

Специальность 09.02.07 Информационные системы и программирование

**КУРСОВОЙ ПРОЕКТ**

**на тему:** Проектирование, дизайн и программная реализация модуля управление телекоммуникационными услугами

по МДК 05.01 Разработка кода информационных систем,   
МДК.05.02 Разработка кода информационных систем

Студент: Шатров Алексей Андреевич

(Фамилия, И.О.)

Курс 3 группа 32ИСП(до)

Защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватели:

(Фамилия, И.О.)

(Фамилия, И.О.)

Москва 2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc196741676)

[1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ 6](#_Toc196741677)

[1.1 Основные принципы разработки информационных систем 6](#_Toc196741678)

[1.1.1 Модульность и декомпозиция процессов 6](#_Toc196741679)

[1.1.2 Взаимодействие компонентов 7](#_Toc196741680)

[1.1.3 Состояние жизненного цикла 8](#_Toc196741681)

[1.1.4 Прозрачность потоков данных 9](#_Toc196741682)

[1.1.5 Автоматизация и точность расчетов 9](#_Toc196741683)

[1.1.6 Безопасность и соответствие стандартам 10](#_Toc196741684)

[1.1.7 Масштабируемость 10](#_Toc196741685)

[1.2 Архитектурные подходы к созданию модуля информационной системы 11](#_Toc196741686)

[1.3 Современные технологии и инструменты разработки информационных систем 12](#_Toc196741687)

[1.4 Анализ существующих решений в области управления телекоммуникационными услугами 15](#_Toc196741688)

[2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ УПРАВЛЕНИЯ телекоммуникационными услугами 18](#_Toc196741689)

[2.1 Выбор платформы и инструментария для разработки модуля 18](#_Toc196741690)

[2.2 Разработка технического задания на модуль управления персональными данными студентов вуза 19](#_Toc196741691)

[2.3 Проектирование базы данных и настройка серверной инфраструктуры 22](#_Toc196741692)

[2.4 Реализация основных функций модуля 24](#_Toc196741693)

[3. ТЕСТИРОВАНИЕ И ОТЛАДКА РАЗРАБОТАННОГО МОДУЛЯ 36](#_Toc196741694)

[4. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ДОКУМЕНТАЦИИ И РУКОВОДСТВА ПО ЭКСПЛУАТАЦИИ 39](#_Toc196741695)

[ЗАКЛЮЧЕНИЕ 40](#_Toc196741696)

[СПИСОК ИСПОЛЬЗОВАННЫх ИСТОЧНИКОВ 42](#_Toc196741697)

# ВВЕДЕНИЕ

Современные телекоммуникационные компании, особенно интернет-провайдеры, сталкиваются с необходимостью эффективного управления своими услугами, клиентской базой и технической инфраструктурой. В условиях высокой конкуренции и растущих требований клиентов к качеству услуг, автоматизация процессов управления становится критически важной. Разработка специализированного модуля для управления телекоммуникационными услугами позволяет оптимизировать работу компании, повысить уровень сервиса и минимизировать операционные издержки.

Актуальность данной работы обусловлена несколькими факторами. Во-первых, интернет-провайдеры предоставляют широкий спектр услуг, включая доступ в интернет, IP-телефонию, облачные сервисы и другие, что требует комплексного подхода к их управлению. Во-вторых, клиенты ожидают удобства и прозрачности при взаимодействии с провайдером, включая возможность самостоятельного управления услугами через личный кабинет. В-третьих, автоматизация процессов учета и биллинга позволяет снизить количество ошибок и ускорить обработку запросов.

Целью курсового проекта является проектирование, дизайн и программная реализация модуля управления телекоммуникационными услугами для интернет-провайдера. Для достижения этой цели необходимо решить следующие задачи:

1. Провести анализ существующих решений и конкурентов в отрасли.
2. Определить функциональные и технические требования к модулю.
3. Разработать архитектуру системы, включая базу данных и пользовательский интерфейс.
4. Создать дизайн-макеты интерфейсов для администратора и клиентов.
5. Реализовать модуль с использованием современных технологий программирования.
6. Провести тестирование и отладку системы.

