОВ

Рисунок 2 – Пример User Persona 1

Пример 2: Офисный работник, который заказывает еду в обеденный перерыв

USER PERSONA



ВОЗРАСТ: 34 года

ПРОФЕССИЯ/ДЕЯТЕЛЬНОСТЬ:
Менеджер по продажам

ЦЕЛИ: Заказывать еду в офис без задержек

ТРУДНОСТИ: Доставки часто опаздывают, трудно выбрать ресторан среди множества предложений

ПОВЕДЕНИЕ: Заказывает еду заранее, предпочитает проверенные рестораны

ЦИТАТА: "Важно, чтобы доставка была вовремя, а еда – качественной."

ЕКАТЕРИНА СМЕРНОВА

Рисунок 3 – Пример User Persona 2

Ответьте на вопросы:

1. Какие различия между созданными персонажами?
2. Какие элементы интерфейса должны учитывать их особенности?
3. Как можно улучшить продукт, чтобы он подходил разным пользователям?

Содержание отчета и его форма

Представить отчёт в формате .doc о создании двух User Persona для цифрового продукта по выбору.

Вопросы для защиты работы

1. Что такое User Persona и какую роль она играет в UX-дизайне?
2. Какие основные методы исследования пользователей используются для создания User Personas?
3. Какие ключевые элементы включает в себя User Persona?
4. Как User Personas помогают в проектировании удобных интерфейсов?
5. Какие ошибки часто допускаются при создании персон пользователей? Приведите примеры.
6. Почему важно основываться на реальных данных, а не на догадках при создании User Personas?
7. Как различие в потребностях разных пользователей влияет на интерфейс цифрового продукта?
8. Как можно использовать User Personas на разных этапах UX-дизайна?
9. Какие факторы следует учитывать при создании персон для мобильных приложений?
10. Как бы вы адаптировали интерфейс, если бы основным пользователем был пожилой человек?

Практическая работа № 3.

Карта пользовательского пути (Customer Journey Map)

Цель и содержание работы: Понять, как пользователи взаимодействуют с продуктом. Узнать, что такое CJM и как ее строить; проанализировать проблемные точки взаимодействия с интерфейсом; определить цели, действия, точки контакта и возможные проблемы на каждом этапе; оформить карту CJM в виде таблицы; проанализировать карту и предложить улучшения UX.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Customer Journey Map (CJM) — это визуальная карта пути пользователя при взаимодействии с продуктом или услугой.

Она показывает все этапы, через которые проходит человек, описывает его цели, эмоции, трудности и точки контакта с сервисом. CJM помогает понять, как можно улучшить пользовательский опыт (UX), устранив проблемы и упрощая важные процессы.

Основные цели CJM:

- Глубокое понимание аудитории – показывает реальные сценарии использования продукта;
- Выявление проблемных зон – помогает обнаружить точки, где пользователь сталкивается с трудностями;
- Оптимизация интерфейса – позволяет адаптировать UX под реальные потребности пользователей;
- Улучшение пользовательского опыта – повышает удовлетворенность клиентов, снижая их негативные эмоции;
- Принятие обоснованных решений – дает фактическую информацию для улучшений продукта.

Чтобы карта была полезной, она должна содержать ключевые элементы пользовательского пути.

Элемент	Описание
Этапы пути пользователя	Основные шаги, которые проходит пользователь (например, поиск информации → регистрация → покупка → поддержка).
Цели пользователя	Чего хочет достичь пользователь на каждом этапе? Какие у него мотивы?
Действия пользователя	Что делает пользователь на каждом этапе? Например, вводит данные, нажимает на кнопку «Оформить заказ».
Эмоции и боли	Что испытывает пользователь на каждом этапе? Радость, раздражение, сомнение? Какие у него возникают трудности?
Точки контакта	Через какие каналы пользователь взаимодействует с продуктом (мобильное приложение, сайт, звонок в поддержку и т. д.)?
Возможности для улучшения	Какие UX-проблемы можно устранить? Как сделать продукт удобнее?

Таблица 1 – Ключевые элементы пользовательского пути

Этапы построения Customer Journey Map.

1. Определение цели карты.

Перед тем как строить карту, нужно задать несколько вопросов:

- Какой сценарий взаимодействия анализируем?
- Какие проблемы хотим выявить?
- Для кого создаем карту?

2. Определение User Persona.

Для каждой карты необходимо выбрать персону пользователя – целевого клиента. Можно использовать персонажей из Лабораторной работы 2.

3. Разделение пути пользователя на этапы

Например, если CJM строится для покупки билетов в кино через мобильное приложение, можно выделить такие этапы:

- Поиск подходящего кинотеатра
- Выбор фильма
- Выбор мест
- Оплата

– Получение билетов

4. Заполнение CJM данными

Заполняем таблицу, добавляя действия, эмоции и трудности пользователя.

Этап	Цель пользователя	Действия	Проблемы	Точки контакта	Возможности для улучшения
1. Поиск кинотеатра	Найти ближайший кинотеатр с удобным временем	Открываем приложение, используем поиск	Раздражение из-за сложных фильтров	Приложение, веб-сайт	Добавить геолокацию для быстрого поиска
2. Выбор фильма	Найти интересный фильм	Просматриваем афишу, читаем отзывы	Непонятное описание фильмов	Приложение, сайт	Добавить короткие видеобзоры
3. Выбор мест	Забронировать лучшие места	Выбираем места на схеме зала	Медленная загрузка схемы, маленькие иконки	Мобильное приложение	Улучшить дизайн схемы зала
4. Оплата	Оплатить билеты	Вводим карту, ждем подтверждение	Боязнь ошибок при оплате	Приложение, SMS, e-mail	Добавить подтверждение оплаты через push-уведомления
5. Получение билетов	Войти в зал по билету	Показываем QR-код на входе	Боязнь забыть билет	Приложение, e-mail	Автоматическое добавление билетов в Apple Wallet/Google Pay

Таблица 2 – CJM данные

5. Анализ данных

- Где пользователь испытывает дискомфорт?
- Какие шаги можно упростить?
- Как улучшить эмоции пользователей?

CJM помогает:

- Оптимизировать интерфейсы под реальные сценарии;
- Улучшить навигацию и пользовательский опыт;
- Найти проблемные места в продукте;
- Разрабатывать более удобные и интуитивные интерфейсы.

Ошибки при создании CJM:

- Обобщение пользователей – важно учитывать разные категории клиентов;
- Игнорирование эмоций – CJM должна отображать не только действия, но и чувства пользователя;
- Недостаток точек контакта – продукт может использоваться на разных платформах (сайт, приложение, соцсети);
- Отсутствие данных – CJM должна основываться на реальных исследованиях.

Методика выполнения лабораторной работы

Задание: Создать Customer Journey Map для одного из следующих сценариев (на выбор):

1. Оформление заказа в интернет-магазине.
2. Регистрация в онлайн-сервисе.
3. Запись к врачу через сайт.
4. Бронирование номера в отеле
5. Оформление подписки на стриминговый сервис
6. Заказ еды через мобильное приложение
7. Покупка авиабилетов онлайн
8. Бронирование столика в ресторане
9. Аренда автомобиля через онлайн-сервис
10. Регистрация на онлайн-курс

Создайте таблицу, заполнив ее ключевой информацией.

Этап	Цель пользователя	Действия	Проблемы	Точки контакта	Возможности для улучшения
1.					
2.					
3.					
4.					
5.					

Таблица 3 – Заполните CJM данные

После построения CJM ответьте на вопросы:

1. Где пользователи сталкиваются с трудностями?
2. Какие эмоции они испытывают на разных этапах?
3. Как можно улучшить UX?

Порядок сдачи лабораторной работы: Представить отчёт, содержащий заполненную CJM таблицу по варианту.

Вопросы для защиты работы

1. Что такое Customer Journey Map (CJM) и зачем она используется?
2. Какие ключевые элементы включает карта пользовательского пути?
3. Как CJM помогает улучшить пользовательский опыт и интерфейс продукта?
4. Чем CJM отличается от User Flow (потока пользователя)?
5. Как определить основные этапы взаимодействия пользователя с продуктом?
6. Какие типичные проблемы можно выявить с помощью CJM?
7. Что такое точки контакта (Touchpoints) и почему они важны в CJM?
8. Как учитывать эмоции и проблемы пользователя при создании CJM?

Практическая работа 4.

Изучение интерфейса Figma

Цель: Освоить интерфейс Figma, научиться работать с основными инструментами и настроить совместную работу.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Figma – это популярный облачный инструмент для проектирования пользовательских интерфейсов. Он позволяет создавать макеты, прототипы и работать над дизайном в команде в режиме реального времени.

Преимущества Figma:

- Доступ через браузер (не требует установки, но есть десктопная версия).
- Автоматическое сохранение изменений.
- Возможность совместной работы над проектами.
- Гибкость в использовании фреймов, фигур, текста и компонентов.
- Поддержка плагинов и интеграция с другими сервисами.

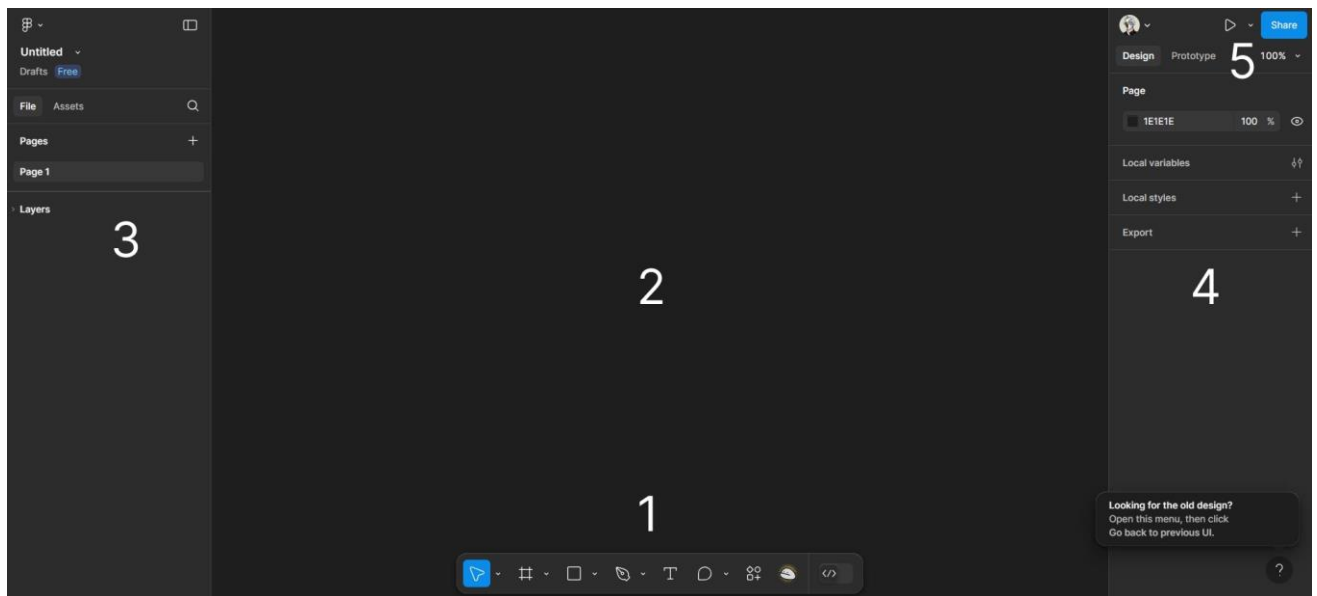
Для работы с Figma необходимо создать учетную запись:

1. Перейдите на figma.com и нажмите Sign up.
2. Введите адрес электронной почты и придумайте пароль.
3. Подтвердите регистрацию через почту и войдите в систему.
4. После входа вы попадете в главное меню, где можно создавать новые проекты и файлы.

При открытии нового проекта в Figma пользователи видят рабочее пространство, которое включает:

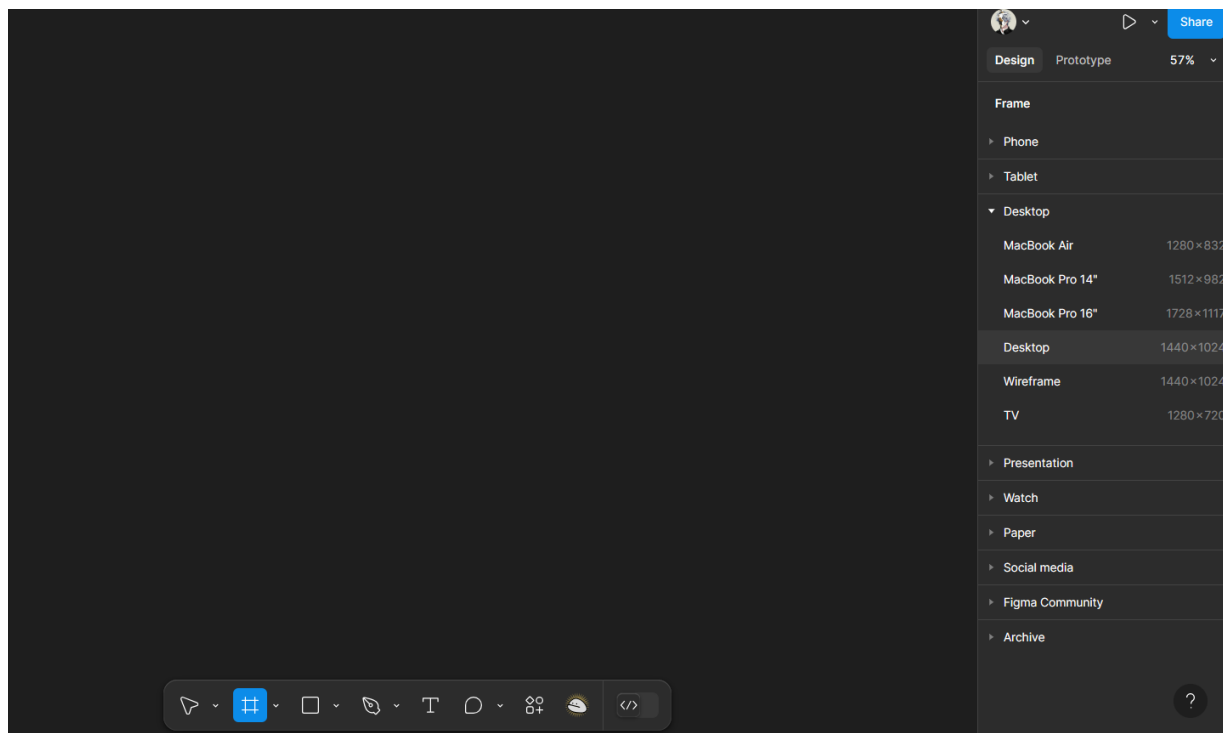
1. **Панель инструментов.** Содержит кнопки для работы с фигурами, текстом и слоями.
2. **Область работы (Canvas).** Главное пространство, где создаются макеты.

