Министерство образования Московской области

Государственное бюджетное профессиональное образовательное учреждение

Московской области

«Сергиево-Посадский колледж»

КУРСОВОЙ ПРОЕКТ

по МДК 05.01 Проектирование и дизайн информационных систем

ПМ.05 Проектирование и разработка информационных систем

РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ ВЗАИМОДЕЙСТВИЯ С КЛИЕНТАМИ

Разработал: Авраменко.С.С.

Группа: ИСП-019

Специальность: 09.02.07

«Информационные системы и программирование»

Руководитель:

преподаватель спец. дисциплин

\_\_\_\_\_\_\_\_\_\_\_\_\_ Карцева М.С

Работа защищена с оценкой

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

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

г. Сергиев Посад

2024

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc3206758)

[1 Основная часть 4](#_Toc3206759)

[1.1 Цель разработки 4](#_Toc3206760)

[1.2 Описание предметной области 4](#_Toc3206761)

[1.3 Используемое программное обеспечение](#_Toc3206775) 6

[2 Технологическая часть 7](#_Toc3206777)

[2.1 Постановка задачи](#_Toc3206778)  [7](#_Toc3206779)

[2.2 Проектирование 8](#_Toc3206780)

[2.3 Руководство оператора 33](#_Toc3206782)

[2.4 Руководство программиста 36](#_Toc3206788)

[2.5 Тестирование и отладка 37](#_Toc3206792)

[Заключение 41](#_Toc3206818)

[Библиография 42](#_Toc3206819)

[Приложение А 44](#_Toc3206821)

[Приложение Б 52](#_Toc3206822)

**Введение**

В данном курсовом проекте рассматривается тема «Разработка клиентской части информационной системы для автоматизации взаимодействия с клиентами». На данный момент существует множество компаний, предоставляющих различные услуги для клиентов. Эффективное взаимодействие с ними является ключевым аспектом успешной работы любой организации и способствует дальнейшему привлечению заинтересованных в предоставляемых услугах лиц. В настоящее время чаще всего используются малоавтоматизированные или вовсе не имеющие автоматизации способы взаимодействия с клиентами. К ним можно отнести: ведение записей на бумажных или электронных носителях, использование чатов для оповещения клиентов, а также социальных сетей для предоставления информации о деятельности компании. Каждый из перечисленных вариантов имеет ограничения, не позволяющие в полной мере обеспечить удобство отслеживания хода оказания услуги клиенту и взаимодействие с ним.

Решением проблемы может стать реализация системы, позволяющей автоматизировать бизнес-процессы организации. Это позволит сократить используемые при работе сервисы, упростить отслеживание и взаимодействие с клиентами, а также структуризировать требуемую для работы информацию, что значительно увеличит общую эффективность компании.

1 Основная часть

1.1 Цель разработки

Программный продукт разрабатывается для следующих целей:

* Автоматизация бизнес-процессов организации;
* Обеспечение удобства работы с клиентами организации;
* Ускорение работы организации;
* Обеспечение организованности информации, требуемой для работы компании.

1.2 Описание предметной области

**Основная деятельность компании** – разработка мобильных приложений.

**Структура организации** включает в себя несколько отделов, контролируемых главами: отдел разработки, финансовый отдел и отдел по работе с клиентами. На рисунке 1 представлена общая структура компании:



Рисунок 1 – Общая структура компании

**Отдел разработки** занимается проектированием и реализацией продукта.

**Финансовый отдел** занимается ведением бухгалтерского и налогового учета.

**Отдел по работе с клиентами** занимается вопросами привлечения взаимодействия с клиентами.

На данный момент в деятельности организации существуют следующие проблемы:

* Распространение информации о компании и привлечение заинтересованных клиентов
* Организация взаимодействия с клиентами
* Отслеживание и контроль за оказанием услуг клиентам
* Планирование дальнейших действий компании

Основные процессы:

**Предоставление информации о деятельности компании, ее услугах и контактах для связи.**

Информация распространяется посредством рекламы.

Предполагается решение проблемы за счет разработки клиентской части сайта, содержащего информацию о компании и ее услугах.

**Обеспечение обратной связи для людей, заинтересованных в предоставляемых услугах.**

Обратная связь обеспечивается за счет телефонного звонка, электронной почты, мессенджеров.

Предполагается решение проблемы за счет размещения на сайте формы обратной связи.

**Предоставление клиенту информации о текущем состоянии проекта.**

Клиент может узнать о текущем состоянии проекта: лично посетив офис, позвонив в компанию, отправив сообщение по электронной почте или в мессенджер.

Предполагается решение проблемы за счет разработки личного кабинета клиента с возможностью просмотра информации о проекте и комментариев по процессу разработки.

**Оповещение клиента об изменении состояния процесса оказания услуги.**

Клиент получает информацию посредством телефонного звонка или личного посещения офиса.

Предполагается решение проблемы за счетотображения информации о состоянии проекта в личном кабинете, отправки автоматизированного сообщения об изменении на электронную почту и/или телефонного звонка.

**Оповещение клиента о необходимости обратной связи для уточнения деталей процесса разработки.**

Клиент получает информацию посредством телефонного звонка или личного посещения офиса.

Предполагается решение проблемы за счет отображения информации о необходимости уточнений по процессу разработки в личном кабинете пользователя, а также отправки автоматизированного сообщения на электронную почту.

**Отслеживание, управление проектами и планирование дальнейших действий компании.**

В организации нет системы отслеживания проектов, что усложняет планирование.

Предполагается решение проблемы за счет разработки личного кабинета администратора с возможностью просмотра списка клиентов и связанных с ними проектов, а также управления ими.

1.3 Используемое программное обеспечение

**Figma** (сервис для проектирования интерфейсов и прототипов. Оно позволяет создавать дизайн макеты, проводить интерактивное тестирование пользовательских интерфейсов, а также работать в команде в реальном времени);

**Visual Studio Code** (текстовый редактор с открытым исходным кодом, поддерживает множество языков программирования, включая HTML, CSS и JavaScript);

**Mozilla Firefox** (популярный веб-браузер с открытым исходным кодом. Он широко используется разработчиками благодаря мощным инструментам для отладки, анализу кода, проверки адаптивности интерфейсов и тестирования веб-приложений);

**draw.io** (веб-приложение для создания диаграмм, схем и прототипов. В нем предоставлены удобные инструменты для визуализации данных, разработки архитектуры приложений и схем взаимодействия компонентов системы);

**Git (**система контроля версий, которая используется для управления изменениями в коде и совместной работы над проектами. Она позволяет разработчикам сохранять историю изменений, отслеживать и возвращаться к предыдущим версиям кода, а также работать над проектами в командах**)**;

**GitHub** (платформа для размещения репозиториев Git и упрощения совместной работы над проектами).

2 Технологическая часть

2.1 Постановка задачи

В рамках курсового проекта под автоматизации подлежит бизнес - процесс взаимодействия с клиентами компании, предоставляющей услуги по разработке программного обеспечения для мобильных устройств.

Объект автоматизации включает в себя набор сущностей:

**Пользователь**: просматривает главную страницу сайта с информацией о компании и может отправить заявку на обратную связь;

**Клиент**: имеет список проектов, разрабатываемых для него компанией, может дать оценку завершенному этапу проекта;

**Администратор**: имеет список заявок и клиентов, может управлять заявками, регистрировать и управлять клиентами, а также добавлять и управлять проектами клиентов.

Автоматизация направлена на упрощение и ускорение процесса взаимодействия между пользователями, клиентами и администраторами компании. Основная цель — минимизация ручного труда, исключение ошибок при обработке информации, а также обеспечение удобного доступа к данным для всех участников процесса.

2.2 Проектирование

2.2.1 Структура информационной системы

Далее рассмотрена детальная структура разрабатываемой части информационной системы.

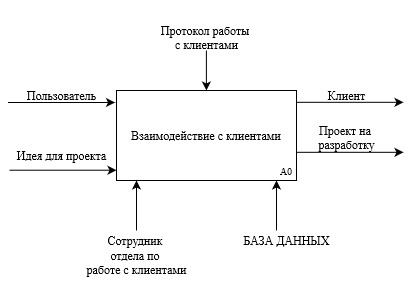


Рисунок 2 – Диаграмма IDEF0

На рисунке 2 изображена диаграмма IDEF, которая иллюстрирует процесс взаимодействия с клиентами на уровне организации. В центре диаграммы находится основной процесс — «Взаимодействие с клиентами». Этот процесс связывает несколько внешних сущностей и источников данных:

Пользователь — предоставляет идеи для проекта, которые становятся входными данными для процесса разработки.

Клиент — участвует в процессе, предоставляя информацию о проекте и оценивая этапу его реализации.

Сотрудник отдела по работе с клиентами — обеспечивает взаимодействие между клиентом и командой разработки.

База данных — используется для хранения информации и данных, связанных с клиентами и проектами.

Протокол работы с клиентами — документ, определяющий общие принципы для работы с клиентами.

Проект на разработку — конечный продукт взаимодействия, который передается клиенту после завершения разработки

Диаграмма демонстрирует структуру взаимодействия между различными участниками процесса и информационными потоками в рамках взаимодействия с клиентами.

