**ОСНОВНАЯ ЧАСТЬ**

**1 Исследование предметной области**

Предметной областью является деятельность кафедры МОСИТ, однако в рамках предмета исследования выделяются два основных процесса:

1. расчёт индивидуальной нагрузки преподавателей;
2. контроль исполнения поручений.

**1.1 Расчёт индивидуальной нагрузки**

Данный процесс включает в себя подпроцессы, такие как:

1. создание индивидуального плана;
2. корректировка индивидуального плана.

**1.1.1 Участники процесса**

Основными участниками процесса являются сотрудники кафедры (Таблица 1).

Таблица – Описание сотрудников

|  |  |
| --- | --- |
| **Сотрудник** | **Описание** |
| Преподаватель | Создаёт индивидуальный план |
| Методист | Корректирует индивидуальный план |

**1.1.2 Данные**

Основной информацией в рамках процесса является индивидуальный план.

Индивидуальный план представляет собой файл в формате электронной таблицы, состоящий из 6 листов.

На первом листе содержится информация о преподавателе:

1. институт;
2. кафедра;
3. фио;
4. должность;
5. учёная степень;
6. учёное звание.

На втором листе содержится список учебных работ за осенний семестр, более конкретно определяющийся как список дисциплин, которые ведёт преподаватель. Также на листе содержится информация о планируемых и фактических затратах часов на выполнение плана.

На третьем листе содержится список учебных работ за весенний семестр, более конкретно определяющийся как список дисциплин, которые ведёт преподаватель. Также на листе содержится информация о планируемых и фактических затратах часов на выполнение плана.

На четвёртом листе содержится сводная информация по учебной работе, в которой подсчитан итог за два семестра. Также на листе содержится список учебно-методических работ с указанием планируемых и фактических затрат часов на их выполнение.

На пятом листе содержится список научно-исследовательских работ и организационно-методических работ с указанием планируемых и фактических затрат часов на их выполнение. Также на листе содержится сводка по общей годовой нагрузке.

На шестом листе содержится приложение к отчёту о работе преподавателя.

В рамках процесса «Создание индивидуального плана» входными данными являются:

1. информация о преподавателе;
2. информация о дисциплинах;
3. информация о группах;

Основанием для создания индивидуального плана является документ о нормах по планированию и учёту труда профессорско-преподавательского состава.

В качестве выходных данных, результатом процесса является индивидуальный план, представляющий собой файл в формате электронной таблицы (Рисунок 1).

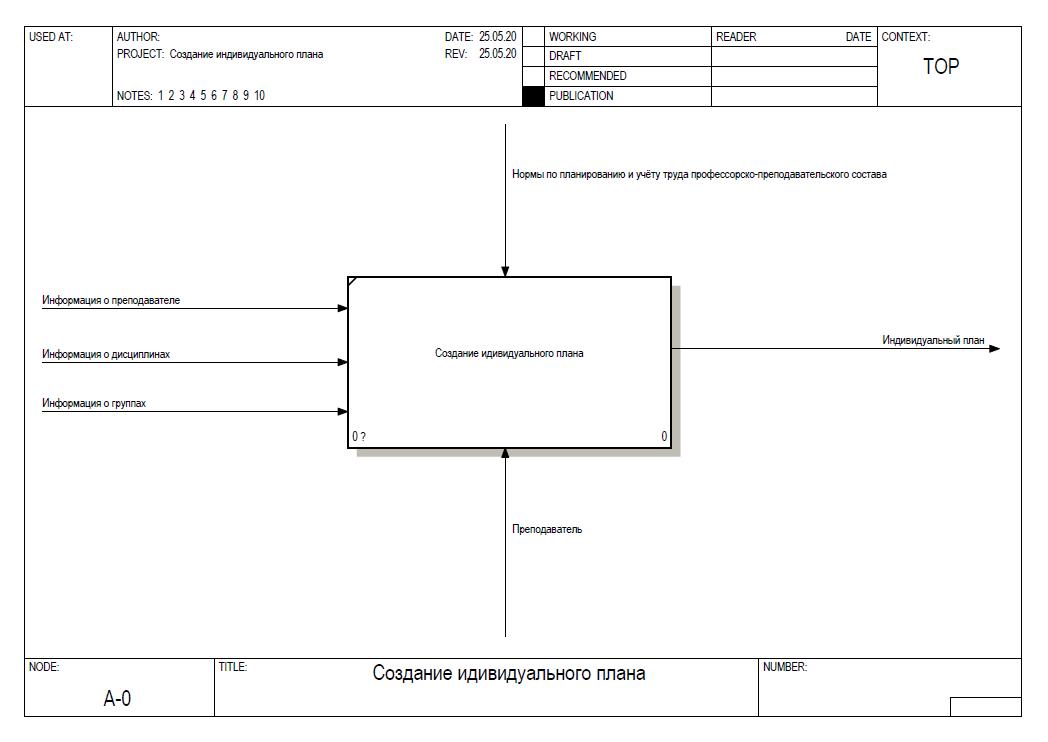


Рисунок – Подпроцесс «Создание индивидуального плана»

В рамках процесса «Корректировка индивидуального плана» основными данными является индивидуальный план преподавателя.

В качестве выходных данных, результатом процесса является скорректированный индивидуальный план (Рисунок 2).

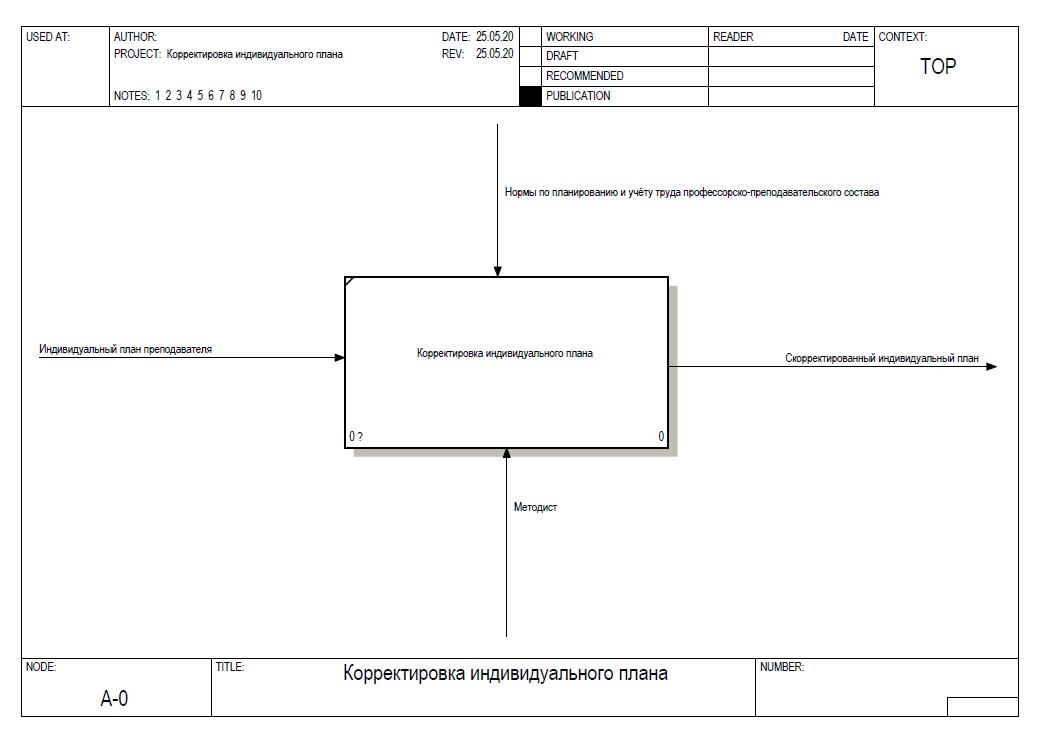


Рисунок – Подпроцесс «Корректировка индивидуального плана»

Стоить отметить, что корректировка индивидуального плана осуществляется вручную, с заполнением 4-го и 5-го листов.

**1.2 Контроль исполнения поручений**

Данный процесс включает в себя подпроцессы:

1. назначение поручения;
2. исполнение поручения.

**1.2.1 Участники процесса**

Основными участниками процессов являются сотрудники кафедры (Таблица 2).

Таблица – Описание сотрудников

|  |  |
| --- | --- |
| **Сотрудник** | **Описание** |
| Преподаватель | Исполняет назначенные поручения |
| Методист |
| Зам. по учебной работе | Исполняет назначенные поручения и также назначает поручения своим подчинённым |
| Зам. по научной работе |
| Зам. по учебно-методической работе |
| Ответственный за МТО |
| Ответственный по работе со студентами |
| Учёный секретарь |
| Заведующий кафедры |

**1.2.2 Данные**

Основной информацией в рамках процесса «Контроль поручений» является поручение.

Поручение представляет собой задачу или встречу, на исполнение которой назначаются подчинённые сотрудники.

Также поручение содержит следующие сведения:

1. постановку задачи или цель встречи;
2. описание;
3. документы;
4. срок исполнения.

Стоит отметить, что назначать поручения может только начальник в распоряжении которого находятся подчинённые сотрудники.

В рамках процесса «Создание поручения» входными данными являются:

1. информация о сотруднике или сотрудниках, которому или которым назначается поручение;
2. постановка задачи или цель встречи;
3. описание;
4. прикрепляемые документы;
5. сроки исполнения.

В качестве выходных данных, результатом процесса является назначенное поручение (Рисунок 3).

