Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Лабораторная работа №3 по дисциплине**

**«Технологии разработки программного обеспечения»**

**ИСПОЛНИТЕЛЬ:**

Васильев Д.А.

Группа ИУ5-11М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2022 г.

Москва - 2022

# Тема работы

Doxygen, документация.

# Цель работы

Изучить базовые возможности и получить навыки создания документации на основе исходного кода программ с применением CASE средств. В материалах рассмотрены принципы составления проектной документации на основе исходного кода программ. Приведены инструкции по работе и команды документирования исходных кодов для автоматического составления документации в среде Doxygen. Рассмотрены примеры использования указанных средств. В заключительной части методических указаний приведены контрольные вопросы, список рекомендуемой литературы и пример задания. Ознакомившись с методическими указаниями и разобрав приведённые в нем примеры, студент может получить у преподавателя свой вариант задания и приступить к его выполнению.

# Задание

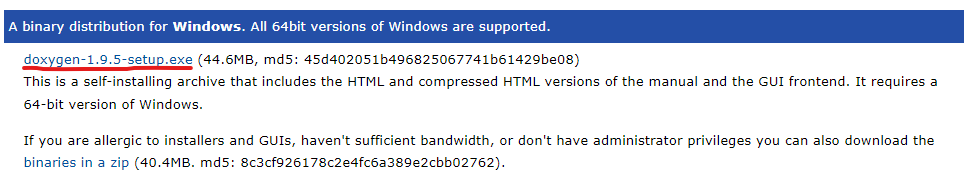
1. Если система документирования Doxygen не установлена, то установить ее.
2. Создать новый проект в Visual Studio.
3. Добавить в проект класс по своему варианту (с методами и свойствами).
4. Выполнить комментирование кода по правилам Doxygen и сохранить проект.
5. Запустить Doxygen и выполнить настройку проекта.
6. Провести документирование и просмотреть полученный результат (html и RTF).
7. Продемонстрировать:

*  краткое и подробное описание класса,
*  краткое и подробное описание функций класса,
*  описание входных и выходных параметров функций,
*  описание переменных класса,
*  описание проекта,
*  описание файла (h и cpp) и добавление ссылки на файл в описание проекта.
*  описание производного класса

# 

## Установка системы документирования Doxygen

На официальном сайте doxygen ([doxygen.nl](https://www.doxygen.nl)) переходим на страницу загрузки дистрибутива и скачиваем необходимую версию ПО.



После этого производим установку в соответствии с указаниями.

## Создание нового проекта

Для демонстрации работы системы документирования была выбрана часть проекта АИС «Образовательный портал», написанная на Django.

## Добавление класса в проект

Вместо создания нового класса продемонстрируем работу системы на уже имеющемся незадокументированном классе. Будем использовать классы моделей (Subject, StudyGroup, Enrollment, Lesson, Attendance)

## Комментирование кода по правилам Doxygen

Код был прокомментирован в соответствии с правилами Doxygen, применительно к языку Python.

### Листинг:

from django.db import models

from django.contrib.auth.models import User

import datetime as dt

***##***

***# @page models Модели***

***# @file models.py***

***#***

***# @brief В данном файле описаны Django модели, обеспечивающие интерфейс доступа к базе данных***

***#***

***# @section schedule\_model\_libs Используемые библиотеки***

***# - datetime***

***# - auth.models.User django model***

***#***

***#***

**class** **Subject**(*models*.*Model*):

    """! Дисциплина обучения

        Интерфейс доступа к объекту БД "Дисциплина обучения"

    """

***## Название дисциплины***

    name = models.CharField(

        verbose\_name = 'Название',

        max\_length = **255**,

    )

***## Дата создания объекта***

    created\_at = models.DateTimeField(verbose\_name = 'Дата и время создания', auto\_now\_add = True)

***## Дата последнего изменения объекта***

    last\_modified = models.DateTimeField(verbose\_name = 'Дата и время последнего изменения', auto\_now = True)