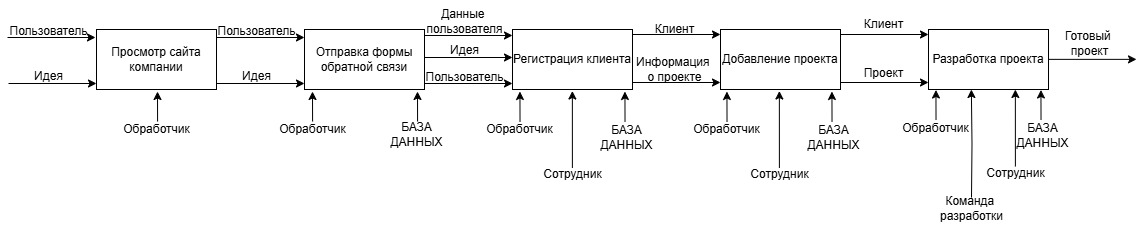


Рисунок 3 – Диаграмма IDEF1

На рисунке 3 представлена расширенная диаграмма IDEF процесса взаимодействия с клиентами, отображающая пошаговое выполнение операций и потоки данных между различными участниками.

Пользователь начинает процесс, просматривая сайт. Далее он может отправить форму обратной связи с интересующей его информацией, после чего данные поступают в систему. Сотрудник обрабатывает данные и связывается по указанным контактным данным. В случае, если Пользователя заинтересовали услуги, он становиться Клиентом. Клиент предоставляет более детализированную информацию о проекте, которую получает сотрудник. Эти данные включают подробности, необходимые для создания и разработки проекта. После добавления проекта в Базу Данных и уточнения деталей, сотрудник передает проект на разработку. Далее происходит отслеживание статуса проекта. В определенные моменты клиент может предоставлять обратную связь о завершении этапов разработки. Если требуется, сотрудник может вносить изменения на основе полученной обратной связи. По окончании всех этапов, проект завершается, и сотрудник передает готовый продукт клиенту. Диаграмма демонстрирует взаимодействие между пользователем, клиентом, сотрудником и базой данных на всех этапах работы над проектом, начиная с поступления запроса и до завершения проекта.

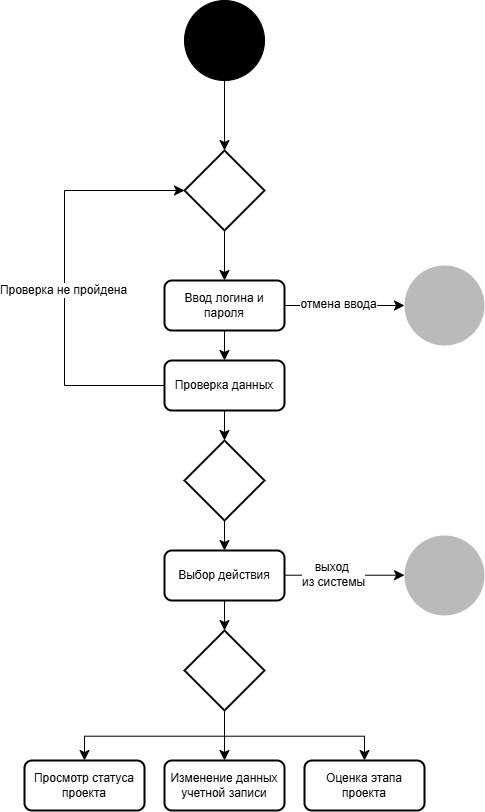


Рисунок 4 – Диаграмма DFD для клиента

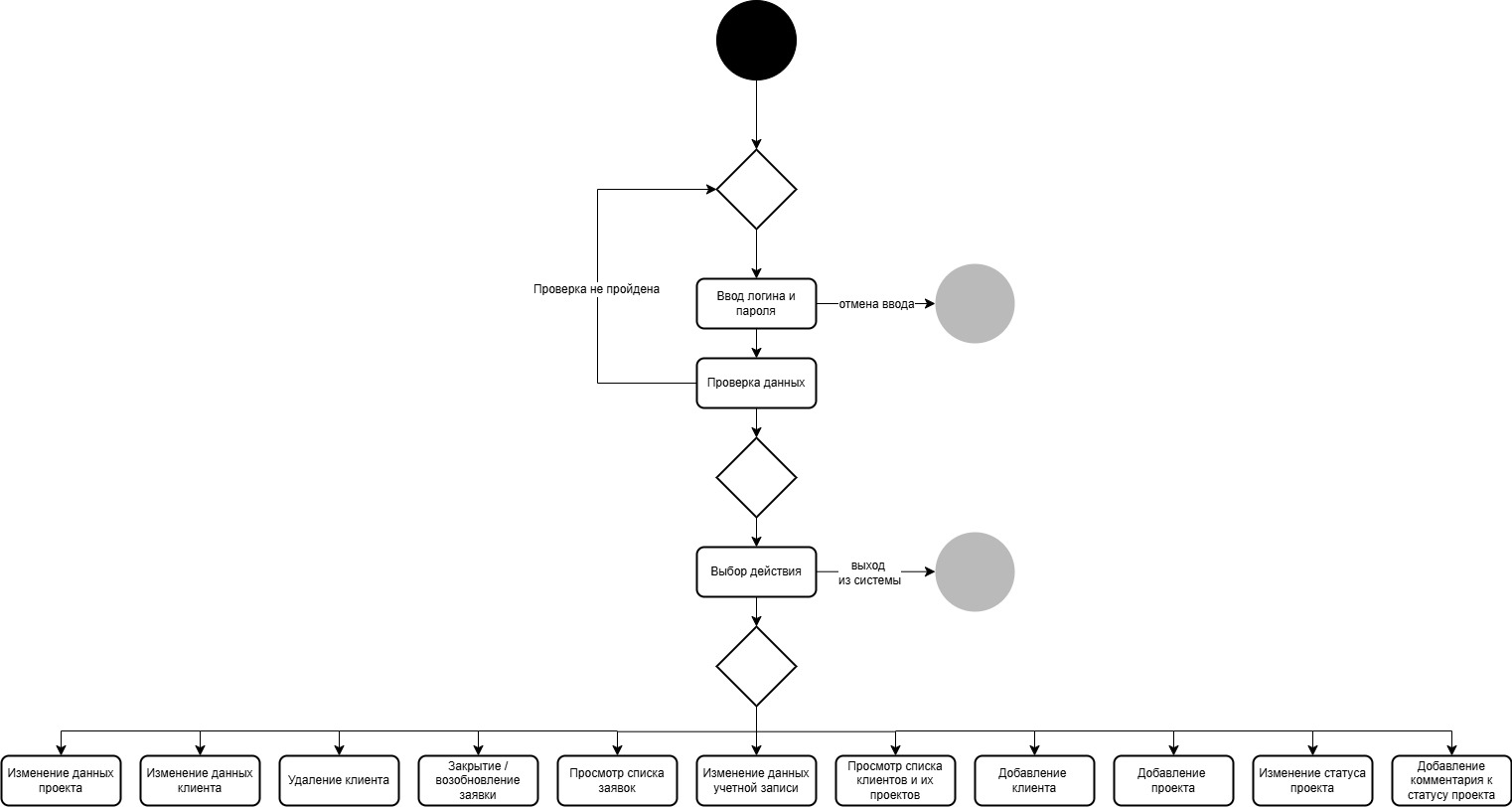


Рисунок 5 – Диаграмма DFD для администратора

На рисунке 4 представлена диаграмма потоков данных (DFD), описывающая процесс взаимодействия пользователя с системой:

Начало процесса — пользователь пытается войти в систему (черный круг).

Ввод логина и пароля — система запрашивает у пользователя учетные данные. Если проверка не пройдена (например, данные введены неправильно), пользователь возвращается на этап ввода данных. Если данные верны, происходит их проверка.

Выбор действий — после успешной проверки данных пользователю предоставляется выбор из следующих опций:

Просмотр статуса проекта, где пользователь может увидеть текущее состояние проекта.

Изменение данных учетной записи, что позволяет пользователю изменить свои персональные данные.

Оценка этапа проекта, где пользователь может дать оценку текущему этапу выполнения проекта.

В случае отмены ввода или выхода из системы происходит завершение процесса.

На рисунке 5 представлена диаграмма DFD для администратора системы:

Ввод учетных данных — аналогично пользователю, администратор должен пройти этап ввода логина и пароля. Если данные не прошли проверку, администратор возвращается на этап ввода.

Выбор действий администратора — после успешного входа администратору предоставляется доступ к функционалу системы:

* Управление проектами — администратор может добавлять новые проекты, редактировать существующие и удалять завершенные или неактуальные проекты. Он также может отслеживать статус каждого проекта и получать уведомления о важных изменениях;
* Управление пользователями — эта функция позволяет администратору управлять учетными записями пользователей.

Он может:

* Добавлять новых пользователей (как клиентов, так и других администраторов);
* Удалять пользователей, которые больше не нуждаются в доступе;
* Изменять данные учетных записей, такие как логин, пароль и личные данные;
* Просмотр статусов проектов — администратор может видеть все текущие проекты и их статусы, включая комментарии и историю изменения статусов. Это позволяет отслеживать прогресс и оперативно реагировать на задержки или проблемы в реализации проектов;
* Просмотр данных клиентов — администратор имеет доступ к информации о клиентах, включая их контактные данные, предпочтения по коммуникации и список проектов, в которых они участвуют. Эта функция позволяет поддерживать актуальную базу клиентов;
* Просмотр истории изменений — администратор может отслеживать всю историю изменений, сделанных в системе, включая изменения проектов, учетных записей, статусов и оценок. Это необходимо для контроля над действиями пользователей и поддержания безопасности данных;
* Оценка этапов проекта — администратор может получать уведомления о негативных оценках этапов от клиентов и принимать меры по исправлению ситуаций. В его компетенции находится контроль качества выполнения проектов и коммуникация с командами для улучшения результатов;
* Управление настройками системы — администратор может настраивать общие параметры системы, например, правила аутентификации, протоколы уведомлений, а также базовые настройки безопасности для предотвращения несанкционированного доступа.

