

## **Лабораторная работа №7**

### **Построение вайрфрейма и простого кликабельного прототипа бизнес-продукта**

**Цель работы:** Познакомиться с прототипированием и такими понятиями, как вайрфрейм, прототип, мокап. Спроектировать вайрфрейм и простой кликабельный прототип продукта.

### **Краткая теория**

**Прототипирование** — это процесс создание интерактивного опыта, который включает создание прототипа на основе разработанных мокапов или вайрфреймов.

В UX/UI-дизайне существует три основных типа структурных схем страниц:

- Вайрфрейм (Wireframe) — отражает структуру продукта.
- Мокап (макет) — отражает визуальную концепцию продукта.
- Прототип — отражает весь динамический функционал продукта.

### **Вайрфрейм**

**Вайрфрейм (каркас, блочная схема)** — это образ продукта низкой точности, с упором на структуру и содержание, схематичное изображение, набросок пользовательского интерфейса будущего продукта. Это первый шаг в процессе прототипирования.

Виды вайрфреймов:

1. Вайрфреймы низкой точности (Low-fidelity). Напоминает грубый набросок или быстрый макет, каркасы низкой точности имеют меньше деталей и быстро создаются. Они более абстрактны, используют прямоугольники и маркировку для представления содержимого приложения. Содержат фиктивный контент, контуры элементов, какие-то образцы текста или символический контент. Например, вместо реальных изображений можно использовать прямоугольник-заполнитель. Пример вайрфрейма низкой точности представлен на рисунке 1.

2. Вайрфреймы высокой точности (High-fidelity). Они включают уровень детализации, который более точно соответствует дизайну фактической деятельности приложения. Вайрфреймы высокого качества включают в себя более реальный контент, оперативный выбор типографики и информацию о размерах изображения. Пример вайрфрейма высокой точности представлен на рисунке 2.

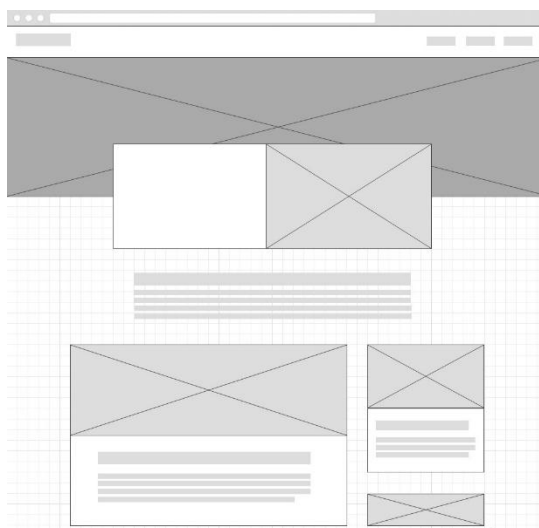


Рисунок 1 – Пример вайрфрейма низкой точности

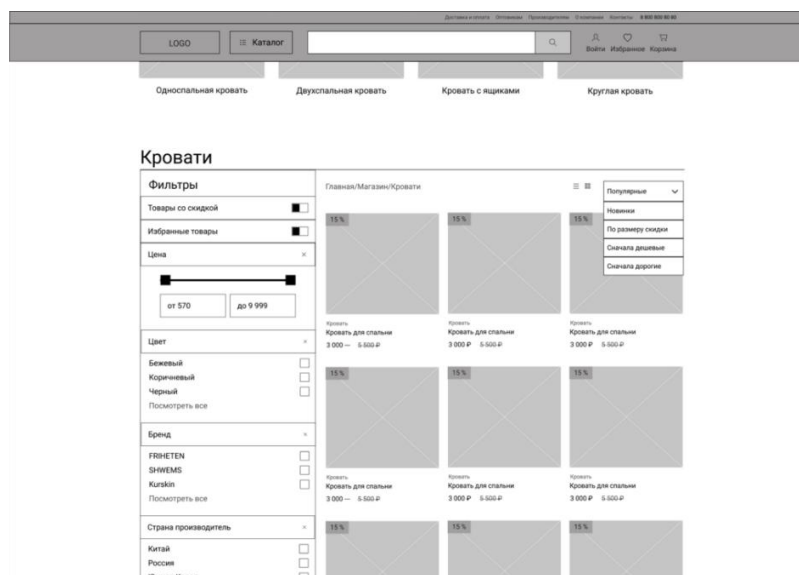


Рисунок 2 – Пример вайрфрейма высокой точности

Рекомендации по созданию вайрфреймов:

- Использовать минимальное количество цветов: чёрный, серый, белый. Этих цветов хватит.
- Использовать стандартные шрифты.
- Не использовать полноцветные изображения.
- Все элементы должны быть простые и аккуратные.

**Вайрфрейм** — чёрно-белый подробный план экрана/страницы продукта. Здесь намечается расположение элементов: кнопок, изображений, текстов.

Пример вайрфрейма низкой (слева) и высокой (справа) точности показан на рисунке 3.