**class** **Meta**:

        """! Мета-информация для корректного отображения в панели администратора

        """

        verbose\_name = "Изучаемая дисциплина"

        verbose\_name\_plural = "Изучаемые дисциплины"

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

        return self.name

***##***

***# @brief Группа обучения.***

***#***

***# Интерфейс доступа к объекту БД "Группа обучения".***

***#***

**class** **StudyGroup**(*models*.*Model*):

***## FK "Изучаемая дисциплина"***

    subject = models.ForeignKey(

        Subject,

        verbose\_name = "Изучаемый предмет",

        on\_delete = models.**SET\_NULL**,

        null=True, blank=True,

    )

***## Название группы***

    name = models.CharField(

        verbose\_name = 'Название',

        max\_length = **255**,

    )

***## Уровень группы. Например: Новички, Средние, Старшие.***

    level = models.CharField(

        verbose\_name = "Уровень группы",

        max\_length = **255**,

        null=True, blank=True,

    )

***## Максимальное количество человек в группе.***

    capacity = models.IntegerField(

        verbose\_name = 'Вместимость',

    )

    created\_at = models.DateTimeField(verbose\_name = 'Дата и время создания', auto\_now\_add = True)

    last\_modified = models.DateTimeField(verbose\_name = 'Дата и время последнего изменения', auto\_now = True)

***## Количество мест доступных для записи в группу.***

***#***

***# @param self Ссылка на переданный объект.***

***#***

***# @throw aslfkjs;dlkjsad adkjflskdjf***

***#***

***# @return Количество мест.***

**@***property*

**def** **vacant\_count**(self):

        group\_enrollments = self.enrollment\_set.filter(deleted=False)

        busy\_count = len(group\_enrollments)

        return self.capacity - busy\_count

**@***property*

**def** **students\_count**(self):

        """! Количество студентов в группе.

        @return Количество студентов

        """

***## Список записей в группу***

        group\_enrollments = self.enrollment\_set.filter(deleted=False)

***## Количество записей в группу***

        busy\_count = len(group\_enrollments)

        return busy\_count

**class** **Meta**:

        """! Мета-информация.

        Мета-информация для корректного отображения в панели администратора

        """

        verbose\_name = "Группа учеников"

        verbose\_name\_plural = "Группы учеников"

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

        return **f**"**{**self.name**}** | **{**self.subject**}** | **{**self.level**}**"

**@***property*

**def** **max\_age**(self):

        """! Максимальный возраст студента в группе.

        @return Максимальный возраст.

        """

        enrollments = self.enrollment\_set.all()

        ages = [enrollment.student.userprofile.age for enrollment in enrollments]

        return max(ages) if ages else None

**@***property*

**def** **min\_age**(self):

        """! Минимальный возраст студента в группе

        @return Минимальный возраст

        """

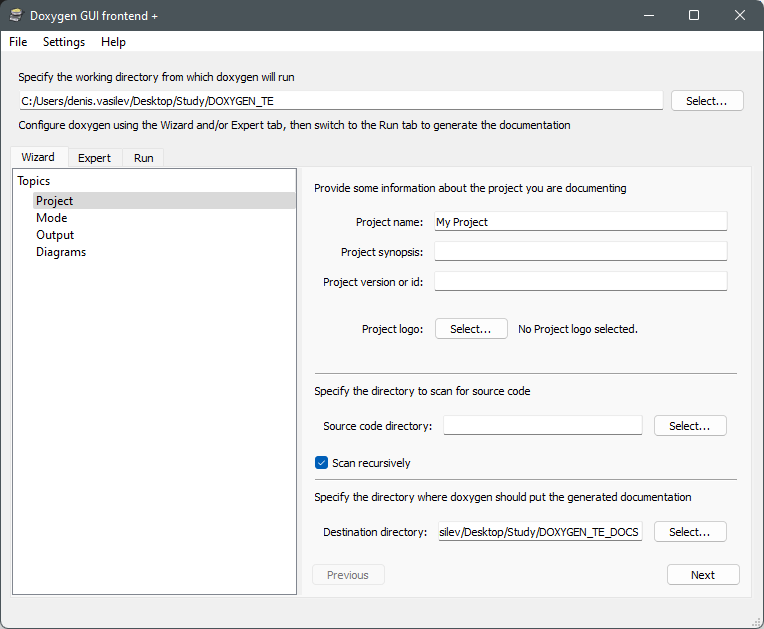
        enrollments = self.enrollment\_set.all()

