Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

**ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

**РАЗРАБОТКА САЙТА-ПРИЛОЖЕНИЯ РИЭЛТОРСКОГО АГЕНТСТВА**

Отчет по производственной практике

|  |
| --- |
| Руководители  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |
| Студент группы 2993  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |

# Содержание

[Раздел 1. Общая (теоретическая) часть 5](#_Toc170640850)

[1.1 Постановка задачи (ТЗ) 5](#_Toc170640851)

[1.2 Обоснование проектных решений 7](#_Toc170640852)

[1.3 Обзор существующих программных систем 9](#_Toc170640857)

[Раздел 2. Практическая составляющая 10](#_Toc170640858)

[2.1 Реализация 10](#_Toc170640859)

[2.2 Тестирование 11](#_Toc170640863)

[Заключение 20](#_Toc170640864)

[Список литературы 20](#_Toc170640865)

[Приложение А 20](#_Toc170640866)

[Приложение Б 20](#_Toc170640867)

[Приложение В 24](#_Toc170640868)

# Аннотация

Этот проект направлен на разработку комплексного веб-приложения для агентства недвижимости. Приложение позволит пользователям просматривать доступную к покупке или аренде недвижимость.

Введение

В современном цифровом мире наличие удобного и функционального веб-сайта является важным инструментом для успеха любого бизнеса. Агентство недвижимости, работающее в условиях высокой конкуренции, не является исключением. Интернет-присутствие позволяет значительно расширить клиентскую базу, предлагая пользователям удобный способ поиска и ознакомления с доступными объектами недвижимости.

Цель данного проекта заключалась в создании веб-сайта для агентства недвижимости, который предоставляет пользователям возможность легко и быстро находить информацию о доступной к покупке недвижимости. В ходе работы над проектом мы стремились обеспечить интуитивно понятный интерфейс, высокую скорость загрузки страниц и адаптивный дизайн для различных устройств.

Создание веб-сайта направлено на решение нескольких ключевых задач:

* Обеспечение удобного и быстрого доступа к информации о недвижимости;
* Обеспечение интуитивно понятного интерфейса для пользователей;
* Гарантия высокой скорости загрузки страниц и адаптивного дизайна для различных устройств.

Основные функциональные возможности сайта:

* Поиск недвижимости по типу.
* Подробные описания объектов с фотографиями и характеристиками.

В отчете подробно описаны этапы разработки веб-сайта: от анализа требований до тестирования и запуска проекта. Также рассмотрены используемые технологии, инструменты и методики, обеспечившие достижение поставленных целей.

# Раздел 1. Общая (теоретическая) часть

## Постановка задачи (ТЗ)

Основные функции

1. Регистрация и авторизация пользователей

Требования:

* Форма регистрации нового пользователя с указанием имени, электронной почты и пароля.
* Форма авторизации для уже зарегистрированных пользователей.
* Восстановление пароля через электронную почту.

2. Вывод недвижимости из базы данных

Требования:

* Отображение списка объектов недвижимости, хранящихся в базе данных.
* Вывод основной информации о каждом объекте: тип недвижимости, адрес, цена, площадь, фотографии.
* Пагинация для удобного просмотра большого количества объектов.

3. Пользовательская фильтрация отображаемой недвижимости по типу

Требования:

* Возможность фильтрации списка недвижимости по типу: коттедж, дом, квартира.

4. Отдельный вход для администраторов

Требования:

* Проверка учетных данных администратора и предоставление доступа к панели администраторов.

5. Панель администраторов для управления недвижимостью в базе данных

Требования:

* Интерфейс для добавления нового объекта недвижимости, включая все необходимые поля (тип, адрес, цена, площадь, фотографии и т.д.).
* Интерфейс для редактирования существующих объектов недвижимости.
* Возможность удаления объектов недвижимости из базы данных.
* Просмотр списка всех объектов недвижимости с возможностью поиска и фильтрации.

Технические требования

Веб-сайт должен быть разработан с использованием современных веб-технологий (HTML, CSS, JavaScript).

