



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
Робототехника и комплексная автоматизация
КАФЕДРА
РК6. Системы автоматизированного проектирования (САПР)

Курсовая работа по дисциплине:

«Разработка информационных систем»

на тему:

**Разработка информационной системы
«Порт»**

Студент РК6-54Б
(Группа)

(Подпись, дата)

Э.Э. Акопян
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Н.В. Пивоварова
(И.О.Фамилия)

2022 г.

Аннотация

Курсовая работа выполнена 07.12.2022г., студенткой гр. РК6-54Б Акопян. Э.Э.
Преподаватель – доцент, к.т.н. Пивоварова Н.В.

Курсовая работа выполнена на 34 страницах.

В данной курсовой работе была разработана информационная система для склада товаров в виде WSGI приложения в среде фреймворка Flask; содержит техническое задание с 2 этапами – этап планирования и этап реализации; в ходе работы пошагово реализовывались пункты тех. задания для двух этапов.

Ключевые слова: диаграмма, архитектура, вариант использования, модель, пользователь.

Оглавление

1.Техническое задание	4
2.Описание предметной области.....	5
3.Конечные пользователи ИС.....	6
4.UML-диаграмма вариантов использования.....	6
5.Описание вариантов использования.....	7
5.1. Работа с запросами	7
5.2. Авторизация.....	11
5.3. Работа с отчетами.....	14
5.4. Основной бизнес-процесс	18
5.5 Расписание.....	23
6.Глобальная архитектура приложения	25
7.Инфологическая модель предметной области в форме UML-диаграммы классов	26
8.Логическая_модель БД.....	28
Заключение	28
Список использованных источников	28

1.Техническое задание

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. На основании выделенных сценариев разработать инфологическую модель предметной области. Описать этапы разработки инфологической модели.
10. Построить логическую модель базы данных.
11. Разработать программную архитектуру информационной системы.
12. Разработать схему хранения модулей информационной системы в файловой системе веб сервера.

2.Описание предметной области

Вариант 28. Порт.

В порт набираются сотрудники, о каждом из которых известно: профессия, уникальный номер, фамилия, дата рождения, адрес, дата приема на работу и, может быть, дата увольнения.

У каждого сотрудника есть табель работы, в котором отмечается дата последнего выхода на работу и количество фактически отработанных часов за все время.

В порту есть причалы, о каждом из которых известно: его тип, длина, глубина у стенки.

В порт приходят корабли, о которых известны: его тип, тоннаж, порт приписки.

Каждый заход корабля в порт регистрируется. Фиксируется дата прихода в порт и ухода из него, причал, сотрудник, который проводил его в порт.

Если корабль требует разгрузки, то создается бригада из сотрудников порта, состав которой заносится в таблицу разгрузок БД. Для каждого члена бригады проставляется количество часов, отработанных на разгрузке. В дальнейшем эти данные переносятся в табель работы сотрудников.

3.Конечные пользователи ИС.

- **Стивидоры:**

отвечает за погрузочно-разгрузочные работы в порту, работает с запросами для получения информации о разгрузках и регистрации судов, назначает работников в бригаду. Реализует основной бизнес-процесс: набирает работников в бригаду для разгрузки.

- **Тельман:**

формирует отчеты по разгрузкам и регистрациям, работа с запросами.

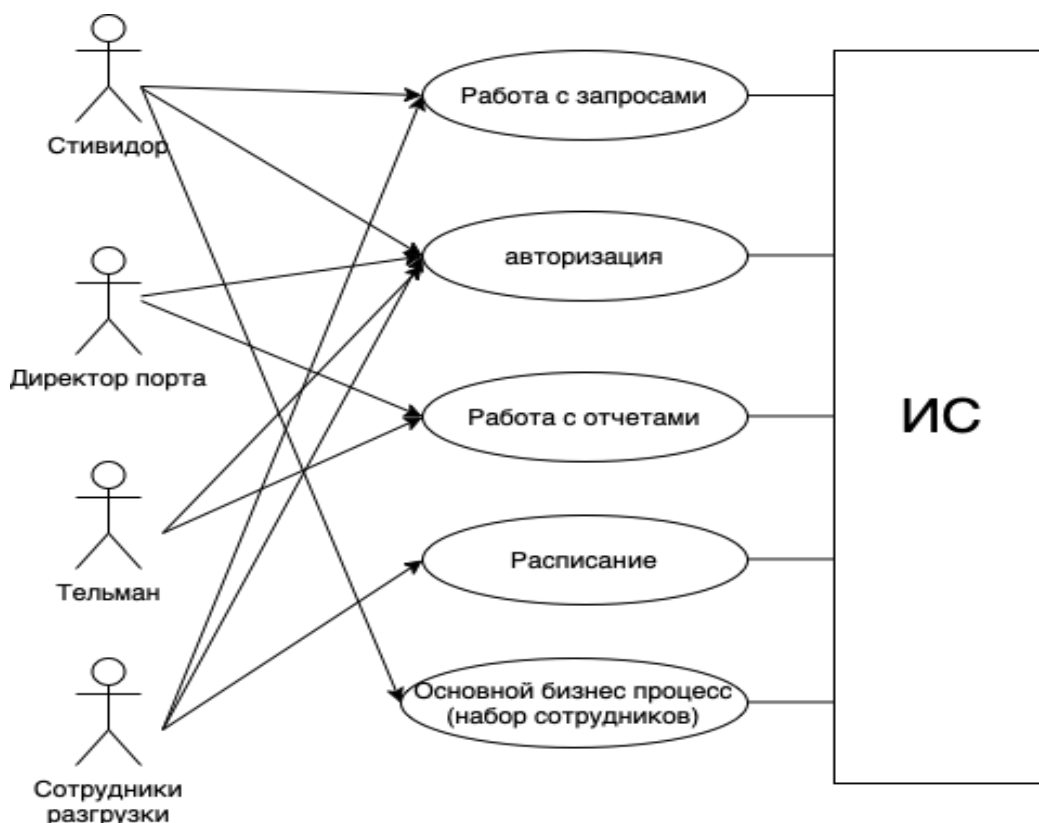
- **Директор порта:**

анализирует данные, в сформированных отчетах.

- **Сотрудники разгрузки:**

(внешний пользователь) просматривает расписание портовых работ в личном кабинете.

4.UML-диаграмма вариантов использования.



Основной сценарий использования

Основной сценарий использования информационной системы (бизнес-процесс) – это набор работников в бригаду для разгрузки.

5.Описание вариантов использования

5.1. Работа с запросами

1) Карточка варианта:

- 1) Название: “Выполнение запроса”
- 2) Предусловие: Авторизация пользователя
- 3) Гарантии: Выполнения select-запроса
- 4) Минимальные гарантии: при неверном вводе будет повторно запрошен ввод и не будет нарушена целостность базы данных.

2) Сценарий конечного пользователя:

- Пользователь входит в систему.
- Отображается форма с выбором варианта использования.
- Отображается форма с вариантами запроса.
- Пользователь начинает работу с нужным запросом.
- Отображение формы с вводом параметра(ов) запроса.
- Ввод данных для формирования sql-запроса:
 - а. При неверном вводе данных система укажет необходимый формат данных.
 - б. Отображения формы с результатами sql-запроса.

Частный сценарий конечного пользователя:

- Пользователь входит в систему.
- Отображается страница с кнопками “Завершить запросы” и “Начать запросы”.
- Пользователь нажимает на “Начать запросы”.
- Отображается страница с вариантами запроса “ Узнать о регистрации кораблей”, “ Узнать о работе сотрудников на разгрузках”, “Вернуться назад”.
- Пользователь нажимает на “ Узнать о регистрации кораблей” и видит формы с вводом параметров “Введите месяц”, “Введите год”, а также кнопку “Вернуться назад”.

