

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ИНФОРМАТИКИ И ЭКОНОМИКИ

Выпускная квалификационная работа

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ
СИСТЕМЫ ОБЕСПЕЧЕНИЯ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ
ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ
ПЕРМСКОГО КРАЯ (НА ПРИМЕРЕ УНИВЕРСИТЕТСКОГО ОКРУГА
ЦИО ПРИ ПГГПУ)

Работу выполнила:
студентка 1243 группы
направления подготовки 09.03.03.
«Прикладная информатика»,
профиль «Прикладная
информатика в государственном и
муниципальном управлении»
Бурцева Анна Владимировна

(подпись)

«Допущена к защите в ГАК»
Зав. кафедрой

(подпись)

«__» _____ 2018 г.

Научный руководитель:
кандидат педагогических наук,
доцент кафедры прикладной
информатики
Худякова Анна Владимировна

(подпись)

ПЕРМЬ
2018

Оглавление

Введение.....	4
ГЛАВА 1. СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ ПЕРМСКОГО КРАЯ.....	6
1.1. Анализ деятельности Университетского округа Пермского края	6
1.2. Мониторинг доступности информации интернет-ресурсов университетских округов Пермского края	10
1.3. Структура ежегодного плана и отчета, предоставляемого центрами инновационного опыта и Университетским округом	17
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УНИВЕРСИТЕТСКИЙ ОКРУГ»	20
2.1. Описание структуры информационной системы.....	20
2.2. Концептуальное проектирование	21
2.2.1. Описание предметной области.....	21
2.2.2. Функциональное проектирование	26
2.2.2.1. Проектирование бизнес-процесса «Регистрация ЦИО»	27
2.2.2.3. Проектирование бизнес-процесса «Формирование плана мероприятий»	32
2.2.2.3. Проектирование бизнес-процесса «Формирование отчета ЦИО»	37
2.2.2.4. Проектирование бизнес-процесса «Формирование отчета Университетского округа».....	41
2.2.3. Объектное проектирование	46
2.2.3.1. Объекты и атрибуты.....	46
2.2.3.2. Связи между объектами.....	58
2.2.4. ER–диаграмма предметной области «Университетский округ»	58
2.3. Логическое проектирование	59
2.3.1. Разработка реляционной модели данных	59
2.3.1.1. Таблицы и атрибуты	60
2.3.1.2. Функциональные зависимости	63
2.3.1.3. Нормализация отношений	64
2.3.1.4. Диаграмма классов	66
2.3.1.5. Реляционные связи	67

2.3.1.6. Разработка отчетных документов и форм для ввода	69
2.3.1.7. Разработка запросов	79
2.3.2. Обеспечение целостности базы данных.....	79
2.3.2.1. Категорная целостность.....	80
2.3.2.2. Ссылочная целостность	81
2.3.2.3. Доменная целостность.....	81
2.4. Проектирование пользовательского интерфейса.....	88
2.4.1. Диаграммы деятельности	90
2.4.2.Эскиз пользовательского интерфейса	93
2.4.3. Описание пунктов пользовательского интерфейса	95
2.5. Разработка и реализация информационной системы «Университетский округ».....	99
ГЛАВА 3. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УНИВЕРСИТЕТСКИЙ ОКРУГ».....	111
3.1. Расчет затрат на информационную разработку автоматизированной системы «Университетский округ».....	111
3.2. Определение возможной цены программного продукта	114
3.3. Расчет основных затрат	115
3.4. Эффективность внедрения системы	116
Заключение.....	122
Библиографический список.....	123
Приложение.....	125

Введение

В Пермском крае с 2008 года начала развиваться новая форма взаимодействия образовательных организаций – Университетские округа, главной задачей которых является координация деятельности образовательных организаций Пермского края, которые получили статус краевого Центра инновационного опыта по результатам конкурсного отбора представленных на экспертизу проектов.

В настоящее время существует три Университетских округа: Университетский округ ПГГПУ, Университетский округ НИУ ВШЭ, Университетский округ ПГНИУ. Они обеспечивают профессиональное взаимодействие педагогов школ Пермского края, оказывают методическую и информационную поддержку образовательным организациям, относящимся к сети центров инновационного опыта (ЦИО).

У каждого Университетского округа есть информационная страница или сайт на сервере для представления результатов деятельности и размещения отчетности. Такое техническое решение не позволяет осуществлять автоматический сбор информации от центров инновационного опыта, входящих в состав Университетского округа, и подробно освещать их деятельность.

Именно поэтому разработка автономной, не привязанной к сайту образовательной организации, информационной системы, обеспечивающей сетевое взаимодействие инновационных образовательных организаций Пермского края, является актуальной.

Объектом исследования является информационная и аналитическая деятельность инновационных образовательных организаций Пермского края. Предметом исследования является автоматизация отчетности по результатам работы Университетского округа.

Целью выпускной квалификационной работы является проектирование и разработка информационной системы для обеспечения сетевого

взаимодействия инновационных образовательных организаций Пермского края (на примере Университетского округа ЦИО при ПГГПУ).

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать деятельность Университетского округа в соответствии с приказом Министерства образования Пермского края от 02.07.2008 №СЭД 26-01-04-66 «Об утверждении Положения о центрах инновационного опыта» и «Положение об университетском округе инновационных образовательных организаций Пермского края»
2. Провести мониторинг доступности информации интернет-ресурсов Университетских округов Пермского края.
3. Изучить структуру ежегодного плана и отчета, предоставляемого центрами инновационного опыта и Университетским округом.
4. Осуществить концептуальное и логическое проектирование информационной системы «Университетский округ».
5. Спроектировать информационную систему «Университетский округ» для удаленного доступа в многопользовательском режиме по технологии «клиент - сервер»
6. Разработать пользовательский интерфейс с помощью программного обеспечения Django.

Результаты выпускной квалификационной работы публиковались и обсуждались на VI Российской научно-практической конференции с международным участием «Наука и образование в обеспечении устойчивого развития человеческого потенциала в условиях перехода к цифровой экономике» (15–16 мая 2018 года).

Выпускная квалификационная работа состоит из введения, трех глав, заключения, библиографического списка и приложения.

ГЛАВА 1. СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ ПЕРМСКОГО КРАЯ

Сетевое взаимодействие - это система связей, позволяющих разрабатывать, апробировать и предлагать профессиональному педагогическому сообществу инновационные модели содержания образования и управления системой образования; это способ деятельности по совместному использованию ресурсов[1].

1.1. Анализ деятельности Университетского округа Пермского края

Университетский округ инновационных образовательных организаций Пермского края (Округ) является формой содержательного и организационного соуправления образовательным процессом в образовательных организациях Округа при методической, научной, кадровой и информационной поддержке ФГБОУ ВПО «Пермский государственный гуманитарно-педагогический университет» (Университет)[2].

Округ представляет собой специально организованную среду (образовательное пространство) вокруг Университета в масштабах Пермского края, в которой объединяются на добровольной основе образовательные организации с целью создания социально-развивающей среды, ориентированной на повышение качества образования, на реализацию моделей личностно-ориентированного и интегрированного образования, осуществления непрерывной переподготовки кадров.

Округ объединяет в сетевое сообщество Центры инновационного опыта (ЦИО), выявленные в результате конкурсного отбора, и другие образовательные организации Пермского края, ведущие инновационную образовательную деятельность.

Округ объединяет интеллектуальные ресурсы образовательных организаций всех уровней общего образования с целью создания единого образовательного пространства в рамках сетевого сообщества,

ориентированного на развитие системы образования Пермского края через развитие социально-развивающей среды, реализацию инновационных образовательных практик, а также осуществление непрерывной подготовки и переподготовки педагогических кадров.

По отношению к образовательным организациям, Округ осуществляет следующие виды деятельности:

а) организация и сопровождение сетевого взаимодействия между инновационными образовательными организациями Пермского края для создания единой информационно-образовательной среды, обеспечивающей «горизонтальное» общение педагогов образовательных учреждений: обмен опытом, трансфер методик и технологий, централизованный сбор данных, обратную связь;

б) обеспечение условий для научно-исследовательской деятельности ЦИО Пермского края;

в) научное обобщение и трансляция результатов инновационной деятельности образовательных организаций;

г) обеспечение координации деятельности ЦИО края по организации повышения квалификации педагогических работников и руководящих кадров на базе ЦИО;

д) создание условий для профессионального развития педагогических кадров ЦИО в рамках системы повышения квалификации Пермского края;

е) осуществление научно-исследовательского, методического сопровождения деятельности ЦИО Пермского края в качестве центров методического сопровождения педагогов для роста и дальнейшего распространения позитивного опыта подготовки специалистов – тьюторов.

Одной из задач деятельности Университетского округа заключается в обеспечение и научно- педагогического взаимодействия образовательных организаций, являющихся краевыми инновационными площадками с Университетом и другими субъектами образования.

Задачи сетевого взаимодействия:

- создание системы научного консультирования, в том числе с использованием новых интернет-технологий;
- организация научно-методической, информационной поддержки уникальных педагогических практик;
- подготовка предложений по повышению квалификации работников ЦИО, по разработке актуальных программ переподготовки и повышения квалификации представителей ЦИО (семинары, тренинги, программы профессиональной переподготовки и повышения квалификации и т.д.);
- развитие научно-исследовательской работы по направлению деятельности образовательных организаций с привлечением обучающихся и преподавателей университета, ЦИО.

Формы реализации сетевого взаимодействия:

– базовая кафедра – структурное подразделение Университета, функционирующие на основании Положения о базовой кафедре и договора между Университетом и образовательной организацией. Для деятельности базовой кафедры характерно возникновение, распространение и накопление новшеств и инициатив, приводящих к значительным изменениям содержания и качества образования. Научно-исследовательскую деятельность базовой кафедры сопровождает научный консультант, из сотрудников профессорско-преподавательского состава Университета, имеющий ученую степень кандидата или доктора наук;

– профессиональная площадка – это образовательное учреждение, осуществляющее инновационный поиск технологий и методов эффективного функционирования системы образования и его элементов. Научно-методическую работу профессиональной площадки сопровождает научный консультант, из сотрудников профессорско-преподавательского состава Университета, имеющий ученую степень кандидата или доктора наук;

– научно-практическая лаборатория – это образовательное учреждение, ведущее экспериментальную научно-исследовательскую работу, в условиях руководства одним или несколькими консультантами;

– «площадка творческих (педагогических) проб» – это образовательная организация, которая делает свои первые исследовательские профессиональные пробы, получают экспертную оценку и формирует стратегию своего будущего развития по выбранному проекту.

Центры инновационного опыта могут создаваться на базе образовательного учреждения Пермского края любого вида и формы собственности по трем направлениям деятельности[3]:

а) ЦИО как центр повышения квалификации, на базе которых высшее учебное заведение осуществляет повышение квалификации работников образования. Такое повышение квалификации организуется как реализация заказа (государственно - общественного) и/или как трансляция имеющегося у учреждения успешного инновационного опыта;

б) ЦИО как научно-методический центр, осуществляющий послекурсовую поддержку и сопровождение педагогов, участвует в формировании заказа на повышение квалификации от профессионального сообщества;

в) ЦИО как центр инновационных разработок в образовательной деятельности, занимающийся инновационными разработками в сфере образовательной деятельности, относящимися к новому содержанию, инновационным технологиям и управлению в системе образования.

Деятельность краевых центров инновационного опыта в системе повышения квалификации Пермского края:

– направлена на организацию повышения квалификации в новых условиях модернизации образования, осуществление эффективного профессионального взаимодействия педагогов региона, оказание качественной методической, дидактической, технологической и

информационной поддержки образовательным учреждениям, относящимся к сети ЦИО.

- может преследовать цели распространения успешного педагогического опыта в профессиональном сообществе всего края, развития потенциала педагогических работников образовательных учреждений, относящихся к сети ЦИО, определения приоритетных задач развития системы образования муниципалитета и путей их решения, представления интересов педагогического сообщества, привлечения педагогов других образовательных учреждений к работе в ЦИО.
- в новой системе повышения квалификации, позволяет обеспечить широкий охват педагогических работников в отдаленных от центра территориях в специальной области образовательной деятельности, обеспечить научно-методическое сопровождение профессиональной деятельности педагогов.
- предполагает постоянную систему отчетности в форме текущих отчетов по реализации программы работы ЦИО и итогового отчета в форме публичного доклада руководителя ЦИО с последующим размещением доклада в сети Интернет.

1.2. Мониторинг доступности информации интернет-ресурсов университетских округов Пермского края

В Пермском крае существуют 3 университетских округа: Университетский округ ПГГПУ[4], Университетский округ НИУ ВШЭ[5], Университетский округ ПГНИУ[6].

Мониторинг интернет-ресурсов Университетских округов проводится с целью определения комплексной оценки степени информационной доступности сайтов, освещающих деятельность Университетских округов.

Методика мониторинга интернет-ресурсов Университетских округов включает в себя исследование и оценку информационного наполнения и

технологических свойств сайта, а также выработку рекомендаций по его совершенствованию[7].

При исследовании содержания интернет-ресурса используется метод вэб-контент-анализа. Данный метод представляет собой изучение экспертами содержания сайта в сети Интернет в режиме он-лайн для определения наличия или отсутствия на сайте выбранных количественных и качественных характеристик. Эффективность данного метода обусловлена максимальной его приближенностью к тем способам поиска информации, к которым прибегают обычные пользователи интернет-сайтов.

Мониторинг базируется на следующих основополагающих принципах:

- принцип открытости: описание методики мониторинга и его результатов публикуется для всеобщего сведения;
- принцип независимости: результаты мониторинга зависят исключительно от оценок экспертов, независимых друг от друга;
- принцип объективности: результаты мониторинга отражают текущую ситуацию в исследуемой сфере и не зависят от субъективного мнения и воли проводивших исследование лиц; эксперты, участвующие в мониторинге, обладают соответствующими квалификацией и знаниями.

В рамках мониторинга исследование содержания сайта происходит по перечню параметров (содержательных и технологических), которые оцениваются по определенным критериям. Результатом мониторинга является интегрированная оценка сайта, которая отображает, насколько совокупность их количественных и качественных характеристик соответствует требованиям информационной доступности сайта.

Под содержательным параметром мониторинга понимается выявленное экспертами свойство официального сайта, характеризующее его информационное наполнение. Под технологическим параметром понимается выявленное экспертами свойство официального сайта, характеризующее технологические, программные и лингвистические средства обеспечения пользования официальными сайтами.

Параметры оцениваются экспертами по следующим критериям:

- наличие;
- полнота;
- актуальность;
- навигационная доступность;
- форматная доступность.

Критерии в зависимости от их сущности разделяются на два типа: количественные (наличие и полнота) и качественные (актуальность, навигационная доступность и форматная доступность).

При мониторинге сайтов оптимально чувствительной является трехбалльная система (шкала), что было определено эмпирическим путем. Использование в исследовании более чувствительной шкалы приведет к систематическим погрешностям.

«Наличие» – количественный критерий, характеризующий факт размещения на интернет-ресурсе информации или сервиса, соответствующих параметру. При анализе параметра по критерию «Наличие» эксперты оценивают Коэффициент наличия параметра (K_1), который может иметь два значения:

- в случае, если соответствующие параметру информация или сервис присутствует на сайте – $K_1 = 1$;
- в случае, если соответствующие параметру информация или сервис отсутствует на сайте – $K_1 = 0$.

«Полнота» – количественный критерий, характеризующий достаточность объема размещенной на сайте информации, для формирования целостного представления по соответствующему параметру.

При анализе параметра по критерию «Полнота» эксперты оценивают Коэффициент полноты информации (K_2), который может принимать одно из трех значений в зависимости от степени полноты информации:

- высокая степень полноты (90-100%) – $K_2 = 1$ (представлен весь необходимый состав сведений и вся информация изложена в полном объеме);
- средняя степень полноты (30-90%) – $K_2 = 0,5$ (представлен весь необходимый состав сведений, но информация изложена не в полном объеме; либо представлен не весь необходимый состав сведений, но информация изложена в полном объеме);
- низкая степень полноты (5-30%) – $K_2 = 0$ (представлена обрывочная информация).

«Актуальность» – качественный критерий, характеризующий степень частоты обновления информации на интернет-ресурсе и сохранения ценности информации на момент ее анализа экспертами. Данный критерий зависит от динамики изменения характеристик информации во времени. При анализе параметра по критерию «Актуальность» эксперты оценивают Коэффициент актуальности информации (K_3), который может принимать одно из трех значений в зависимости от степени актуальности информации:

- высокая степень актуальности – $K_3 = 1$ (размещенная на сайте информация является последней версией такой информации; учтены все изменения информации на момент оценки);
- средняя степень актуальности – $K_3 = 0,5$ (на сайте размещена предпоследняя версия информации; не учтены последние изменения, произошедшие не ранее, чем за 1 месяц до оценки);
- низкая степень актуальности – $K_3 = 0$ (на сайте размещены устаревшие сведения; не учтены изменения, произошедшие ранее, чем за 1 месяц до оценки).

«Навигационная доступность» – качественный критерий, характеризующий удобство поиска размещенной на сайте информации и логичность ее размещения в соответствующем разделе сайта.

При анализе параметра по критерию «Навигационная доступность» эксперты оценивают Коэффициент навигационной доступности (K_4),

который может принимать одно из трех значений в зависимости от степени навигационной доступности размещенной на сайте информации:

- высокая степень навигационной доступности – $K_4 = 1$ (информация доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта; количество таких переходов – не более 3);
- средняя степень навигационной доступности – $K_4 = 0,5$ (информация доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта; количество таких переходов – более 3);
- низкая степень навигационной доступности – $K_4 = 0$ (информация не доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта, или размещена в не соответствующем ее содержанию разделе).

«Форматная доступность» – качественный критерий, характеризующий факт размещения на интернет-ресурсе соответствующей параметру информации в форматах, которые обеспечивают для пользователей удобство ее поиска и возможность ее дальнейшего использования. При анализе параметра по критерию «Форматная доступность» эксперты оценивают Коэффициент форматной доступности (K_5), который может принимать одно из трех значений в зависимости от степени форматной доступности размещенной на сайте информации:

- высокая степень форматной доступности – $K_5 = 1$ (информация размещена в гипертекстовом формате, а также в виде файла в формате, обеспечивающем его сохранение на технические средства пользователя и допускающем после возможность поиска и копирования фрагмента текста (далее файл для скачивания));
- средняя степень форматной доступности – $K_5 = 0,5$ (информация размещена на сайте в гипертекстовом формате и не размещена в виде файла для скачивания);
- низкая степень форматной доступности – $K_5 = 0$ (информация на сайте не размещена в гипертекстовом формате).

В ходе мониторинга интернет-ресурсов Университетских округов Пермского края были получены следующие результаты (Таблица 1).

Мониторинг веб-сайта Университетского округа НИУ ВШЭ:

Критерий 1. «Наличие». $K_1 = 1$. Соответствующая параметру информация или сервис присутствует на сайте.

Критерий 2. «Полнота». $K_2 = 0,5$. На странице сайта «Презентационные мероприятия 2017-2018 учебный год» не заполнено Учебно-методические разработки и Галерея, нет списка новостей от школ ЦИО.

Критерий 3. «Актуальность». $K_3 = 1$. На сайте размещена актуальная информация.

Критерий 4. «Навигационная доступность». $K_4 = 0,5$. Информация доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта; количество таких переходов – более 3.

Критерий 5. «Форматная доступность». $K_5 = 1$. Информация размещена в гипертекстовом формате, а также в виде файла для скачивания.

Мониторинг веб-сайта Университетского округа ПГГПУ:

Критерий 1. «Наличие». $K_1 = 1$. Соответствующая параметру информация или сервис присутствует на сайте.

Критерий 2. «Полнота». $K_2 = 0,5$. На странице сайта «План деятельности» не хватает колонки «Результат» со ссылками на документы, не заполнено «История округа».

Критерий 3. «Актуальность». $K_3 = 0,5$. На сайте размещена предпоследняя версия информации (последний фоторепортаж – 08.06.2017). Список новостей не обновляется с 01.12.17.

Критерий 4. «Навигационная доступность». $K_4 = 0,5$. Информация доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта; количество таких переходов – более 3. Не совсем понятно выделение Мероприятий ЦИО в отдельную страницу, которая содержит только мероприятия одного месяца и не всех школ. Не ясен регламент предоставления и публикации информации о событиях.

Критерий 5. «Форматная доступность». $K_5 = 1$. Информация размещена в гипертекстовом формате, а также в виде файла для скачивания.

Мониторинг веб-сайта Университетского округа ПГНИУ:

Критерий 1. «Наличие». $K_1 = 1$. Соответствующая параметру информация или сервис присутствует на сайте.

Критерий 2. «Полнота». $K_2 = 0,5$. На странице сайта представлена не вся информация о деятельности Университетского округа, нет списка новостей от школ ЦИО.

Критерий 3. «Актуальность». $K_3 = 0$. На сайте не размещена актуальная информация о деятельности Университетского округа.

Критерий 4. «Навигационная доступность». $K_4 = 1$. Информация доступна путем последовательного перехода по гиперссылкам, начиная с главной страницы сайта.

Критерий 5. «Форматная доступность». $K_5 = 1$. Информация размещена в гипертекстовом формате, а также в виде файла для скачивания.

Таблица 1

Результат мониторинга интернет-ресурсов Университетских округов
Пермского края

Веб-сайт	Критерии				
	Наличие (К 1)	Полнот а (К 2)	Актуальност ь (К 3)	Навигационна я доступность (К 4)	Форматная доступност ь (К 5)
Университетский округ ПГГПУ	1	0,5	0,5	0,5	1
Университетский округ НИУ ВШЭ	1	0,5	1	0,5	1
Университетский округ ПГНИУ	1	0,5	0	1	1
Среднее значение	1	0,5	0,5	0,6	1

Таким образом, анализ интернет-ресурсов Университетских округов Пермского края позволяет сделать вывод, что интернет-ресурсы требуют доработки, т.к. в полной мере не обеспечивают сетевое взаимодействие между ЦИО.

1.3. Структура ежегодного плана и отчета, предоставляемого центрами инновационного опыта и Университетским округом

Планирование деятельности Университетского округа представлено в виде: календарного плана реализации проекта и годового плана реализации инновационного проекта.

Календарный план реализации проекта включает в себя:

- Название мероприятия;
- Описание мероприятия;
- Результат;
- Требования к результату;
- Плановый срок исполнения;
- Подтверждающие документы.

Годовой план реализации инновационного проекта:

- Цель на текущий год;
- Задачи по реализации проекта на текущий год;
- Срок реализации проекта;
- Мероприятие;
- Сроки;
- Место проведения;
- Категория участников;
- Ответственный.

Ежегодный отчет, предоставляемый центрами инновационного опыта состоит из:

- Общей информация об образовательной организации (наименование проекта; структурная организация, реализующая проект; ФИО автора проекта; ФИО руководителя проекта; ссылка на раздел на сайте, посвященный проекту);
- Общая информация о проекте (актуальность; цель; задачи; краткое обоснование инновационности проекта; ожидаемые продукты проекта; способы апробации и диссеминации результатов; риски и пути преодоления; механизм самооценки);
- Ресурсное обеспечение проекта (нормативно-правовое обеспечение; кадровое обеспечение проекта; информационное обеспечение проекта; опыт успешно реализованных проектов);
- Реализация инновационного проекта (описание соответствия содержания изначально запланированного проекта и полученных результатов; описание текущей актуальности продуктов; продукты; описание методов и критериев мониторинга качества проекта; результаты самооценки, мониторинга);
- Перечень достигнутых результатов (проведенные мероприятия; выводы по результатам деятельности ЦИО).

Выводы по главе 1

1. Университетские округа - новая форма взаимодействия образовательных организаций. Их главной задачей является координация деятельности образовательных организаций Пермского края, которые получили статус краевого Центра инновационного опыта по результатам конкурсного отбора представленных на экспертизу проектов. Деятельность ЦИО регламентируется планом реализации инновационного проекта и ежегодным отчетом.

2. В настоящее время существует три Университетских округа: Университетский округ ПГГПУ, Университетский округ НИУ ВШЭ, Городской университетский округ при ПГНИУ.

3. Мониторинг интернет-ресурсов Университетских округов показал низкий уровень комплексной оценки степени информационной доступности сайтов, освещающих деятельность Университетских округов. Таким образом, можно сделать вывод, что интернет-ресурсы требуют доработки, т.к. в полной мере не обеспечивают сетевое взаимодействие между ЦИО.

4. Планы и отчеты деятельности ЦИО заполняются в документах Word, что затрудняет их обработку и аналитику. Поэтому разработка информационной системы, обеспечивающей сбор и анализ планов и отчетов, является актуальной.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УНИВЕРСИТЕТСКИЙ ОКРУГ»

Проектирование информационной системы состоит из трёх этапов:

- 1) концептуальное проектирование;
- 2) логическое проектирование;
- 3) проектирование пользовательского интерфейса приложения [8].

2.1. Описание структуры информационной системы

Информационная система обеспечения сетевого взаимодействия ЦИО Пермского края представляет собой «клиент-серверное» приложение, серверная часть информационной системы выполнена в СУБД MySQL, для разработки клиентского приложения использовался веб-фреймворк Django на языке программирования Python. Разрабатываемая информационная система предназначена для 3-х категорий пользователей: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа.

При входе в информационную систему администратор ЦИО проходит авторизацию. После успешного прохождения авторизации администратор ЦИО попадает в главное меню. Администратор ЦИО может добавлять проекты, мероприятия, создавать планы мероприятий и отчеты по закрепленному за ним ЦИО.

При входе в информационную систему научный руководитель ЦИО проходит авторизацию. После успешного прохождения авторизации, открывается страница главного меню. Научный руководитель ЦИО может создавать отчеты и планы мероприятий по закрепленному за ним ЦИО, просмотреть отчеты и планы мероприятий других ЦИО, ознакомиться с планами мероприятий и отчетами Университетского округа.

При входе в информационную систему руководитель Университетского округа проходит авторизацию. После успешного

прохождения авторизации, открывается страница главного меню, в котором руководитель Университетского округа может создать отчеты и планы мероприятий по Университетскому округу, статистический отчет по проведенным мероприятиям, сделать выборку мероприятий за определенный период, просмотреть отчеты и планы мероприятий всех ЦИО.

2.2. Концептуальное проектирование

Концептуальное проектирование – это первый этап проектирования информационных систем. На данном этапе описывается, что должна включать в себя проектируемая информационная система, определяются объекты, их атрибуты и связи, с помощью которых составляется ER-диаграмма.

2.2.1. Описание предметной области

Предметной областью разрабатываемой информационной системы является деятельность Университетского округа по обеспечению сетевого взаимодействия между ЦИО. Университетский округ расположен по адресу ул.Сибирская 26, каб.1.

Об Университетском округе в разрабатываемой базе данных содержится следующая информация:

- 1) Название
- 2) ФИО руководителя Университетского округа
- 3) Адрес
- 4) Телефон
- 5) e-mail
- 6) Сайт

Университетский округ координирует деятельность образовательных организаций Пермского края, которые получили статус Центра

инновационного опыта по результатам конкурсного отбора предоставленных на экспертизу проектов, и с ними Университетский округ заключил договор.

О Центре инновационного опыта в разрабатываемой базе данных содержится следующая информация:

- 1) Название образовательной организации
- 2) Директор образовательной организации
- 3) Руководитель ЦИО
- 4) Адрес
- 5) Телефон
- 6) Сайт

Каждое ЦИО реализует проект и проводит перечень мероприятий в течение года. В конце года ЦИО предоставляет отчет о результатах выполнения проекта и составляет план мероприятий на следующий год.

О проекте в разрабатываемой базе данных содержится следующая информация:

- 1) Название проекта
- 2) ФИО автора проекта
- 3) ФИО руководителя проекта
- 4) Ссылка на раздел на сайте, посвященный проекту
- 5) Актуальность
- 6) Цель
- 7) Задачи
- 8) Краткое обоснование инновационности проекта
- 9) Ожидаемые продукты проекта
- 10) На разработку какого иного инновационного контента
направлен проект
- 11) Способы апробации и диссеминации результатов
- 12) Риски и пути преодоления
- 13) Механизм самооценки

О мероприятиях в разрабатываемой базе данных содержится следующая информация:

- 1) Название мероприятия
- 2) Дата
- 3) Место проведения
- 4) Категория участников
- 5) Ответственный за мероприятие

О результатах проекта в разрабатываемой базе данных содержится следующая информация:

- 1) Нормативно-правовое обеспечение
- 2) Кадровое обеспечение проекта
- 3) Информационное обеспечение проекта
- 4) Опыт успешно реализованных проектов
- 5) Описание соответствия содержания изначально запланированного проекта и полученных результатов
- 6) Описание текущей актуальности продуктов
- 7) Продукты
- 8) Описание методов и критериев мониторинга качества проекта
- 9) Результаты самооценки, мониторинга
- 10) Перечень достигнутых результатов

Разрабатываемая информационная система предназначена для 3-х категорий пользователей: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа.