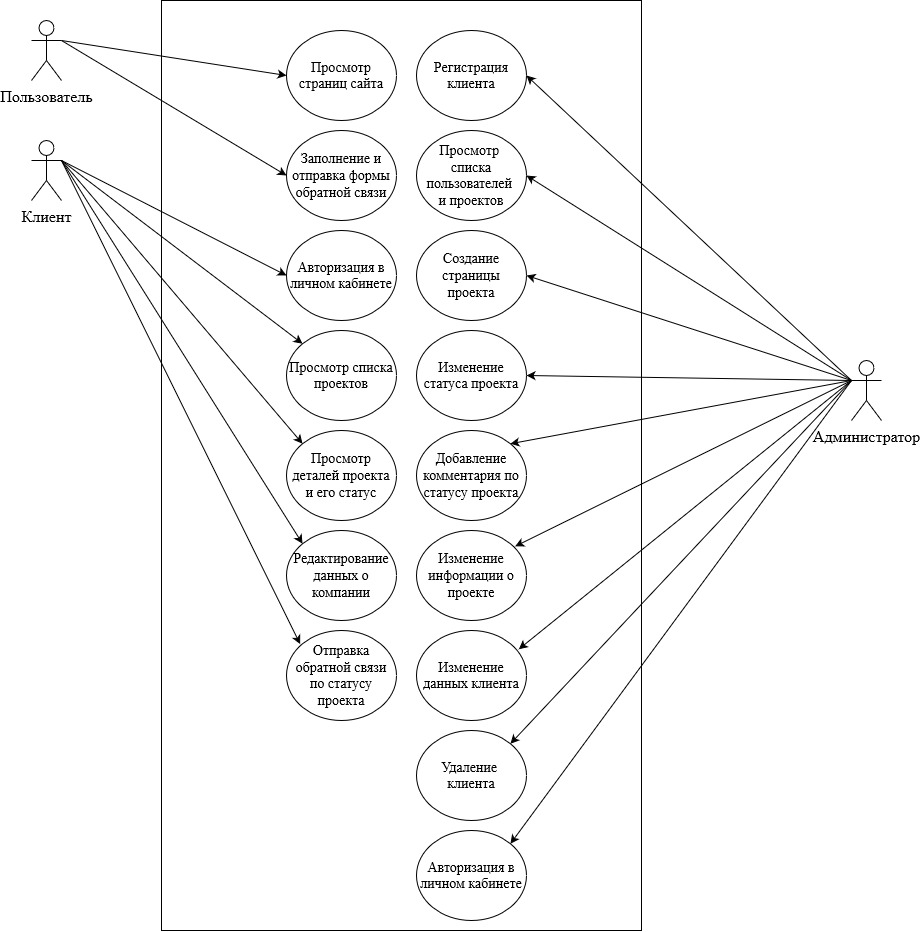


Рисунок 6 – Диаграмма прецедентов

На рисунке 6 изображена диаграмма прецедентов, которая отражает взаимодействие пользователей системы (пользователя, клиента и администратора) с различными функциональными возможностями.

**Пользователь:**

* Просмотр страницы сайта — пользователь может посетить сайт для ознакомления с услугами и информацией о компании;
* Заполнение и отправка формы обратной связи — пользователь может связаться с компанией, заполнив форму обратной связи.

**Клиент:**

* Авторизация в личном кабинете — клиент может войти в систему через свой личный кабинет;
* Просмотр списка проектов — клиент имеет возможность просматривать список всех проектов, связанных с его аккаунтом;
* Просмотр деталей проекта и его статуса — клиент может видеть информацию о текущем состоянии и статусе каждого проекта;
* Редактирование данных о компании — клиент может изменять информацию о своей компании, которая хранится в системе;
* Отправка обратной связи по статусу проекта — клиент может оставлять обратную связь по каждому этапу выполнения проекта.

**Администратор:**

* Регистрация клиента — администратор может добавить нового клиента в систему;
* Просмотр списка пользователей и проектов — администратор имеет доступ к информации обо всех клиентах и проектах;
* Создание страницы проекта — администратор отвечает за создание нового проекта на основе данных, предоставленных клиентом;
* Изменение статуса проекта — администратор обновляет текущий статус проекта в зависимости от его прогресса;
* Добавление комментария по статусу проекта — администратор может добавлять комментарии, связанные с изменениями статуса проекта;
* Изменение информации о проекте — администратор редактирует ключевые данные проекта, такие как сроки, цели и задачи;
* Изменение данных клиента — администратор может обновить информацию о клиенте в случае необходимости;
* Удаление клиента — администратор имеет возможность удалить клиента из системы, если это потребуется;
* Авторизация в личном кабинете — администратор также может входить в систему для управления проектами и клиентами.
* Диаграмма демонстрирует, как каждый тип пользователя взаимодействует с системой, и какие функциональные возможности доступны каждому из них.

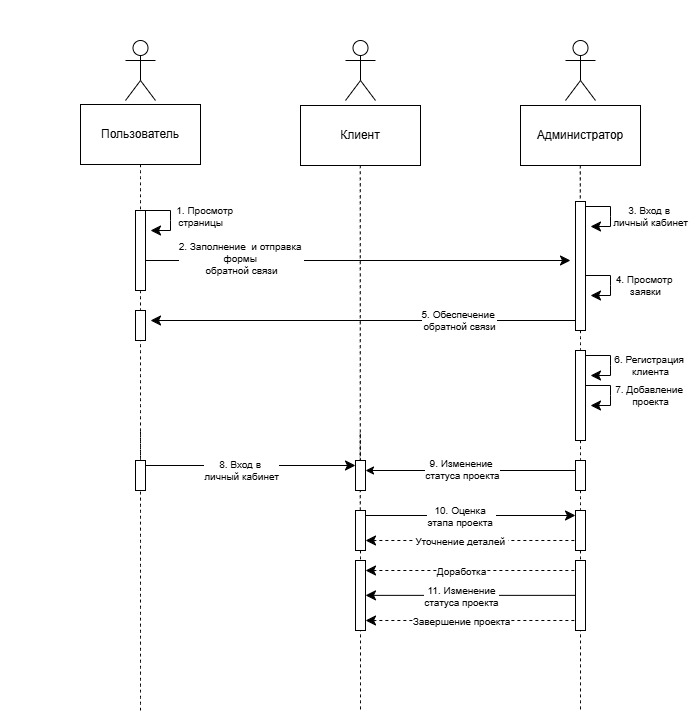


Рисунок 7 – Диаграмма последовательности действий

На рисунке 7 изображена диаграмма последовательности действий между пользователем, клиентом и администратором в рамках процесса разработки проекта.

**Пользователь:**

* Просмотр страницы — пользователь заходит на сайт и изучает предоставленную информацию;
* Заполнение и отправка формы обратной связи — если пользователь заинтересован, он заполняет форму для связи с компанией.

**Администратор:**

* Вход в личный кабинет — администратор авторизуется в системе;
* Просмотр заявки — получает и обрабатывает запрос, отправленный пользователем;
* Регистрация клиента — после обработки заявки администратор регистрирует нового клиента, если он ранее не был зарегистрирован;
* Добавление проекта — создает проект на основе данных, полученных от клиента, и передает его на разработку.

**Клиент:**

* Обеспечение обратной связи — после отправки формы клиент предоставляет необходимые данные о компании и проекте для начала работы/

**Взаимодействие клиента и администратора:**

* Вход клиента в личный кабинет — клиент авторизуется в системе, чтобы отслеживать статус проекта;
* Изменение статуса проекта — администратор обновляет статус проекта по мере выполнения этапов разработки;
* Оценка этапа проекта — клиент оценивает выполненный этап работы, предоставляя обратную связь. Если есть замечания, клиент уточняет детали, что может привести к доработке проекта;
* Изменение статуса проекта — после доработок статус проекта снова изменяется администратором, и работа продолжается;
* Завершение проекта — после успешного завершения всех этапов и их подтверждения проект закрывается.

Диаграмма демонстрирует этапы взаимодействия между всеми участниками процесса, начиная с первичного обращения пользователя и заканчивая завершением проекта через оценку его выполнения клиентом.