а) Удачный сценарий:

- Пользователь вводит 5 месяц, 2020 год и нажимает “Отправить”.
- Пользователь видит описание в виде таблицы со столбцами “ID регистрации”, “Дата прибытия судна”, “Дата отправления судна”, “ID работника”, “ID причала”, “ID корабля” и кнопку “Вернуться к выбору запросов”. Содержимое таблицы:

1	2020-05-01	2020-05-02	3	9	5
2	2020-05-02	2020-05-04	4	10	6
3	2020-05-03	2020-05-04	3	11	7

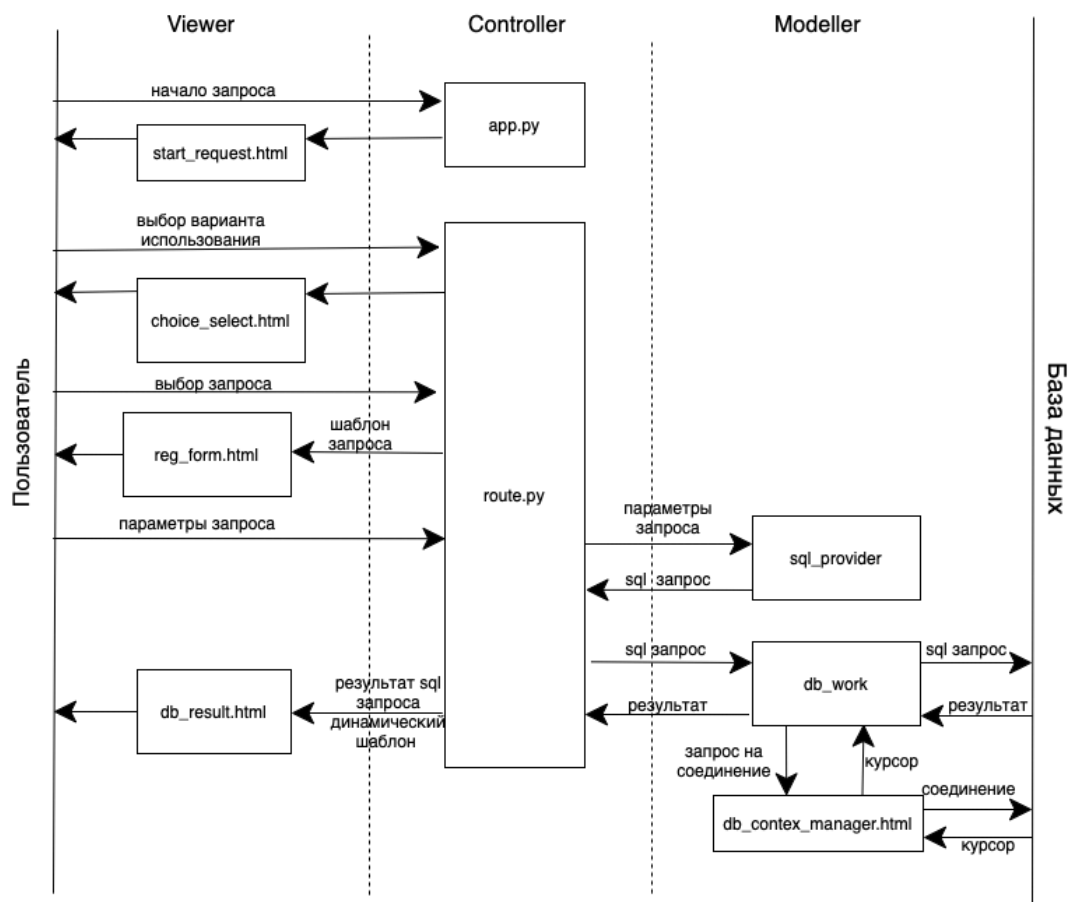
б) Неудачный сценарий:

- Пользователь вводит 6d месяц, 2020 год и нажимает “Отправить”.
- Система выводит сообщение о том, что число должно быть целое от 1 до 12.

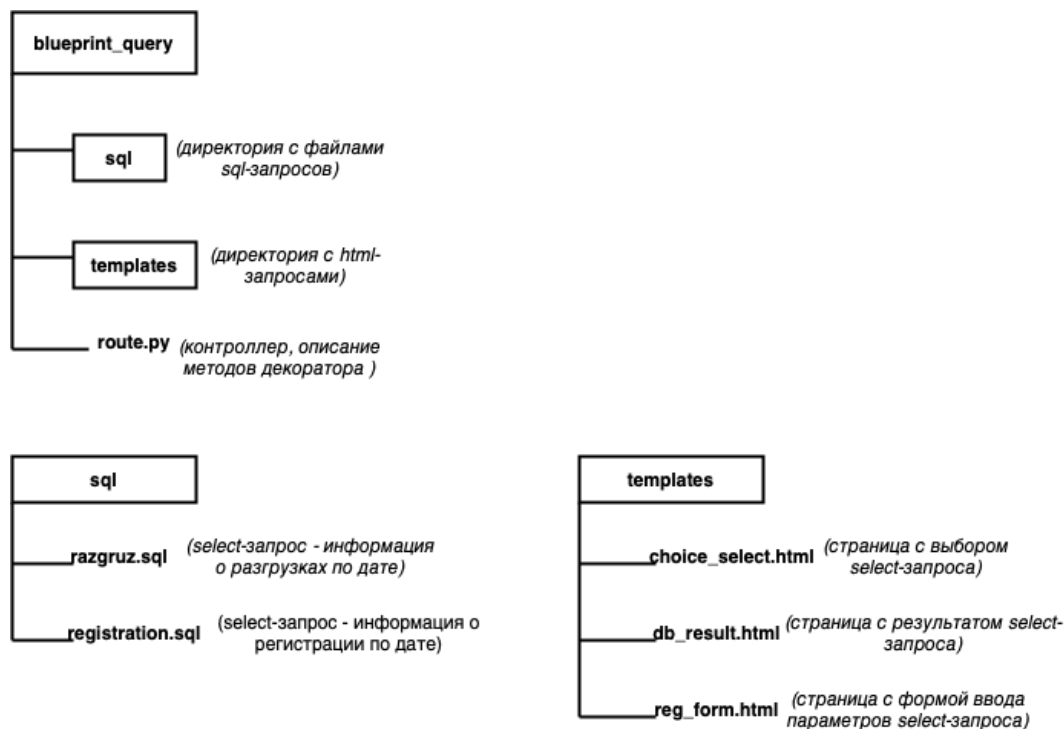
Запросы, реализованные в ИС:

1. Показать все сведения о регистрации кораблей, заходивших в порт в месяц/год.
2. Определить, сколько всего часов отработал на разгрузках каждый сотрудник за месяц/год .

3)UML системная диаграмма последовательности (работа с запросами)



4)Файловая структура(blueprint_query):



5)Требования к шаблону

5.1) Choice_select.html – реализует выбор необходимого запроса, переадресует на формы ввода параметров. На выбор представлены три ссылки: «Узнать о регистрации кораблей» (переадресует обработчик периода регистрации ‘*bp_query.queries*’), «Узнать о работе сотрудников на разгрузках» (переадресует обработчик временного периода разгрузок ‘*bp_query.querie2*’), «Вернуться назад» (для перехода предыдущую html страницу, ссылается на ‘*menu_choice*’). Заголовок шаблона «Какую информацию вы хотели бы получить?» и ссылки на варианты использования расположены сверху по центру. Ссылки имеют синий цвет, при наведении курсора- увеличивается размер шрифта, цвет меняется на белый.

