Минобрнауки России

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

**Институт дополнительного образования**

**Высшая инженерная школа**

**ВЫПУСКНАЯ Квалификационная РАБОТА**

**разработка приложения для сопровождения производства контроллеров присоединения и их последующего учета**

по программе профессиональной переподготовки:

«Разработчик прикладного программного обеспечения (Язык Python)»

Выполнил(а):

Ким Алексей Алексеевич

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель:

преподаватель, Журкин Алексей Михайлович

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2020

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc32846344)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc32846345)

[1.1.Описание устройства контроллера присоединения 5](#_Toc32846346)

[1.2. Требования к функциональности приложения 6](#_Toc32846347)

[1.2.1. Основные требования к приложению 6](#_Toc32846348)

[1.2.2. Детальные требования к приложению 6](#_Toc32846349)

[1.2.3. Дополнительные возможности программы 8](#_Toc32846350)

[1.3. Краткий обзор распространенных приложений и средств для учета контроллеров 10](#_Toc32846351)

[2.ВЫБОР ИСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ПРИЛОЖЕНИЯ 11](#_Toc32846352)

[2.1. Выбор фреймворка 11](#_Toc32846353)

[2.2. Выбор СУБД 12](#_Toc32846354)

[2.2.1.Описание Django ORM 12](#_Toc32846355)

[2.2.2 Проблемы, решаемые при помощи ОRM 13](#_Toc32846356)

[2.3. Выбор инструмента для создания визуального интерфейса приложения 14](#_Toc32846357)

[3. РАЗРАБОТКА МОДУЛЕЙ ПРИЛОЖЕНИЯ 15](#_Toc32846358)

[3.1. Разработка интерфейса приложения 15](#_Toc32846359)

[3.2. Разработка модели 18](#_Toc32846360)

[3.2.1. Разработка требований к функциональности модели 18](#_Toc32846361)

[3.2.2. Разработка классов модели 18](#_Toc32846362)

[3.2.3 Описание классов модели 19](#_Toc32846363)

[3.3. Разработка структуры проекта 21](#_Toc32846364)

[3.3.1 Описание структуры проекта 22](#_Toc32846365)

[ЗАКЛЮЧЕНИЕ 24](#_Toc32846366)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc32846367)

[Приложение 1. Код html-шаблонов 26](#_Toc32846368)

[Приложение 2. Код классов модели 70](#_Toc32846369)

[Приложение 3. Код классов представлений 72](#_Toc32846370)

[Приложение 4. Код модулей-маршрутизаторов urls.py 79](#_Toc32846371)

[Приложение 5. Код служебных модулей проекта 82](#_Toc32846372)

# ВВЕДЕНИЕ

Одним из важнейших аспектов современного мира является автоматизация производств и различных объектов, как бытового, так и промышленного назначения, которая невозможна без использования автоматических систем. В виду того, что данные системы представляют собой сложные комплексы, включающие в себя множество компонентов, то у их производителей и эксплуатантов, в конечном счете, возникает необходимость доступа к информации об этих компонентах для их ремонта или модернизации.

Ключевой частью таких систем являются контроллеры присоединения, используемые для сбора, передачи, обработки и мониторинга информации о состоянии этих систем, а также автоматического управления ими.

Объектом исследования в данной работе является разработка систем учета компонентов, входящих в состав контроллеров присоединения.

Предметом исследования являются программные средства учета компонентов, входящих в состав контроллеров.

Целью работы является разработка прототипа приложения для сопровождения процесса производства контроллеров присоединения и их дальнейшего учета. Под сопровождением в данном случае подразумевается автоматическое заполнение форм эксплуатационной документации, а также доступ к информации о произведенных ранее контроллерах присоединения и входящих в их состав компонентах.

Задачами работы являются:

1. определение необходимой функциональности будущего приложения;
2. выбор инструментов разработки;
3. разработка интерфейса будущего приложения;
4. разработка модулей приложения.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1.Описание устройства контроллера присоединения

Контроллер, как правило, представляет собой металлический корпус с направляющими, в которые устанавливаются платы электроники. Платы электроники различаются физическим исполнением и имеют уникальный заводской номер. Набор плат электроники определяет основную функциональность контроллера. Контроллеры также делятся на несколько типов и отличаются количеством слотов под установку модулей и наличием либо отсутствием лицевой панели управления с экраном. Корпус контроллера рассчитан на весь срок службы и не подлежит модернизации.

Таким образом, основными компонентами контроллера являются входящие в его состав:

1. платы электроники, имеющие конечный ресурс, они подлежат ремонту или модернизации;
2. лицевые панели управления или другие устройства, являющиеся неотъемлемой частью контроллера.

Все эти компоненты необходимо учитывать для дальнейшего доступа к этой информации, что даст возможность обслуживать и модернизировать контроллеры присоединения.

Информацию о составных частях контроллеров нужно хранить и систематизировать, поэтому существует необходимость иметь подходящий для этого инструмент. В качестве такого инструмента должно выступать приложение, взаимодействующее с базой данных, в которой и будет храниться вся необходимая информация.

## 1.2. Требования к функциональности приложения

В данной главе приведены требования для приложения, которое будет использоваться для сопровождения производства контроллеров присоединения и их последующего учета.

### 1.2.1. Основные требования к приложению

1. информация о модулях, и контроллерах в сборе хранится в базе данных;
2. приложение должно работать в браузере, что будет обеспечивать кроссплатформенность;
3. приложение должно предоставлять удобную навигацию по базе данных и возможность поиска информации о модулях и контроллерах по их заводским номерам;
4. файл базы данных должен храниться на сервере и обращение к нему должно осуществляться удаленно с целью доступности базы с разных рабочих мест;
5. приложение должно исключать возможность создания модулей и контроллеров с одинаковыми заводскими номерами;
6. приложение должно исключать возможность привязки одного и того-же модуля к более чем 1 слоту в контроллере и более чем к 1 контроллеру (что физически невозможно).

### 1.2.2. Детальные требования к приложению

База данных должна содержать следующую информацию о модулях:

1. заводской номер модуля;
2. ревизия модуля;
3. наименование модуля (тип модуля);
4. номер партии;
5. фирма-производитель;
6. дата производства модуля;
7. дата ОТК;
8. hash-сумма прошивки;
9. ФИО инженера, проводившего проверку;
10. примечание;
11. заводской номер контроллера, в который входит модуль (заполняется автоматически, после внесения отметки об установке модуля в контроллер);
12. наименование проекта, в который входит модуль (заполняется автоматически, после внесения отметки об установке модуля в контроллер).

База данных должна содержать следующую информацию о контроллерах:

1. номер заказа;
2. заводской номер контроллера;
3. информация о слотах контроллера:
4. номер слота;
5. наименование модуля, входящего в состав слота, либо информация о том, что слот пуст;
6. заводской номер модуля в слоте;
7. номер модуля лицевой панели, либо номер другого оборудования, входящего в состав контроллера (только для комплектации контроллеров с лицевой панелью управления).
8. дата производства контроллера;
9. дата ОТК;
10. наименование проекта;
11. руководитель проекта (ФИО).

Приложение должно иметь функцию поиска модулей по их заводским номерам. Результатом поиска является информация о модуле, которая содержится в Таблице модулей.

Приложение должно осуществлять поиск контроллеров по их заводским номерам. Результатом поиска является наименование проекта, в который входит данный контроллер.

