

Министерство образования Республики Беларусь  
Учреждение образования «Витебский государственный  
университет имени П.М. Машерова»  
Кафедра прикладного и системного программирования

# **РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ**

*Методические рекомендации*

*Витебск  
ВГУ имени П.М. Машерова  
2024*

УДК 004.51(076.5)  
ББК 16.32я73  
Р17

Печатается по решению научно-методического совета учреждения образования «Витебский государственный университет имени П.М. Машерова». Протокол № 2 от 20.12.2023.

Составитель: преподаватель кафедры прикладного и системного программирования ВГУ имени П.М. Машерова **Л.В. Иванова**

**Р е ц е н з е н т ы :**

доцент кафедры информационных технологий и управления бизнесом  
ВГУ имени П.М. Машерова,  
кандидат биологических наук, доцент *А.А. Чиркина*;  
заведующий кафедрой «Математика и информационные технологии»  
УО «ВГТУ», кандидат физико-математических наук, доцент *Т.В. Никонова*

**Р17**      **Разработка пользовательских интерфейсов : методические рекомендации / сост. Л.В. Иванова. – Витебск : ВГУ имени П.М. Машерова, 2024. – 29 с.**

Методические рекомендации содержат краткие теоретические сведения, вопросы и задания для лабораторных работ по дисциплине «Разработка пользовательских интерфейсов».

Данное издание предназначено для студентов факультета математики и информационных технологий специальностей 1-40 01 01 03 Программное обеспечение информационных технологий. Базы данных и программное обеспечение информационных систем и 6-05-0612-01 Программная инженерия дневной и заочной форм получения образования. Также может быть использовано при изучении дисциплины «Бизнес-анализ в IT-проектах» углубленного высшего образования для специальности «информатика и технологии программирования».

УДК 004.51(076.5)  
ББК 16.32я73

© ВГУ имени П.М. Машерова, 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 ВВЕДЕНИЕ В РАЗРАБОТКУ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕР- ФЕЙСОВ .....	5
1.1 Понятие пользовательского интерфейса и пользовател- ского опыта .....	5
1.2 Подходы к разработке пользовательского интерфейса .....	6
1.3 Принципы удобного использования .....	8
2 ЭТАПЫ РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА	10
2.1 Этап исследования .....	10
Лабораторная работа 1. Исследование предметной области .....	12
2.2 Этап разработки пользовательских сценариев .....	13
Лабораторная работа 2. Пользовательские сценарии .....	15
2.3 Этап определения структуры интерфейса .....	15
Лабораторная работа 3. Разработка карты экранов .....	16
2.4 Этап прототипирования интерфейса .....	16
Лабораторная работа 4. Прототипирование .....	20
2.5 Этап определения стилистики .....	21
Лабораторная работа 5. Разработка UI Kit .....	27
2.6 Этап разработки дизайн концепции .....	28
2.7 Этап оформления всех экранов .....	28
Лабораторная работа 6. Вёрстка .....	28
ЛИТЕРАТУРА .....	28

## ВВЕДЕНИЕ

Пользовательский интерфейс является основным способом взаимодействия пользователя с программным обеспечением или веб-приложением и напрямую влияет на пользовательский опыт. Поэтому важно создавать приложения с высоким уровнем удобства использования и привлекательным дизайном. В этих условиях умение создавать web-приложения становится необходимой компетенцией дипломированного специалиста.

Материал соответствует отдельным темам учебной программы курса «Разработка пользовательских интерфейсов» для специальностей 1-40 01 01 03 Программное обеспечение информационных технологий. Базы данных и программное обеспечение информационных систем и 6-05-0612-01 Программная инженерия Витебского государственного университета имени П.М. Машерова дневной и заочной форм получения образования. Также могут быть использованы при изучении дисциплины «Бизнес-анализ в IT-проектах» углубленного высшего образования для специальности «Информатика и технологии программирования».

Учебное издание включает в себя 2 раздела. В первом разделе раскрывается понятие пользовательского интерфейса и основные подходы к его проектированию. Второй раздел посвящен этапам разработки пользовательских интерфейсов и принципам визуального дизайна. Каждая тема содержит краткую справочную информацию, примеры, задания для выполнения в рамках лабораторных работ. Работа с методическими рекомендациями способствует усилению информационной подготовки студентов как обязательной составляющей образовательного процесса, направленной на подготовку специалистов, способных эффективно применять средства информационных и коммуникационных технологий в процессе осуществления своей профессиональной деятельности. Данное издание ориентировано на поддержку проведения лабораторных и практических занятий в компьютерных классах; может также использоваться для самостоятельной работы, индивидуального и дистанционного обучения.

# 1 ВВЕДЕНИЕ В РАЗРАБОТКУ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Создание информационной системы – это достаточно сложный процесс, проходящий несколько этапов. Количество этапов и их назначение могут отличаться, но, как правило, выделяют пять этапов, в том или ином виде присутствующих в процессе разработки ПО всегда. К ним относятся:

1. Анализ требований. На этом этапе ведется работа с заказчиком или будущими пользователями программного обеспечения, формулируются и документируются требования к разрабатываемому ПО.

2. Проектирование. На данном этапе создается некий каркас системы (прототип, прообраз), который удовлетворяет сформированным требованиям.

3. Программирование. На данном этапе с помощью выбранных языков программирования, библиотек и технологий создается программный код, реализующий спроектированную ранее модель предметной области и алгоритмы обработки пользовательских данных.

4. Тестирование. На этом этапе оценивается корректность работы системы, выявляются и исправляются дефекты (работа системы, не совпадающая со сформулированными требованиями).

5. Сопровождение. На этом этапе осуществляется ввод в эксплуатацию разработанного программного обеспечения, консультирование пользователей по возникающим вопросам и проблемам функционирования программного обеспечения (техническая поддержка), исправление обнаруженных ошибок, пропущенных при тестировании,

Проектированием ПИ занимается дизайнер пользовательских интерфейсов (UI Designer). Он является экспертом в области создания эффективных и удобных интерфейсов. UI-дизайнер может подключаться к работе над проектом как на этапе проектирования, так и на этапе анализа требований. Фактически его работа относится к задачам этапа проектирования, однако чем раньше начнется работа над интерфейсом приложения, тем лучше будут согласованы требования, так как визуальный интерфейс легче воспринимается заказчиком и разработчиками.

## 1.1 Понятие пользовательского интерфейса и пользовательского опыта

Пользовательский интерфейс – это набор программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером (или приложением). Основу такого взаимодействия составляют диалоги. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером (приложением), осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи.

Однако необходимо помнить о том, что пользовательский интерфейс не только решает проблему взаимодействия, но и делает это взаимодействие

максимально комфортным. Поэтому перед дизайнером ставится задача – помочь пользователю легко разобраться с приложением и получить при этом удовлетворение от работы с ним. Алан Купер писал: «Пользователя не следует заставлять взаимодействовать с программой дольше, чем абсолютно необходимо для решения той или иной задачи». Поэтому большое внимание уделяется процессу построения пользовательских интерфейсов (UI – User Interface) и выстраиванию пользовательского опыта (UX – User Experience).

**UI** (User Interface – «пользовательский интерфейс») представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными устройствами и аппаратурой. Иными словами, это тот набор кнопок, ссылок, форм, диалоговых окон, иконок, пиктограмм, баннеров, ползунков и лент прокрутки, с помощью которого пользователь управляет продуктом. User Interface – это дизайн визуальной составляющей интерфейса, т.е. то, как выглядит интерфейс, и то, какие физические характеристики приобретает (какого цвета будут элементы интерфейса, удобно ли будет попадать пальцем на кнопки, читабельным ли будет текст и т.д.)

