**1 Анализ задачи**

**1.1 Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

Тема проекта: “маркетплейс «BelAli»”.

Целью нашего проекта является информирование, удобное впитывание полезной информации.

**1.1.2 Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

**Гость:**

1. просмотр страниц сайта

2. просмотр карточки товаров

3. просмотр английской версии сайта

4. просмотр рекламы

5. авторизация

**Пользователь:** все те же функции, что может выполнять гость, но также появляется:

**1.1.3 Описание входной, выходной и условно-постоянной информации**

Таблица 1-Функции программы с описанием с входной, выходной и условно-постоянной информации

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | № Категория пользователей | Наименование процесса | | | Краткое описание алгоритм  выполнения  процесса | Входная  информация | Выходная информация | Условно -  постоянная  информация |
| Продолжение таблицы 1 | | |  |  |  |  |  |  |
| 1 | 2 | 3 | | | 4 | 5 | 6 | 7 |
| 1 | Гость | Просмотр  страниц сайта | | | Любому  пользователю  Разрешение заходить и просматривать  информацию на  сайте | Отсутствует | Отсутствует | Данные сайта |
| 2 | Гость | Просмотр карточки товаров | | | Открыть главную страницу сайта | Главная страница | Карточка | Информация карточки |
| 3 | Гость | Просмотр Английской версии сайта | | | Нажимаем на ссылку ru/en | Выбор ссылки ru/en | Перевод страницы на английскую версию | Страница на английской версииы |
| 5 | Гость | Просмотр меню авторизации | | | Зайти на сайт меню | Авторизация | Переход на главную | Мнею авторизации |

**1.1.4 Нефункциональные требования**

**Требования к применению:**

Интерфейс маркетплэйса должен быть легким, понятным, функциональным и простым в использовании. Он не должен перегружать пользователя ненужной информацией.

**Требования к производительности:**

Требования к производительности наименьшие. Для повышения производительности веб-ресурса нужно иметь хорошее подключение к интернету.

**Требования к реализации:**

Для реализации данного программного продукта будет использоваться сервис Tilda.

**Требования к надёжности:**

Веб-сайт должен иметь возможность самовосстановления после сбоя.

**Требования к интерфейсу:**

При разработке интернет-ресурса использованы красные и чёрные тона по указанию заказчика. Основные разделы сайта должна быть доступны с первой страницы. Должен быть разработан грамотный пользовательский интерфейс.

Для обеспечения удобного использования веб-ресурса, изображения и символы, размещенные на экране, должны быть достаточно крупными и не должны перекрывать другие функциональные элементы. Все страницы сайта должны быть адаптивными для разных устройств и разрешений экрана, чтобы обеспечить его удобное использование независимо от местоположения пользователя.

**1.2 Выбор стратегии разработки и модели жизненного цикла**

Для разработки веб-ресурса следует выбрать стратегию разработки и модель жизненного цикла. Выбор модели жизненного цикла на основе характеристик требований находится в таблице 1.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V- образную, 5 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является RAD модель и инкрементная модель.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

        Вычисления: 5 за каскадную, 5 за V-образную, 4 за RAD, 5 за инкрементную, 2 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная, V-образная и инкрементная модели.

Таблица 5 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 0 за каскадную, 0 за V-образную, 2 за RAD, 1 за инкрементную, 4 за быстрого прототипирования и 2 за эволюционную.

Итог: На основе результатов заполнения табл. 5 подходящей является модель быстрого проектирования.

Таблица 6 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | | Каскадная | | V-образная | | RAD | | Инкрементная | | Быстрого прототипирования | | Эволюционная | |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | | Нет | | Нет | | Нет | | Да | | Да | | Да | |
| 2. | Будет ли проект являться расширением существующей системы? | | Да | | Да | | Да | | Да | | Нет | | Нет | |
| 3. | Будет ли проект крупно- или среднемасштабным? | | Нет | | Нет | | Нет | | Да | | Да | | Да | |
| 4. | Ожидается ли длительная эксплуатация продукта? | | Да | | Да | | Нет | | Да | | Нет | | Да | |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | | Нет | | Да | | Нет | | Да | | Нет | | Да | |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | | Нет | | Нет | | Нет | | Да | | Да | | Да | |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | | Нет | | Нет | | Нет | | Да | | Да | | Да | |
| 8. | Является ли график сжатым? | | Нет | | Нет | | Да | | Да | | Да | | Да | |
| 9. | Предполагается ли повторное использование компонентов? | | Нет | | Нет | | Да | | Да | | Да | | Да | |
|  |  | |  | |  | |  | |  | |  | |  | |
| Продолжение таблицы 6 | |  | |  | |  | |  | |  | |  | |  | |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | | Нет | | Нет | | Нет | | Нет | | Да | | Да | |

Вычисления: 3 за каскадную, 4 за V-образную, 4 за RAD, 8 за инкрементную, 6 за быстрого прототипирования и 8 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является инкрементная и эволюционная модели.

