Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| по ПМ.05 | Проектирование и разработка информационных систем |
| Выполнил | Доничев М.О. |
| Курс, группа | 3 курс, группа ИС-3 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Руководитель практики | Гонгадзе А.С. |

г. Белореченск

2024

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| ВВЕДЕНИЕ |  |
| 1 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ |  |
| 1.1 Анализ предметной области |  |
| 1.2 Техническое задание |  |
| 1.3 Эскизный проект |  |
|  |  |
|  |  |
|  |  |
| ЗАКЛЮЧЕНИЕ |  |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ |  |
| ПРИЛОЖЕНИЕ |  |

ВВЕДЕНИЕ

В эпоху цифровых компании по разработке программных продуктов сталкиваются с необходимостью автоматизации бизнес-процессов для повышения эффективности и конкурентоспособности. Это особенно актуально в условиях стремительного изменения требований рынка и технологий.

Объект исследования является компания «Vermilion», его бизнес-процессы и информационные потоки.

Предмет исследованияразработка и внедрение информационной системы, способной автоматизировать процессы управления проектами, учета ресурсов и взаимодействия с клиентами.

Цель исследованиясоздание информационной системы для компании, которая позволит автоматизировать процессы управления проектами, учета ресурсов и обработки обратной связи от клиентов, с учетом требований заказчика и особенностей компании.

Задачи исследования:

- Анализ структуры, бизнес-процессов и информационных потоков компании.

- Определение требований заказчика к информационной системе.

- Разработка концептуальной модели информационной системы.

- Разработка архитектуры информационной системы.

- Создание базы данных для информационной системы.

- Разработка модулей управления проектами, учета ресурсов и обработки обратной связи.

- Тестирование и отладка информационной системы.

- Внедрение информационной системы в компанию.

- Подготовка документации к информационной системе.

Методы исследования:

- Анализ документов, относящихся к компании и её проектам.

- Использование специализированного ПО для моделирования бизнес-процессов.

- Разработка UML-диаграмм, концептуальной модели и архитектуры системы.

- Разработка базы данных с использованием SQL.

- Java для разработки модулей программирование модулей управления проектами и учета ресурсов.

- Использование специализированного ПО для тестирования и отладки системы.

- Учет требований заказчика и особенностей компании при внедрении.

- Подготовка документации с учетом требований заказчика и нормативных документов.

1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

1.1 Анализ предметной области

Vermilion - это компания занимающаяся web-приложениями по типу: Порталы, CRM-системы, системы управления контентом (CMS), социальные сети. Отлично знает индустрию информационных технологий. Индивидуально подходим к каждому проекту.

Сотрудники компании «Vermilion» разделяются на несколько отделов: Разработка, тестирование, управление проектами, анализ требований, дизайн, бухгалтерия, и юридический отдел. Каждый отдел состоит из людей, обладающих навыками и знаниями, необходимыми для выполнения своих задач.

Отдел Разработки: включает в себя опытных программистов, специализирующихся в различных областях

Отдел Тестирования: состоит из квалифицированных инженеров по обеспечению качества, которые используют различные методики тестирования

Отдел Управления: Проектами состоит из опытных менеджеров проектов, ответственных за планирование, организацию, контроль и завершение проектов в срок и в рамках бюджета.

Отдел Анализа Требований: включает в себя бизнес-аналитиков, которые взаимодействуют с клиентами, собирают и анализируют требования к программному обеспечению, создают подробные технические задания и спецификации.

Отдел Дизайна: состоит из UX/UI дизайнеров, которые создают удобные и привлекательные интерфейсы для программного обеспечения.

Все отделы работают в тесном сотрудничестве, обеспечивая эффективную разработку и выпуск качественного программного обеспечения, а также поддерживая стабильную и эффективную работу компании в целом.

Виды запросов в информационной системе:

1. **Добавление клиентов**
2. **Получить перечень клиентов**
3. **Получить перечень разработчиков, работавших над проектами с указанными характеристиками**
4. **Добавить разработчиков в проект**
5. **Получить количество доступных разработчиков на данный момент**
6. **Получить сведения о конкретном доступном разработчике и его навыках**
7. **Получить список занятых сейчас разработчиков (работающих над проектами)**
8. **Добавления проекта**
9. **Получить данные о проектах, в которых участвует указанный клиент:**
10. **Добавить информацию о разработчике**
11. **Получить сведения о главном разработчике указанного проекта**
12. **Получить сведения о клиентах, с которыми заключены договора на разработку**
13. **Получить сведения о разработчиках, работавших над наибольшим количеством проектов**
14. **Получить сведения о конкретном разработчике, сколько проектов он выполнил**
15. **Реализовать список проектов, запланированных к завершению в течение следующего месяца**
16. **Реализовать удаление, изменение, добавление информации о проекте и разработчике**
17. **Получить список разработчиков, которые не были назначены ни на один проект**
18. **Получить количество багов, отсортированных по приоритету:**
19. **Обновить статус проекта на “завершен”**
20. **Назначить разработчика на проект**

1.2 Техническое задание

1. Общие сведения

1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: Автоматизированная информационная система компании «Vermilion».

Краткое наименование системы: АИС «Vermilion».

1.2 Шифр темы или шифр (номер) договора

Шифр темы: АИС-ГК-28

Номер договора: №2/34-56-58-012 от 12.12.2024.

1.3. Наименование предприятий разработчика и заказчика системы, их реквизиты

Полное наименование заказчика: Частное учреждение профессиональная образовательная организация ТЕХНИКУМ «БИЗНЕС И ПРАВО».

Сокращённое название: ЧУПОО ТЕХНИКУМ «БИЗНЕС И ПРАВО».

Юридический и почтовый адрес: [352632, Краснодарский Край, г Белореченск, ул Чапаева, д. 48](https://yandex.ru/maps/?source=exp-counterparty_entity&text=352632,%20%D0%9A%D1%80%D0%B0%D1%81%D0%BD%D0%BE%D0%B4%D0%B0%D1%80%D1%81%D0%BA%D0%B8%D0%B9%20%D0%9A%D1%80%D0%B0%D0%B9,%20%D0%BC.%D1%80-%D0%BD%20%D0%91%D0%B5%D0%BB%D0%BE%D1%80%D0%B5%D1%87%D0%B5%D0%BD%D1%81%D0%BA%D0%B8%D0%B9,%20%D0%B3.%D0%BF.%20%D0%91%D0%B5%D0%BB%D0%BE%D1%80%D0%B5%D1%87%D0%B5%D0%BD%D1%81%D0%BA%D0%BE%D0%B5,%20%D0%B3%20%D0%91%D0%B5%D0%BB%D0%BE%D1%80%D0%B5%D1%87%D0%B5%D0%BD%D1%81%D0%BA,%20%D1%83%D0%BB%20%D0%A7%D0%B0%D0%BF%D0%B0%D0%B5%D0%B2%D0%B0,%20%D0%B4.%2048).

Контактный телефон: +7 (861) 553 39 12.

ИНН: 2310029652

КПП: 236801001.

БИК: 040349722

ОГРН: 1022301620791

Расчетный счет: 407038106.

Полное наименование разработчика: Доничев Максим.

Юридический и почтовый адрес: 352632, Краснодарский Край, г Майкоп станица Ханская, ул. Краснооктябрьская, д.114

Контактный телефон +7 (928) 482 88 66.

2. Назначение и цели создания системы

Разрабатываемая система предназначена для управления проектами в компании, специализирующейся на разработке веб-приложений, мобильных приложений, систем встраиваемого ПО. Система автоматизирует процессы отслеживания хода проектов, управления ресурсами и обработки ошибок. Она будет предоставлять информацию о деталях проекта, включая данные о клиенте, используемых технологиях, членах команды, сроках и бюджете.

