**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский политехнический университет»**

|  |  |
| --- | --- |
|  |  |
|  |  |

ОТЧЕТ:  
**«Лабораторная работа №4».**

По дисциплине:   
**«Программная инженерия»**

**Выполнил:** Архипов А. И.

**Проверил:** Будылина Е. А.

Москва, 2020

Объектно-ориентированный подход к моделированию бизнес- процессов. Модель классов и модель прецедентов.

Цель: Освоение методики анализа разрабатываемой программы; освоение задач формулирования функциональных и нефункциональных требований к программной реализации отдельных задач и к программе в целом; выработка навыков разработки технического задания.

**Выполнение задания:**

# НАЗНАЧЕНИЕ ПРОГРАММЫ И ЦЕЛИ ЕЕ СОЗДАНИЯ

Разрабатываемое ПО предоставит возможность развернуть облачную инфраструктуру для хранения электронных моделей изделий, чертежей и конструкторской документации. Существующие аналоги на рынке облачных технологий, такие как Autodesk A360 Cloud, Kenesto Cloud, предлагают размещать конструкторскую документацию на своих зарубежных серверах без возможности локального хранения.

Поскольку такая документация может представляет коммерческую или государственную тайну, то, по сведениями из базы данных угроз (БДУ) ФСТЭК России, ее размещение на иностранных облачных серверах представляет собой угрозу информационной безопасности для владельца этой документации. Наш отечественный продукт призван решить эту проблему и предложить размещение документов на серверах заказчика.

# ТРЕБОВАНИЯ К ЗАДАЧЕ «ЗАГРУЗКА МОДЕЛИ В ОБЛАКО»

Для загрузки модели на сервер, клиентская часть программы должна обеспечить пользователя графическим интерфейсом с полями для ввода информации о электронной модели (строковый тип и файловый), кнопкой для отправки модели на сервер. Серверная часть программы должна принять и обработать запрос клиента, создать объект класса модели, скопировать и поместить в базу данных, а файл разместить через файловый сервер. В ответ на запрос программа должна отправить либо ответ о успешной загрузке файлов, либо код ошибки, из-за которой загрузка файла не может быть завершена.