3. **Панель слоев (Layers Panel).** Показывает структуру всех объектов в документе.
4. **Панель свойств (Properties Panel).** Позволяет настраивать выбранные элементы (размер, цвет, эффекты).
5. **Кнопка Share.** Предоставляет доступ к файлу для совместной работы.



Фреймы (Frames) – это контейнеры, которые используются для организации интерфейсов и адаптации к разным экранам.

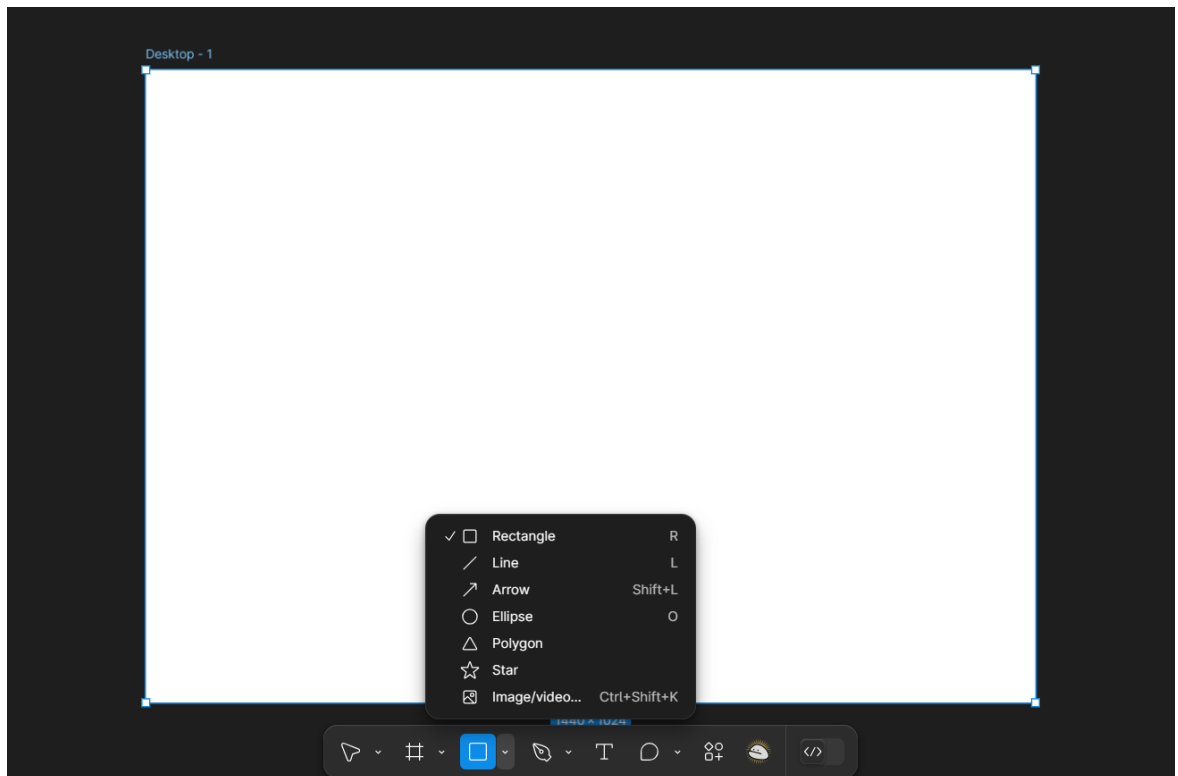
- Создаются с помощью инструмента Frame (F).
- Можно выбрать предустановленные размеры (например, iPhone 13, Desktop 1440px).



- Фреймы могут содержать фигуры, текст и изображения.

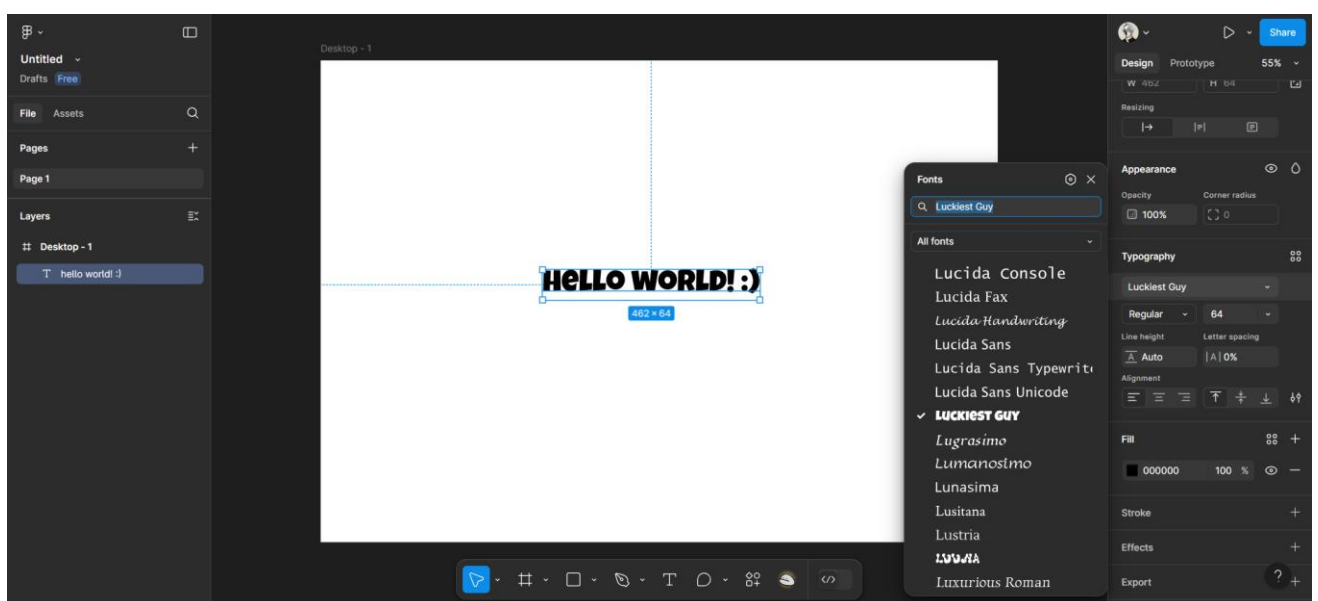
Фигуры (Shapes) – это базовые элементы, из которых строятся интерфейсы.

- Rectangle (R) – прямоугольник.
- Ellipse (O) – круг или овал.
- Line (L) – линия.
- Polygon (Shift + P) – многоугольник.



Текст (Text, T) – используется для заголовков, описаний, кнопок и других текстовых элементов.

- Размер и стиль текста можно настроить в панели свойств.
- Можно использовать готовые шрифты Google Fonts.



Одной из ключевых функций Figma является автоматическое сохранение, что предотвращает потерю данных при сбоях или ошибках.

Совместная работа в Figma:

- Нажмите кнопку Share в правом верхнем углу.
- Введите email или создайте ссылку с правами доступа (просмотр/редактирование).
- Несколько пользователей могут редактировать проект одновременно.
- Комментарии можно оставлять прямо в макете (инструмент Comment, C).

Методика выполнения лабораторной работы

Выполните задания прилагая скриншоты выполнения работы:

1. Зарегистрируйтесь в Figma.
2. Ознакомьтесь с главной страницей, рабочей областью и настройками.
3. Убедитесь, что включено автоматическое сохранение (оно включено по умолчанию в веб-версии).
4. Откройте Figma и создайте новый файл и переименуйте его в "ЛР 4 – Интерфейс Figma (ФИО)".
5. Выберите инструмент Frame (F) и создайте фрейм размером 375x812 px (размер экрана мобильного устройства). Добавьте еще один фрейм для второго экрана.
6. Добавьте заголовок с помощью инструмента Text (T).
7. Используя Rectangle (R), создайте кнопку с текстом.
8. Разместите несколько иконок (например, с помощью плагина Material Icons).
9. В правом верхнем углу нажмите Share и отправьте ссылку на проект однокласснику. Проверьте, как отображается курсор другого пользователя в реальном времени.
10. Убедитесь, что изменения сохраняются автоматически. Экспортируйте готовый макет в формате PNG или PDF.

Контрольные вопросы

1. Что такое Figma и в чем ее основные преимущества по сравнению с другими графическими редакторами?
2. Как зарегистрироваться в Figma и какие способы входа в систему доступны?
3. Какие основные элементы интерфейса Figma необходимо знать для работы?
4. Что такое фрейм (Frame) в Figma и чем он отличается от группы (Group)?
5. Какие инструменты используются для создания фигур и текста в Figma?
6. Как работает автоматическое сохранение в Figma, и можно ли отключить эту функцию?
7. Какие возможности Figma предоставляет для совместной работы?
8. Как предоставить доступ к проекту другому пользователю и какие существуют уровни прав?
9. Каким образом можно оставить комментарий к макету в Figma?
10. Как экспортировать готовый макет из Figma и в каких форматах это можно сделать?

Практическая работа № 5.

Основы Wireframing (каркасные макеты)

Цель работы: Освоить основные принципы создания каркасных макетов (Wireframes) в Figma для проектирования пользовательских интерфейсов.

Формируемые компетенции: ПК-8

Теоретическое обоснование

В UX/UI-дизайне есть три ключевых этапа проектирования интерфейсов:

Тип	Описание	Особенности
Wireframe (каркас)	Структурный набросок интерфейса без цветов и графики.	Фокус на размещении элементов, создании пользовательского пути.
Mockup (макет)	Дизайн страницы с цветами, шрифтами, изображениями.	Фокус на визуальном оформлении, но без интерактивности.
Prototype (прототип)	Интерактивная модель интерфейса, имитирующая поведение конечного продукта.	Фокус на тестировании взаимодействия с пользователем.

Таблица 1 – Ключевые этапы проектирования интерфейсов

Wireframe – это первый шаг в проектировании интерфейса, который помогает определить логику расположения элементов.

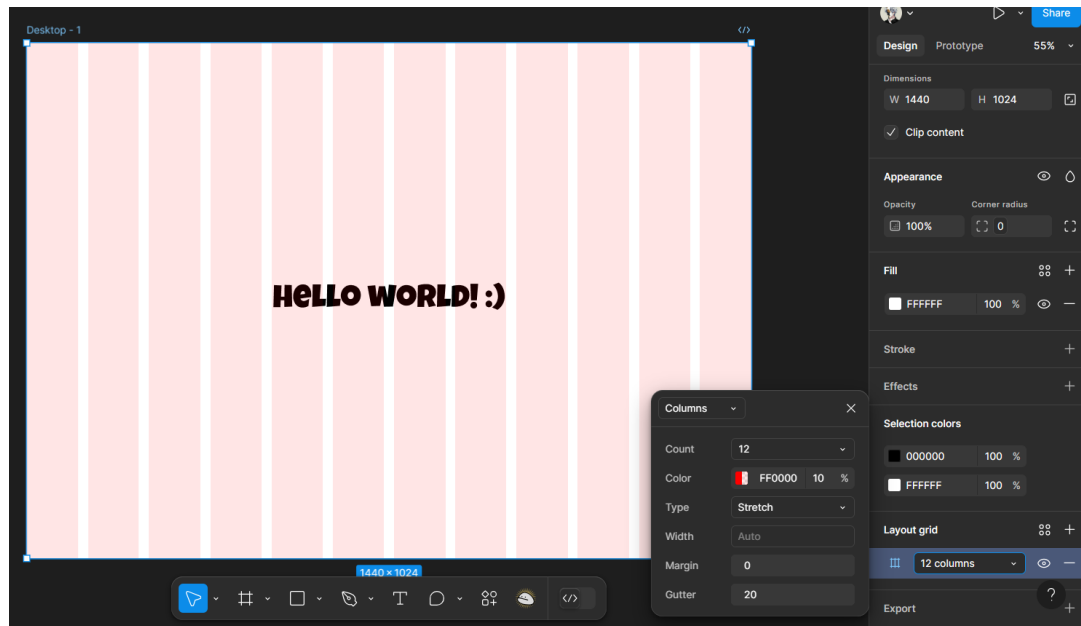
При создании каркасного макета важно избегать излишнего оформления, чтобы сосредоточиться на структуре страницы. В таких макетах используются:

- Только оттенки серого (чтобы не отвлекать внимание цветом).
- Простые геометрические фигуры (для обозначения кнопок, полей, блоков контента).
- Условные обозначения (например, серый прямоугольник вместо изображения).