Администратор ЦИО:

- Просмотр данных об Университетском округе
- Просмотр планов мероприятий Университетского округа
- Просмотр отчетов Университетского округа
- Редактирование, просмотр профиля ЦИО
- Ввод, редактирование, просмотр данных о проекте ЦИО

- Заполнение, редактирование, просмотр, печать плана мероприятий по реализации инновационного проекта ЦИО
- Заполнение, редактирование, просмотр, печать отчетов ЦИО

Научный руководитель ЦИО:

- Просмотр данных об Университетском округе
- Просмотр планов мероприятий Университетского округа
- Просмотр отчетов Университетского округа
- Редактирование, просмотр профиля ЦИО
- Ввод, редактирование, просмотр данных о проекте ЦИО
- Заполнение, редактирование, просмотр, печать плана мероприятий по реализации инновационного проекта ЦИО
- Заполнение, редактирование, просмотр, печать отчетов ЦИО
- Просмотр данных о проектах других ЦИО
- Просмотр мероприятий в рамках проектов других ЦИО
- Просмотр отчетов о проектах других ЦИО

Руководитель Университетского округа:

- Ввод, редактирование, просмотр данных об Университетском округе
- Создание, редактирование, просмотр, печать плана мероприятий Университетского округа
- Создание, редактирование, просмотр, печать отчетов Университетского округа
- Ввод, редактирование, просмотр данных о ЦИО
- Ввод, редактирование, просмотр данных о проекте ЦИО
- Ввод, редактирование, просмотр, печать плана мероприятий по реализации инновационного проекта ЦИО
- Ввод, редактирование, просмотр, печать отчетов ЦИО
- Просмотр статистического отчета по мероприятиям
- Создание и печать выборки мероприятий за определенный период

Исходя из описания предметной области, можно выделить следующие бизнес-процессы:

1. Регистрация ЦИО

- Конкурс инновационных проектов образовательных учреждений
- Регистрация образовательного учреждения в ИС и присвоение ему статуса ЦИО
- Выдача логина и пароля ЦИО для входа в ИС

2. Формирование плана мероприятий

- Заполнение данных о проекте ЦИО
- Добавление мероприятий по реализации проекта ЦИО
- Формирование плана мероприятий ЦИО
- Формирование плана мероприятий Университетского округа

3. Формирование отчета ЦИО

- Сбор данных для отчета ЦИО
- Формирование ежегодного отчета ЦИО
- Формирование итогового отчета ЦИО

4. Формирование отчета Университетского округа

- Сбор данных для отчета Университетского округа
- Формирование ежегодного отчета Университетского округа
- Формирование итогового отчета Университетского округа

2.2.2. Функциональное проектирование

Функциональное проектирование проводилось в среде RAMUS и STARUML. Деятельность Университетского округа представлена контекстной диаграммой (рис.1).

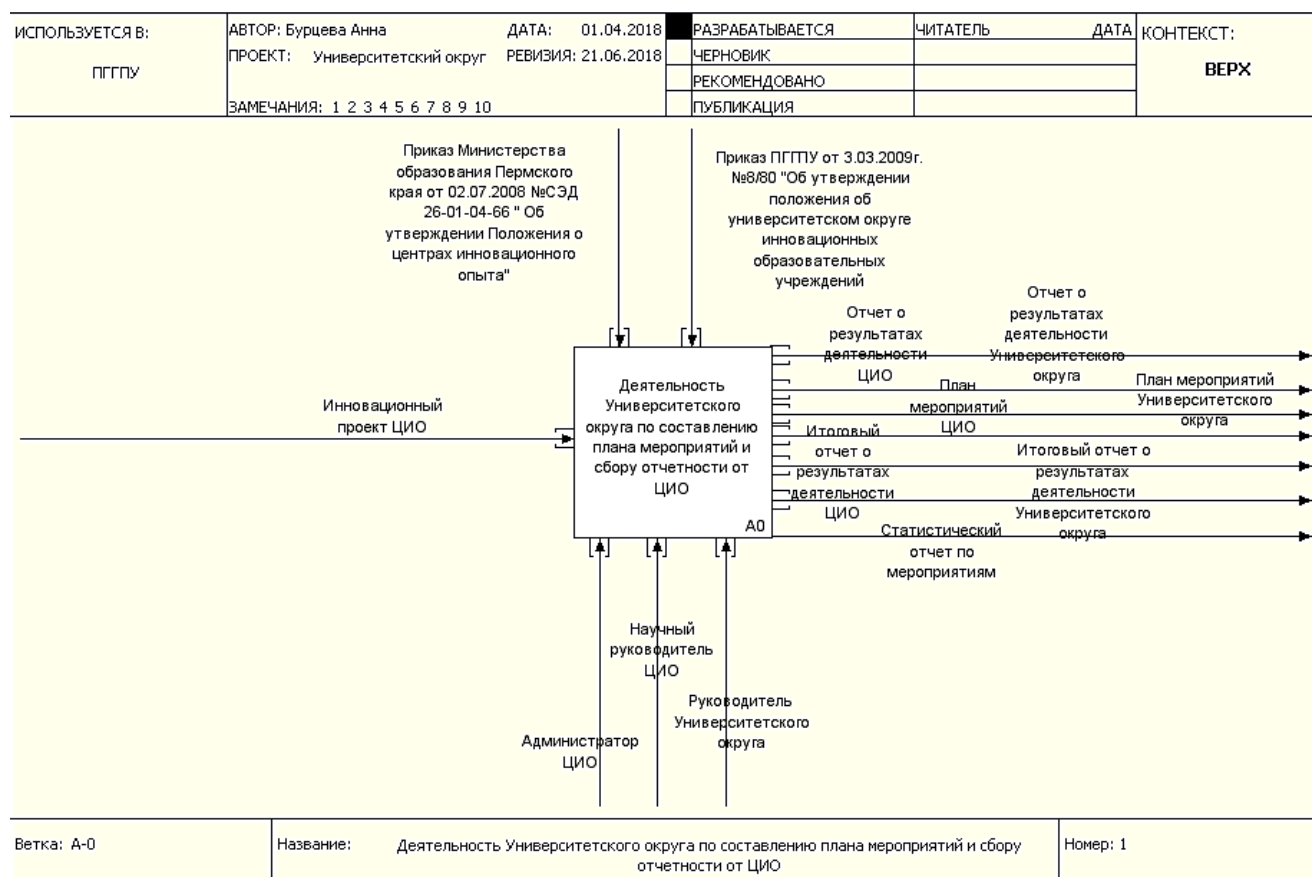


Рис.1 Контекстная диаграмма «Деятельность Университетского округа»

Для автоматизации деятельности в информационной системе рассматриваются бизнес-процессы, которые необходимо автоматизировать.

На рисунке 2 представлена диаграмма декомпозиции контекстной диаграммы, где показано, что деятельность Университетского округа по сбору отчетности от ЦИО и составлению плана мероприятий разбивается на 4 бизнес-процесса (регистрация ЦИО, формирование плана мероприятий, формирование отчета ЦИО, формирование отчета УО) и показывается их взаимосвязь между собой, т.е проводится функциональная декомпозиция –

система разбивается на подсистемы, и каждая подсистема описывается отдельно.

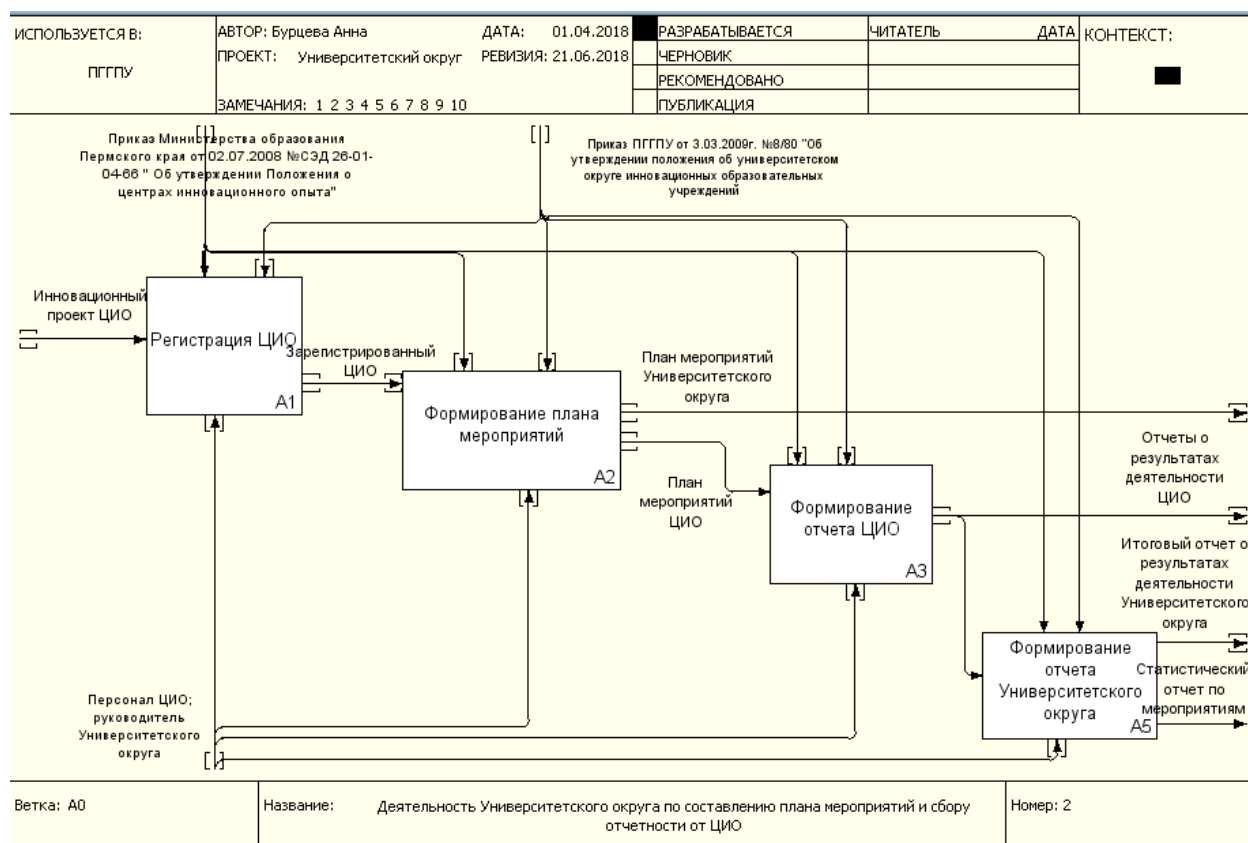


Рис.2 Диаграмма декомпозиции контекстной диаграммы «Деятельность Университетского округа»

2.2.2.1. Проектирование бизнес-процесса «Регистрация ЦИО»

Проектирование бизнес-процесса «Регистрация ЦИО» реализовано с помощью 4-х диаграмм: диаграммы декомпозиции бизнес-процесса «Регистрация ЦИО», диаграммы деятельности бизнес-процесса «Регистрация ЦИО», диаграммы последовательности действий бизнес-процесса «Регистрация ЦИО», диаграммы кооперации бизнес-процесса «Регистрация ЦИО».

Диаграмма декомпозиции бизнес-процесса «Регистрация ЦИО»

Диаграмма декомпозиции бизнес-процесса «Регистрация ЦИО» (рис.3) показывает, что бизнес-процесс «Регистрация ЦИО» состоит из таких подпроцессов, как конкурс инновационных проектов образовательных учреждений, регистрация образовательного учреждения в ИС и присвоение ему статуса ЦИО, выдача логина и пароля ЦИО для входа в ИС.

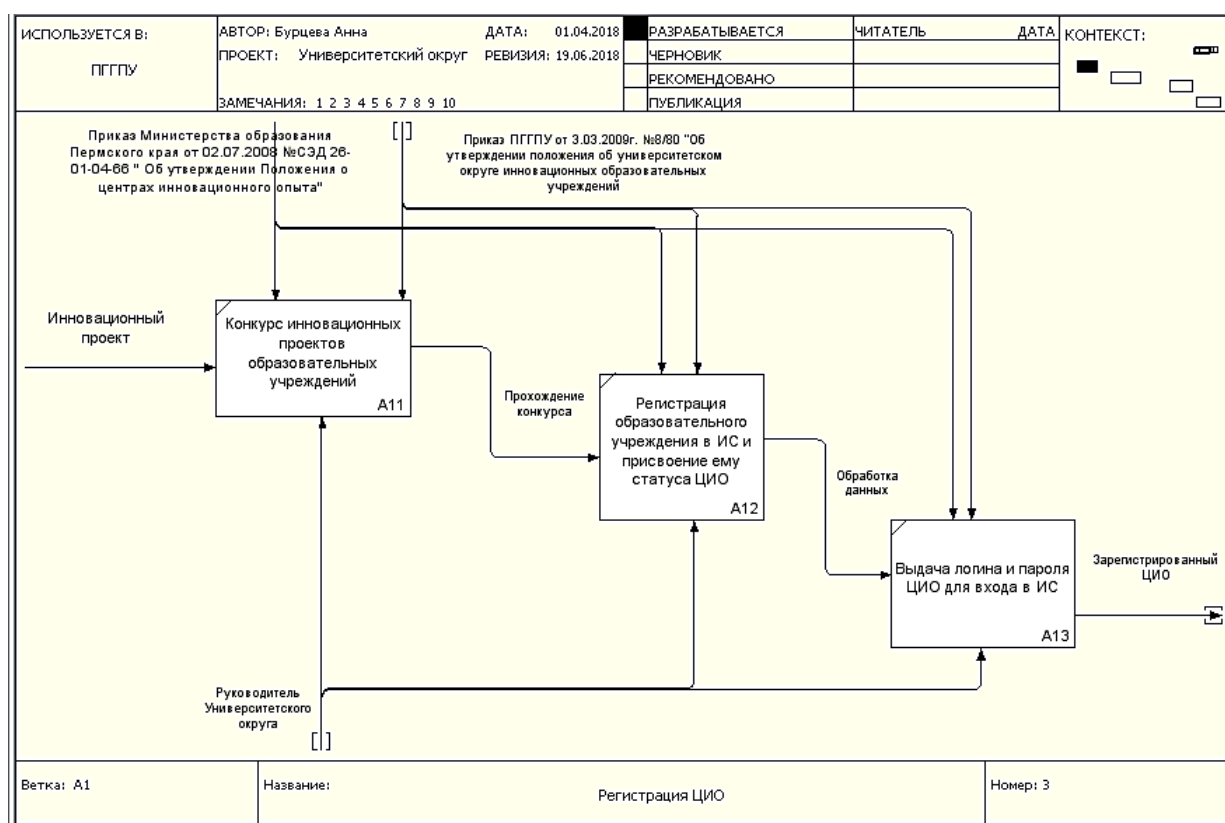


Рис.3 «Диаграмма декомпозиции бизнес-процесса «Регистрация ЦИО»

Диаграмма деятельности бизнес-процесса «Регистрация ЦИО»

Для более детального описания бизнес-процесса «Регистрация ЦИО» используется диаграмма деятельности (рис.4), на которой представлено разложение на составные части бизнес-процесса «Регистрация ЦИО», а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются между собой потоками от выходов

одного узла к входам другого, с указанием их исполнителей (ЦИО, руководитель Университетского округа, система).

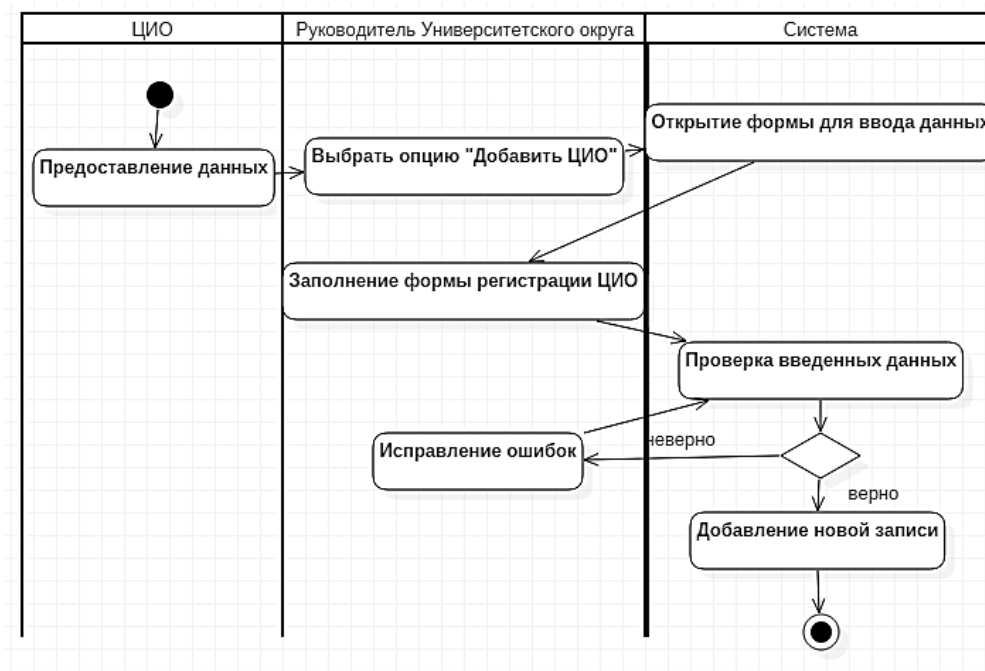


Рис.4. «Диаграмма деятельности бизнес-процесса «Регистрация ЦИО»

Диаграмма последовательности действий бизнес-процесса «Регистрация ЦИО»

Для представления взаимодействий объектов в системе используют диаграммы последовательности и кооперации. Диаграмма последовательности действий (рис.5) отображает обмен сообщениями (табл.2) между объектами во времени.

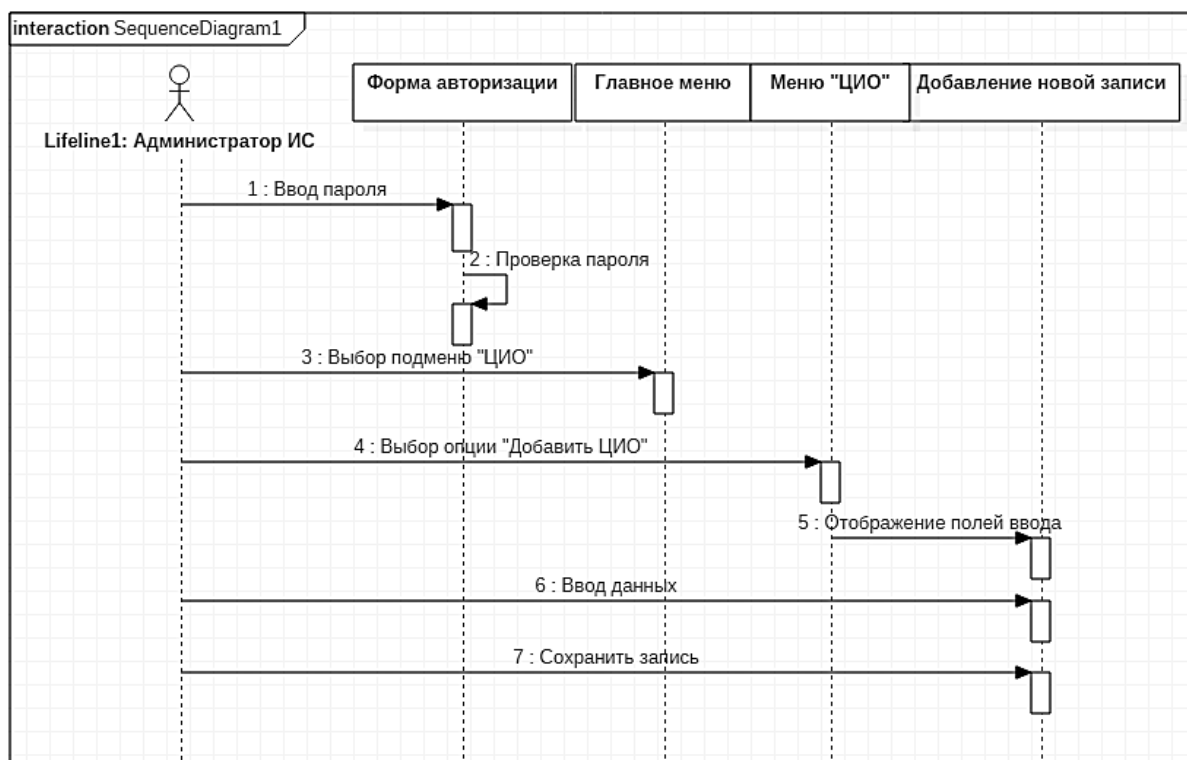


Рис. №5 Диаграмма последовательности действий бизнес-процесса
«Регистрация ЦИО»

Таблица №2. Описание действий бизнес-процесса «Регистрация ЦИО»

Номер сообщения	Объект - отправитель сообщения	Объект - получатель сообщения	Название
1	Администратор ИС	Форма авторизации	Ввод пароля
2	Форма авторизации	Форма авторизации	Проверка пароля
3	Администратор ИС	Главное меню	Выбор подменю «ЦИО»
4	Администратор ИС	Меню «ЦИО»	Выбор опции «Добавить ЦИО»
5	Меню «ЦИО»	Добавление новой записи	Отображение полей ввода

6	Администратор ИС	Добавление новой записи	Ввод данных
7	Администратор ИС	Добавление новой записи	Сохранить запись

Диаграмма кооперации бизнес-процесса «Регистрация ЦИО»

Диаграмма кооперации (рис.6) отображает структуру взаимодействия объектов(администратор ис, форма авторизации, главное меню, меню «ЦИО», добавление новой записи) в системе.

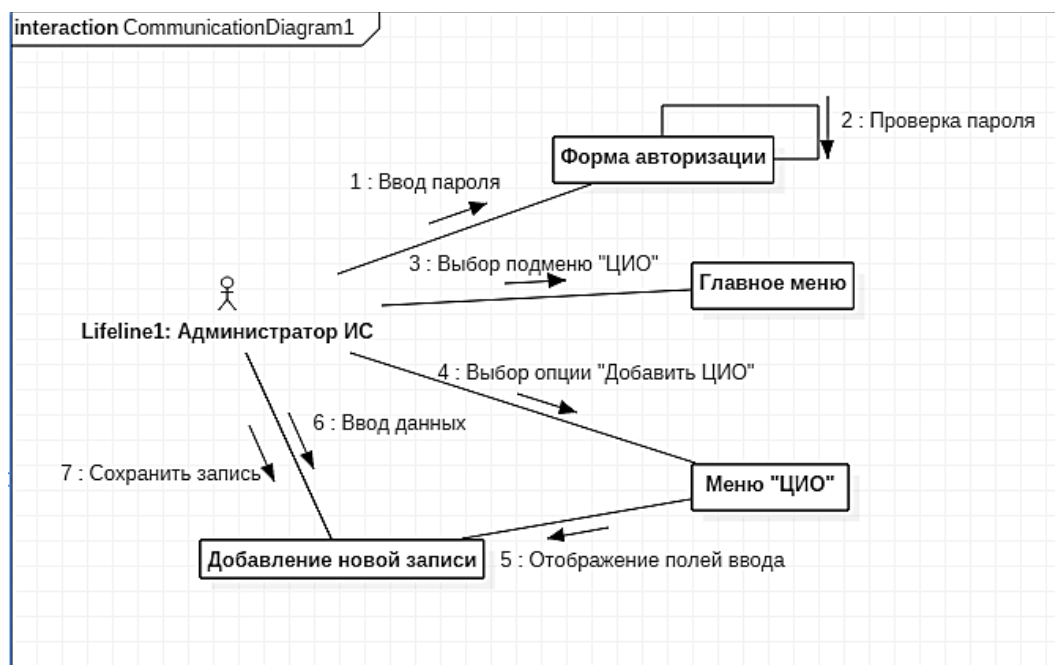


Рис. №6. Диаграмма кооперации бизнес-процесса «Регистрация ЦИО»

2.2.2.3. Проектирование бизнес-процесса «Формирование плана мероприятий»

Проектирование бизнес-процесса «Формирование плана мероприятий» реализовано с помощью 4-х диаграмм: диаграммы декомпозиции бизнес-процесса «Формирование плана мероприятий», диаграммы деятельности бизнес-процесса «Формирование плана мероприятий», диаграммы последовательности действий бизнес-процесса «Формирование плана мероприятий», диаграммы кооперации бизнес-процесса «Формирование плана мероприятий».

Диаграмма декомпозиции бизнес-процесса «Формирование плана мероприятий»

Диаграмма декомпозиции бизнес-процесса «Формирование плана мероприятий» (рис.7) показывает, что бизнес-процесс «Формирование плана мероприятий» состоит из таких подпроцессов, заполнение данных о проекте ЦИО, добавление мероприятий по реализации проекта ЦИО, формирование плана мероприятий ЦИО, формирование плана мероприятий Университетского округа.

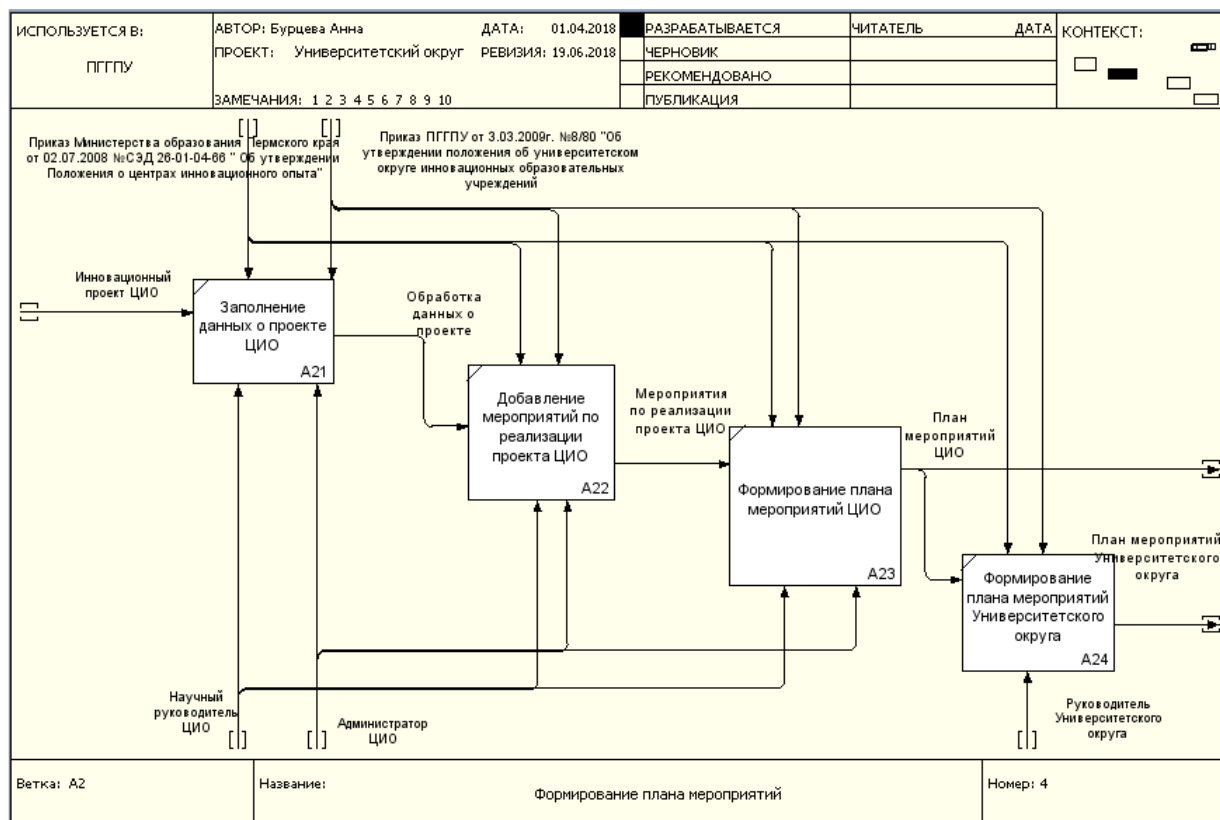


Рис.7 «Диаграмма декомпозиции бизнес-процесса «Формирование плана мероприятий»

Диаграмма деятельности бизнес-процесса «Формирование плана мероприятий»

Для более детального описания бизнес- процесса «Формирование плана мероприятий» используется диаграмма деятельности (рис.8), на которой представлено разложение на составные части бизнес-процесса «Формирование плана мероприятий», а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются между собой потоками от выходов одного узла к входам другого, с указанием их исполнителей (администратор ЦИО, руководитель Университетского округа, система).