Объектом исследования является процесс разработки модуля управления телекоммуникационными услугами для интернет-провайдера. Предметом исследования выступают методы проектирования, дизайна и программирования, обеспечивающие создание эффективного и удобного решения.

# ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

# Основные принципы разработки информационных систем

Разработка информационных систем (ИС) базируется на ряде фундаментальных принципов, обеспечивающих их эффективность, надежность и соответствие требованиям пользователей. В контексте модуля управления телекоммуникационными услугами ключевые принципы были реализованы через интеграцию современных методологий проектирования и визуализации процессов.

* + 1. Модульность и декомпозиция процессов

Назначение: Моделирование функциональной структуры системы, выделение основных процессов и их взаимосвязей.



Рис. 1. IDEF0: Основные процессы модуля

Описание: Диаграмма IDEF0 (Рис. 1) отображает высокоуровневые процессы модуля управления телекоммуникационными услугами. Входные данные включают запросы пользователей, данные клиентов и тарифы. Выходные данные — подключенные услуги, финансовые отчеты и аналитика.

* + 1. Взаимодействие компонентов

**Назначение**: Моделирование последовательности действий в процессе, включая ветвления и параллельные потоки.

  
Рис. 2. UML Activity: Процесс обработки запроса.

Описание: На диаграмме (Рис. 2) представлены этапы обработки запроса клиента:

1. Проверка доступности услуги.
2. Подключение услуги.
3. Формирование счета через биллинг.  
   Ключевые элементы: взаимодействие между клиентом, модулем управления услугами и биллингом.
   * 1. Состояние жизненного цикла

Назначение: Отображение жизненного цикла объекта (например, заявки) и его реакций на события.



Рис. 3. UML State Machine: Жизненный цикл заявки.

Описание: Диаграмма (Рис. 3) показывает переходы состояний запроса: от «Запроса на услугу» через «Проверку», «Оплату», до «Активации» или «Отмены». Например, при ошибке обработки запрос возвращается на этап исправления.

* + 1. Прозрачность потоков данных

Назначение: Детализация процессов, представленных в IDEF0, для анализа подпроцессов и их взаимодействия.



Рис. 4. Декомпозиция процессов управления услугами.

Описание: Декомпозиция (Рис. 4) раскрывает этапы обработки запросов, включая управление тарифами, интеграцию с серверным ПО и веб-интерфейсом. Например, показано, как данные о тарифах влияют на формирование счетов.

* + 1. Автоматизация и точность расчетов

Назначение: Детализация процессов, представленных в IDEF0, для анализа подпроцессов и их взаимодействия.



Рис. 5. Декомпозиция процессов управления услугами.

Описание: Декомпозиция (Рис. 5) раскрывает этапы обработки запросов, включая управление тарифами, интеграцию с серверным ПО и веб-интерфейсом. Например, показано, как данные о тарифах влияют на формирование счетов.

* + 1. Безопасность и соответствие стандартам

Все процессы спроектированы с учетом требований защиты данных (GDPR, PCI DSS). Например, в DFD (Рис. 4) данные пользователей шифруются при передаче в БД, а доступ администратора к управлению услугами ограничен двухфакторной аутентификацией.

* + 1. Масштабируемость

Архитектура системы, представленная в IDEF0 и UML, позволяет добавлять новые модули (например, интеграцию с IoT-устройствами) без перестройки всей системы.

* 1. Архитектурные подходы к созданию модуля информационной системы

Архитектура модуля управления телекоммуникационными услугами строится на принципах, обеспечивающих гибкость, надежность и соответствие требованиям автоматизации процессов. Основной упор сделан на разделение функциональности на независимые компоненты, что позволяет упростить разработку, тестирование и дальнейшее масштабирование системы.

Система разделена на автономные модули, каждый из которых отвечает за конкретную задачу. Например, модуль управления тарифами обрабатывает создание и изменение тарифных планов, а модуль биллинга автоматизирует формирование счетов и интеграцию с платежными шлюзами. Такой подход упрощает внедрение новых функций, таких как мониторинг качества связи или интеграция с IoT-устройствами, без необходимости перестройки всей системы. Декомпозиция процессов помогает детализировать этапы работы: от обработки запроса пользователя до генерации финансовых отчетов.