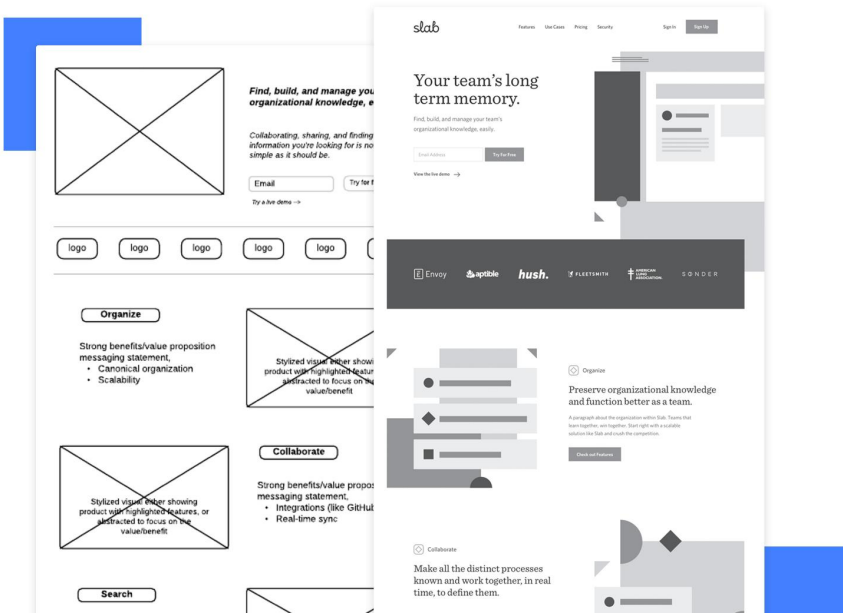


Рисунок 3 – Пример вайрфрейма низкой и высокой точности

Вайрфрейм может состоять из базовых элементов: прямоугольник, эллипс, линии и так далее. Пример основных элементов вайрфрейма представлен на рисунке 4.




Элемент	Назначение
	для структурных блоков, изображений
	иконки, изображения
	изображения
<div>Текст</div> <div>Текст</div>	кнопка с текстом внутри, который соответствует действию кнопки
<input checked="" type="checkbox"/>	checkbox
<input type="radio"/>	radio button

Рисунок 4 – Пример основных элементов вайрфрейма

## Прототип

**Прототип** — это средне или высоко детализированное представление конечного продукта, которое имитирует взаимодействие пользователя с интерфейсом. Прототипы всегда интерактивные.

Прототип необходим для проверки логики функционала. Это симуляция взаимодействия между пользователем и интерфейсом. В прототипах присутствует функционал, и пользователь может выполнять некоторые действия: переходить между экранами/страницами продукта, нажимать на кнопки, открывать модальные окна (обычно это просто создание кликабельности в редакторе для прототипирования).

Может добавлять интерактивности к вайрфреймам или к макетам. В отличие от того, будет определяться его степень детализации. Обычно прототип добавляется уже поверх вайрфрейма и будет являться серединой на пути к высококачественному изображению финального продукта.

Таким образом под прототипом понимается динамическая интерактивная модель продукта.

Пример простого кликабельного прототипа представлен на рисунке 5.

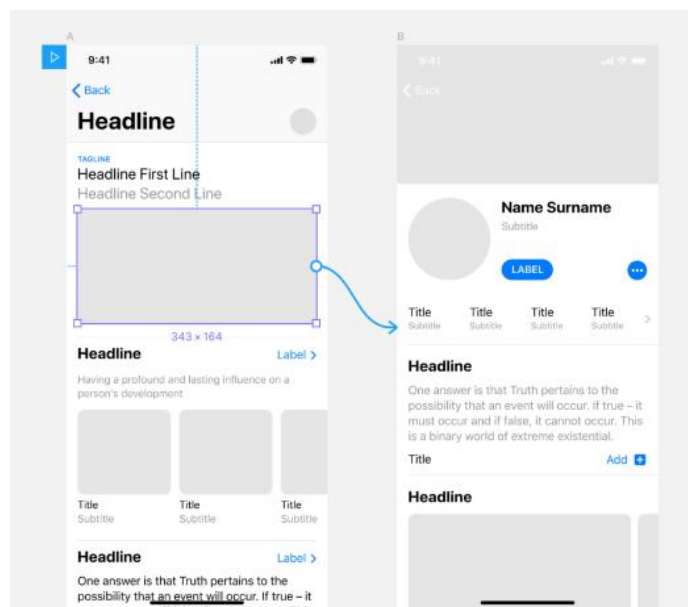


Рисунок 5 – Пример простого кликабельного прототипа

## Мокап (Mockup)

Мокап (mockup или mock-up — макет) является полноцветным и детализированным наброском дизайна. Это живописный аналог вайрфрейма. В нем уже присутствует полная палитра оттенков, конкретные изображения элементов (кнопок, иконок, шапки сайта), фон страниц, текстовые шрифты и прочие графические элементы. Часто мокапы это актуальный графический дизайн.

Мокап является масштабируемой или полноразмерной дизайн-концепцией продукта. Акцент в нем делается на визуальную составляющую. Дизайн здесь накладывается на утвержденный ранее каркас вайрфрейма.

Мокап демонстрирует:

- структуру информации;
- расположение и визуализацию контентных блоков;
- промежуточные этапы создание продукта;
- как будет выглядеть конечный продукт;
- основной функционал в статике.

Пример мокапа мобильного приложения показан на рисунке 6.

