



Министерство науки и высшего образования 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 *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

## **РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА** **К КУРСОВОЙ РАБОТЕ**

по дисциплине: «Базы данных»

Студент

Борисов Сергей Дмитриевич

Группа

РК6-51Б

Тема курсовой работы

Разработка информационной системы  
«График Кораблей (ShipChart)»

Студент

\_\_\_\_\_  
подпись, дата

**Борисов С. Д.**

\_\_\_\_\_  
Фамилия, И.О.

Преподаватель

\_\_\_\_\_  
подпись, дата

**Пивоварова Н.В.**

\_\_\_\_\_  
Фамилия, И.О.

*Москва, 2021 г.*

## **Аннотация.**

Для контроля работ, осуществляемых в порту, контроля загрузок и разгрузок кораблей необходимо разработать информационную систему. Система должна предоставлять наглядное отображение данных о текущем состоянии дел в порту, с возможностью добавления, редактирования, удаления объектов, а так же возможностью просмотра исторических данных.

## **Оглавление**

Задание. Описание предметной области.....	2
Определение конечных пользователей.....	2
Разработка UML диаграммы вариантов использования.....	3
Описание вариантов использования.....	3
Главное меню.....	3
Авторизация и аутентификация.....	4
Основной бизнес-процесс.....	6
Внесение кораблей в базу.....	6
Работа с сотрудниками.....	7
Работа с разгрузками.....	8
Работа со списком работ.....	8
Логическая модель базы данных.....	9
VRMN схема меню запросов и выполнения запросов.....	10
VRMN схема авторизации.....	11
Сценарий.....	11
Файловая структура проекта.....	13

## **Задание. Описание предметной области.**

Требуется разработать информационную систему (далее будем пользоваться сокращением ИС) для сотрудников порта, с помощью которой можно будет отслеживать активность в порту.

Основной бизнес-процесс выглядит следующим образом:

После захода корабля в порт начинается его разгрузка. На разгрузку назначается некоторое количество сотрудников порта.

Данные об участии сотрудников порта в разгрузке кораблей сохраняются отдельно, а затем переносятся в общий график работы сотрудников. Общий график впоследствии используется для начисления заработной платы сотрудникам.

## **Определение конечных пользователей.**

Конечными пользователями ИС являются работники порта, имеющие доступ к заполнению данных в ИС, а так же администраторы, имеющие возможность регистрировать новых сотрудников и выдавать им данные для авторизации.

## Разработка UML диаграммы вариантов использования.

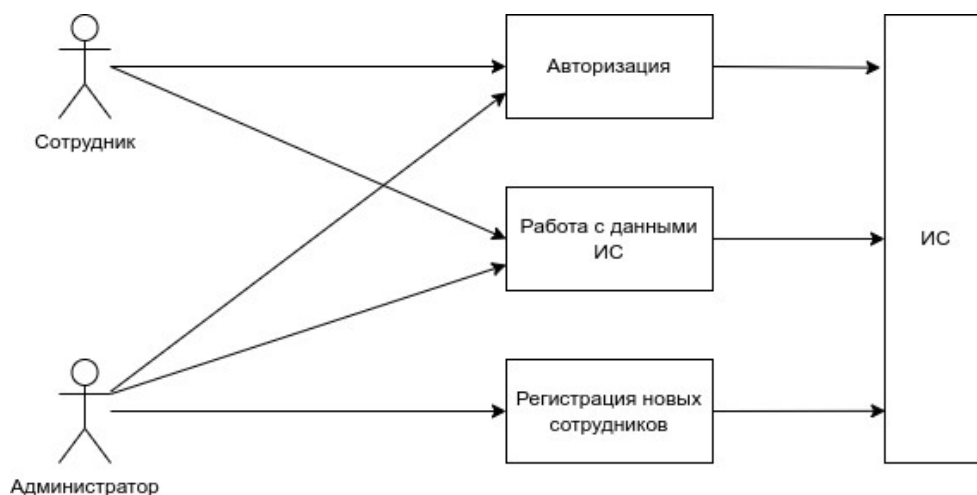


Рис. 1. Диаграмма вариантов использования ИС.

## Описание вариантов использования.

В данном разделе описаны варианты взаимодействия с ИС. Приведены BPMN-диаграммы соответствующих сценариев, описание самих сценариев и даны комментарии по программной реализации. Также, где это необходимо, приведены результаты UI тестов, то есть результаты выполнения сценариев на основе тестовых данных.