        ages = [enrollment.student.userprofile.age for enrollment in enrollments]

        return min(ages) if ages else None

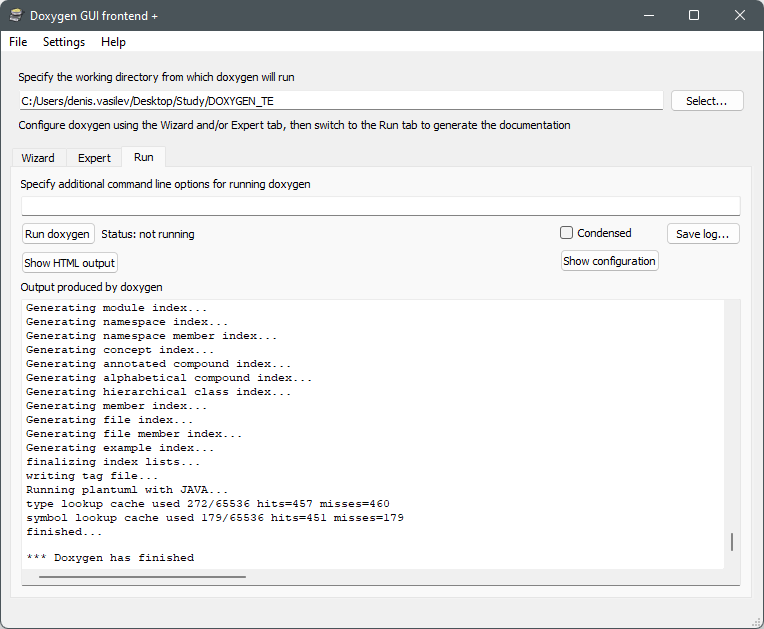
## Запуск и настройка Doxygen

Doxygen был запущен и настроен в соответствии с рекомендациями для проекта на языке Python.