Для удобного проектирования макетов используются инструменты:

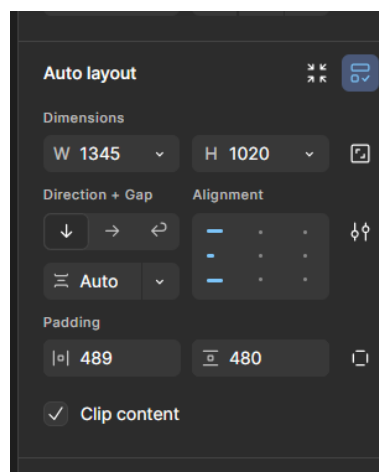
Гриды (Grids) – помогают выравнивать элементы, чтобы интерфейс был удобным и эстетически выверенным.

- Добавляются в панели свойств: Layout Grid → Add Grid.
- Чаще всего используется 12-колоночная сетка для веб-дизайна.



Auto-layout – позволяет автоматически распределять и масштабировать элементы.

- Позволяет создавать гибкие интерфейсы, которые легко адаптируются при изменении размеров.
- Применяется для кнопок, карточек, списков и других элементов.



Использование гридов и auto-layout значительно упрощает работу с макетами и делает их более удобными для адаптации.

Методика выполнения лабораторной работы

Задание: Создать черно-белый каркасный макет главной страницы веб-сайта, используя гриды и auto-layout в Figma.

1. Создайте новый проект в Figma и назовите его "ЛР 5 – Wireframe(ФИО)".
2. Выберите инструмент Frame (F). Создайте фрейм с размерами 1440x900 px (для веб-версии).
3. Выберите фрейм, откройте панель свойств. В разделе Layout Grid нажмите "+" и добавьте 12-колоночную сетку.
4. Разместите header (логотип, меню, поиск).
5. Добавьте блок основного контента (заголовок, текст, изображения-заглушки).
6. Разместите СТА-кнопку (например, "Купить", "Подробнее").
7. Добавьте footer (контакты, соцсети).
8. Группируйте элементы и включите Auto-layout (Shift + A). Настройте отступы, выравнивание и масштабирование элементов.
9. Экпортируйте макет в формате PNG и вставьте в отчёт.

Контрольные вопросы

1. В чем разница между Wireframe, Mockup и Prototype?
2. Почему при создании Wireframe используются только черно-белые цвета?
3. Какие элементы обязательны в каркасном макете?
4. Как добавить гриды (Grids) в Figma и зачем они нужны?
5. Что такое auto-layout и какие преимущества он дает?
6. Как с помощью auto-layout организовать отступы между элементами?
7. Какие инструменты Figma помогают ускорить создание Wireframe?
8. Как адаптировать Wireframe под разные разрешения экранов?
9. Почему Wireframe является важным этапом в UX-дизайне?
10. Как экспортировать готовый Wireframe из Figma?

Лабораторная работа № 6

Сетки и адаптивный дизайн

Цель: Научиться использовать 12-колоночную сетку в Figma, адаптировать макеты под разные разрешения экранов и применять auto-layout для гибкого расположения элементов.

Формируемые компетенции: ПК-8

Теоретическое обоснование

12-колоночная сетка – это самая распространённая система разметки в веб-дизайне. Она делит экран на 12 равных колонок и помогает организовать контент логично и структурировано.

Почему именно 12 колонок?

- **Делимость** – число 12 удобно делится на 2, 3, 4 и 6.
- **Гибкость** – можно использовать разные комбинации колонок (например, 4+4+4 или 6+6).
- **Адаптивность** – легко масштабировать дизайн для разных устройств.

Устройство	Количество колонок	Пример деления
Десктоп (1440px)	12	3+3+3+3 или 6+6
Планшет (768px)	8	4+4 или 2+2+2+2
Мобильный (375px)	4	2+2 или 1+1+1+1

Таблица 1 – Пример структуры 12-колоночной сетки

Адаптивный дизайн позволяет изменять макет в зависимости от устройства пользователя. Это достигается с помощью медиа-запросов (breakpoints) и гибких элементов.

Брейкпоинты (Breakpoints) – контрольные точки, в которых интерфейс изменяет свою структуру:

- Мобильный: 375px – 480px

- Планшет: 768px – 1024px
- Десктоп: 1280px и выше

Auto-layout – это инструмент в Figma, который автоматически подстраивает элементы при изменении размера контейнера.

Как использовать Auto-layout:

1. Выберите группу элементов и нажмите Shift + A.
2. Настройте отступы (padding) и интервалы (spacing) между элементами.
3. Установите тип выравнивания (горизонтальный / вертикальный).
4. Включите гибкое масштабирование (Hug content, Fill container).

Преимущества Auto-layout:

- Элементы автоматически подстраиваются под размер экрана.
- Упрощает редактирование макетов.
- Помогает создавать гибкие адаптивные дизайны.

Методика выполнения лабораторной работы

Задание: Создать адаптивный макет веб-страницы в трёх разрешениях (Desktop, Tablet, Mobile), используя 12-колоночную сетку и auto-layout в Figma.

1. Создайте новый проект в Figma и назовите его "ЛР 6 – Сетки и адаптивный дизайн(ФИО)".
2. Создайте три фрейма с размерами:
 - Десктоп (1440x900 px).
 - Планшет (768x1024 px).
 - Мобильный (375x812 px).
3. Добавьте 12-колоночную сетку. Откройте панель Layout Grid, нажмите "+" и выберите Columns (12 колонок). Настройте ширину колонок, отступы (Gutter) и поля (Margin).
4. Разместите основные элементы страницы:
 - Header (логотип, меню, кнопка входа).
 - Hero-секция (заголовок, описание, СТА-кнопка).
 - Контентные блоки (карточки товаров/услуг).
 - Footer (контакты, соцсети).
5. Используйте auto-layout. Выделите группы элементов и нажмите Shift + A. Настройте отступы и гибкость блоков.
6. Адаптируйте макет для всех разрешений, изменяя расположение элементов.
7. Экпортируйте макеты в формате PNG и вставьте в отчёт.

Контрольные вопросы

1. Что такое 12-колоночная сетка, и почему она популярна?
2. Какие преимущества даёт использование сеток в UI-дизайне?
3. Какие основные брейкпоинты используются в адаптивном дизайне?
4. Как правильно использовать 12-колоночную сетку в Figma?
5. Чем адаптивный дизайн отличается от фиксированного?
6. Что такое auto-layout, и как он помогает в создании адаптивных макетов?
7. Как добавить сетку в Figma?

8. Какие параметры можно настроить в Layout Grid?
9. Почему важно тестировать макет в разных разрешениях?
10. Какие элементы интерфейса сложнее всего адаптировать под разные устройства?

Лабораторная работа № 7

Работа с компонентами в Figma

Цель: Освоить работу с компонентами и вариантами (Variants) в Figma, научиться управлять состояниями UI-элементов и проверять автоматическое обновление инстансов при изменении главного компонента.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Компоненты – это элементы интерфейса, которые можно повторно использовать в дизайне. Их главное преимущество — единое управление всеми копиями (инстансами).

Основные термины:

Компонент (Component) – оригинальный объект, из которого создаются копии.

Инстанс (Instance) – копия компонента, которая наследует его свойства.

Главный компонент (Master Component) – основной элемент, изменения в котором применяются ко всем инстансам.

Варианты компонентов (Variants) позволяют объединить несколько состояний одного элемента в один компонент. Например, кнопка может иметь состояния:

- **Обычная (Default)** – стандартное состояние.
- **Наведение (Hover)** – когда курсор находится над кнопкой.
- **Нажатие (Pressed)** – когда кнопка нажата.

Преимущества вариантов:

- **Экономия времени** – все состояния находятся в одном месте.

- Удобство взаимодействия – можно быстро переключаться между вариантами.
- Простая настройка интерактивности – изменение состояний в режиме прототипирования.

Если изменить главный компонент, то все экземпляры автоматически обновятся. Это позволяет быстро вносить правки без необходимости редактировать каждый элемент вручную.

Как проверить авто-обновление?

1. Создать компонент кнопки.
2. Сделать несколько экземпляров.
3. Изменить главный компонент (например, цвет фона).
4. Проверить, что все экземпляры изменились автоматически.

Ход работы:

Задание: Создать кнопку с тремя состояниями (Default, Hover, Pressed) и проверить работу авто-обновления компонентов.

1. Создать кнопку. Создать фрейм (200x50px). Добавить текст "Купить". Установить цвет фона. Закруглить углы (8px).
2. Создать главный компонент. Выделить кнопку и нажать Ctrl + Alt + K (ПКМ → "Create Component").
3. Настроить варианты (Variants). Выбрать компонент и нажать "Add Variant". Добавить Hover-версию (изменить цвет на более светлый). Добавить Pressed-версию (уменьшить яркость цвета). В панели "Variants" задать названия Default, Hover, Pressed.
4. Добавить интерактивность. Перейти в "Prototype". Связать Default → Hover (триггер "While Hovering"). Связать Hover → Pressed (триггер "On Click").
5. Проверить авто-обновление. Создать несколько экземпляров кнопки. Изменить цвет главного компонента. Проверить, что изменения применились ко всем копиям.

6. Экспортировать проект и отправить архивом с отчётом.

Контрольные вопросы

1. Что такое компонент в Figma?
2. Как создать инстанс компонента?
3. Чем инстанс отличается от главного компонента?
4. Как создать варианты внутри одного компонента?
5. Какие состояния кнопки используются в UI-дизайне?
6. Как связать варианты в режиме прототипирования?
7. Как проверить авто-обновление инстансов?
8. Почему использование компонентов ускоряет работу дизайнера?
9. Можно ли менять отдельные свойства инстанса, не затрагивая главный компонент?
10. Как организовать работу с компонентами в больших проектах?

Лабораторная работа № 8

Дизайн форм и элементов ввода

Цель: Изучить принципы UX-дизайна форм, сосредоточив внимание на минимализме и удобстве для пользователя. Создать форму регистрации с интерактивными состояниями и анимацией ошибок ввода.

Формируемые компетенции: ПК-8

Теоретическое обоснование

UX-дизайн форм — это проектирование интерфейсов ввода данных таким образом, чтобы они были интуитивно понятными, удобными и максимально эффективными для пользователя. Хорошо спроектированная форма позволяет пользователю быстро и без ошибок ввести нужную информацию.

Основные принципы UX-дизайна форм:

1. Минимализм:

Формы должны быть лаконичными и не перегружать пользователя лишней информацией. Это помогает сосредоточить внимание на нужных полях, избегая путаницы. Чем проще форма, тем меньше вероятность ошибок.

2. Удобство и логичность:

Элементы формы должны быть расположены в логическом порядке, чтобы пользователь интуитивно понимал, что и в какой последовательности ему нужно заполнять. Например, сначала поле для имени, потом для email и далее.

3. Предсказуемость и доступность:

Пользователь должен понимать, какие данные от него ожидаются в каждом поле. Важно использовать подсказки и placeholder'ы, чтобы помочь пользователю заполнить форму корректно. Все поля должны быть легко доступными для людей с ограниченными возможностями.

4. Интерактивность:

Элементы ввода должны взаимодействовать с пользователем. Например, при наведении на кнопку она может изменять цвет, а при ошибке в поле должно

отображаться сообщение о проблеме. Это улучшает пользовательский опыт и помогает быстро исправить ошибки.

5. Обратная связь:

Система должна показывать пользователю статус его действий. Например, когда пользователь вводит правильный email, форма должна показать положительное подтверждение, а в случае ошибки — объяснение, что именно не так.

Состояния полей ввода: Каждое поле ввода может иметь несколько состояний, в зависимости от взаимодействия пользователя. Это важно для повышения удобства и понимания, что происходит:

- **Обычное состояние (Default):** Поле ввода готово к заполнению.
- **Фокусное состояние (Focus):** Когда пользователь кликает на поле, оно становится активным и выделяется.
- **Ошибочное состояние (Error):** Если пользователь вводит неверную информацию, например, неправильный email. Обычно это сопровождается красной рамкой и текстом ошибки.
- **Неактивное состояние (Disabled):** Когда поле не доступно для взаимодействия, например, при отсутствии флажка согласия.

Анимации ошибок ввода: Анимация может служить для того, чтобы пользователю было очевидно, что он сделал ошибку, и нужно внести исправления. Например, анимация может включать:

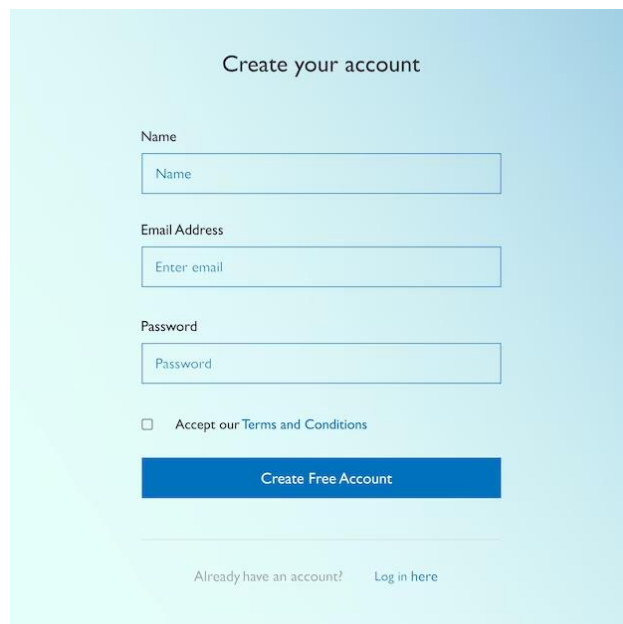
- **Изменение цвета:** Поле становится красным, когда введены неверные данные.
- **Появление подсказки:** Текстовое сообщение или иконка ошибки появляются рядом с полем, чтобы объяснить, что именно нужно исправить.