### Главное меню.

Главное меню – это первое, что видит пользователь при входе в ИС. Оно должно содержать достаточное количество справочной информации для нового пользователя, а также элементы навигации (ссылки) для перехода к остальным разделам системы. При попадании нового пользователя на страницу должна появляться стартовая страница, предлагающая выбрать подсистему, которой необходимо воспользоваться. В данный момент присутствует только одна подсистема «панель администрирования», которой можно воспользоваться, пройдя авторизацию.

В любой момент авторизованный пользователь может выйти из учетной записи и очистить данные сессии, нажав на элемент навигации «выйти», при этом совершается переход на главную страницу. Чтобы снова воспользоваться системой, потребуется вновь авторизоваться.

Ссылки на остальные разделы системы включают в себя раздел управления данными кораблей, раздел управления данными сотрудников порта, раздел управления разгрузками, раздел управления списком работ, производимых в порту.

На рис. 2 приведена BPMN-диаграмма главного меню. Отметим пункт «авторизация», который описывается в следующем разделе.

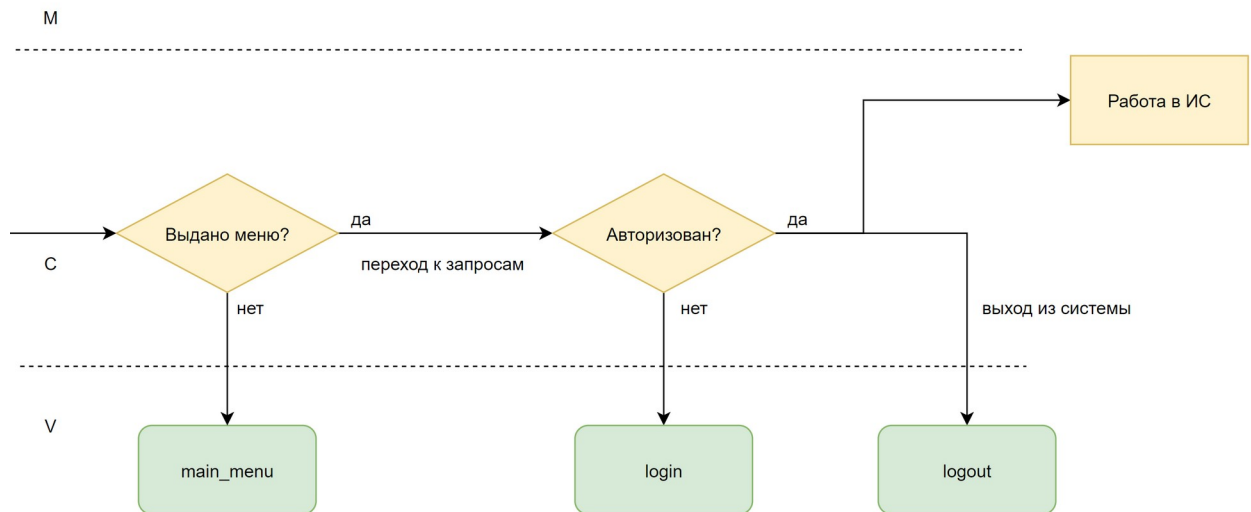


Рис. 2 BPMN-диаграмма главного меню.

Описание сценария:

1. Пользователь начинает сценарий
2. Если не выдано главное меню, происходит выдача меню
3. При попытке перейти к работе происходит идентификация.
4. Если идентификатор не был найден в сессии, производится переход на страницу входа в систему.
5. После идентификации и аутентификации пользователь может переходить к остальным элементам ИС.
6. При выборе элемента «выйти» система перенаправляет пользователя на соответствующую страницу и очищает сессию.

## Авторизация и аутентификация.

В разрабатываемой ИС требуется предусмотреть возможность ограничения доступа для внешних пользователей путем выполнения идентификации и аутентификации, а также разделения доступа между элементами ИС, т. е. авторизации (так, например, у сотрудника не должно быть прав на создание аккаунтов). Для этого был разработан сценарий, при котором система предлагает пользователю войти в систему, а также набор декораторов для выполнения идентификации и авторизации. Их можно применять к функциям-обработчикам запросов, чтобы иметь возможность не допускать пользователя на определенные страницы. Политики для приложения хранятся в файле конфигурации и однократно загружаются, и сохраняются в приложении при его запуске. Данные для идентификации остаются на стороне клиента в cookie файлах браузера. Они содержат такие поля, как идентификатор пользователя, его имя и группа, к которой принадлежит пользователь.

