**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА**

Реализация информационной системы

«Офисные телефоны»

*Москва, 2024 г*

**Описание предметной области**

Служебные телефоны.

Компания хранит сведения о своих сотрудниках в формате: уникальный номер сотрудника, фамилия, адрес, дата рождения, название должности. Дата зачисления на должность, номер отдела, в котором работает сотрудник.

Каждому сотруднику компания предоставляет служебную сотовую связь (ССС) в формате тарифа, включающего в себя число минут, Гб и смс в месяц. Хранится код сотрудника, номер телефона и ID тарифа.

Каждый месяц компания получает счет от оператора мобильной связи. Каждый счет имеет номер, дату выставления, месяц и год, за который выставлен счет, общую сумму счета. В строках счета перечислены номера телефонов, кол-во использованных минут, Гб, смс и сумма к оплате по каждому телефону за указанный период.

Компания сама оплачивает эти счета оператору мобильной связи.

Каждый месяц после получения счета в компании формируются данные о превышениях лимита. Заносится номер телефона, сумма превышения лимита и месяц, и год, в котором это превышение произошло.

Превышение лимита должно быть оплачено самим сотрудником. Данные об оплате также хранятся в превышениях лимита. Для этого предусмотрена дата погашения лимита превышения лимита. Если дата погашения отсутствует, то сотрудник еще не погасил задолженность.

Оплата производится сотрудниками через банк, из которого каждый месяц поступает платежка. Каждая платежка имеет номер и дату создания. В каждой строке платежки указаны: номер телефона, полученная сумма, месяц и год, за который сделан платеж.

**Логическая модель БД**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Техническое задание**

**Техническое задание на этап проектирования**

1. Определить конечных пользователей будущей системы.
2. Составить UML-диаграмму вариантов использования.
3. Выделить основной вариант использования информационной системы (основной бизнес-процесс в предметной области).
4. Разработать систему авторизации пользователей ИС.
5. Разработать системную архитектуру ИС.
6. Для всех вариантов использования разработать главные успешные сценарии и расширения к ним.
7. Разработать системные UML-диаграммы последовательности для всех сценариев с использованием MVC-паттерна.
8. Разработать требования ко всем шаблонам для каждого варианта использования.
9. Разработать инфологическую модель предметной области в форме UML-диаграммы классов.
10. Разработать логическую модель будущей базы данных.

**Техническое задание на этап реализации**

1. Реализовать разработанную на этапе проектирования информационную систему на языке Python в среде фреймворка Flask.
2. Каждый вариант использования оформить, как blueprint.
3. Доступ внутренних пользователей к вариантам использования реализовать с помощью декораторов.

**Основной бизнес-процесс**

Интерфейс для корректировки тарифов сотрудников: выдать тариф, просмотреть свой тариф, просмотреть тариф выбранного сотрудника.

**Сущности системы**:

* Работник
* Счет
* Строки счета
* Платежка
* Строки платежки
* Служебная сотовая связь
* Превышение лимита
* Тариф
* Отчет

Все это записано в виде реляционной базы данных и соединено в информационную систему.

**Конечные пользователи ИС**

Внутренние пользователи: менеджеры, директоры, ассистенты

1. Менеджер. Создает отчеты, просматривает отчеты, просматривает свой тариф
2. Директор. Просматривает отчеты, просматривает свой тариф
3. Ассистент. Выполняет запросы, просматривает свой тариф
4. Администратор. Выдает тариф, просматривает тарифы сотрудников

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

**Сценарий использования «Авторизация»**

1. Карточка варианта

* Название: Авторизация
* Предисловие: Пользователь есть в системе и для него корректно прописаны права
* Гарантия: Пользователь успешно авторизуется как внутренний и переходит к выбору действий для соответствующей группы пользователей
* Минимальная гарантия: В случае ошибки пользователь получит сообщение о том, что были введены неправильные данные