Использование анимации важно для визуальной обратной связи. Это позволяет пользователю сразу понять, что он сделал неправильно, и оперативно исправить ошибку.

Форма регистрации часто является первым взаимодействием пользователя с сервисом или веб-приложением, поэтому она должна быть максимально удобной и простой.

Содержание формы. Типичная форма регистрации включает такие поля:

- Имя
- Email
- Пароль
- Подтверждение пароля
- Согласие с условиями



The image shows a registration form titled "Create your account" on a light blue background. It contains the following elements: a "Name" label above a text input field with placeholder text "Name"; an "Email Address" label above a text input field with placeholder text "Enter email"; a "Password" label above a text input field with placeholder text "Password"; a checkbox labeled "Accept our Terms and Conditions"; a blue button labeled "Create Free Account"; and at the bottom, two links: "Already have an account?" and "Log in here".

Методика и порядок выполнения работы

Задание: Создать форму регистрации с различными полями и состояниями.

1. Создать новый проект в Figma и назвать его "ЛР 8 – Регистрация (ФИО)".
2. Создать форму регистрации с полями:
 - Поля для ввода имени, email, пароля и подтверждения пароля.
 - Чекбокс для согласия с условиями.
 - Кнопка отправки формы (Submit).
3. Установить цвет фона, рамки и текста для каждого поля ввода.
4. Поля для пароля должны быть скрытыми (с использованием символов «•»).

5. Поля email и пароля должны иметь подсказки (например, для email указать формат ввода: "user@example.com").
6. Экспортировать проект и прикрепить архивом с отчётом.

Контрольные вопросы

1. Что такое минимализм в дизайне форм и почему он важен?
2. Какие типы состояний можно настроить для полей ввода? Приведите примеры.
3. Как должна работать анимированная ошибка в формах ввода?
4. Почему важно использовать ошибочные состояния в интерфейсах?
5. Как работает система валидации в формах регистрации?
6. Как можно уменьшить количество ошибок пользователя при заполнении формы?
7. Какие особенности нужно учитывать при проектировании формы для мобильных устройств?
8. Как интерактивность улучшает UX формы?
9. Что такое обратная связь в UX-дизайне формы и как она помогает пользователю?
10. Как можно сделать форму более доступной для людей с ограниченными возможностями?

Лабораторная работа № 9

Цвет и типографика в UI

Цель: Освоить основы работы с цветом и типографикой в UI-дизайне, научиться подбирать цветовую палитру, выбирать подходящие шрифты и проверять контрастность элементов интерфейса.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Цвет — важнейший инструмент визуального восприятия. Он помогает направлять внимание пользователя, создавать эмоциональные ассоциации и улучшать читаемость интерфейса.

Цветовые палитры в UI. При создании интерфейса дизайнер подбирает основную палитру, состоящую из:

- **Основного цвета (Primary Color)** — главный акцент в дизайне (например, цвет кнопок, ссылок, иконок).
- **Дополнительных цветов (Secondary Colors)** — вспомогательные оттенки, поддерживающие основной цвет.
- **Фоновых цветов** — для заднего фона, карточек, элементов интерфейса.
- **Цветов текста** — контрастные и удобочитаемые оттенки для заголовков, основного текста, ссылок.
- **Цветов состояния** (успех, ошибка, предупреждение) — зеленый (успех), красный (ошибка), желтый (предупреждение).

Как выбрать цветовую палитру? Можно использовать сервисы подбора палитр:

- Coolors
- Adobe Color
- Material Palette

Придерживаться одной из цветовых схем:

- Монохромная (разные оттенки одного цвета).
- Аналоговая (цвета, расположенные рядом на цветовом круге).
- Комплементарная (контрастные цвета, расположенные напротив друг друга).

Учитывать доступность (контрастность должна соответствовать WCAG 2.1).

Правильно подобранные шрифты обеспечивают удобочитаемость интерфейса и улучшают пользовательский опыт.

Шрифты делятся на несколько категорий:

- Без засечек (sans-serif) — современные и читаемые (Roboto, Inter, Open Sans).
- С засечками (serif) — для традиционных и официальных сайтов (Times New Roman, Georgia).
- Декоративные — используются для уникального стиля (редко применяются в UI).

Как правильно настроить типографику? Определить шрифтовую иерархию:

- Заголовки (H1, H2, H3) — крупные и выделяющиеся.
- Основной текст — 14-18 px, с достаточным межстрочным интервалом.
- Мелкие подписи — 10-12 px, менее заметные.

По правилам графического дизайна нужно использовать не более 2-3 шрифтов в одном проекте.

Контраст между цветами помогает улучшить читаемость текста и удобство взаимодействия с интерфейсом.

Как проверить контраст? Используем сервис WebAIM Contrast Checker (WebAIM), который проверяет, соответствует ли контрастность цветов стандартам WCAG.

- Минимальный рекомендуемый контраст: 4.5:1 для текста.

- Для заголовков и крупных элементов: 3:1.

Методика и порядок выполнения работы

Задание: Создать UI-макет, используя правильные цветовые сочетания, шрифты и проверяя контрастность.

1. Создать новый проект в Figma и назвать его "ЛР 9 – UI-макет (ФИО)".
2. Подобрать цветовую палитру:
 - Выбрать основной цвет и вспомогательные цвета (используя Coolors или Adobe Color).
 - Определить цвет фона, текста и состояний (ошибка, успех).
 - Добавить палитру в Figma Styles.
3. Выбрать и настроить шрифты:
 - Определить основной шрифт (например, Roboto, Inter).
 - Создать стили заголовков, текста и подписей.
 - Настроить межстрочное расстояние и отступы.
4. Применить цвета и шрифты в макете:
 - Создать заголовок, текстовый блок и кнопку.
 - Оформить интерфейс с использованием цветовой схемы.
5. Сохранить и экспортировать макет, прикрепите его архивом вместе с отчётом.

Контрольные вопросы

1. Какие основные элементы входят в цветовую палитру UI?
2. Какие цветовые схемы применяются в дизайне интерфейсов?
3. Как правильно подбирать сочетания цветов для удобочитаемости?
4. Почему важно учитывать контрастность текста и фона?
5. Как проверить контрастность элементов интерфейса?
6. Какие сервисы помогают подбирать цветовые палитры?
7. Чем отличаются шрифты с засечками и без засечек?
8. Какие размеры текста считаются удобочитаемыми для UI?
9. Сколько шрифтов оптимально использовать в одном проекте?

10.Как правильно выстраивать шрифтовую иерархию?

Лабораторная работа № 10

Создание иконок в Figma

Цель: Освоить работу с векторными инструментами и логическими операциями (Boolean Operations) в Figma для создания пользовательских иконок.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Векторные изображения строятся на основе математических кривых и точек, что делает их:

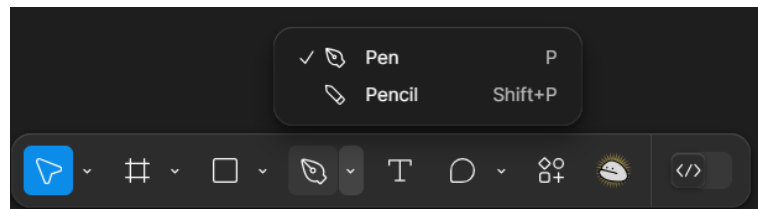
- **Гибкими** – можно менять размер без потери качества.
- **Легковесными** – файлы занимают мало места.
- **Редактируемыми** – можно изменять форму и детали на любом этапе.

Figma позволяет работать с векторной графикой с помощью Vector Tool и Boolean Operations.

Инструменты для создания иконок в Figma

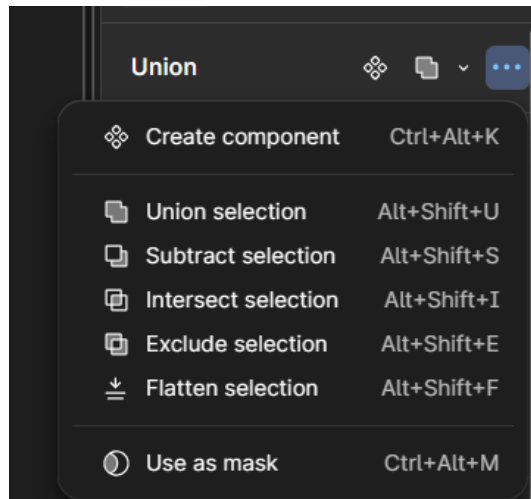
1. Vector Tool (Pen Tool, P)

- Позволяет рисовать фигуры вручную.
- Работает с узлами (точками) и кривыми.

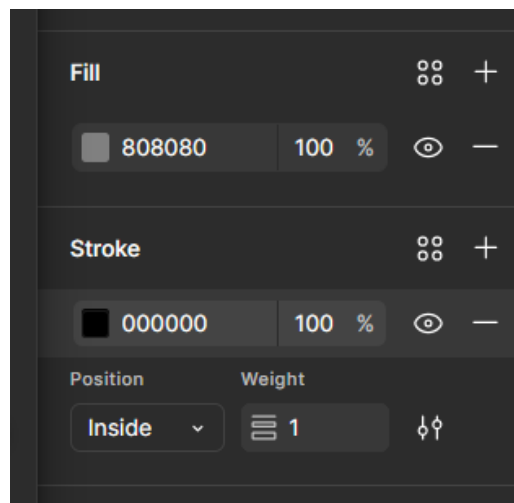


2. Boolean Operations – операции для объединения фигур:

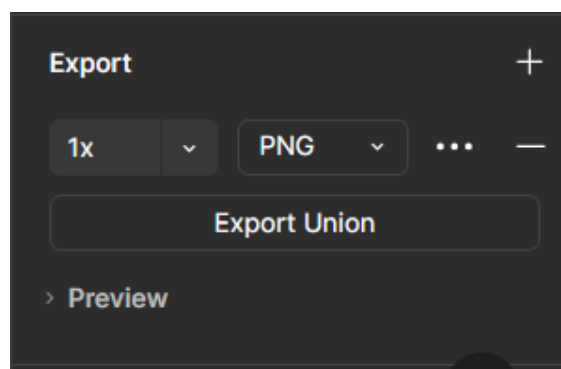
- Union (Объединение) – объединяет несколько фигур в одну.
- Subtract (Вычитание) – вырезает одну фигуру из другой.
- Intersect (Пересечение) – оставляет только пересекающуюся часть фигур.
- Exclude (Исключение) – убирает пересекающиеся части.



3. **Stroke & Fill** – настройка толщины линий и заливки фигур.



4. **Export Panel** – инструмент для сохранения иконок в разных форматах.



Создание иконок: основные принципы

- Простота – иконка должна быть легко узнаваемой.
- Единство стиля – все иконки должны быть одинаковой толщины и формы.
- Масштабируемость – иконки должны быть четкими в разных размерах.

- Оптимальные размеры – стандартные размеры иконок для UI: 16x16, 24x24, 32x32 px.

Figma поддерживает экспорт в SVG, PNG, PDF. Как экспортировать иконки в SVG?

1. Выбрать иконку.
2. Нажать Export в правой панели.
3. Выбрать формат SVG.
4. Сохранить файл.

SVG удобен, потому что:

- Сохраняет векторное качество.
- Позволяет изменять цвет и форму через код.

Имеет малый вес файла.

Методика и порядок выполнения работы

Задание: Создать три иконки в Figma: поиск, настройки, дом.

1. Создать новый фрейм в Figma (размер 24x24 px).
2. Сделать иконки с помощью Vector Tool и Boolean Operations:
 - Иконка поиска – нарисовать круг и добавить линию (лупу).
 - Иконка настроек – использовать Union и Subtract для создания шестеренки.
 - Иконка дома – создать треугольник (крыша) и квадрат (стены), объединить их.
3. Настроить Stroke & Fill:
 - Выбрать толщину линий (2 px).
 - Добавить заливку или оставить контур.
4. Экспортировать иконки в SVG:
 - Выбрать иконки.
 - Нажать Export → выбрать SVG.
5. Сохранить файлы.

Контрольные вопросы

1. В чем преимущества векторной графики перед растровой?
2. Какие инструменты в Figma помогают создавать иконки?
3. Какие Boolean Operations используются для объединения фигур?
4. Какой инструмент позволяет рисовать векторные фигуры вручную?
5. Как создать иконку поиска с помощью Vector Tool?
6. Почему важно соблюдать единый стиль иконок?
7. Какие стандартные размеры иконок используются в UI?
8. Как экспортировать иконки в формате SVG?
9. Почему SVG — лучший формат для иконок?
10. Как сделать иконки масштабируемыми?

Лабораторная работа № 11

Прототипирование и интерактивные связи

Цель: Освоить инструменты прототипирования в Figma, научиться настраивать интерактивные связи между экранами, добавлять ховер-эффекты и тестировать прототип в режиме презентации.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Прототипирование — это процесс создания интерактивной модели интерфейса. Оно помогает:

- Проверить логику навигации до разработки.
- Протестировать взаимодействие пользователей с UI.
- Согласовать структуру интерфейса с командой или заказчиком.

Виды прототипов

1. Низкокачественные (Low-Fidelity, Lo-Fi)

- Простые каркасные макеты (Wireframes).
- Без интерактивности, фокус на структуре.

2. Среднекачественные (Mid-Fidelity, Mid-Fi)

- Добавлены цвета и основные стили.
- Минимальная интерактивность.

