

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**информационной системы «Телефоны и абоненты»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 311ИС-23** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Э.М.Астахова** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Н.И.Кручинкина** |

**Москва 2024**

**Содержание**

[**ВВЕДЕНИЕ** 4](#_Toc185538110)

[1 **АНАЛИЗ** **ПРЕДМЕТНОЙ** **ОБЛАСТИ** 6](#_Toc185538111)

[1.1 Описание предметной области 6](#_Toc185538112)

[1.2 Анализ существующих программных решений 7](#_Toc185538113)

[1.3 Постановка задачи. Структура входной и выходной информации 9](#_Toc185538114)

[2 **ПРОЕКТИРОВАНИЕ** **И РАЗРАБОТКА** **ПРОГРАММНОГО** **МОДУЛЯ** 10](#_Toc185538115)

[2.1 Эскизное проектирование 10](#_Toc185538116)

[2.1.1 Выбор инструментальных средств реализации 11](#_Toc185538117)

[2.2 Проектирование базы данных 12](#_Toc185538118)

[2.3 Разработка архитектуры программного модуля 18](#_Toc185538119)

[2.4 Разработка интерфейса. 20](#_Toc185538120)

[2.4.1 Функциональные возможности интерфейса. Написание программного кода 22](#_Toc185538121)

[3 **ТЕСТИРОВАНИ**Е **И ОТЛАДКА ПРОГРАММНОГО МОДУЛЯ** 24](#_Toc185538122)

[**ЗАКЛЮЧЕНИЕ** 28](#_Toc185538123)

[**СПИСОК** ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc185538124)

[**ПРИЛОЖЕНИЕ** **1** 30](#_Toc185538125)

[**ПРИЛОЖЕНИЕ 2** 31](#_Toc185538126)

[**ПРИЛОЖЕНИЕ 3** 37](#_Toc185538127)

[**ПРИЛОЖЕНИЕ** **4** 42](#_Toc185538128)

# ВВЕДЕНИЕ

Актуальность темы курсового проекта обусловлена растущим спросом конкретных пользователей или организаций на удобство хранения и получения своих данных или данных своих работников.

У каждого абонента есть свой телефон и номер телефона. В информацию, получаемую о них пользователем входят как личные данные самого абонента, так и данные о телефоне и тарифе, который использует этот телефон.

Деятельностью разрабатываемой программы является получение вышеперечисленной информации пользователем (как администратором, так и самим абонентом) по запросу, либо добавление этой информации абонентом, если таковой ещё нет в базе данных (То есть регистрация новых абонентов).

Целью выполнения проекта является разработка программного модуля с данными проектируемой информационной системы, позволяющая пользователям эффективно управлять данными.

Для достижения целей этой курсовой работы (проекта) поставлены следующие задачи:

* проанализировать разные телефонные информационные системы для создания собственного программного модуля;
* разработать программный модуль, включающий телефоны, их абонентов и данные о них;
* внедрить и автоматизировать поиск информации о каждом абоненте для пользователя;
* исследовать возможные функции для автоматизации всех процессов в информационной системе;
* изучить информацию о создании программных модулей;
* определить цели, объект и предмет работы (проекта);
* разработать лаконичный интерфейс;

Объектом исследования является телефонная компания в целом.

Предмет исследования составляет данные и процесс учёта абонентов, информации о них и их телефоны.

Практическая значимость для разрабатываемого программного модуля заключается в том, что правильно разработанное приложение с удобным и интуитивно понятным интерфейсом, а также полезным функционалом создает выигрышную ситуацию как для абонентов (удобство хранения данных, контроль используемых тарифов и экономия времени), так и для оператора (снижение затрат, улучшение обслуживания, увеличение выручки). Это повышает конкурентоспособность оператора и улучшает его репутацию.

Структура проекта:

Курсовой проект состоит из трёх глав: «анализ предметной области», «проектирование и разработка программного модуля» и «тестирование и отладка программного модуля».

В первой главе раскрываются теоретические аспекты темы «Телефоны и абоненты» и ставится техническое задание на выполняемый проект.

Вторая глава описывает и иллюстрирует разработку программного модуля и базы данных.

В третьей главе готовый программный модуль тестируется, выявляются и исправляются ошибки, допущенные при создании кода.

Результатом успешно выполненной курсовой работы будет являться программный модуль, позволяющим пользователям быстро и эффективно получать всю необходимую информацию об абонентах, используя логин и пароль.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Описание предметной области

Предметная область «Телефоны и абоненты» очень актуальна в нашем мире, в котором везде используются мобильная связь и цифровые технологии. При условиях постоянного роста числа абонентов и разнообразия мобильных устройств, крайне важным для многих компаний становится успешное управление информацией о них.

Актуальность проекта по этой теме основывается на необходимости таких задач, как:

* оптимизация бизнес-процессов:

Эффективное управление данными о телефонах и абонентах позволяет операторам связи и другим компаниям оптимизировать свои процессы, улучшить качество обслуживания и повысить свои доходы;

* безопасность хранимых данных пользователей:

Надежность в хранении персональных данных абонентов является основным компонентом обеспечения их безопасности и конфиденциальности. Это является действительно важным в условиях появившейся и растущей угрозы преступности в интернете;

* + анализ принятия решений:

Анализ данных о телефонах и абонентах даёт возможность выявить изменения в спросе и впоследствии принимать более обоснованные решения в организациях;

* + персонализация услуг:

Знание того, в чём нуждаются абоненты и информации об их данных ( телефонах в том числе ) даёт компаниям возможность предлагать услуги, основанные на персональных желаниях абонентов и повысить уровень удовлетворенности клиентов.

Разработка и улучшение систем управления данными в предметной области «Телефоны и абоненты» является актуальной задачей, которая помогает повышению качества жизни в цифровую эпоху.

Предметная область темы «Телефоны и абоненты» описывает взаимодействие между абонентами и используемыми ими телефонами. Она включает в себя информацию об абонентах (личные данные, контактная информация), телефонах (телефонные номера, тарифные планы), а также о самих тарифных планах (стоимость, описание, входящие в план услуги), которые используются с этими телефонами. Система управления такой информацией необходима для магазинов мобильной техники, или любых организаций, работающих с базами данных абонентов и их устройств, а также для самих клиентов.

Цель предметной области – организация, хранение и управление информацией о связи между абонентами и телефонами, обеспечивающая целостность и доступность данных для различных запросов и отчетов.

## 1.2 Анализ существующих программных решений

Предметная область "Телефоны и абоненты" охватывает широкий спектр программных решений, от простых телефонных книг до сложных CRM-систем, используемых для управления взаимодействиями с клиентами.

Из категорий, приемлемых для анализа существующих решений можно выделить такую, как системы операторов мобильной связи. Анализ подобных систем требует рассмотрения нескольких аспектов:

* функциональность личных кабинетов и мобильных приложений;
* особенности управления контактами и данными внутри этих систем;
* интеграция с другими сервисами и общие впечатления (отзывы) пользователей.

В пример можно привести сайты таких мобильных операторов, как "Билайн", "МТС", "Мегафон" и т.д.

В преимущества функционала сайта оператора мобильной связи "Билан" входят такие возможности, как:

* + управление тарифом;
  + управление балансом;
  + управление услугами;
  + обратная связь;
  + возможность управления несколькими номерами в рамках общего тарифа.

Недостатком является ограничение интеграции с внешними сервисами в системе: авторизация может быть упрощена через социальные сети, но нет тесной интеграции с приложениями для управления контактами или другими сторонними сервисами.

Общие впечатления пользователей о сайте также неоднозначны:

В качестве положительной стороны отмечают удобство мобильного приложения "Мой Билайн", наличие детальной истории звонков и трафика.

Но многие пользователи часто критикуют наличие большого количества рекламы и интерфейс личного кабинета на сайте, считая его устаревшим и неудобным. Отсутствие функций управления контактами и своими личными данными является значительным недостатком. Проблемы с производительностью приложения также отмечаются пользователями.

## 1.3 Постановка задачи. Структура входной и выходной информации

Требованиями к выполненному программному модулю является наличие функционала, позволяющего пользователям:

1. регистрировать нового абонента;
2. получать личные данные абонента (включающие в себя ФИО абонента, адрес абонента, дату рождения абонента, электронную почту и тариф, выбранный абонентом) и информацию о тарифе (название тарифного плана, стоимость тарифного плана, краткое описание тарифного плана);
3. выбирать наиболее подходящий для абонента тарифный план и менять его.

При этом при входе в качестве администратора в функционал входят возможности:

* просмотра информации о всех существующих абонентах;
* поиска информации об абоненте, используя только его номер телефона;
* добавления нового тарифа со всей информацией о нём;
* экспорта данных из базы данных.

Интерфейс должен быть лаконичным и интуитивно понятным для каждого пользователя, что позволит администратору также эффективно искать и редактировать необходимую информацию.

# 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ

## 2.1 Эскизное проектирование

Целью эскизного проектирования является проектирование и выделение отдельных независимых компонентов системы и способы решения их задач для дальнейшей разработки программного модуля.

В основные функции разрабатываемого проекта входит возможность пользователя получать данные об абоненте при вводе его регистрационных данных. Обычный абонент может как получить доступ просмотра своих данных сразу после авторизации, так и авторизироваться, внеся свои данные в систему. Если же пользователем является администратор, то после авторизации он также получает доступ к просмотру данных существующих абонентов. Данные в системе делятся на три вида: личные данные абонента с контактной информацией и данные об используемом тарифе.