### 1.2.3. Дополнительные возможности программы

Приложение должно выполнять автоматическое заполнение шаблонов Формуляра на контроллер и Паспорта на модуль в формате .doc или .docx.

Паспорт на модуль содержит следующие поля:

1. дата производства модуля;
2. дата прохождения модулем ОТК;
3. дата установки модуля в контроллер (совпадает с датой производства контроллера);
4. заводской номер контроллера, в который модуль был установлен и порядковый номер слота.

Формуляр на контроллер содержит следующие поля:

1. дата производства контроллера;
2. дата прохождения контроллером ОТК;
3. список модулей, входящих в состав контроллера (наименования и заводские номера);
4. даты прохождения модулями ОТК.

Для заполнения Формуляра на контроллер или Паспорта на модуль в главном окне приложения должно присутствовать меню Заполнение шаблонов, в котором должно быть 3 пункта:

1. формуляр на контроллер,
2. паспорт на модуль,
3. весь проект.

При выборе пункта меню «Паспорт на модуль» должно появиться диалоговое окно с выпадающим списком, в котором нужно выбрать:

1. тип модуля;
2. поле, в которое нужно ввести серийный номер модуля.

После того, как данные будут внесены, необходимо нажать кнопку «Заполнить шаблон», после чего на рабочем столе должен появиться заполненный Паспорт на модуль. Имя файла должно иметь формат «Тип модуля/Заводской номер».

При выборе пункта меню «Формуляр на контроллер» должно появиться диалоговое окно с выпадающим списком, в котором нужно выбрать:

1. тип контроллера (4, 5, 7, 14 слотов, 14 слотов с лицевой панелью управления)
2. заводской номер.

После того, как данные будут внесены, необходимо нажать кнопку «Заполнить шаблон», после чего на рабочем столе должна появиться папка с заполненным Формуляром на контроллер и заполненными Паспортами на модули, входящими в этот контроллер. Имя папки должно иметь формат «Наименование проекта/формуляр/паспорта».

При выборе пункта «Весь проект» должно появиться диалоговое окно с полем, в которое нужно ввести номер заявки на изготовление (Уникальный идентификатор проекта). После заполнения поля необходимо нажать кнопку «Заполнить шаблоны», после чего на рабочем столе должна появиться папка с несколькими подпапками. Количество подпапок должно соответствовать количеству контроллеров, входящих в состав проекта. Имя подпапки должно иметь формат «Заводской номер контроллера». Подпапка должна содержать заполненный Формуляр на контроллер и заполненные Паспорта на модули, входящие в состав этого контроллера.

## 1.3. Краткий обзор распространенных приложений и средств для учета контроллеров

Исходя из ТЗ, становится ясно, что модель данных для сопровождения производства контроллеров присоединения и их последующего учета достаточно проста, поэтому возможно использование таких средств как платформа 1С или электронные таблицы Microsoft Excel, но оба этих метода имеют свои недостатки. Платформа 1С является коммерческой и нуждается в разработке специальной конфигурации, адаптированной под нужды производства. Продукт Microsoft Excel также является коммерческим, а хранение большого количества данных в электронных таблицах быстро становится громоздким и неудобным.

Также у перечисленных выше продуктов отсутствуют специальные функции:

1. заполнение паспортов на модули;
2. заполнение формуляров на контроллеры;
3. заполнение шаблонов (комплекта документации на весь проект);

Анализ рынка программных продуктов, которые бы могли обеспечить необходимую функциональность, показал, что использование этих продуктов возможно, но они являются универсальными и не адаптированными для сопровождения процесса производства контроллеров присоединения. Более того, все эти продукты являются коммерческими. Таким образом, имеется необходимость создания простого, специализированного и инструмента для решения данной проблемы.

# 2.ВЫБОР ИСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ ПРИЛОЖЕНИЯ

Так, как приложение будет использоваться на разных рабочих местах, то вся информация должна храниться на сервере. На предприятиях возможно использование таких операционных систем как Windows, Linux или MacOS, поэтому приложение обязано быть кроссплатформенным. Этому условию отвечают Веб-приложения. Для создания таких приложений с минимальными трудозатратами необходимо выбрать фреймворк, также требуется выбрать СУБД, в которой будет храниться информация.

## 2.1. Выбор фреймворка

Веб-фреймворк - это коллекция модульных инструментов, которые позволяют разработчику абстрагироваться от сложных и рутинных стадий в веб-разработке. Например, большинство веб-сайтов имеют одну и ту-же базовую функциональность: возможность соединиться с базой данных, установить URL маршруты, отобразить содержимое на странице, должным образом управлять безопасностью и так далее. Вместо того, чтобы разрабатывать всю эту функциональность с нуля, программисты с течением времени разработали веб-фреймворки: Django и Flask на Python, Rails на Ruby, и Express на JavaScript [10].

Так как Django написан на Python и адаптирован для работы с ним, то для данного проекта целесообразно выбрать фреймворк Django, как основной инструмент для разработки основы приложения.

Django наследует подход Python «все включено» и по умолчанию поддерживает основные задачи в веб-разработке:

1. аутентификация пользователя;
2. шаблоны, пути и представления;
3. страницу администратора;
4. надежную систему безопасности;
5. поддержку нескольких СУБД.

## 2.2. Выбор СУБД

Данные, с которыми работают Веб-приложения, хранятся в базах данных. База данных – это файл или набор файлов, хранящий структурированную определенным образом информацию. Для обработки этой информации используются особые программы, называемые системами управления базами данных, или СУБД [5].

Базы данных делятся на несколько видов по способу структурирования содержащейся в них информации. Реляционные базы данных служат для хранения информации, организованной в виде связанных друг с другом таблиц, и в настоящее время имеют наибольшее распространение [1].

В качестве СУБД в данной работе будет использоваться SQLite3, т.к. она поддерживается фреймворком Django по умолчанию. Более того, фреймворк Django имеет удобный механизм взаимодействия с базой данных, который ускоряет разработку. Этот механизм называется Django ORM (ObjectRelationalMapper), описание которого приведено далеев п.2.2.1.

### 2.2.1.Описание Django ORM

В Django, взаимодействие с базой данных осуществляется посредством ORM -объектно-реляционного отображения. По-существу, данные, хранимые в базе данных, могут быть инкапсулированы в модели. Главным назначением ORM является передача данных между реляционной базой данных и моделью приложения. ORM автоматизирует эту передачу. ORM, как следует из его названия, соотносит атрибуты объектов к полям таблицы. ORM также может получать данные из таблиц [9].

Модель – это программный модуль, входящий в состав приложения, который служит посредником между остальными его модулями и базой данных. Говоря другими словами, модель – представление базы данных, ее таблиц, полей, индексов и связей в терминологии языка программирования, на котором пишется приложение [1].

### 2.2.2 Проблемы, решаемые при помощи ОRM

ORM имеет определенные преимущества перед традиционным подходом. Главным преимуществом ORM является быстрая разработка. ORM делает проект более переносимым, т.к. в случае использования ORM значительно легче поменять базу данных [3].

В прошлом, веб-разработчикам требовалось знание баз данных. База данных была важным компонентом с самого начала разработки. Языки программирования, используемые для веб-разработки, используют классы и объекты для представления данных. Конкретный класс используется для определенной структуры данных в веб-приложениях. Затем, такая-же схема данных заносится в базу данных. Это требует навыков и знания SQL [7].