Серверная часть должна быть реализована на языке программирования, поддерживающем работу с базами данных (например, Python, PHP, Node.js).

Использование реляционной базы данных (например, MySQL, PostgreSQL) для хранения информации о недвижимости и пользователях.

Адаптивный дизайн для корректного отображения на различных устройствах (ПК, планшеты, смартфоны).

Завершение

Данный проект должен быть завершен в течение 3 месяцев с даты начала работ. Все этапы разработки (анализ требований, проектирование, разработка, тестирование, запуск) должны быть выполнены с учетом стандартов качества и безопасности.

## 1.2 Обоснование проектных решений

Дизайн, ориентированный на пользователя: Основной целью проекта является создание удобной платформы, которая учитывает потребности и предпочтения своих пользователей. Проводя тщательные исследования пользователей, включая опросы, интервью и юзабилити-тестирование, мы получили представление о конкретных требованиях и ожиданиях людей.

Масштабируемость и производительность: Предвидя потенциальный рост числа пользователей в будущем и расширение функциональных возможностей, проектные решения будут уделять приоритетное внимание масштабируемости и оптимизации производительности. Это предполагает использование масштабируемой инфраструктуры, эффективных методов программирования и регулярного тестирования производительности для обеспечения бесперебойной работы при различных нагрузках.

## 1.2.1 Обоснование выбора языков программирования

Выбор языков программирования зависит от многих факторов, например, задачи, которые нужно решить. Выбор языков программирования для разработки веб-приложения обусловлен следующими факторами:

HTML (язык гипертекстовой разметки):

Основа веб-страниц: HTML обеспечивает базовую структуру и содержание интерфейс. Он определяет такие элементы, как заголовки, абзацы, формы и кнопки, которые составляют макет. Семантические элементы HTML обеспечивают доступность приложения для пользователей с ограниченными возможностями и устройств чтения с экрана. Правильная структура HTML помогает поисковым системам понимать и индексировать содержимое приложения.

CSS (каскадные таблицы стилей):

CSS управляет внешним видом и функциональностью. Он стилизует элементы, определяет макеты и создавет визуально привлекательные интерфейсы. Персонализирует пользовательский опыт. Реализует принципы отзывчивого дизайна, чтобы веб-приложение хорошо смотрелось на различных устройствах (настольных компьютерах, планшетах, мобильных телефонах).

JavaScript:

JavaScript добавляет динамическое поведение. Он позволяет реализовать такие функции, как валидация пользовательского ввода, регистрация настроения, обработка данных и интерактивные визуализации. Он взаимодействует с API и внутренними службами для хранения и получения данных о настроении.

## 1.2.2 Инструментальные средства

Для разработки приложения использовались следующие инструментальные средства:

* Visual Studio Code - кроссплатформенная интегрированная среда разработки (IDE), обеспечивающая удобный редактор кода, отладчик и множество полезных расширений для веб-разработки. Она предоставляет широкий набор инструментов для отладки, рефакторинга кода и автоматизации задач.
* Git - распространенная система контроля версий, позволяющая отслеживать изменения в коде, производить откаты и совместную работу над проектом.
* Node.js - программная платформа для выполнения JavaScript-кода на сервере, необходимая для работы с различными инструментами и библиотеками.
* Figma - онлайн-редактор дизайна, который позволяет создавать макеты и прототипы веб-сайтов, приложений и других пользовательских интерфейсов.
* npm - менеджер пакетов для Node.js, позволяющий устанавливать и управлять зависимостями проекта.

## 1.2.3 Обоснование выбора среды программирования

Браузер был выбран в качестве среды выполнения для данного веб-приложения, так как он предоставляет все необходимые возможности для отображения и выполнения HTML, CSS и JavaScript. Современные веб-браузеры поддерживают новейшие стандарты и технологии, что делает их идеальной средой для разработки и запуска веб-приложений.

## 1.2.4 Информационное обеспечение

Для разработки логики работы веб-приложения используется язык программирования JavaScript.