На страничке каждого типа данных будут присутствовать кнопки, предлагающие просмотреть другой тип данных внесённого абонента.

Если данных об абоненте ещё нет в созданной базе данных, то должно быть выведено сообщение о предложении данному абоненту зарегистрироваться. Если это требуется, то пользователь сам может добавить все данные о себе с помощью кнопки «Регистрация». Если пользователь просто допустил ошибку в логине/пароле, то после нажатия кнопки «ОК» на всплывающем сообщении об ошибке, он всё равно вернётся на страницу входа в систему, где сможет заново внести данные для авторизации.

Схематично порядок работы программы представлен в виде блок-схемы в приложении 1 на рисунке 1.1.

### **2.1.1 Выбор инструментальных средств реализации**

Для реализации программного модуля был выбран язык программирования Python. Такой выбор обоснован несколькими факторами:

1. Простота написания и читаемости кода. У языка Python ясная и лаконичная структура, что очень упрощает как разработка, так и отладку и сопровождение кода. Код на этом языке очень лёгок в понимании, поэтому время создания программы и вероятность допуска ошибок снижается.
2. Богатая стандартная библиотека. Python включает в себя обширную стандартную библиотеку со множеством модулей для решения разнообразных задач. Это ускоряет разработку, используя готовые программные решения, а не написание кода с нуля. А также наличие библиотек расширяет функциональность самого языка программирования.
3. Кроссплатформенность. Код может исполняться на разных операционных системах (Windows, MacOS, Linux) без существенных изменений.
4. Лёгкость интеграции. Python просто интегрируется с другими языками программирования и различными технологиями. Это позволяет использовать его как связь в сложных системах.

При создании графического дизайна интерфейса приложения было решено использовать интегрированную среду разработки Tkinter Designer с использованием библиотеки Tkinter. Выбор такого решения:

1. значительно ускоряет и упрощает процесс создания интерфейса посредством кода;
2. позволяет создавать красивый, интуитивно-понятный для будущего пользователя дизайн, что отлично подходит для требований к данному проекту.

Выбор SQLite3 для хранения данных о телефонах, абонентах и тарифах обусловлен его простотой и удобством использования. Это встроенная база данных, не требующая отдельного сервера, что упрощает настройку и работу для выполняемого проекта. Она имеет относительно простой и понятный интерфейс, а также эта база данных является одним файлом, что делает её крайне легко переносимой между различными платформами.

Подключение и работа с базой данных SQLite3 в Python осуществляется с помощью модуля sqlite3, который обеспечивает легкость подключения и работы с базой данных в системе программного модуля.

В качестве среды разработки была выбрана интегрированная среда разработки (IDE) для языка программирования Python, созданная компанией JetBrains – PyCharm. Эта среда разработки предоставляет:

1. современный редактор кода с автодополнением, яркой подсветкой синтаксиса, а также возможностью рефакторинга;
2. пошаговый отладчик с возможностями проверки значений переменных и отслеживания выполнения кода;
3. встроенная поддержка фреймворков тестирования, таких как pytest и др., для написания и запуска тестов.

Для тестирования и дальнейшей отладки кода было выбрано ручное тестирование. Оно позволяет подробно проверить работоспособность программы с точки зрения конечного пользователя и оперативно выявить ошибки в функционале программного модуля.

## 2.2 Проектирование базы данных

Для проектирования и создания программного модуля системы требуется выделить основные объекты и описать их. Сущности базы данных приведены в виде таблицы 1.2 в приложении 1.

Для дальнейшего проектирования, в программный модуль нужно внести базу данных с таблицами и всей нужной для успешной работы приложения информацией.

Выделим набор атрибутов для каждой из сущностей.

Таблица «users\_info» (Информация пользователей) включает всю необходимую информацию об абоненте (Таблица 1). В ней описаны личные данные абонента-8 атрибутов.

В ней атрибут «Tarif\_id» является внешним ключом для связи с таблицей «Tarifes» (Тарифы), а атрибут «User\_id»-внешним ключом для связи с таблицей «Users».

Таблица 1 – атрибуты таблицы «Users\_info».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INT PRIMARY KEY | Уникальное IDинформации о каждом абоненте. |
| User\_id | INT | ID абонента. Внешний ключ для связи с таблицей «Users». |
| Name | TEXT | ФИО абонента. |
| Phone | VARCHAR (12) | Номер телефона абонента. |

Продолжение таблицы 1 – атрибуты таблицы «Users\_info».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Address | TEXT | Адрес абонента. |
| Birthday | DATE | Дата рождения абонента. |
| Email | VARCHAR (255) | Электронная почта абонента. |
| Tarif\_id | INT | ID используемого абонентом тарифа. Внешний ключ для связи с таблицей «Tarifes». |

Таблица «users» (Пользователи) содержит данные, необходимые для безопасности и защиты личных данных абонентов. У каждого абонента есть свой логин и пароль для входа в свой личный кабинет с информацией о них (См. в таблице 2).

Таблица 2-таблица «Users».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| User\_id | INT PRIMARY KEY | Уникальное ID каждого абонента. |
| Login | VARCHAR (255) | Логин абонента. |
| Password | VARCHAR (255) | Пароль абонента. |
| Is\_auth | INT | Отслеживания состояния аутентификации пользователя. |

В таблице «Tarifes» (Тарифы) хранятся данные о существующих тарифных планах, включая их стоимость и количество минут и гигабайт, входящих в тариф (См. таблицу 3).

Таблица 3-таблица «Tarifes».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id\_tarif | INT PRIMAR KEY | Уникальное ID каждого тарифа. |
| Name | VARCHAR (255) | Название тарифа. |

Продолжение таблицы 3-таблица «Tarifes».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Price | INT | Цена тарифа. |
| Description | TEXT | Краткое описание тарифа. |

В таблице «Admins» (Администраторы) хранятся данные для безопасного входа в систему в качестве администратора (См. таблицу 4)

Таблица 4-таблица «Admins».

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID | INT PRIMARY KEY | Уникальное ID для администратора. |
| Login | VARCHAR (255) | Логин для входа в качестве администратора. |
| Password | VARCHAR (255) | Пароль для входа в качестве администратора. |
| Is\_auth | INT | Отслеживания состояния аутентификации пользователя. |

Проектирование структуры базы данных является критическим этапом в разработке любой системы, и программа для телефонов и абонентов не является исключением. Грамотно спроектированная база данных напрямую влияет на производительность, масштабируемость и надежность всего приложения.

Для визуализации и анализа связей между различными атрибутами и сущностями системы интернет-магазина была использована ER-диаграмма (Entity-Relationship Diagram), представленная на рисунке 2. Эта диаграмма наглядно демонстрирует взаимосвязи между ключевыми сущностями, такими как пользователи, тарифы.

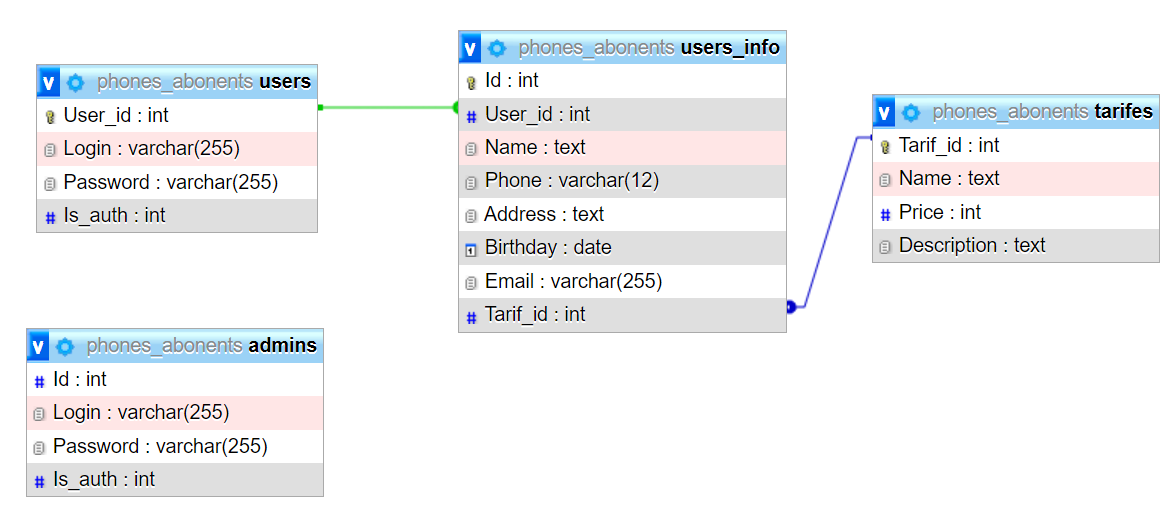


Рисунок 2-ER-диаграмма базы данных «Телефоны и абоненты».

Тщательное проектирование обеспечивает эффективное хранение и извлечение данных, а также оптимизирует взаимодействие между различными частями системы, что в конечном итоге положительно сказывается на работе пользователей и общей производительности приложения. В результате, структура базы данных, визуализованная на ER-диаграмме, служит фундаментом для быстрой, надежной и масштабируемой работы интернет-магазина, обеспечивая бесперебойное взаимодействие между пользователями, товарами, заказами и процессами оплаты.

После создания всех необходимых таблиц, необходимо выделить взаимосвязи между ними.

1. Связь «Пользователь-Данные пользователя».

К одному пользователю прикреплён один блок информации (личных данных) о нём.

Связь схематично представлена на рисунке 3.