**UX** (User Experience – «опыт пользователя») помогает понять пользователя. В UX-дизайне больше психологического аспекта, нежели технологического. UX – это то, какой опыт (впечатление) получает пользователь от работы с вашим интерфейсом. «Опыт» в данном случае – это субъективное восприятие и чувства пользователя, возникающие при взаимодействии с приложением, которые складываются из визуального впечатления, интуитивности (легко ли ориентироваться в приложении), эмоционального отклика (вызывает ли интерфейс положительные эмоции у пользователя), функциональности (насколько эффективно пользователь может решать свои задачи с помощью приложения), доступности (насколько удобен интерфейс для пользователей с различными потребностями, например, людей с ограниченными возможностями).

UX-проектирование пользовательского интерфейса – это проектирование взаимодействия пользователя с интерфейсом. UX-дизайнер разрабатывает сценарии того, как и каких целей может достигать пользователь при взаимодействии с цифровым продуктом.

Пользовательский интерфейс и пользовательский опыт тесно связаны и, как правило, не рассматриваются отдельно друг от друга, хотя у каждого свои задачи. UX нельзя воплотить визуально без UI-дизайна, но и красивый интерфейс, разработанный без глубокого понимания пользователя, скорее всего окажется неудачным.

## **1.2 Подходы к разработке пользовательского интерфейса**

Существует множество рекомендаций от специалистов по проектированию пользовательского интерфейса. Эти рекомендации в той или иной мере применимы как для проектирования интерфейсов настольных и мобильных, так и для веб-приложений.

Одним из подходов разработки интерфейса является разработка *дизайна, ориентированного на пользователя (UCD – User Centered Design)*.

UCD – это итеративный процесс проектирования, где в центре внимания находятся пользователи и их потребности. Вместо того, чтобы начинать с технических спецификаций или бизнес-требований, UCD фокусируется на пользователях, их требованиях, целях и отзывах, даже включая их (пользователей) в процесс проектирования. При этом удобство использования для пользователя находится в центре внимания на протяжении всего процесса разработки.

Важную роль в подходах UCD играет *дизайн-мышление (design thinking)* – процесс, цель которого понять пользователей, определить их возможные проблемы, чтобы разработать решения, которые могут быть не очевидны, но в результате помогают удовлетворить потребности пользователей.

Впервые данная концепция была упомянута в 1969 году ученым-когнитивистом и лауреатом Нобелевской премии Гербертом А. Саймоном в книге «Наука об искусственном». Но основоположником методологии Дизайн-мышления является Дэвид Келли.

**Этапы дизайн-мышления.** Существуют разные методологические ветви Design Thinking. Одна из самых известных – «Двойной Бриллиант» (Double Diamond). Согласно данной модели дизайн-мышление включает в себя следующие этапы (рисунок 1):

- 1) исследование
- 2) анализ и синтез
- 3) разработка решений
- 4) прототипирование.

Эти этапы не всегда последовательны, и команды часто выполняют их параллельно, а не по порядку, и повторяют их итеративно.

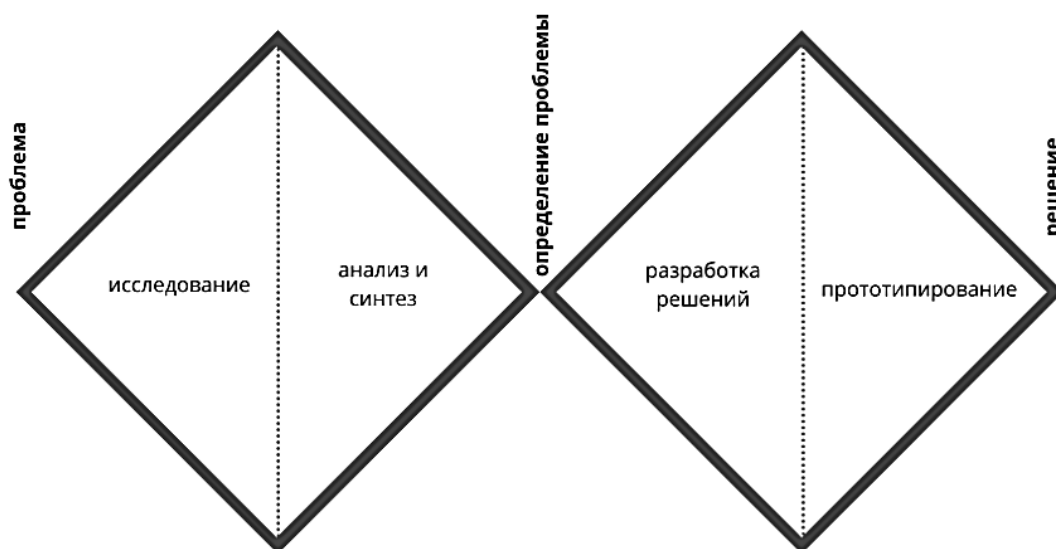


Рисунок 1 – Этапы дизайн-мышления

При данном подходе процесс разработки ПО разбивается на два блока (два «бриллианта»). Первый блок отвечает за поиск верного решения, помогает понять проблему пользователя и изучить ее. Он включает в себя два этапа:

- исследование – эмпирическое понимание проблемы. Данный этап включает консультации с экспертами, чтобы узнать больше о проблемной области, путем наблюдения, взаимодействия, что позволяет понять их опыт и мотивацию, а также погрузиться в физическую среду, чтобы получить более глубокое личное понимание затронутых вопросов. Данный этап имеет решающее значение для процесса проектирования, ориентированного на человека, т.к. он позволяет отказаться от собственных предположений и получить реальное представление о пользователях и их потребностях. Компания собирает всю возможную информацию о пользователях: где, как и почему они используют приложение.

- анализ и синтез – этап изучения полученной информации. Здесь формулируются и выявляются наиболее острые проблемы, которые требуют решений, а также продумываются решения, которые могут быть предложены пользователям для удовлетворения их потребностей.

Второй бриллиант отвечает за верную реализацию решения и включает следующие два этапа:

- разработка решений – этап, цель которого заключается в создании как можно большего количества идей для рассмотрения всех возможных путей решения, сформулированных на предыдущих этапах проблем.

- прототипирование – последний шаг, заключающийся в отсеивании наименее жизнеспособных и реализуемых идей, полученных на предыдущем этапе. На данном этапе происходит прототипирование, тестирование и доработка предыдущих решений. Команда создаёт несколько недорогих, уменьшенных версий приложения, чтобы исследовать идеи, которые были сгенерированы.

### **1.3 Принципы удобного использования**

Ларри Константин, идеолог концепции дизайна, ориентированного на использование, в книге «SoftwareForUse», написанной им в 1999 г. совместно с Люси Локвуд, представил принципы удобного использования, которые должны быть учтены при разработке интерактивных систем:

- структурный принцип: проектирование интерфейса должно вестись целенаправленно, с использованием конструктивных решений, основанных на четких и последовательных моделях, узнаваемых для пользователя;
- принцип простоты: дизайн должен быть простым, общие задачи должны быть понятны, общение между программой и человеком должно происходить на родном для него языке;



– принцип видимости: все необходимые для решения конкретной задачи элементы интерфейса должны быть видимы и не должны отвлекать пользователя посторонней или избыточной информацией.

– принцип обратной связи: дизайн должен информировать пользователей о выполняемых действиях, изменениях состояния или условий, об ошибках или исключениях;

– принцип толерантности: дизайн должен быть гибким и терпимым к действиям пользователей, позволять отмену и повторное выполнение операций, а также предотвращать ошибки, интерпретируя все входные последовательности в разумные действия;

– принцип повторного использования: интерфейс должен использовать согласованные внутренние и внешние компоненты, тем самым уменьшая для пользователей необходимость переосмысления или запоминания их (компонентов) назначения и поведения.

### **Контрольные вопросы**

- 1) Что такое пользовательский интерфейс?
  - 2) Какие подходы к разработке пользовательских интерфейсов существуют?
  - 3) Что такое пользовательский опыт?
  - 4) Что такое дизайн, ориентированный на пользователя?
  - 5) Что такое дизайн-мышление?
- б) Опишите этапы дизайн-мышления.