1. Успешный сценарий
2. Неавторизованный пользователь заходит в систему и попадает на вариант использования «Авторизация»
3. Система присылает форму для ввода логина и пароля
4. Пользователь вводит свой логин и пароль и отправляет их
5. Система идентифицирует и авторизует пользователя
6. Система перенаправляет пользователя на главное меню
7. Исключения
   1. Пользователь не ввел логин или пароль. Система перенаправляет на страницу с сообщением «Логин или пароль не получен» и предлагает вернуться на страницу с авторизацией для повторного ввода логина и пароля.
   2. Пользователь вводит неверный логин или пароль. Система перенаправляет на страницу с сообщением «Данные о пользователе не найдены. Доступ запрещен» и предлагает вернуться на страницу с авторизацией для повторного ввода логина и пароля.
8. Тестовые данные

Для успешного ввода:

|  |  |
| --- | --- |
| **логин** | **пароль** |
| man\_1 | m12345 |
| man\_2 | m23456 |
| adm\_1 | admin1 |
| dir\_1 | d12345 |
| ass\_1 | a12345 |
| ass\_2 | a23456 |

Для неуспешного ввода:

|  |  |
| --- | --- |
| **логин** | **пароль** |
|  | m12345 |
| dir\_1 |  |
| ass\_2 | as234567 |

1. Файловая архитектура реализации варианта использования

├───auth

│ model\_route.py – файл, вызывающий функции для обработки sql запроса и работы с БД

│ и отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ user.sql – содержит шаблон sql запроса для получения id пользователя и его уровня

│ доступа

│

├───static

│ sem1.css – содержит шаблон sql запроса для получения id пользователя и его уровня

│ доступа

│

└───templates

auth\_error.html – страница, содержащая сообщение об ошибке и кнопку для возврата на

страницу для ввода данных

input\_auth.html – страница содержащая поля для ввода логина и пароля и кнопку для

авторизации

1. Список необходимых шаблонов с требованиями

* input\_auth.html – страница для ввода данных. По центру расположен заголовок «Ввод данных» и два поля для ввода, подписанные соответственно, как «логин» и «пароль». Под полями для ввода находится кнопка «Авторизоваться», для отправки введенных данных.

Шаблон содержит поля: login типа text для ввода логина пользователя, password типа password для ввода пароля пользователя. Форма отправляет данные на сервер с помощью метода POST. Данные будут отправлены на текущий URL.

* auth\_error.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопка для перехода на страницу для повторного ввода логина и пароля.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Ввести другой логин или пароль» с URL: {{ url\_for('auth\_bp.auth\_index') }} - ссылка на форму авторизации;

1. Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описание

**Сценарий использования «Взаимодействие с главным меню»**

1. Карточка варианта

* Название: Взаимодействие с главным меню
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь заходит на страницу главного меню и переходит к выбору опций разных взаимодействий с интерфейсом

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Главное меню»

2. Система присылает страницу с главным меню с сообщением об авторизации под соответствующей ролью и выбором кнопок

3. Пользователь выбирает кнопку и нажимает на нее

4. Система перенаправляет пользователя на страницу работы с выбранным вариантом взаимодействия

1. Исключения

Для данного варианта взаимодействия исключений нет.

1. Тестовые данные

Для успешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Функциональность** |
| assistant | «Главное меню» |
| manager | «Главное меню» |

1. Файловая архитектура реализации варианта использования

├───main\_dir

│ app.py – основной файл, рендерящий html страницу главного меню и

│ перенаправляющий на работу с запросами

│

├───static

│ sem1.css – содержит описание стилей элементов html страниц

│

└───templates

main\_menu.html – страница содержащая кнопки для перехода к работе с запросами и

выхода из системы

main\_menu\_error.html – страница с уведомлением об отсутствии прав доступа к

данному варианту взаимодействию и кнопками для возврата в

меню и повторной авторизации

1. Список шаблонов

* main\_menu.html – страница главного меню. Вверху находится заголовок «Главное меню». Под ним находится подпись, уведомляющая о роли авторизации. Ниже располагаются кнопки для работы с разными запросами, кнопка для работы с отчетами, и последней кнопкой является кнопка выхода из системы