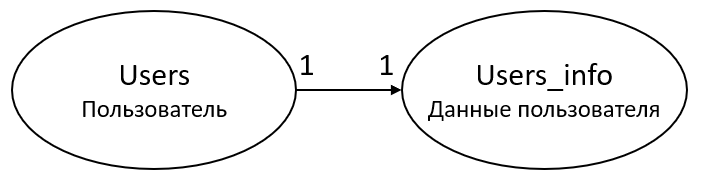


Рисунок 3- Связь «Пользователь-Данные пользователя».

2. Связь «Тариф-Данные пользователя».

У одного пользователя (абонента) может быть только один тарифный план.

Связь представлена на рисунке 4.

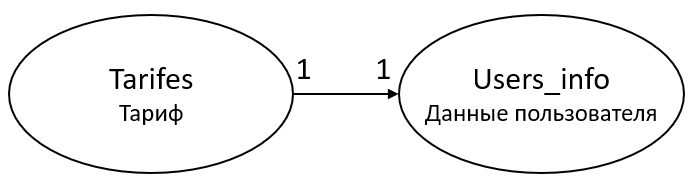


Рисунок 4- Связь «Тариф-Данные пользователя».

## 2.3 Разработка архитектуры программного модуля

Архитектура программного модуля-это разбиение всей системы на связанные части, чтобы продуктивно решать все поставленные задачи, разделяя их между составными проекта.

Практическая часть проекта состоит из базы данных и кода программного модуля, являющимися основными компонентами.

Use-case диаграмма (она же- диаграмма вариантов использования) — это тип диаграмм, который описывает функционал разрабатываемого программного модуля системы, доступного каждой группе пользователей.

На диаграмме изображены графически участники программного модуля и варианты использования ими системы и их взаимодействия в виде эллипсов. Это нужно для лучшего понимания работы программного модуля и требований к системе. См. рисунок 5.

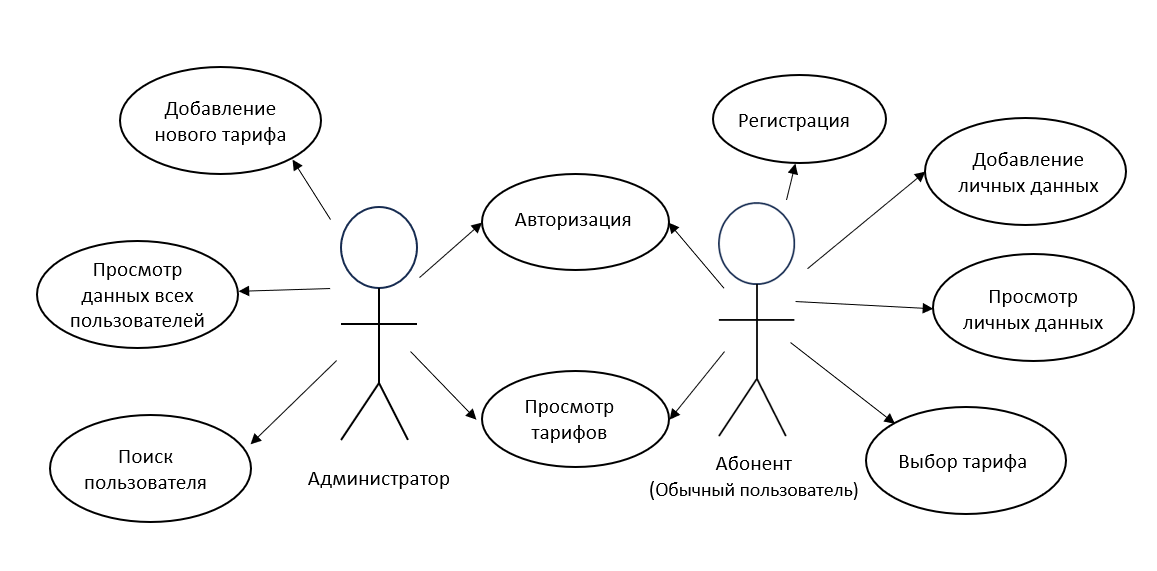


Рисунок 5-Use-case диаграмма.

Администратор имеет возможность просмотра существующих тарифных планов и добавление новых тарифов и информации о них, а также просмотр и поиск зарегистрированных в системе абонентов и экспорт данных обо всех абонентах и тарифах.

В основные функции пользователя входят добавление своих данных (Номера телефона, личных данных, данных о телефоне и используемом тарифе) и выбор/смена тарифного плана.

После анализа теоретической части проекта, а также визуализации (в виде составления таблиц, рисунков и диаграмм) всех необходимых материалов (в том числе-базы данных) и определения ролей, можно приступать к непосредственной разработке кода программного модуля и его тестированию.

## 2.4 Разработка интерфейса.

Интерфейс программы разработан для удобства работы с системой учёта абонентов пользователем. Он должен быть понятен для любого человека, чтобы он смог просто и быстро добавлять и получать все необходимые ему данные.

В основные элементы интерфейса входят такие окна, как:

1. Главное меню: в центре меню находятся две кнопки, предлагающие варианты входа в качестве администратора или в качестве обычного абонента. (См. в приложении 2 на рисунке 2.1)
2. Авторизация: пользователю предлагается ввести свой логин и пароль. В зависимости от ранее выбранной роли это может быть ввод логина и пароля администратора (Приложение 2 рисунок 2.2) или ввод логина и пароля/регистрация пользователя (приложение 2 рисунок 2.3)
3. Регистрация: поля ввода данных о новом пользователе (приложение 2 рисунок 2.4)
4. Выбор тарифного плана: можно выбрать как тариф из категории «Предложения дня» (приложение 2 рисунок 2.5.), так и из всех существующих тарифов (приложение 2 рисунок 2.6).
5. Информация абонента: показ личных данных пользователя после его успешного входа или регистрации и возможность смены тарифа представлены в приложении 2 на рисунке 2.7
6. Меню администратора: расположены кнопки, предлагающие просмотреть информацию о существующих пользователях или тарифах (Приложение 2 рисунок 2.8.).
7. Тарифы для администратора: в этом окне представлены все тарифы и имеется возможность добавить новый с помощью кнопки (Приложение 2 рисунок 2.9.)
8. Добавление тарифа: при добавлении тарифа предлагается ввести все нужные данные, включающие в себя название, стоимость и описание тарифа (Приложение 2 рисунок 2.10.).
9. Список абонентов: окно включает в себя список всех абонентов и их личные данные, а также кнопка предлагает возможность поиска информации о любом абоненте через его номер телефона (Приложение 2 рисунок 2.11.).
10. Поиск абонента: после ввода номера телефона искомого абонента, в поля будут выведены все данные об этом абоненте (Приложение 2 рисунок 2.12.).

Интерфейс программного модуля был реализован с помощью библиотеки Tkinter и **инструмента для ускорения процесса разработки графического интерфейса Tkinter Designer.**

В рабочей области, на которой создаётся интерфейс, используются элементы управления (widgets), такие как кнопки (Button), метки (Label), поля ввода (Entry), фреймы (Frame) и другие.

После размещения элемента, настраиваются его свойства. Эти свойства могут включать текст, отображаемый на кнопке или метке, функции этой кнопки; размер и положение элемента; цвет фона и текста; шрифт; и многие другие параметры, в зависимости от типа элемента управления. Пример на рисунке 6.



Рисунок 6-Создание элементов интерфейса.

### **2.4.1 Функциональные возможности интерфейса. Написание программного кода**

В основе программного модуля системы лежит база данных, которая содержит таблицы для абонентов, администратора и тарифов. Программа обрабатывает вводимые пользователями данные, выполняет их проверку, а также записывает или считывает информацию из базы данных или добавляет её при необходимости.

В функционал интерфейса входит:

1. Авторизация пользователей в качестве абонента (Код см. в приложении 3 на рисунке 2.13) и администратора (код см. в приложении 3 на рисунке 2.14): логин и пароль проверяются на соответствие наличие таковых в базе данных.
2. Регистрация нового пользователя (код смотреть в приложении 3 на рисунке 2.15): данные посредством ввода в ячейки добавляется в соответствующую таблицу базы данных и проверяются на корректность.
3. Выбор тарифного плана: возможен как со страницы «Предложения дня» (код в приложении 3 на рисунке 2.16), так и из списка всех существующих тарифов (Приложение 3 рисунок 2.17).
4. Вывод личных данных для абонента: осуществляется с помощью выборки из соответствующих таблиц в базе данных (Приложение 3 рисунок 2.18).
5. Поиск абонента по его номеру телефона (Приложение 3 рисунок 2.19): программа ищет соответствующий номер телефона в базе данных и выводит данные об абоненте, которому он принадлежит.
6. Добавление нового тарифа (информации о нём) в базу данных (Приложение 3 рисунок 2.20).
7. Экспорт данных об абонентах/тарифах в Excel. (Пример в приложении 3 на рисунке 2.21)

# 3 ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММНОГО МОДУЛЯ

В процессе выбранного ранее ручного тестирования были проверены все функции программного модуля, включая регистрацию и авторизацию пользователей, поиск и вывод информации, выбор и добавление тарифного плана. Тестирование охватило все части модуля и помогло проверить правильность вывода данных из базы данных в систему программы, а также их добавление в базу данных.

Все тест-кейсы описывали свои сценарии проверки, сравнивали ожидаемый результат с фактическим и фиксировались. Результаты проверок либо подтверждали успешный результат совершённого пользователем действия, либо помогали выявить существующие в программном модуле недоработки.