5.2) reg_form.html – шаблон для ввода переменных. Сверху по центру заголовок «Введите временные рамки». Два поля ввода (сверху по центру) с ограничениями: подпись «Введите год» с полем «*year_name*», принимает натуральные числа от 1900 включительно, подпись «Введите месяц » с полем «*month_name*» принимает натуральное число от 1 до 12 включительно.

Обе формы обязательны для заполнения.

Также шаблон включает в себя кнопку «Отправить», расположенную под полями ввода по центру, и ссылку «Вернуться назад» (для перехода на предыдущую html страницу, ссылается на 'br_query.choice'). Также страница должна отображать переменную *message* с сообщениями об ошибках. Ссылки имеют синий цвет, при наведении курсора- увеличивается размер шрифта, цвет меняется на белый.

5.3) db_result.html - страница с результатами sql-запроса. Сверху в центре расположена надпись “Описание”. Под ней таблица с результатами запроса. Названия столбцов *schema* (список), результат запроса *result*(кортеж) и заголовок страницы *search*(строка) передаются шаблону обработчиком.

5.2. Авторизация

1) Карточка варианта:

- Название: “Авторизация”.
- Предусловие: пользователь зарегистрирован и находится в таблице БД.
- Гарантия: пользователь успешно прошел идентификацию.
- Минимальная гарантия: пользователь получает сообщение об ошибке и возможность повторить попытку.

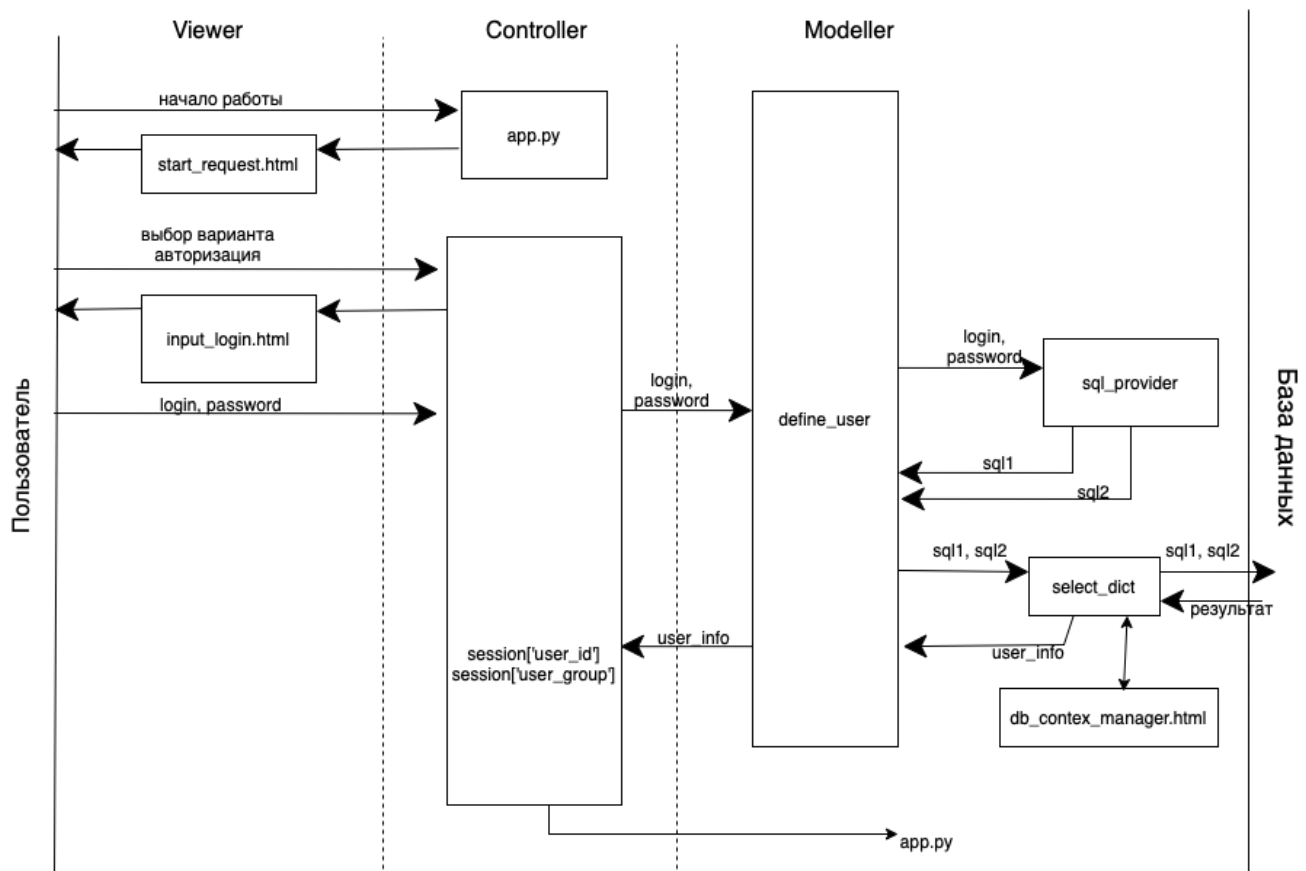
2) Сценарий конечного пользователя

- Пользователь зашел на страницу авторизации.
- Система присылает форму ввода логина и пароля.
- Пользователь вводит логин и пароль и отправляет их.
 - 1) При неверном вводе пользователь получает сообщение, что ввод необходимо повторить.
- Система открывает пользователю доступное меню.
- Пользователю открывается меню.

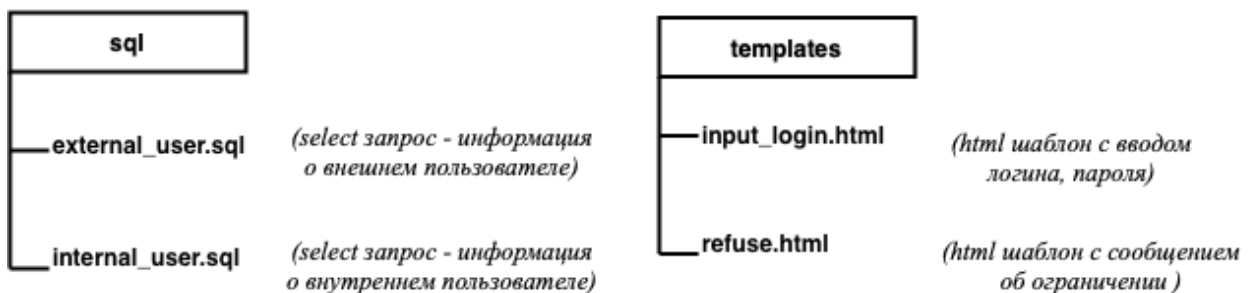
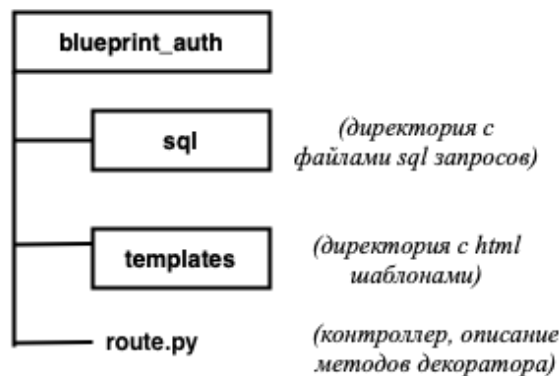
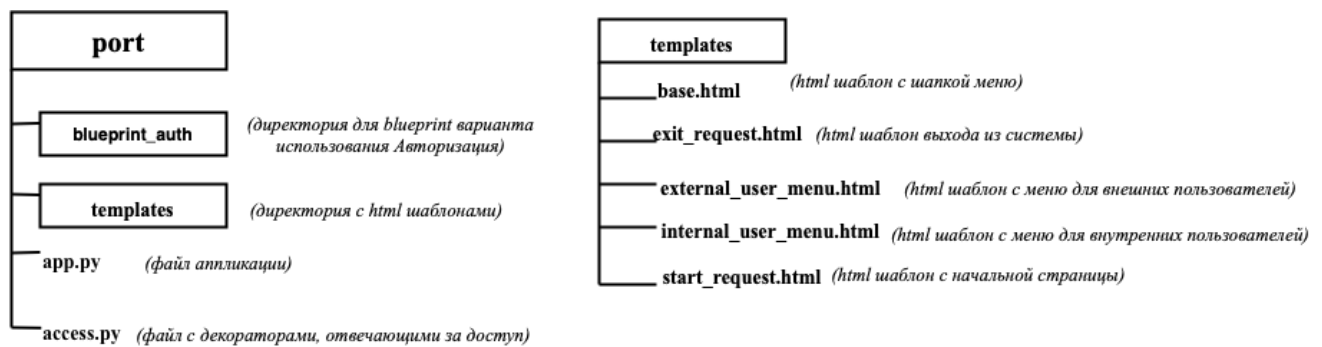
Частный сценарий конечного пользователя:

- Пользователь находится на начальной странице и видит перед собой три ссылки – “Авторизация”, “О порте”, “Завершить работу”.
- Пользователь нажимает на “Авторизация”. Видит перед собой надпись “Введите данные”. Под надписью находятся две формы для логина и пароля.
- Пользователь вводит логин – “dir” и пароль “dir” и жмет на кнопку “Отправить”.
- Пользователь успешно авторизовался как внутренний – Директор порта. Перед ним появилось меню с доступными действиями: “Начать работу с запросами”, “Начать работу с отчетами”, “Начать набор бригады на разгрузку” и “Нажмите для выхода из системы”.

3)UML системная диаграмма последовательности(авторизация)



4)Файловая структура(blueprint_auth):



5) Требования к шаблону

1) **input_login.html** – страница с вводом логина и пароля. Сверху в центре расположена надпись “Введите логин и пароль”, под ней надписи и поля ввода: надпись черного цвета “Логин” и поле “login” (тип “text”), надпись черного цвета “Пароль” и поле “password” (тип “text”). Под формами кнопка “Отправить”. При ошибочном вводе пользователь увидит сообщение “*message*” “Неверный ввод”. Если пользователь не был зарегистрирован – будет сообщение “*message*” “Пользователь не найден”. Под кнопкой “Отправить” – кнопка

“Вернуться назад”. Под всеми описанными выше элементами располагается ссылка “Вернуться назад” (метод ‘menu_choice’).

2) refuse.html - страница с сообщением о недостаточности прав. В центре сверху страницы расположена надпись “Отказано в доступе, авторизируйтесь”, под надписью ссылка “Вернуться в меню” на ‘start_func’.

5.3. Работа с отчетами

1) Карточка варианта:

- Название: “Работа с отчетами”.
- Предусловие: в БД для каждого типа отчета создана таблица. В БД создана хранимая процедура создания отчета.
- Гарантия: при введении нового отчетного периода, будут создаваться новые записи (только один раз).
- Минимальная гарантия: вторичная попытка создания отчета должна закончиться ошибкой.

2) Сценарий конечного пользователя

- Пользователь начинает работу с отчетами.
- Система присылает страницу с выбором "просмотр" и "создать".
- Пользователь нажимает кнопку "создать".
- Система присылает форму для ввода параметров.
- Пользователь вводит параметры и отправляет данные системе.
 - А) Система присылает сообщение "отчет успешно создан" и дает возможность вернуться к списку отчетов.
 - Б) Система присылает сообщение "отчет успешно создан" и дает возможность ввести другие параметры или вернуться к списку отчетов.
- Пользователь нажимает кнопку "просмотр".
- Система присылает форму для ввода параметров просматриваемого отчета.

- Пользователь вводит параметры и отправляет данные системе.
А) Система присылает сообщение "отчет не создан" и дает возможность ввести другие параметры или вернуться к списку отчетов.
Б) Система присылает динамический шаблон с запрошенным отчетом.

Частный сценарий конечного пользователя:

- Пользователь начинает работу с отчетами.
- Система присылает форму с выбором: «Отчет о найме сотрудников» с соответствующими кнопками для выбора действия «Создать» «Смотреть», «Отчет о регистрации кораблей» с аналогичными кнопками выбора.
- Пользователь нажимает кнопку «Создать» для «Отчета о найме сотрудников».
- Система присылает форму для ввода месяца и года.
- Пользователь вводит данные: 5 месяц и 2019 год и нажимает кнопку отправить.
- Система присылает сообщение «Отчет успешно создан» и дает возможность вернуться к списку отчетов.
- Пользователь нажимает кнопку смотреть.
- Система присылает форму для ввода месяца и года для просмотра соответствующего отчета.
- Пользователь вводит данные: 5 месяц и 2019 год и нажимает кнопку отправить
- Система присылает динамический шаблон с запрошенным отчетом.

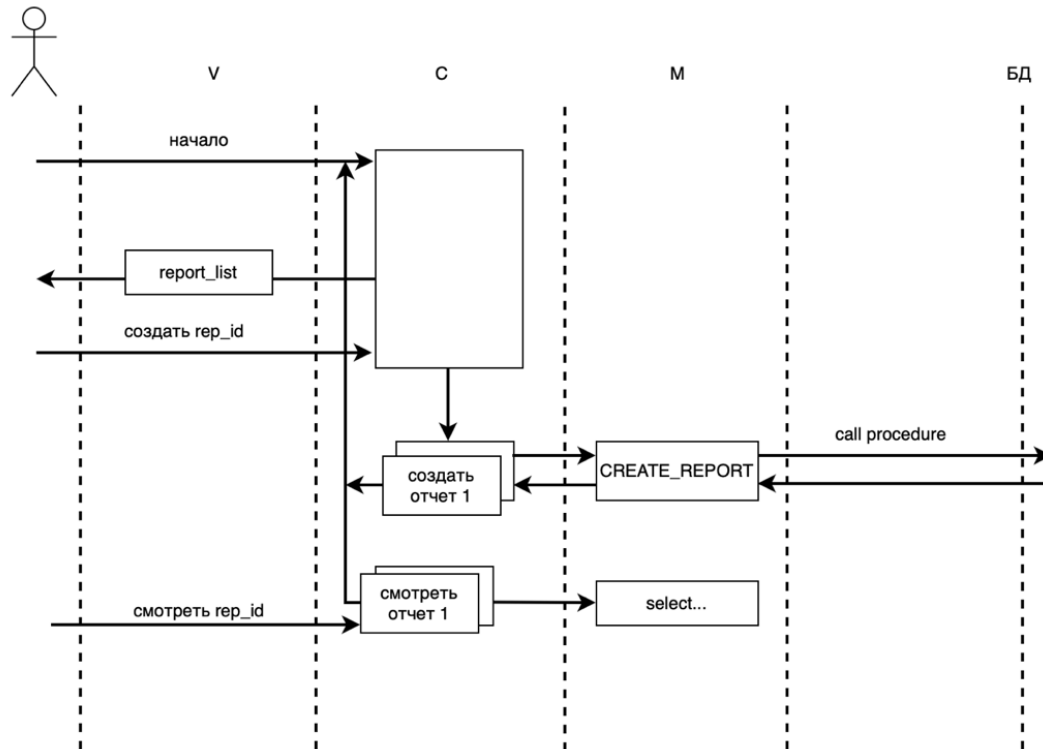
Количество сотрудников	Месяц	Год
8	5	2019

Реализованные отчеты:

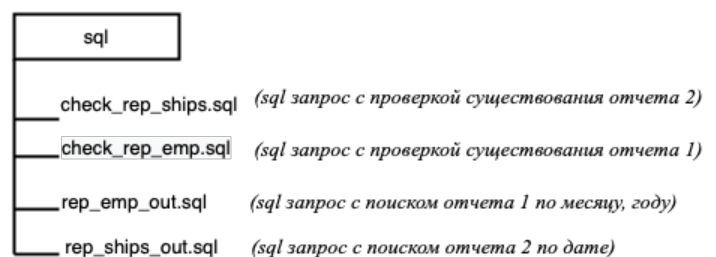
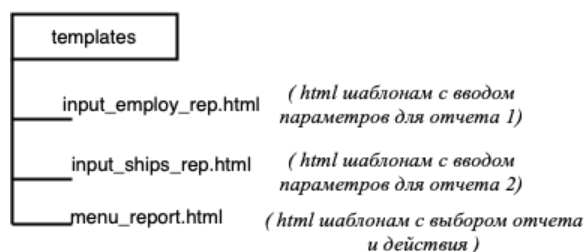
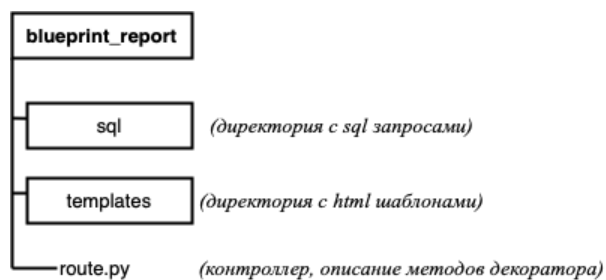
Отчет 1: Показывает количество сотрудников, зарегистрированных в определенный месяц и год

Отчет 2: Показывает количество кораблей, зарегистрированных в порту в определенную дату (дд.мм.гггг)

3) UML системная диаграмма последовательности (работа с отчетами)



4) Файловая структура(blueprint_report):



5) Требования к шаблонам:

1. menu_report.html-

html шаблон для выбора вида отчета и действия к нему. Сверху по центру расположена надпись «Меню отчетов». Входные данные для шаблона – словарь *report_list* содержащий названия отчетов *rep_name* и id отчета *rep_id*. Соответственно, ниже расположены надписи с названиями отчетов и двумя кнопками к каждому виду отчета: кнопка «Смотреть» с именем *view_rep* и кнопка «Создать» с именем *create_rep*. Тут же расположена невидимая кнопка с id отчета для последующего анализа выбора в контроллере. Ниже в центре расположена надпись с сообщением, которое передается шаблону переменной *message*. Ниже расположена ссылка «Вернуться назад» с адресом *menu_choice* для возвращения на предыдущую страницу с вариантами использования .

1. input_emp_rep.html-

html шаблон для ввода параметров отчета о найме сотрудников. Сверху в центре расположена надпись «Введите данные». Далее по центру два поля ввода и две подписи к ним: подпись «Введите месяц» и обязательное поле ввода с именем *month_name* для целых чисел от 1 до 12 , подпись «Введите год» и обязательное поле ввода с именем *year_name* для целых чисел от 1900. Ниже по центру расположена кнопка «Отправить» и вывод текста- сообщений об ошибке, которые передаются шаблону переменными *errone* и *errtwo* . Ниже расположена ссылка «Вернуться назад» с адресом *bp_report.start_report* для возвращения на предыдущую страницу с выбором отчета и действия к нему.

2. input_ships_rep.html-

html шаблон для ввода параметра отчета о регистрации кораблей. Сверху в центре расположена надпись «Введите данные». Далее по центру поле ввода и подпись к нему: подпись «Введите дату» и обязательное поле ввода с именем *date_name* для даты. Ниже по центру расположена кнопка «Отправить» и вывод текста- сообщений об ошибке, которые передаются шаблону переменными *errone* и *errtwo* . Ниже расположена ссылка «Вернуться

назад» с адресом *bp_report.start_report* для возвращения на предыдущую страницу с выбором отчета и действия к нему.

5.4. Основной бизнес-процесс

1) Карточка варианта:

- Название: **“Набор бригады”**.
- Предусловие: в БД создана таблица для записи состава бригады.
- Гарантия: под состав собранной для разгрузки бригады создадутся записи в соответствующей таблице.
- Минимальная гарантия: нет возможности сформировать бригаду на корабль, для которого она уже сформирована.

2) Сценарий конечного пользователя

- Пользователь начинает работу с бизнес-процессом «набор бригады».
- Система присылает страницу с выбором корабля для разгрузки с необходимой информацией по каждому.
- Пользователь нажимает кнопку "набрать бригаду" для конкретного корабля.
- Система присылает форму для ввода параметра – даты для разгрузки корабля собранной бригадой.
- Пользователь вводит дату и отправляет данные системе:
 - А) Система присылает сообщение о неверной дате и дает возможность ввести другие параметры или вернуться к выбору сотрудников для бригады.
 - Б) Система присылает страницу со списком сотрудников для выбора.
- Пользователь нажимает кнопку «добавить» у конкретного сотрудника:

А) Система присылает ту же страницу с добавленным в «корзину» сотрудником.

Б) Система присылает ту же страницу с сообщением - «сотрудник уже выбран» в случае, если сотрудник был выбран ранее в «корзину».

- Пользователь нажимает кнопку "сохранить набор бригады".

А) Система присылает страницу с сообщением о том, что бригада на разгрузку собрана и сохранена.

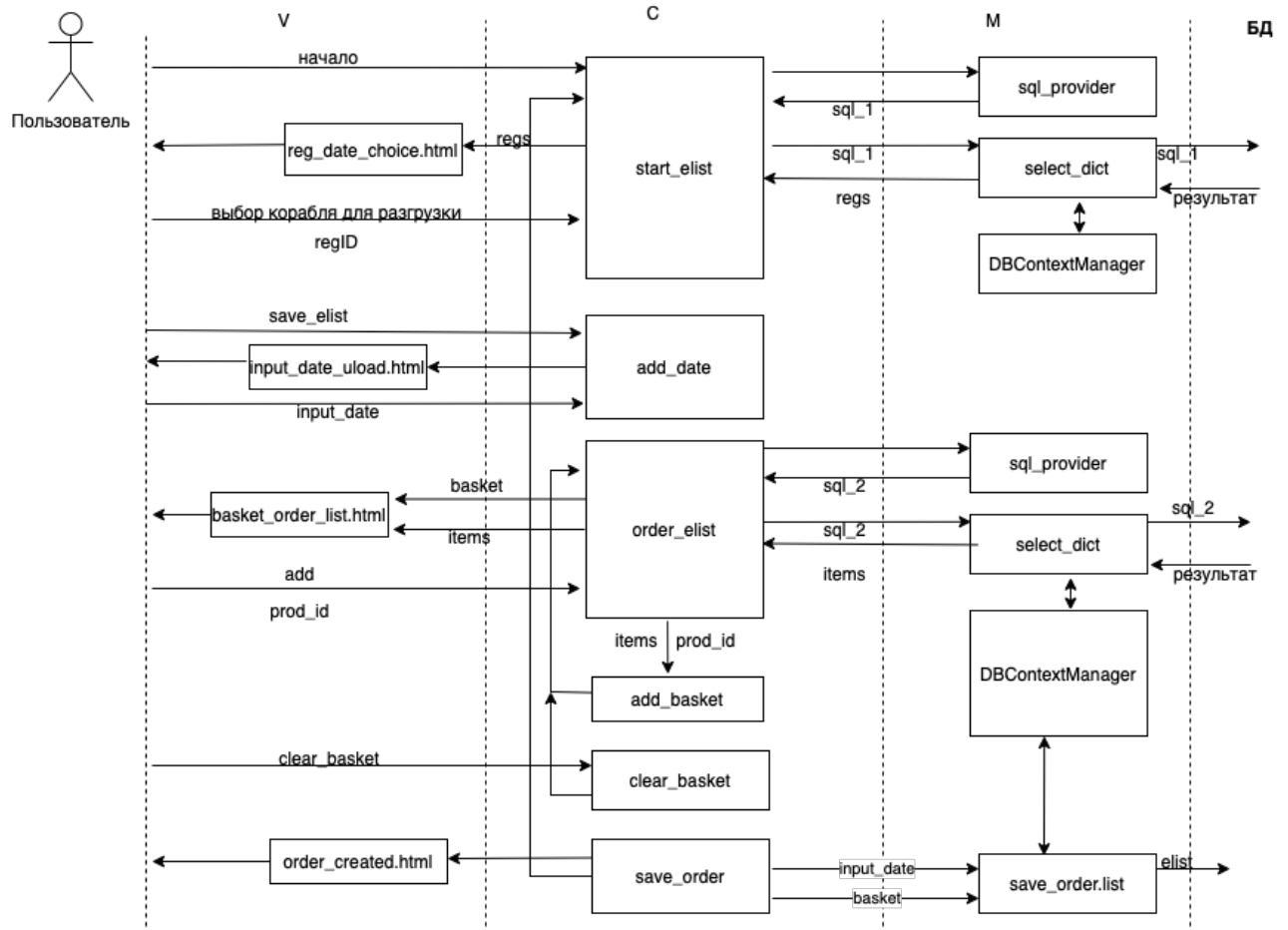
Б) Система присылает страницу с сообщением о том, что произошла ошибка сохранения.