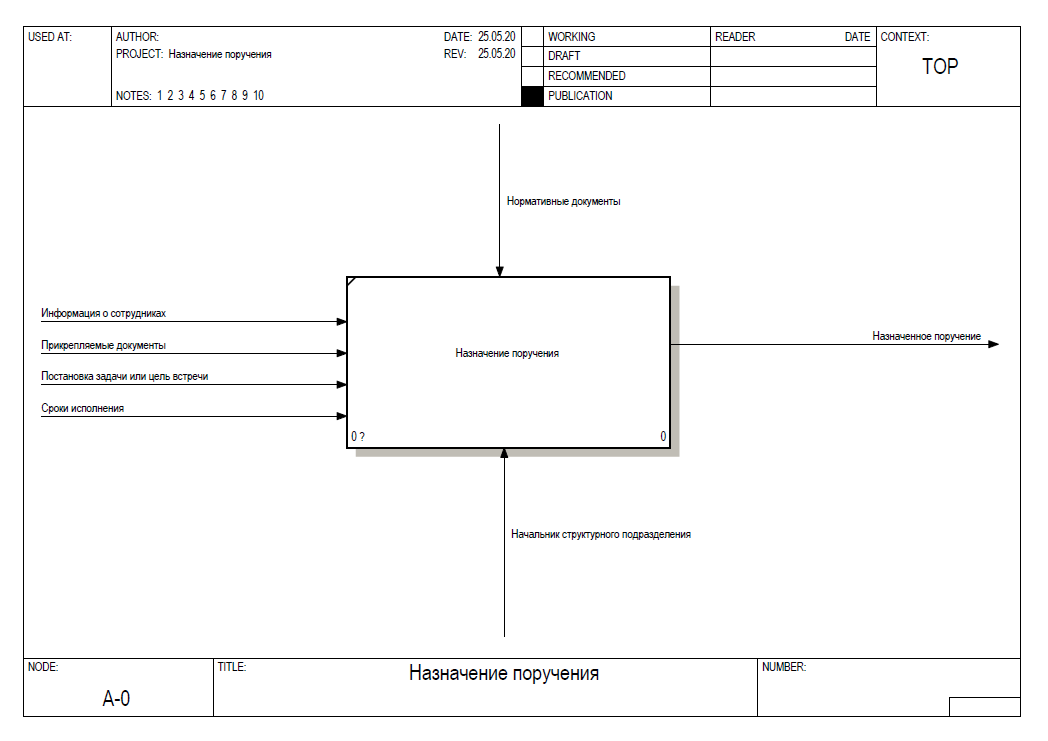


Рисунок – Подпроцесс «Назначение поручения»

В рамках процесса «Исполнение поручения» входными данными является назначенное поручение.

Выходными данными процесса является исполненное поручение (Рисунок 4).

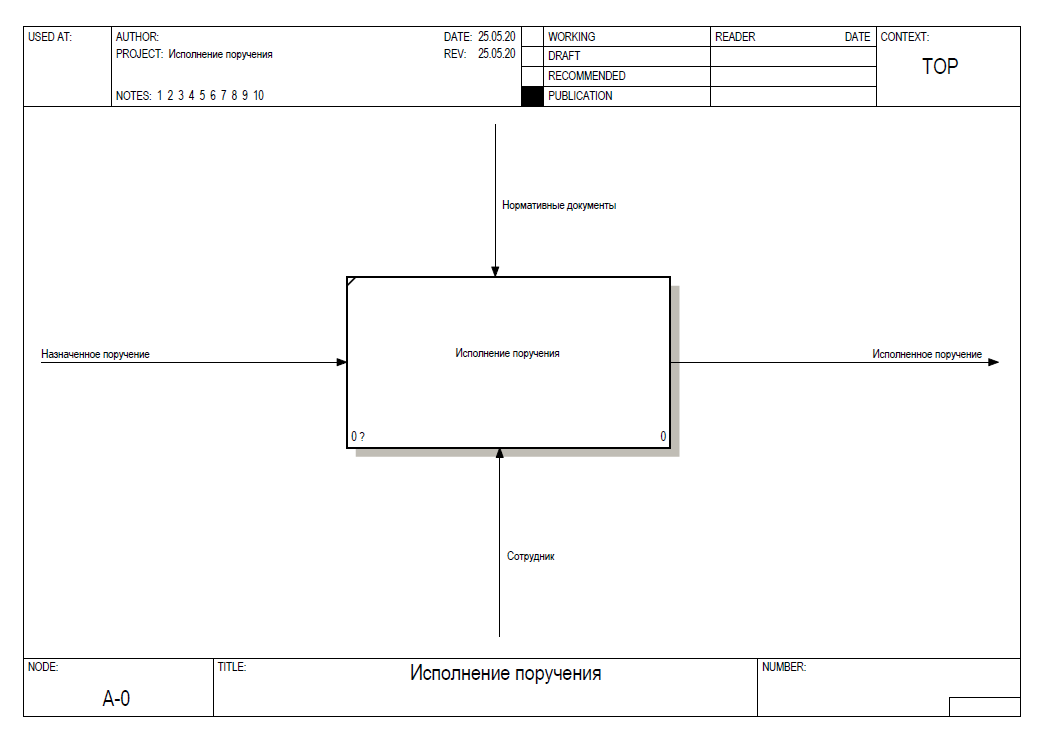


Рисунок – Подпроцесс «Исполнение поручения»

**1.3 Анализ существующих решений**

**1.3.1 Тандем.Университет**

На данный момент для автоматизации процесса расчёта индивидуальной нагрузки преподавателей используется программное решение Тандем.Университет – Модуль «Нагрузка» (Далее Тандем).

По представленной на официальном сайте информации, Тандем имеет следующие преимущества:

1) Технические:

* web-ориентированность;
* модульность решения и возможность модульного внедрения;
* возможность выбора опций в рамках модулей;
* открытая J2EE платформа;
* простота обновлений АРМов — не требуются усилия со стороны конечных пользователей;
* надежность и масштабируемость решения;
* кроссплатформенность;
* быстрота развертывания решения;
* встроенный модификатор скриптов и печатных шаблонов документов;
* покрытие данных системы веб-сервисами и view для построения отчетов внешними средствами, либо с помощью интегрированной Pentaho.

2) Экономические:

* нет необходимости приобретения стороннего программного обеспечения, возможность работы пользователей с системой на свободном программном обеспечении (Open source);
* как правило, не требуется приобретение дорогостоящего серверного оборудования;
* лицензируется инсталляция, а не рабочие места.

3) Развитие:

* поставка с открытыми исходными кодами и правом на доработку решения при условии не распространения третьим лицам;
* широкие интеграционные возможности со сторонними программными продуктами.

**1.3.1.1 Функциональные возможности**

Основными функциями Тандема являются:

1. получение и корректировка сводки планируемого контингента студентов;
2. получение и корректировка планируемых учебных групп (потоков, подгрупп);
3. получение и печать расчета нагрузки на читающих подразделениях (кафедрах);
4. формирование штата, вакансий на кафедрах;
5. распределение строк (часов) нагрузки между преподавателями (вакансиями) на кафедрах, передача часов между кафедрами;
6. контроль объема нагрузки на ставку, должность;
7. коррекция расчета нагрузки, контроль изменений, актуализация расчета с учетом изменений в фактическом контингенте студентов, траекториях обучения, нормах времени, и других факторов;
8. фиксация внеучебной нагрузки в индивидуальных планах преподавателей;
9. получение и печать индивидуальных планов преподавателей, печать учебных поручений;
10. получение фактических учебных групп (групп, подгрупп, потоков) по данным нагрузки и контингента студентов в начале учебного года (семестра);
11. гибкое управление нормами времени и их применением: простые нормы, нормы-формулы, нормы-скрипты;
12. возможность отслеживания показателей (при использовании дополнительных модулей системы) для сравнения планируемой учебной нагрузки с ее фактическим выполнением (по преподавателям, кафедрам, образовательной организации).

**1.3.1.2 Архитектура**

Тандем имеет следующую архитектуру (Рисунок 5).

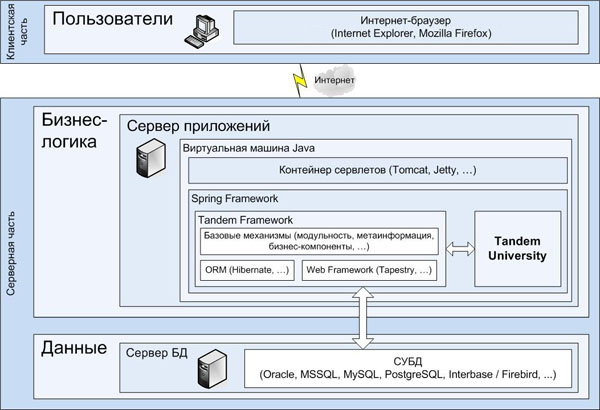


Рисунок – Архитектура Тандема

Архитектура Тандема представляет собой модульную структуру, которая позволяет расширять функционал.

Из рисунка видно, что на базе Spring Framework был разработан Tandem Framework. И на базе него идёт разработана система Tandem University.

Стоит также отметить, что данные могут храниться в различных СУБД, в зависимости от той, которая по умолчанию была предусмотрена в учебном заведении.

**1.3.1.3 Заключение**

Не смотря на то, что основным преимуществом Тандема является открытость к расширению функционала после её приобретения, в свою очередь для того, чтобы внести некоторые изменения в функционал необходимо будет сначала обучить специалиста или студента. Это может вызывать определённые трудности со стороны учебного заведения.

Также текущая функциональность связанная с расчётом нагрузки не позволяет автоматически рассчитать нагрузку для учебно-методической, научно-исследовательской и организационно-методической работ (которые представлены на 4 и 5 листах индивидуального плана).

В дальнейшем при разработке собственной системы следует учитывать возникающие недостатки в работе Тандема и не допускать их в разрабатываемой системе.

**1.3.2 SharePoint**

На данный момент для автоматизации процесса «Контроля поручений» используется SharePoint. SharePoint представляет собой набор программных продуктов предназначенные для обеспечения документооборота и организации интранет сети.

**1.3.2.1 Функциональные возможности**

**1.3.2.2 Архитектура**

**1.3.2.3 Заключение**

**2 Формирование требований к системе**

**2.1 Общие требования**

Система должна предоставлять функции для хранения материалов кафедры.

Система должна предоставлять функции для создания, контроля и исполнения поручений.

Система должна предоставлять функции для работы с индивидуальными планами.

Система должна представлять собой веб-приложения. Исходя из этого следует учесть поддержку браузеров: Chrome, Firefox и Opera.

**2.2 Требования к функциям**

Разделим предоставляемые функции по пользователям.