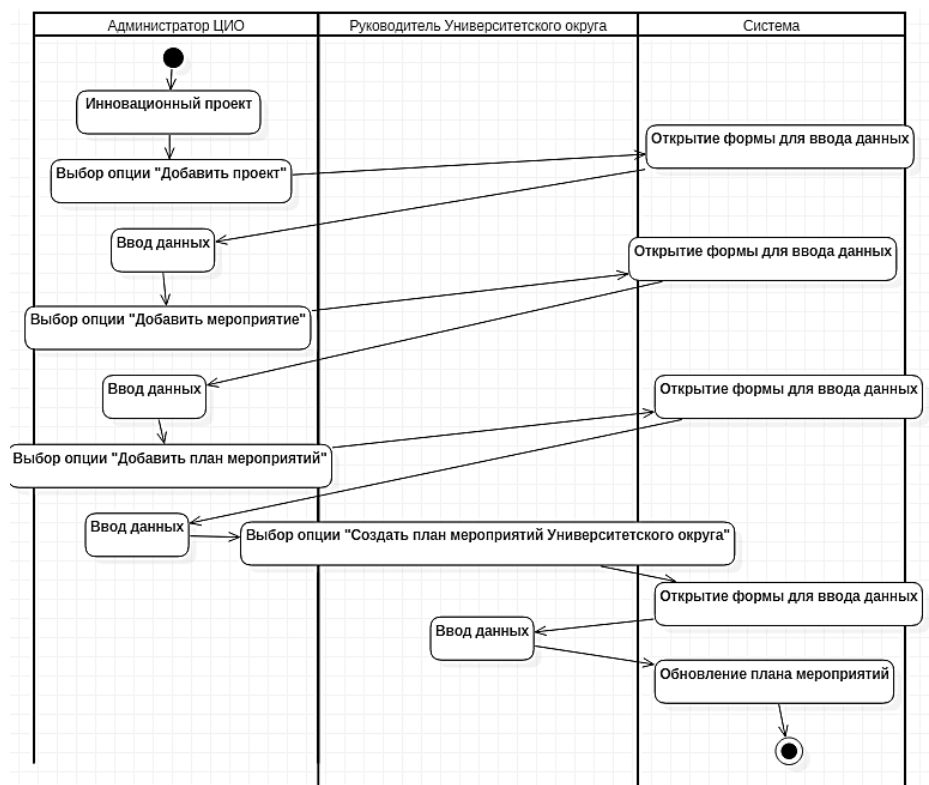


Рис.№8 «Диаграмма деятельности бизнес-процесса «Формирование плана мероприятий»

Диаграмма последовательности действий бизнес-процесса «Формирование плана мероприятий»

Для представления взаимодействий объектов в системе используют диаграммы последовательности и кооперации. Диаграмма последовательности действий (рис.9) отображает обмен сообщениями (табл.3) между объектами во времени.

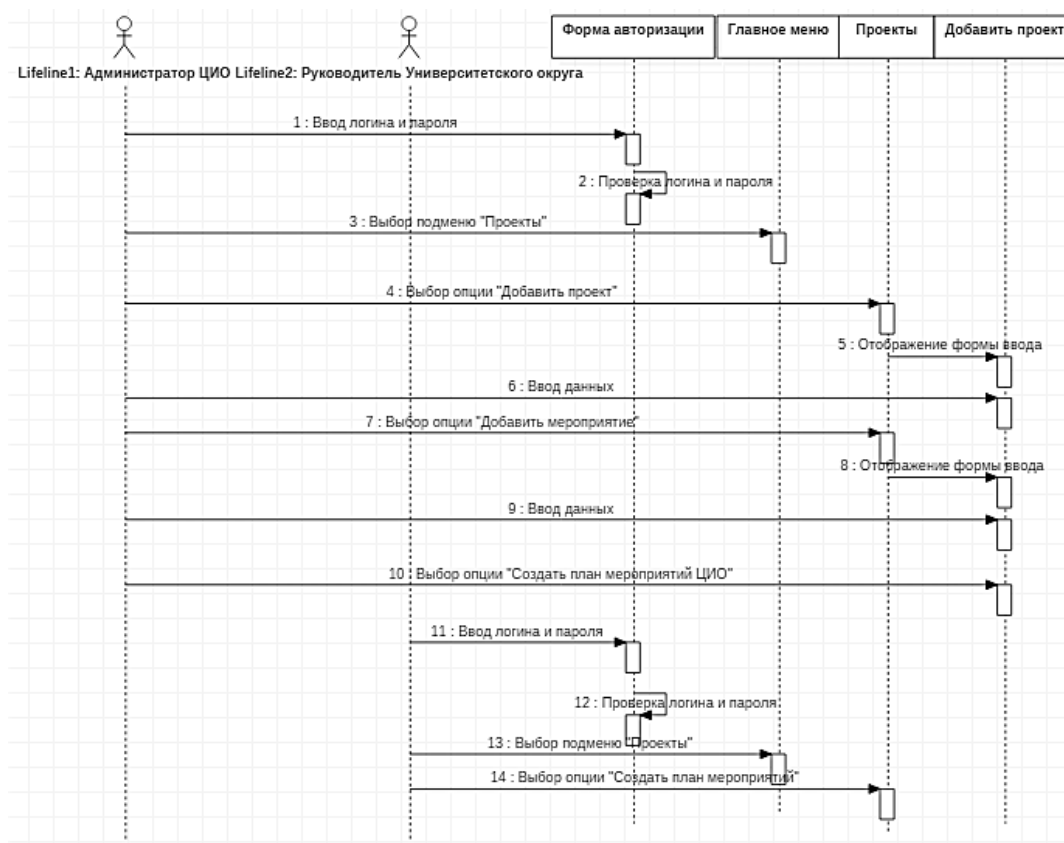


Рис. №9 Диаграмма последовательности действий бизнес-процесса
«Формирование плана мероприятий»

Таблица №3. Описание действий бизнес-процесса «Формирование плана мероприятий»

Номер сообщения	Объект - отправитель сообщения	Объект - получатель сообщения	Название
1	Администратор ЦИО	Форма авторизации	Ввод логина и пароля
2	Форма авторизации	Форма авторизации	Проверка логина и пароля
3	Администратор ЦИО	Главное меню	Выбор подменю «Проекты»
4	Администратор ЦИО	Проекты	Выбор опции «Добавить

			проект»
5	Проекты	Добавить проект	Отражение формы ввода
6	Администратор ЦИО	Добавить проект	Ввод данных
7	Администратор ЦИО	Добавить проект	Выбор опции «Добавить мероприятие»
8	Проекты	Добавить проект	Отражение формы ввода
9	Администратор ЦИО	Добавить проект	Ввод данных
10	Администратор ЦИО	Добавить проект	Выбор опции «Создать план мероприятий ЦИО»
11	Руководитель Университетского округа	Форма авторизации	Ввод логина и пароля
12	Форма авторизации	Форма авторизации	Проверка логина и пароля
13	Руководитель Университетского округа	Главное меню	Выбор подменю «Проекты»
14	Руководитель Университетского округа	Проекты	Выбор опции «Создать план мероприятий»

Диаграмма кооперации бизнес-процесса «Формирование плана мероприятий»

Диаграмма кооперации (рис.10) отображает структуру взаимодействия объектов (Администратор ЦИО, Руководитель Университетского округа Форма авторизации, Главное меню, Проекты, Добавить проект) в системе.



Рис. №10. Диаграмма кооперации бизнес-процесса «Формирование плана мероприятий ЦИО»

2.2.2.3. Проектирование бизнес-процесса «Формирование отчета ЦИО»

Проектирование бизнес-процесса «Формирование отчета ЦИО» реализовано с помощью 4-х диаграмм: диаграммы декомпозиции бизнес-процесса «Формирование отчета ЦИО», диаграммы деятельности бизнес-процесса «Формирование отчета ЦИО», диаграммы последовательности действий бизнес-процесса «Формирование отчета ЦИО», диаграммы кооперации бизнес-процесса «Формирование отчета ЦИО».

Диаграмма декомпозиции бизнес-процесса «Формирование отчета ЦИО»

Диаграмма декомпозиции бизнес-процесса «Формирование отчета ЦИО» (рис.11) показывает, что бизнес-процесс «Формирование отчета ЦИО» состоит из таких подпроцессов, как сбор данных для отчета ЦИО, формирование ежегодного отчета Университетского округа, формирование итогового отчета Университетского округа.

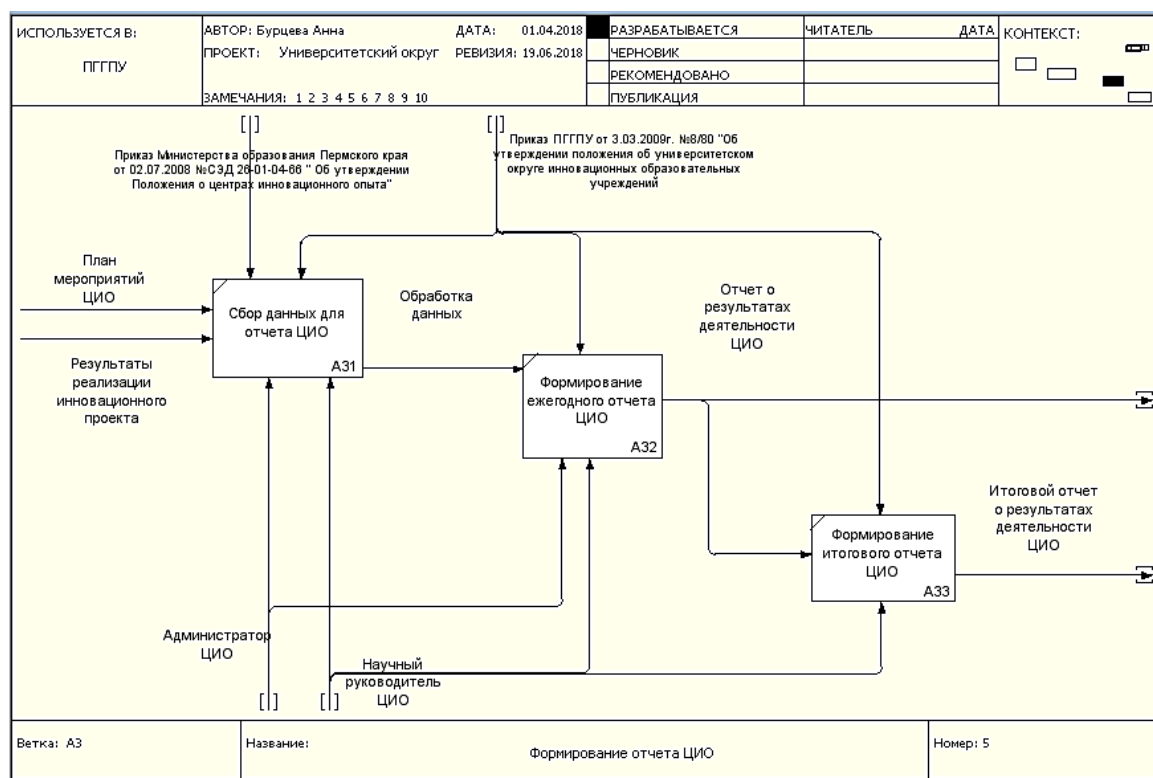


Рис.11 Диаграмма декомпозиции бизнес-процесса «Формирование отчета ЦИО»

Диаграмма деятельности бизнес-процесса «Формирование отчета ЦИО»

Для более детального описания бизнес- процесса «Формирование отчета ЦИО» используется диаграмма деятельности (рис.12), на которой представлено разложение на составные части бизнес-процесса «Формирование отчета ЦИО», а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются

между собой потоками от выходов одного узла к входам другого, с указанием их исполнителей (администратор ЦИО, система).

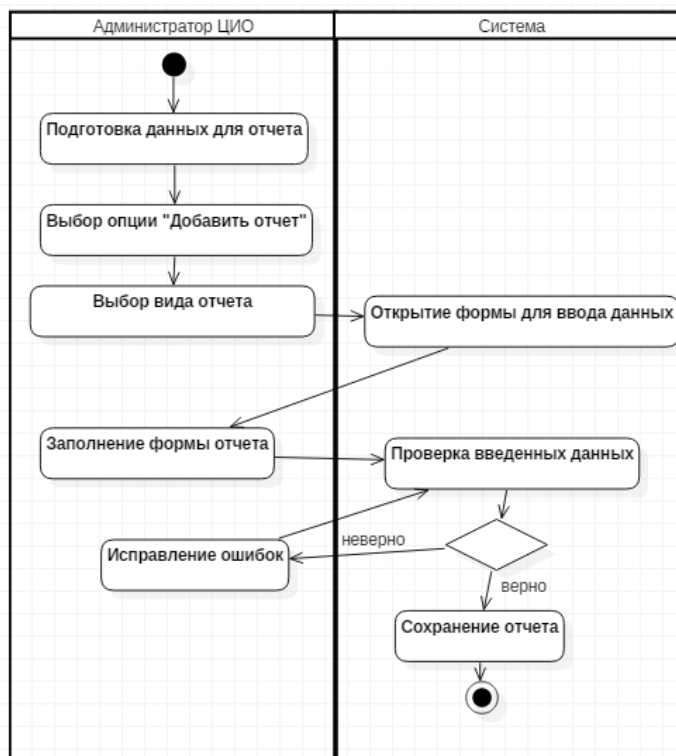


Рис.№12 «Диаграмма деятельности бизнес-процесса «Формирование отчета ЦИО»

Диаграмма последовательности действий бизнес-процесса «Формирование отчета ЦИО»

Для представления взаимодействий объектов в системе используют диаграммы последовательности и кооперации. Диаграмма последовательности действий (рис.13) отображает обмен сообщениями (табл.4) между объектами во времени.

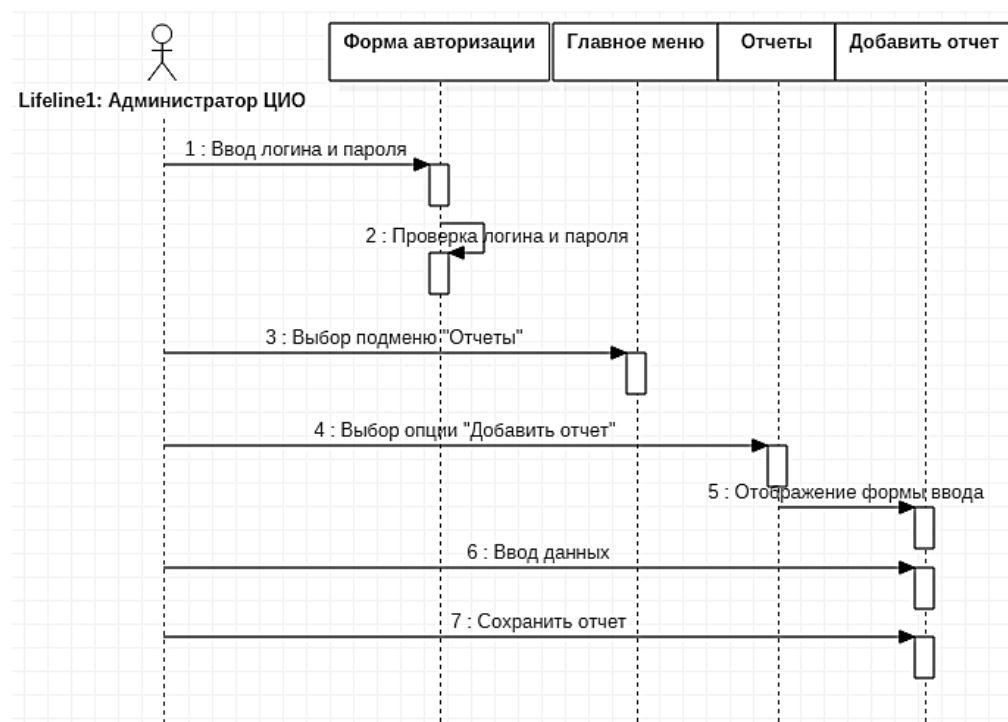


Рис. №13 Диаграмма последовательности действий бизнес-процесса
«Формирование отчета ЦИО»

Таблица №4. Описание действий бизнес-процесса «Формирование отчета»

Номер сообщения	Объект - отправитель сообщения	Объект - получатель сообщения	Название
1	Администратор ЦИО	Форма авторизации	Ввод логина и пароля
2	Форма авторизации	Форма авторизации	Проверка логина и пароля
3	Администратор ЦИО	Главное меню	Выбор подменю «Отчеты»
4	Администратор ЦИО	Отчеты	Выбор опции «Добавить отчет»
5	Отчеты	Добавить отчет	Отражение формы ввода
6	Администратор ЦИО	Добавить отчет	Ввод данных

7	Администратор ЦИО	Добавить отчет	Сохранить отчет
---	----------------------	----------------	-----------------

Диаграмма кооперации бизнес-процесса «Формирование отчета ЦИО»

Диаграмма кооперации (рис.14) отображает структуру взаимодействия объектов (Администратор ЦИО, Форма авторизации, Главное меню, Отчет, Добавить отчет) в системе.

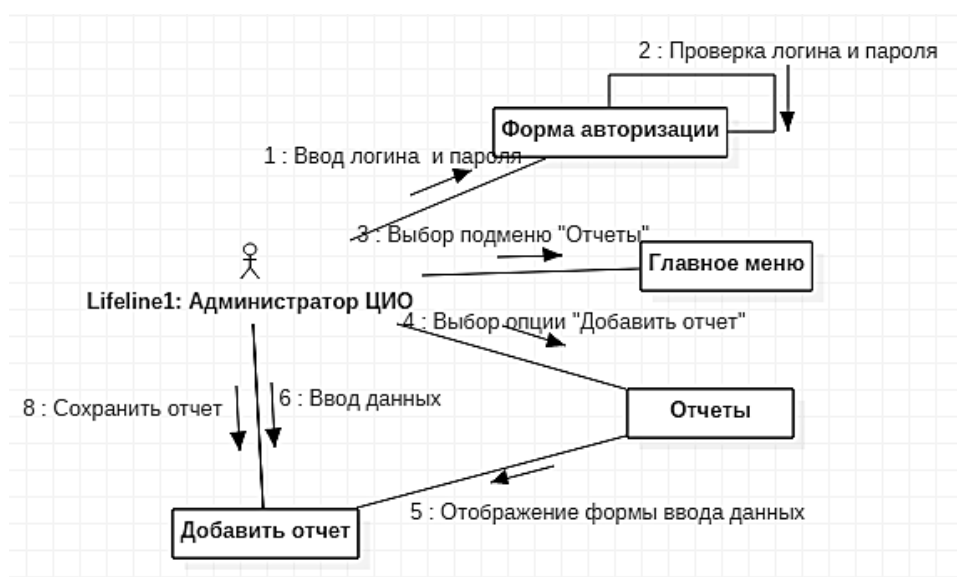


Рис. №14. Диаграмма кооперации бизнес-процесса «Формирование отчета ЦИО»

2.2.2.4. Проектирование бизнес-процесса «Формирование отчета Университетского округа»

Проектирование бизнес-процесса «Формирование отчета Университетского округа» реализовано с помощью 4-х диаграмм: диаграммы декомпозиции бизнес-процесса «Формирование отчета Университетского округа», диаграммы деятельности бизнес-процесса «Формирование отчета Университетского округа», диаграммы последовательности действий бизнес-процесса «Формирование отчета Университетского округа», диаграммы

кооперации бизнес-процесса «Формирование отчета Университетского округа».

Диаграмма декомпозиции бизнес-процесса «Формирование отчета Университетского округа»

Диаграмма декомпозиции бизнес-процесса «Формирование отчета Университетского округа» (рис.15) показывает, что бизнес-процесс «Формирование отчета Университетского округа» состоит из таких подпроцессов, как сбор данных для отчета Университетского округа, формирование ежегодного отчета Университетского округа, формирование итогового отчета Университетского округа.

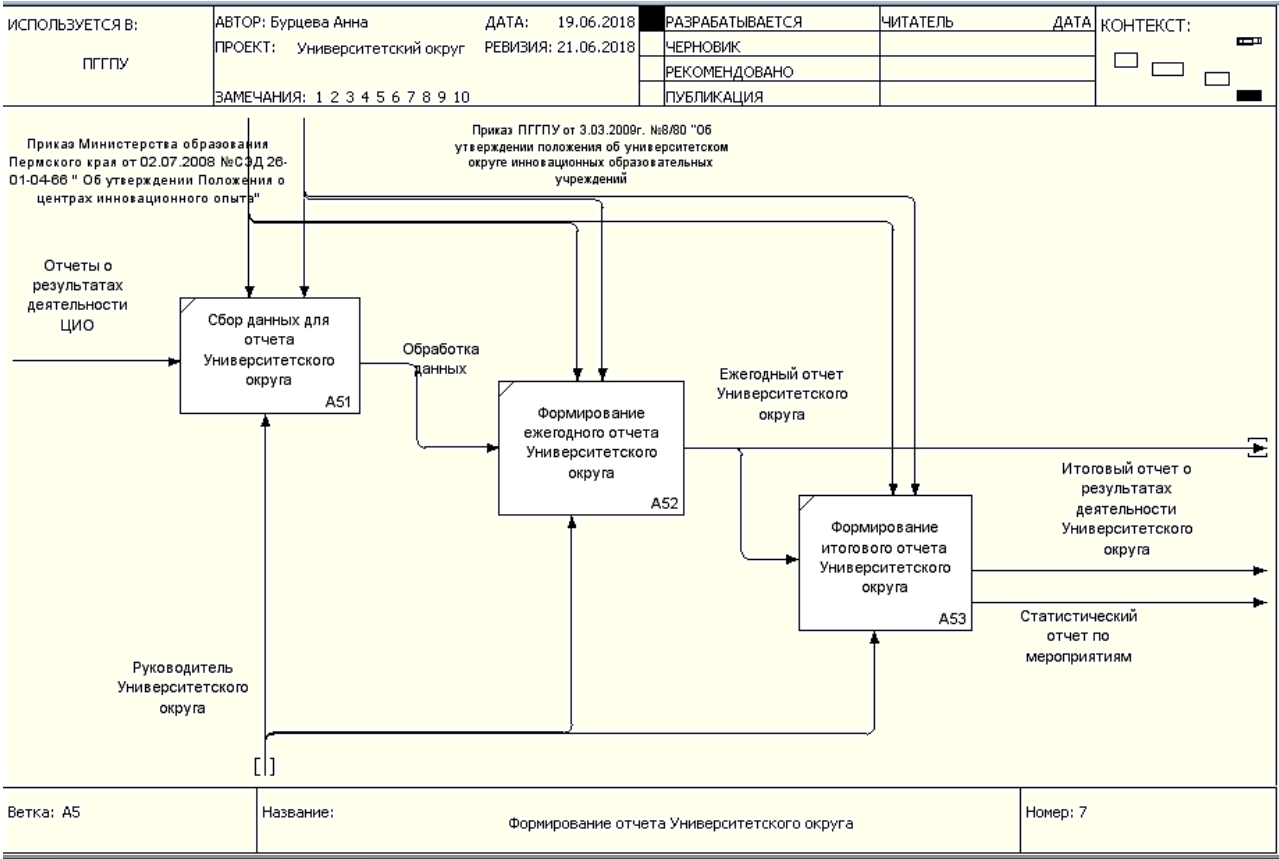


Рис.15 «Диаграмма декомпозиции бизнес-процесса «Формирование отчета Университетского округа»

Диаграмма деятельности бизнес-процесса «Формирование отчета Университетского округа»

Для более детального описания бизнес- процесса «Формирование отчета Университетского округа» используется диаграмма деятельности (рис.16), на которой представлено разложение на составные части бизнес-процесса «Формирование отчета Университетского округа», а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются между собой потоками от выходов одного узла к входам другого, с указанием их исполнителей (руководитель Университетского округа, система).

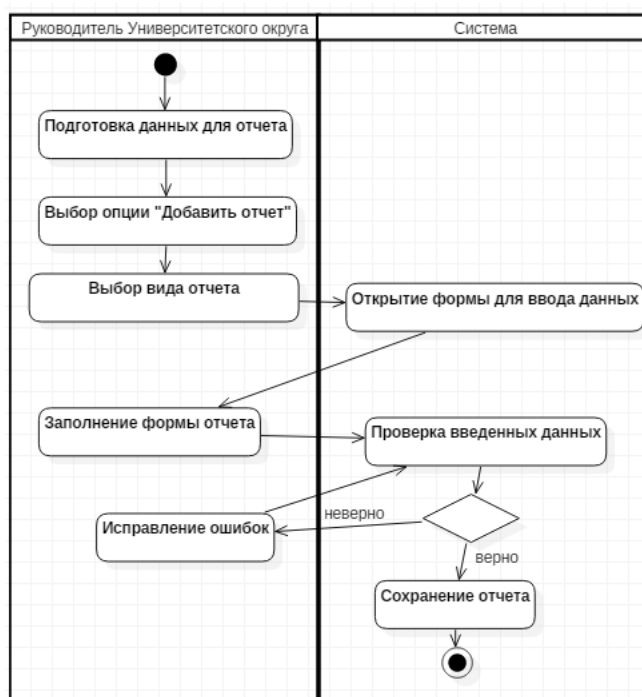


Рис.№16 «Диаграмма деятельности бизнес-процесса «Формирование отчета
Университетского округа»

Диаграмма последовательности действий бизнес-процесса «Формирование отчета Университетского округа»

Для представления взаимодействий объектов в системе используют диаграммы последовательности и кооперации. Диаграмма последовательности действий (рис.17) отображает обмен сообщениями(табл.5) между объектами во времени.



Рис. №17 Диаграмма последовательности действий бизнес-процесса
«Формирование отчета Университетского округа»

Таблица №5. Описание действий бизнес-процесса «Формирование отчета
Университетского округа»

Номер сообщения	Объект - отправитель сообщения	Объект - получатель сообщения	Название
1	Руководитель Университетского округа	Форма авторизации	Ввод логина и пароля
2	Форма авторизации	Форма авторизации	Проверка логина и пароля

3	Руководитель Университетского округа	Главное меню	Выбор подменю «Университетский округ»
4	Руководитель Университетского округа	Университетский округ	Выбор подменю «Отчеты»
6	Руководитель Университетского округа	Отчеты	Выбор опции «Добавить отчет»
7	Отчеты	Добавить отчет	Отражение формы ввода
8	Руководитель Университетского округа	Добавить отчет	Ввод данных
9	Руководитель Университетского округа	Добавить отчет	Сохранить отчет

**Диаграмма кооперации бизнес-процесса «Формирование отчета
Университетского округа»**

Диаграмма кооперации (рис.18) отображает структуру взаимодействия объектов (Руководитель Университетского округа, Форма авторизации, Главное меню, Университетский округ, Отчет, Добавить отчет) в системе.

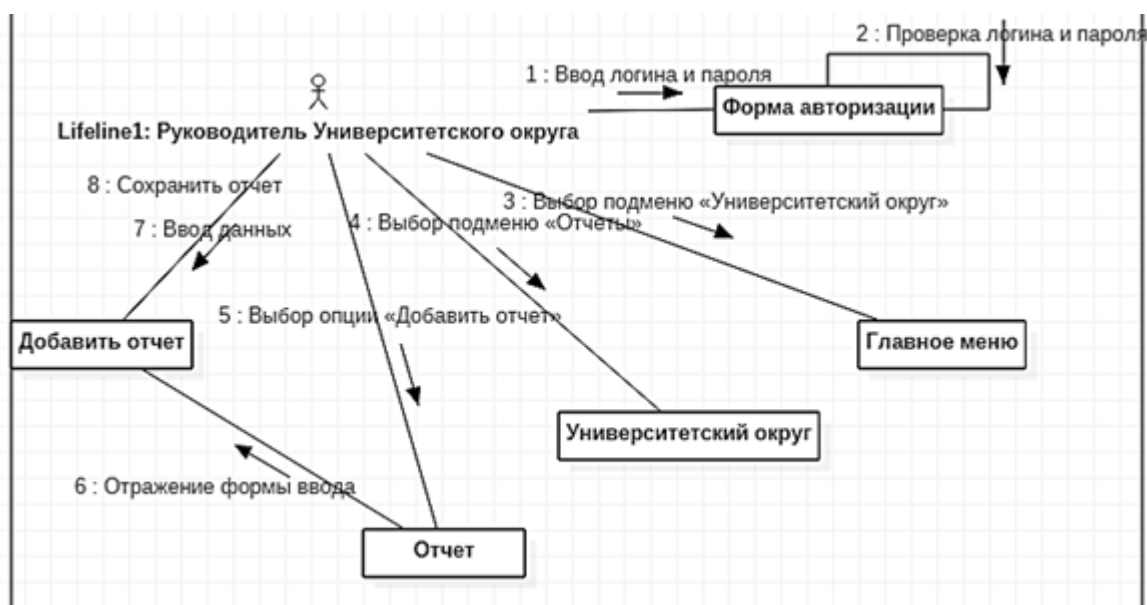


Рис. №18. Диаграмма кооперации бизнес-процесса «Формирование отчета Университетского округа»

2.2.3. Объектное проектирование

Объектное проектирование состоит из выделения объектов и атрибутов, связей между объектами, ER-диаграммы.