BPMN-диаграмма приведена на рис. 3, описание сценария приведено ниже.

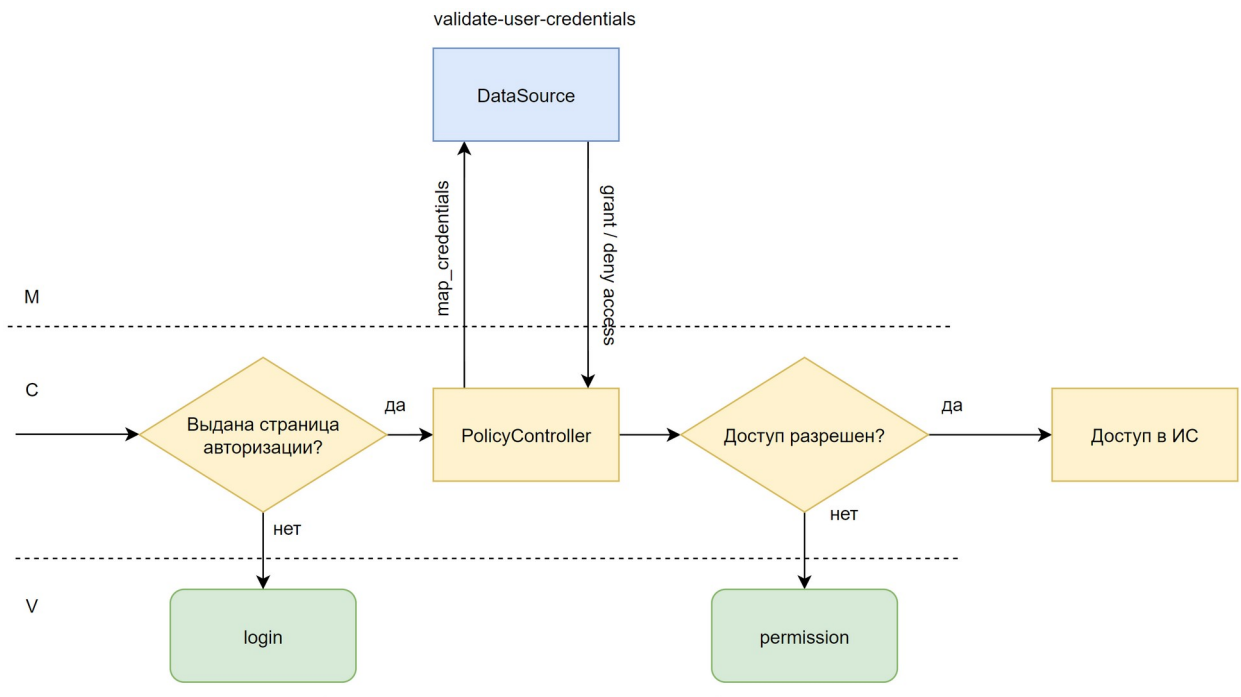


Рис. 3. BPMN-диаграмма аутентификации.

Описание сценария:

1. Пользователь начинает сценарий (произведен выбор опции «войти» в меню, либо было произведено перенаправление с другой страницы).
2. Выдается форма для ввода данных внутренней учетной записи.
3. Пользователь вводит логин и пароль.
4. Если введенные данные корректны, система помещает метаданные о пользователе (идентификатор, имя и данные о группе) в сессию, затем перенаправляет пользователя в главное меню.
5. Если введенные данные некорректны, система уведомляет об этом пользователя, выводя предупреждающее сообщение. Пользователь может ввести данные повторно.

Понятие декоратора, пришедшее в *python* из функционального программирования, позволяет писать экспрессивный код путем «оборачивания» одной функции в другую. Именно с их помощью (а именно, с помощью параметризованных декораторов) производится регистрация функций-обработчиков в приложении во фреймворке *Flask*. Логическим развитием идеи их применения является т. н. *decorator chaining*, т. е. последовательное применение нескольких декораторов к одной функции. При этом большое значение имеет последовательность их применения. Так, на «внешнем» уровне к функции-обработчику должен быть применен *app.route('...')*, далее декоратор прав доступа. Поэтому первым «перехватывает управление» веб-сервер и формирует контекст запроса, далее сначала производится авторизация.

```

def user_group_requires(u_group: int):
    def user_group_requires_f(f):
        @wraps(f)
        def wrapper(*args, **kwargs):
            if session.get('user_id') is None:
                return redirect(url_for('auth.index') + '?redirect_url=' + request.url)
            if int(session.get('u_group')) & u_group or int(session.get('u_group')) & \
                int(getattr(current_app, 'auth_groups')['admin']):
                return f(*args, **kwargs)
            return redirect(url_for('auth.forbidden'))

        return wrapper

    return user_group_requires_f

```

Листинг 4. Исходный код функции-декоратора, реализующие аутентификацию.