Для всех типов пользователей предоставляются следующие функции:

1. хранение материалов;
2. работа с поручениями;
3. просмотр информации профиля;
4. смена пароля.

В свою очередь, хранение материалов включает в себя:

1. загрузку файлов;
2. удаление загруженных файлов;
3. скачивание загруженных файлов;
4. просмотр файлов и папок;
5. создание папок;
6. удаление папок.

В свою очередь, работа с поручениями включает в себя:

1. просмотр назначенных поручений;
2. комментирование поручения;
3. прикрепление поручения;
4. пометка поручения как выполненного.

Прочие функциональные возможности приведены в таблице .

Таблица – Функциональные возможности пользователей

|  |  |
| --- | --- |
| **Пользователь** | **Функциональная возможность** |
| Преподаватель | Просмотр индивидуальных планов |
| Редактирование индивидуальных планов |
| Скачивание индивидуальных планов |
| Методист | Добавление индивидуальных планов |
|  | Изменение индивидуальных планов |
|  | Удаление индивидуальных планов |
| Зам. по учебной работе | Создание поручений, изменение поручений, удаление поручений |
| Зам. по научной работе |
| Зам. по учебно-методической работе |
| Ответственный за МТО |
| Ответственный за работу со студентами |
| Учёный секретарь |
| Зав. кафедрой |

**2.3 Требования к надёжности**

Система должна восстанавливать свою работоспособность (в случае аппаратных сбоев) после корректного перезапуска аппаратных средств.

Также система должна выдавать сообщения пользователю в случае совершения некорректных действий, таких как: ввод некорректных данных, возникновение исключительных ситуаций в самой системе.

**2.4 Требования к безопасности**

Прямой доступ к конфиденциальной информации должен быть исключён.

Доступ к базе данных может быть осуществлён только системным администратором.

Наличие встроенной панели администратора не подразумевается.

**3 Проектирование системы**

**3.1 Проектирование архитектуры**

Архитектура системы является клиент-серверной.

На сервере располагается база данных и само веб-приложение.

На клиенте осуществляется доступ к веб-приложению посредством браузера.

Следует разделить функциональные возможности на основные и сервисные подсистемы.

К основным подсистемам относятся:

1. подсистема индивидуальных планов;
2. подсистема кадров;
3. подсистема поручений;
4. подсистема хранения материалов.

К сервисным подсистемам относятся:

1. подсистема пользователей;
2. подсистема прав доступа.

**3.2 Проектирование базы данных**

Для каждой из подсистем представлены следующие таблицы, на рисунках .

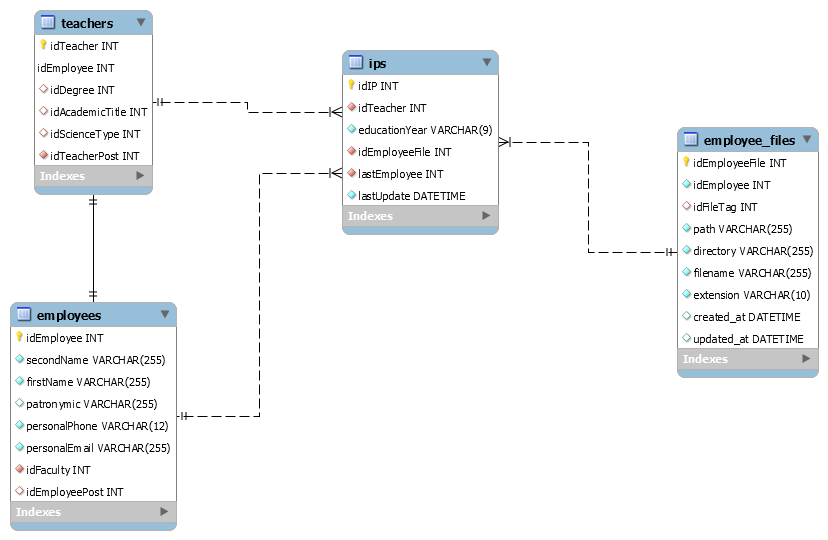


Рисунок – Таблицы подсистемы индивидуальных планов

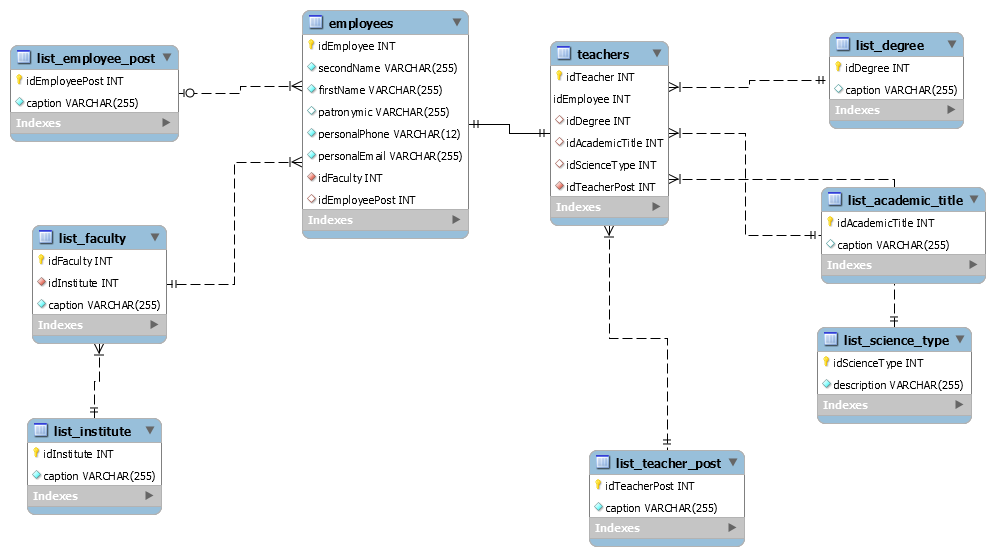


Рисунок – Таблицы подсистемы кадров

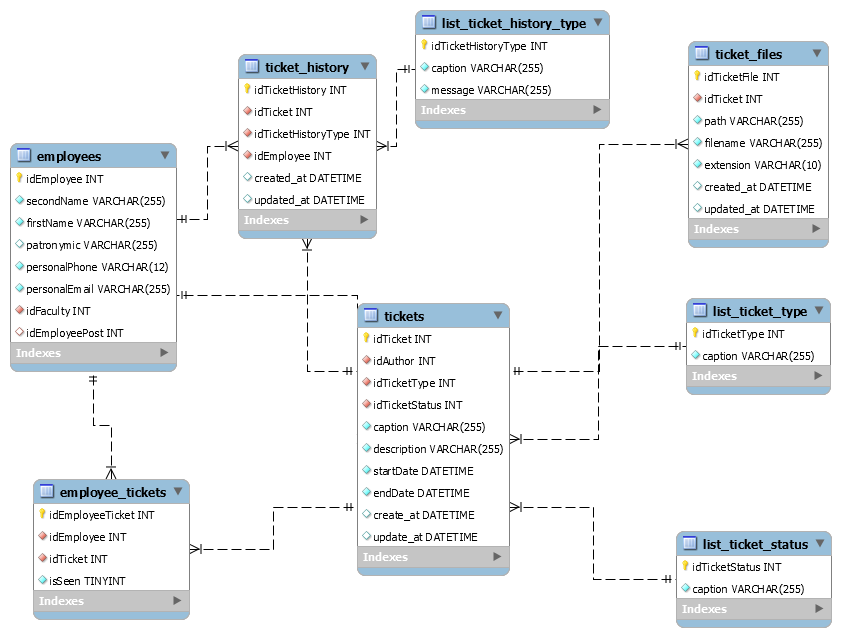


Рисунок – Таблицы подсистемы поручений

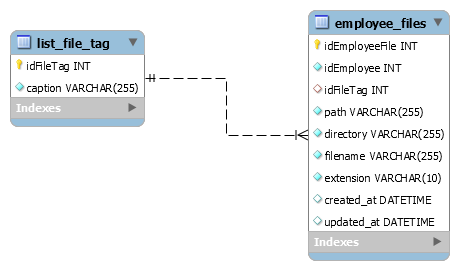


Рисунок – Таблицы подсистемы хранения материалов

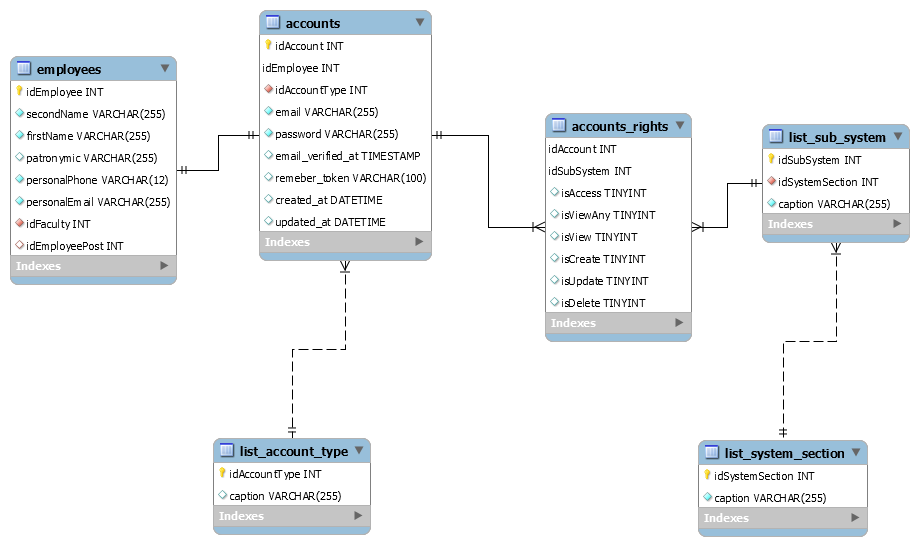


Рисунок – Таблицы подсистемы пользователей и прав доступа

**3.2 Проектирование классов**

**3.2.1 Модели**

Для работы с таблицами непосредственно из кода, необходимо спроектировать так называемые модели, которые повсеместно используются во фреймворке.