Основными целями создания информационной системы для компании Vermilion являются:

-Упрощение отслеживания проектов.

- Предоставление клиентам актуальной информации о статусе проекта.

-Повышение эффективности разработки.

Для достижения поставленных целей информационная система должна решать следующие задачи:

- Отслеживание проектов и управление ресурсами.

- Управление клиентами и коммуникация.

- Управление задачами и отслеживание ошибок.

- Отчетность и аналитика, генерация отчетов о сроках проектов.

3. Характеристика объектов автоматизации

Характеристика объектов автоматизации для компании по разработке программного обеспечения (например, веб-приложений):

В ходе разработки информационной системы для управления проектами в компании Vermilion автоматизируются процессы учета задач, задач для разработчиков, статусов проектов, управления версиями кода, отслеживания ошибок (багов), контроля задач и сроков, и управления ресурсами (в частности, временем разработчиков).

Автоматизируемыми процессами будут такие операции, как создание и назначение задач, отслеживание статусов задач и проектов, управление версиями кода, фиксация ошибок и отслеживание их решения, контроль соответствия сроков, и распределение ресурсов. Система обеспечит быстрый доступ к информации о проектах, задачах, разработчиках, задачах и багах, что позволит упростить работу руководителей проектов, разработчиков, уменьшить вероятность ошибок и повысить качество выполнения проектов.

При внедрении системы будет достигнуто:

- упрощение процессов управления проектами и задачами;

- повышение эффективности работы разработчиков и руководителей проектов;

- увеличение скорости выполнения проектов и уменьшение сроков сдачи.

Объекты автоматизации характеризуются:

необходимостью доступа к базе данных о проектах, разработчиках, задачах и багах;

- возможностью быстрого доступа к информации для принятия оперативных решений;

- потребностью к постоянному доступу к сети Интернет для синхронизации данных.

4. Требования к системе

4.1 Требования к структуре и функционированию системы

Система должная предполагать наличие следующих модулей, представленных на рисунке 1:

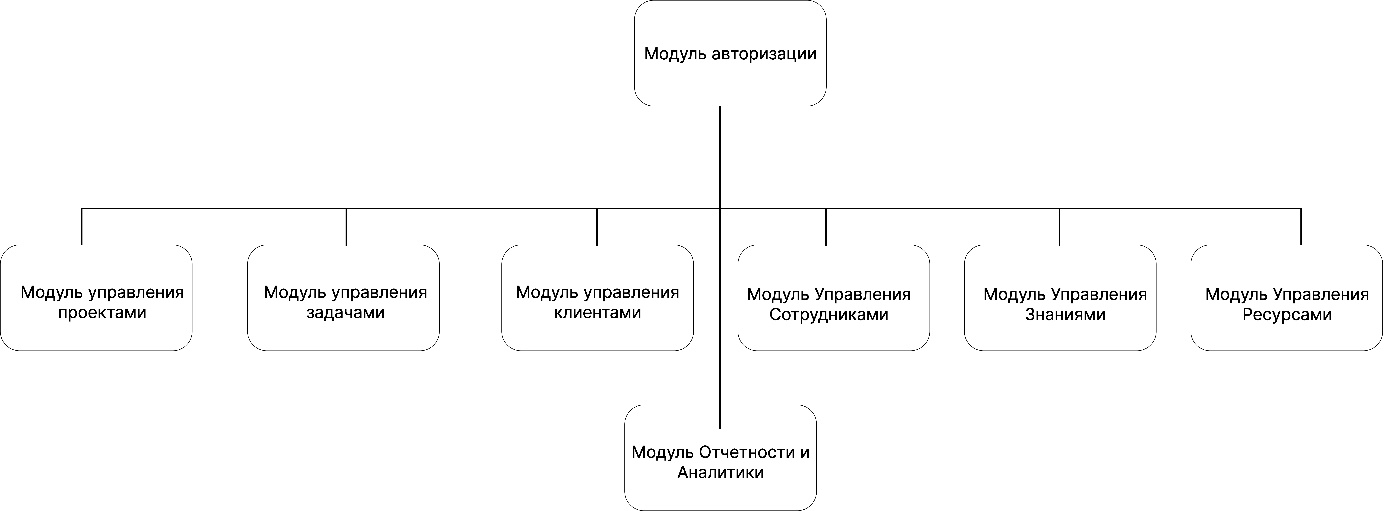
****

Рисунок 1 – Модули разрабатываемой системы

Серверная часть:

- Модуль Авторизации;

- Модуль Управления Проектами;

- Модуль Управления Задачами;

- Модуль Управления Клиентами;

**- Модуль Управления Сотрудниками**

- Модуль Управления Знаниями;

- Модуль Управления Ресурсами

- Модуль Отчетности и Аналитики;

Клиентская часть:

- Интерфейс администратора;

**- Интерфейс клиента;**

На данном этапе необходимо описать примерную архитектуру разрабатываемой системы. Для того, чтобы сделать информационную систему ателье, мы будем использовать MVC архитектуру. Чтобы убедиться, что она нам точно подходит рассмотрим ее подробнее. Модель MVC решает следующие задачи:

- изменение только пользовательского интерфейса, а не бизнес-логики приложения;

- использование в одном приложении разных интерфейсов с возможностью выбора;

- замена реакции приложения на действия пользователя за счет использования другого контроллера.

В более широком плане внедрение модели помогает:

- упростить код большого приложения, сделать его понятным и структурированным, облегчить поддержку, тестирование, повторное использование элементов;

- организовать независимую работу различных отделов, которые занимаются разработкой своей части программного продукта;

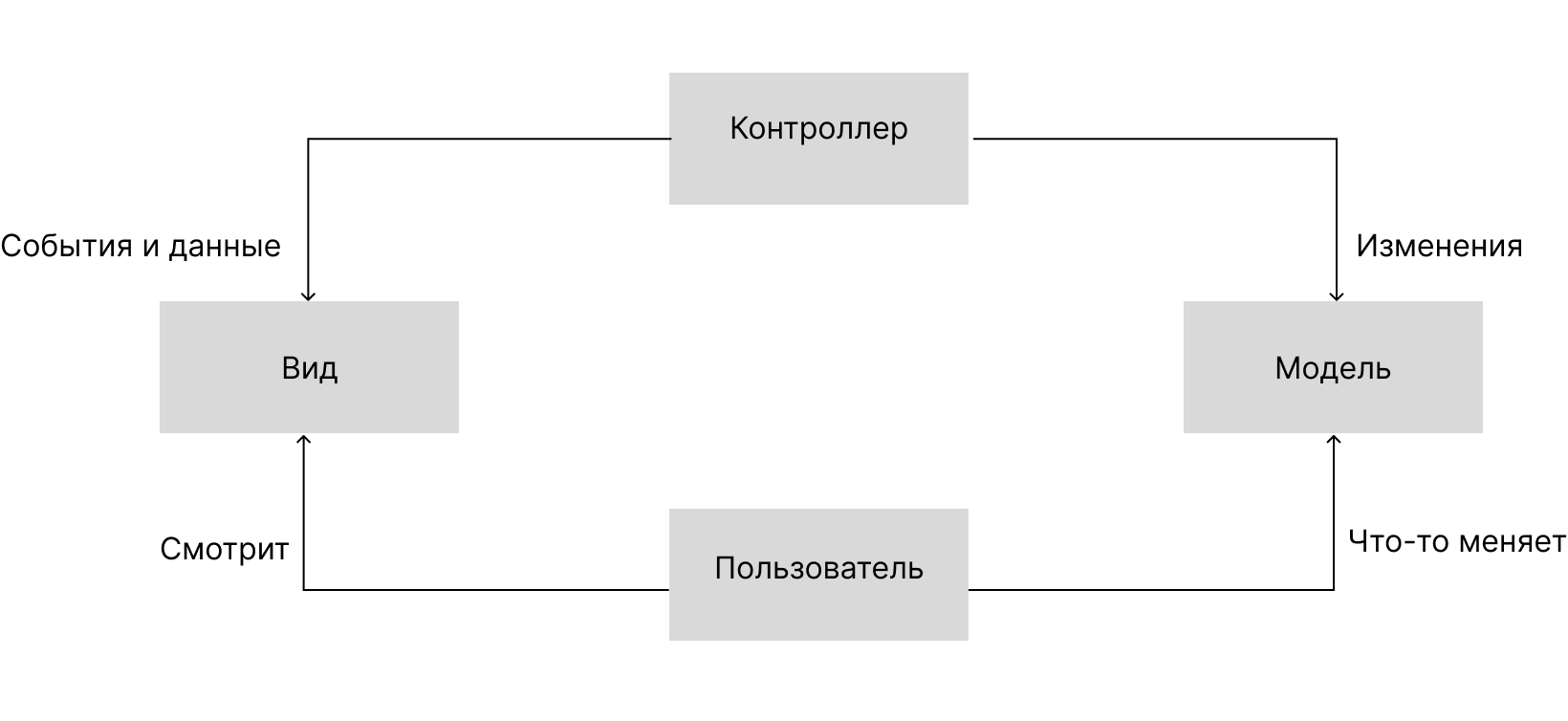
- упростить программную поддержку MVC-приложения за счет модификации отдельных компонентов, а не всей архитектуры.