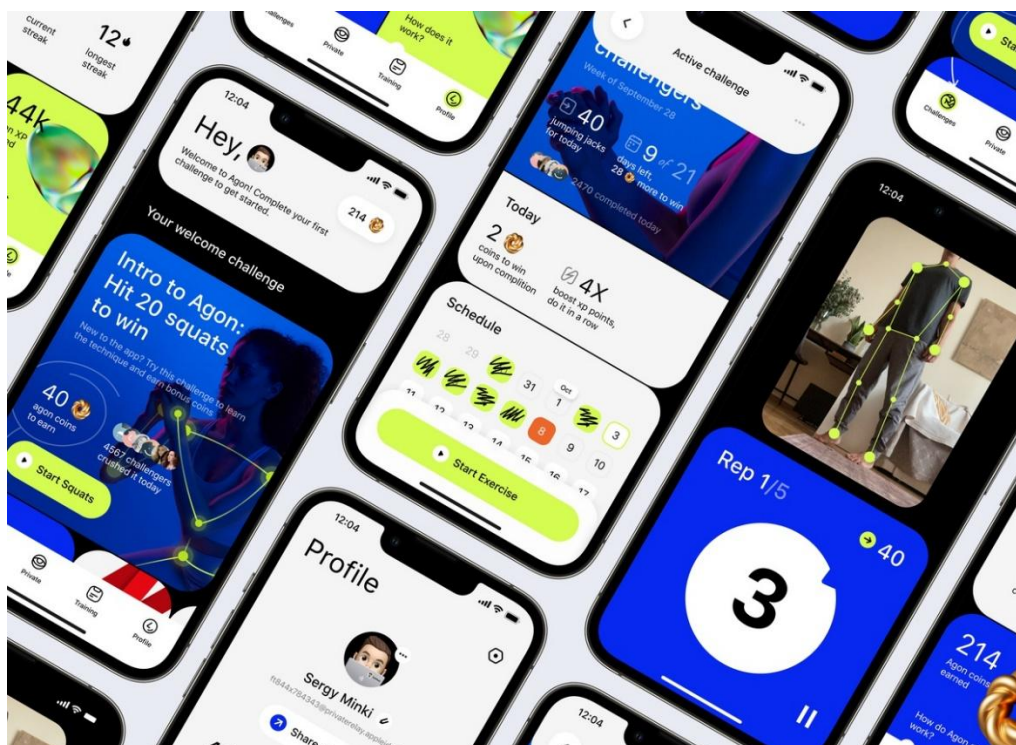


Рисунок 6 – Пример мокапа (макета) мобильного приложения

## Правила чистого UI-дизайна

Необходимо придерживаться нескольких правил для того, чтобы создать хороший UI:

1. Простые шрифты. Использование простых и минималистичных шрифтов сделает тексты и весь дизайн более читабельными. Пример представлен на рисунке 7 (шрифты в первой колонке лучше использовать минимально раз для каких-то крупных заголовков).

Oxygen Mono  
ALGERIAN  
*Pallaya*  
Gigi



Inter  
Montserrat  
Proxima Nova  
Poppins  
Roboto  
Rubik



Рисунок 7 – Пример правила использования простых шрифтов

2. Необходимо выравнивать элементы. Правильные отступы и интервалы создают чистый и упорядоченный макет, что упрощает и ускоряет восприятие информации. Пример представлен на рисунке 8.

#### Заказать образцы цветов

Выберите цвет и нужный размер. Заполните форму с указанием адреса доставки, куда мы сможем отправить Вам образцы.

Отправить



Все элементы выровнены  
по-разному

#### Заказать образцы цветов

Выберите цвет и нужный размер. Заполните форму с указанием адреса доставки, куда мы сможем отправить Вам образцы.

Отправить



Все элементы выровнены  
одинаково

Рисунок 8 – Пример выравнивания элементов

3. Не использовать грязные тени. Рекомендуется использовать более светлые оттенки чёрного, чтобы убрать с тени напряжение и сильный контраст. Пример показан на рисунке 9.



Не используем очень тёмные  
и контрастные тени



Используем более светлые  
и менее контрастные оттенки

Рисунок 9 – Пример правила использования теней

4. Больше свободного пространства. Необходимо предоставлять элементам больше свободного пространства для того, чтобы они смогли дышать. Это способствует облегчению навигации. Пример представлен на рисунке 10.

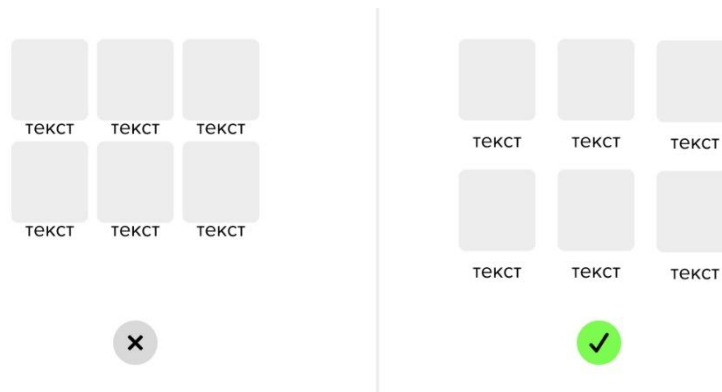


Рисунок 10 – Пример правила использования свободного пространства

5. Использование нужных и полезных элементов. Ограничение количества элементов в одном пространственном блоке повышает концентрацию пользователя на необходимом. Также необходимо использовать правила композиции при группировке элементов. Пример представлен на рисунке 11.