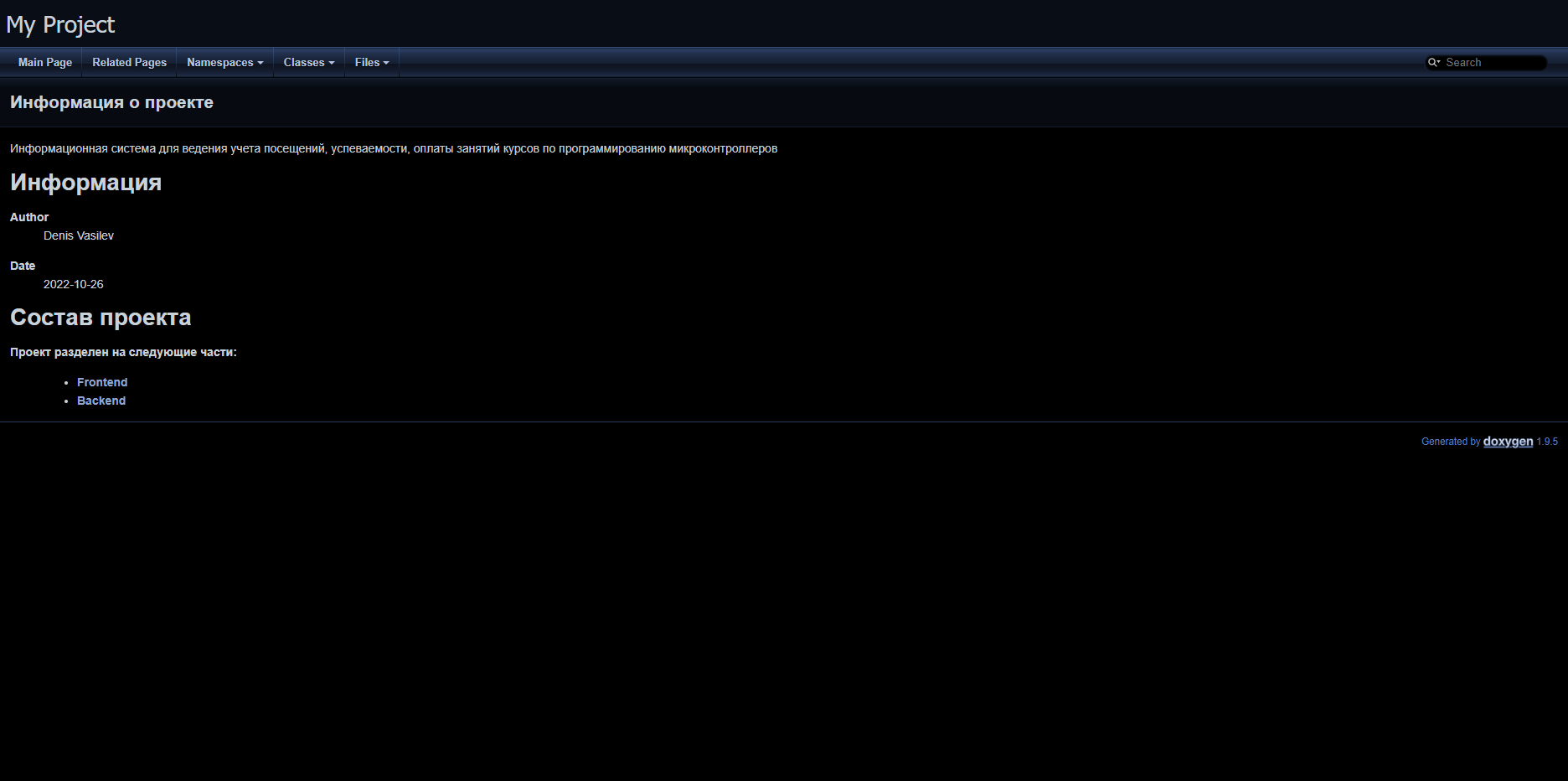


## Запуск документирования

Для запуска документирования на вкладке Run необходимо нажать кнопку “Run doxygen”.



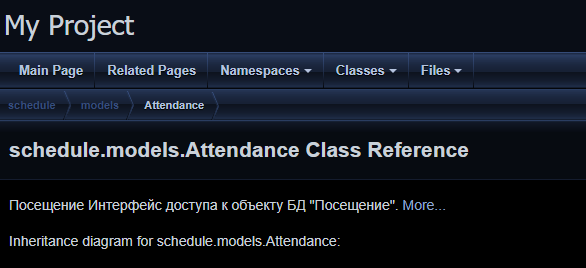
При нажатии на кнопку «Show HTML output» в браузере откроется просмотр результата документирования в формате HTML.