3. Высококачественные (High-Fidelity, Hi-Fi)

- Полностью проработанный UI.
- Анимации, интерактивные элементы.

Figma позволяет создавать Mid-Fi и Hi-Fi прототипы.

Figma поддерживает Prototyping Mode, который позволяет связывать элементы:

- On Click (по нажатию) – переход на другой экран.
- While Hovering (при наведении) – изменение состояния элемента.
- After Delay (с задержкой) – автоматический переход.

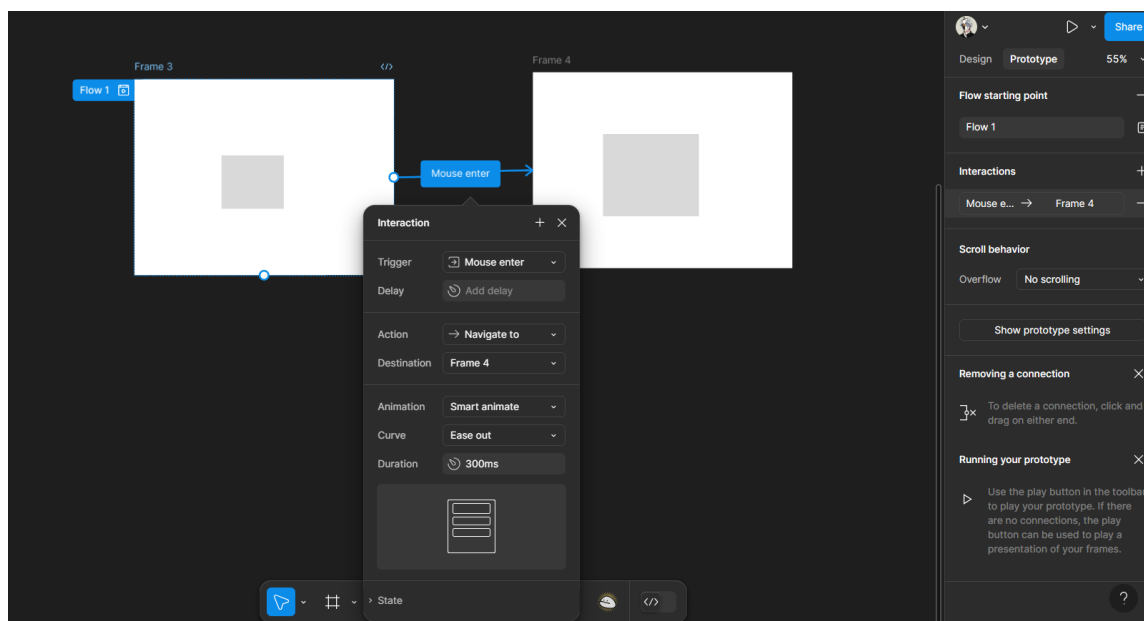
Для создания анимаций можно использовать:

- Instant (мгновенный переход)
- Smart Animate (плавные переходы между состояниями)
- Ease In/Out (замедление начала/конца анимации)

Ховер-эффекты делают интерфейс более отзывчивым и удобным. В Figma можно создать эффект наведения с помощью Variants и настроить переход While Hovering.

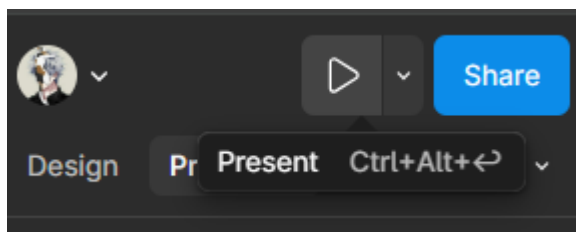
Примеры ховер-эффектов:

- Изменение цвета кнопки.
- Добавление тени.
- Изменение размера элемента.



В Figma есть режим Presentation Mode, где можно:

- Проверить навигацию и работу интерактивных элементов.
- Оценить ощущения пользователя от интерфейса.
- Поделиться прототипом с командой или заказчиком.



Методика и порядок выполнения работы

Задание: Создать интерактивный прототип мобильного приложения (из 3–4 экранов) с хOVER-эффектами и переходами.

1. Создать макет приложения (например, экран входа, список товаров, карточка товара).
2. Настроить переходы между экранами:
 - Выбрать элемент (например, кнопку).
 - Перейти во вкладку Prototype.
 - Привязать к нужному экрану с эффектом On Click.
3. Добавить хOVER-эффекты для кнопок:
 - Создать кнопку в Variants (обычное и наведенное состояние).
 - Настроить связь While Hovering.
4. Проверить прототип в Presentation Mode:
 - Нажать Play в правом верхнем углу.
 - Протестировать все переходы.

Контрольные вопросы

1. Что такое прототипирование и зачем оно нужно?
2. Какие существуют виды прототипов?
3. Как настроить переход между экранами в Figma?
4. Что такое Smart Animate и когда его использовать?
5. Как создать хOVER-эффект для кнопки?
6. Какие типы интерактивных связей поддерживает Figma?
7. Как протестировать интерактивный прототип в Figma?
8. Какие принципы помогают сделать прототип удобным для пользователя?
9. Какие анимации можно использовать при создании прототипа?
10. Как экспортировать и передавать прототип заказчику или разработчикам?

Лабораторная работа № 12

Анимация страниц и экранов

Цель: Освоить инструменты анимации в Figma, научиться создавать плавные переходы между экранами и добавлять интерактивные эффекты для улучшения пользовательского опыта.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Анимация делает взаимодействие с интерфейсом более интуитивным, помогая пользователю понимать:

- Логику переходов между экранами.
- Что происходит при его действиях (например, нажатии на кнопку).
- Какие элементы активны или изменяются.

Принципы хорошей анимации в UI:

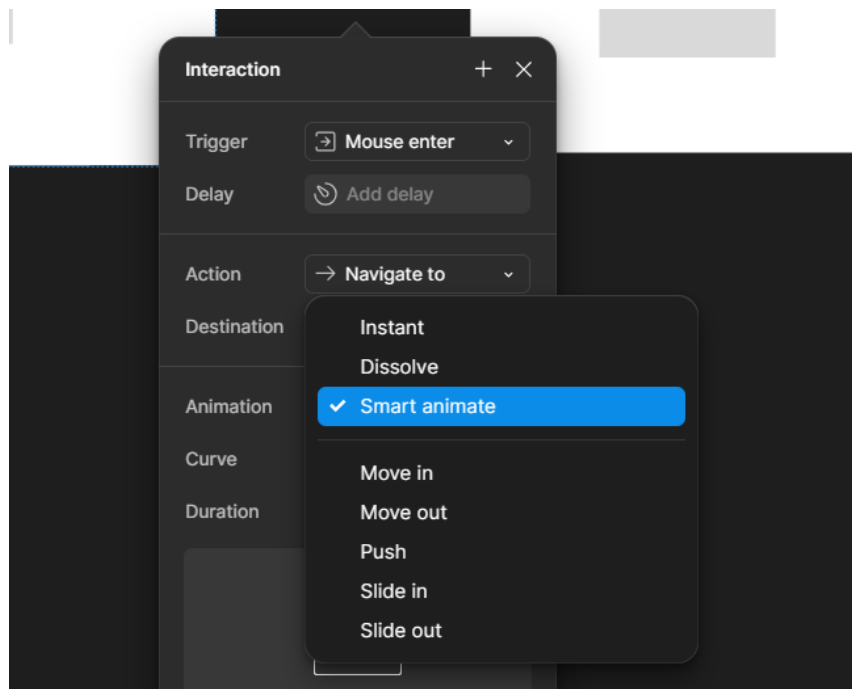
Естественность – движения должны быть плавными и соответствовать физике реального мира.

Скорость и отклик – слишком медленные анимации раздражают, слишком быстрые — теряют смысл.

Не перегруженность – чрезмерное использование анимации отвлекает и мешает.

Figma поддерживает несколько типов анимации:

1. **Instant (мгновенный переход)** – без плавного изменения, просто переключает экраны.
2. **Dissolve (растворение)** – плавное появление нового экрана.
3. **Move In / Move Out** – анимация выезда элемента с разных сторон экрана.
4. **Push** – новый экран сдвигает предыдущий, как в карусели.
5. **Smart Animate** – анализирует разницу между состояниями и создает плавную анимацию.



Основные анимационные эффекты

Анимация появления экрана

- Используется для первого экрана приложения.
- Можно настроить Dissolve, Move In или Smart Animate.

Перелистывание страниц

- Актуально для приложений с карточками или слайдерами.
- Применяется Push или Move In справа/слева.

Свайп-анимация

- Используется в мобильных приложениях (например, пролистывание товаров в интернет-магазине).

Реализуется через Smart Animate с ручным управлением движением.

Методика и порядок выполнения работы

Задание: Создать анимированный прототип мобильного приложения (3–5 экранов) с плавными переходами.

1. Создать макеты экранов приложения (например, экран входа, список товаров, карточка товара).

2. Настроить анимацию появления экрана:
 - Выбрать первый экран.
 - В режиме Prototype добавить анимацию Dissolve или Move In.
3. Добавить эффект перелистывания страниц:
 - Связать экраны через Prototype.
 - Настроить переход Push Left/Right для переключения.
4. Настроить свайп-анимацию:
 - Сделать карточки товаров.
 - Связать их через Smart Animate с движением влево/вправо.
5. Проверить анимацию в режиме презентации (кнопка Play).
6. Оформить презентацию проекта (добавить описание анимаций на отдельном фрейме).

Контрольные вопросы

1. Почему важна анимация в UI/UX-дизайне?
2. Какие принципы качественной анимации существуют?
3. Какие виды анимации поддерживает Figma?
4. Как создать эффект плавного появления экрана?
5. Чем отличается Push от Move In?
6. Как реализовать анимацию перелистывания страниц?
7. Как сделать свайп-анимацию в мобильном приложении?
8. В чем преимущество Smart Animate?
9. Как тестировать анимации в Figma?
10. Какие ошибки могут ухудшить восприятие анимации в UI?

Лабораторная работа № 13

Работа с эффектами (тени, размытие, градиенты)

Цель работы и содержание: Изучить и освоить работу с визуальными эффектами в Figma (тени, размытие, градиенты) для улучшения восприятия интерфейса и повышения его эстетической привлекательности.

Формируемые компетенции: ПК-8

Теоретическое обоснование

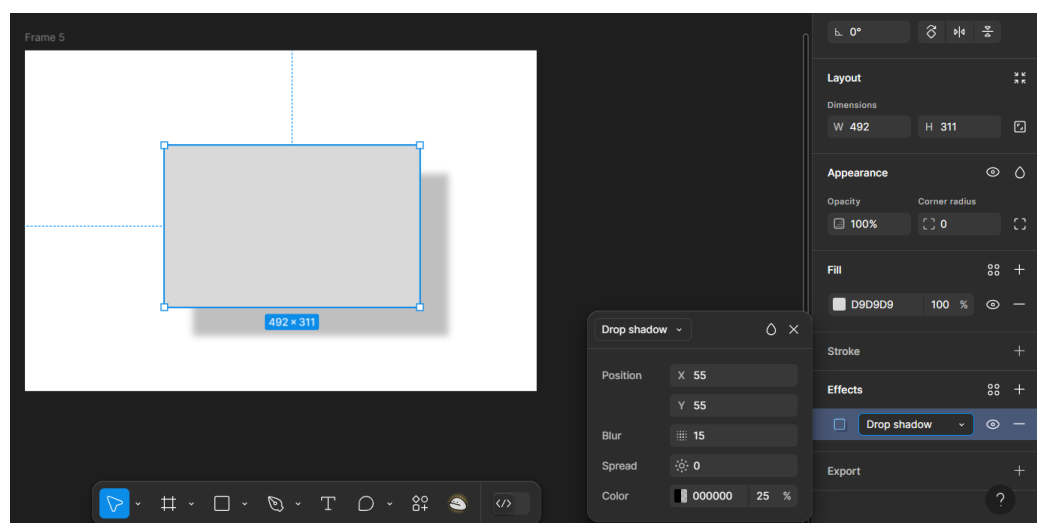
Зачем нужны эффекты в UI-дизайне? Визуальные эффекты помогают:

- Улучшить восприятие элементов интерфейса.
- Создать объем и глубину.
- Направить внимание пользователя.
- Сделать интерфейс более современным и приятным.

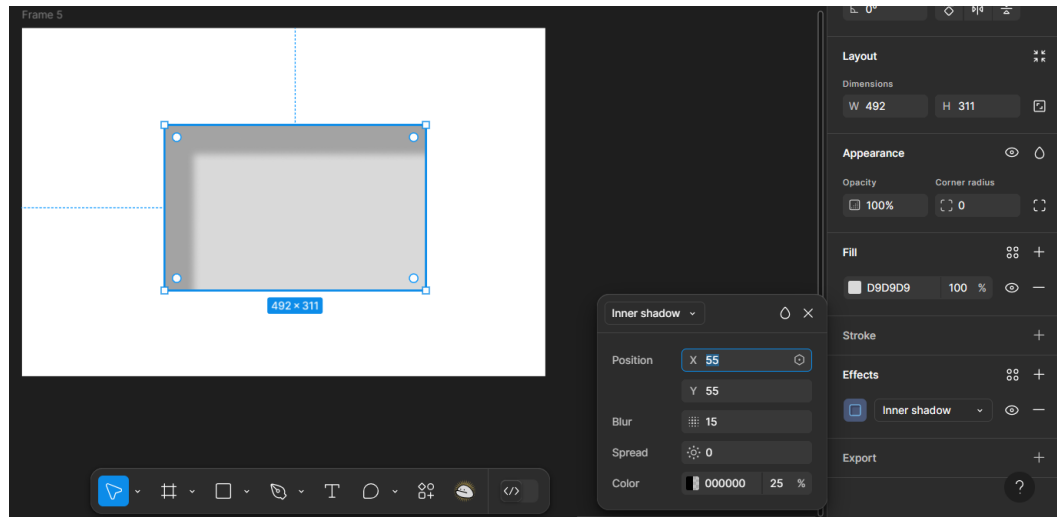
Тень (Shadow) — один из ключевых инструментов, позволяющий добавить объем и выделить элементы.