## 1.3 Обзор существующих программных систем

На рынке существует ряд веб-приложений для отслеживания эмоций. Некоторые из наиболее популярных и широко используемых:

Таблица конкурентов

|  |  |  |
| --- | --- | --- |
| Название | Ссылка | Преимущество |
| «Циан» | <https://www.cian.ru/> | Сервис отличается большой базой партнеров-агентств и риэлторов |
| «Северная долина» | sevdol.ru | Агентство отличается предоставлением ипотеки по материнскому капиталу |
| «D-Apart» | <https://d-apart.ru/> | Веб-приложение предоставляет возможность в кратчайшие сроки выкупить квартиру. |

Преимущества разрабатываемого приложения заключаются в том, что оно легкое, быстро загружаемое и адаптируемое по дизайну, что позволяет использовать его на различных устройствах и при медленном интернет-соединении. Еще одним преимуществом является возможность гибкой настройки приложения в соответствии с потребностями пользователя.

# Раздел 2. Практическая составляющая

## 2.1 Реализация

## 2.1.1 Выбор технических и программных средств

Для разработки приложения прогноза погоды использовались следующие технические и программные средства:

* Операционная система: Windows 11
* Браузер: Opera GX (для тестирования и отладки)
* IDE: Visual Studio Code
* Система контроля версий: Git
* Инструменты разработки: Node.js, npm (менеджер пакетов для Node.js)

## 2.1.2 Практическая реализация алгоритмов

Алгоритм - точное предписание, описывающее порядок действий (план действий), определяющих процесс решения задачи или достижения цели.

Основным алгоритмом, реализованным в приложении, является отображение недвижимости, хранящейся в базе данных:

1. Формирование запроса к базе данных для получения списка недвижимости;
2. Преобразование данных к удобному для работы формату;
3. Ввод полученных данных в конструкцию HTML с помощью php.

## 2.1.3 Используемые методы

1. Получение данных: Код использует манипуляции с php для получения и хранения в базе введенных пользователем данных;
2. Отображение данных: Полученные данные отображаются на веб-странице с помощью php. Код обновляет различные элементы HTML, такие как список отображаемой недвижимости;
3. Обработка событий: Код обрабатывает различные события, такие как отправка формы регистрации, нажатие на кнопки настроек, выборка фильтров при просмотре недвижимости.

Код демонстрирует различные аспекты веб-разработки, включая манипуляции с DOM, обработку событий, использование базы данных и реализацию различных функций для обработки данных и отображения информации на веб-странице.

## 2.2 Тестирование

В современном цифровом ландшафте хорошо функционирующий веб-сайт имеет решающее значение для процветания предприятий и организаций. Однако создание веб-сайта - только половина дела; не менее важно обеспечить его качество и функциональность путем тщательного тестирования. Тестирование веб-сайта - систематический процесс оценки производительности, удобства использования, безопасности и доступности веб-сайта, чтобы убедиться, что он соответствует ожиданиям пользователей и бизнес-целям.

Значение тестирования веб-сайтов

Тестирование веб-сайтов - это не просто вариант, а необходимость по нескольким веским причинам:

Улучшение пользовательского опыта (UX): Успех веб-сайта зависит от его способности обеспечить положительный пользовательский опыт. Тестирование помогает выявить проблемы с юзабилити, навигацией и неработающими функциями, которые препятствуют удовлетворению пользователей. Решив эти проблемы, вы создадите беспроблемный и приятный опыт, что приведет к повышению вовлеченности пользователей и конверсии.

Раннее обнаружение ошибок: Ошибки и дефекты неизбежны при разработке программного обеспечения, но их раннее обнаружение имеет решающее значение для предотвращения дорогостоящих последствий. Тестирование выявляет ошибки на разных этапах, что позволяет своевременно устранить их до того, как они повлияют на пользователей и нанесут ущерб репутации сайта.

Повышение производительности и надежности: Тестирование производительности оценивает такие факторы, как время загрузки страниц, отзывчивость и стабильность при интенсивном трафике. Это помогает обеспечить оптимальное функционирование сайта, предотвратить сбои и замедления, которые расстраивают пользователей и отталкивают их.