Рисунок 2 – MVC архитектура

4.2 Требования к видам обеспечения

Требования к видам обеспечения для функционирования информационной системы управления проектами в компании по разработке программного обеспечения;

Работа клиента и сервера может быть организована следующим образом;

Клиентская часть:

- На компьютерах разработчиков и менеджеров проектов потребуется наличие клиентских приложений.

-  Приложения будут иметь графический интерфейс пользователя, разработанный с учетом требований и удобства использования для различных ролей сотрудников (разработчик, менеджер проекта, администратор).

- В зависимости от роли и прав доступа, сотрудники смогут выполнять различные операции в приложении, такие как:

- Разработчики: Просмотр назначенных задач, обновление статуса задач, создание отчетов об ошибках (багах), управление версиями кода.

-  Менеджеры проектов: Просмотр статуса проекта, назначение задач разработчикам, управление бюджетом проекта, мониторинг сроков, создание отчетов о ходе проекта.

- Администраторы: Управление пользователями, настройка системы, мониторинг использования ресурсов, генерация отчетов.

Серверная часть:

- Сервер базы данных будет хранить всю информацию о проектах, задачах, разработчиках, клиентах, ошибках (багах), технологиях и т.д.

- Серверная часть должна быть реализована с использованием Java и выше.

- Сервер должен обеспечивать RESTful API для взаимодействия с клиентскими приложениями.

- Swagger (или OpenAPI) для документирования API.

- Система будет обеспечивать безопасность данных, авторизацию пользователей и контроль доступа на основе ролей.

В данном случае клиентская часть системы будет представлять собой отдельные приложения, доступные для каждого сотрудника. Приложения будут иметь графический интерфейс и предоставлять необходимый функционал в зависимости от роли пользователя.

Серверная часть будет состоять из сервера базы данных и API. Взаимодействие между клиентскими приложениями и сервером будет осуществляться через RESTful API, обеспечивающий обмен данными и выполнение запросов.

Сущности и база данных:

При анализе предметной области были выделены следующие сущности, на основе которых будет разрабатываться база данных системы:

- Проекты

- Задачи

- Разработчики

- Клиенты

База данных информационной системы должна состоять из следующих связанных таблиц:

1. Projects– таблица, содержащая информацию о проектах.
2. Tasks– таблица, содержащая информацию о задачах.
3. Developers– таблица, содержащая информацию о разработчиках.
4. Clients– таблица, содержащая информацию о клиентах.
5. Role – таблица, содержащая информацию о ролях

Таблица 1: Projects

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| projectId | INTEGER | 11 | Идентификатор проекта | PRIMARY KEY |
| projectName | VARCHAR | 255 | Название проекта | NOT NULL |
| clientId | INTEGER | 11 | Идентификатор клиента | NOT NULL, FOREIGN KEY (Clients.clientId) |
| projectManagerId | INTEGER | 11 | Идентификатор менеджера проекта | NOT NULL, FOREIGN KEY (Developers.developerId) |
| startDate | DATE |  | Дата начала проекта | NOT NULL |
| endDate | DATE |  | Планируемая дата окончания |  |
| budget | DECIMAL |  | Бюджет проекта | NOT NULL |

Таблица 2 - Tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Колонка | Тип данных | Размер | Комментарий | | Ограничения |
| id | INTEGER | 11 | Уникальный идентификатор | | PRIMARY KEY |
| project\_id | INTEGER | 11 | Идентификатор проекта | | NOT NULL, FOREIGN KEY |
| taskName | VARCHAR | 255 | | Название задачи | NOT NULL |
| assignedTo | INTEGER | 11 | | Идентификатор разработчика | NOT NULL, FOREIGN KEY |
| status | VARCHAR | 50 | | Статус задачи | NOT NULL |
| dueDate | DATE |  | | Дата дедлайна |  |
| description | TEXT |  | | Описание задачи |  |

Таблица 3. ManagerDevelopers

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| developerId | INTEGER | 11 | Уникальный идентификатор | PRIMARY KEY |
| firstName | VARCHAR | 255 | Имя разработчика | NOT NULL |
| lastName | VARCHAR | 255 | Фамилия разработчика | NOT NULL |
| Email | VARCHAR | 255 | Электронная почта | UNIQUE, NOT NULL |
| phoneNumber | VARCHAR | 20 | Номер телефона |  |
| specialization | VARCHAR | 255 | Специализация |  |

Таблица 4: Clients

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| clientId | INTEGER | 11 | Уникальный идентификатор клиента | PRIMARY KEY |
| clientName | VARCHAR | 255 | Название компании | NOT NULL |
| contactPerson | VARCHAR | 255 | Контактное лицо |  |
| contactEmail | VARCHAR | 255 | Электронная почта клиента | UNIQUE, NOT NULL |
| phoneNumber | VARCHAR | 20 | Телефон клиента |  |

Таблица 5: Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| developerId | INTEGER | 11 | Идентификатор разработчика | FOREIGN KEY, NOT NULL |
| roleId | INTEGER | 11 | Идентификатор роли | FOREIGN KEY |
| roleName | VARCHAR | 255 | Название роли | NOT NULL |

4.3 Требования к эргономике и технической эстетике

Общие требования:

- Фоновый цвет серого к белому.

Выделение элементов при наведении на кнопки, элементы управления и другие элементы, они должны выделяться путем изменения цвета фона или появления подсветки, не нарушая читаемости.

- Шрифты и размеры: Ясные, читаемые шрифты. Размеры текста должны быть подходящими для комфортного восприятия на различных устройствах (адаптивный дизайн). Использование контрастных цветовых схем для лучшей читаемости.

- Адаптивный дизайн: Дизайн должен корректно отображаться на разных устройствах (разные размеры экранов, разрешения).