2.2.3.1. Объекты и атрибуты

На основании описания предметной области можно выделить следующие объекты, их атрибуты и первичные ключи (табл.6).

Таблица №6. Объекты и атрибуты.

Объект	Атрибут	Первичный ключ
УНИВЕРСИТЕТСКИЙ ОКРУГ	Код_УО	Код_УО
	Название_УО	
	Фамилия_рук	

	Имя_рук	
	Отчество_рук	
	Адрес_УО	
	Телефон_УО	
	E-mail	
	Сайт_УО	
ЦИО	Код_ЦИО	Код_ЦИО
	Название_обр_организаци и	
	Фамилия_дир	
	Имя_дир	
	Отчество_дир	
	Фамилия_рук_ЦИО	
	Имя_рук_ЦИО	
	Отчество_рук_ЦИО	
	Адрес_ЦИО	
	Телефон_ЦИО	
	Сайт	
	Код_УО	
ПРОЕКТ	Код_проекта	Код_проекта
	Название_проекта	
	Фамилия_автора_проекта	
	Имя_автора_проекта	
	Отчество_автора_проекта	
	Фамилия_рук_проекта	

	Имя_рук_проекта	
	Отчество_рук_проекта	
	Ссылка_на_раздел_на_сайте	
	Актуальность	
	Цель	
	Задачи	
	Краткое_обоснование_инновационности_проекта	
	Ожидаемые_продукты_проекта	
	На_разработку_какого_инного_инновационного_контента_направлен_проект	
	Способы_апробации_и_диссеминации_результатов	
	Риски_и_пути_преодоления	
	Механизм_самооценки	
	Код_ЦИО	
МЕРОПРИЯТИЕ	Код_мероприятия	Код_мероприятия
	Название	
	Дата	
	Место_проведения	
	Категория_участников	
	Ответственный	
	Код_проекта	

РЕЗУЛЬТАТ_ПРОЕКТА	Код_результата_проекта	Код_результата_проекта
	Нормативно_правовое_обеспечение	
	Кадровое_обеспечение_проекта	
	Информационное_обеспечение_проекта	
	Опыт_успешно_реализованных_проектов	
	Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов	
	Описание_текущей_актуальности_продуктов	
	Продукты	
	Описание_методов_и_критериев_мониторинга_качества_проекта	
	Результаты_самооценки_мониторинга	
	Перечень_достигнутых_результатов	
	Код_проекта	

Таблица №7. Спецификация объектов.

Объект	Атрибут	Тип атрибута
УНИВЕРСИТЕТСКИЙ ОКРУГ	Код_УО	идентификационный

		атрибут
	Название_УО	описательный атрибут
	Фамилия_рук	описательный атрибут
	Имя_рук	описательный атрибут
	Отчество_рук	описательный атрибут
	Адрес_УО	описательный атрибут
	Телефон_УО	описательный атрибут
	E-mail	описательный атрибут
	Сайт	описательный атрибут
ЦИО	Код_ЦИО	идентификационный атрибут
	Название_обр_организации	описательный атрибут
	Фамилия_дир	описательный атрибут
	Имя_дир	описательный атрибут
	Отчество_дир	описательный атрибут
	Фамилия_рук_ЦИО	описательный атрибут

	Имя_рук_ЦИО	описательный атрибут
	Отчество_рук_ЦИО	описательный атрибут
	Адрес_ЦИО	описательный атрибут
	Телефон_ЦИО	описательный атрибут
	Сайт	описательный атрибут
	Код_УО	идентификационный атрибут
ПРОЕКТ	Код_проекта	идентификационный атрибут
	Название_проекта	описательный атрибут
	Фамилия_автора_проекта	описательный атрибут
	Имя_автора_проекта	описательный атрибут
	Отчество_автора_проекта	описательный атрибут
	Фамилия_рук_проекта	описательный атрибут
	Имя_рук_проекта	описательный атрибут
	Отчество_рук_проекта	описательный атрибут
	Ссылка_на_раздел_на_сайт	описательный

	е	атрибут
	Актуальность	описательный атрибут
	Цель	описательный атрибут
	Задачи	описательный атрибут
	Краткое_обоснование_инн овационности_проекта	описательный атрибут
	Ожидаемые_продукты_про екта	описательный атрибут
	На_разработку_какого_ино го_инновационного_конте нта_направлен_проект	описательный атрибут
	Способы_апробации_и_ди ссеминации_результатов	описательный атрибут
	Риски_и_пути_преодолен я	описательный атрибут
	Механизм_самооценки	описательный атрибут
	Код_ЦИО	идентификационный атрибут
МЕРОПРИЯТИЕ	Код_мероприятия	идентификационный атрибут
	Название	описательный атрибут
	Дата	описательный

		атрибут
	Место_проведения	описательный атрибут
	Категория_участников	описательный атрибут
	Ответственный	описательный атрибут
	Код_проекта	идентификационный атрибут
РЕЗУЛЬТАТ_ПРОЕКТА	Код_результата_проекта	идентификационный атрибут
	Нормативно_правовое_обеспечение	описательный атрибут
	Кадровое_обеспечение_проекта	описательный атрибут
	Информационное_обеспечение_проекта	описательный атрибут
	Опыт_успешно_реализованных_проектов	описательный атрибут
	Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов	описательный атрибут
	Описание_текущей_актуальности_продуктов	описательный атрибут
	Продукты	описательный атрибут
	Описание_методов_и_критериев_мониторинга_качества	описательный атрибут

	ва_проекта	
	Результаты_самооценки_мониторинга	описательный атрибут
	Перечень_достигнутых_результатов	описательный атрибут
	Код_проекта	идентификационный атрибут

Таблица №8. Спецификация атрибутов.

Объект	Атрибут	Вид атрибута	Свойства
УНИВЕРСИТЕТСКИЙ ОКРУГ	Код_УО	счетчик	-
	Название_УО	текстовый	30 символов
	Фамилия_рук	текстовый	20 символов
	Имя_рук	текстовый	20 символов
	Отчество_рук	текстовый	20 символов
	Адрес_УО	текстовый	30 символов
	Телефон_УО	числовой	11 символов
	E-mail	текстовый	30 символов
	Сайт	текстовый	30 символов
ЦИО	Код_ЦИО	счетчик	-
	Название_обр_организации	текстовый	30 символов
	Фамилия_дир	текстовый	20 символов
	Имя_дир	текстовый	20 символов
	Отчество_дир	текстовый	20 символов

	Фамилия_рук_ЦИ О	текстовый	20 символов
	Имя_рук_ЦИО	текстовый	20 символов
	Отчество_рук_ЦИ О	текстовый	20 символов
	Адрес_ЦИО	текстовый	30 символов
	Телефон_ЦИО	числовой	11 символов
	Сайт	текстовый	30 символов
	Код_УО	счетчик	-
ПРОЕКТ	Код_проекта	счетчик	-
	Название_проекта	текстовый	50 символов
	Фамилия_автора_ проекта	текстовый	50 символов
	Имя_автора_прое кта	текстовый	50 символов
	Отчество_автора_ проекта	текстовый	50 символов
	Фамилия_рук_про екта	текстовый	50 символов
	Имя_рук_проекта	текстовый	50 символов
	Отчество_рук_про екта	текстовый	50 символов
	Ссылка_на_раздел _на_сайте	текстовый	50 символов
	Актуальность	текстовый	200 символов
	Цель	текстовый	200 символов
	Задачи	текстовый	200 символов

	Краткое_обоснование_инновационности_проекта	текстовый	200 символов
	Ожидаемые_продукты_проекта	текстовый	200 символов
	На_разработку_какого_иного_инновационного_контента_направлен_проект	текстовый	200 символов
	Способы_апробации_и_диссеминации_результатов	текстовый	200 символов
	Риски_и_пути_преодоления	текстовый	200 символов
	Механизм_самооценки	текстовый	200 символов
	Код_ЦИО	счетчик	-
МЕРОПРИЯТИЕ	Код_мероприятия	счетчик	-
	Название	текстовый	30 символов
	Дата	дата/время	-
	Место_проведения	текстовый	30 символов
	Категория_участников	текстовый	30 символов
	Ответственный	текстовый	30 символов
	Код_проекта	счетчик	-
РЕЗУЛЬТАТ_ПРО	Код_результата_п	счетчик	-

ЕКТА	проекта		
	Нормативно_правовое_обеспечение	текстовый	200 символов
	Кадровое_обеспечение_проекта	текстовый	200 символов
	Информационное_обеспечение_проекта	текстовый	200 символов
	Опыт_успешно_реализованных_проектов	текстовый	200 символов
	Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов	текстовый	200 символов
	Описание_текущей_актуальности_продуктов	текстовый	200 символов
	Продукты	текстовый	200 символов
	Описание_методов_и_критериев_мониторинга_качества_проекта	текстовый	200 символов
	Результаты_самооценки_мониторинга	текстовый	200 символов
	Перечень_достигнутых_результатов	текстовый	200 символов
	Код_проекта	счетчик	-

2.2.3.2. Связи между объектами

Для связи объектов друг с другом используется семантическая методология. В таблице 9 представлены связи между объектами, показатели кардинальности связей и степени участия объектов в данной связи.

Таблица 9. Спецификация связей.

Наименование связи	Объекты	Показатель кардинальности	Степень участия
Входят	УНИВЕРСИТЕТСКИЙ ОКРУГ ЦИО	1:M	Полная Полная
Реализует	ЦИО ПРОЕКТ	1:M	Полная Полная
Содержит	ПРОЕКТ МЕРОПРИЯТИЕ	1:M	Полная Полная
Имеет	ПРОЕКТ РЕЗУЛЬТАТ_ПРОЕКТА	1:1	Полная Полная

2.2.4. ER–диаграмма предметной области «Университетский округ»

На основании выделенных объектов (табл. 7), их атрибутов (табл. 8) и связей между ними (табл. 9) получаем ER-диаграмму предметной области «Университетский округ» (рис. 19).

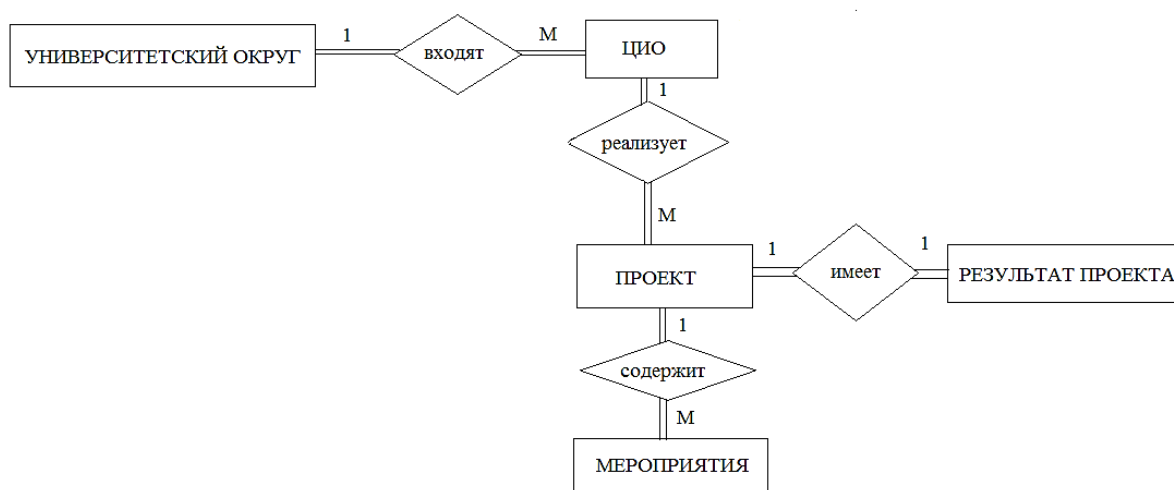


Рис.19. ER-диаграмма предметной области «Университетский округ»

Созданная концептуальная модель информационной системы «Университетский округ» является источником информации для логического проектирования.

2.3. Логическое проектирование

Логическое проектирование – это второй этап проектирования информационной системы, включающий в себя:

1. упрощение концептуальной схемы;
2. разработку реляционной модели;
3. обеспечение целостности базы данных.

2.3.1. Разработка реляционной модели данных

Разработка реляционной схемы включает в себя:

- 1) разработка таблиц с атрибутами;
- 2) определение функциональных зависимостей между атрибутами в каждой таблице;
- 3) приведение каждой таблицы к третьей нормальной форме;
- 4) разработка реляционной схемы данных;
- 5) разработка отчетов;

б) разработка запросов.

2.3.1.1. Таблицы и атрибуты

При преобразовании концептуальных структур данных в реляционные структуры объекты с атрибутами переходят в таблицы с атрибутами (табл.10).

Таблица 10.Таблицы и атрибуты.

Объект	Объект	Атрибут	Первичный ключ
УНИВЕРСИТЕТ СКИЙ ОКРУГ	УНИВЕРСИТЕТС КИЙ ОКРУГ	Код_УО	Код_УО
		Название_УО	
		Фамилия_рук	
		Имя_рук	
		Отчество_рук	
		Адрес_УО	
		Телефон_УО	
		E-mail	
		Сайт	
ЦИО	ЦИО	Код_ЦИО	Код_ЦИО
		Название_обр_орга низации	
		Фамилия_дир	
		Имя_дир	
		Отчество_дир	

		Фамилия_рук_ЦИО	
		Имя_рук_ЦИО	
		Отчество_рук_ЦИО	
		Адрес_ЦИО	
		Телефон_ЦИО	
		Сайт	
		Код_УО	
ПРОЕКТ	ПРОЕКТ	Код_проекта	Код_проекта
		Название_проекта	
		Фамилия_автора_пр оекта	
		Имя_автора_проект а	
		Отчество_автора_пр оекта	
		Фамилия_рук_прое кта	
		Имя_рук_проекта	
		Отчество_рук_прое кта	
		Ссылка_на_раздел_ на_сайте	
		Актуальность	
		Цель	
		Задачи	
		Краткое_обоснован ие_инновационност	

		и_проекта	
		Ожидаемые_продукты_проекта	
		На_разработку_какого_иного_инновационного_контента_направлен_проект	
		Способы_апробации_и_диссеминации_результатов	
		Риски_и_пути_преодоления	
		Механизм_самооценки	
		Код_ЦИО	
МЕРОПРИЯТИЕ	МЕРОПРИЯТИЕ	Код_мероприятия	Код_мероприятия
		Название	
		Дата	
		Место_проведения	
		Категория_участников	
		Ответственный	
		Код_проекта	
РЕЗУЛЬТАТ_ПРОЕКТА	РЕЗУЛЬТАТ_ПРОЕКТА	Код_результата_проекта	Код_отчета
		Нормативно_правовое_обеспечение	

		Кадровое_обеспечение_проекта	
		Информационное_обеспечение_проекта	
		Опыт_успешно_реализованных_проектов	
		Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов	
		Описание_текущей_актуальности_продуктов	
		Продукты	
		Описание_методов_и_критериев_мониторинга_качества_проекта	
		Результаты_самооценки_мониторинга	
		Перечень_достигнутых_результатов	
		Код_проекта	

2.3.1.2. Функциональные зависимости

Рассмотрим функциональные зависимости между атрибутами каждого из отношений (табл.10).

Ф31. Код_УО→Название_УО, Фамилия_рук, Имя_рук, Отчество_рук, Адрес_УО, Тел_УО, E-mail, Сайт.

Ф32. Код_ЦИО→Название_обр_организации, Фамилия_дир, Имя_дир, Отчество_дир, Фамилия_рук_ЦИО, Имя_рук_ЦИО, Отчество_рук_ЦИО, Адрес_ЦИО, Телефон_ЦИО, Сайт.

Ф33.Код_проекта→Название_проекта, Фамилия_автора_проекта, Имя_автора_проекта, Отчество_автора_проекта, Фамилия_рук_проекта, Имя_рук_проекта,Отчество_рук_проекта,Ссылка_на_раздел_на_сайте,Актуальность,Цель,Задачи,Краткое_обоснование_инновационности_проекта, Ожидаемые_продукты_проекта,На_разработку_какого_иного_инновационного_контента_направлен_проект,Способы_апробации_и_диссеминации_результатов,Риски_и_пути_преодоления,Механизм_самооценки.

Ф34.Код_мероприятия→Название, Дата, Место_проведения, Категория_участников, Ответственный.

Ф35.Код_результата_проекта→Нормативно_правовое_обеспечение,Кадровое_обеспечение_проекта,Информационное_обеспечение_проекта,Опыт_успешно_реализованных_проектов,Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов,Описание_текущей_актуальности_продуктов,Продукты,Описание_методов_и_критериев_мониторинга_качества_проекта,Результаты_самооценки_мониторинга,Перечень_достигнутых_результатов.

2.3.1.3. Нормализация отношений

Каждое отношение из таблицы 10 приведем к третьей нормальной форме.

Рассмотрим полученные таблицы (табл.10).

Таблицы УНИВЕРСИТЕТСКИЙ ОКРУГ, ЦИО, ПРОЕКТ, МЕРОПРИЯТИЕ, РЕЗУЛЬТАТ_ПРОЕКТА имеют простые атрибуты (табл. 13), поэтому они приведены к 1НФ.

Таблицы УНИВЕРСИТЕТСКИЙ ОКРУГ, ЦИО, ПРОЕКТ, МЕРОПРИЯТИЕ, РЕЗУЛЬТАТ_ПРОЕКТА находятся в 1НФ и имеют простые первичные ключи (табл. 10), поэтому они автоматически приведены ко 2НФ.

Таблица УНИВЕРСИТЕТСКИЙ ОКРУГ находится во 2НФ и атрибуты (Название_УО, Фамилия_рук, Имя_рук, Отчество_рук, Адрес_УО, Тел_УО, E-mail, Сайт) функционально не зависят друг от друга (Ф31), поэтому таблица УНИВЕРСИТЕТСКИЙ ОКРУГ приведена к 3НФ.

Таблица ЦИО находится во 2НФ и атрибуты (Название_организации, Фамилия_дир, Имя_дир, Отчество_дир, Фамилия_рук_ЦИО, Имя_рук_ЦИО, Отчество_рук_ЦИО, Адрес_ЦИО, Телефон_ЦИО, Сайт) функционально не зависят друг от друга (Ф32), поэтому таблица ЦИО приведена к 3НФ.

Таблица ПРОЕКТ находится во 2НФ и атрибуты (Название_проекта, Фамилия_автора_проекта, Имя_автора_проекта, Отчество_автора_проекта, Фамилия_рук_проекта, Имя_рук_проекта, Отчество_рук_проекта, Ссылка_на_раздел_на_сайте, Актуальность, Цель, Задачи, Краткое_обоснование_инновационности_проекта, Ожидаемые_продукты_проекта, На_разработку_какого_инного_инновационного_контента_направлен_проект, Способы_апробации_и_диссеминации_результатов, Риски_и_пути_преодоления, Механизм_самооценки) функционально не зависят друг от друга (Ф33), поэтому таблица ПРОЕКТ приведена к 3НФ.

Таблица МЕРОПРИЯТИЕ находится во 2НФ и атрибуты (Название, Дата, Место_проведения, Категория_участников, Ответственный) функционально не зависят друг от друга (Ф34), поэтому таблица МЕРОПРИЯТИЕ приведена к 3НФ.

Таблица РЕЗУЛЬТАТ_ПРОЕКТА находится во 2НФ и атрибуты (Нормативно_правовое_обеспечение, Кадровое_обеспечение_проекта, Информационное_обеспечение_проекта, Опыт_успешно_реализованных_проектов, Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов, Описание_текущей_актуальности_продуктов, П

- Класс Проект полностью зависит от класса ЦИО, поэтому между ними определим отношение композиции.
- Класс Мероприятие не зависит от класса Проект и может быть повторно использован самостоятельно, независимо от класса Проект, поэтому создадим между ними отношение ассоциации.
- Класс Результат проекта полностью зависит от класса Проект, поэтому между ними определим отношение композиции.
- Класс Параметры работы с отчетом ЦИО использует класс Результат проекта как параметр своей операции: между ними определим отношение зависимости.
- Класс Параметры работы с отчетом УО использует класс Результат проекта как параметр своей операции: между ними определим отношение зависимости.
- Класс Создать план мероприятий ЦИО использует класс Мероприятие как параметр своей операции: между ними определим отношение зависимости.
- Класс Создать план мероприятий УО использует класс Мероприятие как параметр своей операции: между ними определим отношение зависимости.
- Класс Создать выборку мероприятий за определенный период использует класс Мероприятие как параметр своей операции: между ними определим отношение зависимости.

2.3.1.5. Реляционные связи

Для связей таблиц в реляционной модели данных используется механизм первичных и внешних ключей. Механизм заключается в следующем:

- 1) Из двух таблиц, участвующих в бинарной связи, с учетом спецификации связи (табл.9) и методики определения статуса таблиц одну назначим родительской, другую – дочерней;
 - 2) В дочернюю таблицу добавим атрибут – внешний ключ (копия первичного ключа родительской таблицы);
 - 3) Создадим связь между таблицами по одноименному полю.
- В таблице 11 приведены все связи.

Таблица №11. Реляционные связи.

Связь	Таблицы	Показатель кардинальности	Статус таблицы	Ключи
Входят	УНИВЕРСИТЕТСКИЙ ОКРУГ ЦИО	1:M	Родительская Дочерняя	Код_УО(ПК) Код_ЦИО(ПК) Код_УО(ВК)
Реализует	ЦИО ПРОЕКТ	1:M	Родительская Дочерняя	Код_ЦИО (ПК) Код_проекта(ПК) Код_ЦИО(ВК)
Содержит	ПРОЕКТ МЕРОПРИЯТИЕ	1:M	Родительская Дочерняя	Код_проекта(ПК) Код_мероприятия(ПК) Код_проекта(ВК)
Имеет	ПРОЕКТ РЕЗУЛЬТАТ_ПРОЕКТА	1:1	Родительская Дочерняя	Код_проекта(ПК) Код_результата_проекта(ПК)

				Код_проекта (ВК)
--	--	--	--	---------------------

2.3.1.6. Разработка отчетных документов и форм для ввода

Диаграммы потоков данных используются для описания документооборота и обработки информации. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней (контекстные диаграммы) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процессы становятся элементарными и детализировать их далее невозможно.

Для описания внешних по отношению к системе источников данных, потоков и хранилищ, к которым осуществляется доступ, используют диаграммы потоков. На рисунках 21-26 представлены диаграммы потоков для информационной системы «Университетский округ».

Диаграммы потоков:

- Для контекстной диаграммы

На рисунке 21 представлена диаграмма потоков для контекстной диаграммы Университетского округа, которая описывает деятельность Университетского округа, информация на входе – материально-техническое и кадровое обеспечение работы Университетского округа, результаты деятельности по реализации инновационного проекта, реализация

инновационного проекта, на выходе – услуги Университетского округа, результаты деятельности. Для хранения отчетов имеется 5 хранилищ: сотрудники Центров инновационного опыта, сотрудники Университетского округа, центры инновационного опыта, отчетность ЦИО, отчетность Университетского округа.

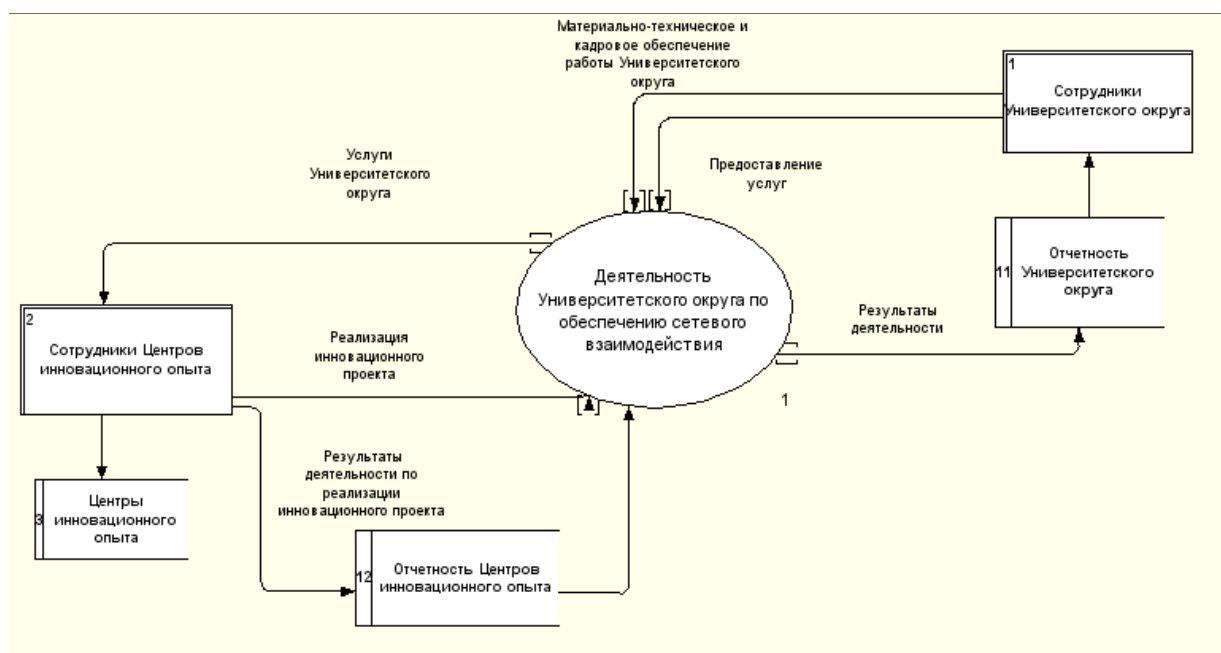


Рис.21 Диаграмма потоков для контекстной диаграммы Университетского округа

- Для диаграммы декомпозиции контекстной

На рисунке 22 представлена диаграмма потоков для декомпозиции контекстной диаграммы Университетского округа, которая раскрывает деятельность Университетского округа более детально, показывая бизнес-процессы. Для хранения информации о ЦИО имеется хранилище Центры инновационного опыта, планов мероприятий ЦИО – хранилище планы мероприятий ЦИО, планов мероприятий Университетского округа – планы мероприятий УО.

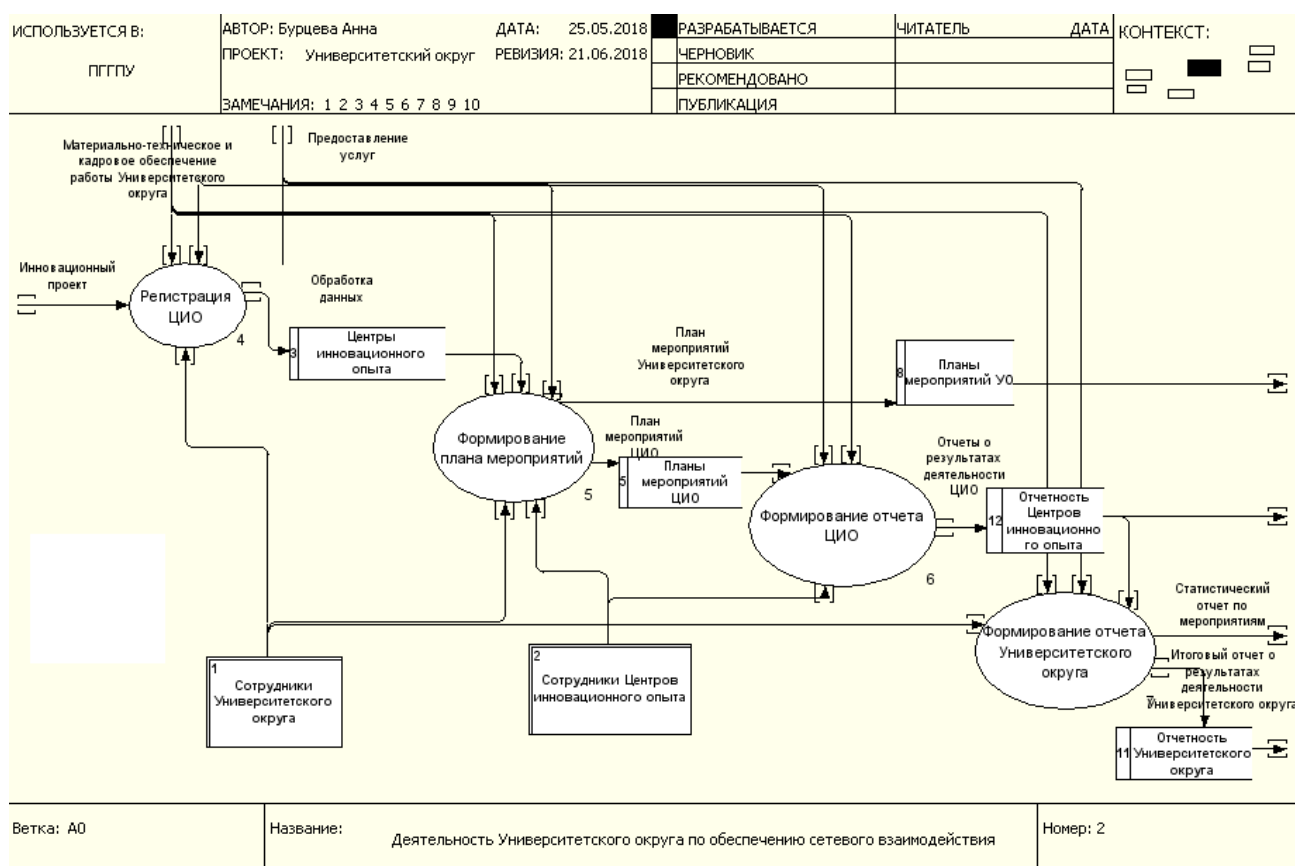


Рис.22 Диаграмма потоков для декомпозиции контекстной диаграммы
Университетского округа

- Для диаграммы декомпозиции каждого бизнес-процесса

Диаграмма потоков для бизнес-процесса «Регистрация ЦИО»

На рисунке 23 представлена диаграмма потоков для бизнес-процесса «Регистрация ЦИО», которая показывает, что бизнес-процесс «Регистрация ЦИО» состоит из таких подпроцессов, как конкурс инновационных проектов образовательных учреждений, регистрация образовательного учреждения в ИС и присвоение ему статуса ЦИО, выдача логина и пароля ЦИО для входа в ИС.