Для обеспечения корректной передачи информации между компонентами система использует четко определенные маршруты данных. Запросы от клиентов поступают через веб-интерфейс, проходят валидацию, сохраняются в базу данных и синхронизируются с модулями управления услугами и биллинга. Это исключает дублирование операций и снижает риск потери данных. Например, при изменении тарифа информация автоматически обновляется в профиле пользователя и передается в биллинг-систему для пересчета счетов.

Логика работы системы учитывает различные сценарии взаимодействия с пользователем. Каждый запрос проходит через этапы проверки доступности услуги, подключения и формирования счета. В случае ошибки (например, недостатка средств или технических ограничений) система предоставляет возможность повторной проверки или отмены операции. Это обеспечивает гибкость и минимизирует ручное вмешательство администратора.

Для реализации архитектурных решений использовались инструменты, поддерживающие модульность и совместную работу. Проектирование процессов выполнялось в Draw.io, что позволило визуализировать взаимодействие компонентов на ранних этапах. Кодовая база разделена на фронтенд (HTML, CSS) и бэкенд (PHP, SQL), что упрощает параллельную разработку. Контроль версий через Git обеспечивает прозрачность изменений, а использование локального сервера OSPanel ускоряет тестирование в среде, близкой к реальной.

Архитектура системы учитывает требования защиты данных. Персональная информация пользователей шифруется при передаче, а доступ администраторов ограничивается двухфакторной аутентификацией. Интеграция с платежными системами соответствует стандартам PCI DSS, что гарантирует безопасность финансовых операций.

Использование описанных архитектурных принципов позволило создать модуль, который:

* Автоматизирует ключевые процессы (биллинг, управление тарифами);
* Обеспечивает прозрачность для пользователей через личный кабинет;
* Легко адаптируется под новые требования (например, добавление услуг IP-телефонии);
* Соответствует отраслевым стандартам безопасности.

Таким образом, выбранные подходы обеспечили не только техническую реализацию, но и выполнение бизнес-целей проекта: снижение операционных издержек, повышение качества сервиса и удовлетворенности клиентов.

* 1. Современные технологии и инструменты разработки информационных систем

Разработка модуля управления телекоммуникационными услугами потребовала применения современных технологий и инструментов, обеспечивающих эффективное проектирование, кодирование, управление данными и контроль версий. В рамках проекта были задействованы следующие решения:

Инструменты разработки

1. Draw.io

* Draw.io использовался для проектирования ключевых диаграмм, включая:
* IDEF0 – визуализация архитектуры системы и бизнес-процессов;
* UML activity – моделирование взаимодействия компонентов модуля;
* DFD – отображение потоков данных между пользователем, администратором и БД;
* UML state machine – моделирование жизненного цикла объекта.

2. Visual Studio Code

Основной редактор для написания кода на PHP, HTML, CSS и SQL. Преимущества использования:

Поддержка синтаксиса и автодополнение для ускорения разработки;

Интеграция с Git через расширения (например, GitLens);

Возможность отладки PHP-скриптов напрямую в редакторе.  
Пример: с помощью VSCode реализована логика обработки запросов пользователей на подключение услуг.

3. phpMyAdmin

Инструмент для проектирования и администрирования базы данных MySQL. С его помощью:

Созданы таблицы БД Пользователей (логины, пароли, тарифы) и БД Услуг (описания, стоимость, условия);

Оптимизированы SQL-запросы для быстрого поиска данных;

Настроены связи между сущностями через внешние ключи.

4. GitKraken

Графический клиент для работы с Git, обеспечивший:

Удобное управление ветками (например, отдельные ветки для разработки интерфейса и бэкенда);

Визуализацию истории коммитов;

Интеграцию с GitHub для синхронизации кода между участниками команды.