Главный экран:

- Меню: Использование навигационной панели (например, горизонтальное меню) для доступа к основным функциям (Проекты, Задачи, Разработчики, Клиенты, Ошибки, Бюджет).

- Таблицы: Использование таблиц JTable для отображения данных.

- Кнопки: Все кнопки должны иметь одинаковую высоту и стилистически объединены. Прямые углы и бардовая заливка.

- Фильтры и сортировка: Должны быть опции сортировки и фильтрации данных в таблицах.

Элементы интерфейса:

- Таблицы: Прозрачные границы ячеек и заголовков.

- Кнопки: Прямые углы и бардовая заливка.

- Поля ввода: Чёткие метки для полей, стилизованный ввод.

Экран авторизации:

- Центрирование: Все элементы страницы должны быть центрированы и по вертикали, и по горизонтали.

- Заголовок: «Авторизация».

- Ширина элементов: Все поля ввода и кнопка «Войти» должны иметь одинаковую ширину.

- Кнопка входа: бардовый цвет, чёрный текст. При наведении — белый.

* 1. Требования к содержанию и оформлению выводимых сообщений

- Отображение всплывающей подсказки в форме выноски снизу от поля ввода в случае ввода некорректного формата, уведомление автоматически исчезает, когда пользователь начнет заново вводить данные;

- Если пользователь вводит неверные данные и нажимает на кнопку войти, то сформируется окно с надписью: «Неверный логин или пароль. Попробуйте заново»;

- Сообщения об ошибках при вводе некорректных данных;

- Автоматическое исчезновение подсказки при начале нового ввода данных в соответствующее поле;

- Сообщения об ошибках при незаполненных обязательных полях (например, «Недопустимое заполнение» или «Пожалуйста, заполните все поля»);

- Отображение ошибки при добавлении, редактировании, удалении данных;

- Сообщение об ошибке при проблемах с сервером (например, «Произошла ошибка при запуске сервера», «Пожалуйста, проверьте соединение к сети и попробуйте снова»).

4.5 Требования к видам обеспечения

Для функционирования системы требуется сервер и клиентских компьютеры, на которых будет производится запуск системы.

Требования для сервера БД:

- Windows 10 и выше;

- Процессор 6 ядер (12 логических потоков), частота – 3—3,40

- Оперативная память 16 Гб и выше

- Свободное дисковое пространство не менее 3 Гб

SSD (твердый диск), тип - SATA2, скорость вращения 7200 об/мин

Требования к клиентским устройствам:

Персональные компьютеры должны быть под управлением Windows 8 или выше. Монитор на клиентских компьютерах должен быть с разрешением не ниже 1280х20 пикселей. Браузер должен быть с поддержкой HTML5 и CSS3.

4.6 Эскизы основных окон системы

На рисунках 3-8 представлены эскизы основных окон системы.