Знание SQL также недостаточно, т.к. реализация SQL слегка отличается от базы к базе. Это стало сложной и затратной по времени задачей. Таким образом, чтобы решить эту проблему, концепция ORM, была применена в веб-фреймворках. ORM автоматически создают схему базы данных опираясь на реализованные классы/модели. Они генерируют SQL запросы из кода Python для конкретной базы данных.ORM позволяет разработчику создать проект используя один язык, которым в случае этой работы является Python [6].

Цель ORM – сделать системы баз данных более доступными, абстрагируя процесс создания запросов: он сам создает код на SQL. К сожалению, этот абстрактный слой не позволяет выполнять низкоуровневые задачи[2].

## 2.3. Выбор инструмента для создания визуального интерфейса приложения

Для быстрого создания кросс браузерного адаптивного приложения необходимо использовать инструмент, включающий в себя html-шаблоны, таблицы стилей CSS и JavaScript плагины. В качестве такого инструмента в данной работе использован фреймворк Bootstrap, как наиболее распространенный и используемый на данный момент [4, 8].

Для удобства работы мной был выбран визуальный онлайн-конструктор html-шаблонов для Bootstrap, расположенный по адресу www.layoutit.com.

# 3. РАЗРАБОТКА МОДУЛЕЙ ПРИЛОЖЕНИЯ

## 3.1. Разработка интерфейса приложения

На основе требований параграфа 1.2. «Требования к функциональности приложения» было разработано 7 html-шаблонов:

1. controller\_form\_filling.html– страницa приложения «Заполнение шаблонов формуляра и паспортов» (рис. 3.1.);

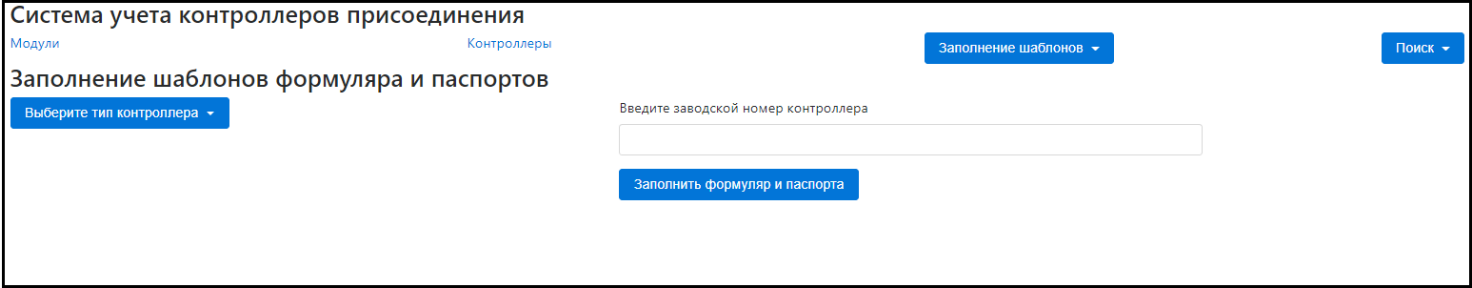


Рисунок 3.1. – Заполнение шаблонов формуляра и паспортов

1. controller\_search.html –страница приложения «Поиск контроллера» (рис. 3.2);

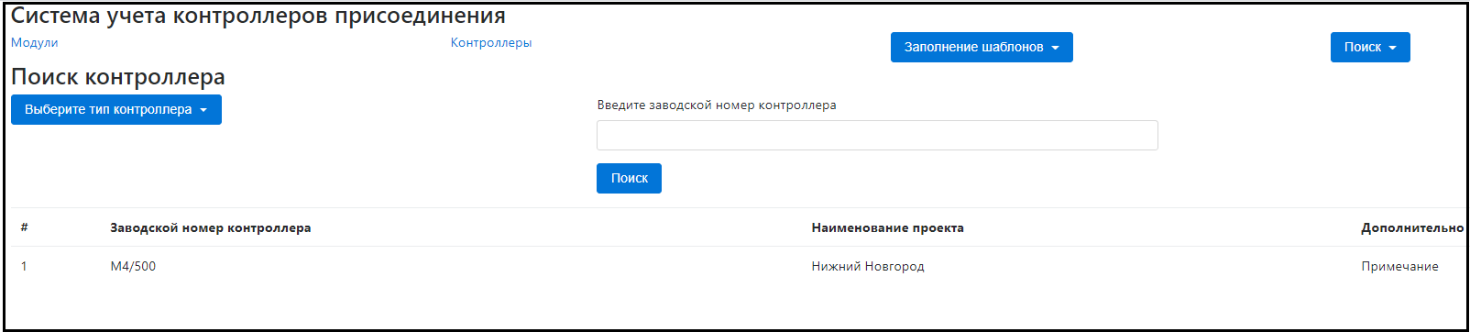


Рисунок 3.2 – Поиск контроллера

1. controllers\_list.html –страница приложения «Вывод списка контроллеров» (рис. 3.3);

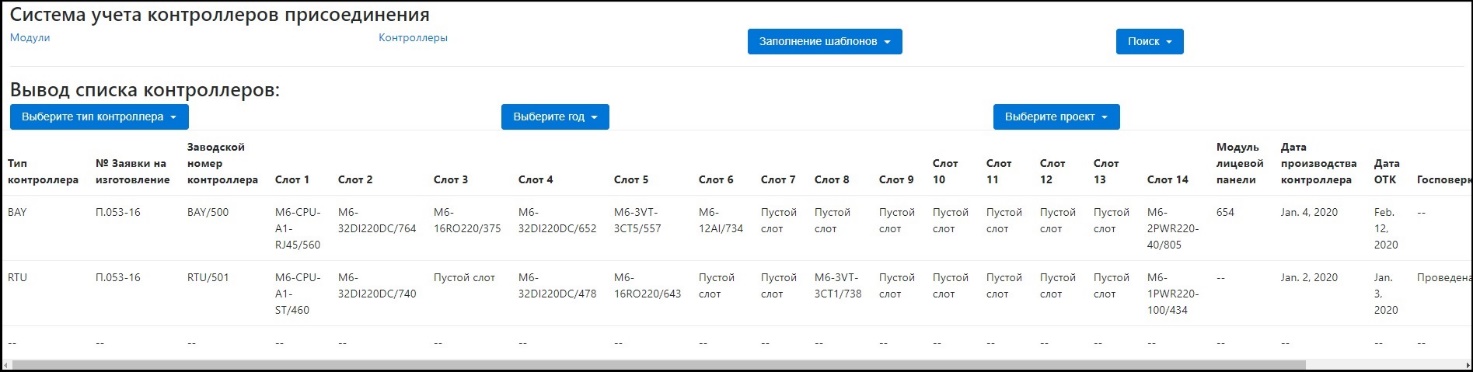


Рисунок 3.3. – Вывод списка контроллеров

1. modules\_list.html – страница приложения «Вывод списка модулей» (рис. 3.4);