Частный сценарий конечного пользователя:

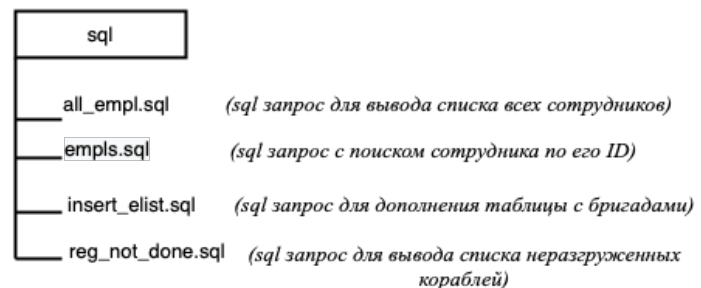
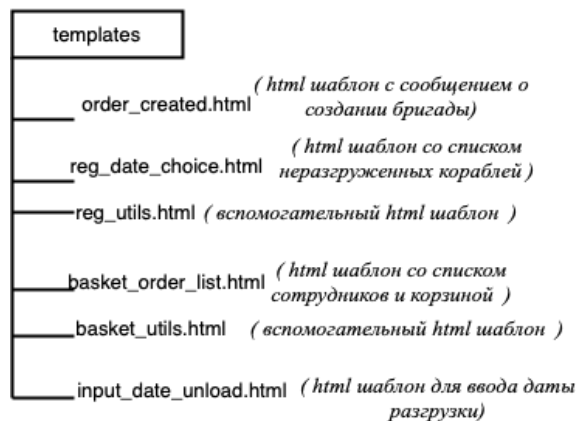
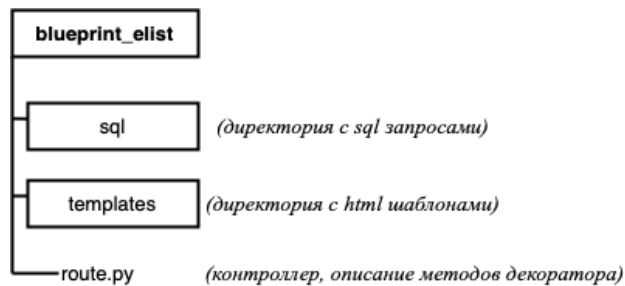
- Пользователь начинает работу с бизнес-процессом «набор бригады».
- Система присылает страницу с выбором корабля для разгрузки с необходимой информацией.
ID регистрации: 21 Дата регистрации: 2020-03-22 Название корабля: Эвелина ID номер причала: 10
- Пользователь выбирает корабль «Эвелина» и нажимает кнопку "набрать бригаду".
- Система присылает форму для ввода параметра – даты для разгрузки корабля собранной бригадой.
- Пользователь вводит дату 02.10.2012 и отправляет данные системе:
- Система присылает сообщение «неверная дата» и дает возможность ввести другие параметры или вернуться к выбору сотрудников для бригады.
- Пользователь вводит дату 20.12.2022 и отправляет данные системе.
- Система присылает страницу со списком сотрудников для выбора.
- Пользователь нажимает кнопку «добавить» у сотрудника «Сидоров»:
- Система присылает ту же страницу с добавленным в «корзину» сотрудником «Сидоров».
- Пользователь нажимает кнопку "сохранить набор бригады".

- Система присылает страницу с сообщением «**Бригада на разгрузку корабля с регистрацией №21 успешно создана**» - успешное формирование бригады на разгрузку.

3) UML системная диаграмма последовательности (набор бригады)



4)Файловая структура(blueprint_elist):



5)Требования к шаблонам:

5.1)order_created.html – статическая html страница с сообщением о том, что «Бригада на разгрузку корабля с регистрацией № *regID* успешно создана», где переменная *regID* передается шаблону. Ниже по центру располагается ссылка на меню выбора корабля для разгрузки ('bp_elist.start_elist').

5.2)input_date_unload.html – html страница для ввода даты разгрузки. Сверху по центру располагается надпись «Введите данные», далее подпись «Введите дату» и поле ввода 'date_name' типа date. Ниже по центру расположена кнопка «Отправить» и сообщение *errone* для отображения ошибок ввода. Ниже приведена ссылка на предыдущую страницу('bp_elist.start_elist').

5.3)reg_date_choice.html – динамический шаблон отображающий список доступных для разгрузки кораблей. Html шаблону в качестве параметра передается список словарей *regs*, где каждый словарь содержит информацию

о корабле. Сверху по центру на странице располагается ссылка на предыдущую страницу ('menu_choice'). Далее надпись «Все неразгруженные корабли» и под ней рамки фиксированного размера с информацией о кораблях, если список regs не пустой, иначе надпись «Все корабли разгружены». Информация по каждому кораблю содержит пункты: regID – id номер регистрации корабля, regCome – дата регистрации, sName – название корабля, pID – id номер причала. Внутри каждой рамки должна быть кнопка «Набрать бригаду», хранящая значение regID.

5.4)basket_order_list.html – динамический html шаблон отображающий список сотрудников доступных для разгрузки и список сотрудников выбранных пользователем. Html шаблону в качестве параметра передается список словарей items, где каждый словарь содержит информацию о сотруднике: id номер сотрудника eID, профессия eProf, фамилия eSurname.

Также шаблону передается список словарей basket_items, где каждый словарь содержит в себе информацию подобно items о каждом выбранном пользователем сотруднике. Сверху располагается надпись «Набор бригады на разгрузку регистрацией номер regID», где regID передается шаблону. Далее по центру расположены ссылки на обработчики: «Сохранить набор бригады» ('bp_elist.save_order'), «Вернуться в меню пользователя»('bp_elist.start_elist'). Под надписью «Все сотрудники» (расположена сбоку справа на странице) располагаются рамки фиксированного размера, содержащие информацию о сотрудниках. Слева расположена надпись «Ваш выбор сотрудников» с оформлением списка сотрудников подобно списку всех сотрудников. Если сотрудник выбирает дважды отображается сообщение message «Сотрудник уже выбран».