В негативное тестирование вошли такие действия, как:

* отсутствие логина/пароля при авторизации. Результат представлен на рисунке 7;

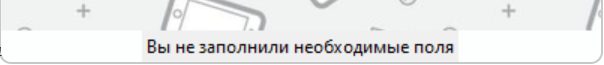


Рисунок 7 – Отсутствие логина/пароля при вводе.

* ввод неверного пароля. Результат см. на Рисунке 8;

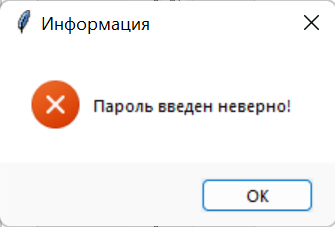


Рисунок 8-Ввод неверного пароля.

* ввод несуществующего логина. Результат см. на Рисунке 9;

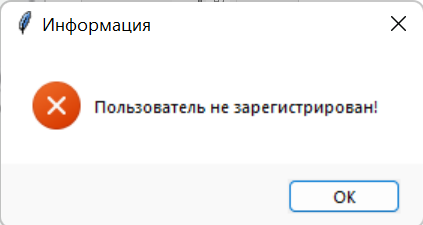


Рисунок 9-Ввод неверного логина.

* добавление информации, не соответствующей требованиям. Результат см. на Рисунке 10;

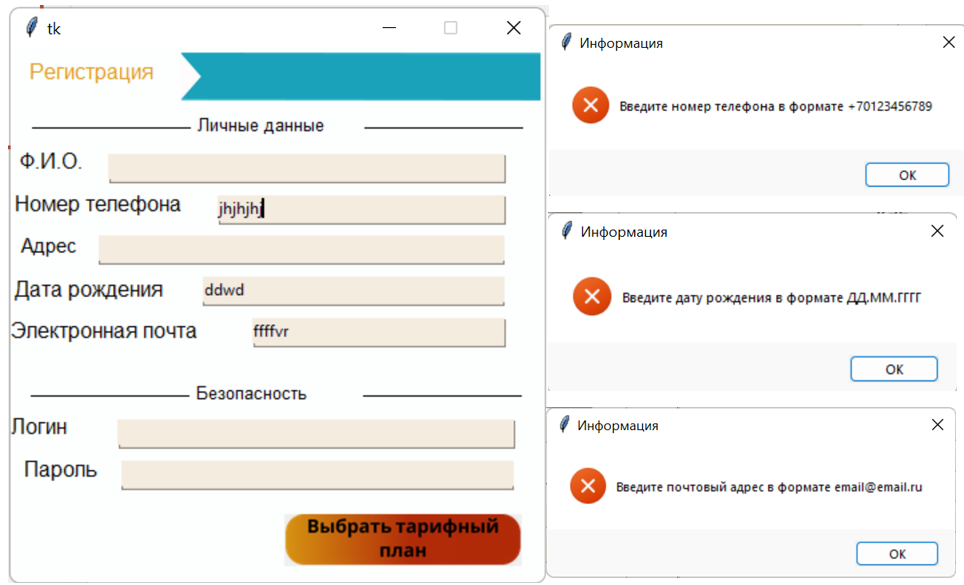


Рисунок 10-Добавление некорректных данных.

* отсутствие заполненных данных в одном или нескольких полях при добавлении нового тарифа администратором. Результат см. на Рисунке 11;

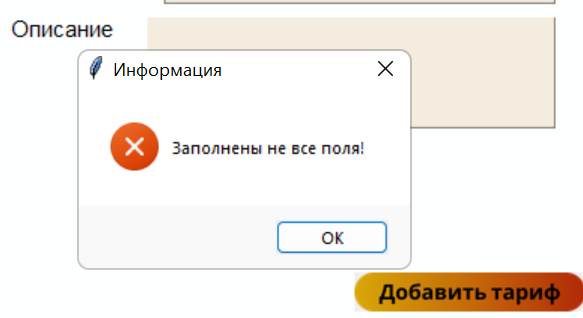


Рисунок 11-Незаполненные поля при добавлении нового тарифа.

В качестве примера позитивного тестирования можно привести экспорт данных о тарифах в Excel. Как выглядит таблица с данными после экспорта показано на рисунке 12.

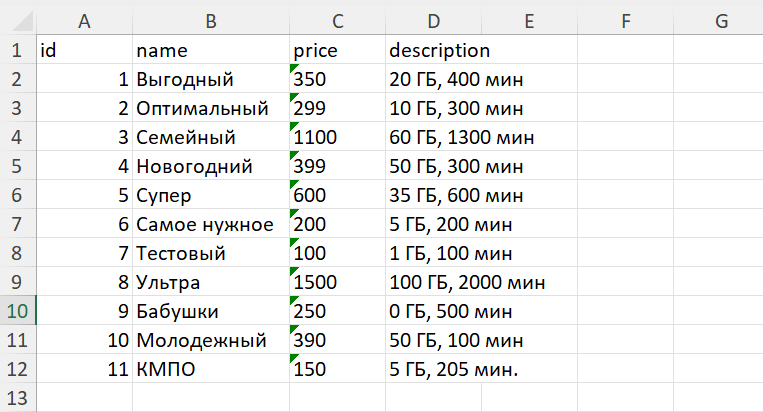


Рисунок 12-Экспортированные данные.

Подробные результаты тест-кейсов описаны в Приложении 4 в Таблице 3.1.

Отладка программного модуля являлась необходимой частью в создании программного модуля. Она помогла выявить существующие ошибки и исправить их.

Отладка выполнялась с помощью «debugger» в используемой среде разработки. Он позволил построчно выполнить программу для более точного обнаружения ошибок в коде, наблюдая за значениями переменных на каждом шаге программы.

# ЗАКЛЮЧЕНИЕ

При выполнении проекта были достигнуты все поставленные в начале выполнения работы цели.

В первую очередь был проведён подробный анализ выбранной предметной области по теме «Телефоны и абоненты». Был проведён анализ существующих программных решений, вследствие чего из них были заимствованы преимущества для проектируемого программного модуля и предупреждены возможные для подобных программ недостатки, что позволило сделать программный модуль более удобным и понятным в использовании для пользователей.

Итоговый программный модуль включает в себя весь необходимый функционал для эффективной работы с приложением и в качестве абонента (обычного пользователя), и в качестве администратора. Приложение является интуитивно понятным, а интерфейс лаконичным и удобным, что позволит любому пользователю быстро адаптироваться в системе.

Модуль позволяет автоматизировать обычные операции, освобождая сотрудников поддержки ПО от необходимости выполнять рутинные задачи в ручном режиме, что повышает производительность системы в целом.

В свою очередь автоматизация учетных процессов позволяет значительно снизить количество ошибок и задержек. В целом это помогает повысить уровень удовлетворённости клиентов и позволяет сотрудникам заниматься другими, более сложными задачами, связанными с обслуживанием программного модуля.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Стандарты и законодательные материалы

1. ГОСТ 34.601-90 "Информационная технология. Системы обработки данных. Техническое задание".
2. ГОСТ 19.105-78 "Единая система программной документации. Техническое предложение на создание программы".
3. ГОСТ 34.201-89 "Информационная технология. Автоматизированные системы. Техническое предложение".
4. ГОСТ 19.106-78. Единая система программной документации (ЕСПД). Требования к программным документам, выполненным печатным способом (с Изменениями №1). – М.: РСТ, 2010. (с Изменением №1, утвержденным в сентябре 1981 г.).

Учебники и учебные пособия

1. Шилдт, Х. «Программирование на языке Python». — М.: ООО «Вильямс», 2019.
2. «Руководство по SQL. Теория и практика». — М.: Вильямс, 2016.
3. «Изучаем Python». — Эрик Мэтиз, 2020.
4. «Используя SQLite». — Джей А. Крейбич, 2010.
5. «Python и SQLite». — Майкл Дрисколл, 2024.
6. «Основы программирования на языке Python: учебное пособие». —С.К. Буйначев, Н.Ю.Боклаг.

Интернет источники

1. Real Python Tutorials <https://realpython.com/>
2. Селектовна И. Д. «Введение в SQL» //https://stepik.org/course/191774/syllabus
3. Идеальная структура проекта на Python - [ <https://realpython.com/python-application-layouts/>], 2023
4. Документация по Python - [ <https://docs.python.org/3/>], 2023
5. Основы Python для начинающих - [ <https://www.learnpython.org/>], 2023

Монография:

1. Ссылка на репозиторий с проектом: [https://github.com/EllinaAst/kurs\_project](https://github.com/EllinaAst/kurs-project)