Повышенная безопасность: Тестирование безопасности выявляет уязвимости, которыми могут воспользоваться злоумышленники, например, SQL-инъекции, межсайтовый скриптинг или утечка данных. Устранение этих уязвимостей позволяет защитить данные пользователей и целостность веб-сайта.

Доступность для всех: Тестирование на доступность гарантирует, что сайт будет удобен для людей с ограниченными возможностями, в соответствии со стандартами доступности, такими как WCAG. Это способствует инклюзивности и расширяет охват веб-сайта для более широкой аудитории.

Виды тестирования веб-сайтов

Тестирование веб-сайтов включает в себя различные типы, каждый из которых фокусируется на определенных аспектах:

Функциональное тестирование: этот основной тип тестирования проверяет, что все функции и возможности работают так, как задумано, в соответствии со спецификациями. Оно включает в себя тестирование регистрации пользователей, входа в систему, ввода данных, поиска и других основных процессов.

Тестирование удобства использования: этот тип тестирования оценивает простоту использования, интуитивность и общий пользовательский опыт. Такие методы, как наблюдение за пользователями, опросы и A/B-тестирование, помогают выявить области для улучшения и обеспечить удобство использования сайта.

Тестирование совместимости: убедиться в том, что сайт корректно отображается и функционирует в различных браузерах, операционных системах, устройствах и разрешениях экрана.

Тестирование производительности: оценивается отзывчивость, стабильность и масштабируемость сайта при различных условиях нагрузки. При этом измеряются такие показатели, как время загрузки страницы, использование ресурсов и одновременная работа пользователей.

Тестирование безопасности: выявление уязвимостей и слабых мест, которые могут быть использованы злоумышленниками. Оно включает в себя тестирование на проникновение, сканирование уязвимостей и аудит безопасности, чтобы обеспечить защиту пользовательских данных и целостности сайта. Тестирование приложения проводилось на различных устройствах и в разных браузерах (Opera GX, Mozilla Firefox, Microsoft Edge) для проверки корректности отображения и функционирования. Были выявлены и исправлены следующие проблемы:

Таблица 2

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_1 | Расшифровка: TestCase\_MoodTracker\_1 |
| Приоритет тестирования | Высокий |  |
| Заголовок/название теста | Проверка ошибки в Email |  |
| Краткое изложение теста | Ввести в форму авторизации некорректные данные для поля Email(незарегистрированные) |  |
| Этапы теста | * 1. Зайти на главную страницу сайта;   2. Нажать левой кнопкой мыши по Кляксе в середине экрана   3. Внести ложные данные в поле Email   4. Внести данные в поле Password   5. Отправляем форму нажатием левой кнопки мыши по кнопке «Sign In» |  |
| Тестовые данные | Ложные данные для поля Email: [321232@yandex.com](mailto:321232@yandex.com)  Данные для поля Password: 4421 |  |
| Ожидаемый результат | Уведомление «email [321232@yandex.com](mailto:321232@yandex.com) has not been registered» |  |
| Фактический результат | Уведомление «email [321232@yandex.com](mailto:321232@yandex.com) has not been registered» |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | После выполнения теста поле должно быть снова очищено для дальнейшего тестирования |  |

Таблица 3

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_2 | Расшифровка: TestCase\_MoodTracker\_2 |
| Приоритет тестирования | Высокий |  |
| Заголовок/название теста | Проверка ошибки в Password |  |
| Краткое изложение теста | Ввести в форму авторизации некорректные данные для поля Email(незарегистрированные) |  |
| Этапы теста | 1. Зайти на главную страницу сайта; 2. Нажать левой кнопкой мыши по Кляксе в середине экрана 3. Внести данные в поле Email 4. Внести ложные данные в поле Password   Отправляем форму нажатием левой кнопки мыши по кнопке «Sign In» |  |
| Тестовые данные | Данные для поля Email: [karina.slake.koo@gmail.](mailto:karina.slake.koo@gmail.)com  Данные для поля Password: 3212 |  |
| Ожидаемый результат | Уведомление «Incorrect password for email karina.slake.koo@gmail.com» |  |
| Фактический результат | Уведомление «Incorrect password for email karina.slake.koo@gmail.com» |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | После выполнения теста поле должно быть снова очищено для дальнейшего тестирования |  |