## Основной бизнес-процесс.

Данная система подразумевает ведение записей о разгрузках кораблей и табелях работ сотрудников порта. Можно выделить 4 основных объекта, над которыми производятся операции: корабль, сотрудник, разгрузка, работа. Для работы со списками этих объектов реализуются страницы, по одной на каждый список: страница занесения в базу кораблей, приходящих в порт, страница управления сотрудниками порта, страница ведения записей о разгрузках, страница со списком работ сотрудников. Для каждого раздела были созданы отдельные шаблоны приложений (blueprints), каждый из которых реализует сетевой API необходимый для работы с данными, а так же контроллеры отображения страниц. Для каждого объекта реализуется полный набор CRUD операций. Для запросов с типом GET возвращается собранная шаблонизатором веб-страница, остальные запросы подразумевают работу с ресурсом. В данном случае для простоты это работает именно так, однако в общем случае можно было бы отделить сетевой API от контроллера страниц, включая задание других URL, или реализовать выдачу HTML либо чистых данных об объектах в формате JSON в зависимости от заголовка Асепт передаваемого с клиента.

## Внесение кораблей в базу.

Данный раздел предназначен для сотрудников или администраторов порта, имеющих данные аккаунтов для доступа к системе.

В ИС создан отдельный «шаблон приложения», *blueprint*, содержащий необходимые функции-обработчики.

Данный раздел отвечает за внесение кораблей пришедших в порт в базу, с целью их дальнейшего использования в других разделах ИС.

Главной задачей этого раздела является реализация операций CRUD для объектов кораблей.

BPMN-диаграмма сценариев приведена на рис. 5.

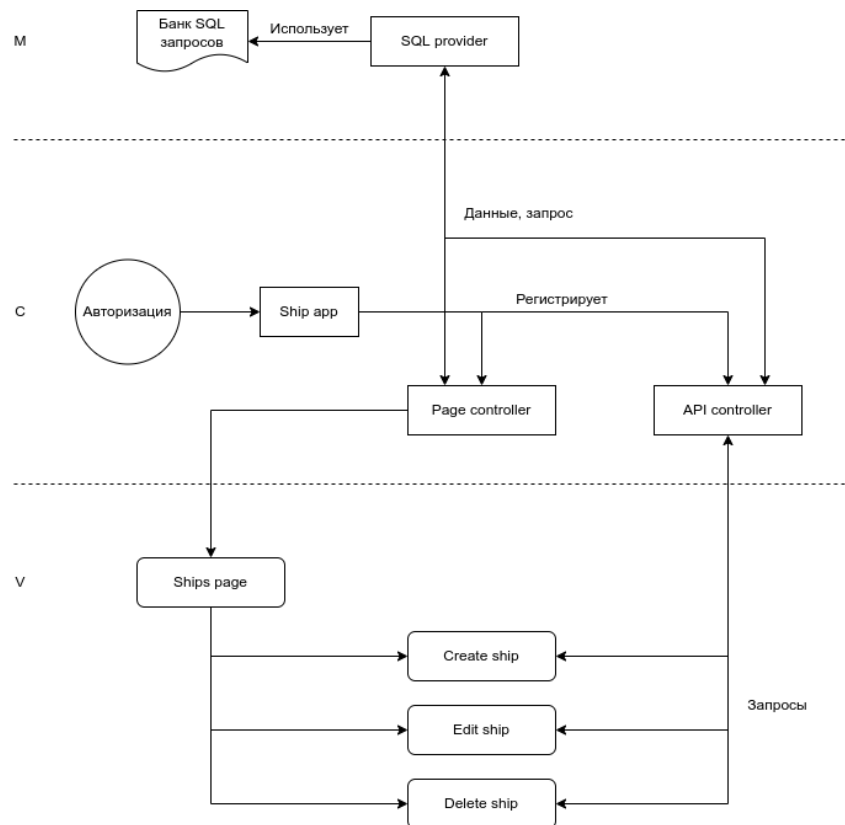


Рис. 5. BPMN-диаграмма управления кораблями

Описание сценария:

1. Сотрудник начинает сценарий (принимается, что авторизация уже была произведена).
2. Система выдает пользователю страницу раздела.
3. Пользователь выбирает опцию меню.
4. Система получает запрос, определяет операции по изменению базы данных (если требуется).
5. Проверенные данные передаются в модель в качестве параметров SQL-запроса.
6. Над объектом данных выполняются определенные операции.
7. На клиентскую страницу возвращается результат запроса.
8. Страница перезагружается для отображения изменений (переход к пункту 1).

## Работа с сотрудниками.

Данный раздел предназначен для добавления сотрудников в базу, работает как предыдущий сценарий, за исключением того, что добавляется возможность уволить сотрудника без удаления из базы, с помощью назначения ему даты увольнения, а так же возможность восстановить уволенного сотрудника в должности.

BPMN-диаграмма данного раздела расположена на рис. 6.

Описание сценария:

1. Сотрудник начинает сценарий (принимается, что авторизация уже была произведена).
2. Система выдает пользователю страницу раздела.
3. Пользователь выбирает опцию меню.
4. Система получает запрос, определяет операции по изменению базы данных (если требуется).
5. Проверенные данные передаются в модель в качестве параметров SQL-запроса.

6. Над объектом данных выполняются определенные операции.
7. На клиентскую страницу возвращается результат запроса.
8. Страница перезагружается для отображения изменений (переход к пункту 1).

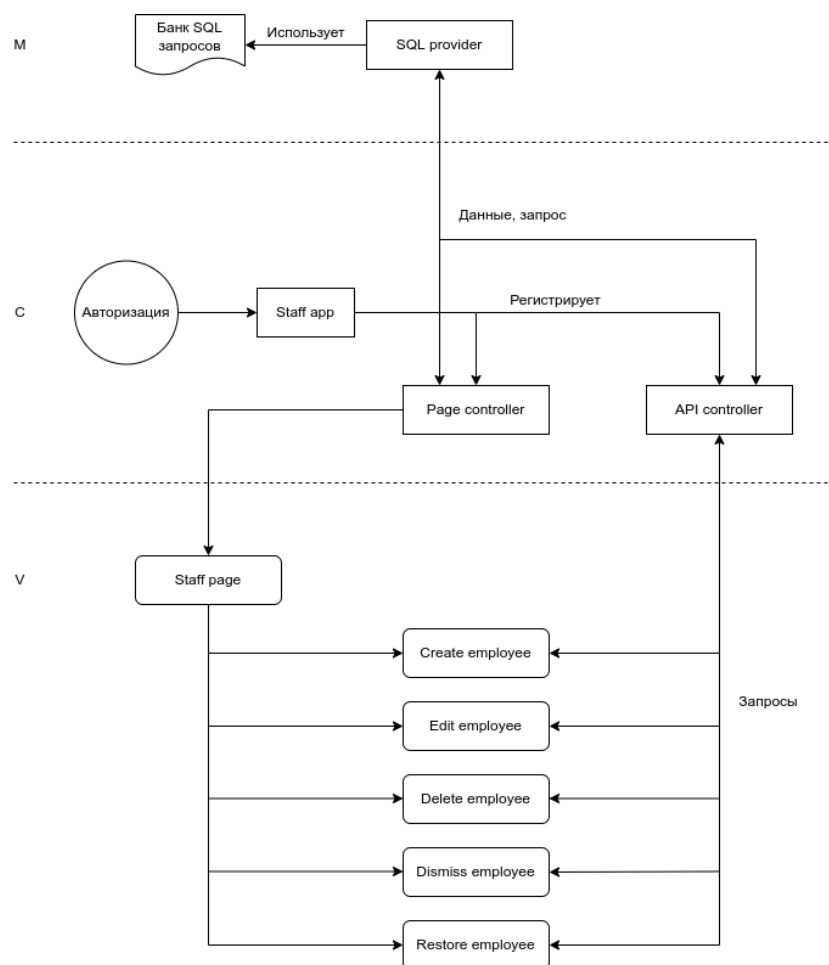


Рис. 5.2.1. BPMN-диаграмма раздела сотрудников.