# ПРИЛОЖЕНИЕ 1

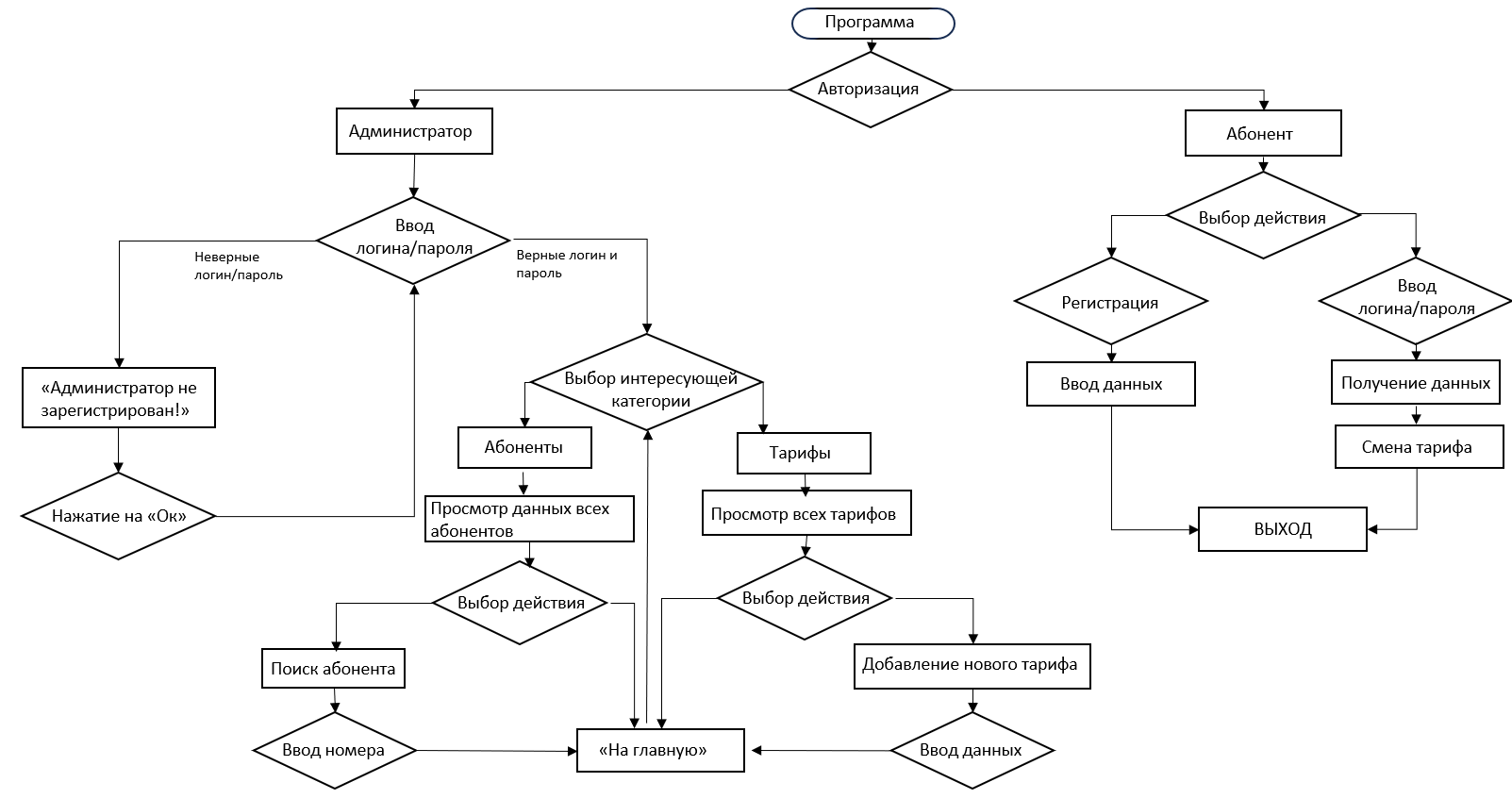


Рисунок 1.1-Блок-схема работы программы.

Таблица 1.2-Объекты базы данных.

|  |  |  |
| --- | --- | --- |
| № | Наименование объекта | Краткое описание |
| 1 | Абонент | Человек, который получает свои данные в системе и может менять свой тариф. |
| 2 | Администратор | Человек, у которого есть возможность просмотра данных обо всех существующих тарифных планах и абонентах. |
| 3 | Тариф | Тарифные планы, один из которых может быть выбран абонентом, а администратор имеет возможность добавлять новые. |

# ПРИЛОЖЕНИЕ 2



Рисунок 2.1-Главное меню

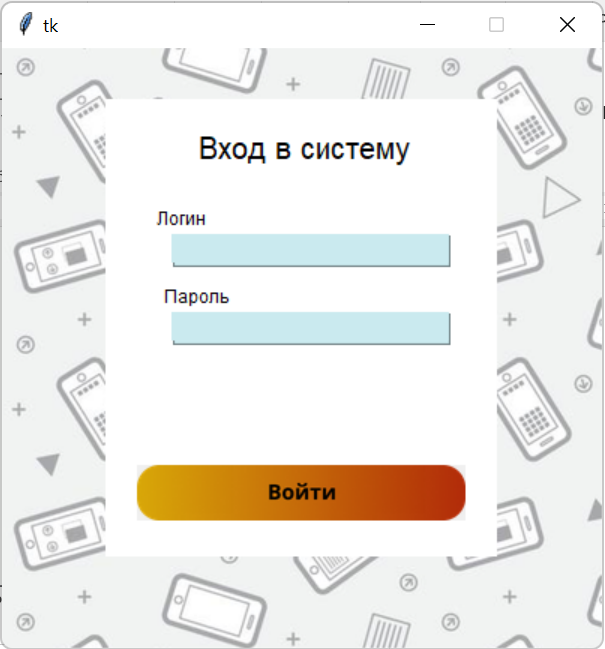


Рисунок 2.2-Вход в качестве администратора.

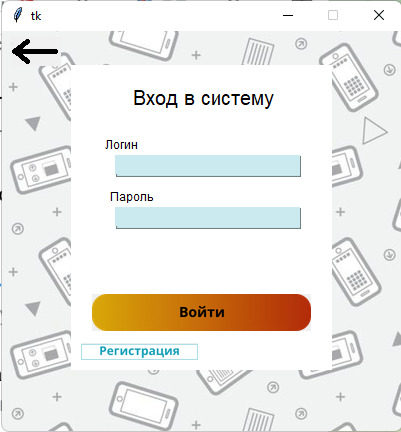


Рисунок 2.3-Вход в качестве абонента.

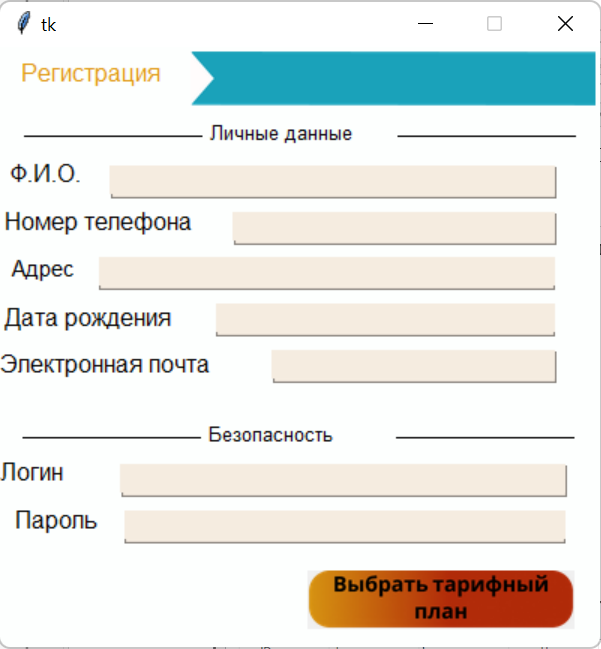


Рисунок 2.4-Регистрация.



Рисунок 2.5-Тарифы «Предложения дня».

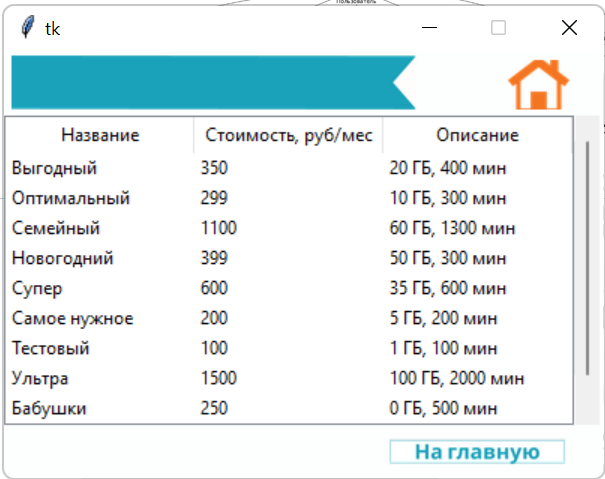


Рисунок 2.6-Все тарифы.

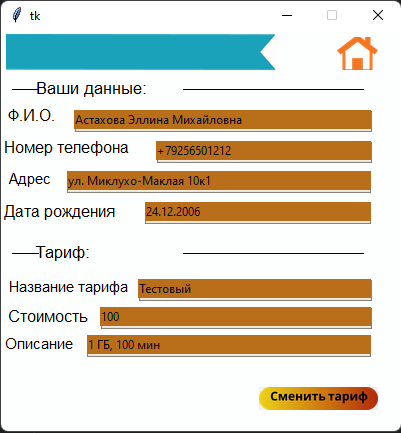


Рисунок 2.7-Информация абонента.

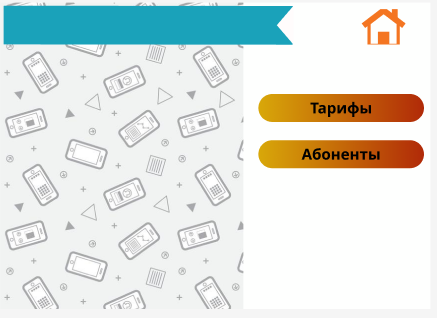


Рисунок 2.8-Меню администратора.