**5.**OSPanel

OSPanel — локальный сервер для разработки, который использовался для развертывания и тестирования модуля в среде, приближенной к реальным условиям. Его ключевые функции в проекте:

* Локальный хостинг: Быстрое развертывание веб-сервера (Apache/Nginx), PHP и MySQL на компьютере разработчика.
* Управление доменами: Создание виртуальных хостов (например, telecom-admin.local) для тестирования интерфейса администратора и клиентских страниц.
* Интеграция с phpMyAdmin: Упрощенное управление базой данных через встроенный инструмент, что ускорило проектирование таблиц и отладку SQL-запросов.
* Поддержка версий PHP: Возможность переключения между версиями PHP (7.4/8.2) для проверки совместимости кода.

Технологии программирования:

1. HTML и CSS

* HTML использовался для создания структуры веб-интерфейсов:
* Формы оформления запросов на услуги;
* Личный кабинет пользователя с виджетом скорости интернета.
* CSS обеспечил адаптивный дизайн:
* Медиа-запросы для мобильных устройств;
* Стилизацию элементов (например, интерактивный калькулятор тарифов).

**2.**PHP

* Язык для реализации серверной логики:
* Обработка данных из форм (проверка корректности ввода);
* Интеграция с БД через SQL-запросы (например, выборка тарифов для отображения);
* Генерация динамического контента (персонализированные уведомления о подключении услуг).

**3.**SQL

Для работы с базой данных:

Написание сложных запросов с JOIN для объединения таблиц пользователей и услуг;

Создание триггеров для автоматического обновления статуса услуги при оплате.

**4.**GitHub

* Платформа для хостинга кода и совместной работы:
* Реализация CI/CD через GitHub Actions для автоматического тестирования;
* Хранение резервных копий проекта;
* Управление задачами через Issues (например, отслеживание багов в модуле биллинга).
  1. Анализ существующих решений в области управления телекоммуникационными услугами

Современные провайдеры домашнего интернета активно используют специализированные системы для автоматизации ключевых процессов: управления клиентской базой, тарификацией, биллингом и технической поддержкой. Наиболее распространены CRM-платформы, такие как Bitrix24 и Microsoft Dynamics, которые предоставляют базовые инструменты для работы с абонентами, но часто требуют доработки под специфику телекоммуникационной отрасли. Например, интеграция с оборудованием для мониторинга скорости интернета или автоматическое формирование счетов с учетом динамических тарифов.

Специализированные решения, такие как Megatec или BSG, предлагают готовые модули для учета телекоммуникационных услуг, включая поддержку IP-телефонии и облачных сервисов. Однако их внедрение сопряжено с высокими затратами и сложностью адаптации под малые и средние компании. Кроме того, многие системы не обеспечивают достаточной гибкости при изменении тарифных планов, что критично для провайдеров, стремящихся быстро реагировать на рыночные изменения.

Отдельного внимания заслуживают облачные платформы, такие как Chargebee или Recurly, которые фокусируются на автоматизации биллинга. Они упрощают управление подписками и интеграцию с платежными системами, но не покрывают весь спектр задач телекоммуникационного провайдера — например, не поддерживают мониторинг качества связи.

Анализ выявил ключевые проблемы существующих решений:

* Недостаточная интеграция между модулями (например, разрыв между биллингом и технической поддержкой);
* Сложность настройки под уникальные требования провайдера;
* Высокая стоимость лицензий и обслуживания.

Эти недостатки стали основой для формулирования требований к разрабатываемому модулю.

**1.5 Постановка задачи и определение требований к модулю управления телекоммуникационными услугами**

Целью разработки модуля является создание гибкой системы, которая объединит управление клиентскими данными, тарификацией, биллингом и мониторингом качества услуг для провайдера домашнего интернета.

Функциональные требования

1. Управление тарифными планами:
   * Возможность создания динамических тарифов с учетом скорости интернета, лимитов трафика и дополнительных опций (например, статический IP);
   * Автоматическое применение изменений тарифов для всех подключенных абонентов.
2. Биллинг и платежи:
   * Интеграция с платежными системами (Robokassa, PayPal) для онлайн-оплат;
   * Формирование счетов с детализацией потребленных услуг.
3. Мониторинг качества услуг:
   * Сбор данных о скорости интернета и uptime через интеграцию с маршрутизаторами;
   * Генерация отчетов для анализа сетевой нагрузки.
4. Клиентский интерфейс:
   * Личный кабинет абонента с возможностью смены тарифа, просмотра статистики использования и подачи заявок в поддержку;
   * Мобильная адаптация интерфейса.