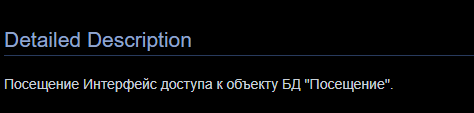
## Демонстрация

### Краткое и подробное описание класса

Краткое описание

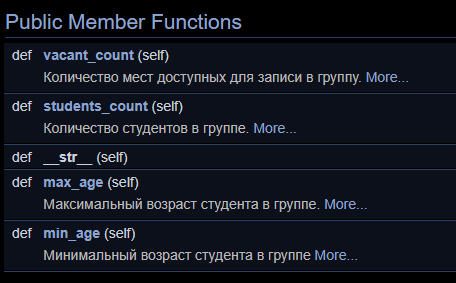


Подробное описание

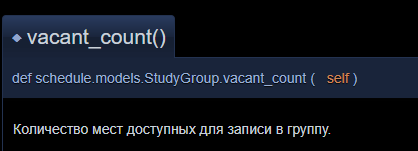


### Краткое и подробное описание функций класса

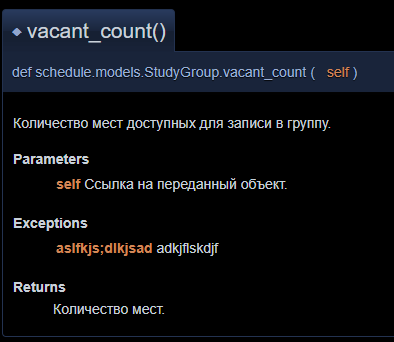
Краткое описание



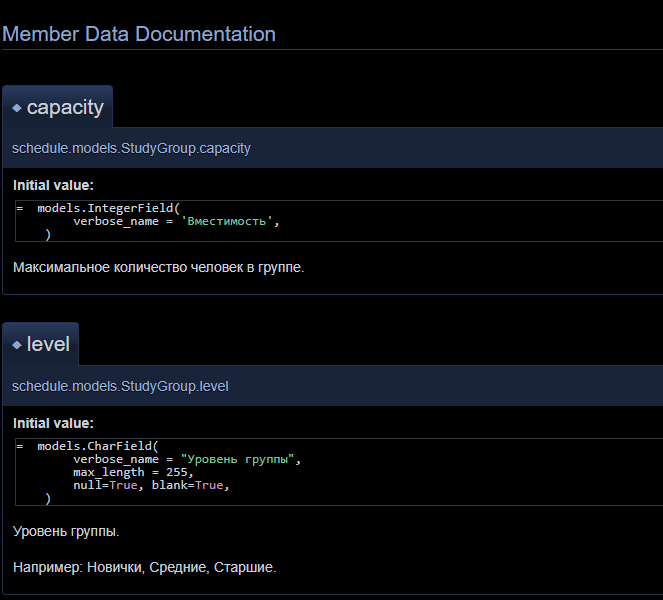
Подробное описание



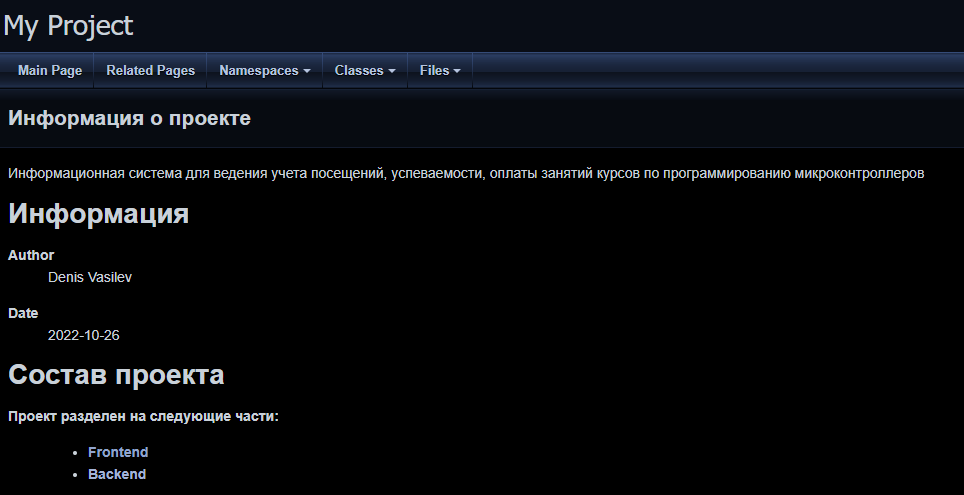
### Описание входных и выходных параметров функций