## 2 ЭТАПЫ РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Полный цикл разработки интерфейса можно разделить на несколько этапов (рисунок 2).

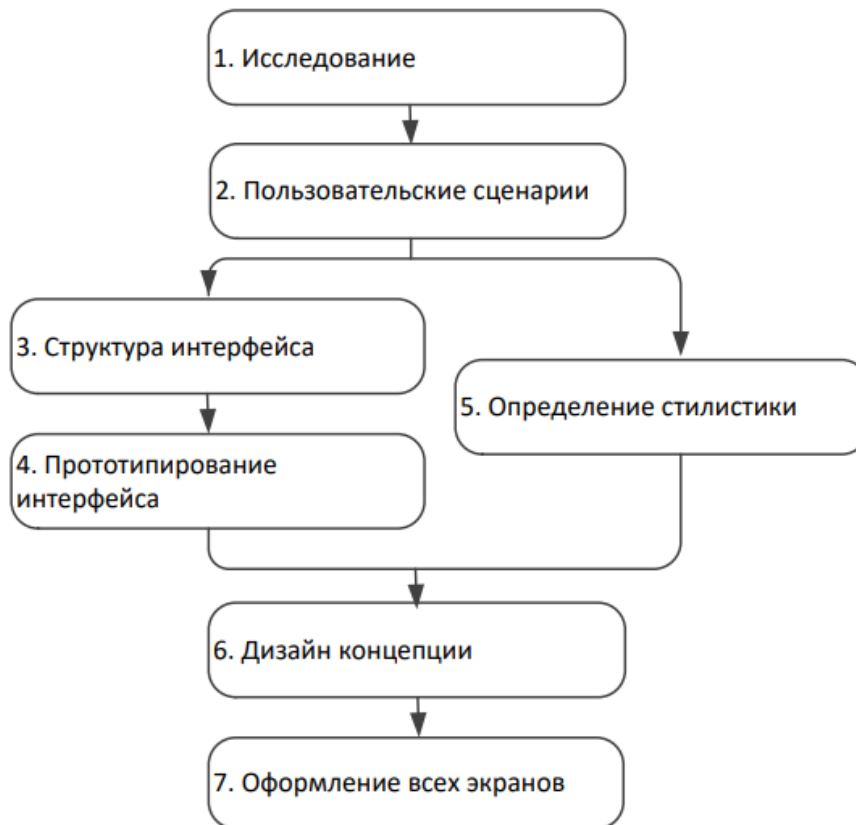


Рисунок 2 – Этапы разработки пользовательского интерфейса

### 2.1 Этап исследования

Прежде чем начать проектирование, необходимо выполнить исследование предметной области, которое позволяет лучше понять цели бизнеса, атрибуты бренда и технические ограничения.

На этапе исследования проводится сбор информации о приложении, о существующих аналогах, о пользователях, осуществляется анализ устройств предполагаемой целевой аудитории. Всё это помогает понять, для кого разрабатывается интерфейс, с какими ограничениями следует его делать (размеры экранов, интерактивность). Данный этап разработки подразумевает под собой сбор, систематизацию и анализ требований к ПО (как правило, это является задачей бизнес-аналитика).

#### **Основы исследования пользовательской аудитории.**

Всегда нужно помнить о том, что каждая группа пользователей уникальна. К примеру, ожидания целевой аудитории нового мобильного теле-

фона будут кардинально отличаться от ожиданий целевой аудитории программного обеспечения. При этом уникален и каждый пользователь отдельно. То, с чем не справится один человек, не представит сложности для другого. Главная задача – понять пользователей в целом, чтобы суметь выделить общие поведенческие шаблоны. Для этого нужно ответить на следующие вопросы:

- Какую цель пользователь ставит при работе с ПО или веб-сайтом?
- Какие задачи выполняет пользователь для достижения этих целей?
- Каковы навыки пользователей в использовании ПО, аналогичного тому, которое Вами разрабатывается?
- Какие трудности испытывают пользователи при работе с подобным ПО?
- Каковы ожидания пользователей?

Ответы на эти вопросы помогут лучше понять потребности пользователей и в конечном счёте разработать приложение с максимально удобным для них интерфейсом.

Существует достаточно много способов анализа пользовательской аудитории.

**Непосредственное наблюдение и интервью.** Интервьюирование чаще всего проводится «на местах», т.е. там, где пользователи фактически используют ПО, например, на работе. Интервью может быть структурированным – состоящим из заранее определенного набора вопросов, или неструктурированным – вопросы задаются в зависимости от ответов респондента.

**Исследование конкретных случаев.** Разрабатывая дизайн специфического программного обеспечения или веб-сайта, необходимо как можно больше знать о реальном контексте использования приложения. Контекст использования приложения – это про условия, в которых человек будет им пользоваться. Здесь нужно задать себе три вопроса: кто будет пользоваться (например, для детей принято делать кнопки и пункты меню большими и контрастными), где будет использоваться (дома, на улице, в транспорте, в горах), когда будет использоваться (в обычной обстановке, когда угодно, в экстремальной ситуации).

**Письменные опросы.** Позволяют получить данные сразу от множества пользователей и дают чёткую картину определённых аспектов целевой аудитории. Здесь необходимо не просто собрать отзывы пользователей, а получить конкретные данные, которые можно использовать для последующего проектирования интерфейса. Поэтому важно продумать сами вопросы, выбрать адресатов опроса и правильно проанализировать их ответы.

**Искусственные образы.** Данный метод не относится к методам сбора данных, а позволяет понять, что делать собранной информацией. Это техника, которая «моделирует» целевую аудиторию. Для каждой группы пользователей создаётся вымышленный представитель, который объединяет

в себе наиболее важные аспекты пользователей своей группы: какие задачи они пытаются выполнить, каковы их конечные цели и т.д. В процессе создания дизайна интерфейса можно задавать себе вопрос «Действительно ли этот вымышленный представитель сделал бы это действие? Что он мог бы сделать вместо этого?»

### **Лабораторная работа 1. Исследование предметной области**

**Задание.** Разработайте техническое задание на некоторое веб-приложение (предметную область выберите согласно одному из вариантов ниже). Приложение должно иметь механизм регистрации и аутентификации. При написании технического задания обязательно провести анализ аналогов вашего приложения. Должны быть расписаны их возможности.

#### **Варианты:**

1) Туристическое агентство. Приложение должно содержать сведения о туристических маршрутах, гостиницах, программе отдыха, заказах туров.

2) Ресторан. Приложение должно содержать сведения о меню ресторана, разделённое на категории, стоимости блюд и должно позволять заказывать столик в ресторане с заказом блюд из меню.

3) Торговая компания. Приложение должно содержать сведения о офисах компании, расположенных в разных городах, служащих компании, объемах продаж офисов за каждый день.

4) Спортивные секции. Приложение должно содержать сведения о спортивных секциях вуза, занимающихся в них студентах, расписании занятий секций, тренерах секции.

5) Агентство недвижимости. Приложение должно содержать сведения об агентстве, о его клиентах, о продаваемых и покупаемых квартирах и иных площадях.

6) Музей. Приложение должно содержать сведения о выставках

7) Городские мероприятия. Приложение должно содержать сведения о концертах, выставках, спортивных мероприятиях в городе.

8) Кинотеатр. Приложение должно позволять пользователям покупать билеты на сеансы, просматривать трейлеры, информацию о киносеансах.

9) Путеводитель. Приложение должно содержать сведения о достопримечательностях, музеях, ресторанах, отелях в определенном городе или стране.

### **Контрольные вопросы**

1) В чём заключается этап исследования предметной области приложения?

2) Опишите способы исследования пользовательской аудитории.

## 2.2 Этап разработки пользовательских сценариев

Пользовательский сценарий – это описание взаимодействия пользователя с приложением. Каждый сценарий основан на конкретной цели, которую необходимо достичь и помогает визуализировать шаги, которые пользователь предпринимает для достижения этой цели, с какими проблемами он может столкнуться и как их решить. Например, покупка продукта, поиск информации, использование услуги. Если некоторую задачу пользователь может решать несколькими способами, то формируется несколько сценариев.