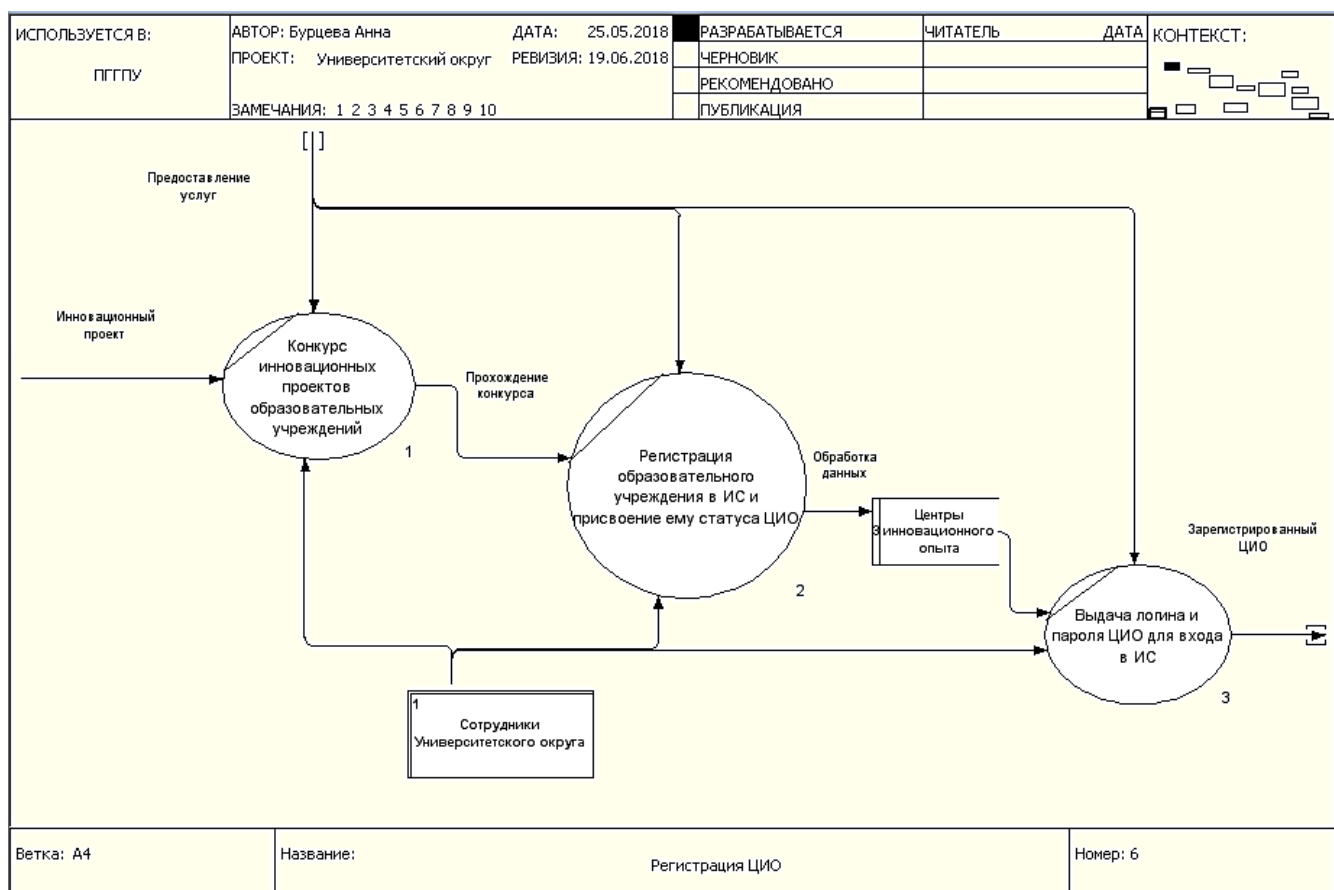


Рис.23 Диаграмма потоков для бизнес-процесса «Регистрация ЦИО»

Диаграмма потоков для бизнес-процесса «Формирование плана мероприятий»

На рисунке 24 представлена диаграмма потоков для бизнес-процесса «Формирование плана мероприятий», которая показывает, что бизнес-процесс «Формирование плана мероприятий» состоит из таких подпроцессов, как заполнение данных о проекте ЦИО, добавление мероприятий по реализации проекта ЦИО, формирование плана мероприятий проекта ЦИО, формирование плана мероприятий Университетского округа. Для хранения проектов ЦИО имеется хранилище проекты ЦИО, мероприятий ЦИО – хранилище мероприятия ЦИО.

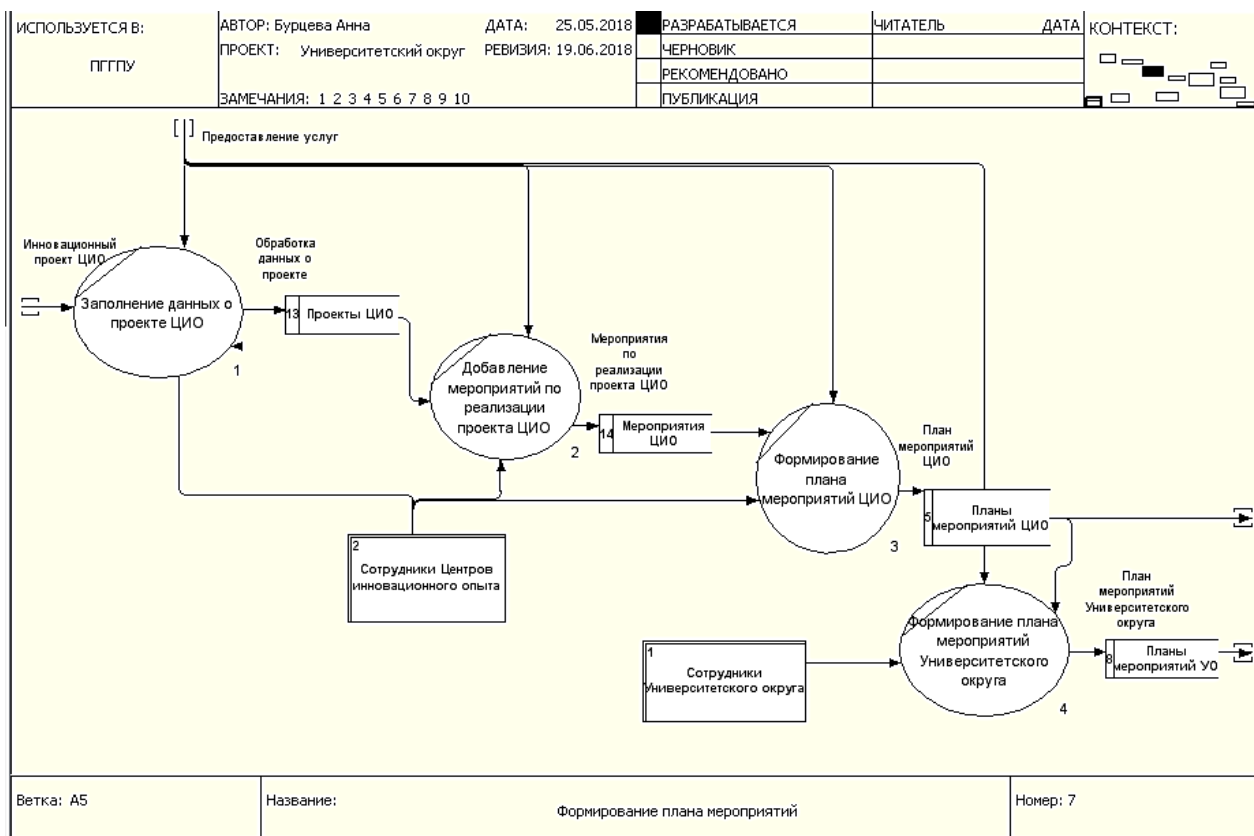


Рис.24 Диаграмма потоков для бизнес-процесса «Формирование плана мероприятий ЦИО»

Диаграмма потоков для бизнес-процесса «Формирование отчета ЦИО»

На рисунке 25 представлена диаграмма потоков для бизнес-процесса «Формирование отчета ЦИО», которая показывает, что бизнес-процесс «Формирование отчета ЦИО» состоит из таких подпроцессов, как сбор данных для отчета ЦИО, формирование ежегодного отчета ЦИО, формирование итогового отчета ЦИО. Для хранения отчетов ЦИО имеется хранилище ежегодные отчеты ЦИО, итоговых отчетов ЦИО – хранилище итоговые отчеты ЦИО.

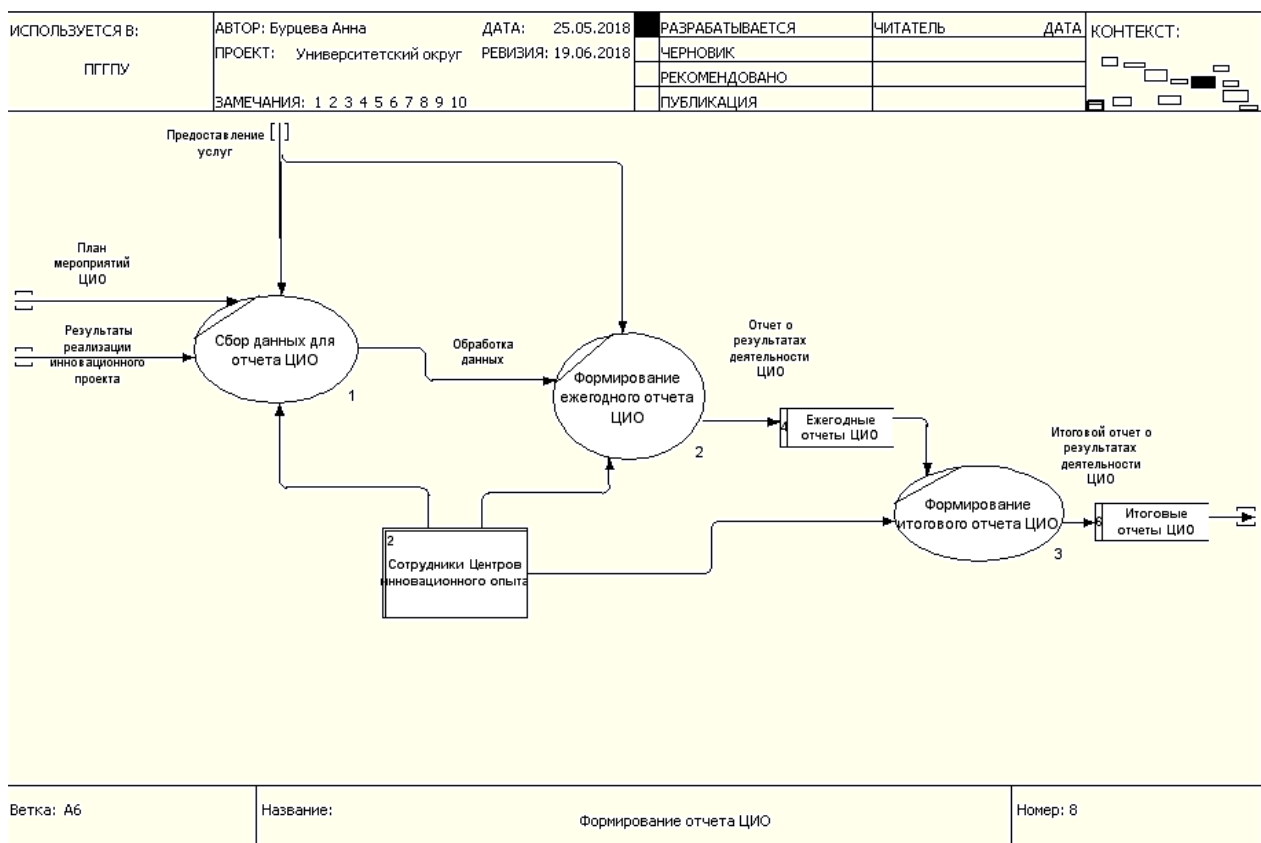


Рис.25 Диаграмма потоков для бизнес-процесса «Формирование отчета ЦИО»

Диаграмма потоков для бизнес-процесса «Формирование отчета Университетского округа»

На рисунке 26 представлена диаграмма потоков для бизнес-процесса «Формирование отчета Университетского округа», которая показывает, что бизнес-процесс «Формирование отчета Университетского округа» состоит из таких подпроцессов, как сбор данных для отчета Университетского округа, формирование ежегодного отчета Университетского округа, формирование итогового отчета Университетского округа. Для хранения отчетов Университетского округа имеется хранилище отчеты УО, итоговых отчетов ЦИО – хранилище итоговые отчеты ЦИО, статистических отчетов по мероприятиям – хранилище статистические отчеты.

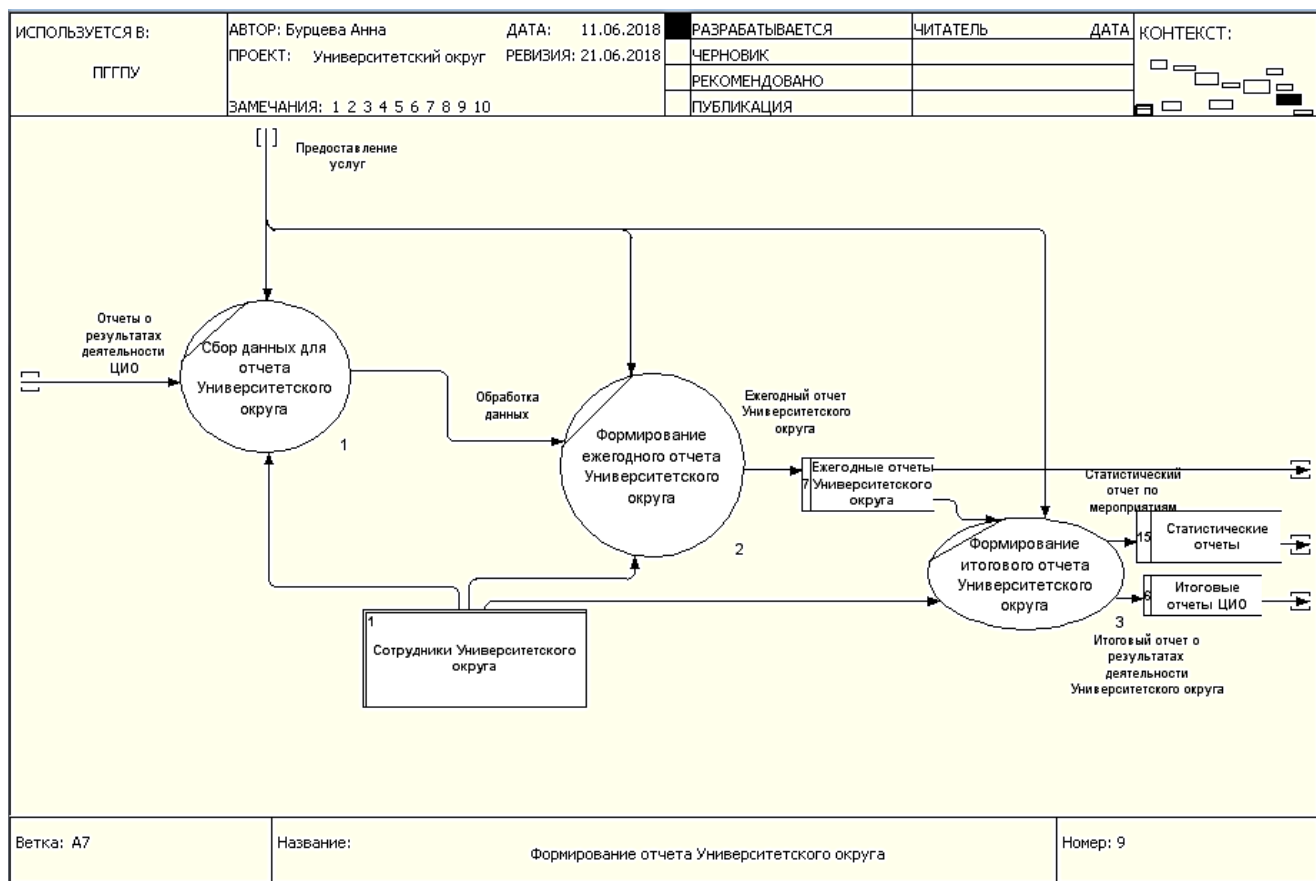


Рис.26 Диаграмма потоков для бизнес-процесса «Формирование отчета Университетского округа»

Диаграммы деятельности для потока событий

Диаграммы деятельности для потока событий «Учет ЦИО» (рис.27), «Работа с проектом» (рис.28), «Работа с мероприятием» (рис.29), «Работа с отчетом» (рис.30), «Работа с планом мероприятий» (рис.31) показывают как реализованы данные потоки событий в системе.

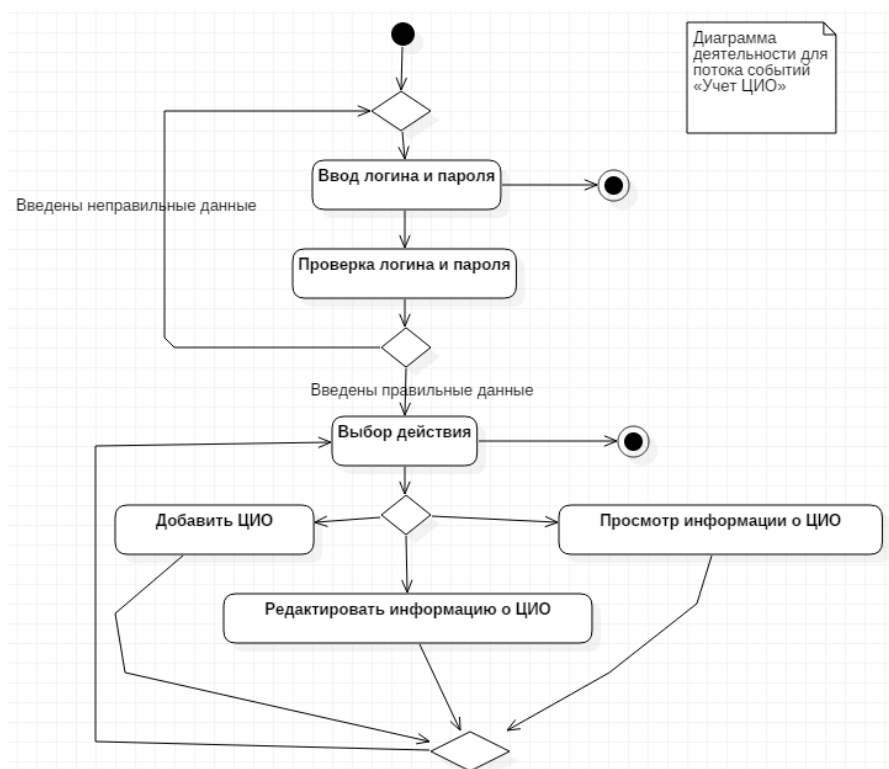


Рис.27 Диаграммы деятельности для потока событий «Учет ЦИО»

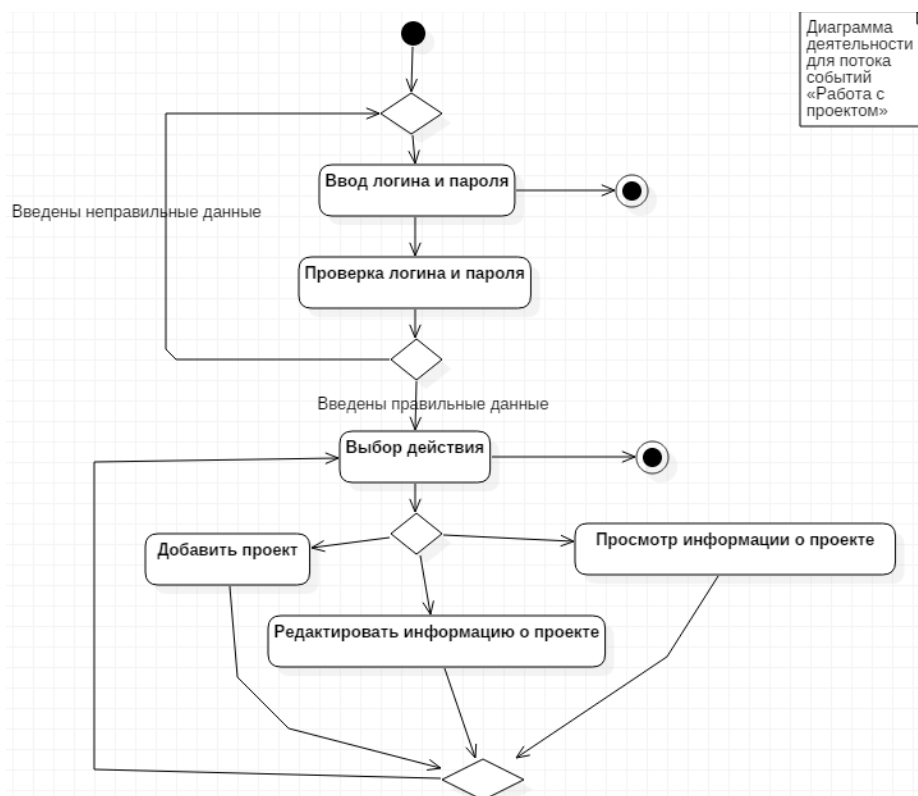


Рис.28 Диаграммы деятельности для потока событий «Работа с проектом»

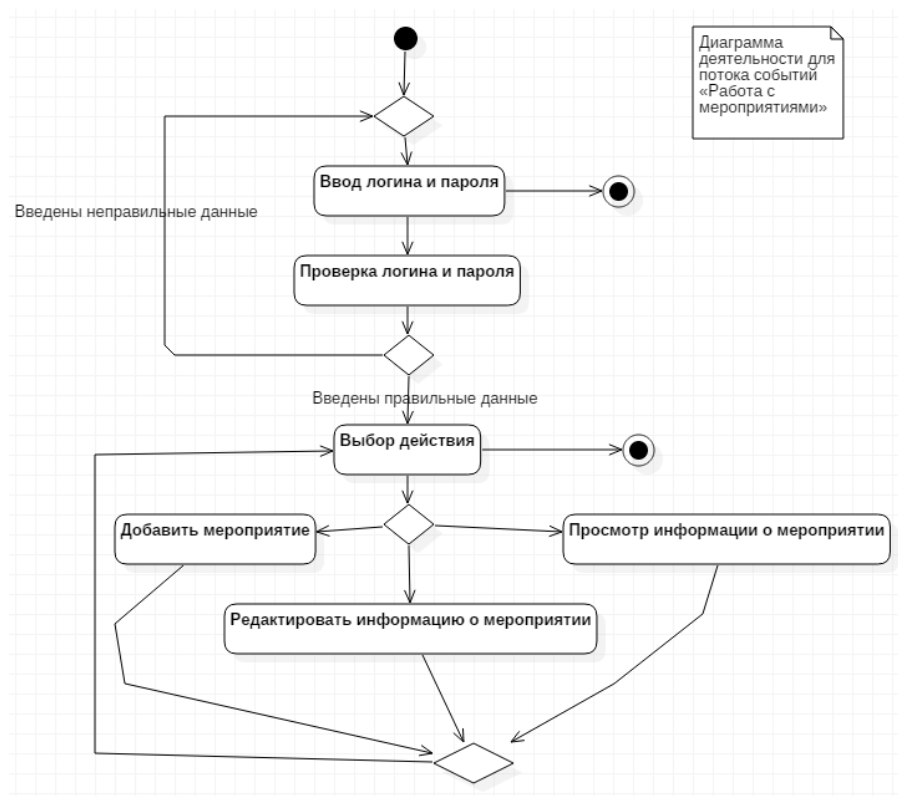


Рис.29 Диаграммы деятельности для потока событий «Работа с мероприятием»

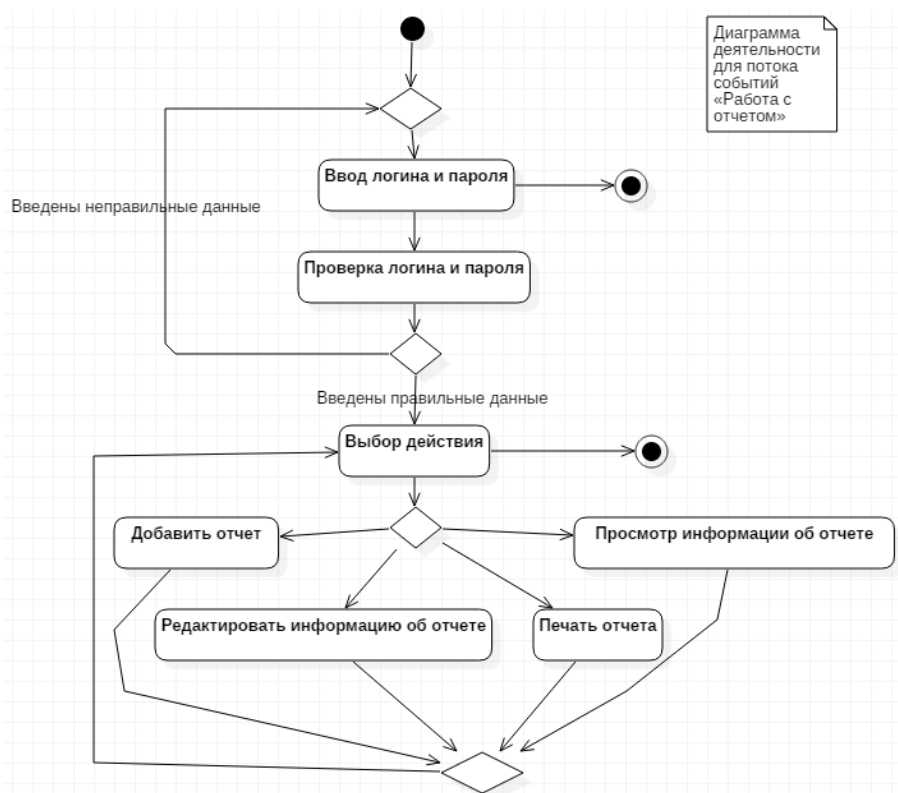


Рис.30 Диаграммы деятельности для потока событий «Работа с отчетом»

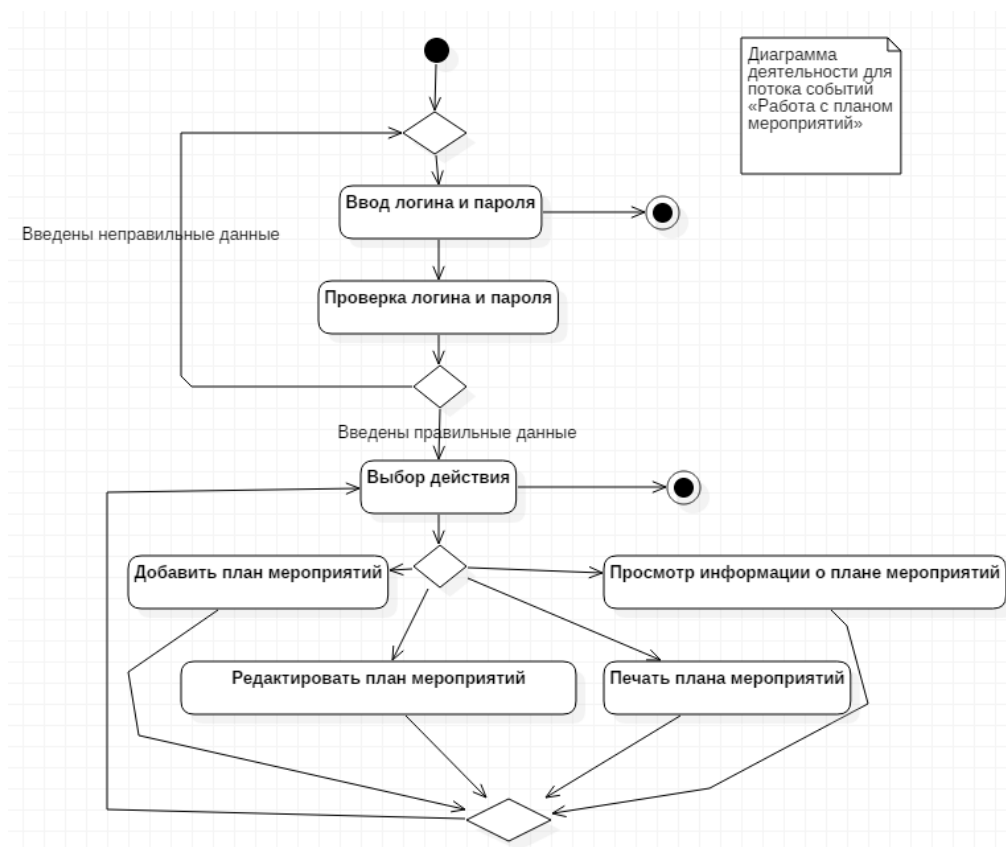


Рис.31 Диаграммы деятельности для потока событий «Работа с планом мероприятий»

Из анализа диаграмм потоков следует, что для правильного функционирования приложения необходимо разработать следующие отчеты:

- 1) Отчет по результатам деятельности Центра инновационного опыта
- 2) Итоговый отчет по результатам деятельности Центра инновационного опыта
- 3) Отчет по результатам деятельности Университетского округа
- 4) Итоговый отчет по результатам деятельности Университетского округа
- 5) План мероприятий ЦИО

- 6) План мероприятий Университетского округа
- 7) Статистический отчет по мероприятиям
- 8) Отчет по выборке мероприятий за определенный период

2.3.1.7. Разработка запросов

Для нормального функционирования приложения необходимо реализовать запросы, которые обеспечат быстрый доступ к данным, хранящимся в базе.