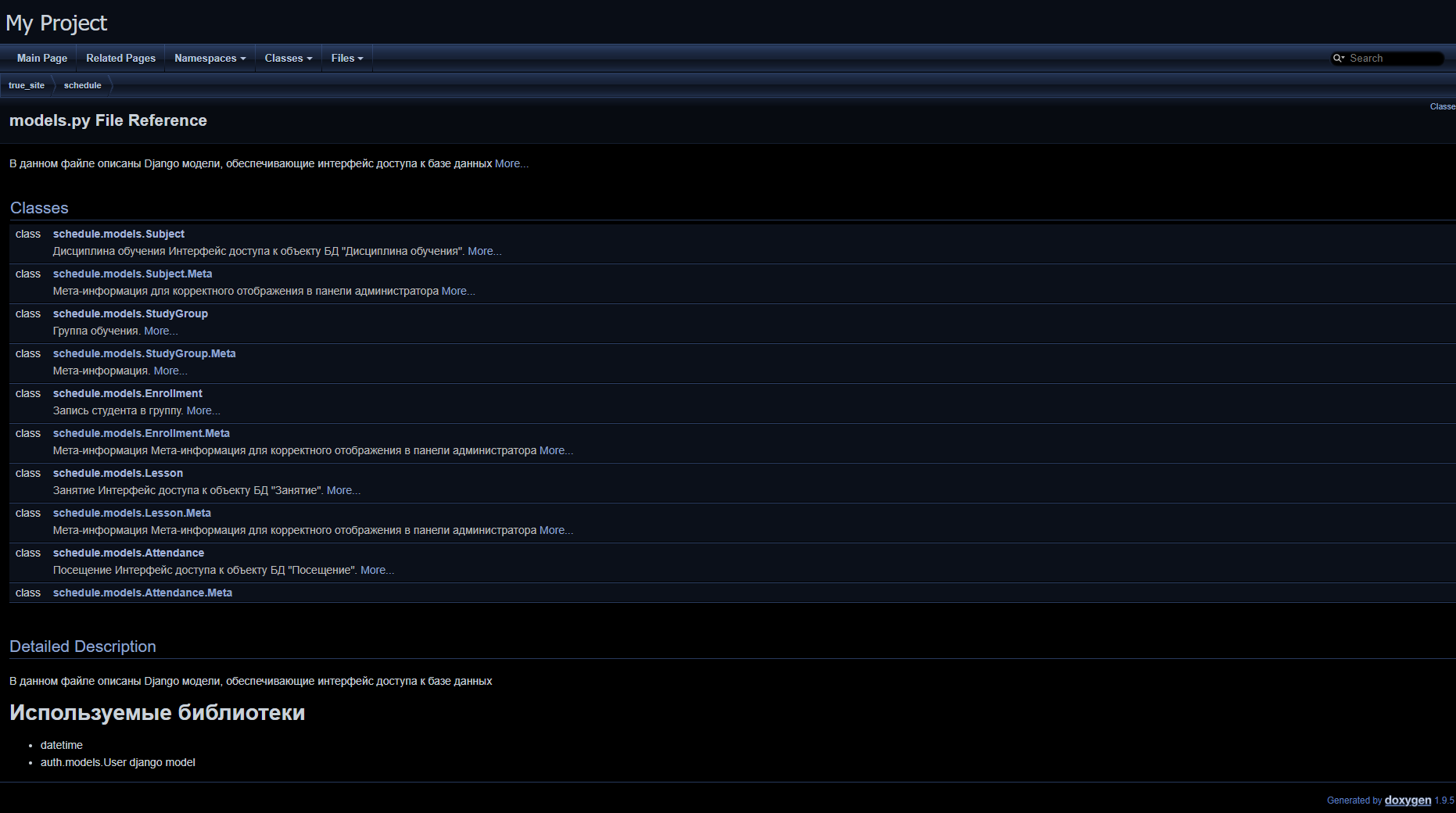
### Описание переменных класса



### Описание проекта



### Описание файла и добавление ссылки на файл в описание проекта



### Описание производного класса



## Вывод

В ходе выполнения работы были приобретены навыки документирования кода, а также освоено и использовано средство документирования Doxygen.