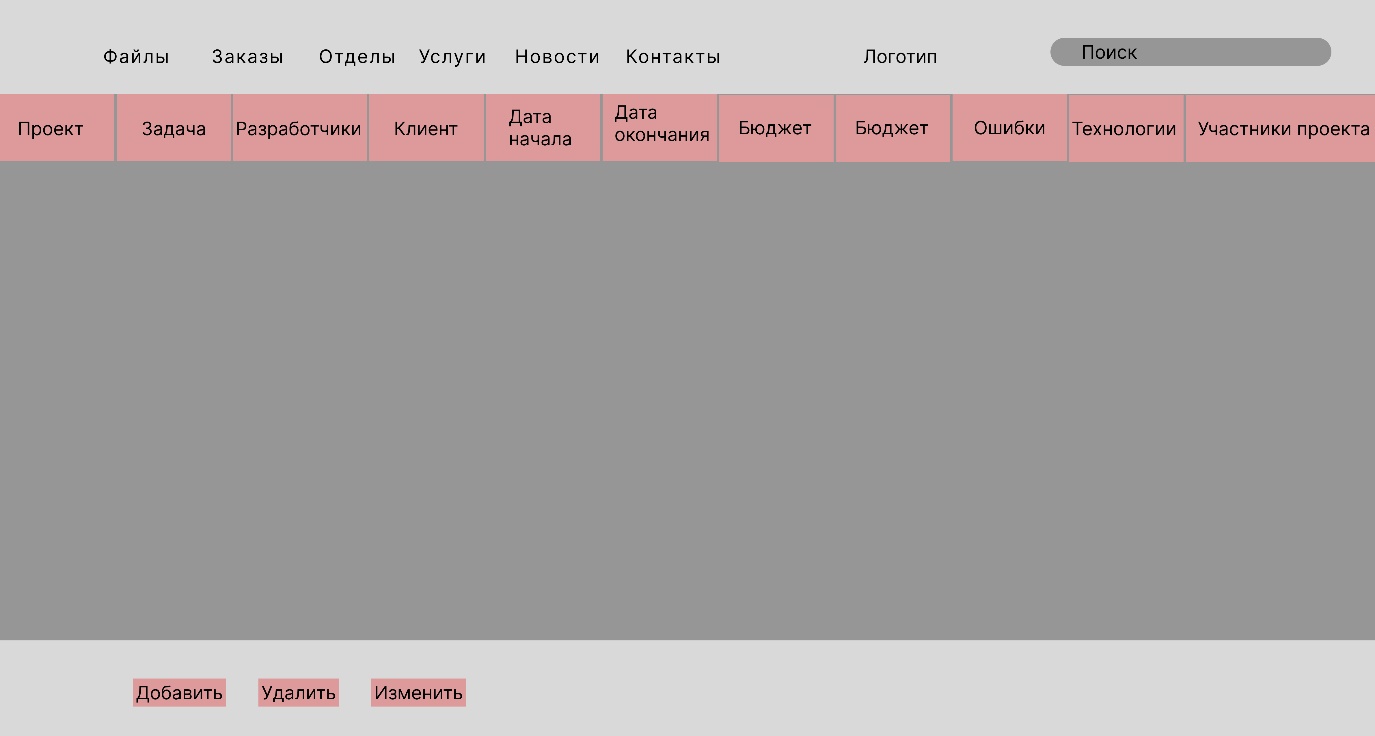


Рисунок 3 – Главный экран

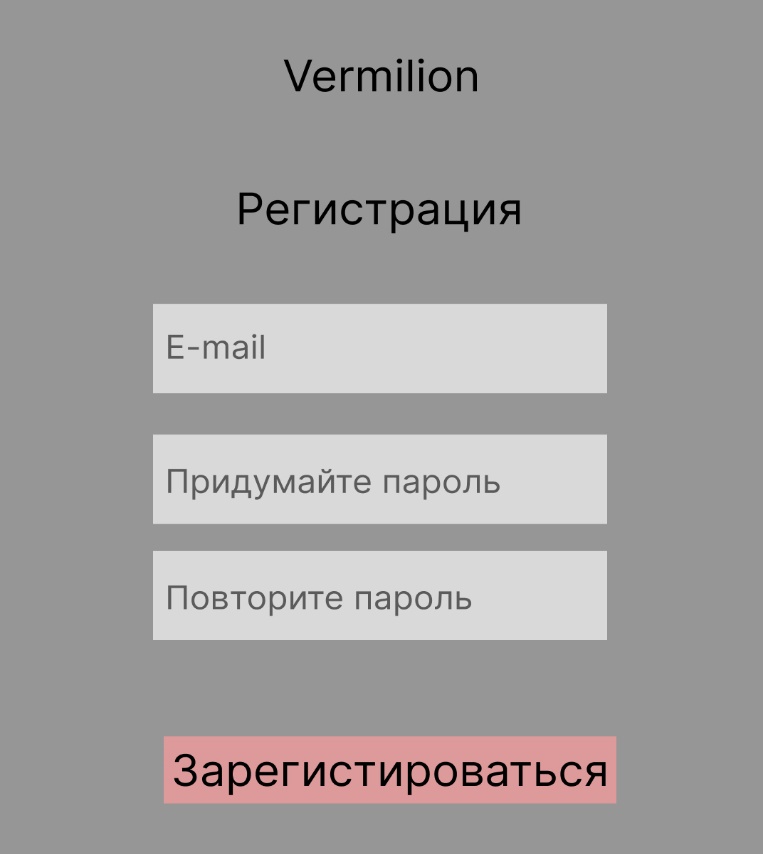


Рисунок 4 – Форма регистрации

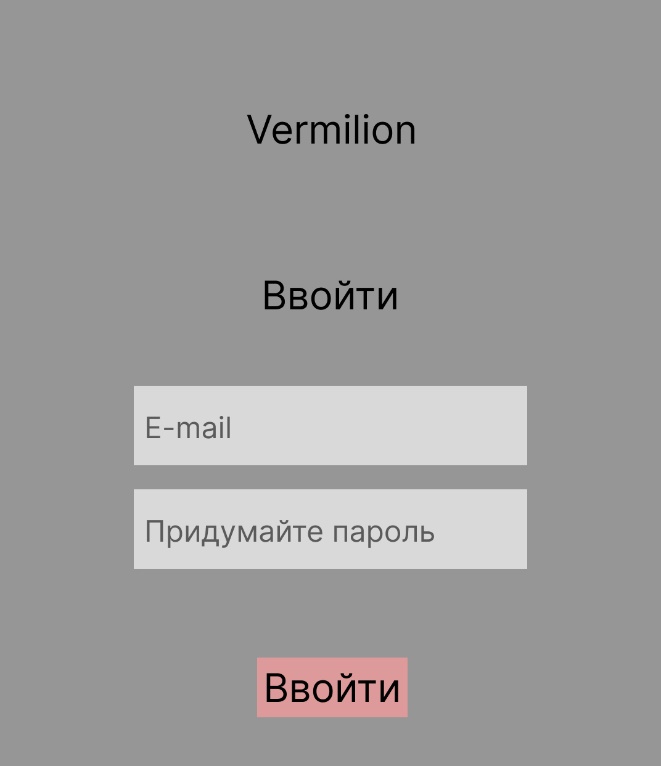


Рисунок 5 – Форма входа



Рисунок 6 – Окно добавление заказа

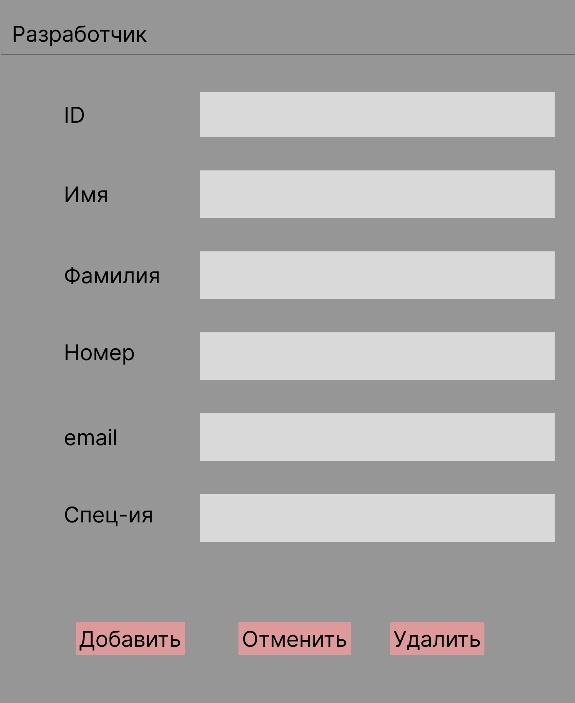


Рисунок 7 – Окно добавление разработчика

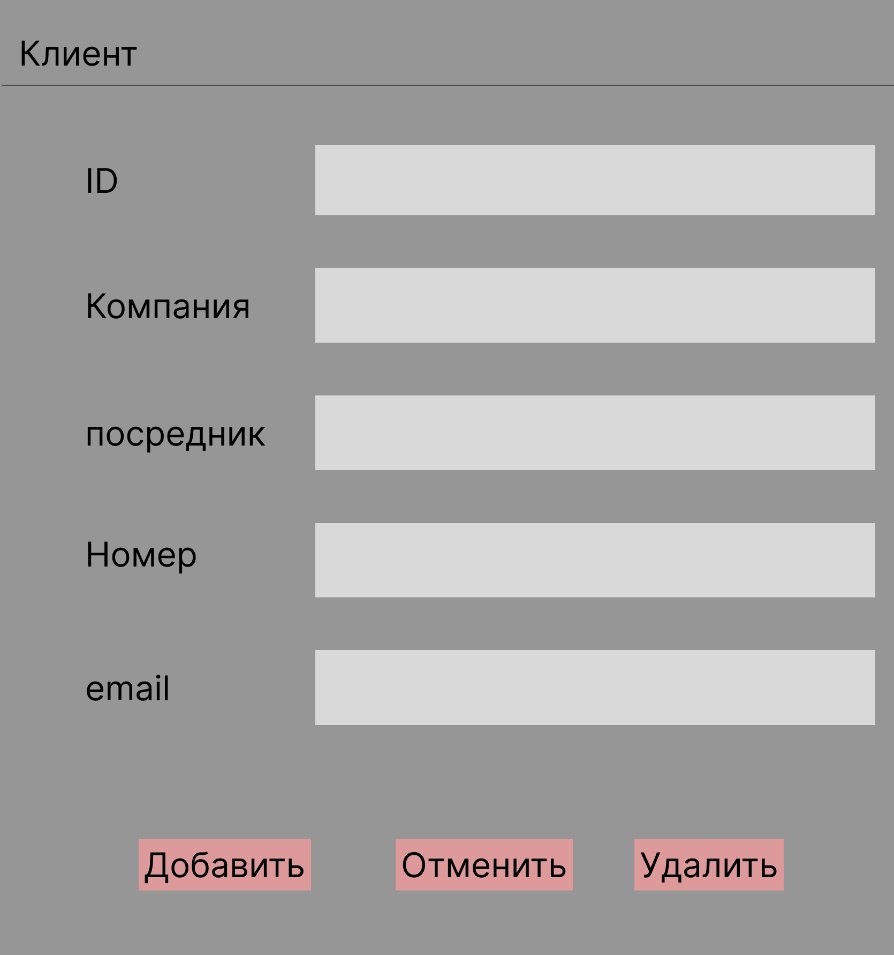


Рисунок 8 – Окно добавление клиента

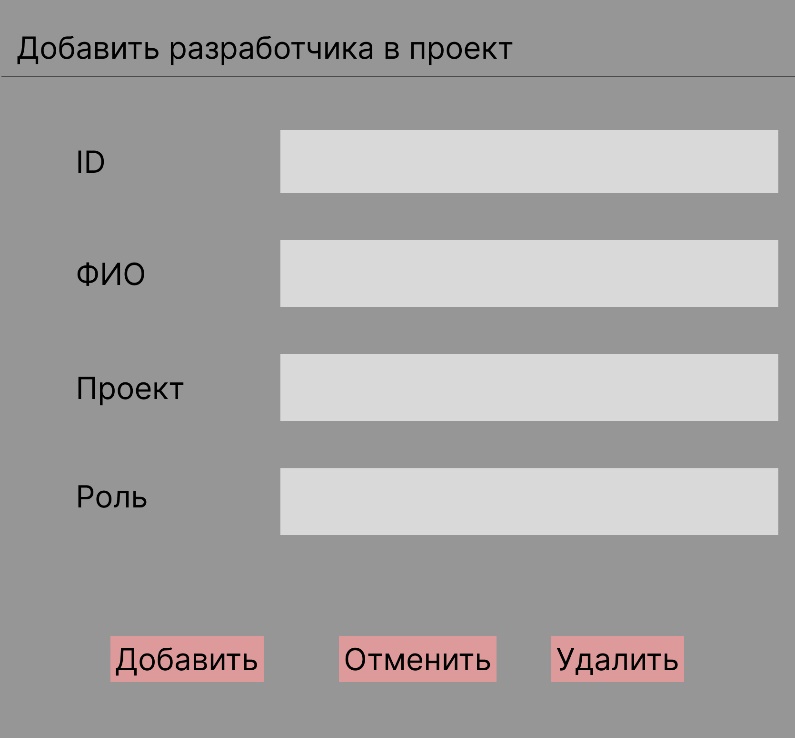


Рисунок 9 – Окно добавление разработчика в проект

1. Состав и содержание работ по созданию системы:

Комплекс работ по созданию информационной системы управления проектами должен осуществляться в несколько этапов:

1. Анализ требований:

- Сбор и анализ требований к системе со стороны заказчика (руководителей компании, менеджеров проектов, разработчиков).

- Определение функциональных и нефункциональных требований (производительность, масштабируемость, безопасность).

- Изучение существующих бизнес-процессов компании в области

1. Проектирование системы:

- Разработка концептуальной модели системы.

- Проектирование архитектуры системы.

- Проектирование базы данных.

- Проектирование пользовательского интерфейса.

1. Разработка системы:

- Создание серверной части системы

- Разработка клиентской части системы.

- Реализация модулей для управления проектами, задачами, разработчиками, клиентами, ошибками и т.д.

- Разработка системы авторизации и управления доступом.

1. Тестирование системы:

- Модульное тестирование отдельных компонентов.

- Интеграционное тестирование взаимодействия между компонентами.

- Системное тестирование всей системы в целом.

- Тестирование производительности и нагрузочное тестирование.

- Тестирование безопасности.

1. Развертывание и внедрение системы:

- Подготовка серверной и клиентской инфраструктуры.

- Установка и настройка системы на серверах и клиентских устройствах.

- Миграция данных из существующих систем.

- Обучение персонала работе с новой системой.

1. Опытная эксплуатация и доработка системы:

- Опытная эксплуатация системы в тестовой среде, максимально приближенной к реальной.

- Сбор отзывов от пользователей.

- Устранение выявленных недочетов и ошибок.

- Доработка системы на основе полученных данных.

1. Ввод системы в промышленную эксплуатацию:

- Окончательный переход на использование новой системы.

- Выведение из эксплуатации старых систем.

1. Техническая поддержка и сопровождение:

- Техническая поддержка пользователей.

- Регулярное обновление системы.

- Развитие системы).

1. Приемка работ:

- Проведение предварительных испытаний на стенде исполнителя.

- Составление акта приема-сдачи.

1. Порядок контроля и приемки системы:

Система передается в виде полностью функционирующего комплекса на базе средств вычислительной техники заказчика и в сроки, установленные договором. Приемка системы осуществляется комиссией в составе уполномоченных представителей заказчика и исполнителя. Весь комплект документации передается заказчику в электронном и печатном виде.

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Заказчик обязан предоставить сервер и персональные компьютеры, требования к которым указаны в разделе 4.2 «Требования к видам обеспечения». Необходимое программное обеспечение для сервера:

- Операционная система: Windows 10

- Spring Framework

- СУБД: MySQL.

Необходимое программное обеспечение для персонального компьютера: - Операционная система: Windows 10;

- Google Chrome(x64)/ Opera(x64)

8. Требования к документированию

Перечень подлежащих разработке комплектов и видов документов:

- Описание анализа предметной области

- Техническое задание

- Эскизный проект

- Сервер

- Клиент

9. Источники разработки

Источниками разработки настоящего Технического Задания являются:

- перечень документации, представленный в разделе 1.2 «Основания для проведения работ»;

- ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;

- ГОСТ 24.601-86 Автоматизированные системы. Стадии создания;

- ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению;

- ГОСТ 24.601 Исследование и обоснование создания АС.

1.3 Эскизный проект

Декомпозиция системы на комплексы и проработка архитектурных решений

Назначение системы АИС «Vermilion» предназначена для комплексного информационно-управленческого обеспечения процессов в части исполнения процессов, описанных ниже:

- Управление проектами и портфелем проектов

- Управление задачами и подзадачами.

- Управление разработчиками и ресурсами.

- Управление ошибками.

- Управление репозиториями кода и версиями.

Информационная система управления разработкой программного обеспечения разрабатывается для повышения эффективности управления проектами и качества разработки продуктов.

Система позволит централизованно хранить и обрабатывать всю информацию о проектах, задачах, разработчиках, ошибках, версиях кода, клиентах и технологиях. Для сотрудников компании система обеспечит удобный доступ ко всем необходимым данным и инструментам управления.

Основными целями создания Системы являются:

- Оптимизация процессов планирования, распределения задач и мониторинга выполнения проектов за счет автоматизации рутинных операций и предоставления актуальной информации о статусе проектов и задач.

- Повышение эффективности использования ресурсов и сокращение сроков разработки за счет улучшенного управления и координации работы команд.

- Улучшение качества программных продуктов благодаря систематическому отслеживанию ошибок.

Реализация вышеперечисленных целей преследует достижение ключевой цели - повышение рентабельности и конкурентоспособности, бюджете, сроках, статусе, команде разработчиков и т.д.

Для достижения поставленных целей информационная система управления гостиничным комплексом «Vermilion» должна решать следующие основные задачи:

- Управление задачами и подзадачами с возможностью назначения ответственных исполнителей, установки приоритетов и дедлайнов, отслеживания статуса выполнения задач.

- Управление базой данных разработчиков с информацией об их специализациях, навыках, доступности для проектов.

- Отслеживание и управление ошибками с возможностью добавления новых ошибок, назначения ответственных разработчиков, отслеживания статуса исправления ошибок, создания отчетов по ошибкам.

- Интеграция с системами управления версиями кода и ведение базы данных версий программного продукта.

На данном этапе рассмотрим, как будет вести себя система при выполнении выделенных функций. Для этого будем использовать диаграммы состояний в нотации UML.