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Ввести другой логин или пароль» с URL: auth\_bp.auth\_index - ссылка на форму авторизации;

Кнопку «Меню запросов» с URL 'query\_bp.query\_menu'

Кнопку «Работа с отчетами» с URL: 'auth\_report\_bp.report\_menu'

Кнопку «Работа с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Нажмите для выхода из системы» с URL: exit\_func'

* main\_menu\_error.html – страница с уведомлением об отсутствии прав доступа к данному запросу. Уведомление расположено по центру, под ним находится кнопка для возвращения в главное меню и кнопка для повторной авторизации.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Вернуться на главное меню» с URL: 'main\_menu'

Кнопку «Ввести другой логин или пароль» с URL: 'auth\_bp.auth\_index'

1. Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Изображение выглядит как текст, линия, снимок экрана, диаграмма

Автоматически созданное описание

**Сценарий использования «Взаимодействие с меню запросов»**

1. Карточка варианта

* Название: Взаимодействие с меню с запросами
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь заходит на страницу меню запросов и переходит к выбору запроса
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Меню запросов»

2. Пользователь находится на странице с главным меню с сообщением об авторизации под соответствующей ролью и выбором кнопок

3. Пользователь выбирает кнопку «Меню запросов» и нажимает на нее

4. Система перенаправляет пользователя на страницу с выбором запросов

1. Исключения

4.1. Пользователь не имеет прав для работы с меню запросов. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другие логин и пароль.

1. Тестовые данные

Для успешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| assistant | «Меню запросов» |

Для неуспешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| manager | «Меню запросов» |
| director | «Меню запросов» |

1. Файловая архитектура реализации варианта использования

│

└──────query

├───templates

│ query\_menu.html – страница содержащая кнопки для перехода к работе с запросами и

возвращения в главное меню

1. Список шаблонов

* query\_menu.html – страница меню запросов. Вверху находится заголовок «Меню запросов». Под ним располагаются кнопки для работы с разными запросами и последней кнопкой является кнопка для перехода в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Показать превышение лимита по каждому телефону» с URL: 'query\_bp.query\_index'

Кнопку «Показать все счета, выставленные за последние n дней» с URL: 'query\_bp.query\_index1'

Кнопку «Список сотрудников» с URL: 'query\_bp.query\_index2'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

1. Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

**Сценарий использования «Взаимодействие с запросом» на примере запроса «Показать превышение лимита по каждому телефону»**

1. Карточка варианта

* Название: Взаимодействие с запросом
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь выполняет запрос и переходит к выбору «выполнить запрос заново», «перейти в меню запросов» или «вернуться в главное меню»
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования взаимодействие с запросом «Показать превышение лимита по каждому телефону»

2. Система присылает страницу с формой ввода года и месяца

3. Пользователь вбивает год и месяц

4. Система перенаправляет пользователя на страницу с результатом запроса

1. Исключения

4.1. Пользователь не имеет прав для работы с выбранным запросом. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню.

4.2. Пользователь не ввел месяц или год. Система перенаправляет на страницу с сообщением «Временной промежуток не получен» и предлагает перейти к повторному вводу, перейти к меню запросов или перейти в главное меню.

4.3. Пользователь ввел некорректные данные месяца или года. Система перенаправляет на страницу с сообщением указания на ошибку и предлагает перейти к повторному вводу, перейти к меню запросов или перейти в главное меню.