Нефункциональные требования

* + Производительность: Обработка до 10 000 транзакций в час;
  + Безопасность: Шифрование персональных данных (SSL), двухфакторная аутентификация для администраторов;
  + Совместимость: Поддержка работы с существующими базами данных провайдера (MySQL);
  + Масштабируемость: Возможность добавления новых модулей (например, интеграция с IoT-устройствами).

Учет потребностей пользователей

Для абонентов критично упрощение взаимодействия:

* + Минимизация шагов при смене тарифа;
  + Прозрачность тарифов без скрытых комиссий;
  + Мгновенные уведомления о проблемах с подключением.

Администраторы нуждаются в централизованном управлении:

* + Единая панель для контроля подключений, финансов и качества услуг;
  + Автоматизация рутинных задач (напоминания о просроченных платежах).

# ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ УПРАВЛЕНИЯ телекоммуникационными услугами

* 1. Выбор платформы и инструментария для разработки модуля

Перед началом верстки сайта следует определиться со средой его разработки. В интернете существует огромное количество программ, не сильно отличающихся по функционалу. Не каждая программа имеет по-настоящему широкий функционал, который позволяет облегчить разработчику процесс написания кода. Для разработки веб-сервиса использовался Visual Studio Code − это легкий и мощный редактор кода с поддержкой множества языков программирования, включая JavaScript, Python и HTML/CSS, что делает его отличным выбором для веб-разработки. Он предлагает встроенный Git-клиент, отладчик и богатую экосистему расширений для повышения продуктивности. Благодаря кроссплатформенности и активному сообществу, VS Code остается одним из самых популярных инструментов среди разработчиков. Интерфейс редактора (см. рисунок 8)



Рис. 6. Интерфейс Visual Studio Code

* 1. Разработка технического задания на модуль управления персональными данными студентов вуза

Таблица 2.1 Техническое задание

| **Параметр** | **Описание** |
| --- | --- |
| **Назначение документа** | Определение требований и порядка разработки модуля управления телекоммуникационными услугами для интернет-провайдера. |
| **Заказчик** | ООО «ТелекомСервис» |
| **Исполнитель** | Шатров Алексей Андреевич, студент группы 32ИСП(до) |
| **Основание для разработки** | Курсовой проект по МДК 05.01 и МДК 05.02 специальности 09.02.07 «Информационные системы и программирование». |
| **Плановые сроки работ** | Начало: 01.02.2025 Окончание: 24.04.2025 Этапы работ указаны в разделе «Порядок контроля и приемки». |
| **Порядок оформления результатов** | Работы сдаются поэтапно. По завершении каждого этапа предоставляются результаты для согласования с заказчиком и преподавателем. |

Назначение и цель создания модуля

| **Параметр** | **Описание** |
| --- | --- |
| **Назначение модуля** | Автоматизация процессов управления услугами интернет-провайдера: биллинг, тарификация, мониторинг качества связи, интеграция с клиентским интерфейсом. |
| **Цель создания** | Повышение эффективности работы компании за счет оптимизации управления услугами, снижения ошибок и улучшения клиентского сервиса. |

Основные требования к модулю

Функциональные требования:

1. Управление тарифами:
   * Создание и редактирование динамических тарифов (скорость интернета, лимиты трафика, дополнительные опции).
   * Автоматическое применение изменений для всех клиентов.
2. Биллинг и платежи:
   * Интеграция с платежными системами (Robokassa, PayPal).
   * Формирование счетов с детализацией услуг.
3. Мониторинг качества услуг:
   * Сбор данных о скорости интернета и uptime через интеграцию с маршрутизаторами.
   * Генерация отчетов для анализа.
4. Клиентский интерфейс:
   * Личный кабинет с возможностью смены тарифа, просмотра статистики, подачи заявок в поддержку.
   * Адаптация под мобильные устройства.