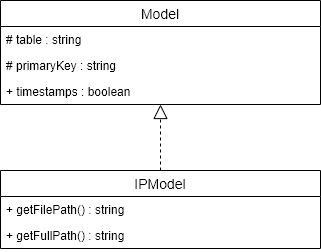


Рисунок – Модели подсистемы индивидуальных планов

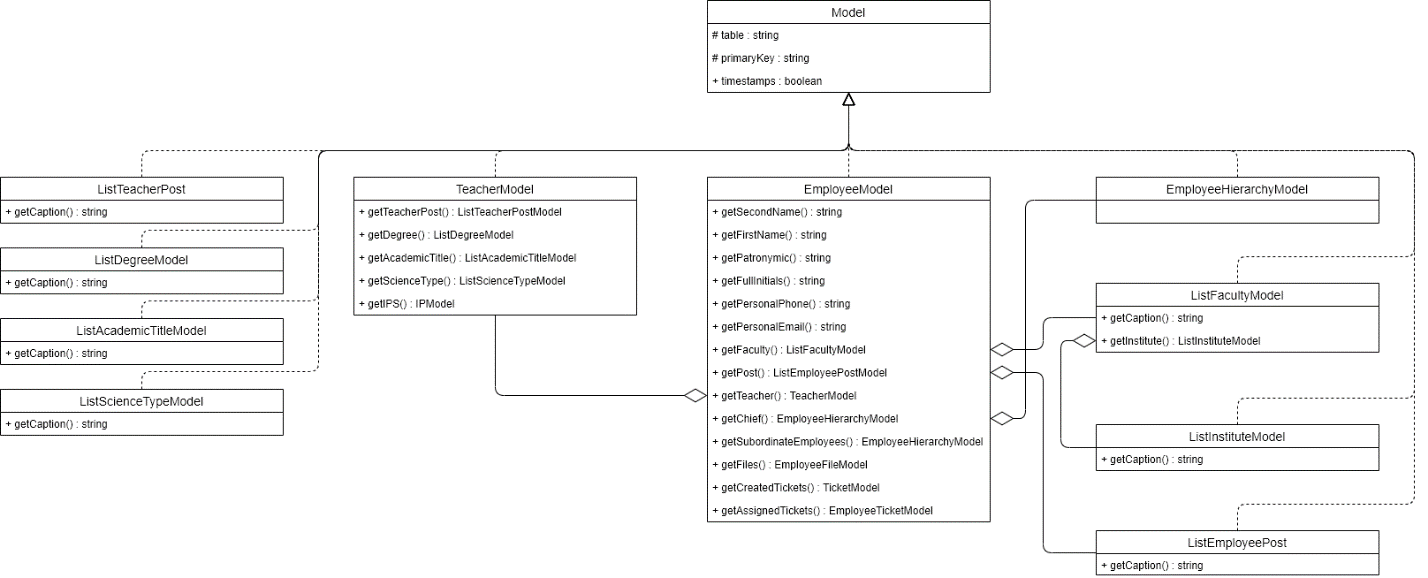


Рисунок – Модели подсистемы кадров

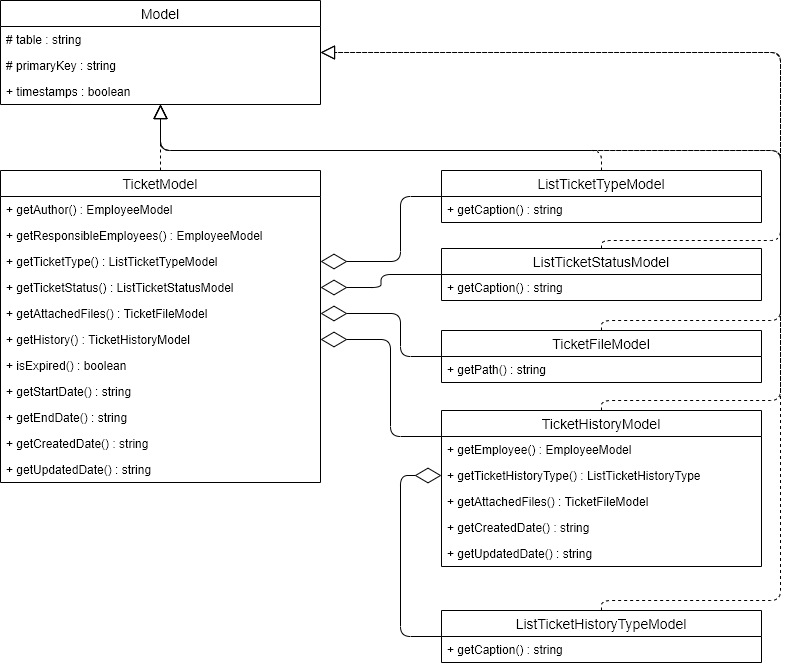


Рисунок – Модели подсистемы поручений

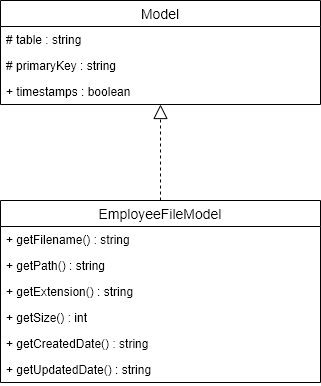


Рисунок – Модели подсистемы хранения материалов

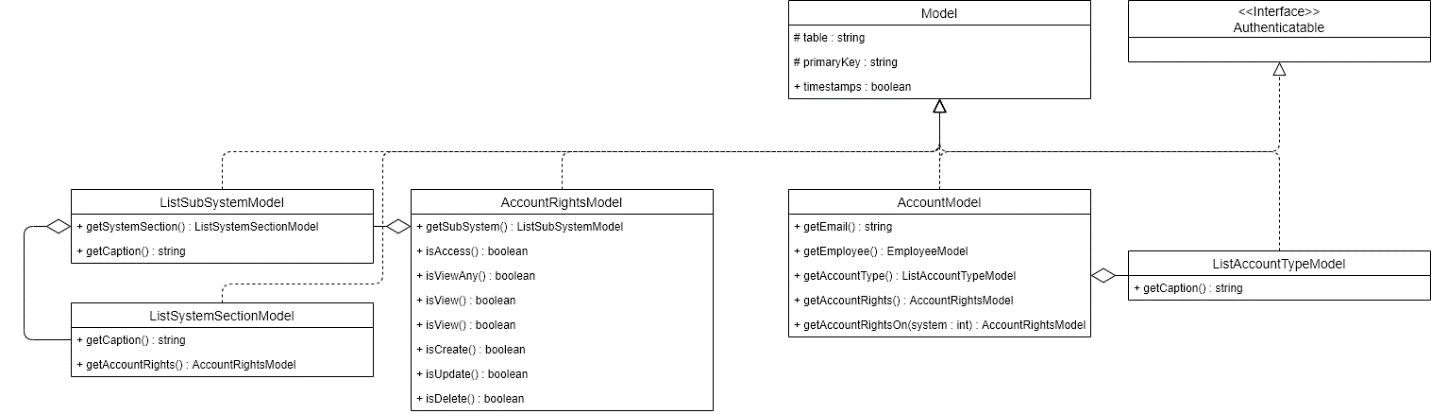


Рисунок – Модели подсистемы пользователей и прав доступа

**3.2.2 Контроллеры**

Для реализации функционала различных подсистем необходимо спроектировать контроллеры.

Стоит разделить контроллеры на страничные и ресурсные. Страничные контроллеры отвечают за то, чтобы возвращать представления для различных страниц. Ресурсные контроллеры отвечают за реализацию CRUD для конкретной модели.