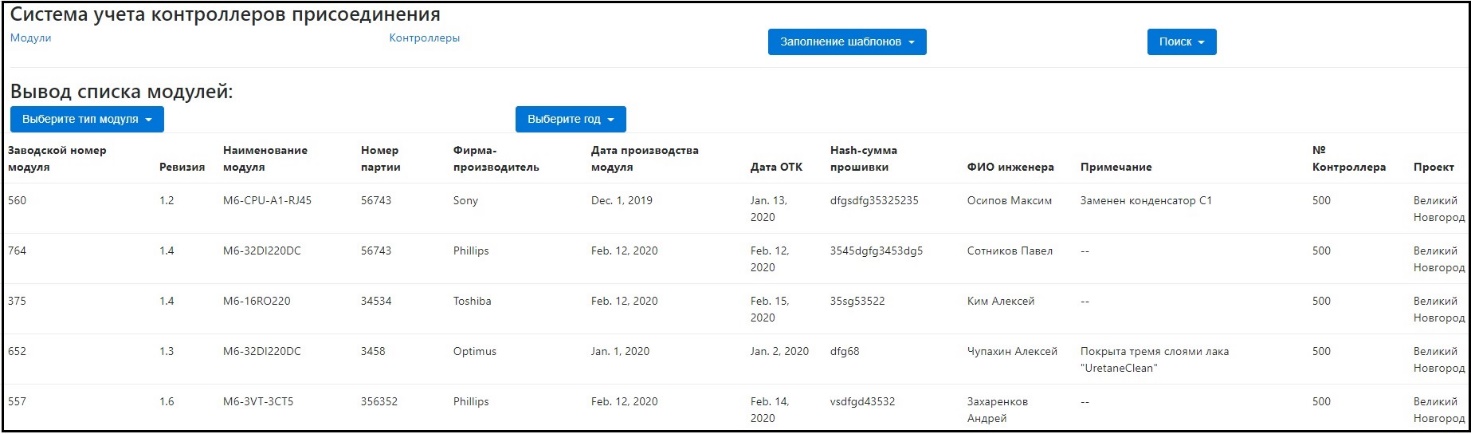


Рисунок 3.4 – Вывод списка модулей

1. module\_passport\_filling.html – страница приложения «Заполнение шаблона паспорта на модуль» (рис. 3.5);

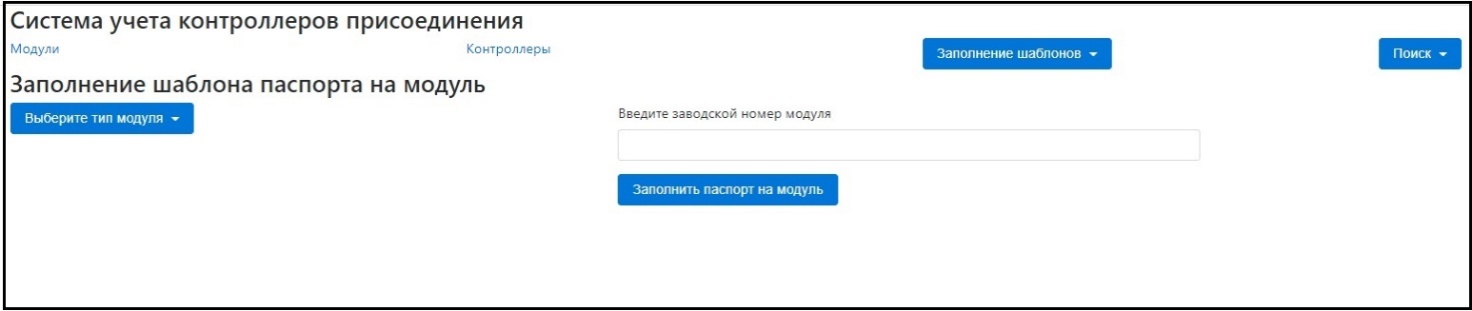


Рисунок 3.5. – Заполнение шаблона паспорта на модуль

1. module\_search.html – страница приложения «Поиск модуля» (рис.3.6);

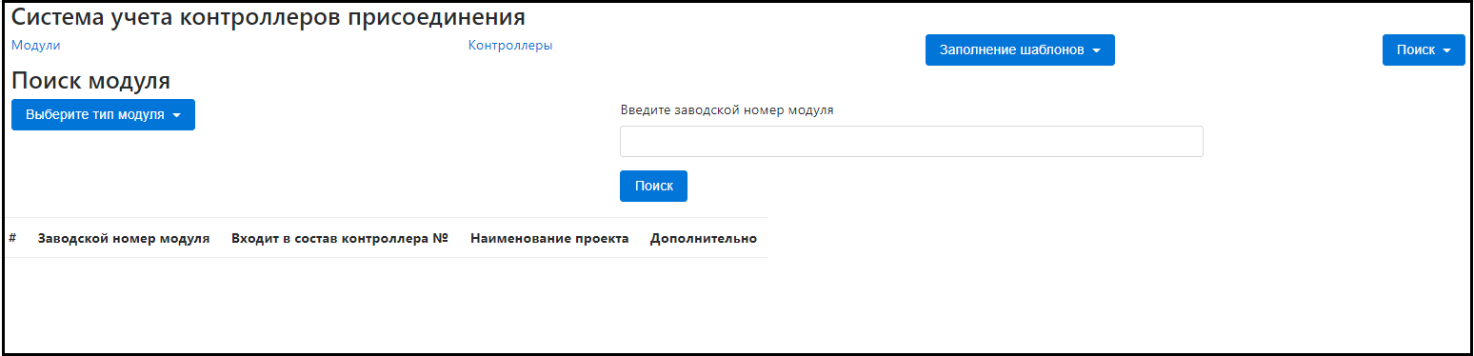


Рисунок 3.6. – Поиск модуля

1. project\_forms\_filling.html – страницa приложения «Заполнение шаблонов формуляров и паспортов на модули для всего проекта» (рис.3.7).

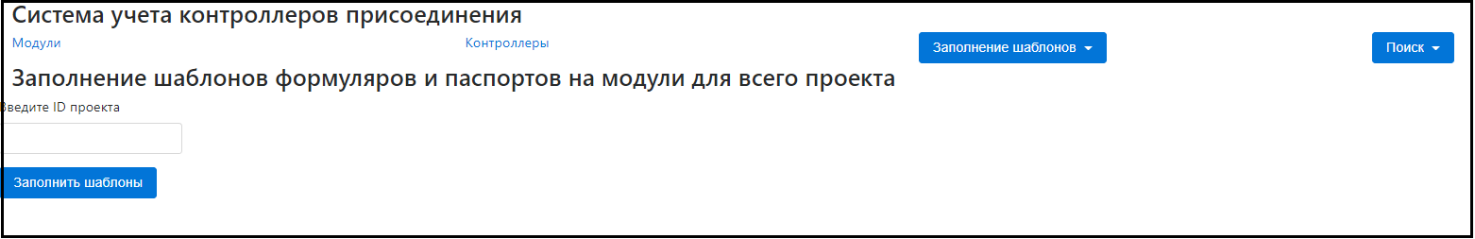


Рисунок 3.7. – Заполнение шаблонов формуляров и паспортов на модули для всего проекта.

Код шаблонов, перечисленных выше, приведен в приложении 1.

## 3.2. Разработка модели

### 3.2.1. Разработка требований к функциональности модели

Требования к функциональности модели должны быть разработаны исходя из требований к базе данных приложения:

1. БД должна содержать минимум пустых полей. Например, если контроллер на 14 слотов имеет только 4 слота с установленными модулями, а в остальных слотах установлены заглушки, то в модели должны отсутствовать пустые поля, отвечающие за слоты;
2. Возможность добавления в БД контроллеров с одинаковыми серийными номерами должна быть исключена, т.к. каждый контроллер имеет свой уникальный серийный номер;
3. Возможность добавления в БД модулей с одинаковыми серийными номерами должна быть исключена, т.к. каждый модуль имеет свой уникальный серийный номер;
4. Должна быть исключена возможность добавления одного модуля в более чем 1 слот и контроллер;
5. Должна быть реализована каскадная связь между основными и подчиненными таблицами для автоматического удаления неактуальных записей.