Нефункциональные требования:

* + Производительность: Обработка до 10 000 транзакций в час.
  + Безопасность: Шифрование данных (SSL), двухфакторная аутентификация для администраторов.
  + Совместимость: Работа с MySQL.
  + Масштабируемость: Возможность добавления новых модулей (например, IoT-устройства).

Требования к дизайну

* + Стиль интерфейса: Современный, минималистичный.
  + Адаптивность: Поддержка мобильных устройств.
  + Цветовая схема: Соответствие корпоративному стилю заказчика.

Информационное обеспечение

* + Языки программирования: PHP (бэкенд), HTML/CSS/JavaScript (фронтенд).
  + База данных: MySQL.
  + Инструменты: Visual Studio Code, Draw.io (для диаграмм), Git (для контроля версий).

Программное обеспечение

Клиентская часть:

* + Браузеры: Google Chrome 85+, Firefox 80+, Safari 14+.

Серверная часть:

* + Веб-сервер: Apache/Nginx.
  + PHP: версии 7.4/8.3.
  1. Проектирование базы данных и настройка серверной инфраструктуры

Для разработки веб-сервиса прежде всего необходимо создать базу данных, которая должна содержать информацию о тарифах, об оформленных на веб-сервисе тарифах и о пользователях системы.

Данные должны храниться в базе данных в виде связанных между собой

таблиц:

а) тарифы – данные о представляемых на сайте изделиях;

б) аккаунты – данные для входа на сайт и ограничения доступа;

в) услуги – данные о выбранных тарифах;

Для разработки веб-сервиса по приему заказов творческих проектов была

выбрана свободная реляционная система управления базами данных MySQL

версии 8.2 для 64-битных систем семейства Windows, используемая через вебинтерфейс администрирования phpMyAdmin, который позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL,просматривать содержимое таблиц и баз данных. Через него были созданы объекты базы данных. В качестве сервера, на котором осуществлялась работа выше перечисленных средств, был выбран Apache версии, поддерживающей PHP 8.1 для 64-битных систем семейства Windows.

Перед созданием базы данных разработаем ER-диаграмму, отображающую

структуру таблиц и связей между ними (см. рисунок 6).

****

Рис. 7. ER-диаграмма базы данных

База данных состоит из пяти таблиц (см. рисунок 4.3) с атрибутами:

1) таблица «users»: код пользователя, логин пользователя, имя пользователя, фамилия пользователя, телефон, пароль, права;

2) таблица «tariff»: код тарифа, название тарифа, скорость;

3) таблица «services»: код услуги, код пользователя, код тарифа, адрес, дата начало тарифа, дата окончания;

****

Рис. 8. База данных в phpMyAdmin

* 1. Реализация основных функций модуля

Перед тем как наполнить сайт контентом, нужно разработать скрипт основных блоков – заголовок, блок отображения контента и подвал. Код для заголовка (см. рисунок 9) и подвала (см. рисунок 10) страницы идентичен для всех страниц.