1. Тестовые данные

Для успешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Месяц, год** |
| assistant | «Показать требования лимита по каждому телефону» | 2020, 3 |
| assistant | «Показать требования лимита по каждому телефону» | 2020, 4 |
| assistant | «Показать все счета, выставленные за последние n дней» | 1000 |
| assistant | «Список сотрудников» | manager |

Для неуспешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Месяц, год** |
| assistant | «Показать требования лимита по каждому телефону» | 2023, 2 |
| assistant | «Показать требования лимита по каждому телефону» | 1800, 4 |
| assistant | «Показать требования лимита по каждому телефону» | 2020, |

1. Файловая архитектура реализации варианта использования

├───query

│ model\_route.py – файл, вызывающий функции для обработки sql запроса и работы с БД

│ и отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ bill.sql – содержит шаблон sql запроса для получения номеров счета, выставленных

│ за последние n дней

│ product.sql – содержит шаблон sql запроса для получения списка номеров телефонов и

│ суммы превышения лимита за заданный год и месяц

│ workers.sql – содержит шаблон sql запроса для получения списка работников

│ определенной должности

│

├───static

│ sem1.css – содержит описание стилей элементов html страниц

│

├───templates

dynamic.html – страница для отображения результатов запроса

dynamic\_bill\_days.html – страница для отображения результатов запроса

dynamic\_role.html – страница для отображения результатов запроса

error.html - страница для отображения ошибок при выполнении запроса

error\_bill.html - страница для отображения ошибок при выполнении запроса

error\_worker.html - страница для отображения ошибок при выполнении запроса

input\_bill\_days.html – страница, содержащая поля для ввода параметров для запроса

input\_category.html – страница, содержащая поля для ввода параметров для запроса

input\_role.html – страница, содержащая поля для ввода параметров для запроса

1. Список необходимых шаблонов с требованиями

* dynamic.html – страница для ввода данных. По центру расположены два поля для ввода, подписанные соответственно, как «введите год» и «введите месяц». Под полями для ввода находятся кнопка «выбрать» для отправки введенных данных и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Переменную title, которая передается из backend в шаблон для отображения заголовка страницы;

Переменную exceeds, которая передается из backend для отображения данных в таблице

Кнопку «Перейти к вводу временного промежутка» с URL: 'query\_bp.query\_index'

Кнопку «Перейти к меню запросов» с URL: 'query\_bp.query\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* error.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопки для перехода на страницу для повторного ввода параметров запрос, для перехода в меню запросов и для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Перейти к вводу» с URL: 'query\_bp.query\_index'

Кнопку «Перейти к меню запросов» с URL: 'query\_bp.query\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

7) Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Изображение выглядит как текст, диаграмма, Технический чертеж, План

Автоматически созданное описание

**Сценарий использования «Взаимодействие с меню отчетов»**

1. Карточка варианта

* Название: Взаимодействие с меню отчетов
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь заходит на страницу меню отчетов и переходит к выбору создания или просмотра отчетов

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Меню отчетов»

2. Система присылает страницу с меню с выбором кнопок «создать отчет» и «посмотреть отчет»

3. Пользователь выбирает кнопку и нажимает на нее

4. Система перенаправляет пользователя на страницу создания или просмотра отчета

1. Исключения

4.1. Пользователь не имеет прав для работы с меню запросов. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другие логин и пароль.

1. Тестовые данные

Для успешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Функциональность** |
| director | «Меню отчетов» |
| manager | «Меню отчетов» |

Для неуспешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| admin | «Меню отчетов» |
| assistant | «Меню отчетов» |

1. Файловая архитектура реализации варианта использования

─report

│

└─────templates

report\_menu.html

* templates
  + report\_menu.html – страница содержащая кнопки для перехода к созданию и просмотру отчетов

1. Список шаблонов

* report\_menu.html – страница меню отчетов. Вверху находится заголовок «Главное меню». Под ним находятся кнопка для создания отчетов и ниже кнопка для просмотра отчетов

Шаблон содержит:

Кнопку «Создание отчетов» с URL: 'report\_bp.report\_index'

Кнопку «Просмотр отчетов» с URL: 'report\_bp.report\_index2'

1. Системная диаграмма последовательности, составленная с использованием MVC-паттерна