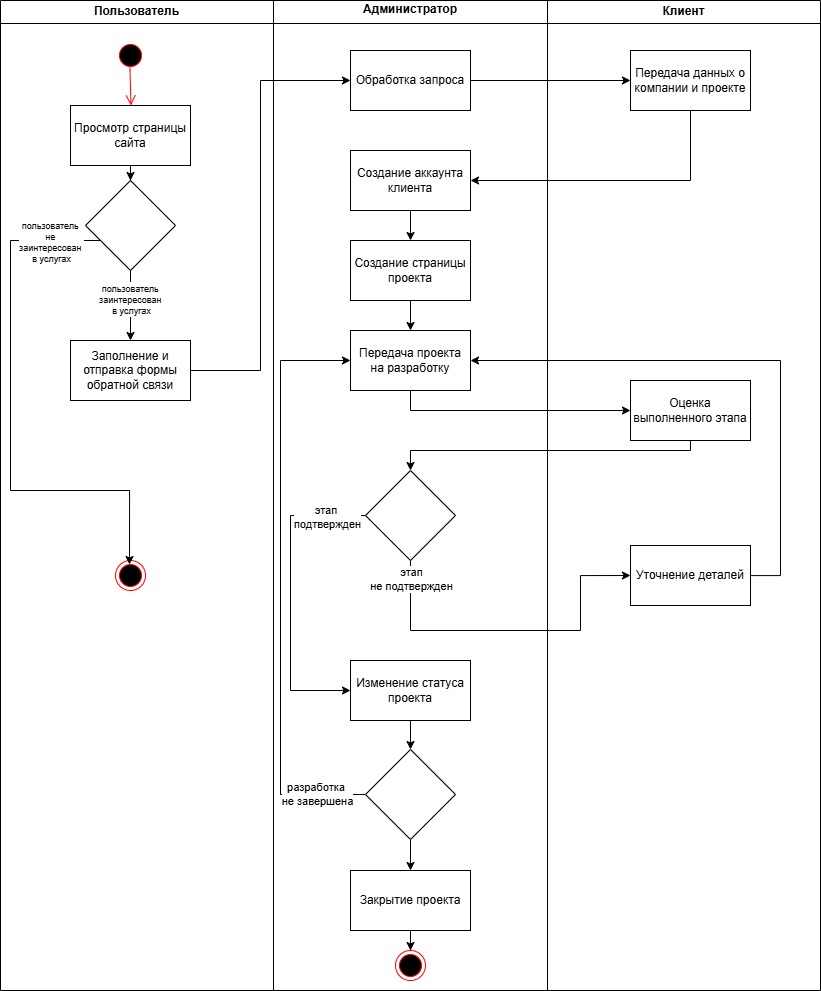


Рисунок 8 – Диаграмма деятельности

На рисунке 8 представлена диаграмма деятельности, отражающая взаимодействия между пользователем, администратором и клиентом в процессе выполнения проекта.

**Пользователь:**

* Начинает с просмотра страницы сайта;
* Если пользователь заинтересован в услугах, он заполняет и отправляет форму обратной связи для запроса услуги.

**Администратор:**

* После получения запроса обрабатывает его, что является первым этапом взаимодействия;
* Создает аккаунт клиента, если клиент новый;
* Создает страницу проекта, на которой будут отражены основные данные и прогресс работы;
* Передает проект на разработку, после чего начинается выполнение этапов проекта.

**Клиент:**

* Передает данные о своей компании и проекте, необходимые для начала работы;
* Оценивает выполненные этапы разработки. Если этап выполнен успешно, он подтверждается;
* Если этап не подтвержден, клиент уточняет детали и предоставляет обратную связь для исправления ошибок или доработки;
* В случае неполного завершения этапа или необходимости доработок, статус проекта изменяется;
* Если проект все еще находится в стадии разработки, он продолжает работу над проектом, обновляя статус;
* После завершения всех этапов и подтверждения выполнения, проект закрывается.

Диаграмма иллюстрирует последовательность действий между всеми участниками процесса: пользователь направляет запрос, администратор организует работу, а клиент оценивает качество выполнения этапов, что определяет дальнейший ход разработки.

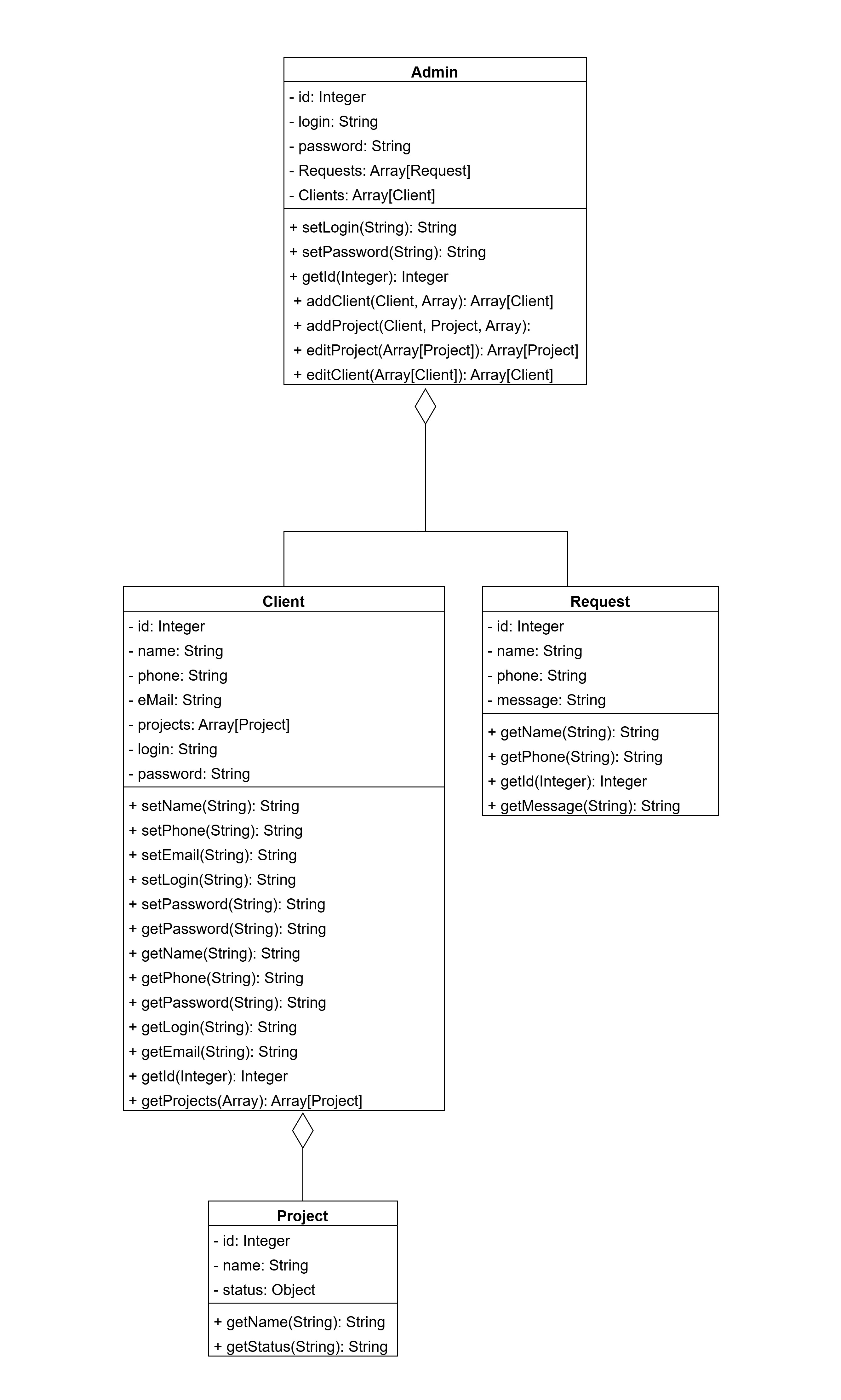


Рисунок 9 – Диаграмма классов

На рисунке 9 представлена диаграмма классов, описывающая структуру системы взаимодействия между администраторами, клиентами, запросами и проектами.

**Admin (Администратор):**

* Класс описывает администратора, который имеет такие поля, как id, login, password, а также списки запросов (Requests) и клиентов (Clients).
* Методы включают:
* setLogin(String): Устанавливает логин администратора.
* setPassword(String): Изменяет пароль администратора.
* getId(Integer): Возвращает идентификатор администратора.
* Работа с клиентами и проектами: методы добавления клиентов и проектов, редактирования информации о клиентах.
* Client (Клиент): класс клиента содержит поля для хранения: идентификатора (id), имени (name), телефона (phone), email (email), списка проектов (projects), логина (login) и пароля (password).

Методы:

* Установление и получение информации, такие как setName, setPhone, setEmail, setLogin, setPassword, а также методы для получения данных (getId, getName, getPhone);
* Работа с проектами: getProjects(), который возвращает список проектов, ассоциированных с клиентом.

**Request (Запрос):**

* Описывает запросы, которые клиенты могут оставлять. Включает поля: идентификатор (id), имя (name), телефон (phone) и сообщение (message).

Методы:

* getName, getPhone, getMessage: получение данных из запроса;
* getId: возвращает идентификатор запроса.

**Project (Проект):**

* Класс, представляющий проект. Поля включают: идентификатор (id), название проекта (name), статус (status).

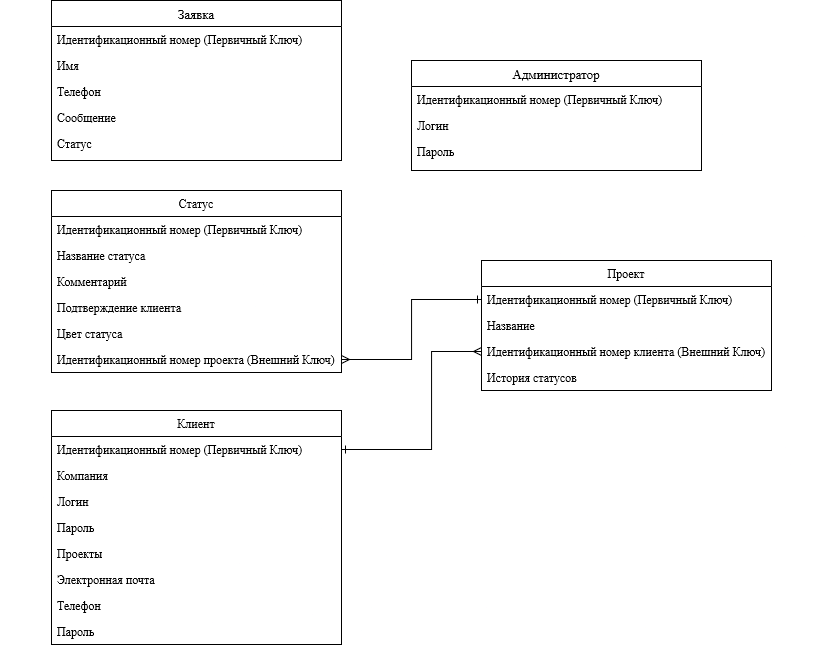
Методы:

* getName, getStatus: методы для получения имени и статуса проекта.