Рис. 9. Код заголовка страниц

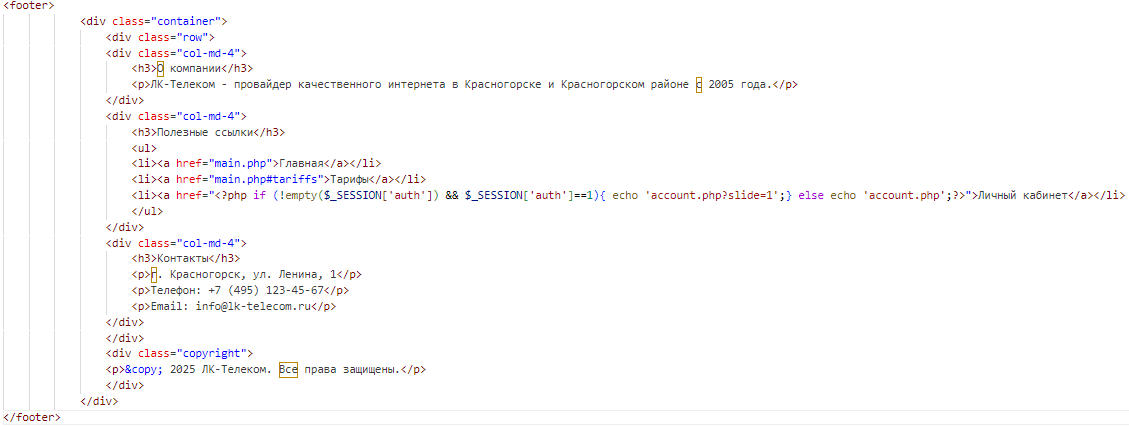


Рис. 10. Код подвала страниц

После того, как был разработан шаблон сайта и прописаны его стили (см. рисунок 10), можно посмотреть результат на главной странице (см. рисунок 11).



Рис. 10. Стили, прописанные в css



Ри. 10. Главная страница веб-сервиса

Так же на главной присутствует раздел “Предоставляемы нами тарифы” (см. рисунок 11), который формируется запросом к базе данных (см. рисунок 12). Подключение к базе данных выполняется в отдельном файле, так как требуется для всех запросов (см. рисунок 13).



Риc. 11. Раздел “Предоставляемы нами тарифы”



Риc. 12. Запрос тарифов



Риc. 13. Подключение к БД

Форма авторизации и регистрации (см. рисунок 14) находится на одной странице.



Риc. 14. Форма авторизации/регистрации

После заполнения поля формы “регистрация” выполняется скрипт для проверки данных на заполненность и добавление записи в БД таблица “users” (см. рисунок 15). Если пользователь заполнить форму “авторизация” то выполнится скрипт на наличие данных в полях ввода и поиск по логину в БД таблица “users” если же такой пользователь существует будет произведена проверка пароля (см. рисунок 16).



Риc. 15. Код регистрации пользователя



Риc. 16. Код авторизации пользователя

После успешной авторизации/регистрации пользователю будет доступен личный кабинет в котором доступно 3 формы Личные данные (см. рисунок 17)., Услуги (см. рисунок 20)., Подключенные услуги (см. рисунок 21).



Риc. 17. Личные данные

В этой форме пользователь может изменить данные код представлен (см. рисунок 18), и произвести выход из системы , код представлен (см. рисунок 19).



Риc. 18 Код редактирование данных



Риc. 19 Код выхода из системы



Риc. 20 Форма «Услуги»

На этой форме пользователь может выбрать необходимый ему тариф. Нажав на кнопку “Выбрать”, после он попадет на страницу оформление тарифа (см. рисунок 21).



Риc. 21 Подключение тарифа



Риc. 22 Код формы подключение тарифа

Здесь пользователю необходимо ввести адрес подключение и подтвердить свой выбор нажатием на кнопку далее подключенная услуга отобразится в форме Подключенные услуги (см. рисунок 22).



Риc. 22 форма подключенные услуги



Риc. 23 Код формы подключенные услуги

Для пользователей с правами администратор доступна страница “Админ панель” (см. рисунок 24) в которой можно отредактировать (см. рисунок 20.4) и удалить каждую запись хранимую в БД (см. рисунок 20.5) . Код выборки записей (см. рисунок 25-28).



Риc. 24 Страница Админ панель



Риc. 25 Код выборки из таблицы “users”



Риc. 26 Код выборки из таблицы “tariff”



Риc. 27 Код выборки из таблицы “services”



Риc. 28 Форма редактирование записи



Риc. 29 Код удаление записей

* 1. ТЕСТИРОВАНИЕ И ОТЛАДКА РАЗРАБОТАННОГО МОДУЛЯ

Чтобы проект на выходе не содержал ошибок, необходимо проведение тестирования. Тестирование позволяет выявить недоработки проекта и осуществить работу по их устранению. Результаты тестирования представлены в таблице 4.1