Таблица 4

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_3 | Расшифровка: TestCase\_MoodTracker\_3 |
| Приоритет тестирования | Высокий | Сильно влияет на пользовательский опыт, а также не позволяет новым желающим зарегистрироваться в сервисе |
| Заголовок/название теста | Проверка работы кнопки «first time» из авторизации в регистрацию |  |
| Краткое изложение теста | Проверка работоспособности функции переворачивания карточки, если пользователь не зарегистрирован |  |
| Этапы теста | * 1. Зайти на главную страницу сайта;   2. Нажать левой кнопкой мыши по Кляксе в середине экрана;   3. Нажать левой кнопкой мыши по надписи «first time» под карточкой авторизации |  |
| Тестовые данные | - |  |
| Ожидаемый результат | Происходит анимация переворачивания карточки |  |
| Фактический результат | Происходит анимация переворачивания карточки |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | Перезагрузить страницу для последующего тестирования |  |

Таблица 5

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_4 | Расшифровка: TestCase\_MoodTracker\_4 |
| Приоритет тестирования | Средний |  |
| Заголовок/название теста | Проверка работы кнопки «first time» из регистрации в авторизацию |  |
| Краткое изложение теста | Проверка работоспособности функции переворачивания карточки, если пользователь только что зарегистрировался и хочет войти в сервис |  |
| Этапы теста | * + 1. Зайти на главную страницу сайта;     2. Нажать левой кнопкой мыши по Кляксе в середине экрана;     3. Нажать левой кнопкой мыши на надпись «first time»;     4. Ввести тестовые данные в поля регистрации;     5. Нажать левой кнопкой мыши по кнопке «Sign Up»;     6. Нажать левой кнопкой мыши по надписи «already registered» |  |
| Тестовые данные | Email: [karina.slake.koo@gmail.com](mailto:karina.slake.koo@gmail.com)  Password: 32122111  Confirm Password: 32122111 |  |
| Ожидаемый результат | Происходит анимация переворачивания карточки |  |
| Фактический результат | Происходит анимация переворачивания карточки |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | Перезагрузить страницу для последующего тестирования |  |

Таблица 6

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_5 | Расшифровка: TestCase\_MoodTracker\_5 |
| Приоритет тестирования | Высокий |  |
| Заголовок/название теста | Проверка формы регистрации |  |
| Краткое изложение теста | Проверка сохранения данных формы регистрации в локальном хранилище |  |
| Этапы теста | * + - 1. Зайти на главную страницу сайта;       2. Нажать левой кнопкой мыши по Кляксе в середине экрана;       3. Нажать левой кнопкой мыши на надпись «first time»;       4. Ввести тестовые данные в поля регистрации       5. Нажать левой кнопкой мыши по кнопке «Sign Up»;       6. Проверить локальное хранилище на наличие новых записей |  |
| Тестовые данные | Email: [moonl@yandex.ru](mailto:moonl@yandex.ru)  Password: m00nly123  Confirm password: m00nly123 |  |
| Ожидаемый результат | Наличие следующих записей в локальном хранилище:  newUserEmail: [moonl@yandex.ru](mailto:moonl@yandex.ru)  newUserPassword: m00nly123 |  |
| Фактический результат | Наличие следующих записей в локальном хранилище:  newUserEmail: [moonl@yandex.ru](mailto:moonl@yandex.ru)  newUserPassword: m00nly123 |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | Перезагрузить страницу для последующего тестирования |  |