### 3.2.2. Разработка классов модели

На основе требований к структуре модели была составлена схема БД (рис. 3.9), используя которую была создана модель, состоящая из 4 классов: Controller, Module, ControllerModule, Slot. Каждый класс представляет собой таблицу в БД, т.е. был использован механизм DjangoORM (рис. 3.8). Данные классы описаны в п. 3.2.3.



Рисунок 3.8. – Django ORM

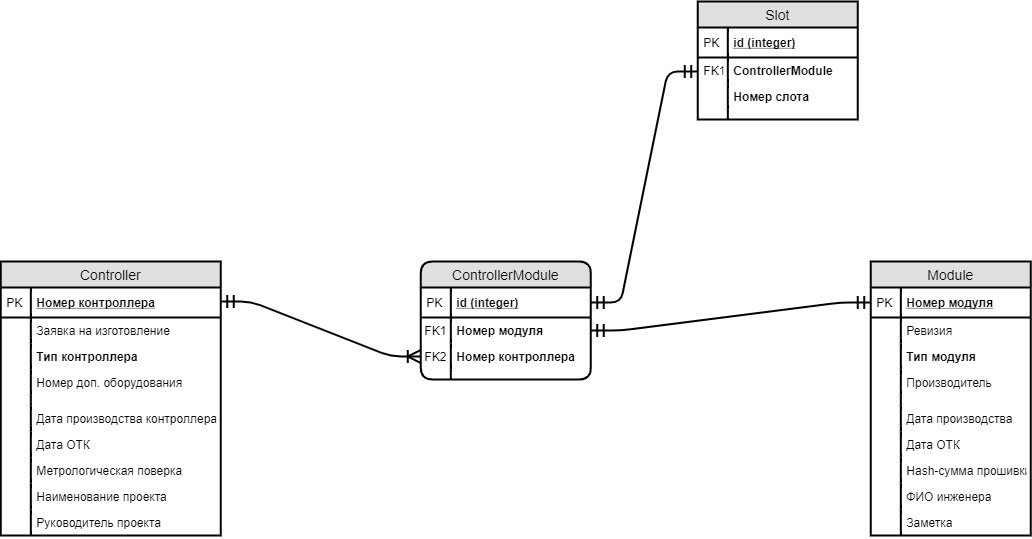


Рисунок 3.9. – Схема БД приложения

### 3.2.3 Описание классов модели

Controller – данная таблица содержит данные о контроллере. Чтобы в таблице отсутствовали пустые записи, относящиеся к пустым слотам (слотам, в которые не установлены модули), информация о слотах в данной таблице отсутствует и перенесена в отдельную таблицу Slot. Первичным ключом(PK) в данной таблице является заводской номер контроллера, что исключает возможность создания контроллеров с одинаковыми серийными номерами.

Module – данная таблица содержит данные о модулях. Первичным ключом (PK) в данной таблице является заводской номер модуля, что исключает возможность создания модулей с одинаковыми серийными номерами.

ControllerModule–данная таблица реализует привязку модуля к единственному контроллеру. Первичным ключом(PK) данной таблицы является переменная типа int. Каждая запись таблицы создает оригинальную связку«ControllerModule». Для реализации этой функциональности внешний ключ(FK1) - «Номер модуля» ссылается на первичный ключ(PK) таблицы Module «Номер модуля»(свойство unique = True), а внешний ключ(FK2) – «Номер контроллера» ссылается на первичный ключ(PK) таблицы Controller «Номер контроллера». Это должно быть сделано для невозможности привязки модуля к более чем 1 контроллеру.

Slot– данная таблица реализует привязку модуля к единственному слоту в контроллере. Первичным ключом (PK) в данной таблице является переменная типа int. Каждая запись таблицы создает оригинальную связку «Контроллер-Модуль-Слот». Для реализации этой функциональности внешний ключ(FK1) – «ControllerModule» ссылается на первичный ключ(PK) таблицы ContollerModule «id» (свойство unique = True), а номер слота задается обязательным параметром «Номер слота».

Код перечисленных выше классов, реализующих таблицы модели, приведен в приложении 2.

## 3.3. Разработка структуры проекта

В качестве фреймворка для создания приложения был выбран фреймворк Django, который использует шаблон проектирования MVT (Модель-Представление-Шаблон), похожий на шаблон проектированияMVC (Модель-Представление-Контроллер).Структура проекта будет создана с использованием этого шаблона. На рисунке 3.10 приведена схема структуры проекта, на которой изображены 4 базовых модуля приложения, которые реализуют шаблон проектирования MVT и обеспечивают функциональность приложения:

1. urls.py
2. views.py
3. models.py
4. имяфайла.html

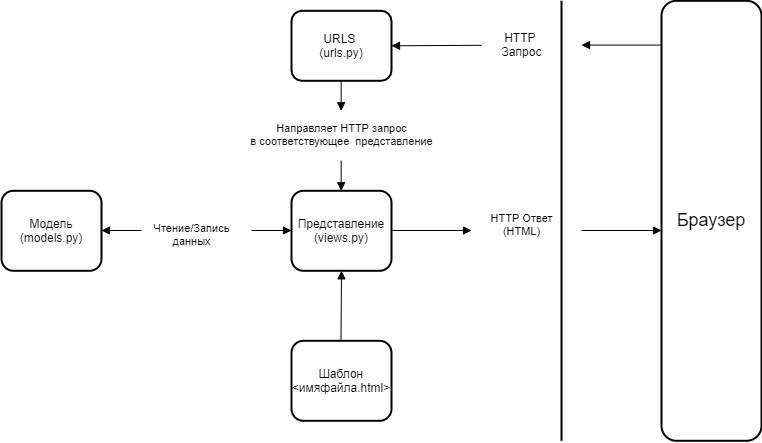


Рисунок 3.10 – Структура проекта

### 3.3.1 Описание структуры проекта

В данном пункте приведено описание базовых модулей приложения.

Модуль models.py – представляет собой Модель. Включает в себя все классы, отвечающие за представление БД: Module, Controller, ControllerModule, Slot.

Модуль-маршрутизатор urls.py – содержит ссылки на конкретные представления и направляет в них HTTP-запросы. В проекте содержится 2 таких файла 1 - в каталоге проекта, 2 – в каталоге приложения. Первый осуществляет маршрутизацию запросов между приложениями, второй – между классами-представлениями в каждом приложении;

Модуль views.py – включает в себя Представления, отвечающие за отрисовку HTML-шаблонов и отправку данных пользователю, а также взаимодействие с Моделью. Состоитиз7классов-представлений:

1. ControllerFormFillingView – обеспечивает работу страницы приложения «Заполнение шаблонов формуляра и паспортов»;
2. ControllerSearchView – обеспечивает работу страницы приложения «Поиск контроллеров»;
3. ControllersListView – обеспечивает работу страницы приложения «Вывод списка контроллеров»;
4. ModulesListView – обеспечивает работу страницы приложения «Вывод списка контроллеров»;
5. ModulePassportFillingView – обеспечивает работу страницы приложения «Заполнение шаблона паспорта на модуль»;
6. ModuleSearchView – обеспечивает работу страницы приложения «Поиск модуля»;
7. ProjectFormsFillingView – обеспечивает работу страницы приложения «Заполнение шаблонов формуляров и паспортов на модули для всего проекта».