На основании перечисленных функций (в описании предметной области), которые должна выполнять ИС для каждой категории пользователей, необходимо разработать следующие запросы:

- 1) Вывести список Центров инновационного опыта
- 2) Вывести список всех отчетов ЦИО
- 3) Вывести список всех проектов ЦИО
- 4) Вывести список всех отчетов Университетского округа
- 5) Вывести список планов мероприятий Университетского округа
- 6) Вывести статистический отчет по проведению мероприятий ЦИО
- 7) Вывести список мероприятий за определенный период

2.3.2. Обеспечение целостности базы данных

Целостность данных — это соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. Она делится на три составляющие: категорную, ссылочную и доменную.

2.3.2.1. Категорная целостность

Категорная целостность ограничивает набор значений первичных ключей базовых отношений. Значения первичных ключевых атрибутов не могут быть отсутствующими, т.к. кортеж не может записываться в БД пока значение его первичного ключа не будет полностью определено, и должны быть индексированными (табл.12) .

Таблица №12. Категорная целостность.

Наименование таблицы ПК	Тип данных	Домен	Ограничения
УНИВЕРСИТЕТСКИЙ ОКРУГ Код_УО	счетчик	целое, >0	Обязательное –да Индексированное- да
ЦИО Код_ЦИО	счетчик	целое, >0	Обязательное –да Индексированное - да
ПРОЕКТ Код_проекта	счетчик	целое, >0	Обязательное –да Индексированное - да
МЕРОПРИЯТИЕ Код_мероприятия	счетчик	целое, >0	Обязательное –да Индексированное- да
РЕЗУЛЬТАТ_ПРОЕКТА Код_результата_проекта	счетчик	целое, >0	Обязательное –да Индексированное - да

2.3.2.2. Ссылочная целостность

Ссылочная целостность ограничивает набор значений внешних ключей базовых отношений. Ссылочная целостность гарантирует согласованность значений ключевых атрибутов в связанных таблицах. Этот вид целостности проверяет наличие ссылок на несуществующие значения (ссылки на отсутствующие значения запрещены в любой СУБД), а также обеспечивает согласованное изменение ссылок во всей ИС при изменении значения ключа (табл.13).

Таблица №13. Ссылочная целостность.

Наименование таблицы ВК	Тип данных	Домен	Ограничения
ЦИО Код_УО	числовой	целое,>0	Обязательное - да
ПРОЕКТ Код_ЦИО	числовой	целое,>0	Обязательное - да
МЕРОПРИЯТИЕ Код_проекта	числовой	целое,>0	Обязательное - да
РЕЗУЛЬТАТ_ПРОЕКТА Код_проекта	числовой	целое,>0	Обязательное - да

2.3.2.3. Доменная целостность

Доменная целостность – достоверность значений неключевых атрибутов. Домен в реляционной модели данных - это допустимое множество значений атрибута. В базах данных такая целостность обычно

обеспечивается условиями на значение, запретом пустых значений, масками ввода, триггерами и хранимыми процедурами (табл.14).

Таблица №14. Доменная целостность.

Наименование таблицы Атрибуты	Тип атрибута	Домен	Обеспечение целостности
УНИВЕРСИТЕТСКИЙ ОКРУГ			
Название_УО	текстовый	<30	Обязательное – да, пустых строк нет
Фамилия_рук	текстовый	<30	Обязательное – да, пустых строк нет
Имя_рук	текстовый	<30	Обязательное – да, пустых строк нет
Отчество_рук	текстовый	<30	Обязательное – да, пустых строк нет
Адрес_УО	текстовый	<30	Обязательное – да, пустых строк нет
Телефон_УО	числовой	>2000000 or<3000000	Обязательное – да

E-mail	текстовый	<30	Обязательное – да, пустых строк нет
Сайт_УО	текстовый	<30	Обязательное – да, пустых строк нет
ЦИО			
Название_обр_организации	текстовый	<30	Обязательное – да, пустых строк нет
Фамилия_дир	текстовый	<30	Обязательное – да, пустых строк нет
Имя_дир	текстовый	<30	Обязательное – да, пустых строк нет
Отчество_дир	текстовый	<30	Обязательное – да, пустых строк нет
Фамилия_рук_ЦИО	текстовый	<30	Обязательное – да, пустых строк нет
Имя_рук_ЦИО	текстовый	<30	Обязательное – да, пустых строк нет

Отчество_рук_ЦИО	текстовый	<30	Обязательное – да, пустых строк нет
Адрес_ЦИО	текстовый	<30	Обязательное – да, пустых строк нет
Телефон_ЦИО	числовой	>2000000 or<3000000	Обязательное – да
Сайт	текстовый	<30	Обязательное – да, пустых строк нет
ПРОЕКТ			
Название_проекта	текстовый	<50	Обязательное – да, пустых строк нет
Фамилия_автора_проекта	текстовый	<30	Обязательное – да, пустых строк нет
Имя_автора_проекта	текстовый	<30	Обязательное – да, пустых строк нет
Отчество_автора_проекта	текстовый	<30	Обязательное – да, пустых строк нет

Фамилия_рук_проекта	текстовый	<30	Обязательное – да, пустых строк нет
Имя_рук_проекта	текстовый	<30	Обязательное – да, пустых строк нет
Отчество_рук_проекта	текстовый	<30	Обязательное – да, пустых строк нет
Ссылка_на_раздел_на_сайте	текстовый	<50	Обязательное – да, пустых строк нет
Актуальность	текстовый	-	Обязательное – да, пустых строк нет
Цель	текстовый	-	Обязательное – да, пустых строк нет
Задачи	текстовый	-	Обязательное – да, пустых строк нет
Краткое_обоснование_инновационности_проекта	текстовый	-	Обязательное – да, пустых строк нет
Ожидаемые_продукты_проекта	текстовый	-	Обязательное – да,

			пустых строк нет
На_разработку_какого_иного_ин новационного_контента_направле н_проект	текстовый	-	Обязатель ное – да, пустых строк нет
Способы_апробации_и_диссемина ции_результатов	текстовый	-	Обязатель ное – да, пустых строк нет
Риски_и_пути_преодоления	текстовый	-	Обязатель ное – да, пустых строк нет
Механизм_самооценки	текстовый	-	Обязатель ное – да, пустых строк нет
МЕРОПРИЯТИЕ			
Название	текстовый	<50	Обязатель ное – да, пустых строк нет
Дата	дата/время	>=Date ()	Обязатель ное – да
Место_проведения	текстовый	<50	Обязатель ное – да, пустых строк нет
Категория_участников	текстовый	<50	Обязатель ное – да, пустых строк нет

Ответственный	текстовый	<30	Обязательное – да, пустых строк нет
РЕЗУЛЬТАТ_ПРОЕКТА			
Нормативно_правовое_обеспечение	текстовый	-	Обязательное – да, пустых строк нет
Кадровое_обеспечение_проекта	текстовый	-	Обязательное – да, пустых строк нет
Информационное_обеспечение_проекта	текстовый	-	Обязательное – да, пустых строк нет
Опыт_успешно_реализованных_проектов	текстовый	-	Обязательное – да, пустых строк нет
Описание_соответствия_содержания_изначально_запланированного_проекта_и_полученных_результатов	текстовый	-	Обязательное – да, пустых строк нет
Описание_текущей_актуальности_продуктов	текстовый	-	Обязательное – да, пустых строк нет
Продукты	текстовый	-	Обязательное – да, пустых строк нет

Описание_методов_и_критериев_мониторинга_качества_проекта	текстовый	-	Обязательное – да, пустых строк нет
Результаты_самооценки_мониторинга	текстовый	-	Обязательное – да, пустых строк нет
Перечень_достигнутых_результатов	текстовый	-	Обязательное – да, пустых строк нет

2.3.3. Реляционная схема данных

Реляционная схема была реализована в СУБД MySQL(рис.32).

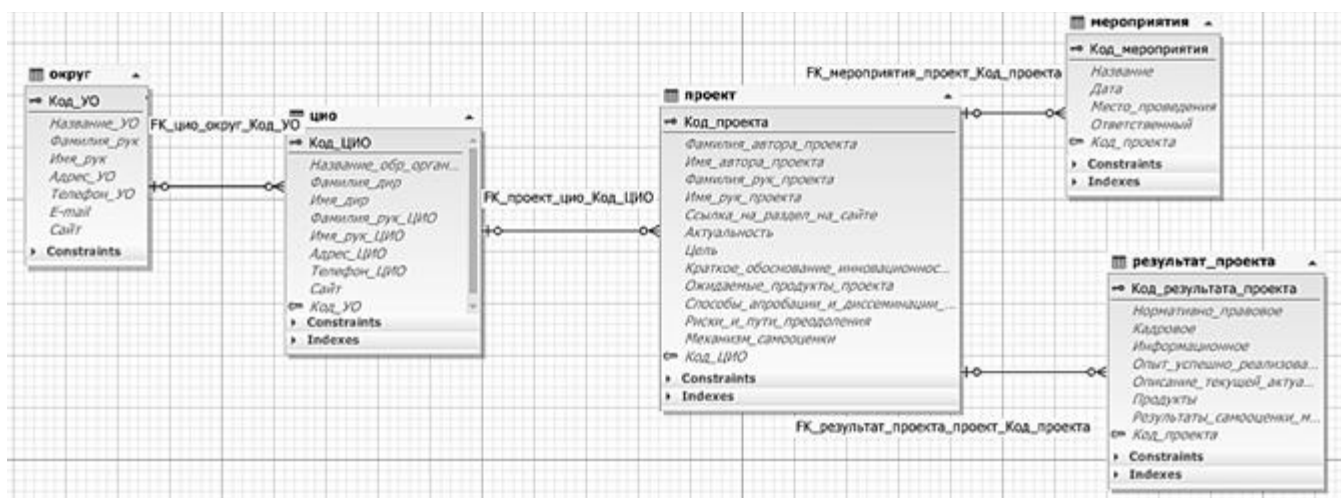


Рис.32 Реляционная схема данных ИС «Университетский округ»

2.4. Проектирование пользовательского интерфейса

Проектирование пользовательского интерфейса является третьим этапом создания информационной системы. Оно включает в себя:

- 1) эскиз пользовательского интерфейса;
- 2) описание пунктов пользовательского интерфейса.

2.4.5. Диаграммы прецедентов

Для проектирования пользовательского интерфейса на основании функционала каждой категории пользователей были разработаны диаграммы прецедентов (рис. 33,34).

На рисунке 33 представлена вспомогательная диаграмма прецедентов, на которой показаны 3 актера: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа, которые, для того, чтобы взаимодействовать с системой с помощью, так называемых вариантов использования: сбор и анализ данных для отчетов, составление планов мероприятий, составление отчетов и планов мероприятий ЦИО, просмотр отчетов и планов других ЦИО, составление отчетов Университетского округа, составление планов мероприятий Университетского округа, просмотр отчетов и планов мероприятий ЦИО, должны сначала пройти аутентификацию пользователя.

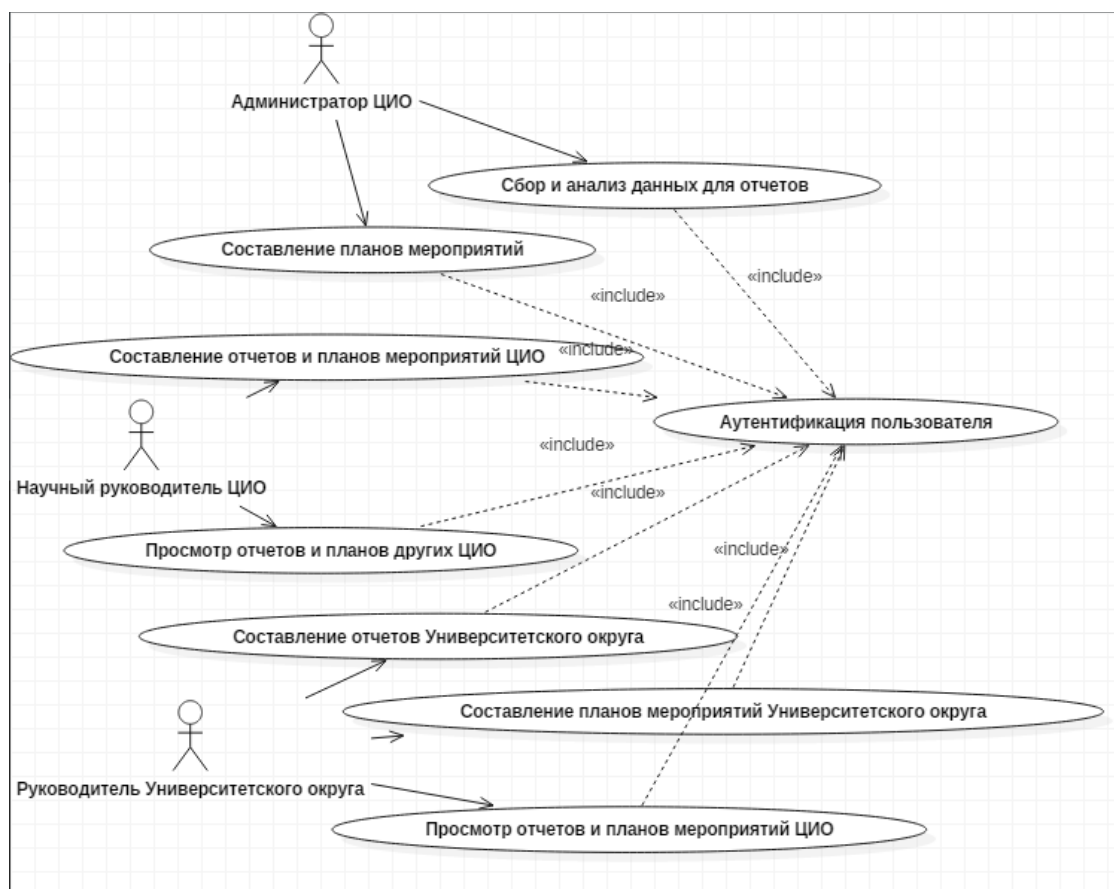


Рис.33 Диаграмма прецедентов (вспомогательная)

На рисунке 34 представлена главная диаграмма прецедентов, на которой показаны 3 актера: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа, взаимодействующие с системой с помощью, так называемых вариантов использования: работа с документацией ЦИО, просмотр документации других ЦИО, просмотр информации ЦИО, работа с документацией Университетского округа, статистика.

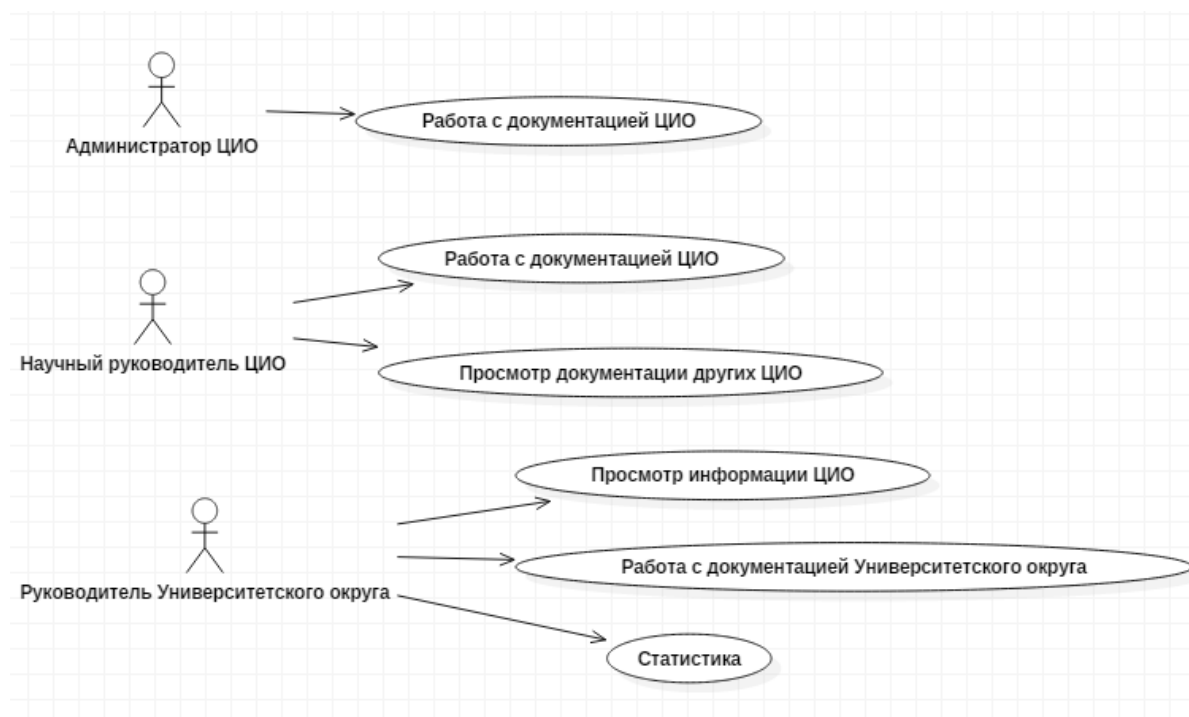


Рис.34 Диаграмма прецедентов (главная)

2.4.1. Диаграммы деятельности

Диаграммы деятельности описывают шаги, которые должна выполнить система после того, как выбрано данное действие.

Последовательности шагов при выборе одного из действий «Ввод информации о ЦИО», «Добавить проект», «Добавить мероприятие», «Добавить план мероприятий», представлены на рисунках 35-38.

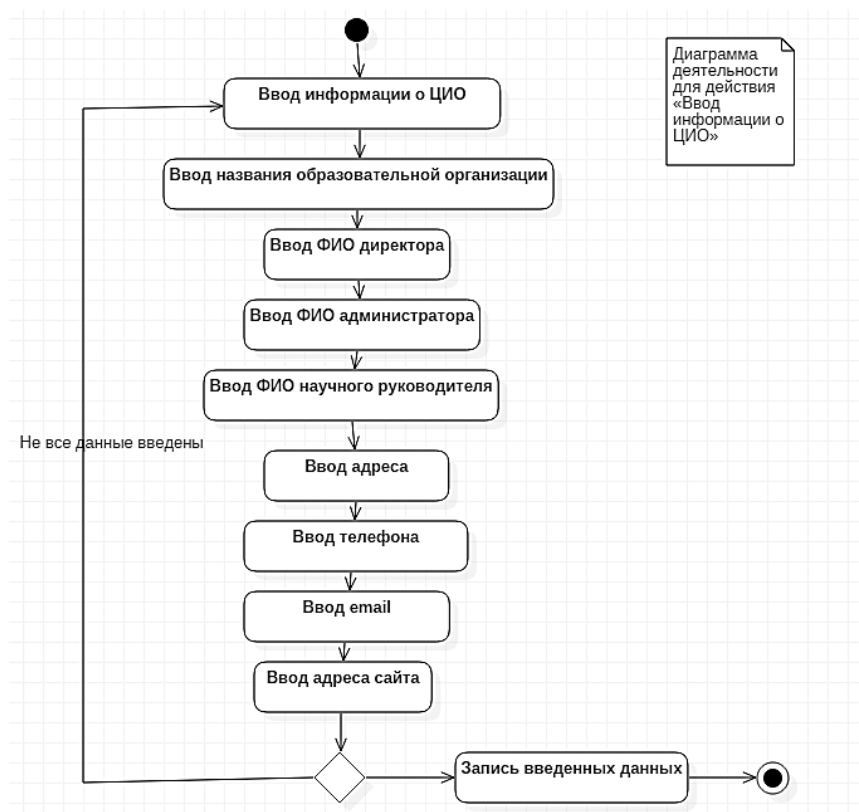


Рис.35 Диаграмма деятельности для действия «Ввод информации о ЦИО»

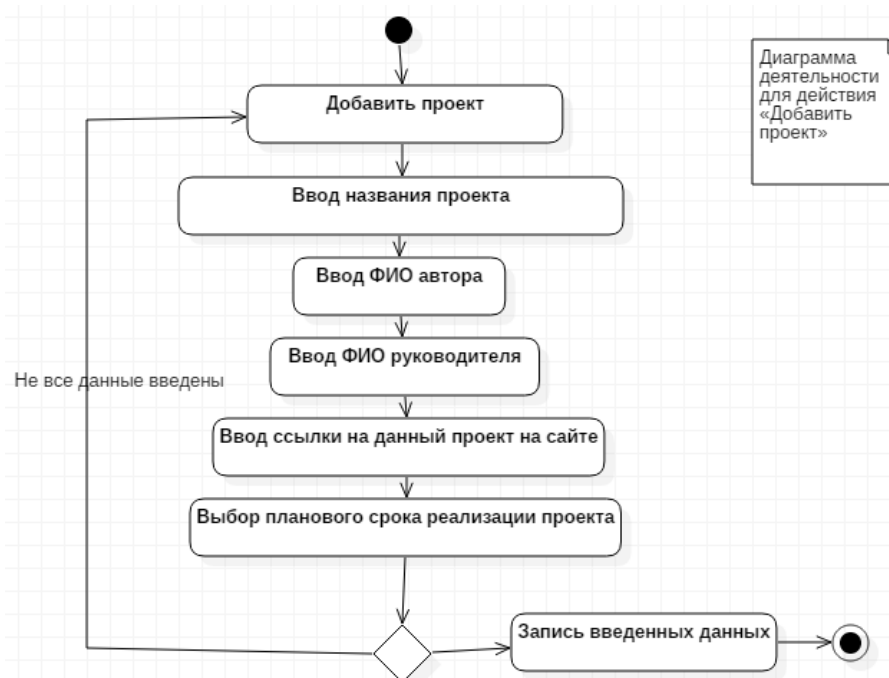


Рис.36 Диаграмма деятельности для действия «Добавить проект»

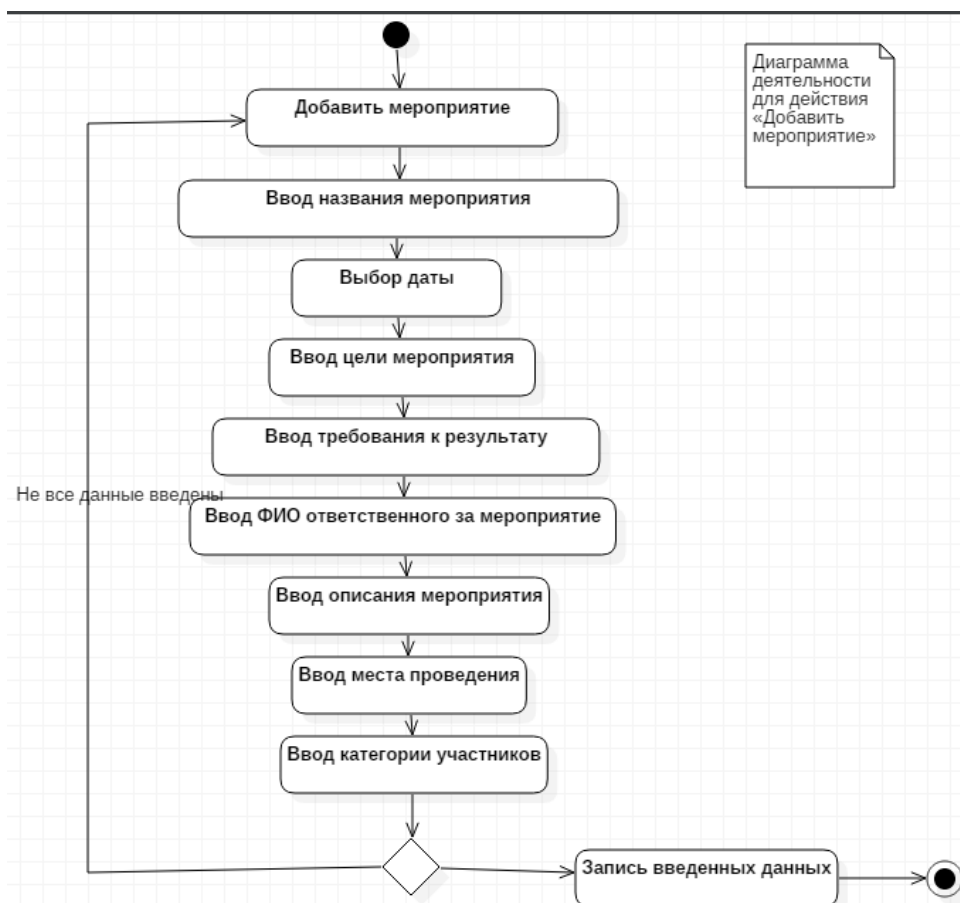


Рис.37 Диаграмма деятельности для действия «Добавить мероприятие»

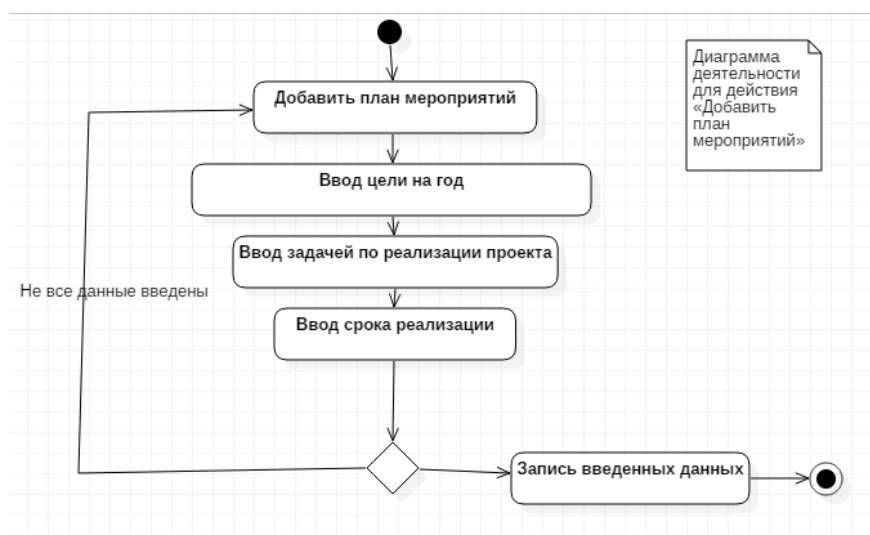


Рис.38 Диаграмма деятельности для действия «Добавить план мероприятий»

2.4.2. Эскиз пользовательского интерфейса

К проектируемой базе данных будут иметь доступ 3 категории пользователей: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа.