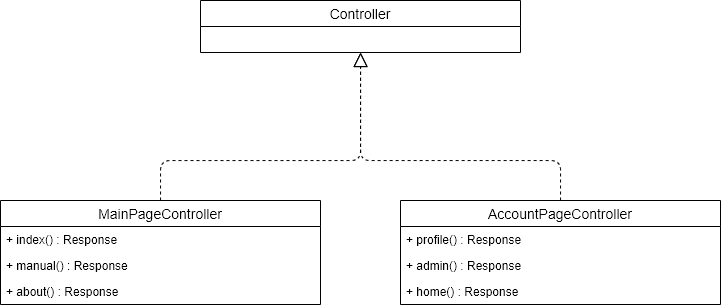


Рисунок – Контроллеры

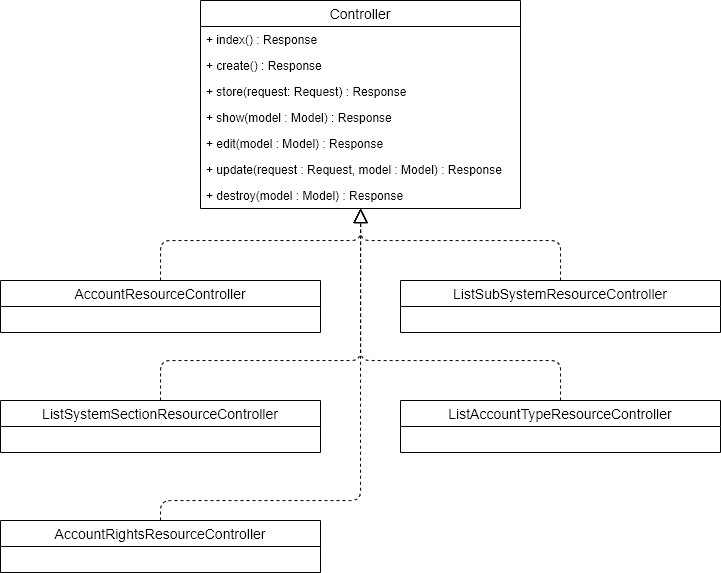


Рисунок – Ресурсные контроллеры подсистемы пользователей

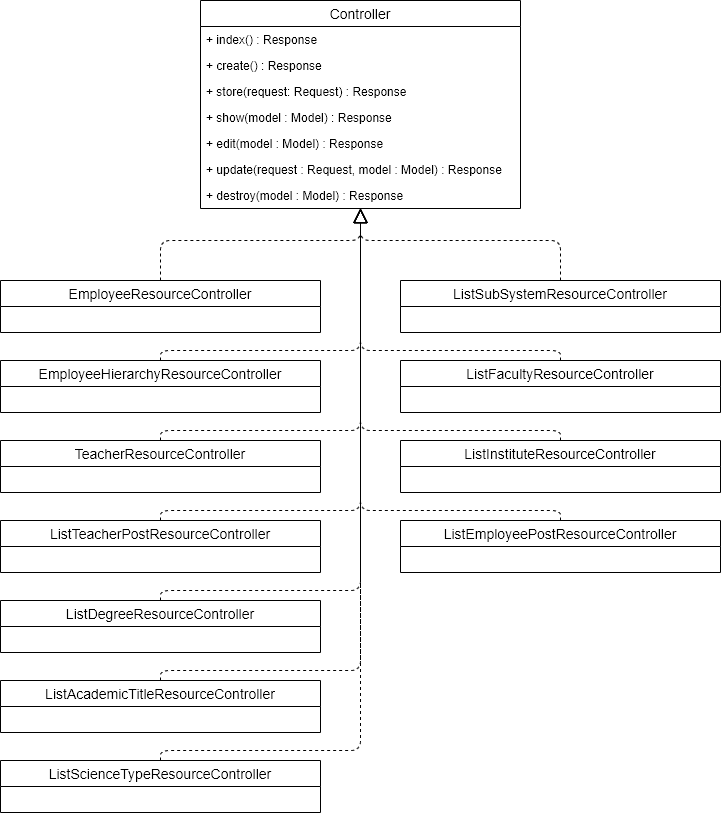


Рисунок – Ресурсные контроллеры подсистемы кадров

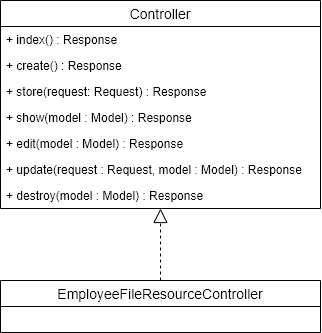


Рисунок – Контроллер подсистемы хранения материалов

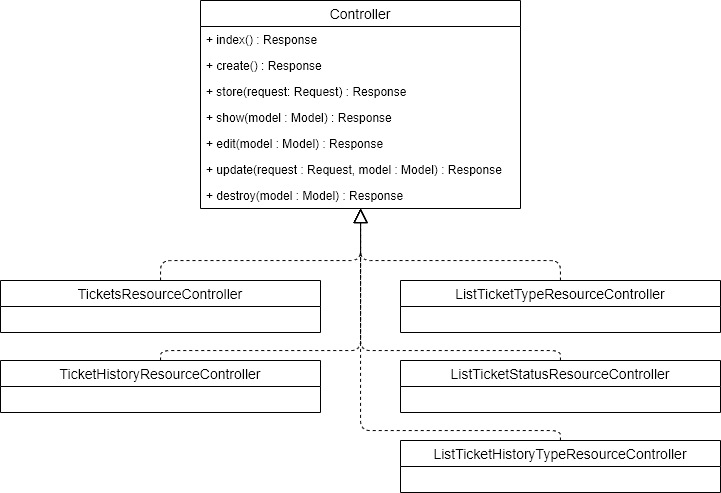


Рисунок – Контроллеры подсистемы поручений

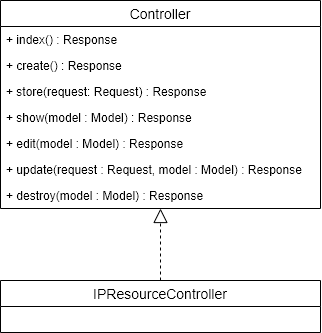


Рисунок – Контроллер подсистемы индивидуальных планов

**4 Реализация системы**

**4.1 Маршруты**

Логика доступа к веб-страницам осуществляется посредством HTTP-маршрутов, которые описываются в файле web.php. Маршруты приведены на рисунке 22.



Рисунок – Маршруты приложения

**4.2 Миграции**

Laravel позволяет создавать файлы миграций, описывающие таблицы. В свою очередь это позволяет на этапе развёртывания сразу создавать базу данных, а также заполнять её данными.

Пример миграции приведён на рисунке 23.



Рисунок – Пример миграции таблицы «Accounts»

**4.3 Сеялки**

Laravel позволяет создавать, так называемые, сеялки, которые отвечают за наполнение базу данных информацией.

Пример сеялки приведён на рисунке 24.



Рисунок – Пример сеялки

**4.4 Модели**

Laravel активно использует паттерн MVC, в связи с этим создание моделей необходимо.

Пример модели представлен на рисунке 25.



Рисунок – Пример модели «AccountModel»

**4.5 Контроллеры**

Создание контроллеров необходимо для поддержки MVC-паттерна. В связи с этим для каждой модели создаётся ресурсы контроллер, реализующий для неё CRUD.

Пример контроллера представлен на рисунке 26.



Рисунок – Пример контроллера «AccountResourceController»

**4.6 Пользовательский интерфейс**

Laravel использует для создания представлений или попросту страниц, шаблонизатор Blade.

В соответствии с этим, все страницы создаются как файлы с расширением blade.php и содержат HTML-вёрстку, а также специальные директивы для шаблонизатора.

Пример кода в шаблона представлен на рисунке 27.



Рисунок – Пример кода blade-шаблона

Пользовательский интерфейс представлен на рисунках 28-35.