Таблица 7

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_6 | Расшифровка: TestCase\_MoodTracker\_6 |
| Приоритет тестирования | Средний |  |
| Заголовок/название теста | Проверка формы регистрации на вывод ошибок |  |
| Краткое изложение теста | Проверка вывода ошибки при попытке зарегистрировать аккаунт с паролем короче 8 символов |  |
| Этапы теста | * + - * 1. Зайти на главную страницу сайта;         2. Нажать левой кнопкой мыши по Кляксе в середине экрана;         3. Нажать левой кнопкой мыши по надписи «first time»;         4. Ввести тестовые данные в поля регистрации;         5. Нажать левой кнопкой мыши по кнопке «Sign Up» |  |
| Тестовые данные | Email: [moonl@yandex.ru](mailto:moonl@yandex.ru)  Password: R213  Confirm password: R213 |  |
| Ожидаемый результат | Уведомление «Password is shorter than 8 characters» |  |
| Фактический результат | Уведомление «Password is shorter than 8 characters» |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | - |  |

Таблица 8

|  |  |  |
| --- | --- | --- |
|  |  | Комментарии |
| Тест № | TC\_MT\_7 | Расшифровка: TestCase\_MoodTracker\_7 |
| Приоритет тестирования | Средний |  |
| Заголовок/название теста | Проверка формы регистрации на вывод ошибок |  |
| Краткое изложение теста | Проверка вывода ошибки при попытке зарегистрировать аккаунт вводя несовпадающие пароли |  |
| Этапы теста | * + - * 1. Зайти на главную страницу сайта;         2. Нажать левой кнопкой мыши по Кляксе в середине экрана;         3. Нажать левой кнопкой мыши по надписи «first time»;         4. Ввести тестовые данные в поля регистрации;         5. Нажать левой кнопкой мыши по кнопке «Sign Up» |  |
| Тестовые данные | Email: [moonl@yandex.ru](mailto:moonl@yandex.ru)  Password: R2134  Confirm password: R2136 |  |
| Ожидаемый результат | Уведомление «Passwords don't match» |  |
| Фактический результат | Уведомление «Passwords don't match» |  |
| Статус | Выполнено |  |
| Предварительное условие | Запустить файл index.html из корневой папки проекта с помощью браузера Opera GX |  |
| Постусловие | - |  |

# Заключение

В рамках этого проекта было разработано полнофункциональное веб-приложение для риэлторского агентства, позволяющее пользователю ознакомиться с актуальными предложениями по продаже и аренде недвижимости.

Планируется дальнейшее развитие и совершенствование приложения.

# Список литературы

1. Документация HTML: <https://developer.mozilla.org/ru/docs/Web/HTML>
2. Руководство по JavaScript: <https://developer.mozilla.org/ru/docs/Web/JavaScript>

# Приложение А

# Приложение Б

Тестируемый код в п. 2.2:

let blob = document.getElementById("blob");

let textsWelcome = document.querySelectorAll(".textOnWelcome");

let signUpForm = document.querySelector("#main");

let card = document.getElementById('card');

let swapCard = document.getElementById('formToSignUp');

blob.addEventListener('click', () => {

textsWelcome.forEach(element => {

element.classList.add('fade-out')

});

signUpForm.setAttribute('style', 'z-index: 2');

blob.setAttribute('style', 'width: 35vw; height: 60vh; animation: none; border-radius: 50px;');

blob.classList.add('no-pseudo');

let active = document.querySelectorAll('.fade-in');

active.forEach((el) => {

el.classList.add('active')

});

});

swapCard.addEventListener('click', () => {

if(card.classList.contains('container') !== false){

card.classList.remove('container');

swapCard.innerText = 'first time';

swapCard.classList.remove('backTitle');

swapCard.setAttribute('style', 'transform: rotateY(0)');

blob.setAttribute('style', 'width: 35vw; height: 60vh; animation: none; border-radius: 50px;');

} else {

swapCard.innerText = 'already registered';

swapCard.setAttribute('style', 'transform: rotateY(180deg)');

card.classList.add('container');

swapCard.classList.add('backTitle');

blob.setAttribute('style', 'width: 35vw; height: 70vh; animation: none; border-radius: 50px;');

};