Таблица 2.2 Тест-кейс «Оформление тарифа на сайте интернет-провайдера»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Заголовок** | **Шаги тестирования** | **Набор тестовых данных** | **Ожидаемый результат** | **Фактический результат** | **Приоритет** | **Статус** | **Описание дефекта** |
| Оформление тарифа на сайте без имение аккаунта | 1. Открыть страницу регистрации.  2. Ввести ФИО.  3. Ввести телефон.  4. Ввести emai.  5. Ввести логин.  6. Ввести пароль.  7. Нажать "Зарегистрироваться".  8. Перейти в аккаунт  9.Открыть раздел “Услуги”  10. Нажать “Выбрать” 11. Выбрать тариф и внести в поле “Адрес подключение” | Фамилия: Иванов  Имя: Иван  Телефон: 8(916)-987-65-40  Email: [ivan@example.com](mailto:ivan@example.com)  Логин: ivanivanov  Пароль: StrongPass123 | Успешная регистрация с последующей авторизацией и оформлением выбранной услуги | Совпадает с ожидаемым |  |  |  |
| Проверка на заполнение поле адрес во время оформление тарифа | 1. Открыть страницу авторизации.  2. Ввести логин.  3. Ввести пароль.  4. Нажать "авторизоваться".  5. Перейти в раздел услуги  6. Нажать выбрать  7. не заполняя поле адрес нажать ‘подключить тариф’ | Логин: Legion  Пароль: 111 | Появится сообщение заполните это поле | Совпадает с ожидаемым |  |  |  |

* 1. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ДОКУМЕНТАЦИИ И РУКОВОДСТВА ПО ЭКСПЛУАТАЦИИ

Веб-пространство для компаний, занимающихся организацией корпоративных мероприятий, представлено разнообразными решениями - от простых сайтов-визиток до сложных многофункциональных порталов. Для определения оптимальной архитектуры и функционального наполнения разрабатываемого веб-ресурса был проведен анализ.

# ЗАКЛЮЧЕНИЕ

Курсовой проект на тему «Проектирование, дизайн и программная реализация модуля управления телекоммуникационными услугами» был успешно выполнен, достигнув поставленных целей и задач. В ходе работы были разработаны и реализованы ключевые компоненты модуля, включая базу данных, пользовательский интерфейс и функциональные возможности для управления услугами интернет-провайдера.

Основные результаты проекта:

1. Анализ и проектирование: Проведен анализ существующих решений в области управления телекоммуникационными услугами, что позволило сформулировать требования к модулю. Были разработаны диаграммы (IDEF0, UML, DFD), которые наглядно отображают архитектуру системы, процессы взаимодействия компонентов и жизненный цикл заявок.
2. Реализация модуля:
   * Создана база данных на MySQL, включающая таблицы для пользователей, тарифов и услуг, с четко определенными связями между ними.
   * Разработан веб-интерфейс с использованием HTML, CSS, PHP и SQL, обеспечивающий удобное взаимодействие пользователей с системой. Интерфейс включает личный кабинет, формы авторизации/регистрации, а также административную панель для управления данными.
   * Реализованы функциональные возможности, такие как подключение тарифов, редактирование персональных данных, мониторинг услуг и автоматическое обновление статусов.
3. Тестирование: Проведено тестирование модуля, включая проверку корректности работы форм, обработки данных и взаимодействия с базой данных. Результаты тестирования подтвердили соответствие системы заявленным требованиям.
4. Документация: Подготовлена пользовательская документация и руководство по эксплуатации, что облегчит внедрение системы и ее дальнейшее использование.

Выводы: Курсовой проект продемонстрировал успешную реализацию модуля управления телекоммуникационными услугами, соответствующего современным требованиям автоматизации и удобства взаимодействия. Разработанная система обеспечивает эффективное управление тарифами, клиентской базой и услугами, что подтверждается пройденным тестированием и наглядной визуализацией процессов через диаграммы IDEF0, UML и DFD. Гибкость архитектуры позволяет легко масштабировать функционал, а интуитивно понятный интерфейс повышает удобство для пользователей и администраторов. Проект готов к внедрению и может стать основой для дальнейшего развития, включая интеграцию с платежными системами и расширенными инструментами аналитики.

# СПИСОК ИСПОЛЬЗОВАННЫх ИСТОЧНИКОВ