ИмяФайла.html – файлы шаблонов. Каждый файл представляет собой одну страницу приложения. Количество файлов шаблонов равняется количеству классов-представлений в модуле views.py. Имена файлов шаблонов соотносятся с названиями классов-представлений, с которыми они связаны.

Код модуля представлений (views.py) и код модулей-маршрутизаторов (urls.py) приведены в приложениях 3 и 4 соответственно. В приложении 5 приведен код служебных модулей проекта.

# ЗАКЛЮЧЕНИЕ

Итогом проделанной работы являются следующие результаты:

1. Проведен анализ предметной области, описан состав контроллеров присоединения, определены основные компоненты контроллеров, которые подлежат учету. Согласно проведенному анализу разработаны детальные требования к приложению;
2. Проведен поиск необходимых инструментов для создания приложения. Для создания приложения был выбран Веб-фреймворк Django, как наиболее отвечающий разработанным требованиям;
3. С помощью визуального онлайн-конструктора шаблонов html-страниц для Bootstrap (layoutit.com) был разработан интерфейс, отвечающий требованиям к приложению;
4. Для реализации требуемой логики работы БД была разработана модель.
5. Разработаны модули приложения, реализующие необходимую функциональность.
6. Не были реализованы некоторые функции клиентской части, такие как поиск по базе данных и заполнение шаблонов формуляров эксплуатационной документации.

Основные цели работы достигнуты, задачи выполнены.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дронов, В.А. Django Практика создания Web-cайтов на Python [Текст]. – СПб.: БХВ-Петербург, 2016. – 528 c.
2. Данжу, Д. Путь Python [Текст]. – СПб.: Питер, 2020. – 256 с.
3. Меле, А.Django 2 в примерах [Текст] / пер. с анг. Д.В. Плотниковой. – М.: ДМК Пресс, 2019. – 408 с.
4. Морето, С. Bootstrapв примерах [Текст] / пер. с анг. Рагимов Р.Н. / Науч. Ред. Киселев, А. Н. – М.: ДМК Пресс, 2017. – 314 с.
5. Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. [Текст]. – СПб.: Питер, 2020. – 512 с.
6. Ravindran, А. Django Design Patterns and Best Practices [Текст]–Birmingham: Packt Publishing Ltd., 2015 – 201с.
7. Greenfield, D. Two Scoops of Django 1.11: Best Practices for Django [Текст]. / Greenfield, A. – Two Scoops Press, 2017. – 515 с.
8. Spurlock, J. Bootstrap [Текст]. –Sebastopol: O’Reilly, 2013. – 109 с.
9. Azzopardi, L.Tango With Django[Текст]. / Maxwell, D.–Lean Publishing, 2016. – 281 с.
10. Vincent, W. Django for Beginners [Текст]. – 2018.- 195 с.

# Приложение 1. Код html-шаблонов

Код шаблона controller\_form\_filling.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Формуляр на контроллер</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3>

Заполнение шаблонов формуляра и паспортов

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип контроллера

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for type in controller\_types %}

<a class="dropdown-item disabled" href="#">{{type}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">

Введите заводской номер контроллера

</label>

<input type="email" class="form-control" id="exampleInputEmail1">

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">

Заполнить формуляр и паспорта

</button>

</form>

</div>

<div class="col-md-4">

</div>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона controller\_search.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Поиск контроллера</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3>

Поиск контроллера

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип контроллера

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for type in controller\_types%}

<a class="dropdown-item disabled" href="#">{{type}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">

Введите заводской номер контроллера

</label>

<input type="email" class="form-control" id="exampleInputEmail1">

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">

Поиск

</button>

</form>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<table class="table">

<thead>

<tr>

<th>

#

</th>

<th>

Заводской номер контроллера

</th>

<th>

Наименование проекта

</th>

<th>

Дополнительно

</th>

</tr>

</thead>

<tbody>

<tr>

<td>

1

</td>

<td>

M4/500

</td>

<td>

Нижний Новгород

</td>

<td>

Примечание

</td>

</tr>

</tbody>

</table>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона controllers\_list.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Список контроллеров</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<hr>

<div class="row">

<div class="col-md-12">

<h3>

Вывод списка контроллеров:

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип контроллера

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<!--<a class="dropdown-item" href="#">M4</a> <a class="dropdown-item" href="#">M5</a> <a class="dropdown-item" href="#">M7</a> <a class="dropdown-item" href="#">RTU</a> <a class="dropdown-item" href="#">BAY</a> <a class="dropdown-item" href="#">Compact</a>-->

{% for type in controller\_types %}

<a class="dropdown-item" href="#">{{type}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите год

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<!--<a class="dropdown-item" href="#">M4</a> <a class="dropdown-item" href="#">M5</a> <a class="dropdown-item" href="#">M7</a> <a class="dropdown-item" href="#">RTU</a> <a class="dropdown-item" href="#">BAY</a> <a class="dropdown-item" href="#">Compact</a>-->

{% for date in qc\_dates %}

<a class="dropdown-item" href="#">{{date}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите проект

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<!--<a class="dropdown-item" href="#">M4</a> <a class="dropdown-item" href="#">M5</a> <a class="dropdown-item" href="#">M7</a> <a class="dropdown-item" href="#">RTU</a> <a class="dropdown-item" href="#">BAY</a> <a class="dropdown-item" href="#">Compact</a>-->

{% for name in project\_names %}

<a class="dropdown-item" href="#">{{name}}</a>

{% endfor %}

</div>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<table class="table" id="bagua-table">

<thead>

<tr>

<th>

Тип контроллера

</th>

<th>

№ Заявки на изготовление

</th>

<th>

Заводской номер контроллера

</th>

<th>

Слот 1

</th>

<th>

Слот 2

</th>

<th>

Слот 3

</th>

<th>

Слот 4

</th>

<th>

Слот 5

</th>

<th>

Слот 6

</th>

<th>

Слот 7

</th>

<th>

Слот 8

</th>

<th>

Слот 9

</th>

<th>

Слот 10

</th>

<th>

Слот 11

</th>

<th>

Слот 12

</th>

<th>

Слот 13

</th>

<th>

Слот 14

</th>

<th>

Модуль лицевой панели

</th>

<th>

Дата производства контроллера

</th>

<th>

Дата ОТК

</th>

<th>

Госповерка

</th>

<th>

Название проекта

</th>

<th>

ФИО Руководителя проекта

</th>

</tr>

</thead>

<tbody>

{% for controller in controllers %}

<tr>

<td>

{{controller.controller.controller\_type}}

</td>

<td>

{{controller.controller.manufacturing\_request\_number}}

</td>

<td>

{{controller.controller}}

</td>

{% for slot in controller.slot %}

{%if slot%}

<td>

{{slot.controller\_module.module}}

</td>

{% else %}

<td>

{{"Пустой слот"}}

</td>

{% endif %}

{% endfor %}

<!--<td>-->

<!--{{controller.controller.another\_equipment\_numbers}}-->

<!--</td>-->

{% if controller.controller.another\_equipment\_numbers %}

<td>

{{controller.controller.another\_equipment\_numbers}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