})

let btn\_reg = document.getElementById("btn\_reg");

function registerUser(){

email = document.getElementById('signUpEmail').value;

password = document.getElementById('signUpPassword').value;

confPassword = document.getElementById('signUpConfirmPassword').value;

if(password == confPassword && password.length >= 7) {

localStorage.setItem('newUserEmail', email);

localStorage.setItem('newUserPassword', password);

}

else if(password != confPassword) alert(`Passwords don't match`);

else if(password.length <= 7) alert(`Password is shorter than 8 characters`)

}

let btn\_log = document.getElementById("btn\_log");

function loginUser(){

email = document.getElementById('signInEmail').value;

password = document.getElementById('signInPassword').value;

const registeredUser = localStorage.getItem("newUserEmail");

const registeredPassword = localStorage.getItem("newUserPassword");

const validUser = email == registeredUser;

const validPassword = password == registeredPassword;

if(validUser && validPassword){

console.log('u did it');

document.location.replace('pages/recEmotion.html');

}

else if(!validUser) alert(`email ${email} has not been registered. `);

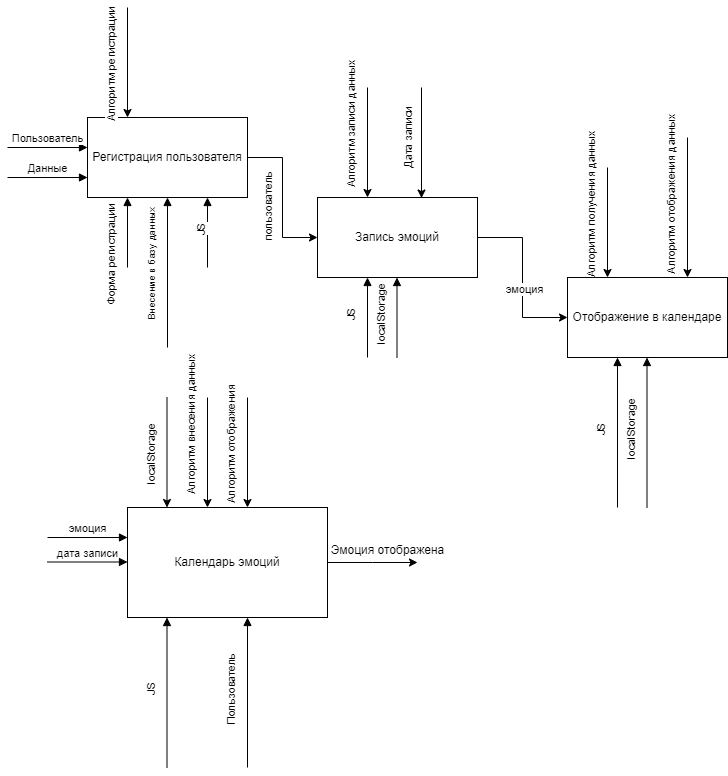
else if(!validPassword) alert(`Incorrect password for email ${email}`);

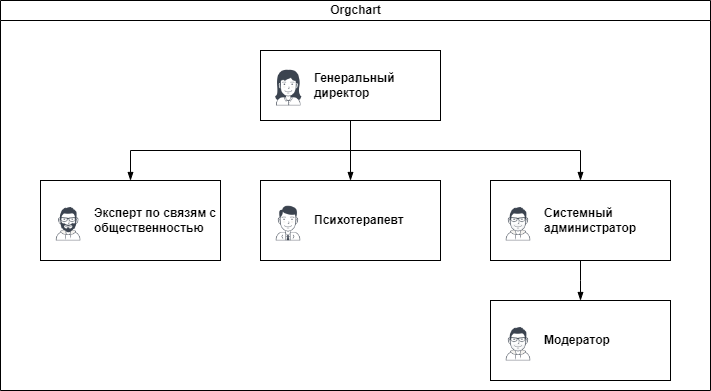
}

btn\_log.addEventListener('click', () => loginUser());

btn\_reg.addEventListener('click', () => registerUser());