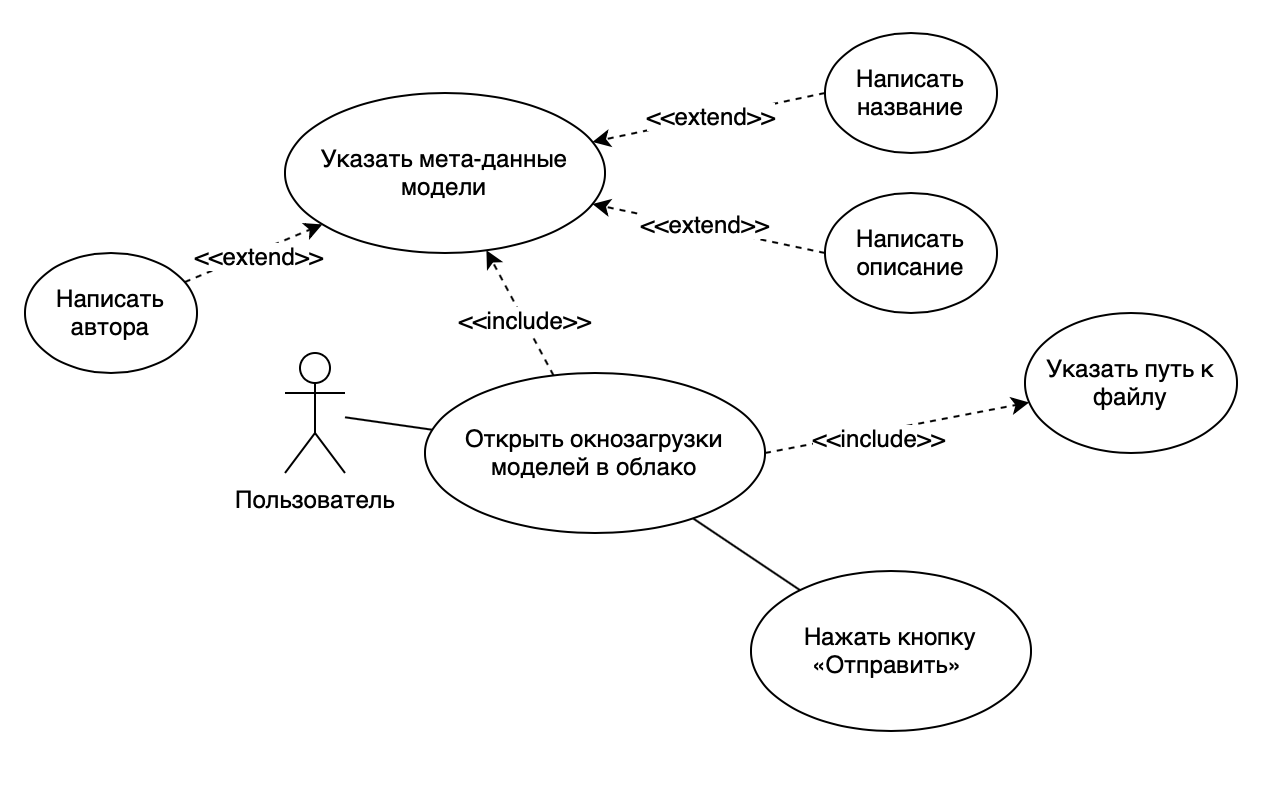


Рис. 1. Диаграмма вариантов использования UML

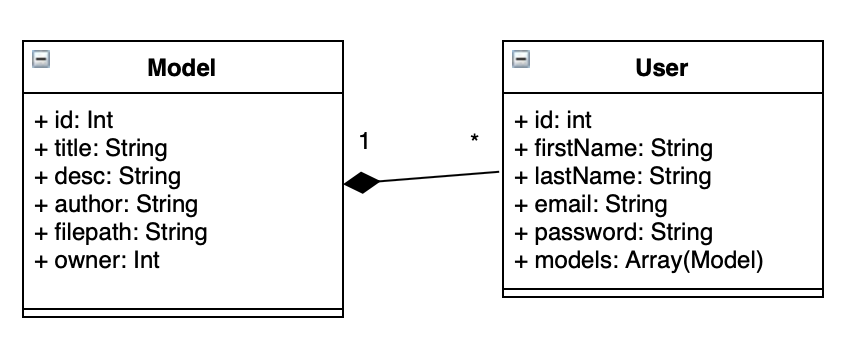


Рис. 2. Диаграмма классов UML

## Временной регламент реализации задачи

Время исполнения задачи программой неограниченно поскольку зависит от размера входящего файла модели, а также сетевого подключения и загруженности сетевого трафика как сети, так и компьютера пользователя.

## Требования к входным данным

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Ограничения |
| Название модели | Строка | До 255 символов |
| Описание модели | Многострочный текст | До 2000 символов |
| Файл модели | Файл | Формат: SAT, C3D, STEP, STP, IGS, IGES, JT, X\_T, X\_B, XMT\_TXT, XMP\_TXT, XMT\_BIN, XMP\_BIN, STP, STEP, STL, WRL, GRDECL, TXT, ASC, XYZ, C3D, GLTF |
| Автор (ФИО) | Строка | До 255 символов |

## Требования к выходным данным

В случае успеха, программа должна успешно закрыть подключение клиента с сервером и обновить страницу клиента. В случае ошибки, программа должна уведомить пользователя о возникших проблемах, таких как: отсутствие свободного места на ЖД, неверный формат файла, отсутствие прав для загрузки и т.д.

## Требования к преобразованию входной информации к машиночитаемого виду

Модель должна пройти обработку через конвертер для дальнейшего визуализации через браузер по технологии WebGL. Файл модели будет импортирован с помощью ядра C3D. Программа строт B-Rep модель исходного файла внутри памяти ЭВМ. Затем твердотельная модель претерпевает процессы триангуляции и создания триангуляционной сетки модели. В конце алгоритма программа выдает набор файлов модели, готовых к визуализации через браузер.

## Требования к программной реализации задач

Структура данных HTML-формы должна использовать тип содержимого **multipart/form-data**. Сообщение должно делиться на несколько частей. Клиентская часть программы должна внедрять в него поля, введенные пользователем (название модели - title, описание - desc, название файла - filename), и вложенный файл (поле file). Структура должна кодироваться помощью программного кода, реализованного через скрипт на языке JavaScript.

Сообщение должно передаваться по протоколу HTTP методом POST на серверную часть программы и обрабатываться с HTTP-сервером. Сервер должен быть реализован на платформе Node.js с использованием библиотеки Express. Ответ на запрос должен быть отправлен в формате JSON со следующей структурой: свойство code, которое обозначает код операции (успешно или не успешно), свойство message – причина возникновения ошибки на стороне, если есть.

## Функциональные требования

* Программа должна указать пользователю на недопустимый формат электронной модели (Допустимые форматы: SAT, C3D, STEP, STP, IGS, IGES, JT, X\_T, X\_B, XMT\_TXT, XMP\_TXT, XMT\_BIN, XMP\_BIN, STP, STEP, STL, WRL, GRDECL, TXT, ASC, XYZ, C3D, GLTF)
* Программа должна указать на поля, в случае их не заполнения пользователем и попросить их заполнить
* Программа должна проинформировать пользователя о том, что модель отправляется на сервер
* Программа должна проинформировать процент загруженной части модели на сервер
* Программа должна проинформировать пользователя о том, что модель успешно загружена

## Требования к прикладному программному обеспечению

Для стабильной работы ПО требуется ОС семейства Windows Server.