Пример. Пользовательский сценарий «Обновить аватарку в профиле». Шаги для выполнения сценария:

- 1) перейти в профиль;
- 2) нажать на аватарку;
- 3) выбрать файл;
- 4) подтвердить или изменить кадрирование изображения;
- 5) сохранить.

Составленные списки шагов для каждой задачи помогают определить слишком долгий путь для решения конкретной задачи относительно остальных.

По степени детализации и технической проработки пользовательские сценарии делятся на четыре группы:

- пользовательские истории
- концептуальные сценарии
- конкретные сценарии
- сценарии использования.

**Пользовательские истории.** Это самый насыщенный подробностями вариант: рассказ, схемы, видео, фотографии – все, что помогает описать опыт взаимодействия. У каждого пользователя может быть своя история и свой специфический опыт. Если говорить о покупке: кто-то берет для себя, кто-то в подарок, кто-то подарит маме, кто-то – жене, у кого-то все хорошо с финансами, а кто-то собирал деньги на покупку, кого-то интересует кредит и рассрочка. Опыт в этих пользовательских историях будет отличаться.

К. Вигерс предлагает использовать для написания пользовательских историй следующий шаблон (хотя возможны и другие стили): «Как <тип пользователя>, я хочу <цель>, чтобы <причина>

Пример. «Как любитель кино, я хочу, чтобы на моем смартфоне была вся информация о предстоящих премьерах в местных кинотеатрах, чтобы мне не приходилось проверять сайты каждого из них по отдельности»

Пример. «Через месяц у нас юбилей свадьбы, я отложил деньги и собираюсь выбрать подарок жене. У меня есть предел по стоимости, я знаю, что жена любит серебро и носит серьги».

**Концептуальные сценарии.** Создаются с помощью объединения похожих пользовательских историй и отсеечения лишнего и несущественного.

Такой подход обобщает множество пользовательских историй в несколько сценариев.

Пример. Концептуальный сценарий «Покупка товара»

«Пользователь готовит деньги заранее, ищет через форму поиска, имеет ограниченный бюджет – сразу использует фильтры, принимает решение долго, несколько раз заходит в корзину и сравнивает товары».

**Конкретные сценарии.** Составляются с позиции персонажа (каждой группе целевой аудитории сопоставляется персонаж). Здесь еще меньше абстракции и больше конкретики. В конкретных сценариях прописывается путь достижения цели персонажем.

**Сценарии использования.** Описывают пользовательский опыт по шагам: кто, что, каким образом и в какой последовательности делает. Это должен быть наиболее детализированный и технически проработанный вариант. Единой формы для разработки пользовательских сценариев не существует. Поэтому итоговый результат может сильно отличаться. Наиболее полно описанный сценарий использования включает в себя название, первичных актёров (пользователей), предусловие, основной сценарий, альтернативные сценарии.

*Название* представляет собой краткое описание сценария. Например, «покупка товара».

*Предусловие* – это условие, при котором данный сценарий использования может быть инициирован. Предусловие может быть упоминанием о выполнении других сценариев использования.

*Основной сценарий* – это последовательность шагов, описывающая процесс решения задачи, которой посвящен сценарий использования. Шаги удобно последовательно нумеровать. При этом важна формулировка каждого шага, например, вместо «добавил товар в корзину» следует писать «пользователь нажимает на кнопку «Добавить товар в корзину»».

*Альтернативные сценарии* – это сценарии, в которых процесс развития событий на каком-либо шаге чем-либо заметно отличается от основного. Считается недопустимым прописывать альтернативный сценарий для альтернативного сценария.

Пример. Сценарий использования «Добавить товар в корзину»

Название	Добавление товара в корзину
Первичные актёры	Пользователь (авторизованный)
Предусловие	Открыта карточка товара из каталога
Основной сценарий	1. Пользователь выбирает количество товара, которое он хочет переместить в корзину 2. Пользователь нажимает кнопку «Добавить в корзину» 3. Система проверяет наличие выбранного количества товара.

	<p>4. Система добавляет выбранный товар в корзину. В иконке «Корзина» система выводит маркер – количество добавленного товара в корзину.</p> <p>5. Пользователь получает уведомление об успешном добавлении товара в корзину</p>
Альтернативный сценарий № 1	<p>4.1 Пользователь получает сообщение об отсутствии нужного количества товара на складе</p> <p>4.2 Система переходит к шагу 1</p>

## Лабораторная работа 2. Пользовательские сценарии

**Задание.** В соответствии с функционалом Вашего приложения (см. лабораторную работу 1) разработать

- пользовательские истории;
- на основе пользовательских историй сформировать концептуальные сценарии;
- построить пользовательские сценарии с учетом основных концептуальных сценариев.

### Контрольные вопросы

- 1) Что такое пользовательская история?
- 2) Что такое концептуальный сценарий?
- 3) Что такое конкретный сценарий?
- 4) Что такое сценарий использования?
- 5) Что включает в себя сценарий использования?

### 2.3 Этап определения структуры интерфейса

Полученный на предыдущем этапе список шагов ложится в основу структуры интерфейса. На данном этапе становится известно количество экранов, их краткое содержание. Здесь строятся так называемые **диаграммы пользовательского пути (User Flow Diagrams)**. User flow (в пер. с англ. путь пользователя) – это наглядное представление последовательности действий, которые выполняет пользователь для достижения значимой для себя цели в приложении, и охватывает некоторую отдельную функцию.

User flow применяют, чтобы разработать интуитивно понятный интерфейс, который позволит пользователям добиться желаемого за наименьшее количество шагов. Кроме того, создание подобных диаграмм позволяет команде по разработке вносить корректировки и менять логику приложения еще до создания дизайна и кода.

Объединение всех таких диаграмм представляет собой **карту экранов (User Flow Map)**, которая представляет собой схему, состоящую из экранов или страниц приложения с минимальными деталями дизайна (рисунок 3). Карта экранов помогает продумать логику и структуру интерфейса приложения, определить основные элементы и функции на каждом экране.