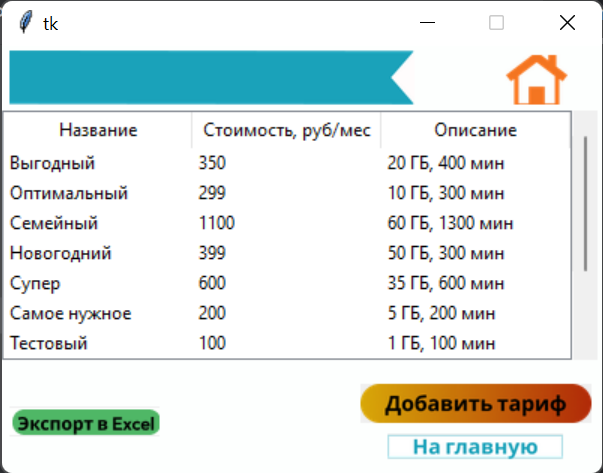


Рисунок 2.9-Список тарифов для администратора.

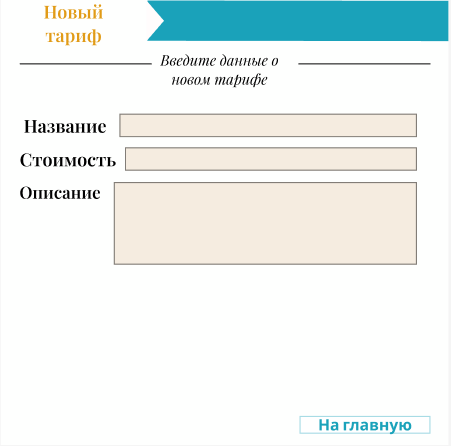


Рисунок 2.10-Добавление нового тарифа.

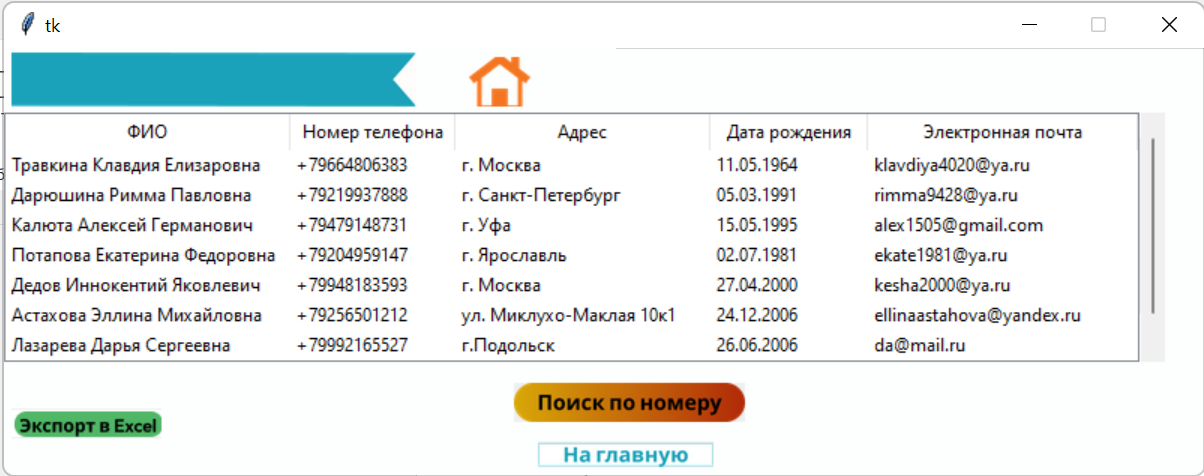


Рисунок 2.11-Список личных данных абонентов.

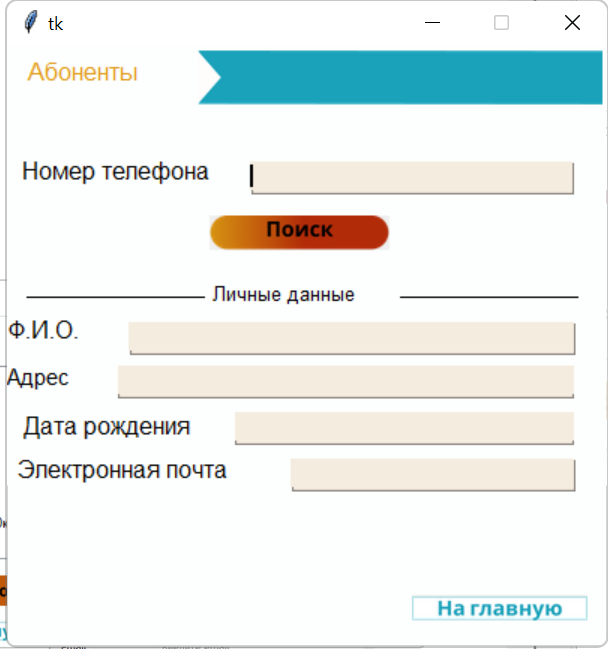


Рисунок 2.12-Поиск данных абонента по номеру.

# ПРИЛОЖЕНИЕ 3

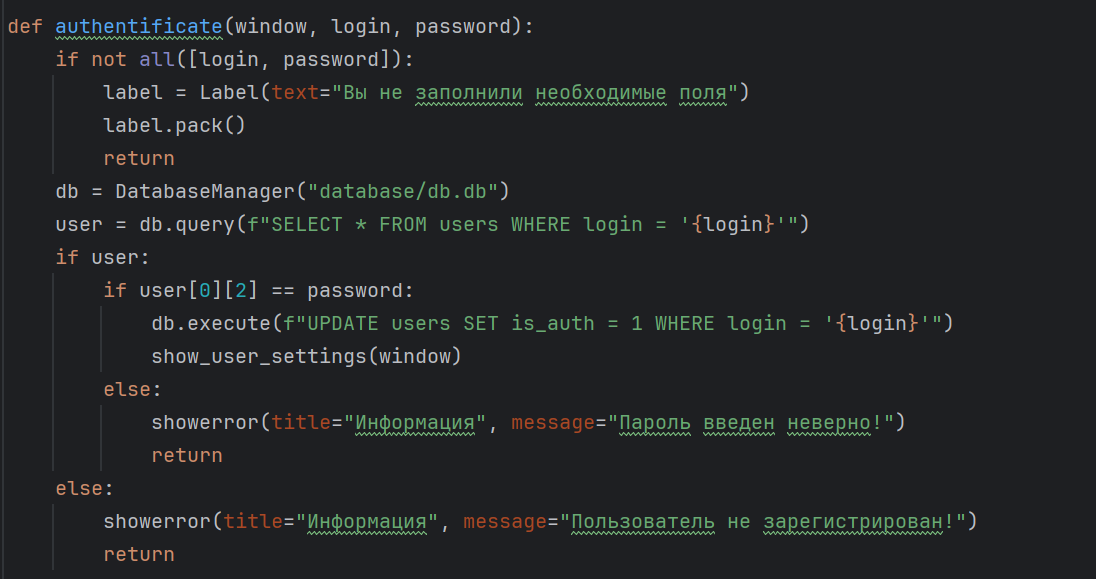


Рисунок 2.13-Аутентификация в качестве абонента.

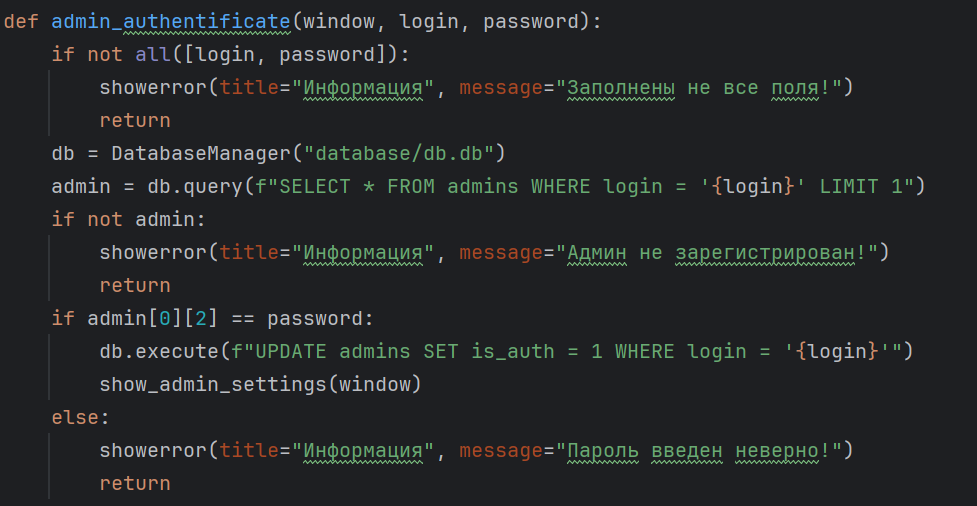


Рисунок 2.14-Аутентификация в качестве администратора.