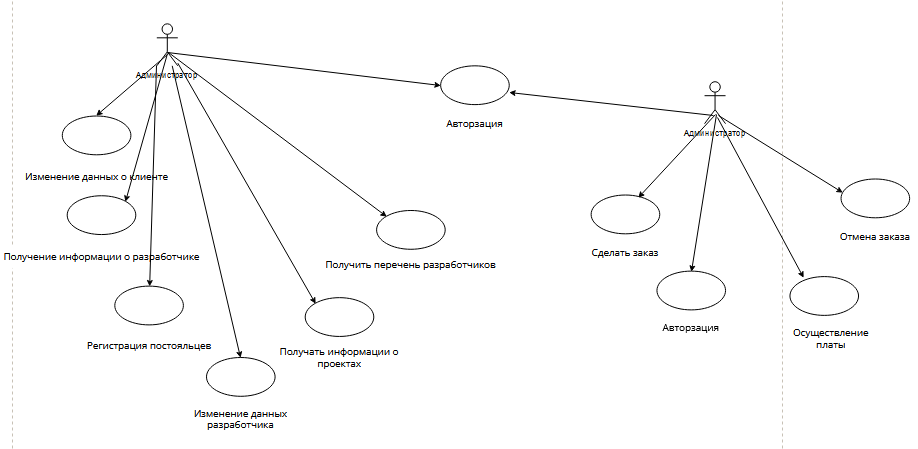


Рисунок 10 – Диаграмма варианта использования

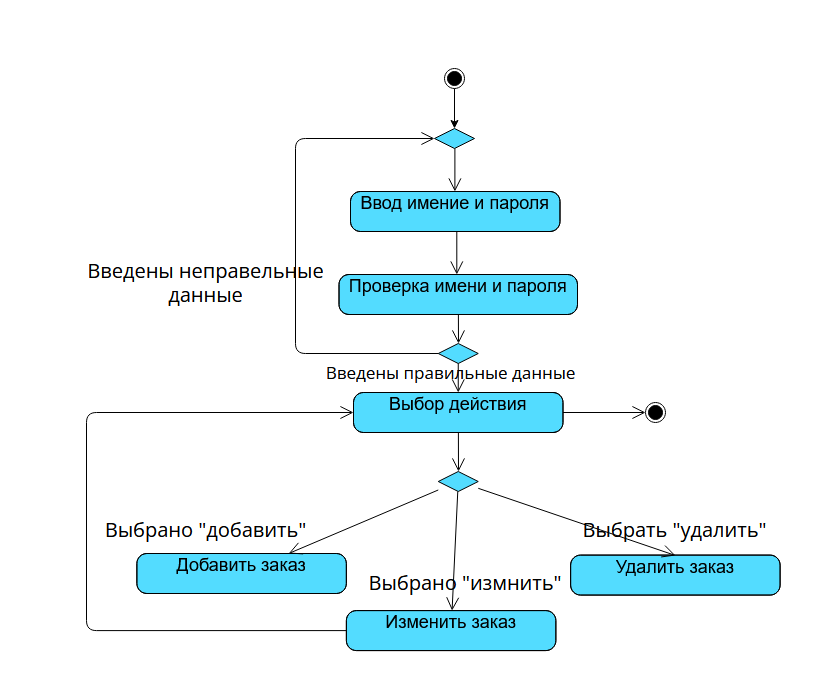


Рисунок 11 – Диаграмма деятельности

Клиент - серверная архитектура

- изменение только пользовательского интерфейса, а не бизнес-логики приложения;

- использование в одном приложении разных интерфейсов с возможностью выбора;

- замена реакции приложения на действия пользователя за счет использования другого контроллера.

В более широком плане внедрение модели помогает:

- упростить код большого приложения, сделать его понятным и структурированным, облегчить поддержку, тестирование, повторное использование элементов;

- организовать независимую работу различных отделов, которые занимаются разработкой своей части программного продукта;

- упростить программную поддержку MVC-приложения за счет модификации отдельных компонентов, а не всей архитектуры.