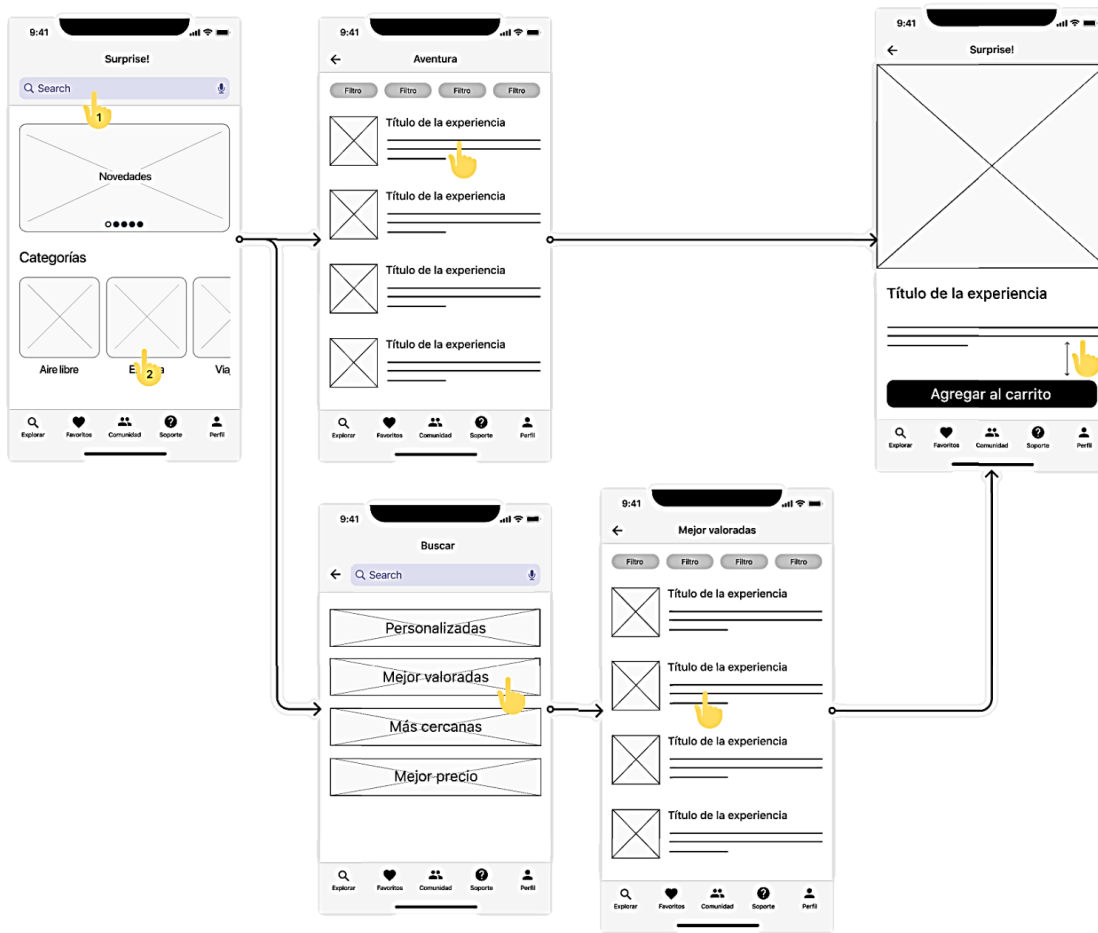


Рисунок 3 – Фрагмент карты экранов

### Лабораторная работа 3. Разработка карты экранов

**Задание.** Разработайте карту экранов для мобильной версии вашего приложения. Основная задача – проработать расположение ключевых элементов интерфейса для достижения целей пользователей и способы перехода между страницами. Карта экранов должна быть разработана с помощью онлайн-сервиса [figma.com](https://www.figma.com).

#### Контрольные вопросы

- 1) Что такое диаграмма пользовательского пути?
- 2) Что такое карта экранов?

#### 2.4 Этап прототипирования интерфейса

Процесс создания интерфейса приложения, позволяющего заказчику протестировать работу с ним и оценить удобство использования, называют *прототипированием*. Прототипом может являться набор сверстанных HTML-страниц для web-приложения, настольное или мобильное приложение, содержащее визуальные формы, рисунки в одном из графических форматов, изображающие внешний вид приложения, схематичные рисунки от



руки, демонстрирующие лишь концепцию пользовательского интерфейса (например, расположение элементов интерфейса и их состав). Способ прототипирования и его объём определяются исходя из специфики проекта. Например, для рекламных сайтов определяющим является именно внешний вид, а не его функциональные возможности, соответственно, и прототип будет ориентирован, прежде всего, на эффектный внешний вид. Для приложения в области бухгалтерского учёта важнее функционал, поэтому на первых этапах пользовательский интерфейс достаточно спроектировать схематично.

В большинстве случаев реализуется два схематичных прототипа: *черновой* и *финальный*. Черновой прототип представляет собой схематичные изображения экранов, связанные между собой, с основными зонами без деталей. Такой прототип позволяет более наглядно определить, на сколько объёмным будет приложение, как много информации будет на каждом экране (рисунок 4). На финальном прототипе появляются все кнопки, тексты, элементы форм и пр.

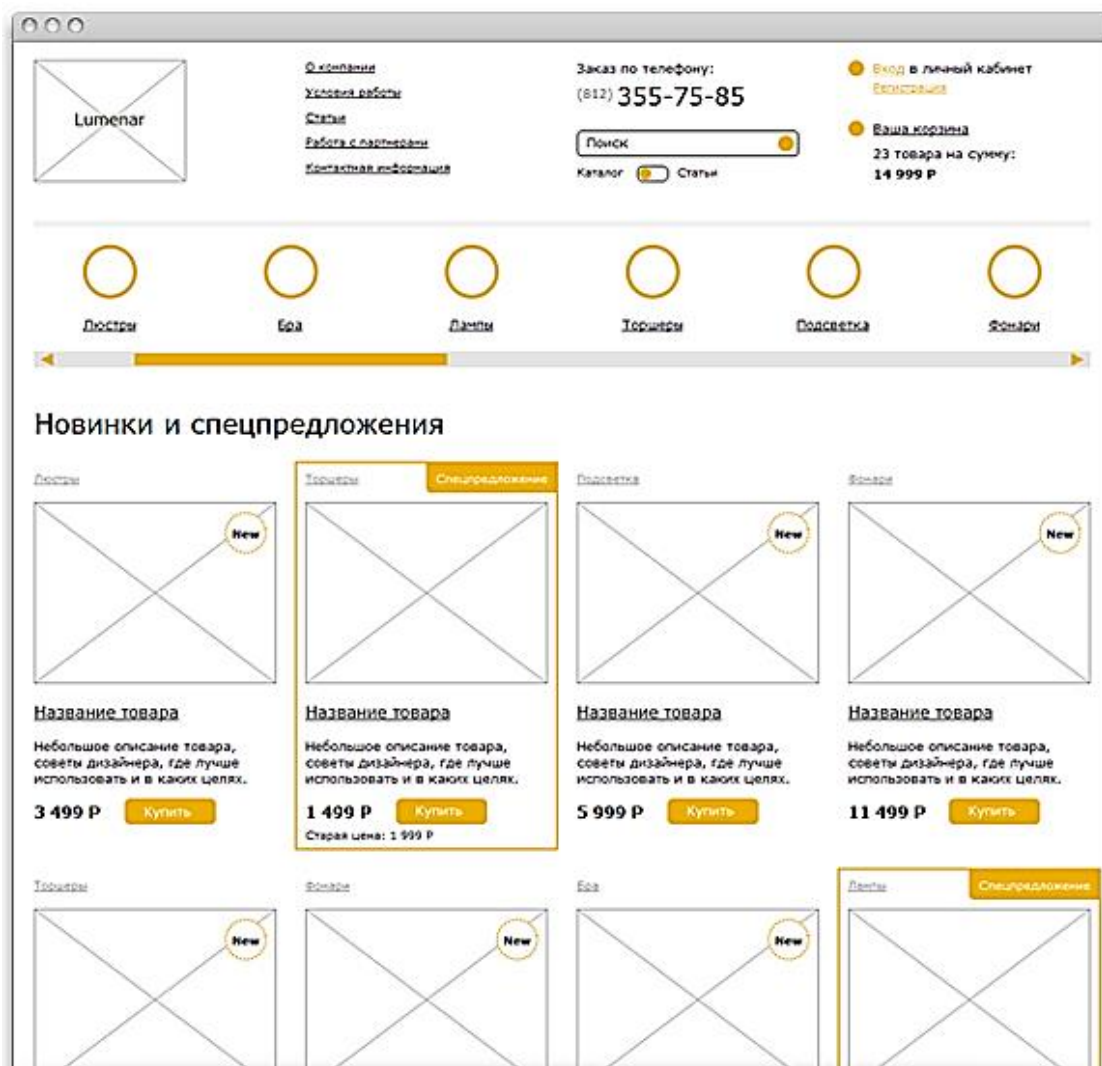


Рисунок 4 – Прототип приложения

## Адаптивный пользовательский интерфейс

В контексте разработки веб-приложений нужно уделять особое внимание адаптивному пользовательскому интерфейсу, чтобы обеспечить комфортное взаимодействие пользователей с приложением на различных устройствах (компьютеры, планшеты, смартфоны) с различными характеристиками (размер экрана, разрешение, ориентация и др.).

Дизайнер обычно создаёт отдельные прототипы веб-страниц (экранов) для различных типов устройств.