**Изображение выглядит как текст, линия, диаграмма, снимок экрана

Автоматически созданное описание**

**Сценарий использования «Взаимодействие с отчетами» на примере «Создание отчета1»**

1. Карточка варианта

* Название: Создание отчета1
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь создает отчет и переходит к выбору «создать другой отчет», «перейти к работе с отчетами» или «вернуться в главное меню»
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Взаимодействие с отчетами»

2. Система присылает страницу с формой выбора типа отчета, года и месяца

3. Пользователь выбирает тип отчета, год и месяц

4. Система перенаправляет пользователя на страницу с результатом выполнения отчета

1. Исключения

4.1. Пользователь не имеет прав для работы с отчетами. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другой логин или пароль.

4.2. Ввел неправильные данные отчета. Система перенаправляет на страницу с сообщением «Отчет за m месяц и y год не найден» и предлагает посмотреть другой отчет, перейти к работе с отчетами или вернуться на страницу с главным меню.

4.3. Отчет по введенным данным уже существует. Система перенаправляет на страницу с сообщением «Отчет за m месяц и y год уже сущетвует» » и предлагает посмотреть другой отчет, перейти к работе с отчетами или вернуться на страницу с главным меню.

1. Тестовые данные

Для успешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Тип отчета, месяц, год** |
| manager | «Создать отчет» | Отчет1, июль, 2023 |
| manager | «Создать отчет» | Отчет1, март, 2020 |
| manager | «Посмотреть отчет» | Отчет1, март, 2020 |
| manager | «Посмотреть отчет» | Отчет2, сентябрь, 2021 |
| manager | «Посмотреть отчет» | Отчет2, сентябрь, 2021 |

Для неуспешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Тип отчета, месяц, год** |
| manager | «Посмотреть отчет» | Отчет2, январь, 2022 |
| manager | «Создать отчет» | Отчет1, ноябрь, 2024 |

1. Файловая архитектура реализации варианта использования

├──report

│ model\_route.py - файл, вызывающий функции для обработки sql запроса и работы с БД и

│ отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ report2\_view.sql – содержит шаблон sql запроса для просмотра отчета №2 (платежки)

│ report\_create.sql – содержит шаблон sql запроса для создания отчета

│ report\_view.sql – содержит шаблон sql запроса для просмотра отчета №1

│ (превышения)

│

├───static

│ sem1.css – содержит описание стилей элементов html страниц

│

└───\_\_templates

input\_report.html – страница, содержащая поля для выбора параметров для отчета

message.html – страница для отображения сообщения о результате создания отчета

output\_report.html – страница для отображения результатов просмотра отчета

rep\_error2.html – страница для отображения ошибок при просмотре отчета

1. Список необходимых шаблонов с требованиями (пример для запроса «Показать превышение лимита по каждому телефону»)

* input\_report.html – страница для ввода данных. По центру расположены три поля для ввода, подписанные соответственно, как «отчет», «месяц» и «год». Под полями для ввода находятся кнопка «отправить» для отправки введенных данных, а также «перейти к работе с отчетами» для выбора создания/просмотра отчета и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Поле rep\_id для выбора отчета

Поле rep\_month для выбора месяца

Поле rep\_year для выбора года

Метода POST, с помощью которого форма отправляет данные на сервер. Данные будут отправлены на текущий URL

Кнопку для отправки формы «Отправить» - кнопка отправляет форму на сервер

Кнопку «Перейти к работе с отчетами» с URL: 'report\_bp.report\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* message.html – страница для отображения сообщения о результате создания отчета. По центру расположено сообщение о результатах создания отчета. Ниже находятся три кнопки: «создать другой отчет» для повторного ввода параметров отчета, «перейти к работе с отчетами» для выбора создания/просмотра отчета и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщений пользователю;

Кнопку «Создать другой отчет» с URL: 'report\_bp.report\_index'

Кнопку «Перейти к работе с отчетами» с URL: 'report\_bp.report\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* output\_report.html – страница для отображения результатов просмотра отчета

Шаблон содержит:

Переменную prod\_title, которая передается из backend в шаблон для отображения заголовка страницы;

Переменная columns, которая передается из backend для отображения заголовков столбцов таблицы.