На рисунке 39 представлен эскиз главного меню пользовательского интерфейса проектируемой информационной системы «Университетский округ». На рисунке 40 представлен эскиз меню «Администратор ЦИО», на рисунке 41 – «Научный руководитель ЦИО», на рисунке 42 – «Руководитель Университетского округа».

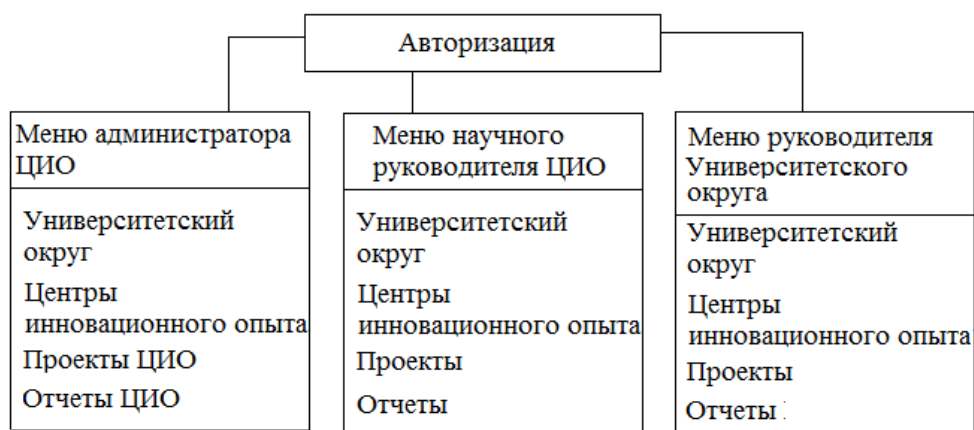


Рис.39 Эскиз главного меню



Рис.40 Эскиз меню «Администратор ЦИО»



Рис.41 Эскиз меню «Научный руководитель ЦИО»



Рис.42 Эскиз меню «Руководитель Университетского округа»

2.4.3. Описание пунктов пользовательского интерфейса

Рассмотрим подробное описание пунктов пользовательского меню проектируемой информационной системы. Все пункты описаны в таблицах 15 – 28 .

Описание главного меню

В таблице 15 описаны пункты меню «Главное меню».

Таблица 15. Меню «Главное меню»

Команда	Назначение
Авторизация	Открываются поля для ввода логина и пароля
Выход	Выход из приложения

Описание меню «Администратор ЦИО»

В таблицах 16 – 18 описаны команды меню «Администратор ЦИО» и соответствующих подменю.

Таблица 16. Главное меню «Администратор ЦИО»

Команда	Назначение
Университетский округ	Просмотр данных об Университетском округе
Центры инновационного опыта	Просмотр и редактирование профиля ЦИО, добавление отчетов и планов мероприятий ЦИО

Таблица 17. Подменю «Университетский округ»

Команда	Назначение
Информация об Университетском округе	Просмотр сведений об Университетском округе

Таблица 18. Подменю «Центры инновационного опыта»

Команда	Назначение
Профиль ЦИО	Просмотр, редактирование информации о ЦИО
Проекты	Добавление, просмотр, редактирование проектов ЦИО
Мероприятия	Добавление, просмотр, редактирование мероприятий ЦИО
Отчеты	Добавление, просмотр, редактирование отчетов ЦИО

Описание меню «Научный руководитель ЦИО»

В таблицах 19 – 23 описаны команды меню «Научный руководитель ЦИО» и соответствующих подменю.

Таблица 19. Главное меню «Научный руководитель ЦИО»

Команда	Назначение
Университетский округ	Просмотр данных об Университетском округе, просмотр отчетов и планов мероприятий Университетского округа, статистических отчетов
Центры инновационного опыта	Просмотр, поиск, редактирование данных о ЦИО, добавление отчетов и планов мероприятий ЦИО
Проекты	Просмотр данных обо всех проектах ЦИО
Отчеты	Просмотр данных обо всех отчетах ЦИО

Таблица 20. Подменю «Университетский округ»

Команда	Назначение
Информация об Университетском округе	Просмотр сведений об Университетском округе
План деятельности	Просмотр планов мероприятий Университетского округа
Отчеты	Просмотр отчетов, статистического отчета Университетского округа

Таблица 21. Подменю «Центры инновационного опыта»

Команда	Назначение
Информация о ЦИО	Просмотр, редактирование информации о ЦИО
Проекты	Добавление, просмотр, редактирование проектов ЦИО
Мероприятия	Добавление, просмотр, редактирование мероприятий ЦИО
Отчеты	Добавление, просмотр, редактирование отчетов ЦИО

Таблица 22. Подменю «Проекты»

Команда	Назначение
Список проектов	Просмотр, редактирование проектов ЦИО

Таблица 23. Подменю «Отчеты»

Команда	Назначение
Список отчетов	Просмотр, редактирование отчетов ЦИО

Описание меню «Руководитель Университетского округа»

В таблицах 24 – 28 описаны команды меню «Руководитель Университетского округа» и соответствующих подменю.

Таблица 24. Главное меню «Руководитель Университетского округа»

Команда	Назначение
Университетский округ	Просмотр данных об Университетском округе, добавление отчетов и планов мероприятий Университетского округа, формирование статистического отчета
Центры инновационного опыта	Просмотр, поиск, редактирование данных о ЦИО, добавление отчетов и планов мероприятий ЦИО
Проекты	Редактирование, просмотр данных обо всех проектах ЦИО
Отчеты	Редактирование, просмотр данных обо всех отчетах ЦИО

Таблица 25. Подменю «Университетский округ»

Команда	Назначение
Информация об Университетском округе	Редактирование, просмотр сведений об Университетском округе
План деятельности	Создание, редактирование, просмотр планов мероприятий Университетского округа

Отчеты	Создание, редактирование, просмотр отчетов, статистического отчета Университетского округа
--------	--

Таблица 26. Подменю «Центры инновационного опыта»

Команда	Назначение
ЦИО	Просмотр информации о ЦИО
Проекты	Добавление, просмотр, редактирование проектов ЦИО
Мероприятия	Добавление, просмотр, редактирование мероприятий ЦИО
Отчеты	Добавление, просмотр, редактирование отчетов ЦИО

Таблица 27. Подменю «Проекты»

Команда	Назначение
Список проектов	Просмотр, редактирование проектов ЦИО

Таблица 28. Подменю «Отчеты»

Команда	Назначение
Список отчетов	Просмотр, редактирование отчетов ЦИО

2.5. Разработка и реализация информационной системы «Университетский округ»

Спроектированная информационная система «Университетский округ» соответствует предъявленным требованиям заказчика и реализована с помощью фреймворка Django на языке программирования Python с использованием СУБД MySQL.

На рисунке 43 изображена начальная страница информационной системы «Университетский округ».

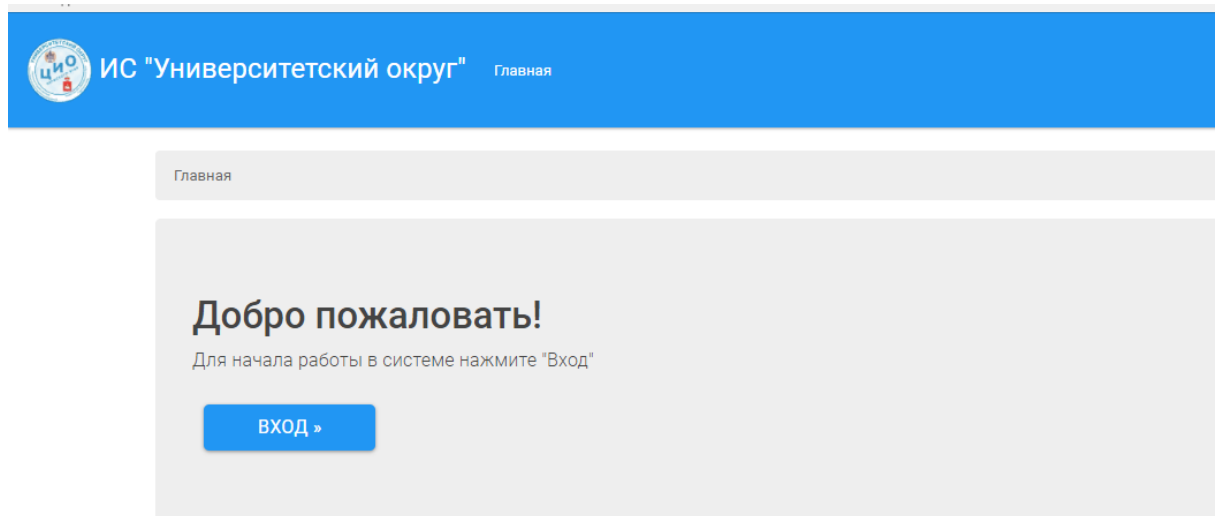


Рис.43 Начальная страница информационной системы
«Университетский округ»

Чтобы войти в информационную систему «Университетский округ» нужно авторизоваться, форма для авторизации представлена на рисунке 44.

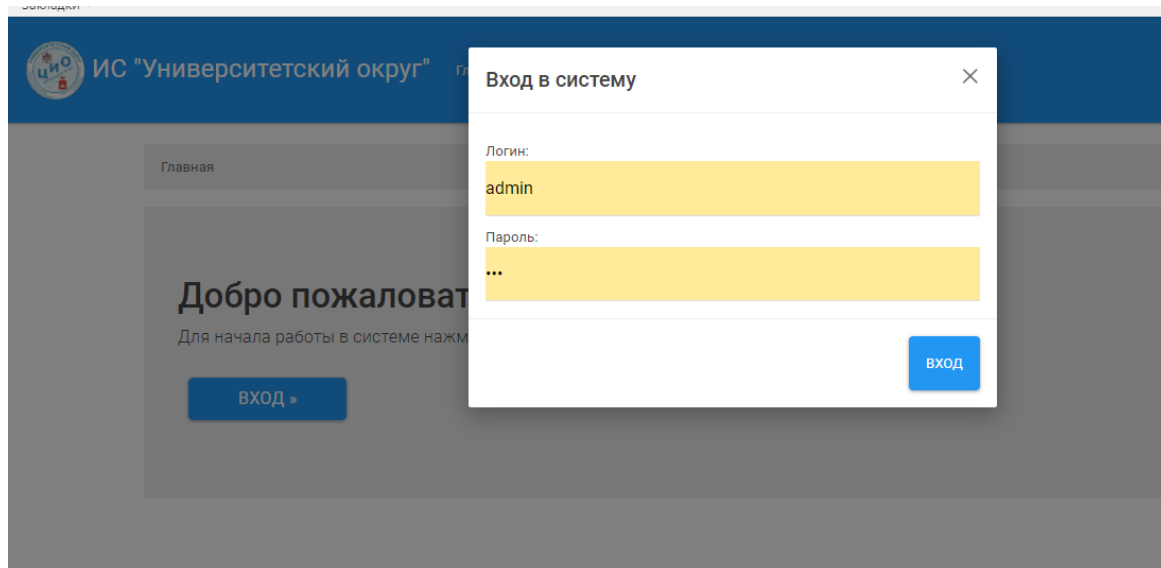


Рис.44 Авторизация пользователя

После успешной авторизации открывается меню соответствующего пользователя. Меню администратора ЦИО, научного руководителя ЦИО,

руководителя Университетского округа выглядят следующим образом (рис.45 - 47).

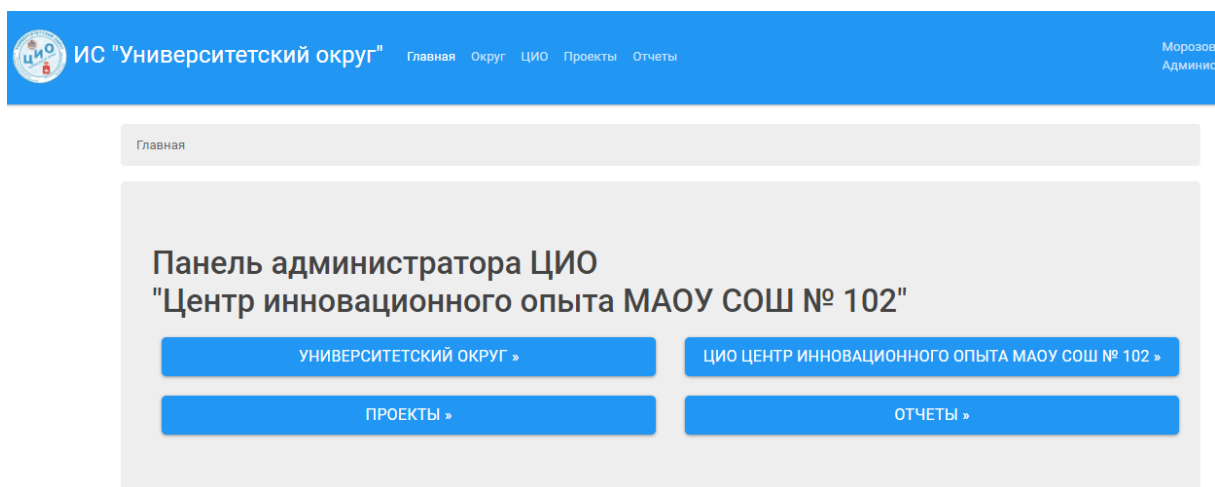


Рис. 45 Меню администратора ЦИО

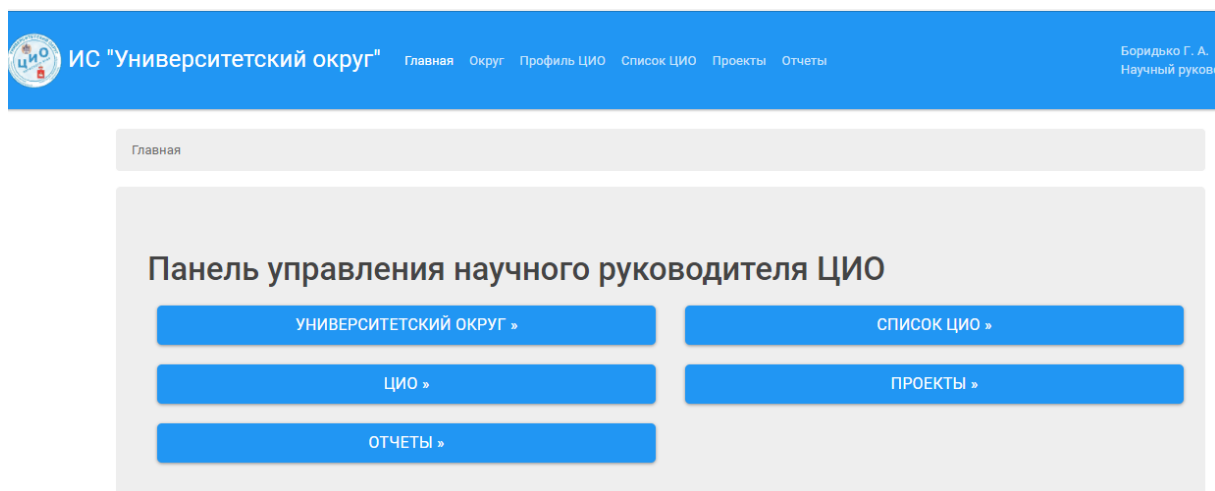


Рис. 46 Меню научного руководителя ЦИО

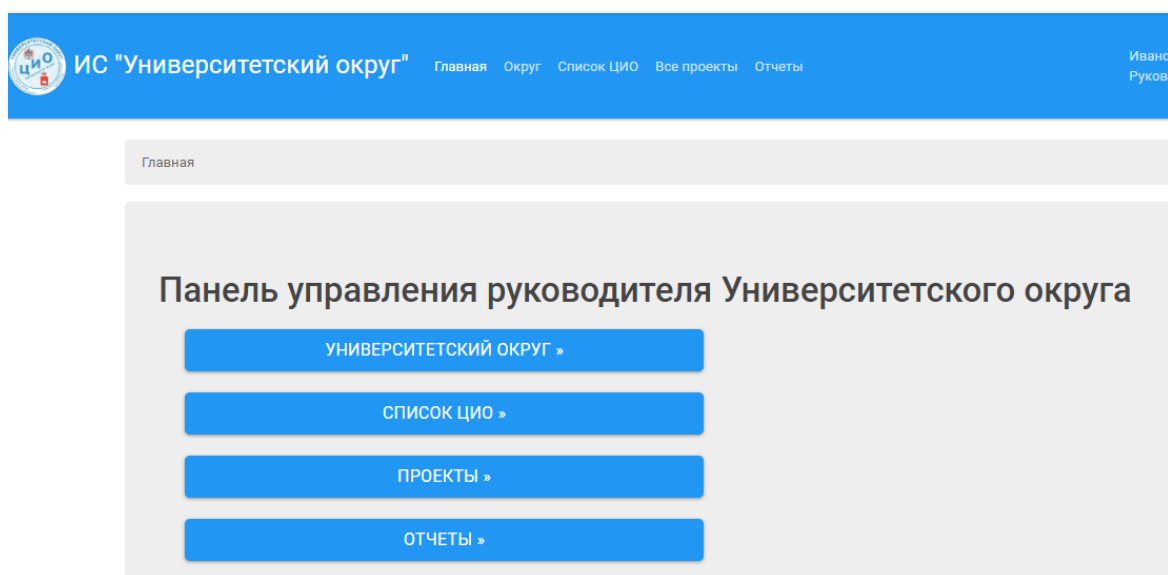


Рис. 47 Меню руководителя Университетского округа

Рассмотрим меню администратора ЦИО, основной его задачей является создание планов мероприятий и отчетов по результатам деятельности ЦИО, также он может редактировать профиль ЦИО, добавлять проекты и мероприятия (рис.48-57).

Редактировать ЦИО

Название образовательной организации:
Центр инновационного опыта МАOU СОШ № 102

Директор:
Ларионова Наталья Александровна

Администратор:
Морозов Михаил Олегович (admincio)

Научный руководитель:
Иванова Светлана Николаевна (иванова)

Адрес:
г. Пермь, ул. Мира 92

Телефон:
8 (342) 226-03-80

Email:

Наименование: Центр инновационного опыта МАOU СОШ № 102

Руководитель: Ларионова Наталья Александровна

Научный руководитель: Иванова Светлана Николаевна

Администратор: Морозов Михаил Олегович

Адрес: г. Пермь, ул. Мира 92

Телефон: 8 (342) 226-03-80

email: scool-102@rambler.ru

РЕДАКТИРОВАТЬ ОБЩИЕ СВЕДЕНИЯ

Рис.48 ЦИО

Редактировать проект

Название проекта:
Единая информационно-коммуникационная среда как средство

ФИО автора:
Романюк Галина Ивановна

ФИО руководителя:
Ростовщикова Юлия Александровна

Ссылка на раздел на сайте, посвященный проекту:
<http://schoolperm.ru/index.php/tsio>

Плановый срок реализации:
3 года

СОХРАНИТЬ

Показать 2 записей

№	Название
1	Управление развитием субъектной среды школы в контексте реализации
2	Единая информационно-коммуникационная среда как средство

Записи с 1 по 2 из 2 записей

Поиск:

Руководитель	Последняя	Следующая
Иванова Светлана Николаевна	1	
Ростовщикова Юлия Александровна		

+ ДОБАВИТЬ ПРОЕКТ

Рис.49 Редактирование проекта

Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития

Проект	Общие сведения	План мероприятий	Мероприятия	Отчеты
Наименование:	Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития			
Автор:	Романюк Галина Ивановна			
Руководитель:	Ростовщикова Юлия Александровна			
Ссылка на раздел на сайте, посвященный проекту:	http://schoolperm.ru/index.php/tsio			
Срок реализации:	3 года			

Рис.50 Проект ЦИО

Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития

Проект	Общие сведения	План мероприятий	Мероприятия	Отчеты
РЕДАКТИРОВАТЬ ОБЩИЕ СВЕДЕНИЯ				
Актуальность:	В условиях динамично меняющегося мира, постоянного совершенствования и усложнения технологий, информатизация сферы образования приобретает фундаментальное значение. Данное направление развития образовательной отрасли, как подчеркивается в государственных документах, признается важнейшим национальным приоритетом.			
Цель:	Повышение эффективности управления образовательным процессом ДОУ и обеспечение максимально благоприятных условий для раскрытия интеллектуально-творческого потенциала детей дошкольного возраста посредством использования возможностей современных информационных технологий.			
Задачи:	Создать единое информационное пространство ДОУ. Подготовить кадры к эффективному использованию в управленческом, воспитательно-образовательном процессе информационных и коммуникационных технологий. Предоставить участникам образовательного процесса свободный доступ к компьютерной технике, к информационным ресурсам, программным средствам. Интегрировать, внедрить информационные и коммуникационные технологии в процессы управления, воспитания и обучения. Создать банк дидактических и методических материалов, разработок по использованию информационных и коммуникационных технологий в работе ДОУ.			
Краткое обоснование	Информатизация образования – это комплексный, многоплановый, ресурсоемкий процесс, в котором участвуют дети, педагоги, и администрация ДОУ. Это создание единого информационного образовательного пространства ДОУ, города, района, страны; использование информационных технологий.			

Рис.51 Общие сведения

Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития

Проект	Общие сведения	План мероприятий	Мероприятия	Отчеты
+ ДОБАВИТЬ ПУНКТ ПЛАНА				
Показать ▼ записей		Поиск:		
№	Год	Цель на год	Сроки реализации	
1	2018	Разработка, освоение и апробация способов управления и диагностики развития СИПУ в информационной образовательной среде школы.	1 год	

Рис.52 Создание плана реализации проекта

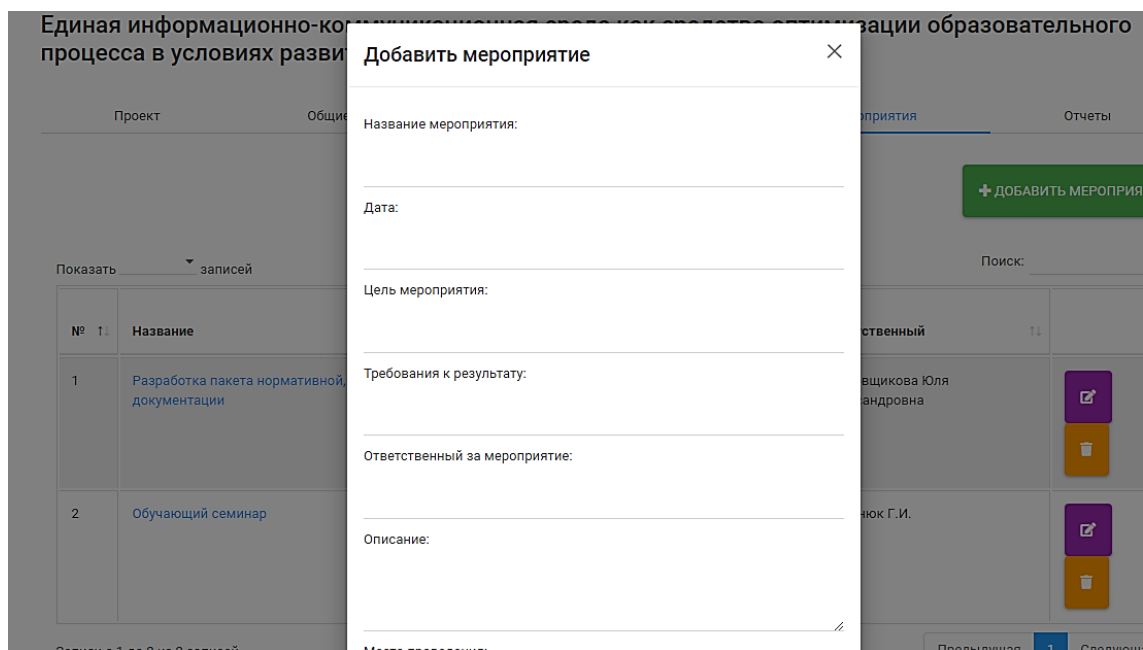


Рис.53 Добавление мероприятий

22 июня 2018 г. 8:14

План мероприятий по реализации инновационного проекта ЦИО на 2018 г Центр инновационного опыта МАОУ СОШ № 102

Тема проекта Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития

Цель проекта на 2018 год: Разработка, освоение и апробация способов управления и диагностики развития СИПУ в информационной образовательной среде школы.

Задачи по реализации проекта на 2018 год: Создать единое информационное пространство ДООУ. Подготовить кадры к эффективному использованию в управленческом, воспитательно-образовательном процессе информационных и коммуникационных технологий. Предоставить участникам образовательного процесса свободы

Срок реализации: 1 год

№	Мероприятие	Сроки	Место проведения	Категория участников	Ответственный
1	Разработка пакета нормативной, методической документации	25 апреля 2018 г.	г.Пермь, ул. Мира 92	Преподаватели, специалисты	Ростовщикова Юлия Александровна
2	Обучающий семинар	23 августа 2018 г.	г.Пермь, ул. Мира 92	Преподаватели	Романюк Г.И.

Рис.54 Печатная форма плана мероприятий ЦИО

Отчет по проекту Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития за 2018 год

ПЕЧАТЬ ОТЧЕТА

Общая информация	Ресурсное обеспечение	Реализация проекта	Результаты	Выводы
Информация в данном разделе формируется автоматически на основе данных проекта и ЦИО				
Общая информация об образовательной организации				
Наименование проекта:	Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития			
Образовательная организация, реализующая проект:	Центр инновационного опыта МАОУ СОШ № 102			
ФИО автора проекта:	Романюк Галина Ивановна			
ФИО руководителя проекта:	Ростовщикова Юлия Александровна			
Ссылка на раздел на сайте, посвященный проекту:	http://schoolperm.ru/index.php/tsio			

Общая информация о проекте

Рис.55 Отчет ЦИО

22 июня 2018 г. 8:18

Отчёт по результатам деятельности ЦИО Центр инновационного опыта МАОУ СОШ № 102 за 2018 год

1. Общая информация об образовательной организации

- 1.1 Наименование проекта: Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития
- 1.2 Образовательная организация, реализующая проект: Центр инновационного опыта МАОУ СОШ № 102
- 1.3 ФИО автора проекта: Романюк Галина Ивановна
- 1.4 ФИО руководителя проекта: Ростовщикова Юлия Александровна
- 1.5 Ссылка на раздел на сайте, посвященный проекту: <http://schoolperm.ru/index.php/tsio>

2. Общая информация о проекте

- 2.1 Актуальность: В условиях динамично меняющегося мира, постоянного совершенствования и усложнения технологий, информатизация сферы образования приобретает фундаментальное значение. Данное направление развития образовательной отрасли, как подчеркивается в государственных документах, признается важнейшим национальным приоритетом.
- 2.2 Цель: Повышение эффективности управления образовательным процессом ДООУ и обеспечение максимально благоприятных условий для раскрытия интеллектуально-творческого потенциала детей дошкольного возраста посредством использования возможностей современных информационных технологий.
- 2.3 Задачи: Создать единое информационное пространство ДООУ. Подготовить кадры к эффективному использованию в управленческом, воспитательно-образовательном процессе информационных и коммуникационных технологий. Предоставить участникам образовательного процесса свободный доступ к компьютерной технике, к информационным ресурсам, программным средствам. Интегрировать, внедрить информационные и коммуникационные технологии в процессы управления, воспитания и обучения. Создать банк дидактических и методических материалов, разработок по использованию информационных и коммуникационных технологий в работе ДООУ.
- 2.4 Краткое обоснование инновационности проекта: Информатизация образования – это комплексный, многоплановый, ресурсоемкий процесс, в котором участвуют дети, педагоги, и администрация ДООУ. Это

Рис.56 Печатная форма ежегодного отчета ЦИО

**Итоговый отчёт по результатам деятельности ЦИО Центр
инновационного опыта МАОУ СОШ № 102 по реализации проекта
Единая информационно-коммуникационная среда как средство
оптимизации образовательного процесса в условиях развития**

1. Общая информация об образовательной организации

- 1.1 Наименование проекта: Единая информационно-коммуникационная среда как средство оптимизации образовательного процесса в условиях развития
- 1.2 Образовательная организация, реализующая проект: Центр инновационного опыта МАОУ СОШ № 102
- 1.3 ФИО автора проекта: Романюк Галина Ивановна
- 1.4 ФИО руководителя проекта: Ростовщикова Юлия Александровна
- 1.5 Ссылка на раздел на сайте, посвященный проекту: <http://schoolperm.ru/index.php/tsio>

2. Общая информация о проекте

- 2.1 Актуальность: В условиях динамично меняющегося мира, постоянного совершенствования и усложнения технологий, информатизация сферы образования приобретает фундаментальное значение. Данное направление развития образовательной отрасли, как подчеркивается в государственных документах, признается важнейшим национальным приоритетом.
- 2.2 Цель: Повышение эффективности управления образовательным процессом ДОУ и обеспечение максимально благоприятных условий для раскрытия интеллектуально-творческого потенциала детей дошкольного возраста посредством использования возможностей современных информационных технологий.
- 2.3 Задачи: Создать единое информационное пространство ДОУ. Подготовить кадры к эффективному использованию в управленческом, воспитательно-образовательном процессе информационных и коммуникационных технологий. Предоставить участникам образовательного процесса свободный доступ к

Рис.57 Печатная форма итогового отчета ЦИО

Научный руководитель ЦИО также может добавлять проекты мероприятия, создавать планы мероприятий и отчеты, также он может просматривать планы мероприятий и отчеты других ЦИО.