Общий итог: в итоге заполнения табл. 3 – 6 наиболее подходящей является инкрементная модель.

**1.3 Инструменты разработки**

Инструменты, используемые при разработке и написании сопутствующей документации:

Figma – будет использоваться для создания UX/UI макетов проекта

Web-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;

Microsoft Office Word – для написания документации к программному продукту;

Microsoft PowerPoint — программа подготовки презентаций и просмотра презентаций, являющаяся частью Microsoft Office.

Разработка проекта будет происходить на компьютере со следующими параметрами:

процессор: Intel Core i3-10110U CPU 2.10GHz 2.59 GHz

объем оперативной памяти: 8 GB;

объем места на жёстком диске: 256 GB;

видеокарта: Intel graphics

**2. Проектирование задачи**

**2.1 Разработка структуры сайта, системы меню, навигации**

Для успеха сайта очень важно его проектирование, так как оно влияет на то, как пользователи взаимодействуют с сайтом и насколько он эффективен. Качественный сайт должен быть простым в использовании, иметь удобную навигацию, быстро загружаться и удовлетворять запросы пользователей.

**2.2 Разработка UML-диаграмм**

Смотреть приложение А.

**3. Реализация**

**3.1 Руководство программиста**

**Руководство программиста**

Программный продукт разработан с использованием таких языков: «Html» и «Css». Данные языки позволяют создавать полноценные веб-сайты, разберёмся с каждым компонентом по подробнее.

Html является языком разметки, который определяет структуру веб-страницы. Он используется для создания элементов, таких как:

**Сематические теги:**

<header></header>;

<main></main>;

<footer></footer>;

**Параграфы:**

<p>Ferz <br> Снуд</p>;

<p>31,00</p>;

<p>Adidas <br> Рюкзак LIN</p>;

**Изображения:**

<img src="img/App store.svg">**;**

<img src="img/Instagram.svg">;

**Ссылки:**

<a href="../Html/private\_office.html">Личный кабинет</a>;

<a href="index-en.html">RU/EN</a>;

CSS представляет собой язык стилей, который управляет внешним видом элементов, созданных с использованием «Html». С его помощью:

**Задаются цвета**:

background-color: black;

color: #FFFEFE;

background-color: rgb(5, 5, 119);

**Шрифты:**

font-size: 30px;

font-family: 'Inter', sans-serif;

**Расположение элементов:**

display: flex;

flex-direction: column;

justify-content: space-around;

**Анимации:**

a:hover{

    color: white;

}

Вместе эти две технологии формируют основу веб-разработки, предоставляя средства для создания структуры «Html», стилей «Css» на веб-страницах. Этот двойной подход обеспечивает возможность создания современных веб-приложений, способных к разнообразным и мощным функциональностям.

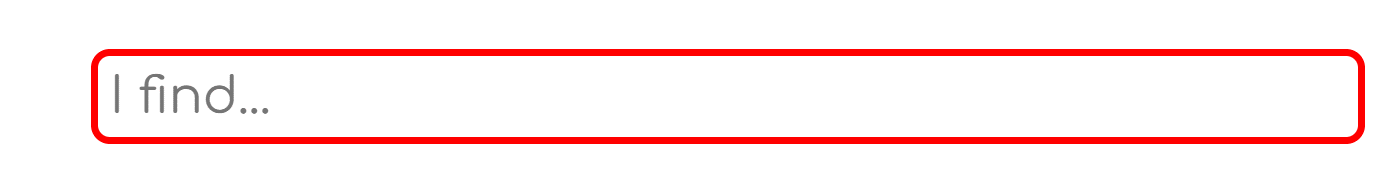
В разработке веб-приложений широко используются компоненты. Формы и вводные компоненты, включая **поля ввода** (рисунок 1)

Рисунок 1 - Поле ввода

Изображение выглядит как текст, снимок экрана, Шрифт, белый

Автоматически созданное описание**Кнопки** (рисунок 2)