Рисунок 11 – Пример правильного использования элементов

6. Более простой дизайн. В большинстве случаев наличие большого количества эффектов и стилей портит пользовательский интерфейс. Рекомендуется не делать слишком много эффектов, например, так как тени, градиенты, жирные обводки элементов. Пример показан на рисунке 12.

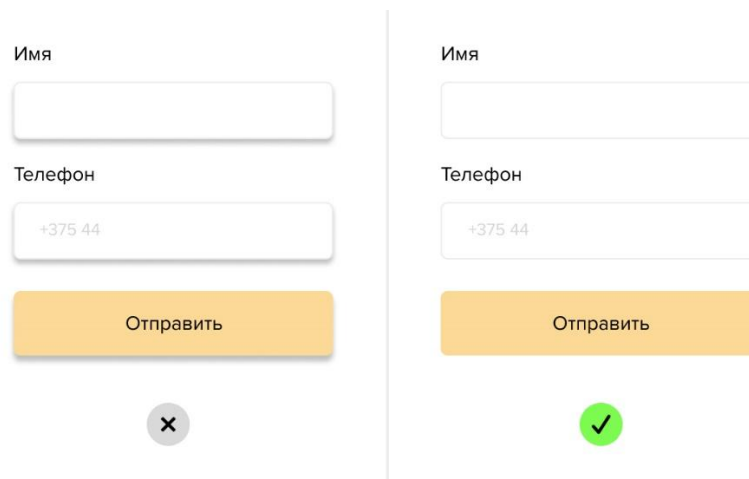


Рисунок 12 – Пример правильного использования эффектов

7. Баланс и выравнивание. Необходимо всегда помнить, что у всех элементов в интерфейсе есть вес. Вес можно определить по цвету, размеру, текстуре. Необходимо соблюдать баланс, правила композиции и правильно выравнивать элементы.

8. Быть последовательным и соблюдать единообразие в интерфейсе. Согласованный единообразный пользовательский интерфейс подразумевает использование одинаковых шаблонов проектирования, одинаковые элементы, отступы, а также согласованные команды во всём интерфейсе. Пример представлен на рисунке 13.

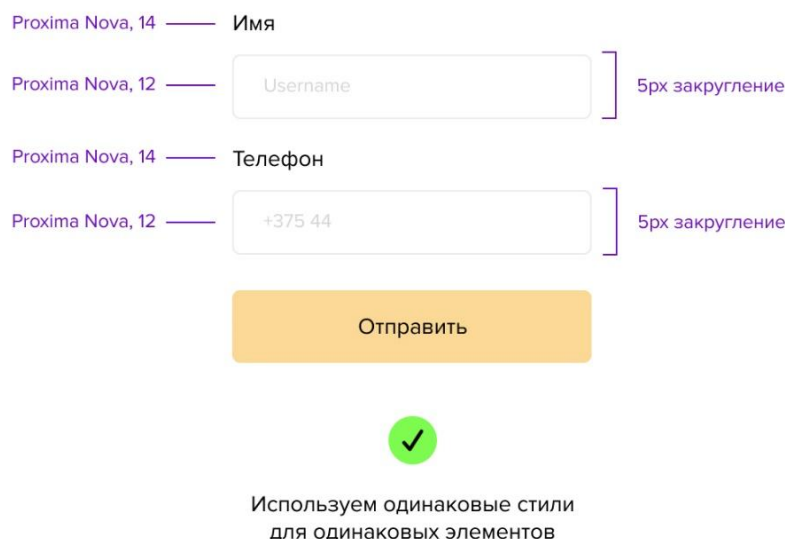


Рисунок 13 – Пример единообразного интерфейса

9. Единая система шрифтов и начертаний. Необходимо использовать одну систему шрифтов, начертаний, толщины для основного текста, заголовков, подзаголовков и так далее. Даже небольшая разница в размере между двумя типами шрифта может исказить контрасты и иерархию.



10. Читаемые заголовки. Необходимо делать заголовки более акцентными и контрастными, для того, чтобы направить взгляд пользователя от наиболее важных вещей к менее важным. Пример показан на рисунке 14.

The image shows two versions of a form side-by-side, separated by a vertical line. Both forms have the same text: 'Заказать образцы' (Order samples), 'Выберите цвет и нужный размер. Заполните форму с указанием адреса доставки, куда мы сможем отправить Вам образцы.' (Select color and size. Fill out the form with delivery address), and an 'Отправить' (Send) button. The left version has a grey 'x' icon below the button, indicating it is incorrect. The right version has a green checkmark icon below the button, indicating it is correct. The key difference is the header 'Заказать образцы', which is bold and black in the right version but regular weight in the left version.

Рисунок 14 – Пример создания читаемых заголовков

11. Визуальная иерархия элементов. Строгая визуальная иерархия — основной принцип успешного пользовательского интерфейса. Состоит в расположении элементов интерфейса таким образом, чтобы объяснить уровень важности каждого элемента и направить пользователей к выполнению желаемого действия. Пример представлен на рисунке 15 (дизайн не путает пользователя и создаётся иерархия, в которой есть призыв к целевому действию).