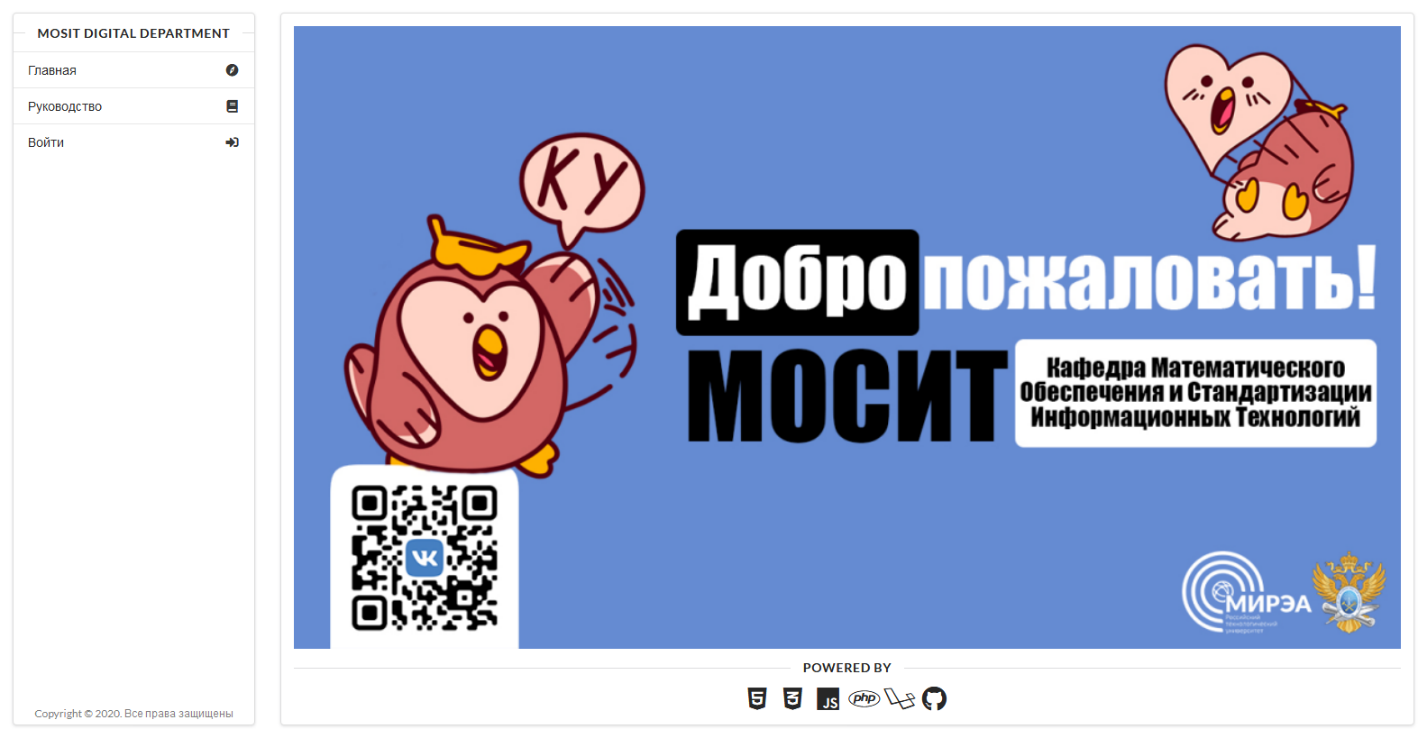


Рисунок – Главная страница



Рисунок – Рабочая область

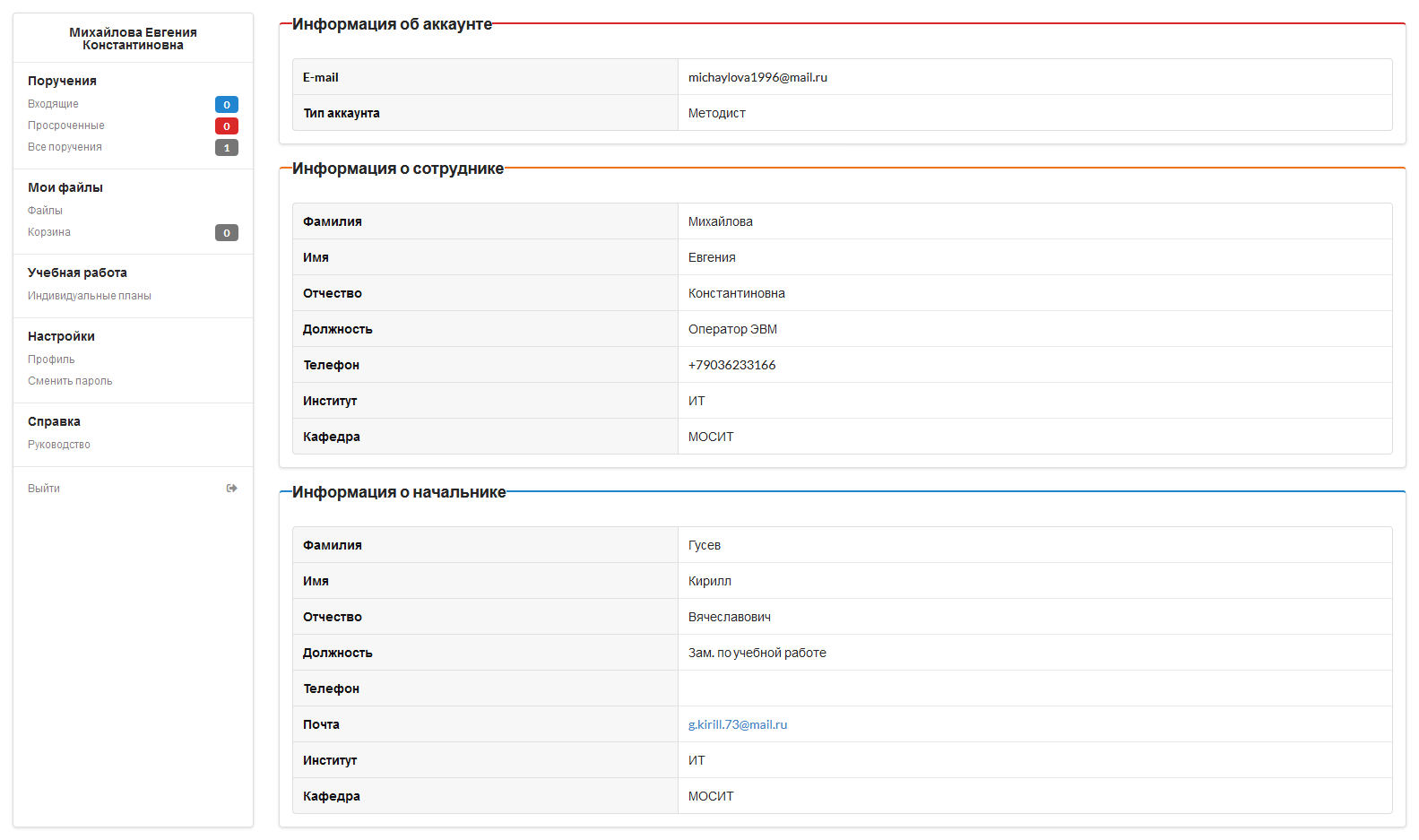


Рисунок – Профиль пользователя

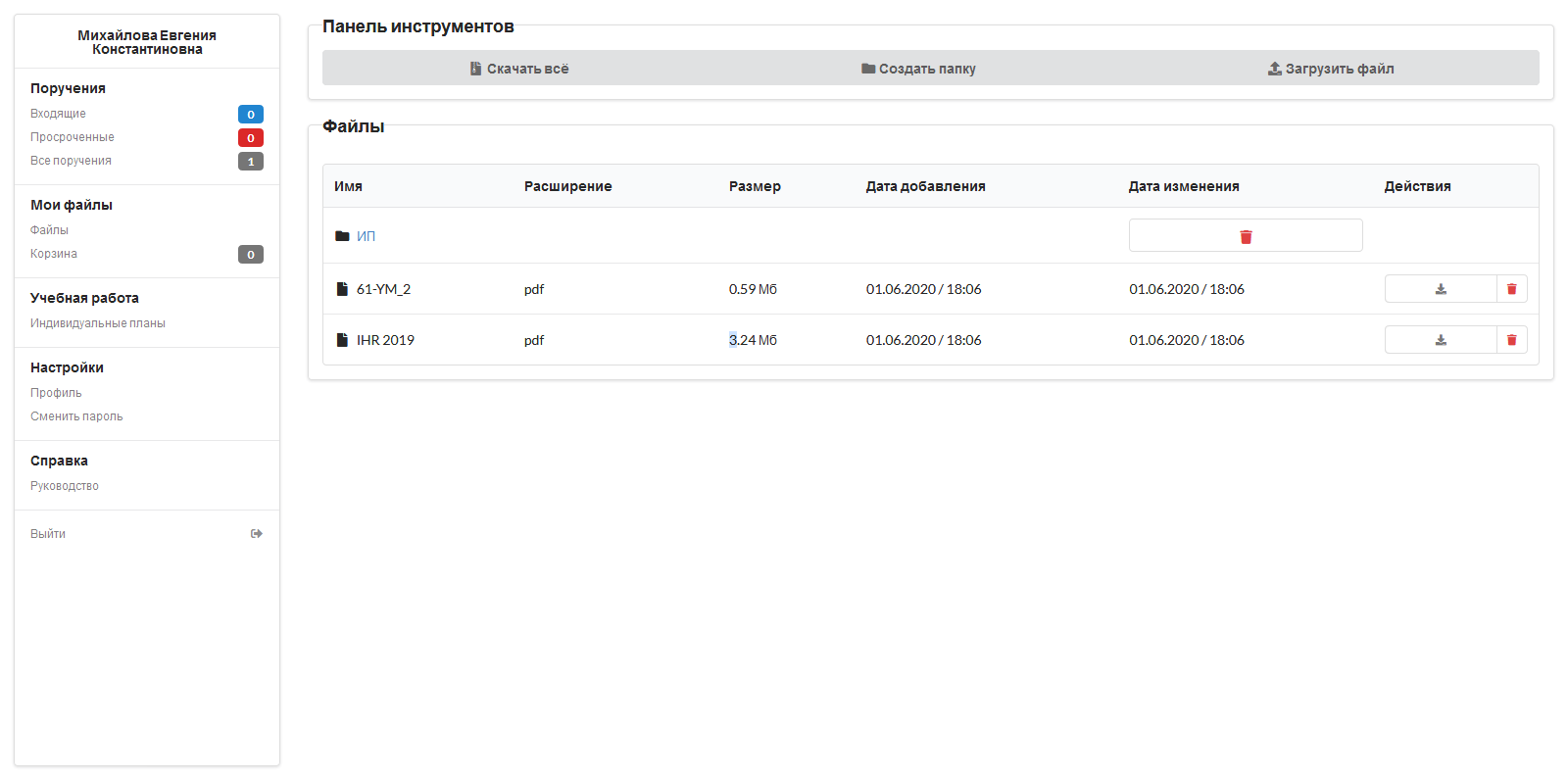


Рисунок – Мои файлы

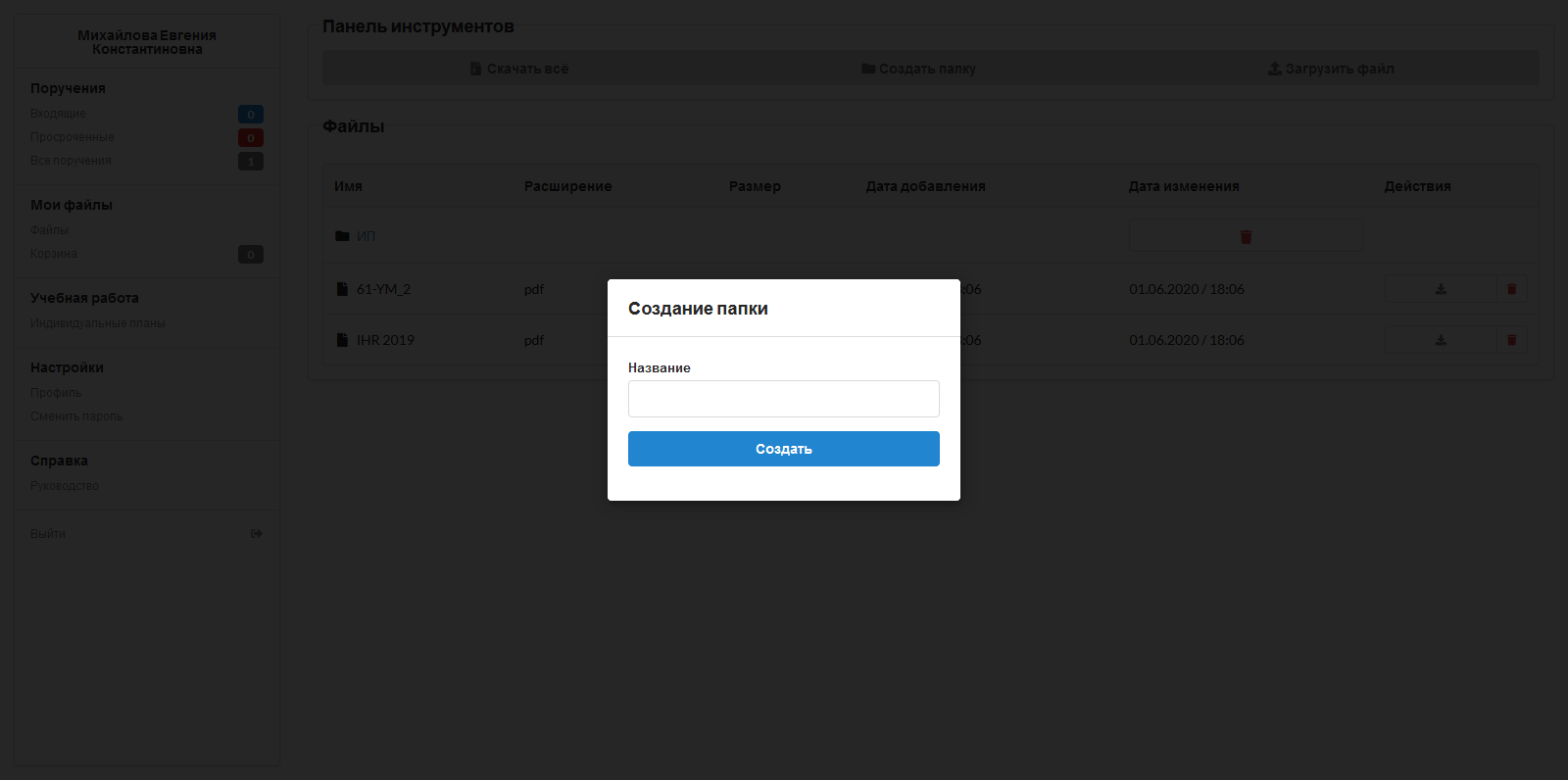


Рисунок – Модальное окно «Создание папки»