Связи между классами:

* Класс Admin ассоциирован с классами Client и Request, что указывает на то, что администратор управляет запросами и клиентами;
* Класс Client связан с классом Project, указывая на то, что у клиента может быть множество проектов;
* Диаграмма показывает взаимодействие между различными сущностями системы и методы, которые используются для их управления.

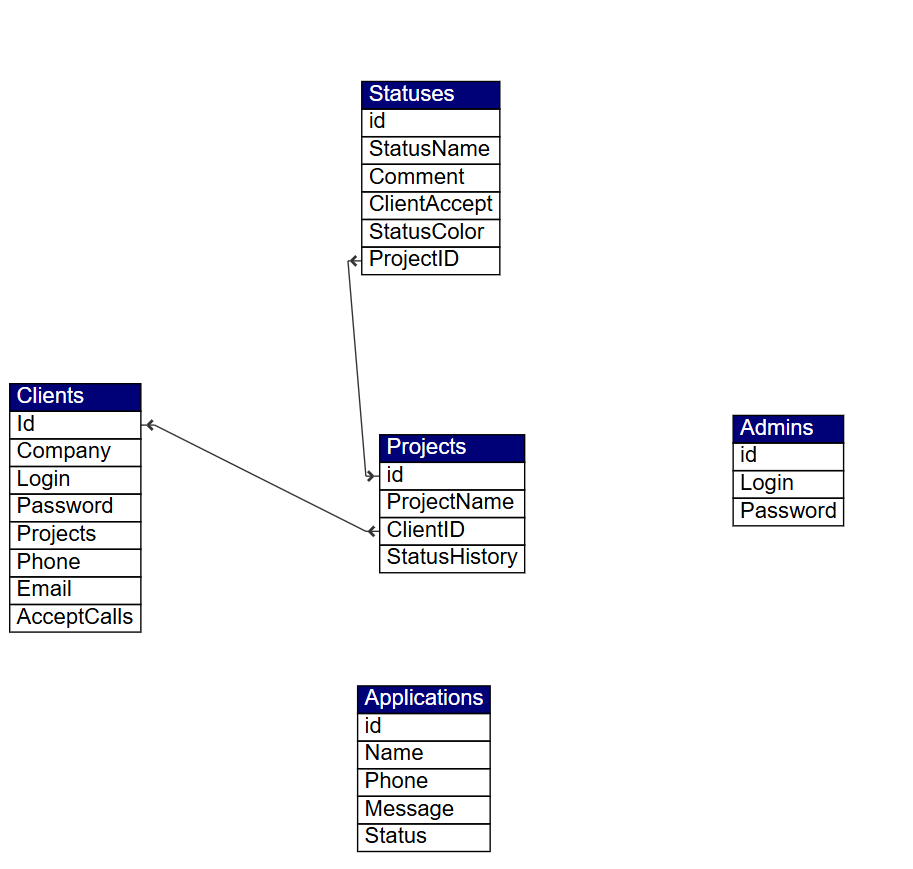
2.2.2 Структура организации данных



## Рисунок 10 – ER диаграмма

На рисунке 10 представлена ER диаграмма. Она соответствует Третьей Нормальной Форме (3НФ) исходя из следующих условий:

* Первая нормальная форма (1НФ): **атомарность атрибутов** (каждое поле таблицы должно содержать только одно значение), **уникальность строк** (каждая строка в таблице должна быть уникальной);
* Вторая нормальная форма (2НФ): **соответствие 1НФ** (таблица должна быть в 1НФ), **отсутствие частичных зависимостей** (все неключевые атрибуты должны зависеть от всего первичного ключа, а не от его части);
* Третья нормальная форма (3НФ): **cоответствие 2НФ** (таблица должна быть в 2НФ), **отсутствие транзитивных зависимостей** (никакой неключевой атрибут не должен зависеть от другого неключевого атрибута).



## Рисунок 11 – реализация Базы Данных

На рисунке 11 представлена реализованная База Данных, имеющая следующую структуру:

**Заявки (Applications):**

* **id** - Уникальный идентификатор заявки;
* **Имя (name)** - Имя человека, который оставил заявку;
* **Телефон (phone\_number)** - Контактный номер телефона заявителя;
* **Сообщение (message)** - Текстовое сообщение или запрос от заявителя;
* **Статус (status)** - Статус заявки.

**Проекты (Projects):**

* **id** - Уникальный идентификатор проекта;
* **Название (name)** - Название проекта;
* **id Клиента (client\_id)** - Внешний ключ, ссылающийся на таблицу **Клиенты**, указывающий на клиента, которому принадлежит проект;
* **История статусов (status\_history)** - Ссылка на историю статусов проекта.

**Статусы (Statuses):**

* **id** - Уникальный идентификатор статуса;
* **Название статуса (status\_name)** - Название текущего статуса проекта;
* **Комментарий (comment)** - Дополнительная информация или описание текущего статуса;
* **Подтверждение клиента (client\_evaluation)** - Флаг, который указывает, подтвердил ли клиент этот статус (1 — подтверждено, 0 — не подтверждено);
* **Цвет статуса (color)** - Цвет, ассоциированный с этим статусом;
* **id Проекта (project\_id)** - Внешний ключ, ссылающийся на таблицу **Проекты**, который указывает, к какому проекту относится статус.

**Клиенты (Clients):**

* **id** - Уникальный идентификатор клиента;
* **Компания (company\_name)** - Название компании клиента;
* **Логин (login)** — Логин для доступа клиента к системе;
* **Пароль (password)** — пароль клиента;
* **Проекты (projects)** - Список проектов, ассоциированных с клиентом;
* **Телефон (phone\_number)** - Контактный номер телефона клиента;
* **Email** - Электронная почта клиента;
* **Прием звонков (accept\_calls)** - Флаг, который указывает, принимает ли клиент звонки (1 — принимает, 0 — не принимает).

**Администраторы (Administrators):**

* **id** - Уникальный идентификатор администратора;
* **Логин (login)** - Логин администратора для доступа к системе;
* **Пароль (password)** – Пароль администратора.