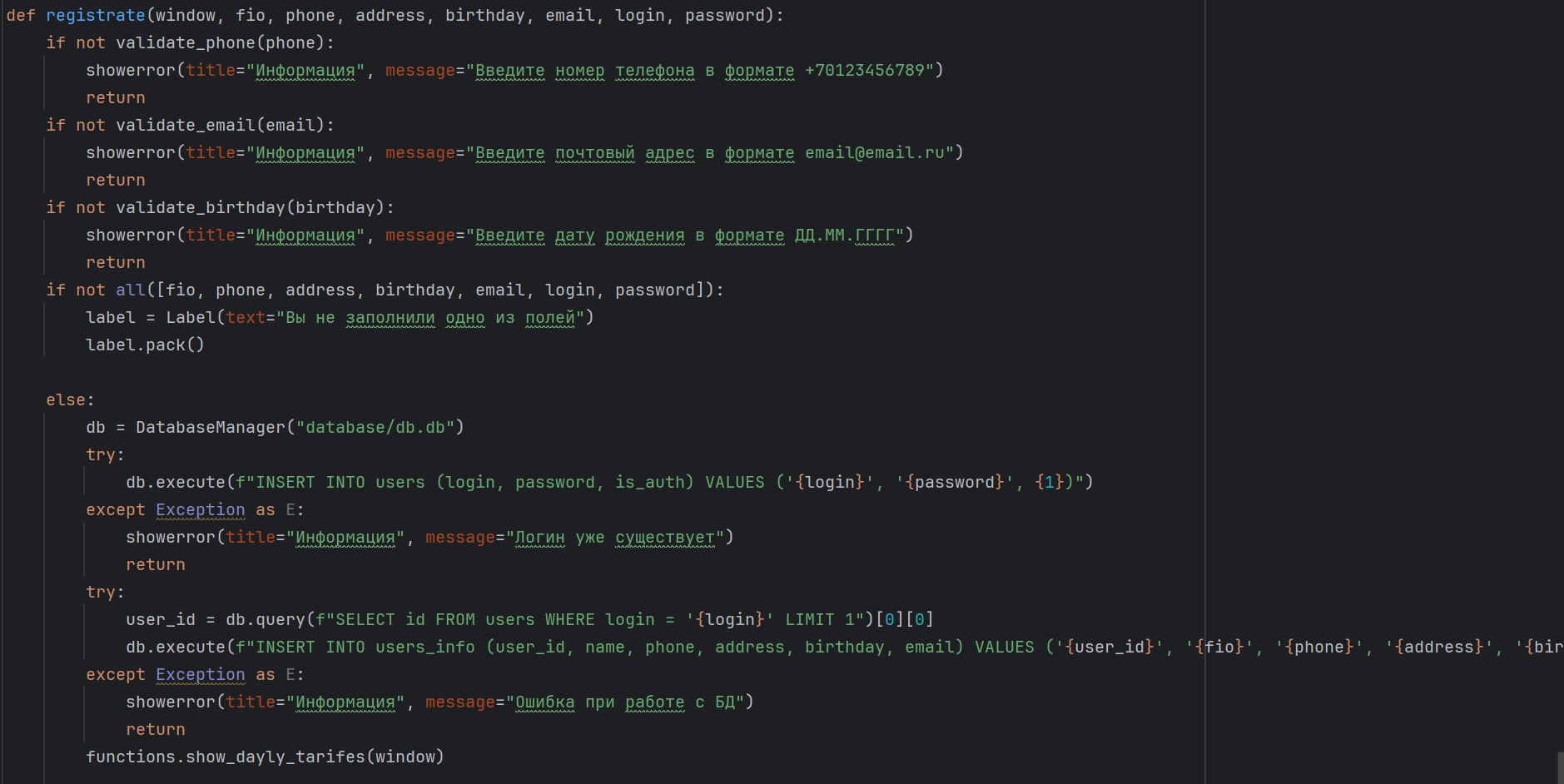


Рисунок 2.15-Регистрация абонента.

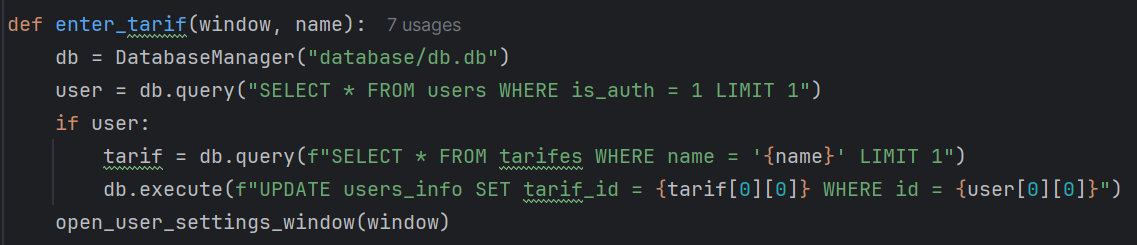


Рисунок 2.16-Выбор тарифа из списка.

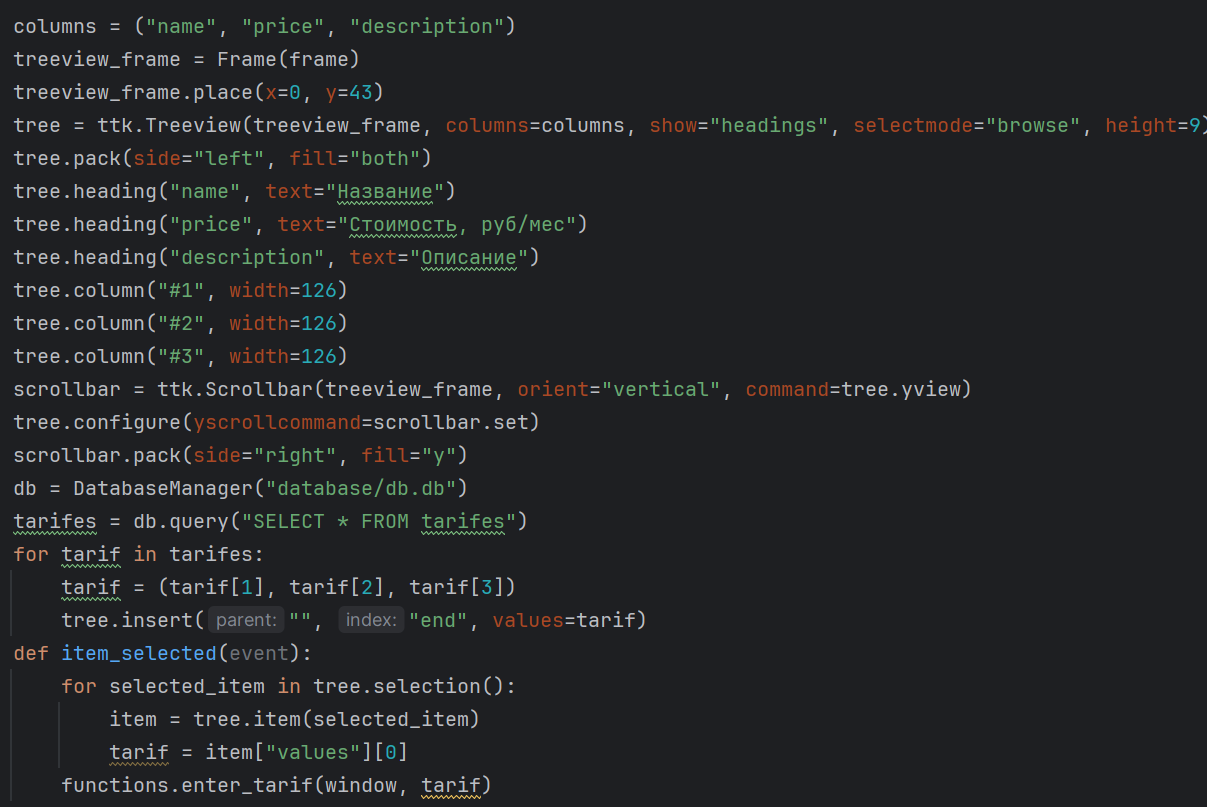


Рисунок 2.17-Вывод таблицы тарифов с возможность выбора одного из них.

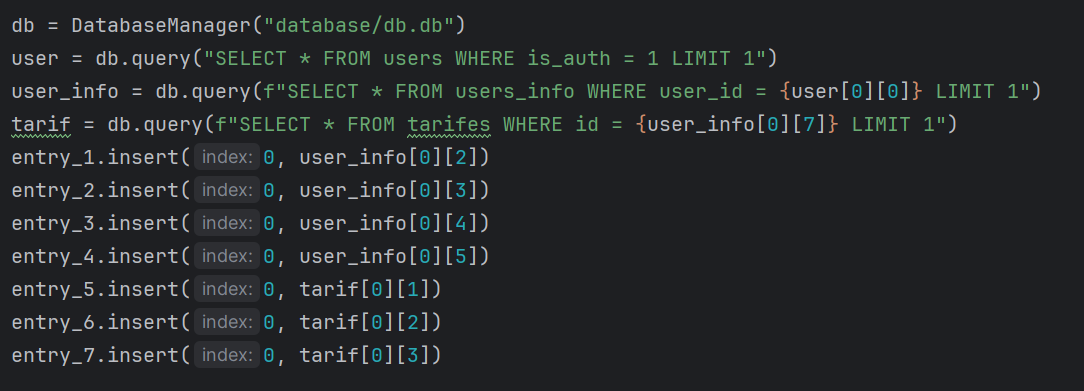


Рисунок 2.18-Вывод личных данных для абонента.