Виды теней:

- Внешние (Drop Shadow) — создают ощущение приподнятого элемента.



- Внутренние (Inner Shadow) — используются для эффектов углубления.



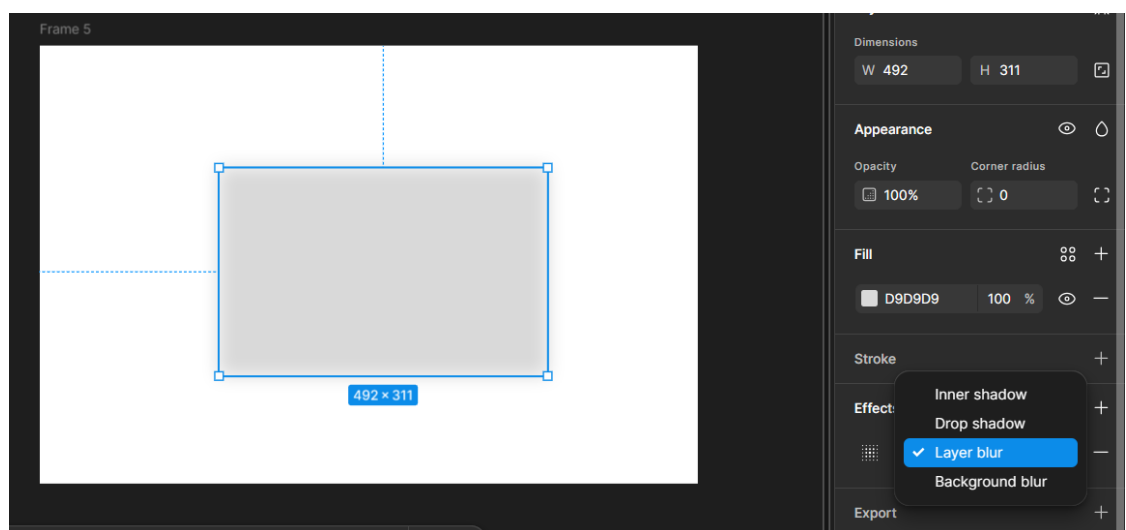
Хорошая тень:

- Размытие (Blur) — 4–12px для мягкости.
- Смещение (Offset) — небольшое, чтобы тень не выглядела «висящей».
- Прозрачность (Opacity) — 10–30% для естественного вида.

Размытие используется для создания эффектов глубины и фокусировки. Виды размытия:

Background Blur — размывает фон, но оставляет четкие контуры элемента.

Layer Blur — размывает весь объект.

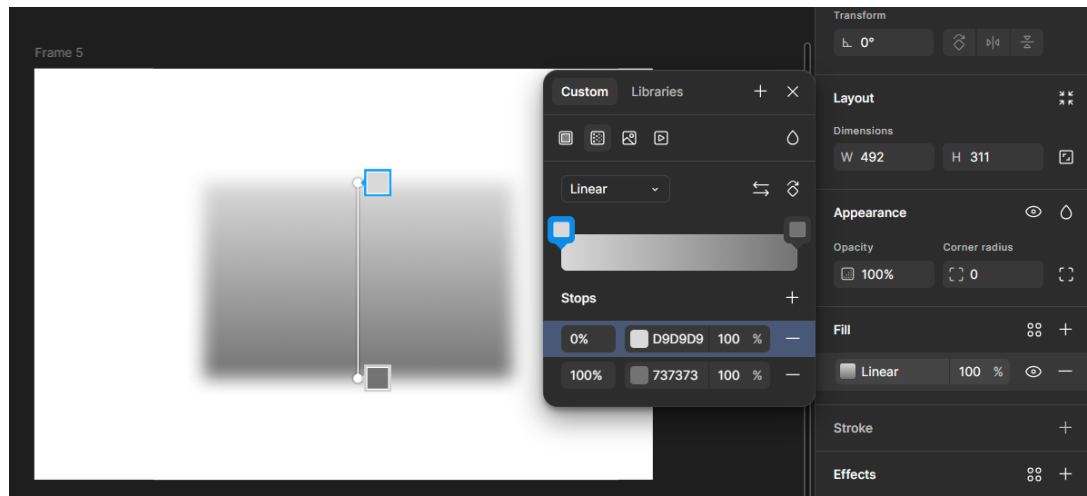


Применение размытия:

- Размытый фон в модальных окнах (эффект «стекла»).
- Размытие изображений для создания акцента на переднем плане.

Градиент — плавный переход между цветами. Основные типы:

Линейный (Linear Gradient) – градиент в одном направлении (горизонтально, вертикально, под углом).



Радиальный (Radial Gradient) – градиент, расходящийся от центра.

Angular Gradient – градиент по кругу.

Как использовать градиенты?

- Для фонов страниц.
- В кнопках и карточках.
- В декоративных элементах.

Советы по градиентам:

- Использовать 2–3 цвета для чистоты дизайна.
- Держать контрастность на среднем уровне.
- Подбирать цвета из одной палитры.

Методика и порядок выполнения работы

Задание: Создать карточку товара с тенями, размытием и градиентным фоном.

1. Создать кнопку с мягкой тенью:

- Выбрать Rectangle.
- Добавить Drop Shadow с параметрами:
 - Blur: 10px
 - Offset: 2px

- Opacity: 20%
 - Настроить разные состояния (обычное, наведение, нажатие).
2. Настроить размытый фон для модального окна:
 - Создать прямоугольник и применить Background Blur (20–30px).
 - Добавить текст и кнопку закрытия.
 3. Создать градиентный фон:
 - Выбрать фон страницы.
 - Применить Linear Gradient (например, от светлого голубого к темно-синему).
 4. Проверить в режиме просмотра (Prototype).
 5. Сохранить и экспортировать проект.

Контрольные вопросы

1. Какие бывают виды теней и в чем их отличие?
2. Как настроить мягкую тень в Figma?
3. Для чего используется размытие в UI-дизайне?
4. Чем Background Blur отличается от Layer Blur?
5. Как сделать эффект размытого модального окна?
6. Какие типы градиентов существуют?
7. Где уместно использовать градиенты в интерфейсе?
8. Как выбрать правильные цвета для градиента?
9. Какие ошибки могут ухудшить восприятие теней?
10. Почему важно не злоупотреблять эффектами в UI?

Лабораторная работа № 14

Дизайн-системы и UI-киты

Цель работы и содержание: Освоить принципы создания дизайн-систем и UI-китов для эффективной и согласованной работы над интерфейсами.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Дизайн-система — это единый набор правил, компонентов и стандартов, которые обеспечивают консистентность и удобство разработки интерфейсов.

Элементы дизайн-системы:

- Компоненты (кнопки, поля ввода, карточки, модальные окна).
- Стили (цвета, шрифты, отступы).
- Сеточные структуры (гриды и отступы).
- Гайдлайны (правила использования компонентов).

Зачем нужна дизайн-система?

- Ускоряет процесс разработки.
- Делает интерфейсы визуально согласованными.
- Облегчает взаимодействие дизайнеров и разработчиков.

UI-кит — это набор компонентов (кнопок, иконок, полей ввода и др.), оформленных в едином стиле, который можно повторно использовать в разных проектах.

Компоненты UI-кита:

- Кнопки (разные состояния: обычная, наведение, нажатие, неактивная).
- Формы (поля ввода, чекбоксы, переключатели).
- Шрифты (заголовки, подзаголовки, основной текст).
- Цветовые палитры (основные и вспомогательные цвета).

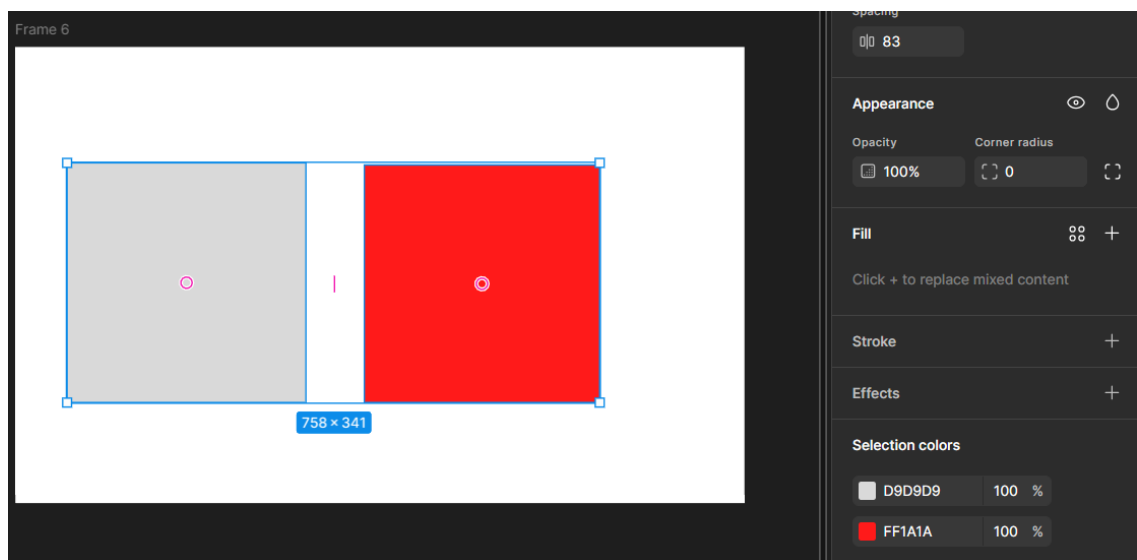
UI-КИТ	ДИЗАЙН-СИСТЕМА
Набор готовых компонентов	Структурированная система правил
Чаще используется для отдельных проектов	Применяется для всей экосистемы продукта
Может быть изменен под каждый проект	Следует строгим гайдлайнам

Таблица 1 – Разница между UI-китом и дизайн-системой

Figma Styles — инструмент для управления стилями в проекте.

Что можно настроить через Styles?

- Цвета (основной, второстепенный, фоновый, текстовый).



- Шрифты (размеры заголовков, межстрочный интервал, стили текста).



- Эффекты (тени, градиенты, размытие).

Преимущества использования Styles:

- Изменения автоматически применяются ко всем элементам.
- Упрощает работу команды (все работают по единым стандартам).
- Экономит время при масштабировании проекта.

Методика и порядок выполнения работы

Задание: Создать UI-кит в Figma, который включает основные элементы интерфейса, стили и компоненты, готовые для передачи разработчикам.

1. Создание кнопок.

Разработать кнопку с разными состояниями. Каждое состояние должно быть оформлено в едином стиле.

Состояния кнопок:

- Default (обычное) – стандартный вид кнопки.
- Hover (наведение) – кнопка слегка изменяет цвет/тень.
- Active (нажатие) – кнопка становится темнее или меняет стиль.
- Disabled (неактивное) – кнопка становится полупрозрачной или серой.

Нужно использовать авто-layout для автоматической подгонки размеров. Применить компоненты и варианты (Variants) для быстрого переключения состояний. Задать общие стили (цвета, шрифты) через Figma Styles.

Результат: Готовый компонент «Кнопка» с четырьмя состояниями.

2. Разработка полей ввода.

Создать стильные поля ввода для формы регистрации или авторизации.

Состояния полей ввода:

- Default (обычное) – стандартное поле с подсказкой (Placeholder).
- Focused (активное) – поле с выделенной границей.
- Filled (заполненное) – отображается введенный текст.
- Error (ошибка) – красная рамка, сообщение об ошибке.

Нужно использовать авто-layout для правильного размещения элементов. Добавить иконки (например, «глаз» для показа пароля). Настроить варианты (Variants) для быстрого переключения между состояниями.

Результат: Поле ввода с четырьмя состояниями, готовое к использованию в UI-ките.

3. Настройка текстовых и цветовых стилей (Figma Styles)

Нужно создать набор текстовых и цветовых стилей, которые будут использоваться во всем UI-ките.

Текстовые стили:

- Заголовки (H1, H2, H3).
- Основной текст.
- Подписи (Label, Placeholder, Error).

Цветовые стили:

- Основные цвета (Primary, Secondary).
- Фоновые цвета (Light, Dark).
- Цвета текста.

В разделе Styles → Text Styles задать шрифты, межстрочные интервалы и размеры. В разделе Styles → Color Styles определить основные цвета.

Результат: Готовые текстовые и цветовые стили для удобного применения.

4. Организация UI-кита и передача разработчикам.

Нужно подготовить UI-кит для командной работы и передачи разработчикам.

Собрать все элементы (кнопки, поля ввода, стили) на одной странице. Разделить элементы по категориям: Buttons, Inputs, Typography, Colors. Добавить подписи и комментарии, объясняющие использование компонентов.

Проверить работу компонентов и стилей (например, при изменении основного цвета все кнопки должны обновляться автоматически).

Результат: Готовый UI-кит, который можно использовать в реальных проектах.