5.5 Расписание

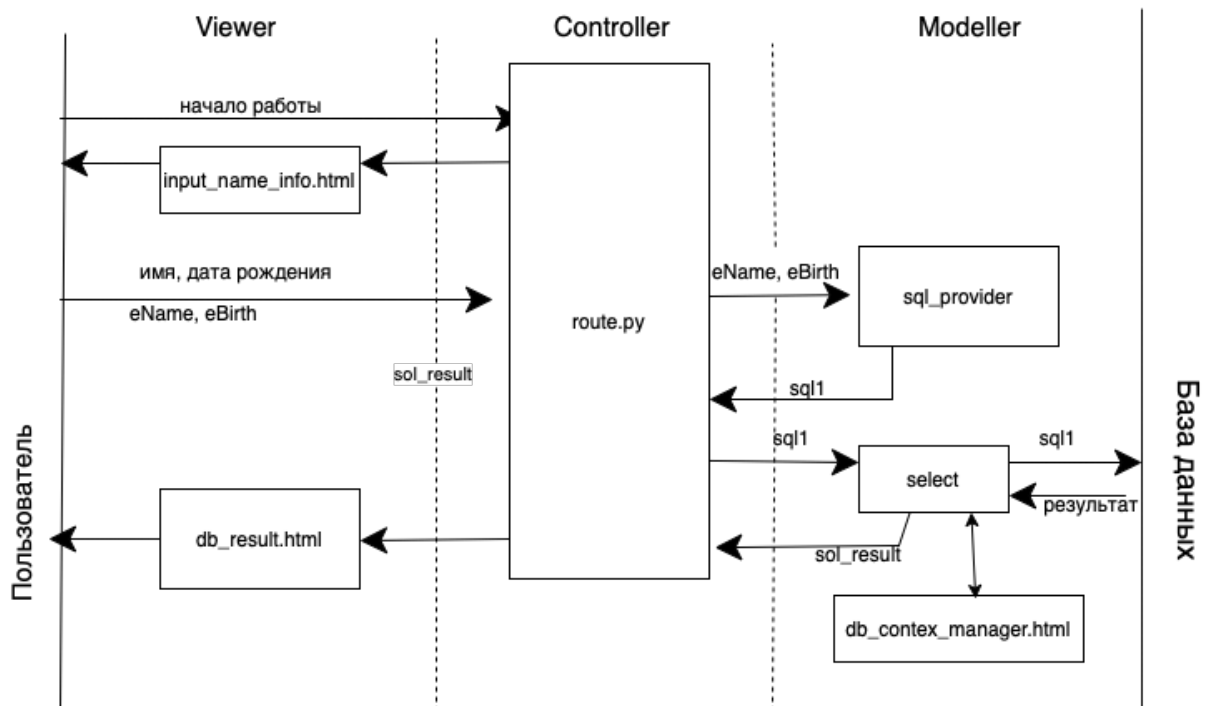
1) Карточка варианта:

- Название: “**Расписание**”.
- Предусловие: авторизован внешний пользователь .
- Гарантия: пользователь получит расписание разгрузок.
- Минимальная гарантия: пользователь получит сообщение об ошибке, БД осталась в согласованном состоянии

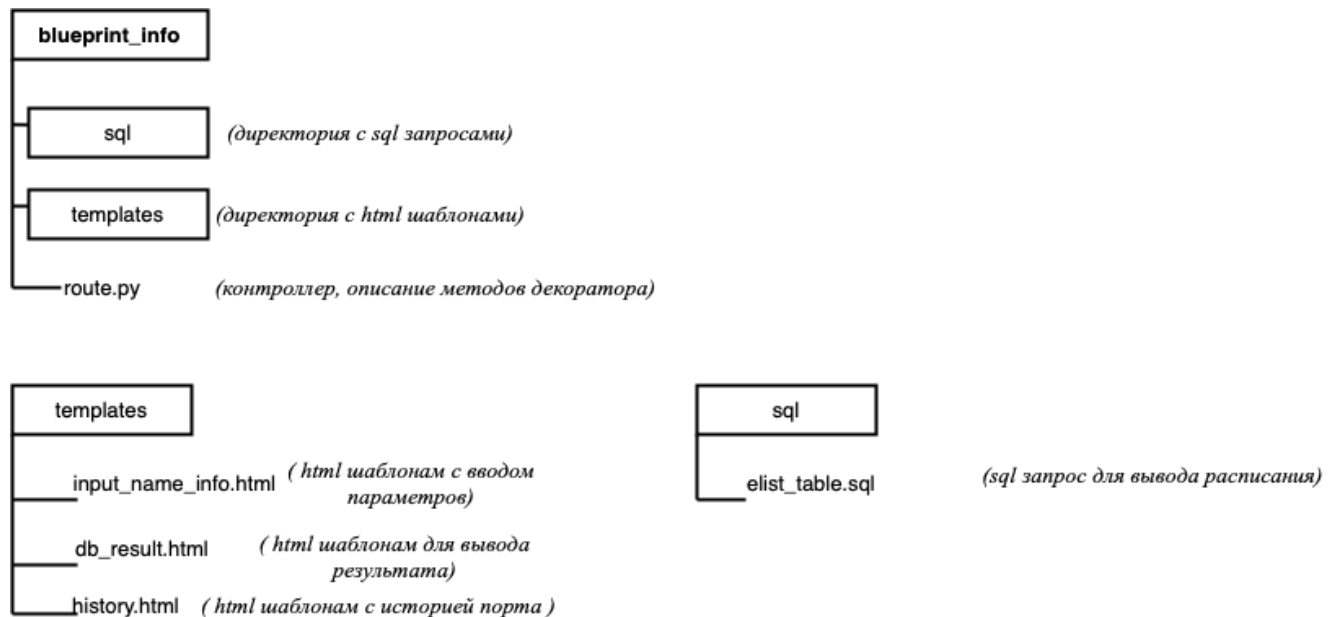
2) Сценарий конечного пользователя

- Пользователь начинает работу с расписанием.
- Система присылает страницу с вводом параметров.
- Пользователь вводит параметры
 - А) При неверном вводе отображается сообщение об ошибке
 - Б) Система выполняет запрос и присылает пользователю расписание разгрузок.

3) UML системная диаграмма последовательности (расписание)



4)Файловая структура(blueprint_elist):



5)Требования к шаблонам:

5.1) db_result - шаблон с результатами select-запроса `db_result.html`. Сверху в центре расположена надпись “Описание”. Под ней таблица с результатами запроса. Шаблону в качестве параметров передаются списки - `schema`, `result`, хранящие название столбцов и содержимое таблицы соответственно. Информация в таблице должна быть отображена шрифтом черного цвета, задний фон в таблице – белого цвета.

5.2)input_name_info – шаблон с вводом параметров для отображения расписания. Надпись «Введите фамилию» и поле `input_surname`, надпись «Введите дату рождения» и поле `input_date` типа `date`. Под всеми описанными выше элементами должна быть ссылка “Вернуться назад” (метод `bp_info.start_info`).