The image shows two versions of a registration form side-by-side, separated by a vertical line. Both forms have fields for 'Имя' (Name) and 'Телефон' (Phone) with a placeholder '+375 44'. Below the fields are two buttons: 'Отправить' (Send) and 'Назад' (Back). The left version has both buttons in the same orange color, with a grey 'x' icon below them. The right version has the 'Отправить' button in orange and the 'Назад' button in light grey, with a green checkmark icon below them. Below the forms, text explains that in the left design, buttons of the same color can confuse the user, while in the right design, the primary button's color encourages the user to act.

Рисунок 15 – Пример создания правильной иерархии

12. Делать различными placeholders, labels и введенные пользователем данные. Пример представлен на рисунке 16.

Рисунок 16 – Пример создания элементов формы

13. Использовать одну систему отступов/интервалов. Рекомендуется использовать одну систему отступов. Вокруг логических блоков должны быть установлены отступы одинакового размера (например, вверху и внизу, а также слева и справа). Пример представлен на рисунке 17.

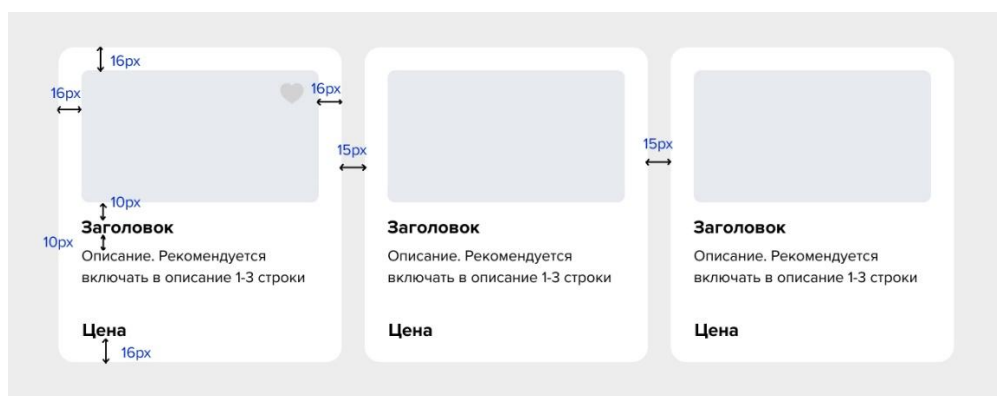


Рисунок 17 – Пример использования единой системы интервалов

14. Связанные элементы должны быть расположены ближе друг к другу. И наоборот, несвязанные элементы должны быть расположены подальше друг от друга. Пример представлен на рисунке 18.

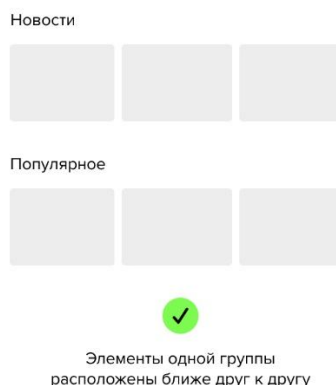


Рисунок 18 – Пример связанных элементов

15. Элементы одного «типа» должны быть расположены на одинаковом расстоянии друг от друга. Пример представлен на рисунке 19.



Рисунок 19 – Пример расположения элементов одного «типа»

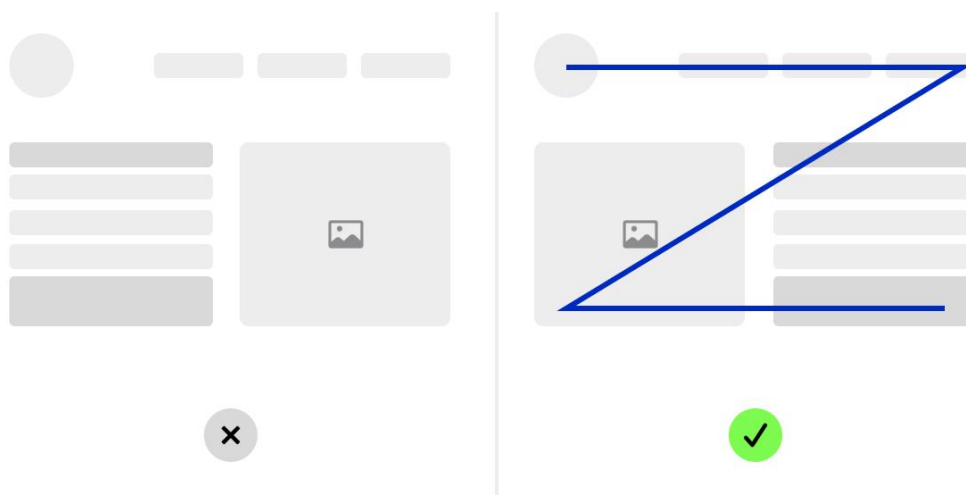
16. Более крупные элементы должны иметь больше свободного пространства вокруг себя.

17. Пространство между различными логическими блоками должно быть больше, чем пространство между заголовком и текстом внутри каждого блока.

18. Рекомендуется использовать 3 размера шрифта: для заголовка, подзаголовка и основного текста. При помощи разного размера, толщины и стиля шрифтов, можно направить внимание пользователя к важным элементам.