## Работа с разгрузками.

Данный раздел по схеме работы аналогичен предыдущим двум, за тем исключением, что после загрузки основной страницы асинхронно подгружаются данные с сервера посредством API без явного действия пользователя (загружается список кораблей и сотрудников). Этот раздел предоставляет следующий функционал: CRUD для разгрузок, перенос данных о работах сотрудников в таблицу работ для выделенных разгрузок, назначение даты завершения для разгрузок.

## Работа со списком работ.

Данный раздел так аналогичен предыдущему разделу с точки зрения сценария работы, и предоставляет следующий функционал: CRUD для работ, возможность пометить оплаченные работы.



## Логическая модель базы данных.

Для работы ИС необходима база данных (БД), содержащая таблицы, хранящие информацию о кораблях, сотрудниках, разгрузках и данные о их работе, а также таблица для аутентификации, хранящая логины, пароли пользователей.

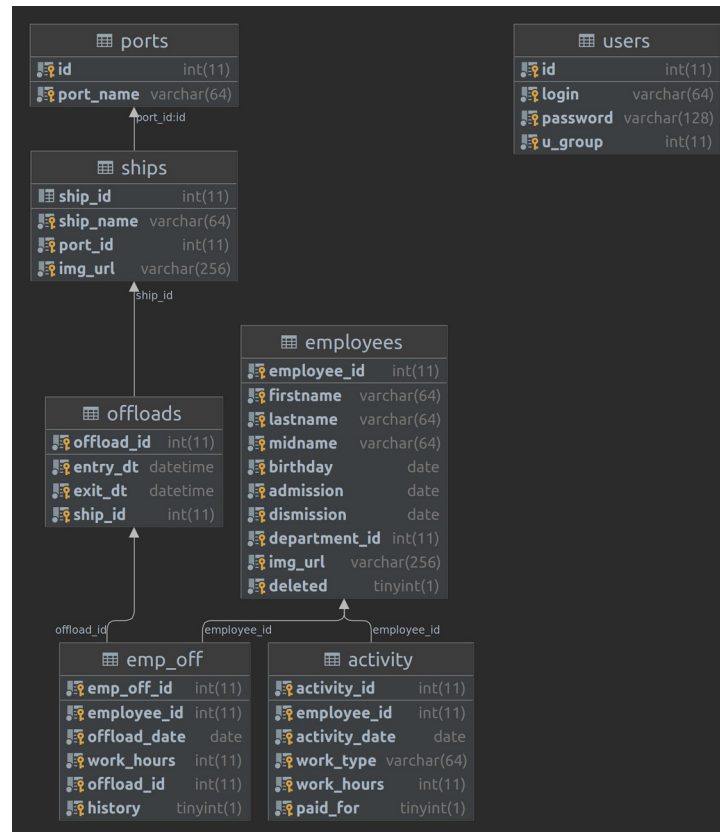
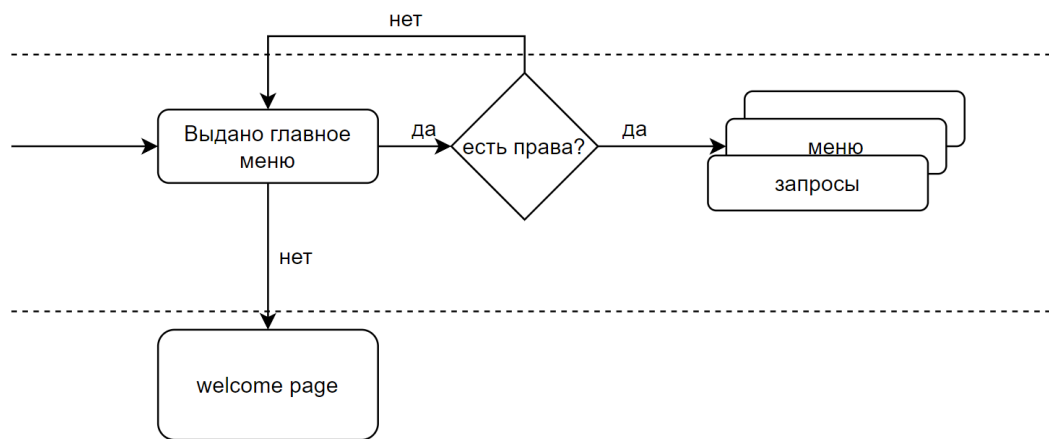
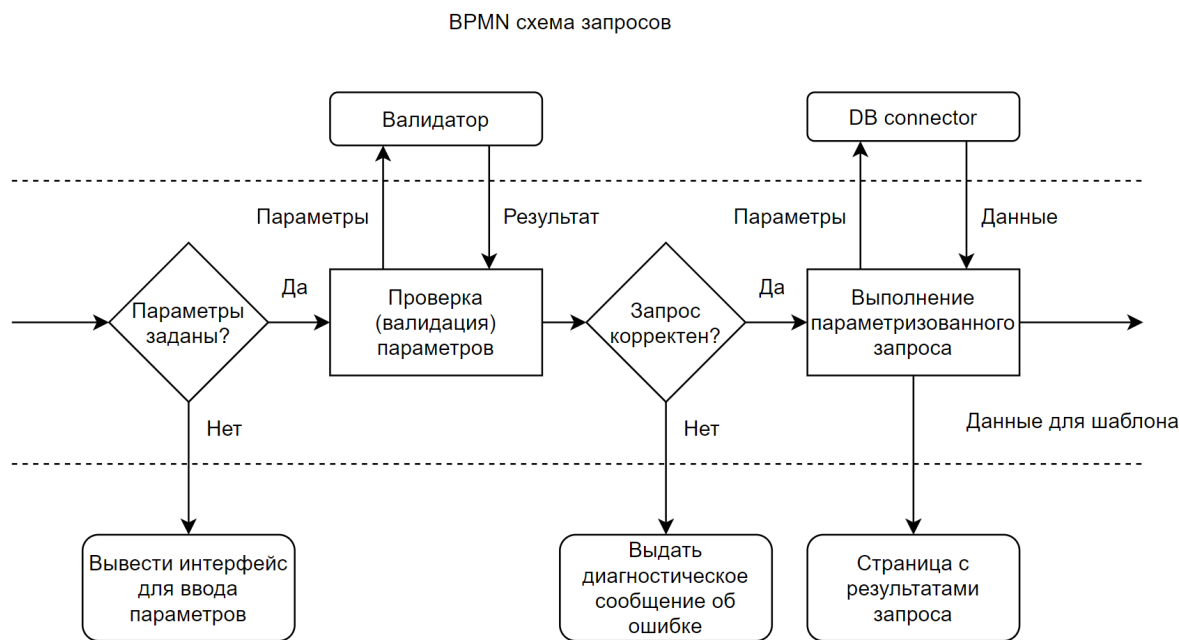
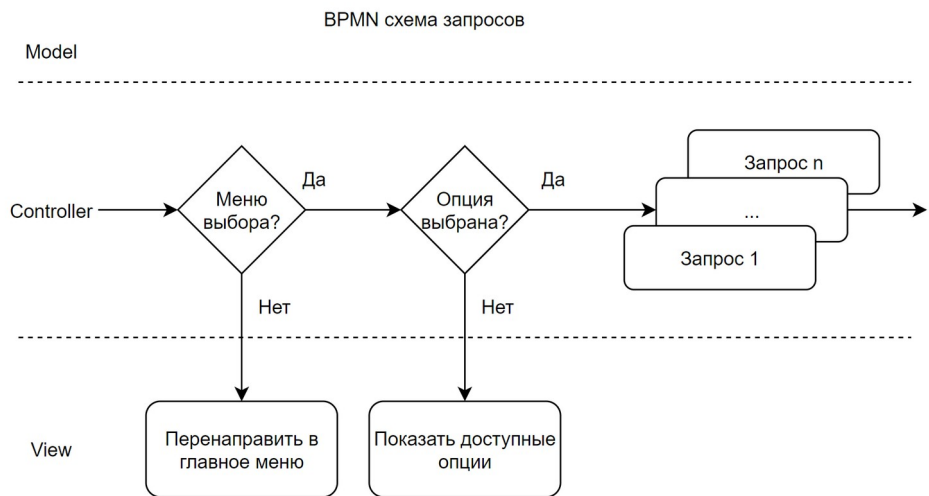