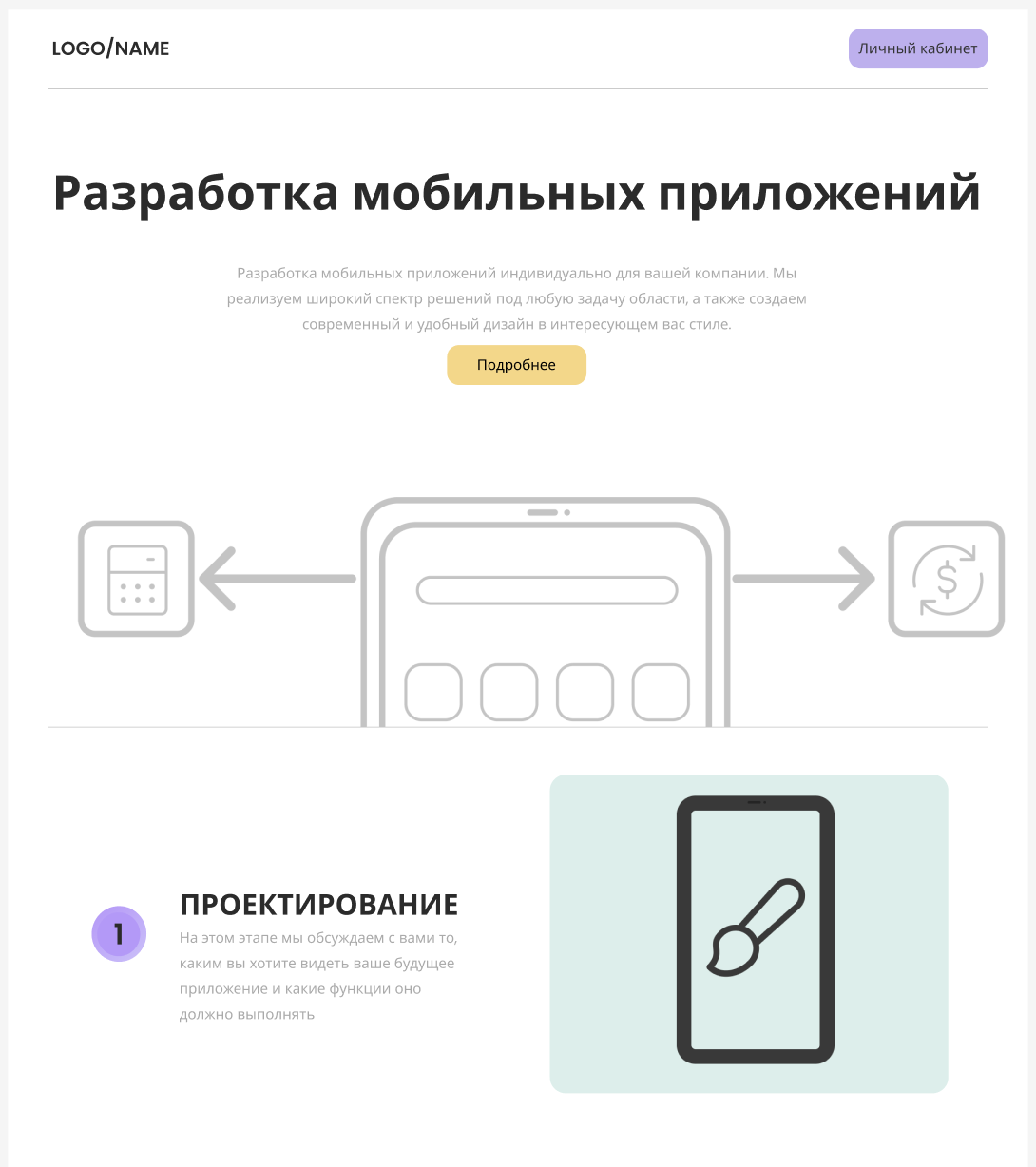
**Связи:**

* **Проекты** связаны с **Клиентами** через поле id Клиента, что указывает, какой клиент владеет проектом;
* **Статусы** связаны с **Проектами** через поле id Проекта, что указывает, к какому проекту относится конкретный статус;
* **История статусов** и поле История статусов в таблице **Проекты** позволяют отслеживать все изменения статусов для каждого проекта.

2.2.3 Стиль и структура дизайна сайта

Дизайн разработан с учетом современных стандартов, сочетающих в себе: минимализм (акцент на ключевую информацию и визуальную чистоту), интерактивность и анимации элементов.

Цветовая палитра состоит преимущественно из светлых оттенков, сочетание которых создает строгий и сдержанный, но при этом открытый стиль. Шрифты подобраны в соответствии с деятельностью организации имеют стиль Sans Serif.



## Рисунок 12 – фрагмент главной страницы сайта

На рисунке 12 представлен фрагмент главной страницы сайта. Структура сайта состоит из:

* Главная страница (содержит информацию о компании, кнопку авторизации в личном кабинете и форму обратной связи);
* Личный кабинет Администратора (содержит списки заявок, клиентов и проектов с возможностью отобразить информацию о каждом проекте, раздел настроек для изменения данных Администратора и кнопку выхода);
* Личный кабинет Клиента (содержит список проектов с возможность просматривать информацию, раздел настроек для изменения данных Клиента и кнопку выхода).

2.2.4 Программные функции

Функция generatePass(), генерирует пароль длиной 8 символов, который включает:минимум 2 цифры, минимум 1 специальный символ, другие символы могут быть любыми из набора:

function generatePass() {

const symbols = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&\*()";

const symbolsArr = symbols.split('');

let passSymbols = [];

let nums = 0;

let symbolsCount = 0;

let iterations = 0;

while (passSymbols.length < 8 && iterations < 100) {

let rand = symbolsArr[Math.floor(Math.random() \* symbolsArr.length)];

if (!passSymbols.includes(rand)) {

if (!isNaN(rand) && nums < 2) {

nums++;

passSymbols.push(rand);

} else if (isNaN(rand) && symbolsCount < 1) {

symbolsCount++;

passSymbols.push(rand);

} else {

passSymbols.push(rand);

}

}

iterations++;

}

if (passSymbols.length < 8) {

while (passSymbols.length < 8) {

let rand = symbolsArr[Math.floor(Math.random() \* symbolsArr.length)];

passSymbols.push(rand);

}

}

return passSymbols.join('');

}

Функция CreateClient() создаёт новый объект клиента и добавляет его в локальное хранилище:

function CreateClient(){

let client = {

"Id":`${Math.max(...getData('CLIENTS','Id'))+1}`,

"Company":`${clientAddPopupForm.company.value}`, // Используется в качестве имени клиента.

"Login":`${clientAddPopupForm.login.value}`,

"Password":generatePass(),

"Projects": "0", //количество проектов клиента.

"Phone":`${clientAddPopupForm.phone.value}`,

"Email":`${clientAddPopupForm.email.value}`,

"AcceptCalls":"1"

}

let clients = JSON.parse(localStorage.getItem('CLIENTS'));

clients.push(client);

localStorage.setItem('CLIENTS',JSON.stringify(clients));

}

Функция UpdateFromForm() позволяет обновить записи в таблице, хранящейся в локальном хранилище, на основе данных из формы. Она ищет нужный элемент по его идентификатору, обновляет его поля и сохраняет изменения в таблице:

function UpdateFromForm(tableName, id, form){

const table = getTable(tableName);

let updateElementIndex = 0;

if(form.name === 'projectEdit'){

updateElementIndex = table.findIndex(element => element.id === id);

}

else{

updateElementIndex = table.findIndex(element => element.Id === id);

}

const formData = new FormData(form);

for (const [key, value] of formData.entries()) {

if(key === 'passReset' && form.name === 'clientEdit'){

if (value === 'on') {

table[updateElementIndex]['Password'] = generatePass();

}

}

else{

table[updateElementIndex][key] = value;

}

}

updateTable(tableName, table);

}

2.3 Руководство оператора

Для посетителей

Просмотр информации о компании:

* Перейдите на главную страницу сайта;
* Изучите разделы с информацией о компании и её услугах.

Как связаться с компанией:

* Нажмите кнопку «Связаться с нами» в верхней части страницы (шапке) или в нижнем блоке сайта;
* Заполните форму: укажите своё Имя (обязательное поле), введите контактный Телефон (обязательное поле), при необходимости добавьте Сообщение (поле необязательное);
* Нажмите кнопку «Отправить», чтобы отправить запрос.

Для клиентов

Вход в личный кабинет:

* Нажмите кнопку «Личный кабинет» в верхней части страницы;
* Заполните форму авторизации: укажите Логин и Пароль, установите галочку в поле «Запомнить», если хотите сохранить сессию;
* Нажмите кнопку «Отправить» для входа.

Работа с проектами:

* После входа вы увидите список всех ваших проектов;
* Для просмотра информации о конкретном проекте выберите его из списка.

Действия при изменении статуса проекта:

* Откройте проект, чтобы просмотреть историю изменений статусов;
* Ознакомьтесь с информацией о текущем статусе;
* Для подтверждения этапа нажмите кнопку галочки;
* Если вы хотите отклонить этап, нажмите кнопку крестика, чтобы отправить запрос на пересмотр.

Для администраторов:

Вход в личный кабинет администратора:

* Нажмите кнопку «Личный кабинет» в верхней части страницы;
* Заполните форму авторизации, указав Логин и Пароль;
* Установите галочку «Запомнить», если хотите сохранить сессию;
* Нажмите «Отправить».

Работа с клиентами:

* Нажмите кнопку «Клиенты» для просмотра списка зарегистрированных клиентов.

Для добавления клиента:

* Нажмите кнопку «Добавить клиента»;
* Заполните форму с данными нового клиента;
* Нажмите «Добавить».

Для редактирования клиента:

* Нажмите на значок карандаша рядом с нужным клиентом;
* Обновите данные и при необходимости установите галочку «Сброс пароля»;
* Нажмите «Сохранить».

Для удаления клиента:

* Нажмите на значок корзины рядом с нужным клиентом.

Работа с заявками:

* Нажмите кнопку «Заявки», чтобы открыть список входящих запросов.

Если есть новые заявки:

* Выберите заявку для просмотра;
* Нажмите на значок стрелки вниз для просмотра сообщения;
* Свяжитесь с клиентом по указанному номеру;
* После завершения обработки заявки нажмите галочку, чтобы закрыть её.

Управление проектами:

Для изменения статуса проекта:

* Откройте список проектов, нажав стрелку вниз;
* Выберите проект, щёлкнув на его ID или название;
* Заполните форму изменения статуса (Название статуса, Описание, Цвет);
* Нажмите «Изменить статус».

Для редактирования проекта:

* Нажмите значок карандаша рядом с нужным проектом;
* Внесите изменения в поля Название проекта и/или Описание;
* Нажмите «Сохранить».

Для добавления нового проекта:

* Нажмите стрелку вниз в списке проектов;
* Заполните форму с данными нового проекта;
* Нажмите «Добавить».

Управление настройками администратора

* Нажмите на значок шестерёнки в верхней части страницы;
* Внесите изменения в настройки;
* Нажмите «Сохранить» для сохранения изменений.

2.4 Руководство программиста

2.4.1 Файловая структура:

* CSS (содержит файлы таблиц стилей);
* fonts (содержит файлы подключенных шрифтов);
* icons (содержит файлы иконок, используемых в системе);
* img (содержит файлы изображений, используемых в системе);
* js (содержит файлы JavaScript с программным кодом, проект имеет модульную систему: **main.js –** основные функции, **DataBase.js –** функции для работы с Базой Данных, **validation.js –** функции для валидации полей, **auth.js –** функции для системы авторизации, **register.js –** функции для системы регистрации, **admin.js –** функции для личного кабинета администратора), **client.js –** функции для личного кабинета пользователя, **project.js –** функции для работы с проектом);
* admin.html (страница для личного кабинета администратора);
* client.html (страница для личного кабинета клиента);
* index.html (главная страница сайта).