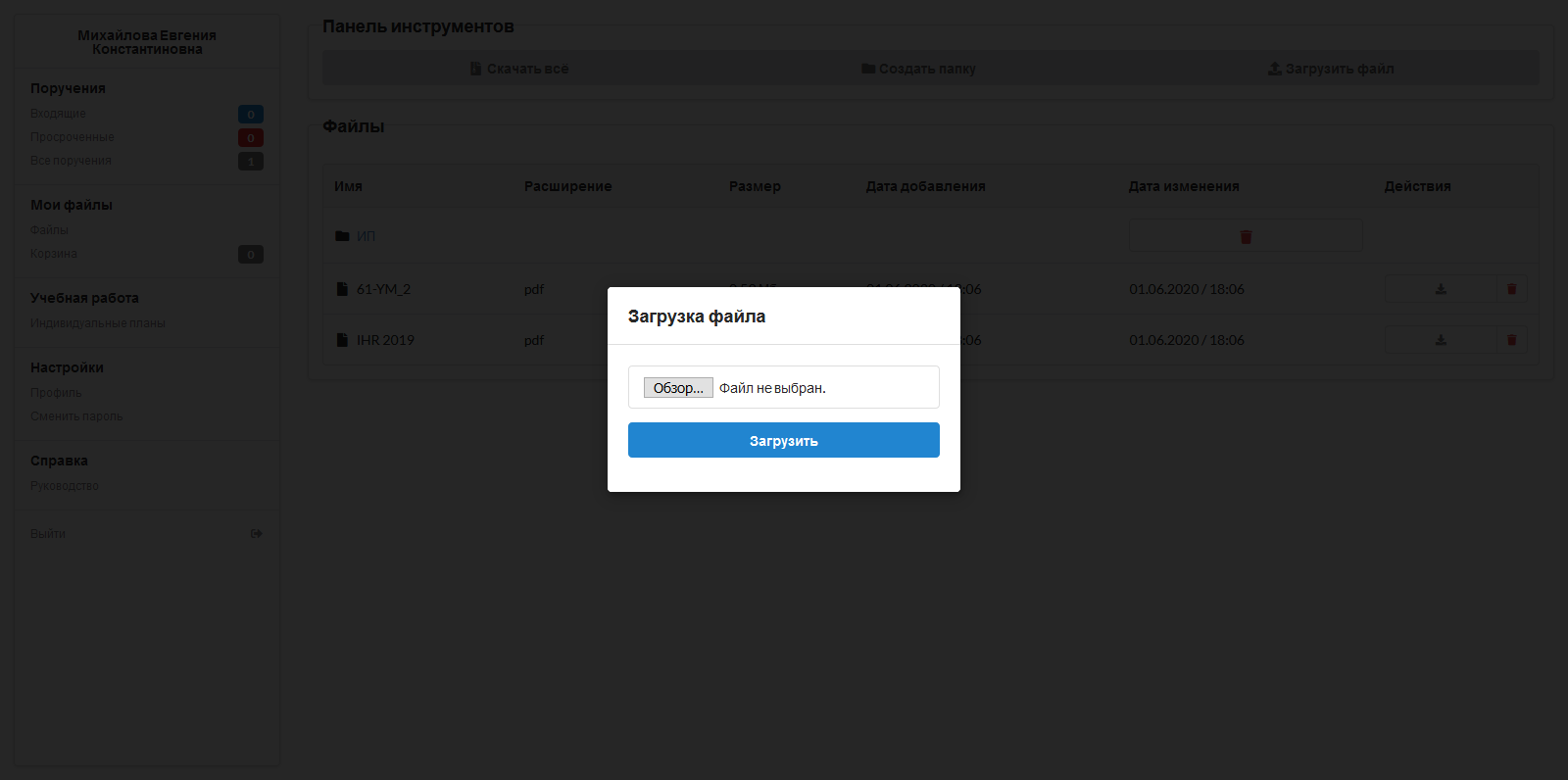


Рисунок – Модальное окно «Загрузка файлов»