Контрольные вопросы

1. Что такое дизайн-система и зачем она нужна?
2. Чем UI-кит отличается от дизайн-системы?
3. Какие элементы входят в UI-кит?
4. Какие преимущества дает использование Figma Styles?
5. Как правильно организовать цвета и шрифты в Figma Styles?
6. Какие состояния должны быть у кнопок?
7. Как передать UI-кит разработчику?
8. Как сделать UI-кит масштабируемым?
9. Почему важно поддерживать единые стили в проекте?
10. Какие ошибки часто встречаются при создании UI-китов?

Лабораторная работа № 15

Мобильный дизайн в Figma

Цель работы и содержание: Освоить создание мобильного интерфейса в Figma. Настроить адаптивность UI-дизайна. Проверить удобство взаимодействия с элементами на небольшом экране.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Дизайн мобильных приложений отличается от веб-дизайна несколькими ключевыми моментами:

Размер экрана – мобильные устройства имеют ограниченное пространство, поэтому важно учитывать компактность и удобство элементов.

Касание вместо кликов – пользователи взаимодействуют с интерфейсом пальцами, что требует увеличенных зон нажатия (минимум 44x44 px).

Глубина навигации – мобильные приложения часто используют bottom navigation (нижнюю навигацию) или hamburger-меню.

Адаптивность – дизайн должен корректно отображаться на разных моделях смартфонов с различными разрешениями.

Скорость работы – важно минимизировать количество элементов и эффектов, замедляющих загрузку.

При создании мобильного дизайна следует учитывать рекомендации ведущих платформ:

- Material Design (Google) – гайдлайн для Android-приложений.
- Human Interface Guidelines (Apple) – стандарты для iOS-приложений.

Каждая из этих систем содержит рекомендации по размерам кнопок, цветам, шрифтам и анимациям.

Основные компоненты мобильного интерфейса

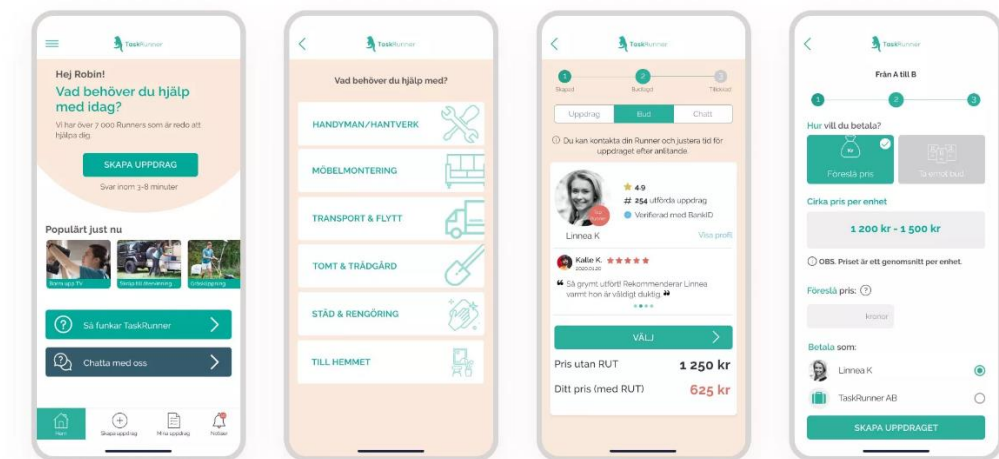
App Bar (верхняя панель) – отображает заголовок экрана и основные действия.

Navigation Bar (нижняя панель навигации) – содержит ключевые разделы приложения.

Кнопки (Primary & Secondary Buttons) – основные интерактивные элементы.

Формы и поля ввода – используются для регистрации, поиска и ввода данных.

Карточки (Cards) – группируют информацию в удобном виде.



Методика и порядок выполнения работы

Задание: Разработать главный экран мобильного приложения, настроить его адаптивность и проверить удобство взаимодействия с элементами.

1. Создайте новый проект в Figma и выберите Frame → iPhone 14 Pro (или другой размер экрана).
2. Включите layout grid (4 или 8 колонок) для удобного выравнивания элементов.
3. Определите основной цвет и шрифты (используйте Figma Styles).
4. Разработка главного экрана
 - Добавьте App Bar (верхнюю панель) с названием приложения.
 - Разместите ключевые элементы (например, поле поиска, список категорий, карточки контента).
 - Добавьте кнопки действий (например, «Купить», «Добавить в корзину»).
5. Настройка адаптивности UI
 - Используйте Auto Layout для гибкого масштабирования компонентов.
 - Создайте три варианта дизайна: iPhone 14 Pro, iPhone SE, Android (Pixel 7).

- Проверьте, как элементы перестраиваются при изменении размеров экрана.
6. Проверка удобства взаимодействия
- Используйте Figma Prototype Mode, чтобы настроить кликабельные зоны.
 - Проверьте, удобно ли нажимать на кнопки (минимальный размер 44x44 px).
 - Подготовьте несколько сценариев взаимодействия (например, клик по карточке открывает детали товара).

Результат: Готовый главный экран мобильного приложения, который адаптируется под разные устройства и удобен для пользователя.

Контрольные вопросы

1. Что такое мобильный дизайн, и какие особенности он имеет по сравнению с веб-дизайном?
2. Почему важно учитывать размер экрана при проектировании мобильных интерфейсов?
3. Какие рекомендации по мобильному дизайну предоставляет Material Design для Android?
4. Что такое «адаптивный дизайн» и как его можно реализовать в Figma?
5. Какие минимальные размеры интерактивных элементов (кнопок, полей ввода) рекомендуется использовать для мобильных устройств?
6. Что такое Auto Layout, и как он помогает в создании адаптивных интерфейсов?
7. Какие основные элементы мобильного интерфейса должны быть на главном экране приложения?
8. Каковы ключевые различия в дизайне для iOS и Android?
9. Какие параметры нужно учитывать при проектировании кнопок и полей ввода для мобильного интерфейса?
10. Как можно проверить удобство взаимодействия с элементами мобильного интерфейса в Figma?

Лабораторная работа № 16

Тестирование UI и исправление ошибок

Цель работы и содержание: Освоить создание мобильного интерфейса в Figma. Настроить адаптивность UI-дизайна. Проверить удобство взаимодействия с элементами на небольшом экране.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Тестирование интерфейса помогает выявить проблемные зоны в дизайне, которые могут негативно сказаться на опыте пользователей (UX). Даже самый красивый и продуманный интерфейс может быть неудобным и сложным для восприятия, если не учесть важные аспекты доступности, контрастности и удобочитаемости.

Одним из ключевых аспектов UI-дизайна является контрастность между текстом и фоном, а также доступность интерфейса для пользователей с особыми потребностями (например, слабовидящих).

Контрастность:

- Для того чтобы текст был легко читаемым, контраст между цветом текста и фоном должен быть достаточным.
- Рекомендуемый показатель контрастности для текста на фоне: не менее 4.5:1 для нормального текста и не менее 3:1 для крупного текста.

Доступность:

- Интерфейс должен быть доступен для всех пользователей, включая людей с ограниченными возможностями. Это включает использование доступных шрифтов, четкие кнопки, поддержку экранных читалок и т. д.

Для проверки контрастности можно использовать инструменты, такие как **WebAIM Contrast Checker**.

Удобочитаемость текста зависит от нескольких факторов:

Шрифт: Четкость шрифта, его размер и межстрочный интервал.

Цветовая схема: Высокий контраст между текстом и фоном, избегание чрезмерно ярких или раздражающих цветов.

Размер текста: Текст должен быть достаточного размера, чтобы его было легко читать на мобильных устройствах и экранах с низким разрешением.

После проведения тестирования и получения отзывов от пользователей, необходимо выявить ошибки и устранить их:

Проблемы с навигацией: Например, если кнопки слишком мелкие или неудобные для нажатия.

Некорректный контраст: Если текст не читается на фоне, нужно изменить цвета.

Ошибки в пользовательских сценариях: Например, поля ввода, которые не соответствуют реальным потребностям пользователей.

Важно провести тестирование с реальными пользователями, чтобы понять, как они взаимодействуют с интерфейсом. Для этого можно использовать методы:

Тестирование с пользователем (User Testing): Пользователь выполняет заранее подготовленные задачи, а вы наблюдаете за его действиями.

Интервью и опросы: Сбор обратной связи от пользователей по поводу удобства использования.

Тесты А/В: Проведение экспериментов с двумя вариантами дизайна, чтобы понять, какой из них работает лучше.

Методика и порядок выполнения работы

Задание 1: Проверка контрастности текста и фона с использованием WebAIM Contrast Checker.

1. Откройте ваш макет в Figma.
2. Выберите текстовые элементы, такие как заголовки, абзацы и кнопки.
3. Используя WebAIM Contrast Checker (инструмент доступный онлайн), введите значения цветов текста и фона. Оцените контрастность каждого текстового элемента.

4. Если контрастность для нормального текста (обычный размер шрифта) меньше 4.5:1, а для крупного текста — меньше 3:1, необходимо выбрать другие цвета для текста или фона, чтобы улучшить читаемость.
5. Запишите, какие элементы требуют корректировки по контрастности и внесите изменения в макет.

Задание 2: Оцените текстовые элементы по следующим критериям:

- Шрифт: Он должен быть легко читаемым. Избегайте слишком сложных или декоративных шрифтов.
- Размер шрифта: Для текста на экранах рекомендуется использовать размер шрифта не менее 14px для обычного текста и 18px для заголовков.
- Межстрочный интервал (leading): Между строками должен быть достаточный интервал (не менее 1.5x высоты шрифта), чтобы текст не выглядел слишком плотным.
- Цвет и контраст: Текст должен хорошо контрастировать с фоном, чтобы быть видимым на разных устройствах (мобильных и настольных экранах).

Задание 3: Исправление ошибок в дизайне

Если контрастность не соответствует требованиям (менее 4.5:1 для обычного текста и менее 3:1 для крупного), внесите следующие изменения:

1. Поменяйте цвет текста или фона. Выберите более контрастные цвета.
2. Используйте цветовые схемы, соответствующие стандартам доступности.

Если текст слишком мелкий, увеличьте размер шрифта. Если шрифт слишком декоративный или трудночитаемый, выберите более простые и распространенные шрифты, такие как Roboto, Arial, Helvetica.

Убедитесь, что между строками текста достаточный интервал, чтобы облегчить восприятие информации.

Если ваши элементы интерфейса не имеют текстовых меток или подсказок, добавьте их. Например, добавьте вспомогательные тексты или подсказки для кнопок и форм.

Убедитесь, что все изображения имеют альтернативные текстовые описания, чтобы они могли быть прочитаны пользователями с нарушениями зрения.

Добавьте поддержку навигации с клавиатуры для пользователей, которые не могут использовать мышь.

Задание 4: Тестирование с реальными пользователями

Сформулируйте несколько задач для пользователей, которые должны быть выполнены в вашем интерфейсе. Например:

- Зарегистрироваться в системе.
- Заполнить форму для отправки заявки.
- Найти информацию о продукте на сайте.
- Перейти к настройкам профиля.

Подготовьте краткие инструкции для участников тестирования, объясняя, что они должны сделать. Убедитесь, что они не испытывают давления и могут выражать свои мнения открыто.

Пригласите 3-5 реальных пользователей (например, одноклассников или знакомых), чтобы протестировать интерфейс.

Пусть пользователи выполняют предложенные задачи. Запишите их действия, отмечайте, где они испытывают трудности, где задерживаются или не понимают, что делать.

Обратите внимание на поведение пользователей, а не только на их слова. Иногда то, что они говорят, не всегда соответствует тому, что они на самом деле делают.

Внимательно записывайте все проблемы, с которыми сталкиваются пользователи. Пример таких проблем:

- Неясные или неудобные кнопки.

- Неинтуитивная навигация.
- Проблемы с заполнением форм (например, отсутствие подсказок при ошибках).

Проанализируйте собранные данные. Какие проблемы были наиболее частыми? Что вызвало трудности у большинства пользователей?

После анализа внесите необходимые изменения в интерфейс. Это может быть изменение цвета кнопок, улучшение текста подсказок или изменение расположения элементов.

Отчет о тестировании должен включать:

1. Перечень выявленных проблем, таких как недостаточный контраст, трудности с навигацией или непонятные кнопки.
2. Предложения по исправлениям для каждого из найденных проблем.
3. Результаты тестирования с пользователями: что пользователи думали о вашем интерфейсе, какие трудности они испытывали, что им понравилось.
4. Если проводилось A/B тестирование (например, два разных варианта кнопки), опишите, какой из вариантов был более успешным и почему.

Приложите изображения или ссылки на улучшенные версии макетов в Figma, чтобы показать, какие изменения были внесены.

Контрольные вопросы

1. Что такое контрастность в дизайне, и почему она важна для доступности интерфейса?
2. Какие инструменты можно использовать для проверки контрастности текста и фона в UI?
3. Какие ошибки могут возникнуть при несоответствии контрастности текста и фона стандартам доступности?
4. Как оценить удобочитаемость текста в интерфейсе? Какие факторы следует учитывать при выборе шрифта?
5. Что такое доступность в дизайне, и какие основные принципы необходимо учитывать при создании доступного интерфейса?

6. Какую роль играют альтернативные текстовые описания для изображений в контексте доступности?
7. Как проводятся юзабилити-тестирования на реальных пользователях, и какие данные следует собирать в процессе?
8. Какие методы и техники можно использовать для выявления проблем с навигацией в интерфейсе во время тестирования?
9. Как проводить тестирование доступности для пользователей с ограниченными возможностями? Приведите примеры.
10. Какие действия необходимо предпринять после выявления ошибок в интерфейсе на основе пользовательских тестов? Как исправления могут повлиять на общий пользовательский опыт?