19. Более важные элементы рекомендуется увеличивать, а менее важные – уменьшать. Главное, чтобы крупных элементов на один компонент было не больше двух.

20. Не забывать про паттерны сканирования. Необходимо помнить, что существуют две главные траектории, по которым человек считывает информацию — F-паттерн и Z-паттерн. Ключевой контент должен располагаться именно по этим маршрутам. Пример показан на рисунке 20.



## Рисунок 20 – Пример использования Z-паттерн

21. Не использовать placeholder в качестве label. Пример представлен на рисунке 21.

The image shows two examples of email input fields. On the left, a placeholder 'Введите email' is used as the label, which is marked with a red 'X' indicating it is incorrect. On the right, a separate label 'Email' is placed above the input field containing 'bsty@gmail.com', which is marked with a green checkmark indicating it is the correct approach.

## Рисунок 21 – Пример правильного использования placeholder и label

22. Добавлять свободное пространство кнопке. Пример показан на рисунке 22.



## Рисунок 22 – Пример правильно спроектированной кнопки

23. Рекомендуется использовать формы, состоящие из одной колонки пример показан на рисунке 23, либо, в котором две колонки, но порядок размещения полей идёт по горизонтали, пример показан на рисунке 24.

The image compares two form layouts. The left layout is a two-column form with fields for 'Имя' and 'Фамилия' in the first row, and 'Email' and 'Телефон' in the second row, followed by an 'Отправить' button. It is marked with a red 'X'. The right layout is a single-column form with fields for 'Имя', 'Фамилия', and 'Email' stacked vertically, followed by an 'Отправить' button. It is marked with a green checkmark, indicating it is the recommended design.

## Рисунок 23 – Пример формы, состоящей из одной колонки

Имя

Фамилия

Email

Телефон

Город

Адрес

Рисунок 24 – Пример формы, где поля размещены по горизонтали

24. Рекомендуется не использовать символы только верхнего регистра в кнопках. Пример показан на рисунке 25.



Рисунок 25 – Пример правильно оформленного текста внутри кнопки

25. Основной контент страницы/экрана должен располагаться в пределах сетки и быть выровнен по ней. Пример показан на рисунке 26.

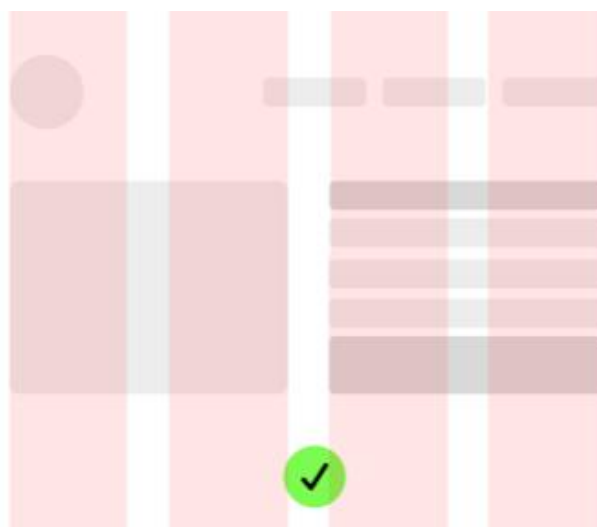


Рисунок 26 – Пример выравнивания контента внутри сетки

## Карточка

**Карточка** — это компонент интерфейса, который визуально обозначает группу логически связанных между собой элементов. Обобщённая структура карточки представлена на рисунке 27.

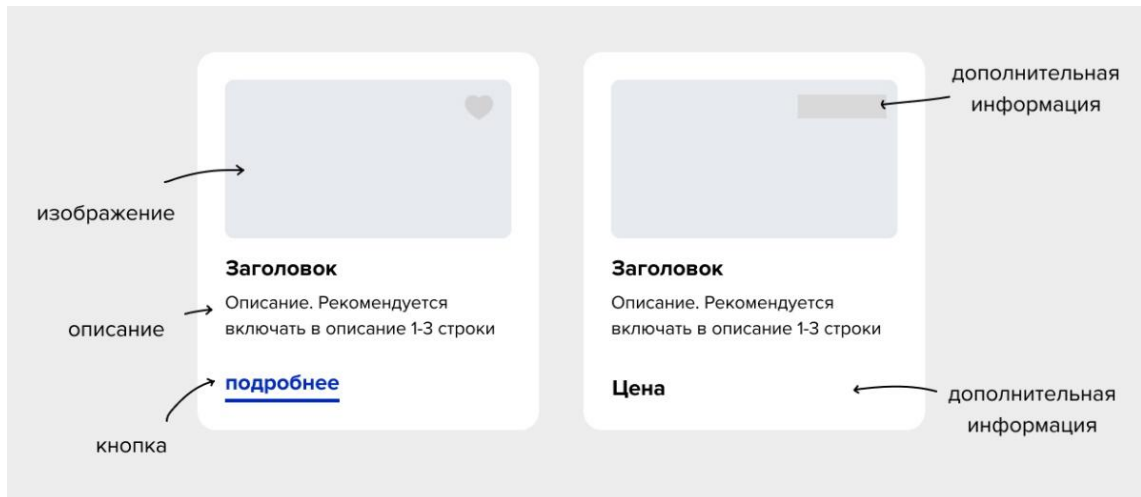


Рисунок 27 – Обобщённая структура карточки

### Правила составления карточек:

1. Необходимо обеспечивать хороший контраст между карточкой и фоном. Можно сделать карточку и фон разных цветов, или добавить на карточку рамку или тень.