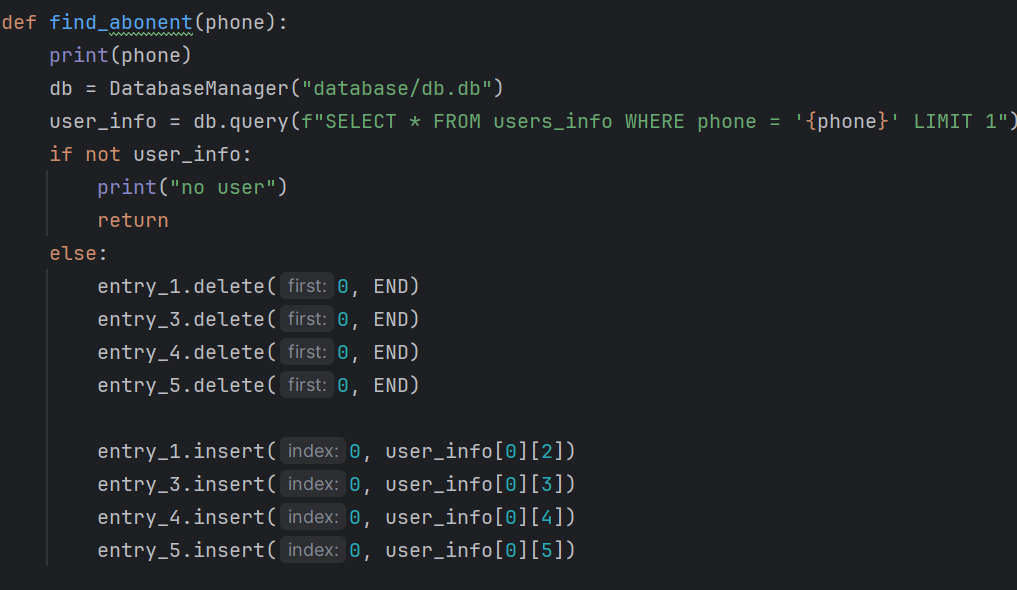


Рисунок 2.19-Поиск пользователя.

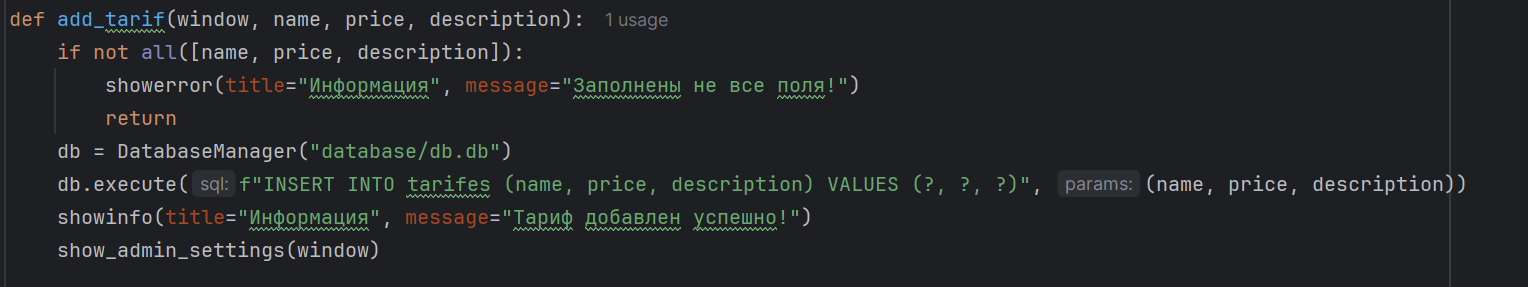


Рисунок 2.20-Добавление тарифа.



Рисунок 2.21-Функция экспорта данных о тарифах.

# ПРИЛОЖЕНИЕ 4

Таблица 3.1-План тестирования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **1** | Авторизация в системе (позитивный) | 1.Пользователь находится на странице входа в личный кабинет  2. Пользователь был ранее зарегистрирован в системе | 1. Ввести в поля «Логин» и «Пароль» логин и пароль пользователя  2. Нажать кнопку «Войти» | Пользователь перенаправляется на страницу личного кабинета | Пользователь перенаправляется на страницу личного кабинета |
| **2** | Авторизация в системе (негативный) | 1.Пользователь находится на странице входа в личный кабинет  2. Пользователь не был ранее зарегистрирован в системе | 1. Ввести в поля «Логин» и «Пароль» логин и пароль пользователя  2. Нажать кнопку «Войти» | Пользователь на странице авторизации получает сообщение «Пользователь не зарегистрирован!» | Пользователь на странице авторизации получает сообщение «Пользователь не зарегистрирован!» |

Продолжение таблицы 3.1-План тестирования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **3** | Регистрация (позитивный) | 1. Пользователь находится на странице регистрации  2. Пользователь не был ранее зарегистрирован в системе | 1. Ввести в обязательные поля «ФИО», «Номер телефона», «Адрес», «Дата рождения», «Электронная почта», «Логин», «Пароль» данные.  2. Нажать кнопку | Пользователь на странице регистрации получает сообщение «Пользователь успешно зарегистрирован.» | Пользователь на странице регистрации получает сообщение «Пользователь успешно зарегистрирован.» |
| **4** | Регистрация (негативный) | 1. Пользователь находится на странице регистрации  2. Пользователь был ранее зарегистрирован в системе | 1. Ввести в обязательные поля «ФИО», «Номер телефона» и т.д. «Логин», «Пароль» данные.  2. Нажать кнопку | Пользователь на странице регистрации получает сообщение «Логин уже существует» | Пользователь на странице регистрации получает сообщение «Логин уже существует» |

Продолжение таблицы 3.1-План тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **5** | Ввод данных для регистрации (негативный) | 1. Пользователь находится на странице регистрации  2. Пользователь не был ранее зарегистрирован в системе | 1. Ввести данные во все необходимые поля.  2.Ввести в обязательное поле «Номер телефона» невалидные данные.  3. Нажать кнопку | Пользователь на странице регистрации получает сообщение «Введите номер телефона в формате +70123456789» | Пользователь на странице регистрации получает сообщение «Введите номер телефона в формате +70123456789» |
| **6** | Ввод данных для регистрации (негативный) | 1. Пользователь находится на странице регистрации  2. Пользователь не был ранее зарегистрирован в системе | 1. Ввести данные во все необходимые поля.  2.Ввести в обязательное поле «Дата рождения» невалидные данные.  3. Нажать кнопку | Пользователь на странице регистрации получает сообщение «Введите дату рождения в формате ДД.ММ.ГГГГ» | Пользователь на странице регистрации получает сообщение «Введите дату рождения в формате ДД.ММ.ГГГГ» |

Продолжение таблицы 3.1-План тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **7** | Ввод данных для регистрации (негативный) | 1. Пользователь находится на странице регистрации  2. Пользователь не был ранее зарегистрирован в системе | 1. Ввести данные во все необходимые поля.  2.Ввести в обязательное поле «Электронная почта» невалидные данные.  3. Нажать кнопку | Пользователь на странице регистрации получает сообщение «Введите Электронную почту в формате email@email.ru» | Пользователь на странице регистрации получает сообщение «Введите Электронную почту в формате email@email.ru» |
| **8** | Поиск по номеру телефона (позитивный) | 1. Администратор находится на странице поиска абонента  2. Абонент был ранее зарегистрирован в системе | 1. Ввести номер телефона в поле.  2. Нажать кнопку «Поиск». | Администратор получает все данные об искомом абоненте. | Администратор получает все данные об искомом абоненте. |

Продолжение таблицы 3.1-План тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **9** | Поиск по номеру телефона (негативный) | 1. Администратор находится на странице поиска абонента  2. Абонент не был ранее зарегистрирован в системе | 1. Ввести номер телефона в поле.  2. Нажать кнопку «Поиск». | Администратор на странице поиска получает сообщение «Данный абонент не зарегистрирован» | Ничего не выводится. |
| **10** | Добавление тарифа (позитивный) | 1.Администратор находится на странице создания тарифа | 1. Ввести в необходимые поля «Название», «Стоимость», «Описание» данные.  2. Нажать на кнопку «Добавить тариф». | Пользователь на странице выбора получает сообщение «Тариф добавлен успешно!» | Пользователь на странице выбора получает сообщение «Тариф добавлен успешно!» |
| **11** | Добавление тарифа (негативный) | 1.Администратор находится на странице создания тарифа | 1. Оставить одно из полей незаполненным.  2. Нажать на «Добавить тариф». | Администратор получает сообщение «Заполнены не все поля!» | Администратор получает сообщение «Заполнены не все поля!» |

Продолжение таблицы 3.1-План тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **12** | Выбор тарифа (позитивный) | 1. Пользователь находится на странице с выбором тарифных планов. | 1. Нажать на нужный тариф. | Выбранный тарифный план успешно добавляется и отображается в данных абонента. | Выбранный тарифный план успешно добавляется и отображается в данных абонента. |
| **13** | Смена тарифа (позитивный) | 1. Пользователь находится на странице с выбором тарифных планов. | 1. Нажать на нужный тариф. | Выбранный тарифный план успешно добавляется и отображается в данных абонента. | Выбранный тарифный план успешно добавляется и отображается в данных абонента. |
| **14** | Экспорт данных о тарифах (позитивный) | 1.Администратор находится на странице со списком всех тарифов. | 1.Нажать на кнопку «Экспорт в Excel» | Открывается Проводник, где можно создать новый файл, куда экспортируется таблица с данными о тарифах из базы данных.  После создания/выбора файла появляется надпись «Успешно! Данные экспортированы» | Открывается Проводник, где можно создать новый файл, куда экспортируется таблица с данными о тарифах из базы данных.  После создания/выбора файла появляется надпись «Успешно! Данные экспортированы» |

Продолжение таблицы 3.1-План тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **15** | Экспорт данных об абонентах (позитивный) | 1.Администратор находится на странице со списком всех абонентов. | 1.Нажать на кнопку «Экспорт в Excel» | Открывается Проводник, где можно создать новый файл, куда экспортируется таблица с данными об абонентах из базы данных.  После создания/выбора файла появляется надпись «Успешно! Данные экспортированы» | Открывается Проводник, где можно создать новый файл, куда экспортируется таблица с данными б абонентах из базы данных.  После создания/выбора файла появляется надпись «Успешно! Данные экспортированы» |