Основными принципами при проектировании адаптивного веб-дизайна являются:

1) *проектирование для мобильных устройств с самых ранних этапов.* Проектирование начинается с адаптивной версии веб-сайта для мобильных устройств. На этом этапе дизайнеры стремятся правильно передать смысл и основные идеи с использованием небольшого экрана и всего одной колонки. Содержимое при необходимости сокращают, удаляя второстепенные информационные блоки и оставляя самое важное.

2) *работа с медиазапросами*, которые определяют тип устройств (проекторы, смартфоны, мониторы, телевизоры и пр.) и условия (как правило связанные с разрешением экрана). На соответствующий запрос и ответ будут применяться соответствующие устройству параметры отображения из файла стилей `css`.

3) применение гибкого макета на основе сетки (`grid-based layout`) и использование гибких изображений (`flexible images`).

Подходы к созданию адаптивных веб-приложений развиваются. Первым из них был так называемый *резиновый дизайн*, когда контент растягивался или сжимался. Для разных разрешений стационарных мониторов такой подход был вполне применим, но адаптировать дизайн сайта для маленьких экранов смартфонов он уже не мог. Изображение и текст становились слишком мелкими. Поэтому появились более функциональные методы.

*Адаптивный макет.* В этом случае прототип веб-приложения создается на основе нескольких макетов, фиксированных для стандартных размеров экранов. Фактически дизайнер проектирует несколько разных дизайнов, между которыми выполняется автоматическое переключение.

*Отзывчивый макет.* При таком подходе создается проект веб-приложения с гибкой сеткой, гибкими изображениями и медиазапросами. Для такого дизайна не имеет значения, открыто веб-приложение на мобильном устройстве или на десктопе, т.к. макет в любом случае будет выглядеть хорошо.

*Смешанный макет.* Если необходимо сделать макет максимально адаптированным к всевозможным размерам экранов, можно скомбинировать адаптивный и отзывчивый макет. Такой подход часто используют при создании мобильной версии веб-приложения на основе десктопной.

**Расположение элементов в адаптивном макете.** После определения общей инфраструктуры приложения и элементов пользовательского интерфейса, дизайнеры организуют их в сетку, которая обеспечивает единый каркас и схему расположения всех основных блоков и элементов. Использование сетки не является обязательным, это лишь инструмент, который помогает создавать симметричные, упорядоченные и легко масштабируемые макеты.

Существует несколько видов сеток:

- колоночная сетка – состоит только из вертикальных колонок. Объекты располагаются, опираясь на эту сетку (один блок может занять две колонки, другой – четыре и т.д.);
- модульная сетка характеризуется наличием как вертикального деления, так и горизонтального;
- иерархическая сетка – состоит из блоков, размещенных интуитивно и не поддающихся никаким закономерностям.

Однако с точки зрения разработки адаптивного макета наибольший интерес представляют колоночные и модульные сетки.

В самом простом понимании модульная сетка – это решетка из ячеек, где одна из них взята за основную единицу измерения (*модуль*), а остальные ячейки равны или кратны ей. Как правило, сетка делит экран на несколько крупных горизонтальных и вертикальных областей.

Качественно спроектированная сетка задействует понятие *шага*, то есть минимального расстояния между элементами. Например, если шаг сетки составляет четыре пиксела, то все расстояния между элементами и группами должны быть кратны четырем.

На сегодняшний день существует большое количество технологий, использующих модульные сетки. Среди них:

- Bootstrap (фреймворк для веб-разработки). Bootstrap характерен своей продуманной и гибкой системой модульных сеток, в основе которой лежит, как правило, 12-колоночный макет. Сами колонки не обязательно имеют одинаковую ширину, они могут быть скомбинированы в любое число колонок.
- Grids (программное обеспечение для создания модульных сеток).

### **Особенности проектирования пользовательского интерфейса мобильного приложения**

Графический интерфейс для мобильных приложений требует учета особенностей пользовательского взаимодействия с мобильным устройством, в частности расположения большого пальца руки т.к. чаще всего пользователи держат устройство правой рукой, и используют для работы с экраном только большой палец. По этой причине навигационные кнопки чаще размещают внизу экрана, а верхний левый угол идеально подходит для важной информации.

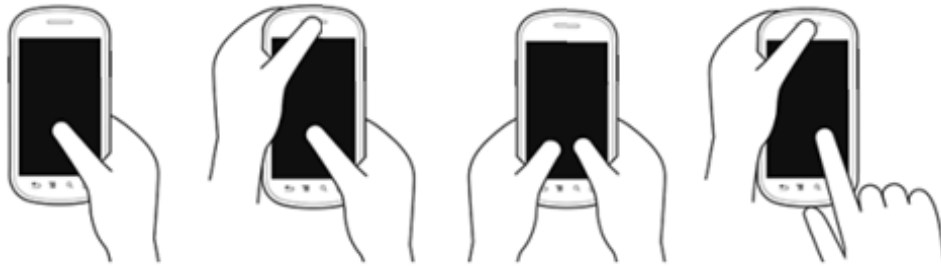


Рисунок 5 – Положение рук при работе со смартфоном

Типовые жесты – это ещё один вид взаимодействия с мобильным устройством. Сегодня пользователи уже привыкли к возможности использовать разные жесты для различных ситуаций: проведение пальцем вправо или влево для перелистывания страниц, двойное касание для увеличения изображения, зажатие и перетаскивание для перемещения объектов, проведение пальцем вверх или вниз для регулировки громкости или яркости экрана и др. Важно учитывать схемы наиболее удобных для человека жестов в проектировании пользовательского интерфейса, чтобы получить интуитивно понятный и эффективный интерфейс, который будет приятен в использовании.



Рисунок 6 – Схемы жестов

#### **Лабораторная работа 4. Прототипирование**

**Задание.** Разработайте прототипы для всех страниц десктопной версии вашего приложения, для мобильной версии и планшета. Основная задача – проработать расположение элементов интерфейса. Прототипы должны быть разработаны с помощью онлайн-сервиса [figma.com](https://www.figma.com).

#### **Контрольные вопросы**

- 1) Что такое прототип?
- 2) Назовите виды прототипов.

- 3) Что такое адаптивный дизайн?
- 4) Назовите виды адаптивного дизайна.
- 5) Что такое модульная сетка?

## 2.5 Этап определения стилистики

После этапа исследования и параллельно с этапами проектирования определяется будущая стилистика интерфейса. Это может быть подготовка нескольких наборов изображений готовых страничек приложения с иллюстрациями, кнопками, шрифтовыми композициями, связанными между собой стилистически.

Существует множество концепций, позволяющих четко и грамотно сформировать структуру интерфейса и красиво представить ее пользователям, например: *material design*, *flat design* и др.

**Material design.** Это система для создания дизайна интерфейсов, которую разработала компания Google. В данной системе содержатся рекомендации по оформлению интерфейса, который будет привычен и понятен большинству пользователей.

Основные принципы *Material Design*:

- все элементы интерфейса представляются как слои цифровой бумаги, которые располагаются на разной высоте и имеют тени. Это помогает пользователям отличить главные элементы от второстепенных и делает интерфейс интуитивно понятным;

- все элементы, которые есть на экране, не могут просто так появляться и исчезать. Каждый элемент должен быть анимирован последовательно, чтобы подсказывать пользователю последовательность его действий;

- акцентирование внимания на графике, цвете и шрифтах. Дизайн должен быть максимально естественным и логичным, все элементы – сбалансированы и гармоничны по отношению друг к другу, чтобы у «юзера» было комфортно и он понимал, что ему необходимо делать здесь и сейчас!

- применение единого адаптивного дизайна, но на каждом устройстве будет отражаться свой взгляд на одни и те же элементы, при этом неизменными остаются цвета, иконки и пространственные отношения.

**Flat design**, в отличие от предыдущего подхода, является концепцией «плоского» дизайна, при котором все элементы управления (иконки, кнопки, меню) лишены тени и визуального объема. Объекты выглядят нереалистичными, но при этом ничто не отвлекает пользователя от решения его задачи.