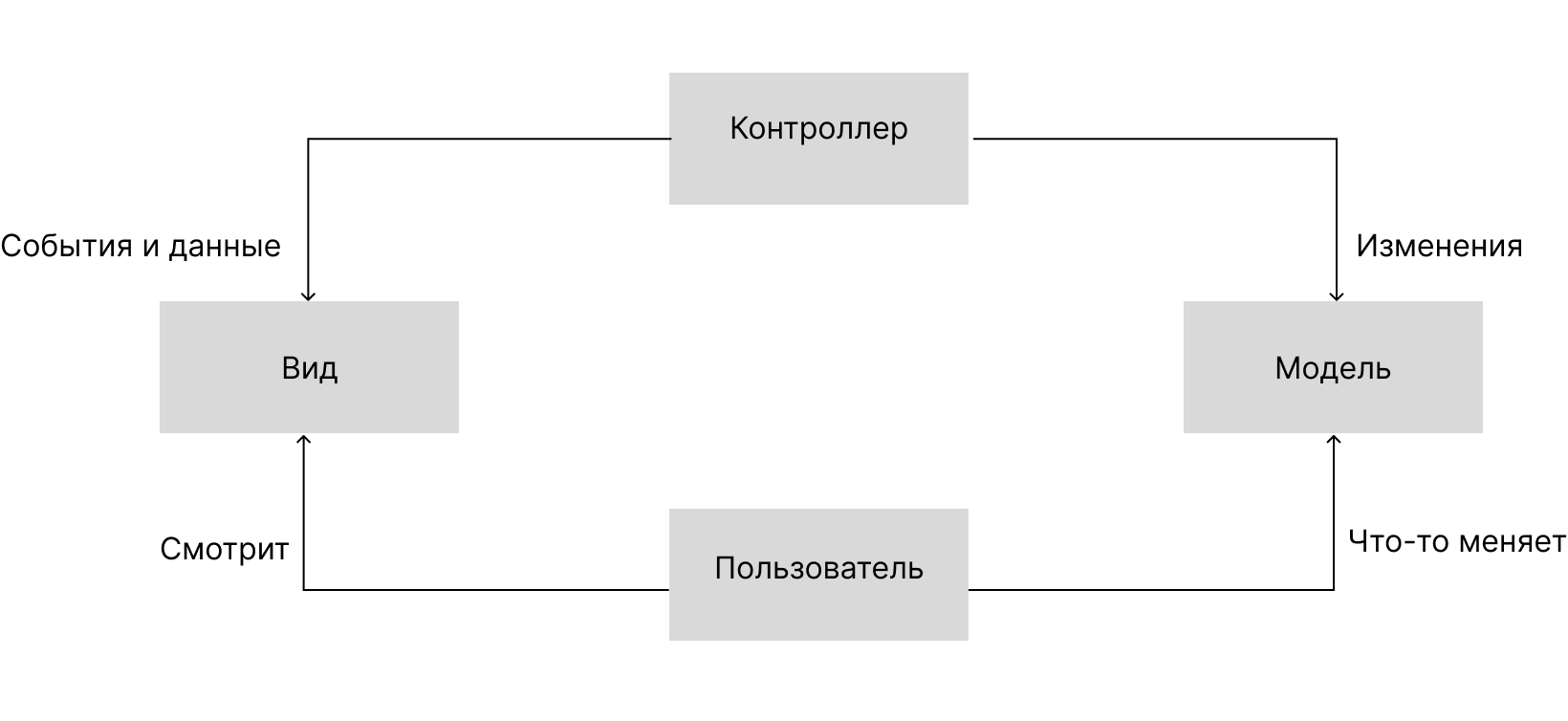


Рисунок 11 – Диаграмма деятельности

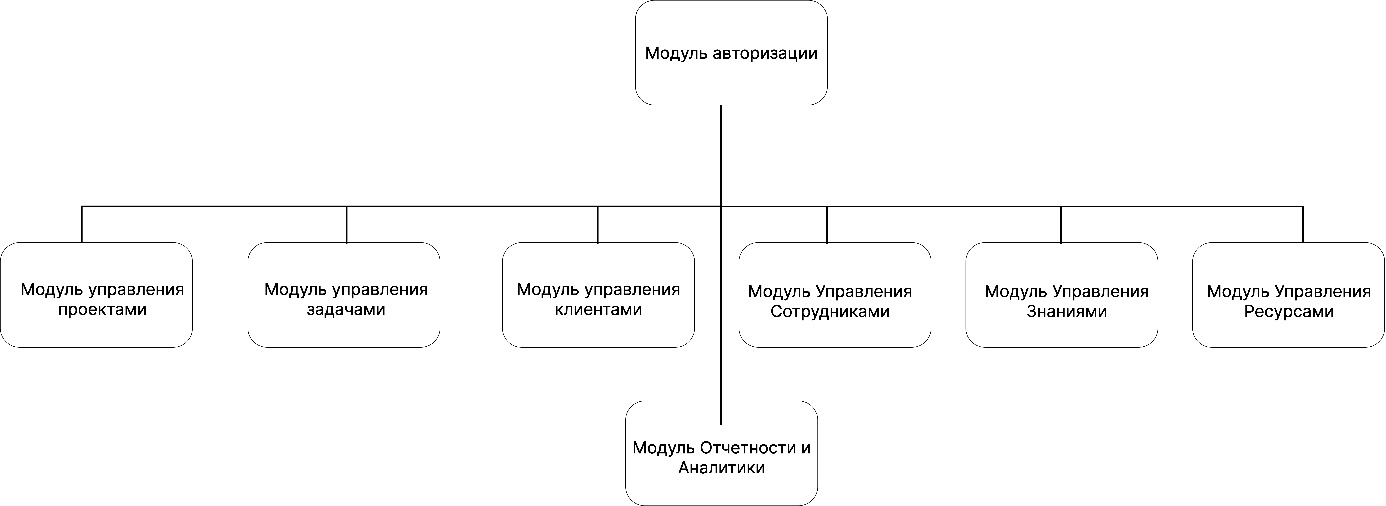
****

Рисунок 1 – Модули разрабатываемой системы

Серверная часть:

- Модуль Авторизации;

- Модуль Управления Проектами;

- Модуль Управления Задачами;

- Модуль Управления Клиентами;

**- Модуль Управления Сотрудниками**

- Модуль Управления Знаниями;

- Модуль Управления Ресурсами

- Модуль Отчетности и Аналитики;

В состав персонала, необходимого для обеспечения эксплуатации АИС «Vermilion» в рамках соответствующих подразделений, необходимо выделение следующих ответственных лиц, представленных в таблице 9:

Таблица 8 – Состав персонала, необходимого для обеспечения эксплуатации АИС «Vermillion»

|  |  |  |  |
| --- | --- | --- | --- |
| Должность |  | Количество | Подсистема |
| Администратор |  | 1 | Администрирование БД |
| Сотрудники |  | 5-10 | Управление номерным фондом |

Данные лица должны выполнять следующие функциональные обязанности:

- Пользователь:

- Управление статусами номеров (свободно, занято, ремонт и т.д.);

- Администратор: внесение правок в базу данных

1.Projects– таблица, содержащая информацию о проектах.

2.Tasks– таблица, содержащая информацию о задачах.

3.Developers– таблица, содержащая информацию о разработчиках.

4.Clients– таблица, содержащая информацию о клиентах.

Таблица 1: Projects в базе данных

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| projectId | INTEGER | 11 | Уникальный идентификатор проекта | PRIMARY KEY |
| projectName | VARCHAR | 255 | Название проекта | NOT NULL |
| clientId | INTEGER | 11 | Идентификатор клиента | NOT NULL, FOREIGN KEY |
| projectManagerId | INTEGER | 11 | Идентификатор менеджера проекта | NOT NULL, FOREIGN KEY |
| 1 | 2 | 3 | 4 | 5 |
| startDate | DATE | - | Дата начала проекта | NOT NULL |
| endDate | DATE | - | Планируемая дата окончания проекта |  |
| budget | DECIMAL | - | Бюджет проекта | NOT NULL |
| status | VARCHAR | 50 | Статус проекта | NOT NULL |

Данные о проектах размещаются в таблице Projects:

Таблица 2 - Tasks в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| id | INTEGER | 11 | Уникальный идентификатор задачи | PRIMARY KEY |
| project\_id | INTEGER | 11 | Идентификатор проекта | NO NULL, FOREIGN KEY |
| taskName | VARCHAR | 255 | Название задачи | NOT NULL |
| assignedTo | NTEGER | 11 | Идентификатор разработчика | NOT NULL, FOREIGN KEY |
| Status | VARCHAR | 50 | Статус задачи | NOT NULL |
| dueDate | DATE | - | Дата дедлайна задачи |  |
| description | TEXT | - | Описание задачи |  |

Данные о задачах размещаются в таблице Tasks;

Таблица 3. Developers в базе данных

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| developerId | INTEGER | 11 | Уникальный идентификатор разработчика | PRIMARY KEY |
| firstName | VARCHAR | 255 | Имя разработчика | NOT NULL |
| 1 | 2 | 3 | 4 | 5 |
| lastName | VARCHAR | 255 | Фамилия разработчика | NOT NULL |
| Email | VARCHAR | 255 | Электронная почта разработчика | UNIQUE, NOT NULL |
| phoneNumber | VARHAR | 20 | Номер телефона разработчика |  |
| specialization | VARCHAR | 255 | Специализация разработчика |  |

Данные о разработчиках размещаются в таблице Developers:

Таблица 4: Clients в базе данных

| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| --- | --- | --- | --- | --- |
| clientId | INTEGER | 11 | Уникальный идентификатор клиента | PRIMARY KEY |
| clientName | VARCHAR | 255 | Название компании | NOT NULL |
| contactPerson | VARCHAR | 255 | Контактное лицо |  |
| contactEmail | VARCHAR | 255 | Электронная почта клиента | UNIQUE, NOT NULL |
| phoneNumber | VARCHAR | 20 | Телефон клиента |  |

Данные о ролях в группе размещаются в таблице Role;

Таблица 5: Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Колонка | Тип данных | Размер | Комментарий | Ограничения |
| developerId | INTEGER | 11 | Идентификатор разработчика | FOREIGN KEY, NOT NULL |
| roleId | INTEGER | 11 | Идентификатор роли | FOREIGN KEY |
| roleName | VARCHAR | 255 | Название роли | NOT NULL |

Данные о клиентах размещаются в таблице Clients:

2. Разработка операционной среды или требований к ней

Серверная часть: ОС Windows 10, СУБД MySQL

Клиентские рабочие места: ОС Windows 10

Использование сети Интернет для удаленного доступа.

3. Разработка методов анализа качества системы и метрик оценки.

Метрики показателей качественной оценки:

- Надежность: Количество сбоев и их частота.

- Производительность: Время отклика системы на запросы.

- Возможность системы обрабатывать увеличивающиеся объемы данных и пользователей.

- Уровень защиты данных, количество инцидентов безопасности.

- Удобство использования: Оценка удобства интерфейса пользователями.

Методы анализа качества:

- Тестирование на надежность и отказоустойчивость.

- Производительность и нагрузочное тестирование.

- Безопасное тестирование, включая тестирование на проникновение.

- Оценка удобства использования на основе обратной связи пользователей.

4. Планирование перспектив создания АСУ

Основные концепции технологий разработки:

- Используемые технологии: Java для клиентской части, Spring Framework для серверной части, MySQL для базы данных.

- Подход к разработке: Agile с итеративным развитием и частыми релизами.

Требования к программным средствам:

- Среда разработки: IntelliJ IDEA для разработки на Java.

База данных по автоматизации и контролю работ:

- Разработка базы данных для отслеживания задач и прогресса разработки.

- Использование систем мониторинга и отчетности для контроля качества кода и системы в целом.

Оформление пояснительной записки:

- Подробное описание архитектурных решений и технологий.

- Технические требования и спецификации.

- План внедрения и тестирования системы.

- Документация по эксплуатации и поддержке системы.

5. Эскиз пользовательского интерфейса АИС

Эскизы пользовательского интерфейса представлены в разделе 4.6 «Эскизы основных окон системы»