2.5 Тестирование и отладка

Для проверки правильности работы системы, проведем несколько тест-кейсов:

**Заполнение и отправка заявки на обратную связь:**

Таблица 1 – тест функционала отправки заявки на обратную связь

| № | Действие | Ожидаемый результат | Фактический результат |
| --- | --- | --- | --- |
| 1 | Двойное нажатие по кнопке «Связаться с нами» | Закрытие формы | Соответствует ожидаемому |
| 2 | Случайное закрытие формы | Сохранение значений формы | Соответствует ожидаемому |
| 3 | Отправка формы с пустыми полями | Выдача ошибки для незаполненных полей, форма не отправлена | Соответствует ожидаемому |
| 4 | Отправка формы с полями только из пробелов | Выдача ошибки для незаполненных полей, форма не отправлена | Выдача ошибки только для поля «Телефон» |
| 5 | Отправка формы с полями только из пробелов и заполнением полей с ошибками | Выдача ошибки для незаполненных полей, форма не отправлена | Форма отправлена с пустым значением Имя и стандартным значением поля Сообщение |
| 6 | Ввод некорректного значения в поле Телефон | Выдача ошибки | Соответствует ожидаемому |
| 7 | Повторное нажатие кнопки «Отправить» | Нажатие невозможно | Соответствует ожидаемому |
| 8 | Самостоятельное закрытие формы сразу после нажатия кнопки «Отправить» | Форма отправлена, значения не изменены | Соответствует ожидаемому |
| 9 | Изменение значений полей формы сразу после нажатия кнопки «Отправить» | Форма отправлена, значения не изменены | Соответствует ожидаемому |

Продолжение таблицы 1

**Форма авторизации:**

Таблица 2 – тест функционала формы авторизации

| № | Действие | Ожидаемый результат | Фактический результат |
| --- | --- | --- | --- |
| 1 | Двойное нажатие по кнопке «Личный кабинет» | Закрытие формы | Соответствует ожидаемому |
| 2 | Случайное закрытие формы | Сохранение значений формы | Соответствует ожидаемому |
| 3 | Отправка формы с пустыми полями | Выдача ошибки для незаполненных полей, форма не отправлена | Соответствует ожидаемому |
| 4 | Отправка формы с полями только из пробелов | Выдача ошибки для незаполненных полей, форма не отправлена, | Соответствует ожидаемому |
| 5 | Отправка формы с полями только из пробелов и заполнением полей с ошибками | Выдача ошибки для незаполненных полей, форма не отправлена | Форма не отправлена, но ошибка для поля Логин не выдана. Для поля Пароль учитываются пробелы |
| 6 | Ввод в поля Логин и Пароль несуществующих данных для авторизации | Выдача ошибки «Данные не верны» | Соответствует ожидаемому |
| 7 | Ввод в поле Логин или Пароль несуществующих данных для авторизации | Выдача ошибки «Данные не верны» | Соответствует ожидаемому |
| 8 | Повторное нажатие кнопки «Отправить» | Нажатие невозможно | Соответствует ожидаемому |
| 9 | Самостоятельное закрытие формы сразу после нажатия кнопки «Отправить» | Форма отправлена, значения не изменены | Соответствует ожидаемому |
| 10 | Изменение значений полей формы сразу после нажатия кнопки «Отправить» | Форма отправлена, значения не изменены | Соответствует ожидаемому |
| 11 | Авторизация без выхода из текущего аккаунта | Переход в личный кабинет ранее авторизованного пользователя | Соответствует ожидаемому |

Продолжение таблицы 2

Заключение

В ходе курсового проекта была разработана клиентская часть информационной системы для взаимодействия с клиентами. Проект реализован в соответствии с Техническим Заданием (Приложение А).

В рамках дальнейшей поддержки возможно улучшение главной страницы сайта, безопасности Программного Обеспечения, а также разработка и внедрение расширенной системы обмена сообщениями между Клиентом и Администратором. Это дополнительно может повысить скорость работы компании и сократить использование сторонних сервисов.

На текущий момент система требует исправления выявленных ошибок и проведения дополнительного тестирования. После проверки стабильности возможно её внедрение в рабочую среду организации.

Библиография

1. Современный учебник JavaScript [Электронный ресурс]. – <https://learn.javascript.ru/> (дата обращения – 20.10.2024);
2. MDN Web Docs: JavaScript [Электронный ресурс]. – <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения – 23.10.2024);
3. MDN Web Docs: HTML [Электронный ресурс]. – <https://developer.mozilla.org/ru/docs/Web/HTML> (дата обращения – 12.10.2024);
4. MDN Web Docs: CSS [Электронный ресурс]. – <https://developer.mozilla.org/ru/docs/Web/CSS> (дата обращения – 14.10.2024);
5. W3Schools: JavaScript [Электронный ресурс]. – https://www.w3schools.com/js/ (дата обращения – 21.10.2024);
6. W3Schools: HTML [Электронный ресурс]. – https://www.w3schools.com/html/ (дата обращения – 20.10.2024);
7. FreeCodeCamp: JavaScript [Электронный ресурс]. – https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/ (дата обращения – 18.10.2024);
8. CSS Tricks [Электронный ресурс]. – <https://css-tricks.com/> (дата обращения – 8.10.2024);
9. DeepL Translator [Электронный ресурс]. – <https://www.deepl.com/> (дата обращения – 6.10.2024);
10. Руководство по GitHub [Электронный ресурс]. – <https://guides.github.com/> (дата обращения – 2.10.2024);
11. Can I Use [Электронный ресурс]. – <https://caniuse.com/> (дата обращения – 20.10.2024);
12. Stack Overflow [Электронный ресурс]. – <https://stackoverflow.com/> (дата обращения – 20.10.2024);
13. CodePen [Электронный ресурс]. – <https://codepen.io/> (дата обращения – 20.10.2024);
14. JSFiddle [Электронный ресурс]. – <https://jsfiddle.net/> (дата обращения – 20.10.2024);
15. Test Cases in Software Testing [Электронный ресурс]. – https://www.guru99.com/test-case.html (дата обращения – 20.11.2024);
16. A List Apart [Электронный ресурс]. – <https://alistapart.com/> (дата обращения – 20.10.2024);
17. DevDocs [Электронный ресурс]. – <https://devdocs.io/> (дата обращения – 6.11.2024);
18. Руководство по современному дизайну интерфейсов [Электронный ресурс]. – https://www.invisionapp.com/inside-design/ (дата обращения – 12.10.2024);
19. Google Material Design [Электронный ресурс]. – <https://material.io/> (дата обращения – 12.10.2024);
20. Nielsen Norman Group: Руководства по UX-дизайну [Электронный ресурс]. – <https://www.nngroup.com/> (дата обращения – 3.10.2024).

**Приложение А**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на курсовой проект**

**Разработка клиентской части информационной системы для взаимодействия с клиентами**

(полное наименование курсового проекта)

**Студента** Авраменко Степана Сергеевича

**группа** ИСП - 019 **специальность** 09.02.07

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc1)

[1 Цель разработки 3](#_Toc2)

[2 Назначение 3](#_Toc3)

[3 Общие требования 3](#_Toc4)

[4 Функциональные требования 3](#_Toc5)

[4.1 Web-страница с информацией о компании 3](#_Toc6)

[4.2 Регистрация и авторизация 4](#_Toc7)

[4.3 Личный кабинет клиента 4](#_Toc8)

[4.4 Личный кабинет администратора 4](#_Toc9)

[4.5 Список клиентов 4](#_Toc10)

[4.6 Список заявок на обратную связь 5](#_Toc11)

[4.7 Возможности администратора 6](#_Toc12)

[4.8 Возможности клиента 6](#_Toc13)

[4.9 Возможности посетителя 6](#_Toc14)

[4.10 Форма обратной связи 6](#_Toc15)

[4.11 Страница проекта (клиент) 7](#_Toc16)

[4.12 Статусы (этапы) проекта 7](#_Toc17)

[4.13 Валидация 7](#_Toc19)

[5 Технологические требования 8](#_Toc20)

[6 Требования к программным средствам 8](#_Toc21)

[7 Требования к техническим средствам 8](#_Toc22)

[8 Требования по надежности 8](#_Toc23)

[9 Эстетические и эргономические требования 8](#_Toc24)

1 Цель разработки

Повышение эффективности бизнес-процессов, основанных на взаимодействии с клиентами компании за счет внедрения средств автоматизации.

2 Назначение

Разрабатываемый продукт должен увеличить распространенность информации о компании, а также предоставить удобный способ для взаимодействия между клиентом и представителем организации.

3 Общие требования

Дизайн-макет главной страницы должен быть разработан с учетом современных стандартов и адаптации для мобильных и десктопных устройств. Текст должен быть SEO-оптимизированным. Разрабатываемая система должна иметь 3 группы пользователей: «**Пользователь**» (любой зашедший на сайт, не являющийся клиентом или администратором и не имеющий личный кабинет), «**Клиент**» (клиент компании, имеющий личный кабинет и список проектов), «**Администратор**» (сотрудник компании, имеющий личный кабинет с возможностями Администратора).

4 Функциональные требования

4.1 Web-страница с информацией о компании и предоставляемых услугах

Страница содержит разделы: шапка (header) с логотипом, меню навигации и кнопкой для входа в личный кабинет, секция с заголовком и краткой информацией об основной деятельности компании, секция с описанием компании, секция с общим описанием процесса разработки проекта, секция с контактной информацией и адресом, подвал (footer) с формой обратной связи.