Переменная products, которая передается из backend для отображения данных в таблице.

Кнопку «Посмотреть другой отчет» с URL: 'report\_bp.report\_index2'

Кнопку «Перейти к работе с отчетами» с URL: 'report\_bp.report\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* rep\_error2.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопки для перехода на страницу для повторного ввода параметров просмотра отчета, для перехода к работе с запросами и для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщения страницы;

Кнопку «Посмотреть другой отчет» с URL: 'report\_bp.report\_index2'

Кнопку «Перейти к работе с отчетами» с URL: 'report\_bp.report\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

7) Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Создание отчета:

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

Просмотр отчета:

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

**Основной бизнес-процесс**

**Сценарий использования «Взаимодействие с меню работы с тарифами»**

1. Карточка варианта

* Название: Взаимодействие с меню работы с тарифами
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь заходит на страницу меню работы с тарифами и переходит к выбору просмотра своего тарифа или раздачи тарифа

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Меню работы с тарифами»

2. Система присылает страницу с меню с выбором кнопок «посмотреть свой тариф» и «выдать тариф»

3. Пользователь выбирает кнопку и нажимает на нее

4. Система перенаправляет пользователя на страницу просмотра или раздачи тарифа

1. Исключения

Для данного варианта взаимодействия исключений нет.

1. Тестовые данные

Для успешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Функциональность** |
| assistant | «Меню работы с тарифами» |
| manager | «Меню работы с тарифами» |

1. Файловая архитектура реализации варианта использования

─report

│

└─────templates

business\_menu.html

* templates
  + business\_menu.html – страница содержащая кнопки для перехода на страницу просмотра или раздачи тарифа

1. Список шаблонов

* business\_menu.html – страница меню работы с тарифами. Вверху находится заголовок «Работа с тарифами». Под ним находятся кнопка для просмотра своего тарифа и ниже кнопка для раздачи тарифов

Шаблон содержит:

Кнопку «Посмотреть свой тариф» с URL: 'business\_bp.tariff\_check'

Кнопку «Посмотреть тариф сотрудника» с URL: 'business\_bp.tariff\_view'

Кнопку «Выдать тариф» с URL: 'business\_bp.tariff\_change'

1. Системная диаграмма последовательности, составленная с использованием MVC-паттерна

**Изображение выглядит как текст, линия, снимок экрана, диаграмма

Автоматически созданное описание**

**Сценарий использования «Просмотр своего тарифа»**

1. Карточка варианта

* Название: Просмотр своего тарифа
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь просматривает данные о своем тарифе и переходит к выбору «перейти к работе с тарифами» или «перейти к главному меню»
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Просмотр своего тарифа»

2. Система присылает страницу с данными о тарифе авторизованного пользователя

1. Исключения

4.1. Пользователь не имеет прав для просмотра своего тарифа. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другой логин или пароль.

1. Тестовые данные

Для успешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| manager | «Посмотреть свой тариф» |
| manager | «Посмотреть свой тариф» |

Для неуспешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| admin | «Посмотреть свой тариф» |

1. Файловая архитектура реализации варианта использования

├──business

│ model\_route.py файл, вызывающий функции для обработки sql запроса и работы с БД и

│ отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ tariff\_view.sql - содержит шаблон sql запроса для просмотра своего тарифа

│

├───static

│ sem1.css – содержит описание стилей элементов html страниц

│

└───\_\_templates

output\_tariff.html – страница для отображения результатов просмотра своего тарифа

bus\_rep\_error2.html – страница для отображения ошибок при просмотре тарифа

1. Список необходимых шаблонов с требованиями (пример для запроса «Показать превышение лимита по каждому телефону»)

* output\_tariff.html – страница для отображения результатов просмотра тарифа

Шаблон содержит:

Переменную prod\_title, которая передается из backend для отображения заголовка страницы

Переменную columns, которая передается из backend для отображения данных в таблице.