Рисунок 2 - Кнопка

Изображение выглядит как текст, одежда, снимок экрана, рукав

Автоматически созданное описание Собирают данные от пользователей. Компоненты содержимого, такие как **карточки** (рисунок 3)

Рисунок 3 - Карточка товара

Изображение выглядит как аквариум, снимок экрана, риф

Автоматически созданное описание**Окна регистрации** (рисунок 4)

Рисунок 4 - Форма регистрации

Изображение выглядит как текст, снимок экрана, Шрифт, символ

Автоматически созданное описаниеПредоставляют информацию и взаимодействие. **Графические элементы** (рисунок 5)

Рисунок 5 - Графический элемент

Все они являются основными строительными блоками для разработки веб-приложений.

Этот компонентный подход упрощает разработку, делает код более читаемым и обеспечивает повторное использование, создавая тем самым современные и привлекательные веб-приложения.

**4 Тестирование**

**4.1 Тесты на использование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результате тестирования предоставлен в таблице 7.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест | | Ожидаемый результат | Физический результат | Результат  тестирования |
| 1 | 2 | | 3 | 4 | 5 |
| 1 | Ввод значений в поле «Email» | | Ввод значений | Значения были введены | Выполнено |
| 2 | Ввод значений в поле «Пароль» и что бы значения были не видны | | Значения вводятся, и они скрыты | Значения ввелись и их не видно | Выполнено |
| 3 | Проверка меню чек-бокс «Запомнить» | | Окно должно активироваться при нажатии и пароль запомнился браузером | Окно активировался и пароль сохранился | Выполнено |
| 4 | Проверка кнопка «Войти» | | Переход на главную форму | Переход осуществился | Выполнено |
| 5 | Проверка поля ввода «Я ищу…» | | Возможность вводить данные | Значения ввелись | Выполнено |
| 6 | Переход по ссылке «Instagram» | | Переход на страницу | Переход осуществился | Выполнено |
|  |  | |  |  |  |
| Продолжение таблицы 7 | |  |  |  |  |
| 7 | Переход по ссылке «RU/EN» | | Переход на форму с английским языком | Страница перевелась на английский язык | Выполнено |
| 8 | Адаптивная страница | | Страница должна подстраиваться под основные виды экрана | Страница адаптировалась | Выполнено |

**4.2 Отчёт о результатах тестирования**

При разработке программного продукта было решено множество проблем, например, не осуществлялся переход к следующей странице по  нажатию кнопки или открывалось окно с формой, поэтому в будущем пользователь не столкнется с  данными проблемами.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

**5. Руководство пользователя**

Цель данного проекта заключается в создании макет плэйса с интуитивно понятным интерфейсом, его описанием и структурированностью.

Создаваемый сайт будет рассчитан на любого рода пользователей, которые хотят приобрести товар.

Для того, чтобы открыть сайт необходимо открыть в редакторе код «main.html».

Изображение выглядит как аквариум, снимок экрана, риф

Автоматически созданное описаниеПосле открытия сайта мы видим форму входа (рисунок 6).

Рисунок 6 – Форма авторизации

Изображение выглядит как текст, снимок экрана, Веб-сайт, Реклама в Интернете

Автоматически созданное описаниеПосле входа на сайт нас приветствует главная форма (рисунок 7).

Рисунок 7 – Главная форма

Изображение выглядит как текст, одежда, шляпа

Автоматически созданное описаниеИзображение выглядит как текст, одежда, снимок экрана, Футболка

Автоматически созданное описаниеПри наведение на карточку товара у на появляется подсветка и появляются hover эффекты (рисунок 8).

Рисунок 8 – Главная форма сайта

Изображение выглядит как текст, снимок экрана, Веб-сайт, Реклама в Интернете

Автоматически созданное описаниеПри нажатии на ссылку «RU/EN» (рисунок 9) страница переводится на английский язык (рисунок 10)

Рисунок 9 - Кнопка «RU/EN»

Изображение выглядит как текст, снимок экрана, мода

Автоматически созданное описание

Рисунок 10 - Перевод страницы на английскую версию

Изображение выглядит как Шрифт, логотип, Графика, символ

Автоматически созданное описаниеПри нажатии на ссылку «Instagram» (рисунок 11) открывается страница с Instagram (рисунок 12)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описаниеРисунок 11 - Ссылка «Instagram»

Рисунок 12 - Переход на страницу