2. Правильно сбалансировать размеры шрифтов. Для повышения читабельности контента и создания более понятной карточки, необходимо правильно выбирать размеры шрифтов. Совет по выбору размеров шрифта:

- Для заголовков рекомендуется использовать шрифт размером от 18px до 96px, в зависимости от размеров карточки.
- Подзаголовок должен быть на 2-10px меньше, чем заголовок.
- Для основного текста рекомендуется использовать размер шрифта не менее 14px.

3. Использовать одну систему отступов. Лучше использовать размер отступов, который будет кратен 4px. Необходимо быть последовательным и использовать одну систему отступов на всей карточке. Слева и справа карточки должны быть одинаковые отступы, сверху и снизу одинаковые отступы, между элементами должны быть одинаковые отступы и так далее. Пример показан на рисунке 28.

4. Фиксированная высота. Карточки должны быть с фиксированной высотой и усечённым текстом и не должны менять свой размер в зависимости от количества контента.

5. Ограничить контент внутри карточки. Рекомендуется не перегружать информацию в карточке. Необходимо предоставить только ту информацию, которая сможет дать пользователю представления о том, захочет ли он

взаимодействовать в дальнейшем с тем, что презентует карточка, или нет. Пример показан на рисунке 29.

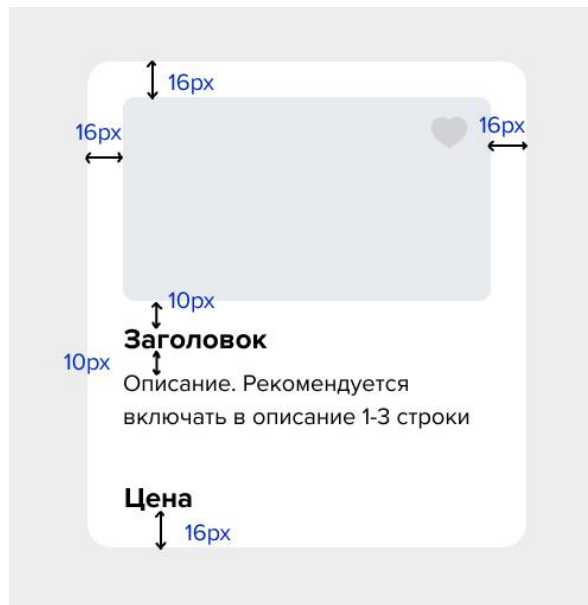


Рисунок 28 – Пример единой системы отступов

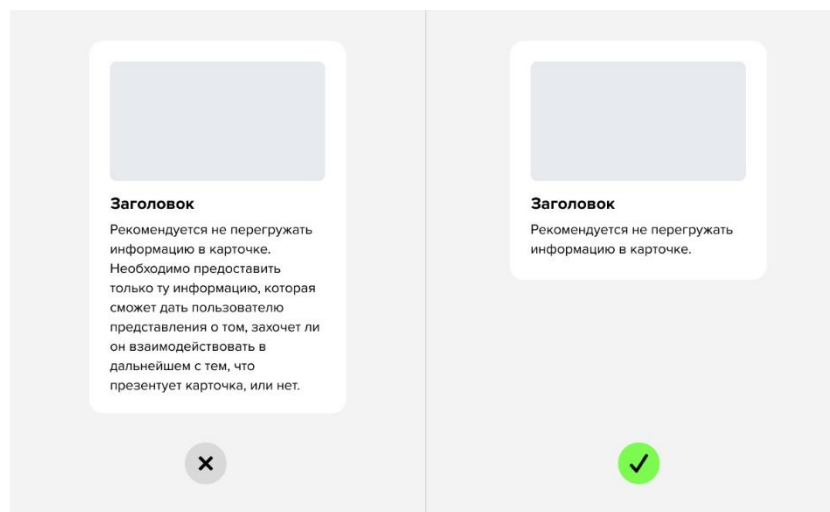


Рисунок 29 – Пример ограничения контента внутри карточки

6. Тестировать содержимое карточки. Необходимо всегда проверять содержимое карточки при разработке. Рекомендуется создавать карточки в соответствии с тем, что мы не знаем точно из сколько строк будет состоять заголовок или подзаголовок, и лучше сразу это предусмотреть, чтобы затем не нарушать иерархию. Пример показан на рисунке 30.

7. Не нужно использовать везде линии. Не используйте разделительные линии там, где можно просто пробелом отделить элементы друг от друга. Пример представлен на рисунке 31.