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* bus\_rep\_error2.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопки для перехода на страницу для работы с отчетами и для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщения страницы;

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

7) Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Просмотр тарифа:

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описание

**Сценарий использования «Просмотр тарифа сотрудника»**

1. Карточка варианта

* Название: Просмотр тарифа сотрудника
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь просматривает данные о тарифе выбранного сотрудника и переходит к выбору «посмотреть тариф другого сотрудника», «перейти к работе с тарифами» или «перейти к главному меню»
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Просмотр тарифа сотрудника»

2. Система присылает страницу с данными о тарифе выбранного сотрудника

1. Исключения

2.1. Пользователь не имеет прав для просмотра тарифов сотрудников. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другой логин или пароль.

2.2 Выбранному сотруднику не выдан тариф. Система перенаправляет на страницу с соответствующим сообщением и предлагает выбору «посмотреть тариф другого сотрудника», «перейти к работе с тарифами» или «перейти к главному меню»

1. Тестовые данные

Для успешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Сотрудник** |
| admin | «Посмотреть тариф сотрудника» | Potter |
| admin | «Посмотреть тариф сотрудника» | Malfoy |

Для неуспешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Сотрудник** |
| admin | «Посмотреть свой тариф» | Lovegood |

1. Файловая архитектура реализации варианта использования

├──business

│ model\_route.py - файл, вызывающий функции для обработки sql запроса и работы с БД и

│ отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ tariff\_view.sql - содержит шаблон sql запроса для просмотра тарифа

│ выбранного работника

│ get\_name.sql - содержит шаблон sql запроса для получения фамилии сотрудника по id

├───static

│ sem1.css – содержит описание стилей элементов html страниц

│

└───templates

bus\_error3.html – страница для отображения ошибок при просмотре тарифа

input\_view\_worker.html - страница, содержащая поля для выбора фамилии работника

output\_worker\_tariff.html – страница для отображения результатов просмотра

тарифа выбранного работника

1. Список необходимых шаблонов с требованиями (пример для запроса «Показать превышение лимита по каждому телефону»)

* output\_worker\_tariff.html – страница для отображения результатов просмотра тарифа выбранного работника. По центру расположено сообщение о результатах просмотра тарифа. Под ней таблица с результатами. Ниже находятся три кнопки: «Посмотреть тариф другого сотрудника» для повторного ввода параметров, «Перейти к работе с тарифами» для выбора посмотреть/выдать тариф и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Переменную title, которая передается из backend для отображения заголовка страницы

Переменную columns, которая передается из backend для отображения данных в таблице.

Кнопку «Посмотреть тариф другого сотрудника» с URL: 'business\_bp.tariff\_view'

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* bus\_error3.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопки для просмотра тарифа другого сотрудника, перехода на страницу для работы с тарифами и для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщения страницы;

Кнопку «Посмотреть тариф другого сотрудника» с URL: 'business\_bp.tariff\_view'

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* input\_view\_worker.html – страница, содержащая поля для выбора фамилии сотрудника. По центру расположено одно поле для ввода, подписанное, как «сотрудник». Под полем для ввода находятся кнопка «выбрать» для отправки введенных данных, а также «вернуться к работе с тарифами» для выбора просмотра/выдачи тарифа и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Поле worker для выбора сотрудника

Кнопка "Выбрать" для оправки данных на текущий URL с помощью метода POST

Кнопку «Вернуться к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

7) Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Просмотр тарифа сотрудника:

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

**Сценарий использования «Раздача тарифа»**

1. Карточка варианта

* Название: Раздача тарифа
* Предисловие: Пользователь авторизован и его права определены.
* Гарантия: Пользователь выдает тариф другому пользователю и переходит к выбору «выдать новый тариф», «перейти к работе с тарифами» или «перейти к главному меню»
* Минимальная гарантия: В случае ограничения прав пользователь получит соответствующее сообщение.