6.Глобальная архитектура приложения

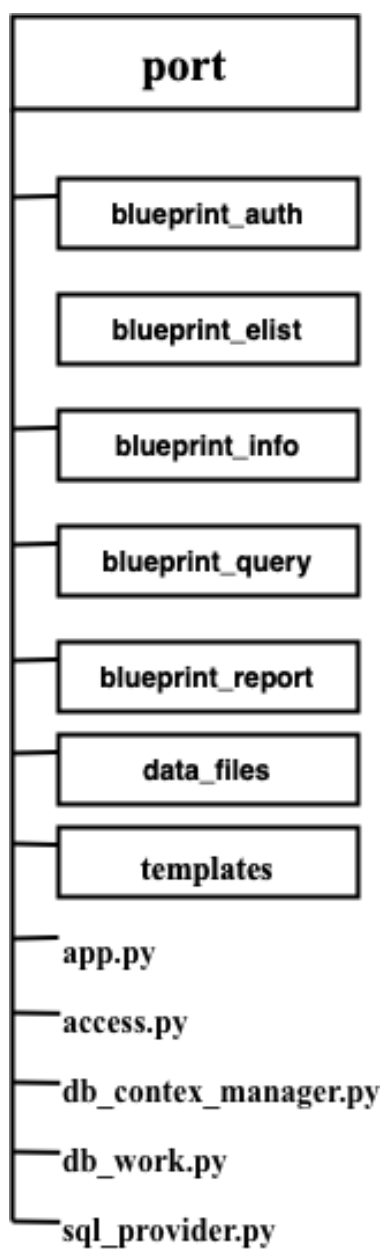


Рис.6 Глобальная архитектура приложения

7.Инфологическая модель предметной области в форме UML- диаграммы классов

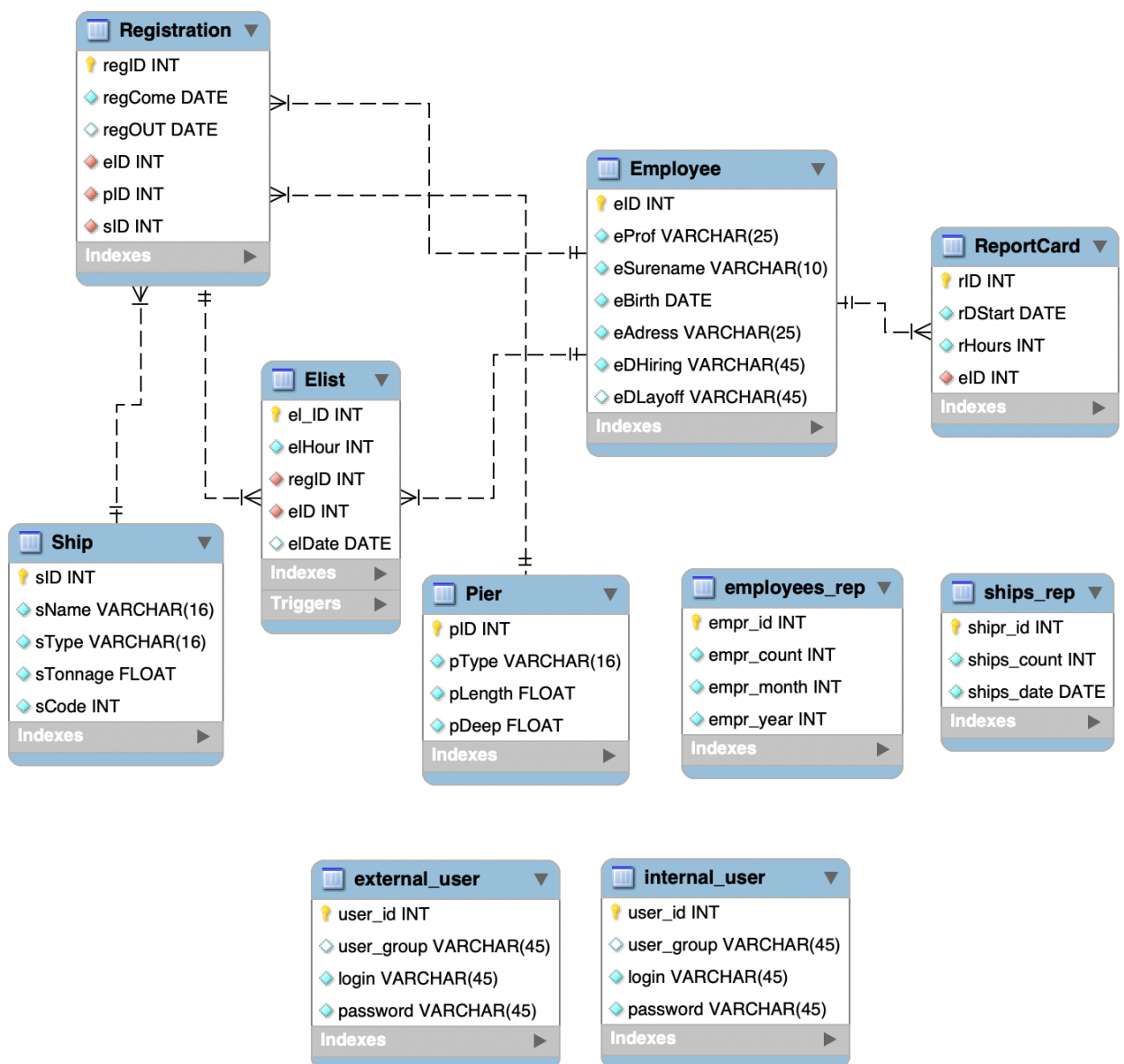


Рис.7 Инфологическая модель предметной области в форме UML-диаграммы классов

8. Логическая модель БД

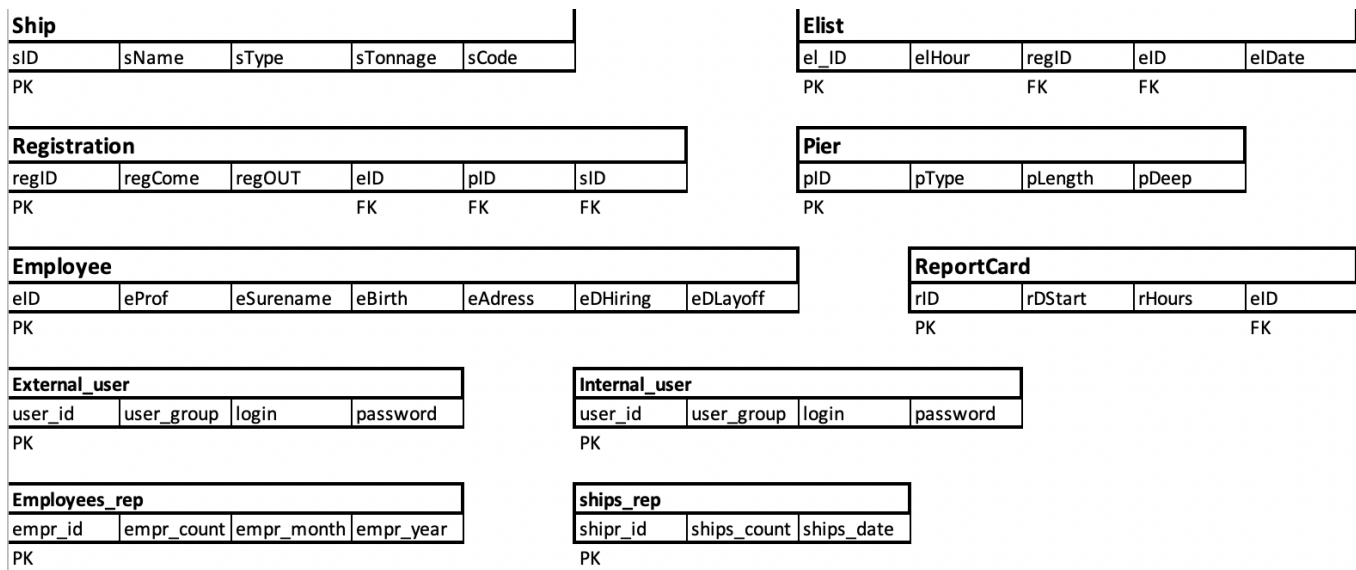


Рис.8 Логическая модель Базы данных

Заключение

В данной работе описана реализации информационной системы для порта. Реализация выполнялась поэтапно в соответствии с пунктами технического задания. Задача реализации информационной системы была выполнена с использованием языка программирования Python в среде фреймворка Flask.

Список использованных источников

- 1) Лекции по курсу «Разработка информационных систем» и «Базы данных»