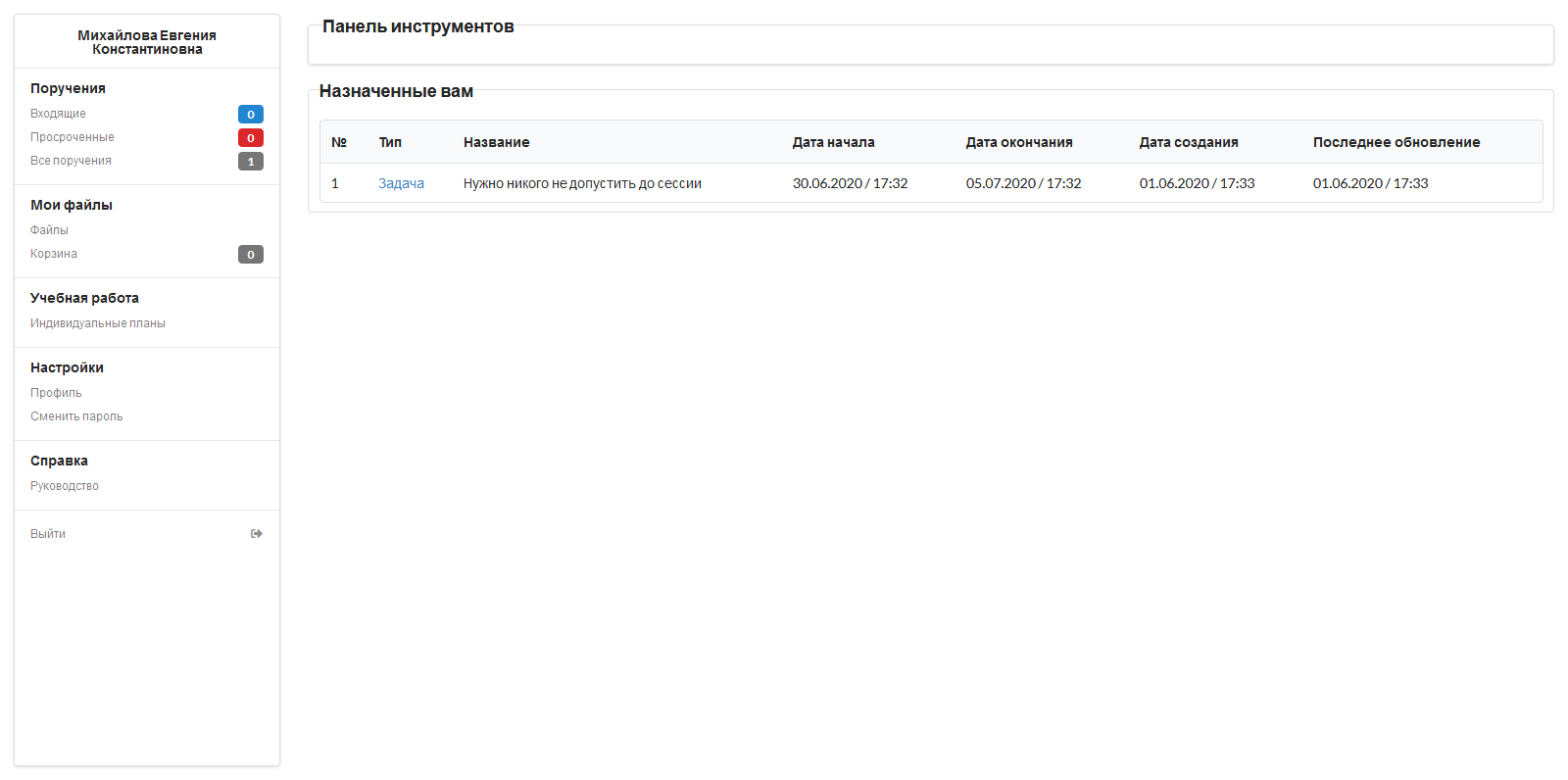


Рисунок – Список всех поручений

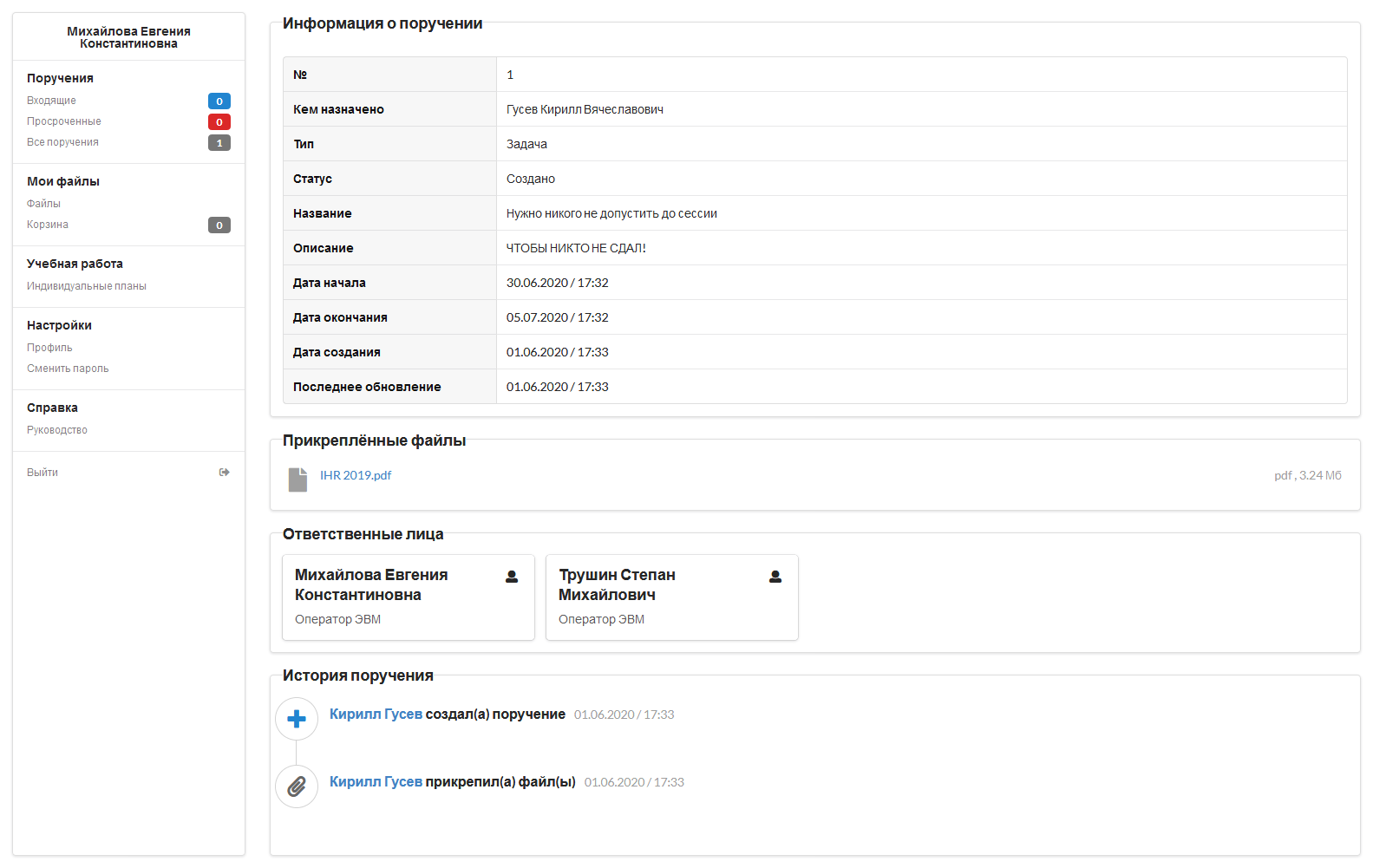


Рисунок – Информация о поручении

**4.3 Документация**

Документация на исходный код была создана с помощью doxygen (Рисунок 36-37).

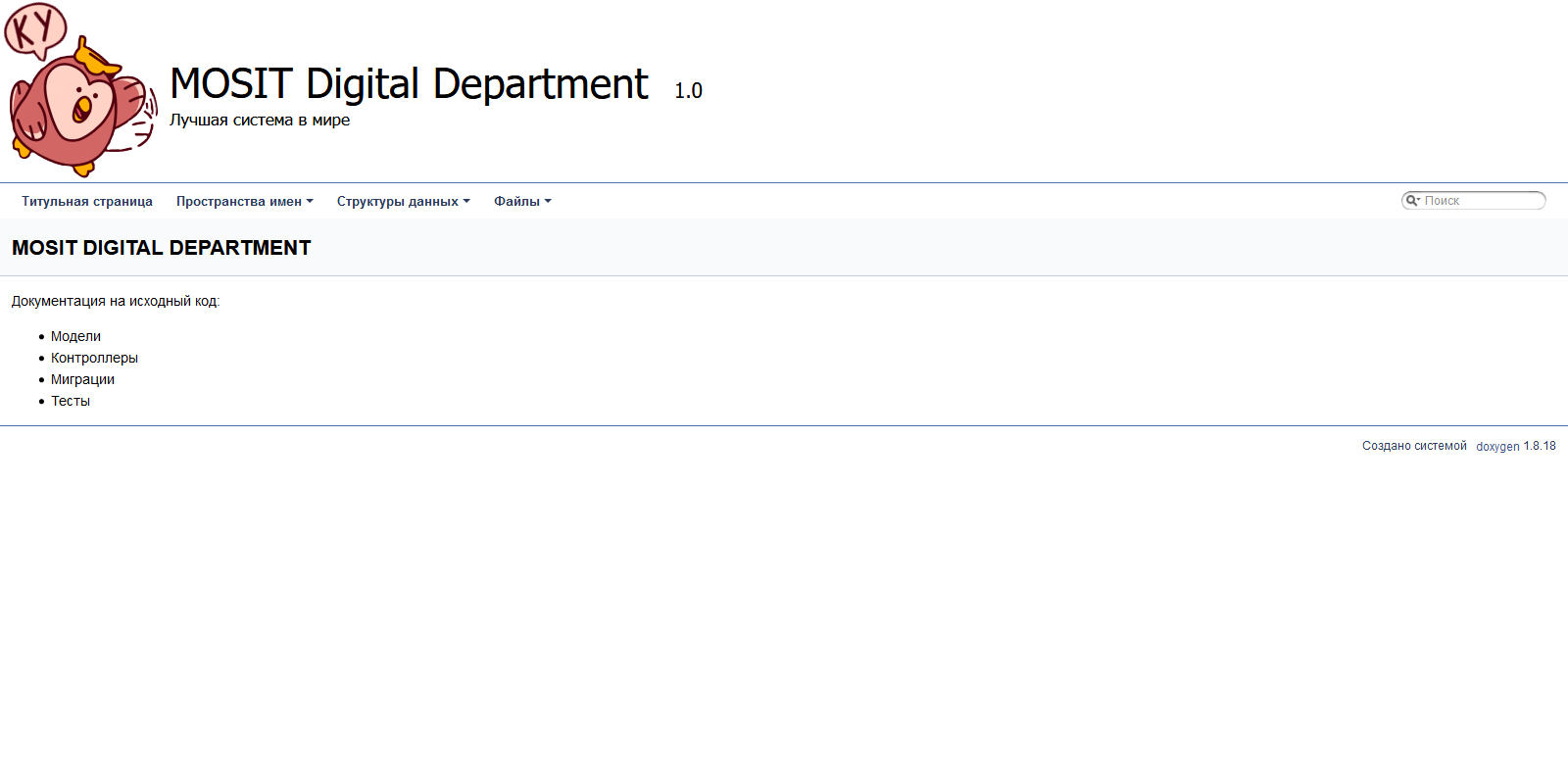


Рисунок – Документация на исходный код

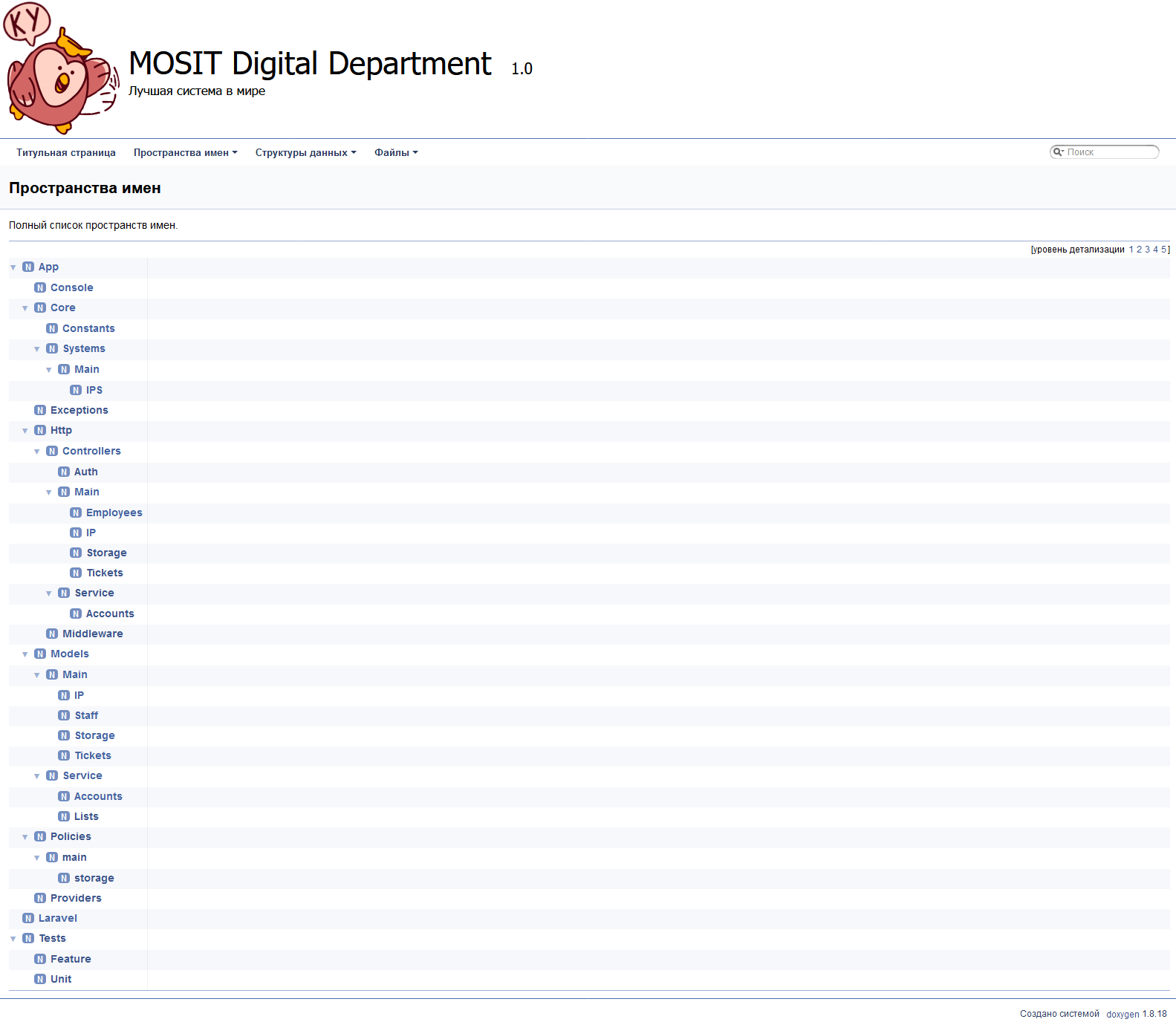


Рисунок – Пространства имён