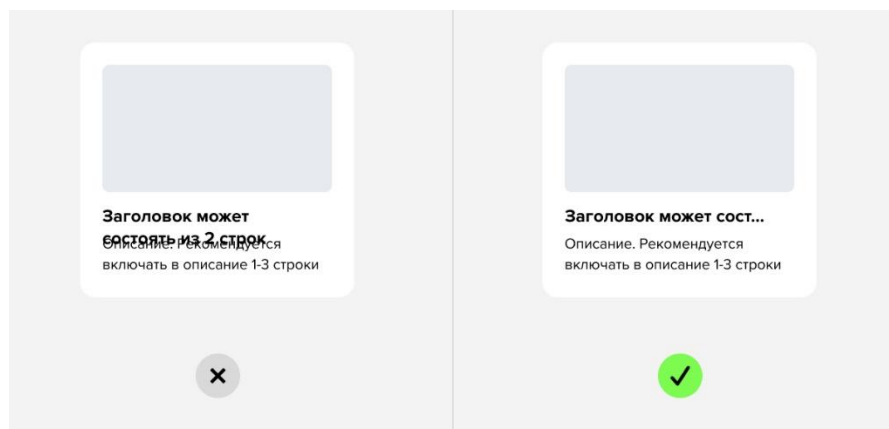


Рисунок 30 – Пример правильного тестирования карточки

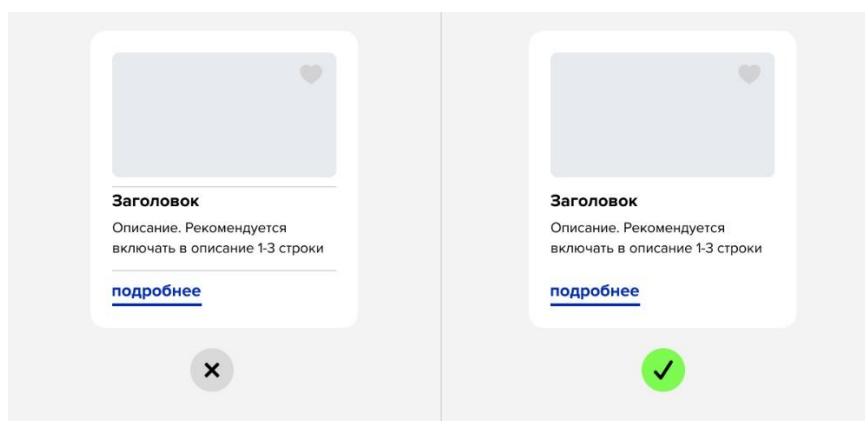
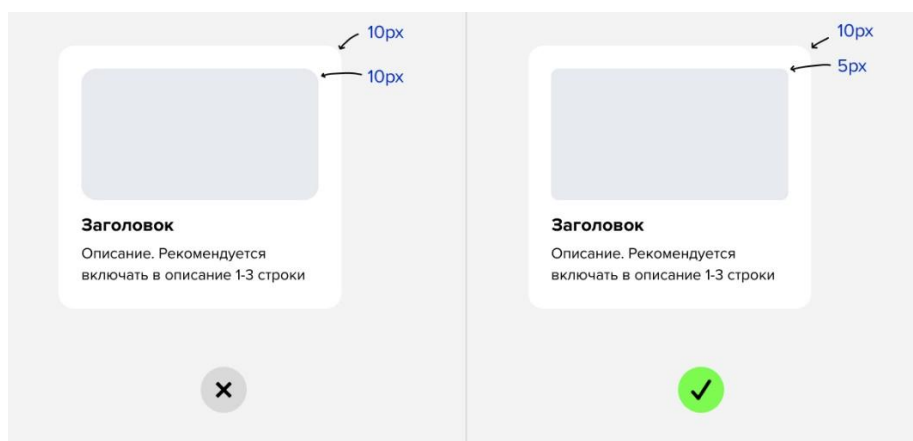


Рисунок 31 – Пример правильного использования пробелов

8. Ограничить текстовые ссылки. Рекомендуется не использовать ссылки в контенте, который вызывает большую когнитивную нагрузку.

9. Угловой радиус. Рекомендуется не использовать тот же угловой радиус внутри карточки (угловой радиус изображений, кнопок), что и для внешнего радиуса самой карточки. Правильно будет использовать половину основного радиуса в качестве радиуса изображения. Пример показан на рисунке 32.





## Рисунок 32 – Пример правильного использования пробелов

### Задание к лабораторной работе №5

1. Обязательно ИЗУЧИТЬ все правила чистого UI-дизайна.
2. Выбрать редактор для прототипирования, можно использовать Figma и установить плагин для прототипирования (например, Wireframe, там есть много готовых элементов для схематичного отображения), InVision, Miro, Webflow.
3. Создать фреймы нужного размера. Для разработки необходимо выбрать десктопный фрейм размером 1920px.
4. При помощи сетки и направляющих определить и разграничить рабочую область (рабочая область — та часть фрейма, где будет располагаться основной контент, без полей слева и справа). Например, если размер фрейма 1920px, то ширина рабочей области может быть 1400px.
5. На основе выявленного функционала, разработанных user flow, правил чистого UI-дизайна создать вайрфрейм всего продукта (по теме, выбранной в лабораторной работе №2) средней детализации.
6. Поверх созданного вайрфрейма создать простой кликабельный прототип (без UI, мокап в данной лабораторной работе не создаётся).

### Контрольные вопросы к лабораторной работе №5

1. Что такое прототипирование?
2. Что такое вайрфрейм?
3. Какие есть виды вайрфреймов?
4. Для чего необходимы вайрфреймы?
5. Что такое прототип?
6. Для чего необходим прототип?
7. Чем прототип отличается от вайфрейма?
8. Что такое мокап (макет)?
9. Перечислите правила чистого UI-дизайна.