# Приложение В

ДиаграммыIDEF0 и декомпозиция

Организационная диаграмма

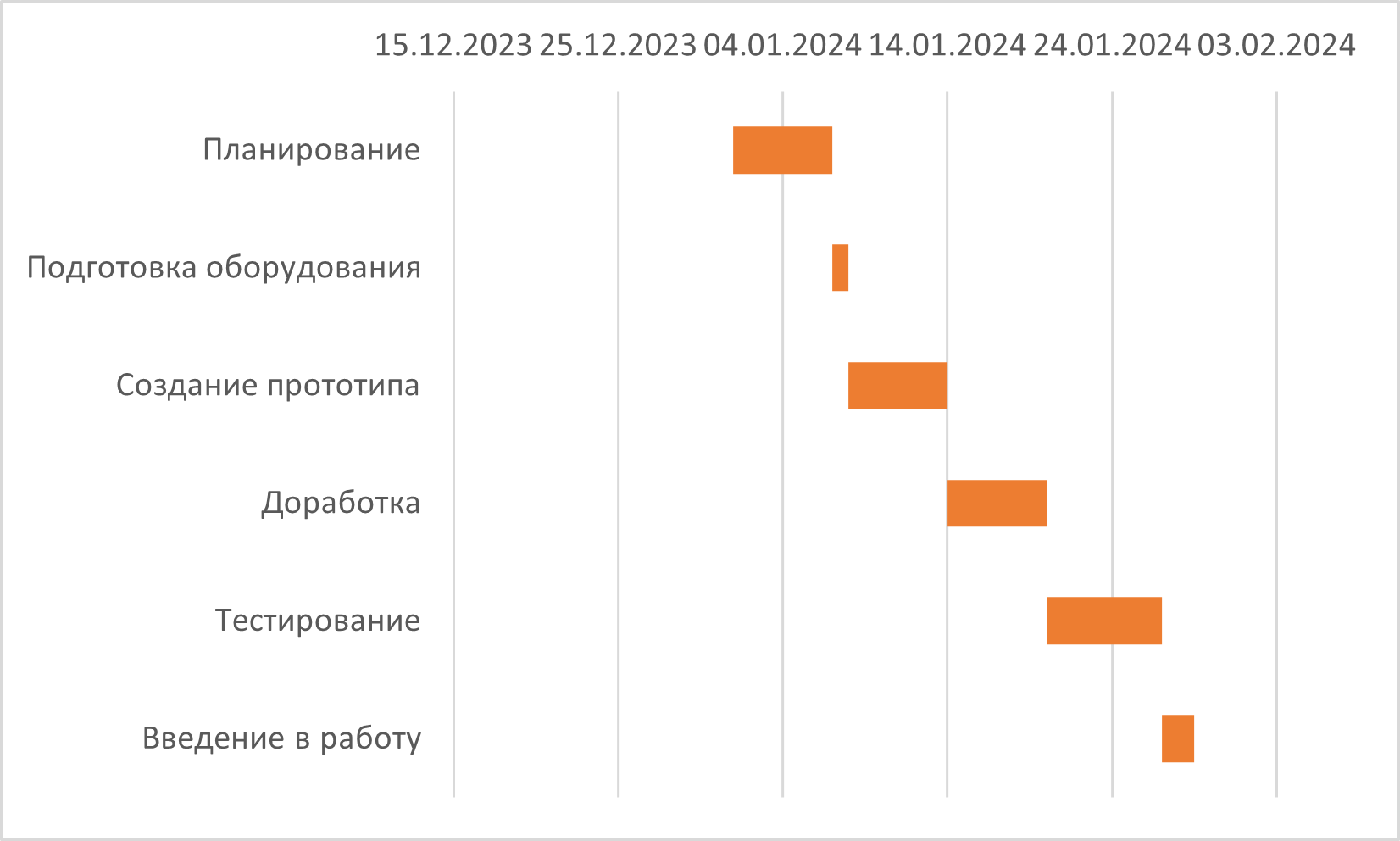


Диаграмма Ганта