Основные принципы *Flat design*:

- отсутствие лишних эффектов. Все изображения двухмерные и находятся в одной плоскости. Такая графика не предполагает использования текстур, бликов и демонстрации объема;

- использование простых и узнаваемых очертаний предметов. Flat design не акцентирует внимание на реалистичности, в его основе лежат схематичные изображения, которые будут понятны пользователю;
- использование насыщенных цветовых контрастов. Если нет текстур и теней, цвет становится единственным средством, позволяющим визуальнo отделить один объект от другого. Поэтому flat design невозможен без контрастов.
- ориентация на опытных пользователей. Мы знаем, что значит кнопка Play, бургерное меню или значок «Домой», т.к. подобные иконки встречаются нам постоянно на разных ресурсах.
- плоский дизайн должен одинаково хорошо читаться и в крупном, и в мелком масштабе.
- обеспечение быстрой загрузки страниц. Плоские векторные объекты весят мало, это ускоряет работу сайтов и приложений.

Результатом работы на данном этапе является, так называемый, UI Kit (User Interface Kit – набор для пользовательского интерфейса). Это полный набор элементов и компонентов для дизайна интерфейса: шрифтов, иконок, форм и других элементов, которые помогают пользователю взаимодействовать с приложением.

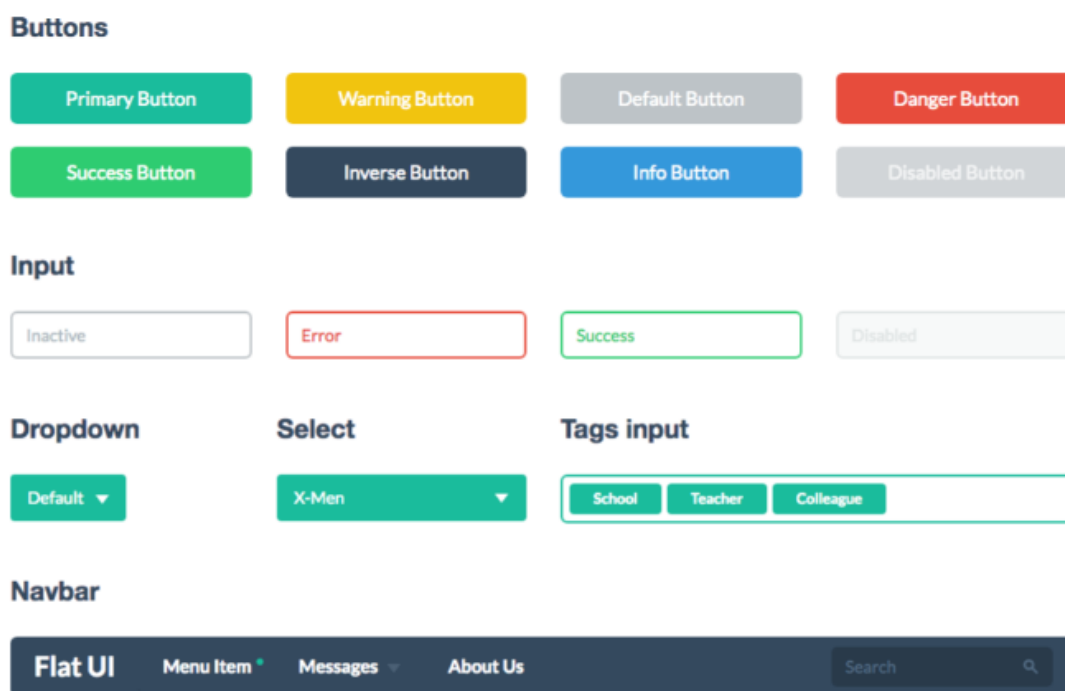


Рисунок 7 – Фрагмент UI Kit

UI Kit позволяет создать интерфейс в едином стиле. Что входит в UI Kit: Цвета и шрифты, которые используются в дизайне.

Цвет является одним из наиболее очевидных элементов дизайна, как для пользователя, так и для дизайнера. С помощью цвета можно привлечь внимание к определенной части сайта. Цвета в интерфейсах обычно подбираются исходя из цветовых схем.

*Монохроматическая цветовая схема* основана на использовании нескольких оттенках одного цвета с разной степенью черного и белого. Поэтому монохроматические цвета хорошо сочетаются, создавая успокаивающий эффект.



Рисунок 8 – Монохроматическая цветовая схема

*Аналоговые цветовые схемы* создаются из цветов, лежащих на цветовом круге рядом на друг с другом. Один цвет используется как доминирующий, в то время как другие используются для обогащения схемы.



Рисунок 9 – Аналоговая цветовая схема

*Дополнительные цветовые схемы.* В основе такой схемы лежат только два цвета, которые сильно контрастируют. Эта схема используется для привлечения внимания пользователя. При использовании данной схемы важно выбрать доминирующий цвет, а *дополнительный* цвет использовать для акцентов.



Рисунок 10– Дополнительная цветовая схема

*Триадическая цветовая схема* основывается на выборе трёх цветов, которые должны лежать на одинаковом расстоянии друг от друга на цветовом круге.



Рисунок 11 – Триадическая цветовая схема

*Тетрадные цветовые схемы* используют четыре цвета, расположенных на цветовом круге на равном расстоянии друг от друга (образуя квадрат). Как правило один цвет используется как основной, а три остальных – акцентные.

*Естественная цветовая схема* использует набор цветов, которые встречаются в природе и поэтому кажутся наиболее гармоничными и приятными для глаз.

При разработке графического интерфейса часто бывает сложно выбрать цветовую схему. Однако, с помощью определенных правил, можно создать гармоничную цветовую палитру.

**Правило ограничения в количестве цветов.** Применение цвета к дизайн-проекту имеет много общего с балансом. Чем больше цветов используется, тем труднее его достичь. Считается, что лучших результатов можно добиться, если придерживаться не более трех основных цветов, например, следуя правилу 60% + 30% + 10%. Такие пропорции предназначены для обеспечения баланса цветов: 60% – доминирующий оттенок, 30% – вторичный цвет, который поддерживает основной, при этом они достаточно отличаются, чтобы их можно было разделять, 10% – цвет акцента (это может быть цвет для призыва к действию или цвет для другого элемента, который необходимо выделить).

**Правило «сначала в оттенках серого».** Подход, при котором работа над интерфейсом начинается с его проектирования в оттенках серого, позволяет сосредоточиться на композиции, структуре и распределении элементов интерфейса без отвлечения на цветовые решения. Кроме того, серый интерфейс может быть полезен при работе над дизайном для людей с ограниченными возможностями зрения, так как он позволяет яснее оценить контраст. Цвет добавляется на последних этапах для создания более привлекательного и эмоционально насыщенного пользовательского опыта.



**Правило отсутствия чисто черного цвета.** В реальной жизни чистый черный почти никогда не встречается. Все «черные» объекты вокруг нас имеют некоторое количество света, отражающегося от них, а это значит, что они не черные, а темно-серые. Кроме того, черный цвет будучи чистым и контрастным, будет «пересиливать» другие цвета, создавая неестественное выделение.

**Выделение важного с помощью цвета.** Цвет – это инструмент, который помогает направить взгляд. Поэтому нужно использовать контрастные цвета, если необходимо что-то выделить. Как правило, высокий контраст – лучший выбор для важного контента или ключевых элементов. Если вы хотите, чтобы пользователи увидели или кликнули по элементу, сделайте его выделяющимся.

**Значение цвета.** Известно, что цвета имеют присущие им значения и влияют на эмоции пользователя. Выбирая цветовую гамму, нужно думать не только о внешнем виде интерфейса приложения, но и о том, какие ощущения он вызывает. Например, синий символизирует уверенность, спокойствие, успех, фиолетовый – очарование, гламур, престиж, зеленый – прочность, экологичность, надежность, желтый – современность, задор, красный – уверенность в себе, экспертность, мотивацию.