<!--<td>-->

<!--{{controller.controller.controller\_production\_date}}-->

<!--</td>-->

{% if controller.controller.controller\_production\_date %}

<td>

{{controller.controller.controller\_production\_date}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

<!--<td>-->

<!--{{controller.controller.qc\_date}}-->

<!--</td>-->

{% if controller.controller.qc\_date %}

<td>

{{controller.controller.qc\_date}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

<!--<td>-->

<!--{{controller.controller.metrology\_check}}-->

<!--</td>-->

{% if controller.controller.metrology\_check %}

<td>

{{'Проведена'}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

<!--<td>-->

<!--{{controller.controller.project\_name}}-->

<!--</td>-->

{% if controller.controller.project\_name %}

<td>

{{controller.controller.project\_name}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

<!--<td>-->

<!--{{controller.controller.project\_manager\_name}}-->

<!--</td>-->

{% if controller.controller.project\_manager\_name %}

<td>

{{controller.controller.project\_manager\_name}}

</td>

{% else %}

<td>

{{"--"}}

</td>

{% endif %}

</tr>

{% endfor %}

{% for i in a %}

<td>

{{i}}

</td>

{% endfor %}

</tbody>

</table>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона module\_passport\_filling.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Заполнение паспорта на модуль</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3>

Заполнение шаблона паспорта на модуль

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип модуля

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for name in module\_names %}

<a class="dropdown-item disabled" href="#">{{name}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">

Введите заводской номер модуля

</label>

<input type="email" class="form-control" id="exampleInputEmail1">

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">

Заполнить паспорт на модуль

</button>

</form>

</div>

<div class="col-md-4">

</div>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона module\_search.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Поиск модуля</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3>

Поиск модуля

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип модуля

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for name in module\_names%}

<a class="dropdown-item" href="#">{{name}}</a>

{% endfor%}

</div>

</div>

</div>

<div class="col-md-4">

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">

Введите заводской номер модуля

</label>

<input type="email" class="form-control" id="exampleInputEmail1">

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">

Поиск

</button>

</form>

</div>

<div class="col-md-4">

</div>

</div>

<div class="row">

<div class="col-md-12">

<table class="table">

<thead>

<tr>

<th>

#

</th>

<th>

Заводской номер модуля

</th>

<th>

Входит в состав контроллера №

</th>

<th>

Наименование проекта

</th>

<th>

Дополнительно

</th>

</tr>

</thead>

<tbody>

<tr>

<td>

1

</td>

<td>

TB - Monthly

</td>

<td>

01/04/2012

</td>

<td>

Default

</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона modules\_list.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Список модулей</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<hr>

<div class="row">

<div class="col-md-12">

<h3>

Вывод списка модулей:

</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите тип модуля

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for name in original\_module\_names %}

<a class="dropdown-item" href="#">{{name}}</a>

{% endfor %}

</div>

</div>

</div>

<div class="col-md-4">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Выберите год

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

{% for year in original\_production\_years %}

<a class="dropdown-item" href="#">{{year}}</a>

{% endfor %}

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<table class="table">

<thead>

<tr>

<th>

Заводской номер модуля

</th>

<th>

Ревизия

</th>

<th>

Наименование модуля

</th>

<th>

Номер партии

</th>

<th>

Фирма-производитель

</th>

<th>

Дата производства модуля

</th>

<th>

Дата ОТК

</th>

<th>

Hash-сумма прошивки

</th>

<th>

ФИО инженера

</th>

<th>

Примечание

</th>

<th>

№ Контроллера

</th>

<th>

Проект

</th>

</tr>

</thead>

<tbody>

{% for mod\_cont in modules\_controllers %}

<tr>

<td>

{{mod\_cont.module.factory\_number}}

</td>

{% if mod\_cont.module.revision %}

<td>

{{mod\_cont.module.revision}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.module\_name %}

<td>

{{mod\_cont.module.module\_name}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.consignment\_number %}

<td>

{{mod\_cont.module.consignment\_number}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.manufacturer %}

<td>

{{mod\_cont.module.manufacturer}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.production\_date %}

<td>

{{mod\_cont.module.production\_date}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.qc\_pass\_date %}

<td>

{{mod\_cont.module.qc\_pass\_date}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.hash %}

<td>

{{mod\_cont.module.hash}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.engineer\_name %}

<td>

{{mod\_cont.module.engineer\_name}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.module.note %}

<td>

{{mod\_cont.module.note}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.controller.factory\_number %}

<td>

{{mod\_cont.controller.factory\_number}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

{% if mod\_cont.controller.project\_name %}

<td>

{{mod\_cont.controller.project\_name}}

</td>

{% else %}

<td>

{{'--'}}

</td>

{% endif %}

</tr>

{% endfor%}

</tbody>

</table>

</div>

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

Код шаблона project\_forms\_filling.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>На весь проект</title>

<meta name="description" content="Source code generated using layoutit.com">

<meta name="author" content="LayoutIt!">

<link href="{% static "myapp/css/bootstrap.min.css" %}" rel="stylesheet"/>

<link href="{% static "myapp/css/style.css" %}" rel="stylesheet">

</head>

<body>

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h3>

Система учета контроллеров присоединения

</h3>

</div>

</div>

<div class="row">

<div class="col-md-3">

<a href="{% url "modules\_list" %}">Модули</a>

</div>

<div class="col-md-3">

<a href="{% url "controllers\_list" %}">Контроллеры</a>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Заполнение шаблонов

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "controller\_form\_filling" %}">Формуляр на контроллер</a> <a class="dropdown-item" href="{% url "module\_passport\_filling" %}">Паспорт на модуль</a> <a class="dropdown-item" href="{% url "project\_forms\_filling" %}">Весь проект</a>

</div>

</div>

</div>

<div class="col-md-3">

<div class="dropdown">

<button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown">

Поиск

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="{% url "module\_search" %}">Поиск модуля</a> <a class="dropdown-item" href="{% url "controller\_search" %}">Поиск контроллера</a>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3>

Заполнение шаблонов формуляров и паспортов на модули для всего проекта

</h3>

</div>

</div>

<div class="row">

<form role="form">

<div class="form-group">

<label for="exampleInputEmail1">

Введите ID проекта

</label>

<input type="email" class="form-control" id="exampleInputEmail1">

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">

Заполнить шаблоны

</button>

</form>

</div>

<div class="col-md-4">

</div>

</div>

<script src="{% static "myapp/js/jquery.min.js" %}"></script>

<script src="{% static "myapp/js/bootstrap.min.js" %}"></script>

<script src="{% static "myapp/js/scripts.js" %}"></script>

<script src="{% static "myapp/js/livetable.js" %}"></script>

</body>

</html>

# Приложение 2. Код классов модели

from django.db import models

# Create your models here.

class Module(models.Model):

factory\_number = models.CharField(primary\_key=True, max\_length=30) # 3D/526 3D/527

revision = models.FloatField(null=True, blank=True)

module\_name = models.CharField(null=True, max\_length=30)

consignment\_number = models.IntegerField(null=True, blank=True)

manufacturer = models.CharField(max\_length=30, blank=True)

production\_date = models.DateField(null=True, blank=True)

qc\_pass\_date = models.DateField(null=True, blank=True)

hash = models.TextField(blank=True)

engineer\_name = models.CharField(max\_length=30, blank=True)

note = models.TextField(blank=True)

def \_\_str\_\_(self):

return f'{self.module\_name}/{self.factory\_number}'