Лабораторная работа № 17-18

Финальный проект

Цель работы и содержание: Разработать полноценный интерфейс для мобильного или веб-приложения, создать несколько экранов с интерактивными переходами, подготовить проект для передачи разработчику и представить свой проект аудитории.

Формируемые компетенции: ПК-8

Теоретическое обоснование

Процесс разработки экрана интерфейса.

На этом этапе важно понимать, что каждый экран должен не только выглядеть красиво, но и обеспечивать удобство пользователя.

Экран должен быть логично структурирован, иметь понятные кнопки и элементы управления, а также предлагать удобный путь для выполнения задач пользователя.

Используйте гриды и авто-layout для упорядочивания элементов на экране и обеспечения адаптивности интерфейса.

Интерфейс должен быть интерактивным, а переходы между экранами — плавными и интуитивно понятными.

Переходы могут быть как простыми, так и сложными, в зависимости от потребностей проекта. Например, можно использовать:

- Плавное исчезновение и появление экранов.
- Анимацию для перехода между страницами.
- Ховер-эффекты для кнопок.
- Анимацию для динамичных элементов (например, свайпы, анимация кнопок).

Подготовка макета к передаче разработчику.

Все элементы интерфейса должны быть документированы и правильно структурированы для того, чтобы разработчики могли легко перенести дизайн в код.

Важно, чтобы использовались стили и компоненты для унификации дизайна и упрощения работы с проектом.

Экспортируйте все изображения и элементы в нужных форматах (например, SVG для иконок, PNG/JPG для изображений).

Подготовьте описание взаимодействий, элементов интерфейса и их состояния, чтобы разработчики понимали, как все должно работать.

Презентация проекта

Презентация — это важный момент, где вам нужно будет представить свою работу и объяснить, почему вы приняли те или иные дизайнерские решения.

Ожидается, что вы расскажете о:

- Ключевых экранах и их функциональности.
- Навигации между экранами и переходах.
- Анимациях и взаимодействиях.
- Почему выбран именно такой стиль и цветовая палитра.
- Как макет будет работать в реальных условиях (например, для мобильного приложения или веб-сайта).

Методика и порядок выполнения работы

Варианты для финального проекта:

1. Мобильное приложение для онлайн-шопинга.
2. Приложение для планирования задач и проектов.
3. Финансовое приложение для личного бюджета.
4. Сервис для заказа такси.
5. Приложение для отслеживания здоровья.
6. Приложение для медитации и релаксации.
7. Веб-сайт для бронирования путешествий.
8. Мобильное приложение для социальных сетей.
9. Образовательная платформа для онлайн-курсов.
10. Приложение для аренды велосипедов или скутеров.
11. Приложение для поиска рецептов и составления меню.

12. Сервис для аренды квартир.
13. Приложение для планирования путешествий.
14. Приложение для ресторанных заказов.
15. Веб-приложение для выставок и событий.
16. Приложение для онлайн-обучения с видеоуроками и заданиями.
17. Мобильное приложение для покупки билетов на мероприятия (концерты, театры).
18. Сервис для знакомства с новыми людьми.
19. Мобильное приложение для заказа продуктов и товаров с доставкой на дом.
20. Приложение для поиска вдохновения и идей (пример – Pinterest).

Задание 1: Составьте сценарий пользователя: что он должен делать на каждом экране (регистрация, поиск, покупка, настройка профиля и т.д.).

Задание 2: Создайте ключевые экраны:

Экран 1: Главная страница (должна содержать информацию о том, что предлагает приложение).

Экран 2: Страница регистрации или входа.

Экран 3: Главный экран с функциональными элементами (например, для интернет-магазина — каталог товаров).

Экран 4: Страница настроек или профиля.

Экран 5: Страница, подтверждающая действие (например, подтверждение покупки или успешной регистрации).

Задание 3: Настройте переходы между экранами. Используя возможности Figma, создайте прототип с переходами между экранами, где каждый элемент, например, кнопка или ссылка, будет вести на другой экран.

Настройте различные анимированные переходы, такие как:

- Плавное исчезновение/появление.
- Перелистывание экранов.
- Динамичные переходы, например, с анимацией кнопок.

Тестируйте поведение макета в режиме презентации.

Задание 4: Сформируйте UI-кит для вашего проекта, включающий кнопки, поля ввода, карточки, иконки и другие элементы, которые будут использоваться на всех экранах.

Убедитесь, что элементы в UI-ките универсальны и могут быть использованы многократно.

Подготовка файлов для экспорта: Экспортируйте все изображения и графику в нужных форматах, таких как SVG для иконок и PNG/JPG для изображений.

Убедитесь, что структура файлов чистая и упорядоченная для удобства разработчиков.

Задание 5: Подготовьте описание всех переходов и взаимодействий. Опишите, какие действия должен выполнять пользователь на каждом экране (например, что происходит при нажатии на кнопку "Регистрация", или какой экран появляется после того, как пользователь завершает покупку).

Задание 6: Подготовка презентации. Оформите слайды, где подробно изложите ключевые экраны и их функциональность. Объясните логику переходов и анимаций, которые вы использовали. Расскажите о вашем выборе цветовой палитры, шрифтов и других визуальных решений. Покажите свой прототип в действии, демонстрируя, как пользователь будет взаимодействовать с интерфейсом.

После выполнения лабораторной работы студенты должны:

1. Создать полноценный интерактивный макет с 5+ экранами, настроенными переходами и анимациями.
2. Подготовить макет для передачи разработчику с документированными взаимодействиями и экспортированными файлами.
3. Презентовать свой проект, объяснив принятые дизайнерские решения и продемонстрировав работу интерфейса в реальных условиях.

Контрольные вопросы

1. Что такое интерактивные переходы в Figma и как они улучшают пользовательский опыт?

2. Какие ключевые экраны должны присутствовать в любом цифровом продукте?
3. Как подготовить макет к передаче разработчику? Какие элементы необходимо экспортировать?
4. Какие инструменты в Figma используются для создания анимаций и взаимодействий?
5. Какие принципы удобства использования нужно учитывать при проектировании финального макета?
6. Что такое UI-кит, и зачем он нужен в процессе разработки интерфейса?
7. Как правильно настроить адаптивность интерфейса в Figma?
8. Какие ошибки могут возникнуть при проектировании интерактивного прототипа и как их избежать?
9. Какие элементы интерфейса необходимо документировать для передачи разработчику?
10. Как правильно презентовать свой дизайн-проект, чтобы аргументировать принятые дизайнерские решения?

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное
автономное образовательное
учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

«UX/UI дизайн»

Направление подготовки 09.03.04 «Программная инженерия» Направленность
(профиль) «Инженерия сложных программных систем»
Квалификация выпускника: бакалавр

Ставрополь

2026

ВВЕДЕНИЕ

Цель изучения дисциплины «UX\UI дизайн» является подготовка будущих специалистов в области цифрового дизайна, способных разрабатывать интуитивно понятные интерфейсы и улучшать взаимодействие пользователей с цифровыми продуктами, а также компетенций будущего бакалавра по направлению подготовки 09.03.04 Программная инженерия.

Задачами обучения являются:

изучение основ UX\UI-дизайна, его принципов, методологий и современных тенденций;

изучение процесса исследования пользователей, выявления их потребностей и построения User Personas;

изучение методов проектирования пользовательского пути (Customer Journey Map) для создания удобного взаимодействия с продуктом;

изучение принципов информационной архитектуры и основ wireframing (создания каркасных макетов);

обретение навыков работы с инструментарием Figma для проектирования интерфейсов, создания интерактивных прототипов и совместной работы над проектами;

изучение основ типографики, цветовых решений, дизайн-систем и UI-китов для создания визуально целостных и удобных интерфейсов;

обретение навыков оформления и презентации финального проекта, подготовки макетов к передаче разработчикам.

Дисциплина «UX\UI» относится к циклу Б1 вариативная часть, дисциплины по выбору (Б1.В.04). Ее освоение происходит в 3 семестре. Для успешного освоения дисциплины «UX\UI дизайн» необходимы знания, полученные в ходе освоения дисциплин:

- Введение в специальность.

Знания, полученные в результате освоения дисциплины «UX\UI дизайн» могут быть использованы при подготовке отчетов по преддипломной практике и выпускной квалификационной работы.

В ходе изучения дисциплины «UX/UI дизайн» у обучающегося формируются набор следующих компетенций:

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-8 – Способен адаптировать и применять методы искусственного интеллекта и машинного обучения для решения профессиональных задач, включая интеграцию готовых моделей в программные продукты	ИД-1ПК-8 Проектирует пользовательские интерфейсы с учетом возможности персонализации на основе ML-алгоритмов (рекомендательные системы, предсказание действий пользователя, адаптивный контент) ИД-2ПК-8 Разрабатывает сценарии взаимодействия (user flows), включающие обработку и визуализацию результатов работы AI-моделей (например, отображение распознанных объектов, confidence scores, альтернативных вариантов) ИД-3ПК-8 Учитывает в дизайне интерфейсов особенности представления данных, полученных от ML-моделей: индикаторы уверенности, пояснения результатов (explainable AI), обработку неопределенности ИД-4ПК-8 Проводит юзабилити-тестирование интерфейсов с AI-функциями, оценивая понятность их работы для пользователя, выявляя проблемы восприятия и доверия к рекомендациям системы	Результаты проектирования пользовательского интерфейса с учетом возможности персонализации на основе ML-алгоритмов. Не владеет необходимыми навыками и знанием проектирования пользовательского интерфейса для достижения результата. Результаты демонстрация умения разрабатывать сценарии взаимодействия (user flows), включающие обработку и визуализацию результатов работы AI-моделей Результаты демонстрация учета в дизайне интерфейсов особенности представления данных, полученных от ML-моделей: индикаторы уверенности, пояснения результатов (explainable AI), обработку неопределенности Результаты умений проводить юзабилити-тестирование интерфейсов с AI-функциями, оценивая понятность их работы для пользователя, выявляя проблемы восприятия и доверия к рекомендациям системы

1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Самостоятельная работа студентов является важнейшим условием формирования научного способа познания. Она проводится накануне каждого лабораторного занятия и включает изучение необходимого для выполнения лабораторных работ и подготовки к практическим занятиям теоретического материала, а также подготовки отчетов по выполненным лабораторным и практическим занятиям. Подготовленный материал оформляется в виде отчета. Самостоятельные занятия (СЗ) являются одной из активных форм обучения. Самостоятельные занятия по дисциплине «UX/UI дизайн» имеют целью: - закрепить и углубить знания, полученные студентами на лекциях и в процессе лабораторных и практических занятий; - привить практические навыки в оформлении организационных, распорядительных и информационно-справочных документов. Самостоятельные занятия проводятся в специализированных аудиториях, оборудованных средствами вычислительной техники и возможностью пользования Internet-ресурсами, в библиотеке. Предлагаемые методические рекомендации содержат информацию для студентов, необходимую при подготовке и проведении лабораторных занятий по дисциплине «Основам проектной деятельности».

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа по дисциплине «UX\UI дизайн» состоит из двух частей:

1. Самостоятельное изучение вопросов;
2. Самостоятельная разработка дизайн проекта на базе умений и навыков, полученных на лабораторных работах.

3. СОДЕРЖАНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

3.1. Темы для самостоятельного изучения

1. Психология восприятия информации в интерфейсах.
2. Основные UX-метрики и способы их измерения.
3. Основные принципы анимации в UI.
4. Ошибки при создании анимации в интерфейсе.

В процессе самостоятельного изучения вопросов курса студент изучает предложенные им по рекомендуемому преподавателем списку литературы и предоставляет преподавателю краткий конспект, по которому в дальнейшем проводится индивидуальное собеседование.

На зачетной неделе студент должен предоставить готовый проект.

4. ОРГАНИЗАЦИЯ КОНТРОЛЯ ЗНАНИЙ, ПОЛУЧЕННЫХ ПРИ ВЫПОЛНЕНИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В процессе собеседования и защиты самостоятельного проекта: Оценка «отлично» выставляется студенту, если теоретическое содержание вопросов самостоятельного изучения освоено полностью, без пробелов; исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует в ответе дополнительный материал все предусмотренные программой задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному; анализирует полученные результаты; проявляет самостоятельность при выполнении заданий.

Оценка «хорошо» выставляется студенту, если теоретическое и практическое содержание вопросов самостоятельной работы освоено полностью, необходимые практические компетенции в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка «удовлетворительно» выставляется студенту, если теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, большинство предусмотренных программой заданий выполнено, но в них имеются ошибки, при ответе на поставленный вопрос студент допускает неточности, недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении программного материала. Практические навыки освоены частично.

Оценка «неудовлетворительно» выставляется студенту, если он не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, необходимые практические компетенции не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, качество их выполнения оценено числом баллов, близким к минимальному.

5. РЕКОМЕНДАЦИИ ПО РАБОТЕ С ЛИТЕРАТУРОЙ И ИСТОЧНИКАМИ

5.1 Рекомендации студентам по организации работы с литературой и источниками Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «UX\UI дизайн», которые прямо или косвенно относятся к отрабатываемой теме. Изучая литературу и источники, студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные на ваш взгляд мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, студенту следует выписать их в виде цитат с полным указанием библиографических источников. Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо выписывать все выходные данные по используемой литературе и источникам. Важным этапом при работе с рекомендуемой литературой и источниками является изучение законодательных и нормативных актов федерального, регионального, местного и ведомственного уровней. При изучении Указов Президента РФ, Законов и Кодексов РФ, постановлений, положений, рекомендаций и т.д., студент должен выяснить все изменения и дополнения, которые могли быть внесены после их выхода в свет. Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «UX\UI дизайн».