При выборе гаммы для приложения, необходимо принимать во внимание и тот факт, что цвет может иметь разные значения в различных культурах. Культурный аспект для цветной символики может быть очень сильным, поэтому важно знать о том, какова целевая аудитория сайта (или приложения).

Типографика – это фундамент хорошего дизайна интерфейса, она определяет читабельность и восприятие информации.

Считается, что большое количество гарнитур шрифта в интерфейсе может путать пользователя, при этом интерфейс, который использует везде только один шрифт, кажется безвкусным. Поэтому рекомендуется использовать гарнитуру одного-трех шрифтов: базовый шрифт и акцидентный (например, для заголовков). В некоторых случаях вводятся дополнительные шрифты для меню, важной информации, цитат, для мелкого текста и т.д. В веб-разработке также важно спланировать единую общую схему размеров отступов для всех элементов, иерархию заголовков и навигационных элементов (например, для древовидного меню на сайте). Все последующее оформление информации желательно строить на базе выбранной общей схемы.

Существует четыре основных типа шрифтов:

1) шрифт с засечками (serif) – это любая гарнитура шрифта, которая содержит завершающие штрихи.

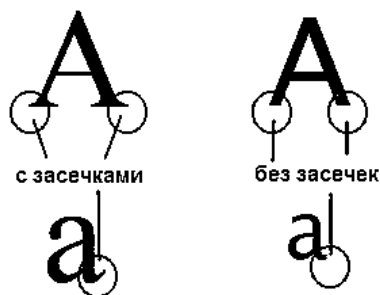


Рисунок 12– Шрифты с засечками и без засечек

2) без засечек (sans-serif) – это любая гарнитура шрифта, конечные штрихи которой не имеют никаких расширений, пересекающих штрихов или других украшений.

3) рукописный или курсив. Эти шрифты обычно выглядят по большей части как написанные пером или кистью, они будут включать те, которые кажутся рукописными, даже хотя и не являются курсивом. Одной из причин отказа от использования подобных шрифтов, особенно в основном тексте, является трудность чтения.

4) моноширинный шрифт отличается единой фиксированной шириной всех символов. Эти шрифты часто используются для вывода программного кода.

**Выбора шрифта.** В 1878 году профессор Эмиль Джаваль (Emile Javal) из парижского университета установил, что взгляд при чтении не движется плавно вдоль линии текста, а совершает серию прыжков, которые он назвал *саккадическими прыжками*. Засечки на знаках корректируют движение взгляда читателя, придавая ему горизонтальное направление, а отсутствие засечек способствует вертикальному направлению взгляда, т.е. перескакиванию между строками. Более того, существуют исследования, которые показывают, что читатели предпочитают тексты, написанные шрифтом с засечками, называя их более разборчивыми.

**Контраст и цвет.** Цветовой акцент в тексте может сделать весь дизайн чрезвычайно привлекательным. Размещение двух цветов с низким значением контраста рядом друг с другом сделает текст очень трудным для чтения.

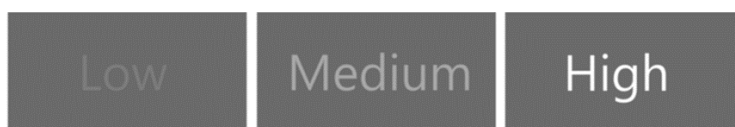


Рисунок 13 – Контраст фона и текста

Особое внимание следует уделять соблюдению наивысшего контраста текста: располагать темный текст на светлом фоне, или наоборот – светлый текст на темном фоне. Однако для больших кусков текста лучше остановиться на традиционном сочетании темного цвета для шрифта и светлого

для страницы. Нежелательно использовать яркие фоновые заливки и графические обои, затрудняющие восприятие текстовой информации. Если все же они используются, то в этом случае рекомендуется применять двуцветные шрифты (черные символы с белой окантовкой).

1) Поля для ввода, например, поле для поиска или авторизации. У текстовых полей может быть до семи-восьми вариантов отображения: обычный, при наведении, при нажатии, в момент ввода, после ввода, при ошибке и т.д.

2) Кнопки и ссылки, которые обычно представлены минимум в трех вариантах: в обычное время, в момент наведения, в момент нажатия, в ситуациях, когда элемент неактивен (например, когда кнопку нельзя нажать).

3) Иконки – графические элементы, необходимые для создания управляющих и информационных компонентов, например, кликабельное изображение «Корзины» в интернет-магазинах.

4) Элементы навигации, хедеры и футеры также могут быть включены в UI Kit, т.к. они содержат типовые элементы и часто выглядят шаблонно (например, футеры почти всегда содержат список контактов и вспомогательный блок навигации).

5) Выпадающие меню, всплывающие окна, поле поиска и другие шаблонные компоненты. В UI Kit включают набор элементов навигации для мобильных и десктопных версий (верхние и боковые панели, сэндвич-панели).

6) Виджеты – интерактивные блоки, которые отображают часто обновляемую информацию, например, погоду, курс валют и пр.

### **Лабораторная работа 5. Разработка UI Kit**

**Задание.** Разработать UI Kit для Вашего приложения (см. лабораторная работа 1). Лабораторная работа должна быть выполнена с помощью онлайн-сервиса [figma.com](https://www.figma.com).

Элементы, которые обязательно должны быть отображены:

- основные цвета;
- логотип;
- типографика (заголовки всех уровней, основной текст, ссылки, списки);
- элементы форм (поля ввода, кнопки, выпадающие списки, переключатели);
- различные состояния для полей ввода, кнопок и ссылок;
- иконки;
- управляющие элементы.

### **Контрольные вопросы**

- 1) В чём заключается этап определения стилистики интерфейса?
- 2) Чем отличается концепция Material Design от Flat design?
- 3) Что такое UI Kit?
- 4) Что включает в себя UI Kit?

## 2.6 Этап разработки дизайн концепции

Дизайн концепция призвана показать будущий вид всего приложения. Если предыдущий этап определения стилистики только дал направление, то дизайн концепция призвана объединить выбранное направление с имеющимся содержанием интерфейса. Дизайн концепция может быть представлена любым объемом, но его стараются минимизировать для экономии времени. Обычно концепция представлена 1 – 3 экранами интерфейса. Если речь идет о веб-приложении (или сайте), то стараются показать вид одной и той же страницы для нескольких устройств.

## 2.7 Этап оформления всех экранов

Когда очередь доходит до оформления всех экранов, становится ясно правильно ли подобран кегль, хорошо ли сочетается толщина линий иконок с текстом, не конфликтует ли оформление форм (кнопок, полей ввода) с другими элементами экрана и т.д.

Планом для оформления всех экранов являются структура и схематичный прототип интерфейса, но бывают случаи отхождения от этого плана. Например, на этом этапе может выясниться, что всплывающее окно будет намного нагляднее и эффективнее, чем разъезжающийся блок информации посреди экрана.

## Лабораторная работа 6. Вёрстка

**Задание.** Сверстать разработанные макеты дизайна из лабораторной работы 4 с учётом карты экранов (из лабораторной работы 3) и UI Kit (из лабораторной работы 5). Вёрстка должна быть адаптивной, т.е. корректно отображаться на мобильном телефоне, планшете и мониторе компьютера или ноутбука. Минимальная ширина устройства – 440px, максимальная – 1920px.

## ЛИТЕРАТУРА

1. Тидвелл Дж. Разработка пользовательских интерфейсов. 2-е изд. – СПб.: Питер, 2011. – 480с.
2. Купер А., Рейман Р., Кронин Д. Интерфейс. Основы проектирования взаимодействия. – Пер.с англ. – СПб.: Символ-Плюс, 2009. – 688с.
3. Уильямс Робин. Дизайн. Книга для недизайнеров. 4-е изд. – СПб.: Питер, 2016. – 240с.
4. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. – СПб.: Символ-Плюс, 2004. – 272 с.
5. Унгер Р., Чендлер К. UX-дизайн. Практическое руководство по проектированию опыта взаимодействия. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 336 с.