class Controller(models.Model):

factory\_number = models.CharField(primary\_key=True, max\_length=30)

manufacturing\_request\_number = models.CharField(max\_length=30, blank=True)

controller\_type = models.CharField(null=True, max\_length=30)

another\_equipment\_numbers = models.CharField(max\_length=30, blank=True)

controller\_production\_date = models.DateField(null=True, blank=True)

qc\_date = models.DateField(null=True, blank=True)

metrology\_check = models.NullBooleanField(blank=True)

project\_name = models.CharField(max\_length=30, blank=True)

project\_manager\_name = models.CharField(max\_length=30, blank=True)

def \_\_str\_\_(self):

return f'{self.controller\_type}/{self.factory\_number}'

class ControllerModule(models.Model):

module = models.ForeignKey('Module', unique=True, on\_delete=models.CASCADE)

controller = models.ForeignKey('Controller', on\_delete=models.CASCADE)

class Meta:

unique\_together = (('module', 'controller'),)

def \_\_str\_\_(self):

return f'{self.controller}-{self.module}'

class Slot(models.Model):

controller\_module = models.ForeignKey('ControllerModule', unique=True, on\_delete=models.CASCADE)

slot\_number = models.IntegerField(editable=True)

class Meta:

unique\_together = (('slot\_number', 'controller\_module'),)

def \_\_str\_\_(self):

return f'{self.controller\_module}-{self.slot\_number}'

# Приложение 3. Код классов представлений

from datetime import datetime

from django.shortcuts import render

from django.views.generic import View

from django.http import HttpResponse

from .models import Module, Controller, Slot, ControllerModule

# Create your views here.

class ControllerFormFillingView(View):

def get(self, request):

controllers = Controller.objects.all()

controller\_types = []

for controller in controllers:

try:

if controller.controller\_type in controller\_types:

continue

else:

controller\_types.append(controller.controller\_type)

except:

pass

return render(request, 'myapp/controller\_form\_filling.html', {'controller\_types': controller\_types})

class ControllerSearchView(View):

def get(self, request):

controllers = Controller.objects.all()

controller\_types = []

for controller in controllers:

try:

if controller.controller\_type in controller\_types:

continue

else:

controller\_types.append(controller.controller\_type)

except:

pass

return render(request, 'myapp/controller\_search.html', {'controller\_types': controller\_types})

class ControllersListView(View):

def get(self, request):

a = ['--' for i in range(23)]

controller = Controller.objects.all()

controllers = []

controller\_types = []

qc\_dates = []

project\_names = []

for cont in controller:

slots = Slot.objects.select\_related().all().filter(

controller\_module\_\_controller\_\_factory\_number=cont.factory\_number).order\_by('slot\_number')

new\_slots = [None for i in range(1, 15)]

j = 0

for i in range(1, 15):

try:

if slots[j].slot\_number == i:

new\_slots[i-1] = slots[j]

j += 1

except:

pass

controllers.append({'controller': cont, 'slot': new\_slots})

try:

if cont.controller\_type in controller\_types:

continue

else:

controller\_types.append(cont.controller\_type)

except:

pass

try:

if cont.qc\_date.year in qc\_dates:

continue

else:

qc\_dates.append(cont.qc\_date.year)

except:

pass

try:

if cont.project\_name in project\_names:

continue

else:

project\_names.append(cont.project\_name)

except:

pass

return render(request, 'myapp/controllers\_list.html', {'controllers': controllers,

'a': a,

'controller\_types': controller\_types,

'qc\_dates': qc\_dates,

'project\_names': project\_names})

class ModulePassportFillingView(View):

def get(self, request):

modules = Module.objects.all()

module\_names = []

for module in modules:

try:

if module.module\_name in module\_names:

continue

else:

module\_names.append(module.module\_name)

except:

pass

return render(request, 'myapp/module\_passport\_filling.html', {'module\_names': module\_names})

class ModuleSearchView(View):

def get(self, request):

modules = Module.objects.all()

module\_names = []

for module in modules:

try:

if module.module\_name in module\_names:

continue

else:

module\_names.append(module.module\_name)

except:

pass

return render(request, 'myapp/module\_search.html', {'module\_names': module\_names})

class ModulesListView(View):

def get(self, request):

modules = Module.objects.all()

modules\_controllers = []

original\_module\_names = []

original\_production\_years = []

for module in modules:

try:

controller\_module = ControllerModule.objects.select\_related().all().filter(module\_\_factory\_number=module.factory\_number)

modules\_controllers.append({'module': module, 'controller': controller\_module[0].controller})

except:

modules\_controllers.append({'module': module, 'controller': None})

try:

if module.module\_name in original\_module\_names:

continue

else:

original\_module\_names.append(module.module\_name)

except:

pass

try:

if module.production\_date.year in original\_production\_years:

continue

else:

original\_production\_years.append(module.production\_date.year)

except:

pass

return render(request, 'myapp/modules\_list.html', {'modules\_controllers': modules\_controllers,

'original\_module\_names': original\_module\_names,

'original\_production\_years': original\_production\_years})

class ProjectFormsFillingView(View):

def get(self, request):

return render(request, 'myapp/project\_forms\_filling.html')

# Приложение 4. Код модулей-маршрутизаторов urls.py

Код модуля urls.py (директория проекта)

"""proj URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/3.0/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('myapp/', include('myapp.urls')),

]

Код модуля urls.py (директория приложения)

from django.urls import path, include

from django.http import HttpResponse

from .views import ControllerFormFillingView, ControllerSearchView,ControllersListView,\

ModulePassportFillingView, ModuleSearchView, ModulesListView, ProjectFormsFillingView

from .models import Module, Controller, Slot

def get\_controllers(request):

controllers = Slot.objects.select\_related()

response = '<body>'

for controller in controllers:

response += f'{controller}<br>'

response += '</body>'

return HttpResponse(response)

urlpatterns = [

path('controller\_form\_filling/', ControllerFormFillingView.as\_view(), name='controller\_form\_filling'),

path('controller\_search/', ControllerSearchView.as\_view(), name='controller\_search'),

path('controllers\_list/', ControllersListView.as\_view(), name='controllers\_list'),

path('module\_passport\_filling/', ModulePassportFillingView.as\_view(), name='module\_passport\_filling'),

path('module\_search/', ModuleSearchView.as\_view(), name='module\_search'),

path('modules\_list/', ModulesListView.as\_view(), name='modules\_list'),

path('project\_forms\_filling/', ProjectFormsFillingView.as\_view(), name='project\_forms\_filling'),

path('simple\_controllers/', get\_controllers),]

# Приложение 5. Код служебных модулей проекта

Код модуля settings.py (настройки проекта)

***"""***

Django settings for proj project.

Generated by 'django-admin startproject' using Django 3.0.1.

For more information on this file, see

https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.0/ref/settings/

"""

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '299a5h!eu8t\*9)$u!=u2q2!v#d&ugy((-ti7a(wowgzov\*yc6='

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'myapp',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'proj.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'proj.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

# Password validation

# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC\_URL = '/static/'

Код модуля apps.py

from django.apps import AppConfig

class MyappConfig(AppConfig):

name = 'myapp'

Код файла admin.py

from django.contrib import admin

from .models import Controller, Module, Slot, ControllerModule

# Register your models here.

admin.site.register(Controller)

admin.site.register(Module)

admin.site.register(ControllerModule)

admin.site.register(Slot)