1. Успешный сценарий

1. Авторизованный пользователь начинает сценарий использования «Раздача тарифа»

2. Система присылает страницу с формой выбора фамилии сотрудника и номера тарифа

3. Пользователь выбирает фамилию сотрудника и номер тарифа

4. Система перенаправляет пользователя на страницу с сообщением о выдаче тарифа

1. Исключения

4.1. Пользователь не имеет прав на раздачу тарифов. Система перенаправляет на страницу с сообщением «У вас нет доступа на эту функциональность» и предлагает вернуться на страницу с главным меню или ввести другой логин или пароль.

Тестовые данные

Для успешного сценария:

|  |  |  |
| --- | --- | --- |
| **Роль пользователя** | **Выбранная функциональность** | **Сотрудник** |
| admin | «Выдать тариф» | Potter |

Для неуспешного сценария:

|  |  |
| --- | --- |
| **Роль пользователя** | **Выбранная функциональность** |
| assistant | «Выдать тариф» |
| director | «Выдать тариф» |

1. Файловая архитектура реализации варианта использования

├──business

│ model\_route.py файл, вызывающий функции для обработки sql запроса и работы с БД и

│ отлавливающий возможные ошибки

│ route.py – основной файл, рендерящий html страницы и запускающий model\_route.py

│

├───sql

│ tariff\_change.sql – содержит шаблон sql запроса для изменения тарифа

│

├───static

│ sem1.css

│

└───templates

input\_worker.html - страница, содержащая поля для выбора параметров выдачи тарифа (фамилия сотрудника и № тарифа)

bus\_rep\_error2.html – страница для отображения ошибок при выдаче тарифа

tariff\_message.html – страница для отображения сообщения о результате выдачи тарифа

1. Список необходимых шаблонов с требованиями

* input\_worker.html – страница для ввода данных. По центру расположены два поля для ввода, подписанные соответственно, как «сотрудник» и «тариф». Под полями для ввода находятся кнопка «выбрать» для отправки введенных данных, а также «перейти к работе с тарифами» для выбора просмотра/выдачи отчета и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Поле worker для выбора сотрудника

Поле tariff\_id для выбора тарифа

Кнопку «Выбрать» для отправки формы на сервер. Форма отправляет данные на сервер методом POST.  Данные будут отправлены на текущий URL.

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* tariff\_message.html – страница для отображения сообщения о результате выдачи тарифа. По центру расположено сообщение о результатах выдачи тарифа. Ниже находятся три кнопки: «выдать новый тариф» для повторного ввода параметров, «перейти к работе с тарифами» для выбора создания/просмотра тарифа и «Перейти к главному меню» для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщения страницы;

Кнопку «Выдать новый тариф» с URL: 'business\_bp.tariff\_change'

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

* bus\_rep\_error2.html – страница ошибки. По центру располагается сообщение, соответствующее ошибке. Под ним кнопки для перехода на страницу для работы с отчетами и для возвращения в главное меню.

Шаблон содержит:

Переменную message, которая передается из backend в шаблон для отображения сообщения страницы;

Кнопку «Перейти к работе с тарифами» с URL: 'business\_bp.business\_menu'

Кнопку «Перейти к главному меню» с URL: 'main\_menu'

7) Системная диаграмма последовательности, составленная с использованием MVC-паттерна

Выдача тарифа:

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

**Заключение**

В результате работы была создана информационная система на языке Python в среде фреймворка Flask для компании, которая предоставляет служебную сотовую связь сотрудником своей фирмы. Были разработаны варианты использования для пользователей с предусмотренными правами доступа. Также был реализован основной бизнес-процесс для внутренних пользователей, а именно просмотр тарифа и выдача тарифа сотрудникам фирмы.