4.2 Регистрация и авторизация

* проверка существования (занятости) регистрируемого логина;
* авторизация клиента и администратора по логину и паролю;
* регистрация клиента через личный кабинет администратора;
* возможность запомнить данные для входа в личный кабинет клиента и администратора.

4.3 Личный кабинет клиента

* список проектов;
* Информация о выбранном проекте;
* страница настроек с возможностью изменить логин, название компании, номер телефона для связи, включения/выключения возможности связи по телефону, e-mail и пароль;
* кнопка для выхода из личного кабинета.

4.4 Личный кабинет администратора

* список клиентов и связанных с ними проектов;
* список заявок на обратную связь;
* страница настроек с возможностью изменить логин и пароль;
* кнопка для выхода из личного кабинета.

4.5 Список клиентов

* Список клиентов содержит поля: id компании, название компании, количество проектов, кнопка открытия модального окна для изменения данных клиента, кнопка открытия модального окна для удаления клиента и его проектов (возможно удаление только пользователя вместе с проектами, отдельный проект удалить невозможно), кнопка раскрытия списка проектов;
* при наличие обратной связи от клиента фоновый цвет строки изменяется;
* Сортировка списка по наличию обратной связи от клиента;
* При клике на кнопку раскрытия списка проектов, в раскрывшемся меню можно увидеть id, названия всех проектов, связанные с клиентом, кнопку «редактировать проект» для каждой строки и кнопу «добавить проект» . При нажатии на кнопку «добавить проект» открывается модальное окно с полями «название проекта», «краткое описание» и кнопками «отмена», для закрытия окна и «добавить» для регистрации проекта. При нажатии на кнопку «изменить проект» открывается модальное окно с полями «название проекта», «краткое описание» и кнопкой «изменить» для изменения данных проекта. При клике на название происходит переход на страницу проекта;
* Кнопка «добавить клиента»: при нажатии открывается модальное окно с полями «Наименование компании», «Телефон», «E-mail», и кнопками «отмена» для закрытия окна и «добавить» для регистрации клиента, при этом на указанный электронный адрес отправляется данные для входа: логин и сгенерированный пароль;
* При нажатии на кнопку для изменения или удаления клиента открывается модальное окно с полями «Наименование компании», «Телефон», «E-mail» и кнопками «сбросить пароль», нажатие на которую генерирует новый пароль и отправляет его на e-mail клиента, «отмена» для закрытия окна и «изменить» для принятия изменений.

4.6 Список заявок на обратную связь

* Список заявок содержит поля: id заявки, имя, номер телефона, кнопка «завершить» (после нажатия сменяется кнопкой «Удалить») и кнопка раскрытия меню. При клике по кнопке «завершить» заявка считается закрытой.
* В раскрываемом меню указан текст из поля «описание» формы обратной связи.

4.7 Управление клиентами и связанными проектами, возможности администратора

* Просмотр списка клиентов и переход на страницы связанных с ними проектов;
* Просмотр списка заявок на обратную связь;
* Изменение данных клиента;
* Удаление клиента;
* Регистрация клиента;
* Добавление проекта;
* Изменение состояние проекта (в том числе добавление комментария);
* Изменение данных о проекте;
* Авторизация в личном кабинете.

4.8 Возможности клиента

* Просмотр проектов;
* Изменение данных учетной записи;
* Авторизация в личном кабинете;
* Отправка обратной связи (реакции) на завершенный этап (статус) разработки.

4.9 Возможности посетителя

* Просмотр сайта;
* Отправка формы обратной связи.

**4.10 Форма обратной связи**

* Содержит поля для заполнения: Имя, Телефон, Описание;
* Имеет кнопку для отправки формы.

4.11 Страница проекта (клиент)

* Название компании, для которой разрабатывается проект;
* Название проекта;
* Текущее состояние проекта и комментарий;
* Список предыдущих состояний (этапов) проекта с комментариями и возможностью отреагировать на результат (подтвердить, если клиента устраивает результат и отказать, если результат клиента не устраивает).

**4.12 Страница проекта (администратор)**

* Название компании, для которой разрабатывается проект;
* Название проекта;
* Текущий статус проекта и комментарий;
* Список предыдущих статусов (этапов) проекта с комментариями и реакцией клиента;
* Поля для ввода нового статуса и комментария, меню выбора цвета текста и кнопка «изменить статус» для применения изменений. При изменении статуса, предыдущий добавляется в историю;

4.13 Статусы (этапы) проекта

* При добавлении проекта ему присваивается статус «Начат»;
* Администратор может задать новый статус для проекта и цвет текста.

4.14 Валидация

* Все поля для ввода не должны быть пустыми;
* Формы авторизации и регистрации: поля для имени и фамилии содержат только буквы, первая – заглавная; в поле с номером телефона используются только цифры, первый символ «+» ставиться автоматически; в поле e-mail возможно использовать только цифры и английские буквы; поле «название компании» может содержать любые символы).

5 Технологические требования

HTML (HyperText Markup Language) не ниже 5-й версии, CSS (Cascading Style Sheets) не ниже 3-ей версии, Java Script.

6 Требования к программным средствам

Операционная система семейства Windows или UNIX c возможностью подключения к сети Интернет. Web-браузер с поддержкой технологий, указанных в пункте 5 «Технологические требования».

**7 Требования к техническим средствам**

Персональный Компьютер (ПК) с поддержкой современной архитектуры центрального процессора (ЦП). Наличие графического ядра в ЦП или графического процессора.

8 Требования по надежности

Изделие должно надежно (без сбоев и ошибок) функционировать в заданных условиях и иметь защиту от неправильных вводов исходных данных, предусматривать блокировку некорректных действий пользователя при работе с системой.

**9 Эстетические и эргономические требования**

Программный продукт не должен вызывать утомления и раздражения глаз пользователя, демонстрационные экраны не должны содержать мелких шрифтов.

**Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_** *Карцева Мария Сергеевна*

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** *Авраменко Степан Сергеевич*

(подпись)                               (Ф.И.О)

**«10» сентября 2024г.**

**Приложение Б**

**Реализация Базы Данных**

START TRANSACTION;

CREATE TABLE `Admins` (

  `id` int(11) NOT NULL,

  `Login` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Password` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

INSERT INTO `Admins` (`id`, `Login`, `Password`) VALUES

(1, 'admin1', 'pass123'),

(2, 'admin2', 'pAsS321');

CREATE TABLE `Applications` (

  `id` int(11) NOT NULL,

  `Name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Phone` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Message` text COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Status` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

INSERT INTO `Applications` (`id`, `Name`, `Phone`, `Message`, `Status`) VALUES

(1, 'Иван', '+79161234567', 'Интересуюсь разработкой мобильного приложения', 'Не обработано');

CREATE TABLE `Clients` (

  `Id` int(11) NOT NULL,

  `Company` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Login` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Password` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Projects` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Phone` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `Email` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `AcceptCalls` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

INSERT INTO `Clients` (`Id`, `Company`, `Login`, `Password`, `Projects`, `Phone`, `Email`, `AcceptCalls`) VALUES

(1, 'ООО \"ТехПром\"', 'ivanov123', 'password1', 'Проект1', '+79991112233', 'ivanov@techprom.ru', 1);

CREATE TABLE `Projects` (

  `id` int(11) NOT NULL,

  `ProjectName` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `ClientID` int(11) NOT NULL,

  `StatusHistory` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

INSERT INTO `Projects` (`id`, `ProjectName`, `ClientID`, `StatusHistory`) VALUES

(1, 'Проект1', 1, 1);

CREATE TABLE `Statuses` (

  `id` int(11) NOT NULL,

  `StatusName` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL DEFAULT 'проект начат',

  `Comment` text COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `ClientAccept` tinyint(1) NOT NULL,

  `StatusColor` varchar(96) COLLATE utf8mb4\_unicode\_ci NOT NULL DEFAULT '#32CD32',

  `ProjectID` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

INSERT INTO `Statuses` (`id`, `StatusName`, `Comment`, `ClientAccept`, `StatusColor`, `ProjectID`) VALUES

(1, 'Начат', 'Проект был запущен', 1, '#32CD32', 1);

ALTER TABLE `Admins`

  ADD PRIMARY KEY (`id`);

ALTER TABLE `Applications`

  ADD PRIMARY KEY (`id`);

ALTER TABLE `Clients`

  ADD PRIMARY KEY (`Id`);

ALTER TABLE `Projects`

  ADD PRIMARY KEY (`id`),

  ADD KEY `Client-Project` (`ClientID`);

ALTER TABLE `Statuses`

  ADD PRIMARY KEY (`id`),

  ADD KEY `Status-Project` (`ProjectID`);

ALTER TABLE `Admins`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

ALTER TABLE `Applications`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

ALTER TABLE `Clients`

  MODIFY `Id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

ALTER TABLE `Projects`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

ALTER TABLE `Statuses`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

ALTER TABLE `Projects`

  ADD CONSTRAINT `Client-Project` FOREIGN KEY (`ClientID`) REFERENCES `Clients` (`Id`) ON DELETE CASCADE ON UPDATE NO ACTION;

ALTER TABLE `Statuses`

  ADD CONSTRAINT `Status-Project` FOREIGN KEY (`ProjectID`) REFERENCES `Projects` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION;

COMMIT;