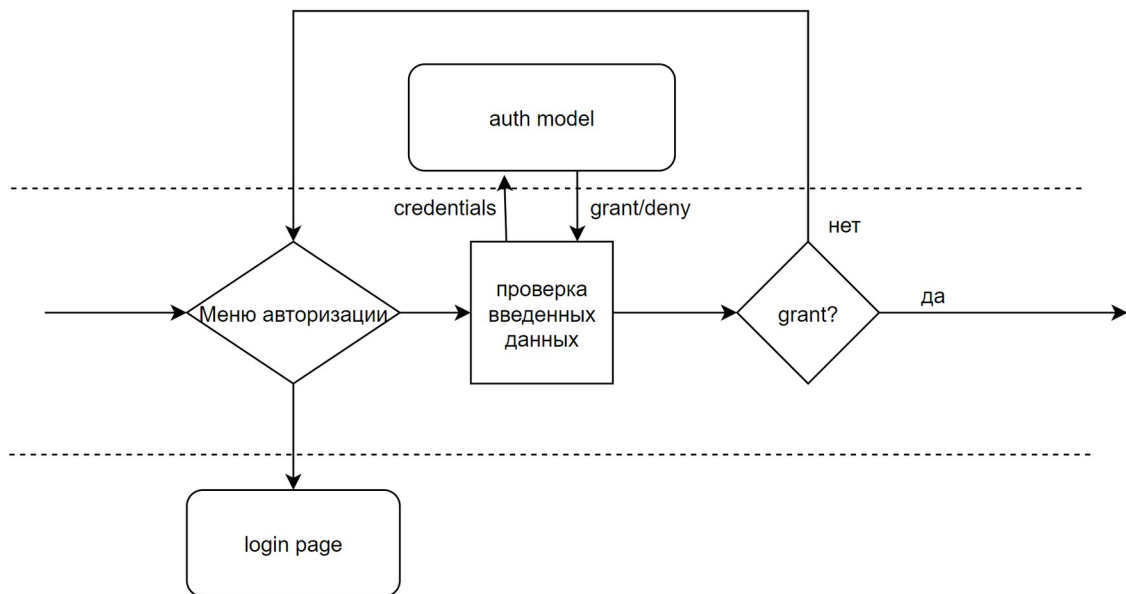
Рис. 6.1. Логическая модель БД.

Таблица **ports** хранит список портов, таблица **ships** хранит список кораблей, имеет внешний ключ, ссылающийся на таблицу портов (one-to-one), таблица **offloads** хранит список разгрузок, имеет внешний ключ, ссылающийся на таблицу с кораблями (one-to-one), участвующими в разгрузках, таблица **emp\_off** (employee offloads) хранит список сотрудников, участвовавших в разгрузках, имеет внешние ключи, ссылающиеся на таблицу разгрузок (many-to-one) и таблицу сотрудников (one-to-one), таблица **employees** хранит список сотрудников, таблица **activity** хранит список активностей сотрудников (табель работ), имеет внешний ключ, ссылающийся на таблицу сотрудников (one-to-one), а так же таблица **users**, которая хранит список зарегистрированных аккаунтов и групп для каждого аккаунта, которые хранятся как целое число, которое с применением побитовой маски позволяет определить, к каким группам принадлежит этот аккаунт (предполагается строго определенное число групп).

**BPMN схема меню запросов и выполнения запросов**



## BPMN схема авторизации



## Сценарий

1. Пользователь начинает сценарий
2. Выдаем форму для ввода пароля и логина (авторизации)
  - a. (проверяем через контроллер прав, делаем запрос в системную таблицу с ролями)
  - b. пропускаем дальше, иначе выводим информацию о неверно введенных параметрах учетной записи
  - c. 4. в случае успеха система запоминает пользователя путем добавления данных о группе в сессию
  - d. сессия хранится на стороне клиента в виде куки до завершения сессии браузера (закрытия приложения)
  - e. в случае инкогнито, иначе остается в локальном хранилище (не имеет expiration date?)
3. Система предоставляет пользователю доступ к меню:

- a. Если выбран элемент меню, переход в соответствующий элемент меню
  - b. Если выбран выход из системы, производится выход из системы
  - c. Если элемент меню не выбран, ничего не происходит
4. Отображение меню запросов:
- a. При попадании в меню запросов пользователь может выбрать интересующий его запрос
  - b. Если выбран элемент меню, переход на страницу выбора параметров
  - c. Если выбрана опция возврата в главное меню, вернуться в главное меню
5. Выполнение запросов:
- a. Пользователю предоставляется возможность ввести параметры запроса
  - b. Если параметры введены, производится их передача в элемент валидации. Если параметры некорректные, выводится диагностическое сообщение об ошибке
  - c. Если параметры корректны, они передаются в слой бизнес-логики, который обрабатывает данный запрос и выводит результат с перенаправлением на соответствующую страницу или происходит возврат данных (если обращение было к контроллеру API).

## Файловая структура проекта