Руководитель Университетского округа может редактировать информацию об Университетском округе (рис.58), формировать отчеты по результатам деятельности Университетского округа и планы мероприятий Университетского округа. План мероприятий Университетского округа (рис.59-60) автоматически складывается из мероприятий, запланированных ЦИО. Отчет по результатам деятельности Университетского округа формируется из отчетов ЦИО, итоговый отчет Университетского округа формируется на основе ежегодных отчетов Университетского округа, статистический отчет (рис.61) формируется на основе проведенных и не проведенных мероприятий ЦИО, также можно сделать выборку мероприятий за определенный период (рис.62-63).

Университетский округ

Общие сведения	План мероприятий	Отчеты
РЕДАКТИРОВАТЬ СВЕДЕНИЯ		
Наименование:	Университетский округ ПГГПУ	
Руководитель:	Колесников Андрей Константинович	
Адрес:	г.Пермь, ул.Сибирская, 26	
Телефон:	8 (342) 212-68-52	
email:	okrug@pspu.ru	
Сайт:	http://pspu.ru/university/universitetskij-okrug/ob-okruge/uchastniki-universitetskogo-okruga	



Рис.58 Университетский округ

Университетский округ

Общие сведения	План мероприятий	Отчеты
+ ДОБАВИТЬ		
Показать записей		
Поиск:		
№	Год	
1	2018	📄 🔍 ✎ 🗑
2	2017	📄 🔍 ✎ 🗑

Показывать 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 253

План мероприятий Университетского округа на 2018 г.

№	ЦИО	Мероприятие	Сроки	Место проведения	Категория участников	Ответственный
1	Центр инновационного опыта МАОУ СОШ № 102	Разработка пакета нормативной, методической документации	25 апреля 2018 г.	г.Пермь, ул. Мира 92	Преподаватели, специалисты	Ростовщикова Юля Александровна
2	Центр инновационного опыта МАОУ "Гимназия № 7" г. Перми	конференция	30 июня 2018 г.	ПГТПУ	Преподаватели	Иванова М.Ю.
3	Центр инновационного опыта МАОУ "Гимназия № 7" г. Перми	Курсы	10 августа 2018 г.	г. Пермь, ул. Целинная, 29Б	Преподаватели	Морозов О.Ю.
4	Центр инновационного опыта МАОУ СОШ № 102	Обучающий семинар	23 августа 2018 г.	г.Пермь, ул. Мира 92	Преподаватели	Романюк Г.И.
5	Центр инновационного опыта МАОУ "Гимназия № 7" г. Перми	Форум	12 октября 2018 г.	ПГТПУ	Преподаватели	Макарова Д.А.

Рис.60 Печатная форма плана мероприятий Университетского округа

22 июня 2018 г. 11:22

Статистический отчет по реализации проектов ЦИО в
Университетском округе за 2018 год

ЦИО: Центр инновационного опыта МАОУ "Гимназия № 7" г. Перми

Проект: Достижение личностных и метапредметных результатов образования в рамках образовательной программы
"Школа социального творчества"

№	Название мероприятия	Статус
1	конференция	Мероприятие запланировано
2	Курсы	Мероприятие запланировано
3	Форум	Мероприятие проведено
4	конкурс	Мероприятие проведено

Запланировано мероприятий: 4, из них проведено 2

ЦИО: Центр инновационного опыта МАОУ "Средняя общеобразовательная школа № 135" г. Перми

Проект: клавпавп

№	Название мероприятия	Статус
1	курсы повышения квалификации	Мероприятие проведено

Запланировано мероприятий: 1, из них проведено 1

Рис.61 Печатная форма статистического отчета Университетского округа

Отчет по мероприятиям
×

Начало периода:

01.08.2017

Окончание периода:

01.12.2018

ПЕЧАТЬ ОТЧЕТА

Рис.62 Выбор периода для выборки мероприятий для отчета

22 июня 2018 г. 9:18

Отчет по мероприятиям ЦИО за период с 1 декабря 2017 г. по 1 декабря 2018 г.

№	ЦИО	Название	Дата	Место	Категория участников	Ответственный	Статус
1	Центр инновационного опыта MAOY СОШ № 102	Разработка пакета нормативной, методической документации	25 апреля 2018 г.	г.Пермь, ул. Мира 92	Преподаватели, специалисты	Ростовщикова Юлия Александровна	Мероприятие проведено
2	Центр инновационного опыта MAOY "Гимназия № 7" г. Перми	конференция	30 июня 2018 г.	ПГТПУ	Преподаватели	Иванова М.Ю.	Мероприятие запланировано
3	Центр инновационного опыта MAOY "Гимназия № 7" г. Перми	Курсы	10 августа 2018 г.	г. Пермь, ул. Целинная, 29Б	Преподаватели	Морозов О.Ю.	Мероприятие запланировано
4	Центр инновационного опыта MAOY СОШ № 102	Обучающий семинар	23 августа 2018 г.	г.Пермь, ул. Мира 92	Преподаватели	Романюк Г.И.	Мероприятие запланировано
	Центр						

Рис.63 Выборка мероприятий за определенный период

Выводы по главе 2

1. Информационная система «Университетский округ» позволяет автоматизировать процесс сбора информации по планированию и отчетам по результатам деятельности ЦИО и Университетского округа; имеет 3 категории пользователей: администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа.
2. Во время проектирования информационной системы применялись такие CASE – средства как Ramus для моделирования бизнес-процессов, StarUML для построения диаграммы прецедентов, диаграммы классов, диаграмм деятельности, последовательности действий, кооперации.
3. Серверная часть информационной системы разработана с помощью СУБД MySQL.
4. При разработке пользовательского интерфейса информационной системы использовался фреймворк Django на языке программирования Python.

ГЛАВА 3. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УНИВЕРСИТЕТСКИЙ ОКРУГ»

Основной задачей проведения экономического исследования по внедрению информационной системы «Университетский округ» является определение величины экономического эффекта от внедрения продукта.

Целью проводимых расчетов является определение сроков окупаемости внедряемой системы.

3.1. Расчет затрат на информационную разработку автоматизированной системы «Университетский округ»

Расчет себестоимости программы

Расчет себестоимости программного продукта – это, как правило, совокупность затрат на разработку программного продукта, которые подразделяются на следующие статьи расходов:

- Основная заработная плата;
- Накладные расходы.

Расчет заработной платы производится в соответствии с трудоемкостью программного продукта.

Плановый фонд рабочего времени за месяц – 22 дня по 8 часов (3 часть Федерального закона от 22.07.2008 N 157-ФЗ Трудового кодекса РФ). Итого 176 часов в месяц. На разработку программного продукта потрачено 2 месяца. Таким образом, общий объем времени, потраченный на разработку программы, составил $T_{ож} = 352 (176*2)$ часов.

Средний размер заработной платы разработчика на языке программирования python по данным сайта <https://perm.hh.ru> в г.Пермь составляет 60 000 рублей.

Часовая тарифная ставка определяется ($C_ч$) определяется по формуле (1):

$$C_{\text{ч}} = \frac{\text{Оклад}}{\Phi_{\text{рв}}}, \quad (1)$$

где $\Phi_{\text{рв}}$ – плановый фонд рабочего времени за месяц, из расчета 22 рабочих дня по 8 часов.

$$C_{\text{ч}} = \frac{60000}{176} = 340 \text{ руб. в час}$$

Основная заработная плата за разработку программы определяется по формуле (2):

$$ЗП_{\text{осн}} = C_{\text{ч}} \cdot T_{\text{ож}}, \quad (2)$$

где $T_{\text{ож}}$ - время написания программы, которое составляет 2 месяца.

$$ЗП_{\text{осн}} = 340 \cdot 352 = 119\,680 \text{ руб.}$$

Также необходимо включить затраты, связанные с обслуживанием и организацией производства. В данном случае – это расходы на электроэнергию, потребляемую компьютером (ноутбуком) за время разработки программы.

Ниже в таблице 29 выполнены расчеты по затрате ресурсов на электроэнергию по формуле (3):

$$\text{Сумма} = (M \cdot C) \cdot T, \quad (3)$$

где:

- M - мощность, кВт;
- C - стоимость 1 кВт/час, руб.;
- T - время работы оборудования, час.

Согласно постановлению РСТ Пермского края от 20.12.2017 №40-э цена (тариф) на электрическую энергию для населения и приравненным к

нему категориям потребителей Пермского края на 2018 год с 01.01.2018 по 30.06.2018 составляет 3,77 руб./кВтч (одноставочный тариф).

Мощность ноутбука Packard bell easynote TE рассчитывается следующим образом: входное напряжение — 240 вольт, максимальный потребляемый ток — 1.6 А. Действительная входная мощность составляет не более 80% от максимальной мощности. Из этого следует, что мощность ноутбука равна $240 \cdot 1.6 \cdot 0.8 = 288$ ватт, что соответствует энергопотреблению 0.288 кВт*час.

Таблица 29. Затраты на электроэнергию

Вид оборудования	Кол-во (шт.)	Мощность, кВт	Стоимость 1 кВт/час, руб.	Время работы оборудования, час	Сумма затрат, руб.
Ноутбук Packard bell easynote TE	1	0,288	3,77	352	382,2
Итого	-	-	-	-	382,2

Сумма накладных расходов = затраты на электроэнергию.

Сумма накладных расходов = 382,2 руб.

Эксплуатационные материалы

Программное обеспечение, которое использовалось при разработке, такое как Windows 7, MS Office 2010, Ramus, StarUML, платформа Django, MySQL приобретены не специально для реализации системы, поэтому их стоимость не включается в расчет затрат.

Расчет затрат наглядно представлен в таблице 30.

Таблица 30. Калькуляция затрат на разработку программного продукта

Статья затрат	Сумма затрат, руб.
Общая сумма з/п	119 680
Накладные расходы	382,2
Эксплуатационные материалы	0
Итого:	120 062,2

Таким образом, затраты на разработку программы составляют 120 062,2 рублей.

3.2. Определение возможной цены программного продукта

Рассчитаем плановый уровень прибыли с условием, что процент рентабельности составляет 3,6% на 2018 г. в розничной торговле по данным Федеральной Налоговой Службы. Прибыль рассчитывается по формуле (4):

$$\Pi = \frac{C_{\text{полн}} \cdot P}{100}, \quad (4)$$

где:

- $C_{\text{полн}}$ – себестоимость;
- P - процент рентабельности.

$$\Pi = \frac{120\,062,2 \cdot 3,6}{100} = 4\,322,24 \text{руб.}$$

Цена программного продукта равна сумме полной себестоимости и прибыли и рассчитывается по формуле (5):

$$\Pi = C_{\text{полн}} + \Pi \quad (5)$$

$$\Pi = 120\,062,2 + 4\,322,24 = 124\,384,4 \text{ руб.}$$

НДС и цена программного продукта с НДС определяется по формулам (6) и (7):

$$\text{НДС} = \frac{\Pi \cdot 18}{100} \quad (6)$$

$$\text{НДС} = \frac{124\,384,4 \cdot 18}{100} = 22\,389,2 \text{ руб.}$$

$$\text{Цена} = \text{НДС} + \Pi \quad (7)$$

$$\text{Цена} = 22\,389,2 + 124\,384,4 = 146\,773,6 \text{ руб.}$$

Таким образом, предполагаемая стоимость программного продукта составляет 146 773,6 рублей.

3.3. Расчет основных затрат

К основным затратам относятся приобретение информационной системы и обучение персонала. Совокупность основных затрат на внедрение информационной системы «Университетский округ» приведена в таблице 31.

Таблица 31. Затраты на внедрение ИС «Университетский округ»

№	Статья затрат	Стоимость за единицу	Количество	Сумма, руб.
1	Приобретение программы	146 773,6 руб./шт.	1 шт.	146 773,6
2	Обучение	200 руб./ч.	1 ч.	200

	администратора ЦИО			
3	Обучение научного руководителя ЦИО	200 руб./ч.	1 ч.	200
4	Обучение руководителя Университетского округа	200 руб./ч.	1 ч.	200
	Итого:	—	—	147 373,6

Основной статьей расходов на внедрение является приобретение программного продукта. Затраты составляют 146 773,6рублей.

Обслуживанием программного продукта занимаются администратор ЦИО, научный руководитель ЦИО, руководитель Университетского округа, затраты ограничиваются обучением сотрудников администрированию продукта. Стоимость обучения составляет 600 руб.

Стоимость внедрения ИС «Университетский округ» составляет 147 373,6 руб.

3.4. Эффективность внедрения системы

Оценка стоимости выигранного рабочего времени.

Методист-специалист Университетского округа, который выполняет функцию руководителя Университетского округа, тратит на анализ и сбор отчетности 1 месяц в год, в другое время он занимается организацией конференций и мероприятий для школьников.

Отсюда следует, что методист тратит 176 часов в год на обработку отчетности.

Стоимость одного рабочего часа у методиста равна 113 рублей, она рассчитывается следующим образом: общая заработная плата в месяц/количество часов в месяц.

Общая заработная плата специалиста-методиста Университетского округа ПГПУ составляет 20000 рублей.

Общая заработная плата - заработная плата сотрудников, которая является суммой основной заработной платы, дополнительной заработной платы, а также премиальных выплат.

Расчет общей ежемесячной заработной платы производим по формуле (8):

$$ЗП_{\text{Общ}} = ЗП_{\text{Осн}} + ЗП_{\text{Доп}} + ЗП_{\text{Прем}} + РК, \quad (8), \text{ где}$$

- $ЗП_{\text{Общ}}$ – общая заработная плата;
- $ЗП_{\text{Осн}}$ – основная заработная плата;
- $ЗП_{\text{Доп}}$ – дополнительная заработная плата;
- $ЗП_{\text{Прем}}$ – премиальные выплаты;
- РК – надбавка по районному коэффициенту.

Дополнительная заработная плата - это вознаграждение за труд сверх установленных норм, за трудовые успехи и изобретательность и за особые условия труда, которое включает:

- выплаты за непроработанное время, предусмотренные законодательством о труде и коллективными договорами;
- оплата времени отпусков;
- плату за выполнение государственных и общественных обязанностей;
- выходное пособие при увольнении и др.

Ежемесячная дополнительная заработная плата равна 1/12 от основной заработной платы, так как отпуск работнику оплачивается исходя из среднего значения основной заработной платы за год. Расчет дополнительной заработной платы производится по формуле (9).

$$ЗП_{\text{Доп}} = ЗП_{\text{Осн}} / 12 \quad (9)$$

Основная заработная плата методиста составляет 20 000 руб., исходя из формулы 9, получим результат: $ЗП_{\text{доп}} = 20\,000 / 12 = 1666$ руб.

Премияльные выплаты составляют 25% от основной заработной платы, из этого следует, что премиальные выплаты составляют $20\,000 * 0,25 = 5\,000$ руб.

Надбавка районного коэффициента содержит % районного коэффициента к основной заработной плате, в зависимости от территориального расположения его применения.

Районный коэффициент в Пермском края составляет 1,15%. Исходя из этого, надбавка по районному коэффициенту составляет: $20\,000 * 0,0115 = 230$ руб.

Подставляя данные в формулу 8, получаем сумму общей ежемесячной заработной платы:

$$ЗП_{\text{общ}} = 20000 + 1666 + 5000 + 230 = 26\,896 \text{ руб.}$$

Также в каждой образовательной организации, которая получила статус ЦИО по результатам конкурса инновационных проектов, возникает необходимость в сотруднике, который бы занимался формированием отчетности, планов мероприятий и сдачей документации методисту Университетского округа. Поэтому эти обязательства обычно передают заведующему или преподавателю образовательного учреждения, и они за год примерно тратят на работу с отчетностью около 30 часов.

Стоимость одного рабочего часа у преподавателя равна 215 рублей, она рассчитывается следующим образом: общая заработная плата в месяц/количество часов в месяц.

Согласно трудовому кодексу РФ норма выработки количества часов составляет 144, так как в соответствии со ст. 333 ТК РФ для педагогических работников устанавливается сокращенная продолжительность рабочего времени — не более 36 ч в неделю. [9]

Общая заработная плата учителя по Перми составляет 31070 рублей[10].

Основная заработная плата методиста составляет 31070руб., исходя из формулы 9, получим результат: $ЗП_{доп} = 31070 / 12 = 2\,589$ руб.

Премияльные выплаты составляют 25% от основной заработной платы, из этого следует, что премиальные выплаты составляют $31070 * 0,25 = 7\,767,5$ руб.

Надбавка районного коэффициента содержит % районного коэффициента к основной заработной плате, в зависимости от территориального расположения его применения.

Районный коэффициент в Пермском края составляет 1,15%. Исходя из этого, надбавка по районному коэффициенту составляет: $31070 * 0,0115 = 357,305$ руб.

Подставляя данные в формулу 8, получаем сумму общей ежемесячной заработной платы:

$$ЗП_{общ} = 31070 + 2\,589 + 7\,767,5 + 230 = 41\,656,5 \text{руб.}$$

Далее рассчитываем экономию в год.

Для этого мы умножаем стоимость часа на часы в год: для методиста Университетского округа получаем 19 888рублей($113 \text{руб.} * 176$), для преподавателя образовательной организации 6 450руб($215 \text{руб.} * 30$).

Так как информационной системой будут пользоваться: 1 методист и 30 администраторов ЦИО (сотрудники обр.учреждений), получаем сумму 116 638руб.

Годовая экономия составит 48 114,4рублей (195 488рублей вычесть стоимость внедрения 147 373,6 рублей).

Срок окупаемости

Рассчитаем срок окупаемости продукта по формуле (10):

$$T = K / П, \quad (10),$$

где

- К – единовременные капитальные затраты при внедрении;
- П – годовая экономия, руб.

В нашем случае $K = 147\,373,6$ руб., $P = 48\,114,4$ руб.

Подставляя данные значения в формулу (10), получаем: $T = 3,06$. Умножим на 365 дней в году и получим срок окупаемости продукта в 1 116,9 дней.

Таким образом, внедрение ИС «Университетский округ» примерно окупится через 3 года и 20 дней.

При использовании информационной системы «Университетский округ» появляется ряд преимуществ:

- Экономия времени на сбор необходимой информации о мероприятиях УО;
- Экономия времени на передачу отчетности;
- Обеспечение сетевого взаимодействия ЦИО.

На основании приведенных расчетов можно сделать вывод о том, что за счет экономии времени ИС может значительно повысить эффективность взаимодействия центров инновационного опыта.

Выводы по главе 3

1. Рассчитаны затраты на разработку и реализацию ИС «Университетский округ» для Университетского округа ПГПУ.
2. Стоимость программного продукта составила 146 773,6 руб.
3. Внедрение ИС «Университетский округ» окупится через 3 года и 20 дней.

Заключение

В результате выполнения выпускной квалификационной работы была спроектирована и разработана информационная система «Университетский округ» на примере Университетского округа Пермского Государственного Гуманитарно-Педагогического Университета.

Информационная система «Университетский округ» позволяет автоматизировать процесс сбора, хранения и обработки информации по планированию мероприятий и отчетам по результатам деятельности ЦИО и Университетского округа. Система спроектирована для удаленного доступа в многопользовательском режиме по технологии «клиент-сервер».

Для организации хранения данных в информационной системе было выполнено концептуальное и логическое проектирование. Практическая реализация данных моделей выполнена в СУБД MySQL. При разработке пользовательского интерфейса использовался фреймворк Django на языке программирования Python.

Внедрение ИС «Университетский округ» примерно окупится через 3 года и 20 дней.

При использовании информационной системы «Университетский округ» появляется ряд преимуществ:

- Экономия времени на сбор необходимой информации о мероприятиях УО;
- Экономия времени на передачу отчетности;
- Обеспечение сетевого взаимодействия ЦИО.

Поэтому можно сделать вывод, что внедрение информационной системы «Университетский округ» является целесообразным и экономически обоснованным.

Библиографический список

1. Осяк С.А., Газизова Т.В., Колокольникова З.У., Лобанова О.Б., Храмова Л.Н., Коршунова В.В. СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ В ПЕДАГОГИЧЕСКОМ ОБРАЗОВАНИИ // Современные проблемы науки и образования. – 2015. – № 1-1.; URL: <http://www.science-education.ru/ru/article/view?id=18081> (дата обращения: 04.06.2018).
2. Приказ ПГГПУ от 3.03.2009г. №8/80 "Об утверждении положения об университетском округе инновационных образовательных учреждений"
3. Приказ Министерства образования Пермского края от 02.07.2008г. №СЭД-26-01-04-66 «Об утверждении Положения о центрах инновационного опыта»
4. Университетский округ ПГГПУ [Электронный ресурс]//URL: <http://pspu.ru/university/universitetskij-okrug> (дата обращения:28.04.18)
5. Университетский округ НИУ ВШЭ [Электронный ресурс]//URL: <https://perm.hse.ru/okrug> (дата обращения:28.04.18)
6. Университетский округ ПГНИУ [Электронный ресурс]//URL: <http://www.psu.ru/universitetskaya-zhizn/universitetskij-okrug> (дата обращения:28.04.18)
7. Павлов И.Ю., Голубева Е.Г., Голубев В.О. Методика мониторинга официальных сайтов органов власти. Санкт-Петербург, 2011.
8. Симакина Н.И. Проектирование информационных систем [Электронный ресурс]. – Режим доступа: <https://moodle.pspu.ru/course/view.php?id=596> – (Дата обращения: 05.06.2017)
9. Трудовой кодекс Российской Федерации [Электронный ресурс]//URL: http://www.consultant.ru/document/cons_doc_LAW_34683/bb999f080ef12c25c328e1b5e4ffd35319d8bea6/ (Дата обращения: 21.06.2018 г.)

10. Зарплаты учителей в России [Электронный ресурс]//URL:
<http://visasam.ru/russia/rabotavrf/zarplata-uchitelya-v-rossii.html> (дата
обращения: 21.06.18)

```

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from django.contrib.auth.models import User

from common import models

admin.site.unregister(User)

class UserProfileInline(admin.StackedInline):
    model = models.UserProfile
    verbose_name = u'Профиль пользователя'
    verbose_name_plural = u'Профили пользователя'

class UserProfileAdmin(UserAdmin):
    inlines = [UserProfileInline]

admin.site.register(User, UserProfileAdmin)
admin.site.register(models.UniversityDepartment)
admin.site.register(models.CenterOfInnovationExperience)
admin.site.register(models.Project)
admin.site.register(models.ProjectReport)
admin.site.register(models.Event)
admin.site.register(models.EventFile)
admin.site.register(models.YearPlan)
admin.site.register(models.UniversityDepartmentYearPlan)
admin.site.register(models.UniversityDepartmentReport)

ADMIN_CIO = 'ADMIN_CIO'
SCIENTIFIC_DIRECTOR = 'SCIENTIFIC_DIRECTOR'
UNIVERSITY_DEPARTMENT_HEAD =

USER_ROLES = (
    (ADMIN_CIO, 'Администратор ЦИО'),
    (SCIENTIFIC_DIRECTOR, 'Научный руководитель ЦИО'),
    (UNIVERSITY_DEPARTMENT_HEAD, 'Руководитель
университетского округа'),
)

```

```

from django.contrib.auth.models import User
from django.db import models

from common.constants import USER_ROLES, SCIENTIFIC_DIRECTOR,
SYSADMIN, ADMIN_CIO, UNIVERSITY_DEPARTMENT_HEAD, \
MINISTRY_OF_EDUCATION

def get_user_full_name(self):
    return "%s %s %s (%s)" % (self.profile.last_name, self.profile.first_name,
self.profile.middle_name, self.username)

User.add_to_class("__str__", get_user_full_name)

class UserProfile(models.Model):
    user = models.OneToOneField(User, verbose_name='Пользователь',
related_name='profile', on_delete=models.CASCADE)
    first_name = models.CharField('Имя', max_length=255, null=True)
    last_name = models.CharField('Фамилия', max_length=255, null=True)
    middle_name = models.CharField('Отчество', max_length=255,
blank=True, null=True)
    role = models.CharField('Роль в системе', choices=USER_ROLES,
default=SCIENTIFIC_DIRECTOR, max_length=50)

    def __str__(self):
        return self.full_name

    @property
    def cio_admin(self):
        return
        CenterOfInnovationExperience.objects.filter(administrator=self.user).first()

    @property
    def cio_scientific_head(self):
        return
        CenterOfInnovationExperience.objects.filter(scientific_director=self.user).fi
rst()

    @property
    def is_sysadmin(self):
        return self.role == SYSADMIN

```

```
@property
def is_scientific_director(self):
    return self.role == SCIENTIFIC_DIRECTOR
```

```
@property
def is_admin_cio(self):
    return self.role == ADMIN_CIO
```

```
@property
def is_university_department_head(self):
    return self.role == UNIVERSITY_DEPARTMENT_HEAD
```

```
@property
def is_ministry_of_education(self):
    return self.role == MINISTRY_OF_EDUCATION
```

```
@property
def full_name(self):
    return "%s %s %s" % (self.last_name, self.first_name, self.middle_name or
    ")
```

```
@property
def short_name(self):
    first_name_i = self.first_name[0]
    middle_name_i = ( ' ' + self.middle_name[0] + '.') if self.middle_name is not
    None else "
    return "%s %s.%s" % (self.last_name, first_name_i, middle_name_i)
```

```
class UniversityDepartment(models.Model):
    name = models.CharField('Название', max_length=255)
    head = models.ForeignKey(User,
    related_name='university_departments',
    verbose_name='Руководитель',
    on_delete=models.CASCADE,
    null=True)
    address = models.TextField('Адрес')
    phone = models.CharField('Телефон', max_length=20)
    email = models.EmailField('Email')
    website = models.URLField('Сайт')
```

```
@property
def projects(self):
```

```
return
Project.objects.filter(center_of_innovation_experience__university_department=self)
```

```
class Meta:
    verbose_name = 'Университетский округ'
    verbose_name_plural = 'Университетские округа'

    def __str__(self):
        return self.name
```

```
class CenterOfInnovationExperience(models.Model):
    university_department = models.ForeignKey(UniversityDepartment,
        related_name='centers_of_innovation_experience',
        verbose_name='Университетский округ',
        on_delete=models.SET_NULL,
        null=True)
    name = models.CharField('Название образовательной организации',
        max_length=255)
    head = models.CharField('Директор', max_length=255)
    administrator = models.ForeignKey(User,
        related_name='admin_centers_of_innovation_experience',
        verbose_name='Администратор',
        on_delete=models.SET_NULL,
        blank=True,
        null=True)
    scientific_director = models.ForeignKey(User,
        related_name='sd_centers_of_innovation_experience',
        verbose_name='Научный руководитель',
        on_delete=models.SET_NULL,
        blank=True,
        null=True)
    address = models.TextField('Адрес', default='')
    phone = models.CharField('Телефон', max_length=20, default='')
    email = models.EmailField('Email', default='')
    website = models.URLField('Сайт', default='')
```

```
class Meta:
    verbose_name = 'Центр инновационного опыта'
    verbose_name_plural = 'Центры инновационного опыта'
```

```
@property
def events(self):
```

```
return Event.objects.filter(project__center_of_innovation_experience=self)
```

```
@property
```

```
def reports(self):
```

```
return
```

```
ProjectReport.objects.filter(project__center_of_innovation_experience=self)
```

```
def __str__(self):
```

```
return self.name
```

```
class Project(models.Model):
```

```
center_of_innovation_experience
```

```
=
```

```
models.ForeignKey(CenterOfInnovationExperience,
```

```
related_name='projects',
```

```
verbose_name='Центр инновационного опыта',
```

```
on_delete=models.CASCADE)
```

```
name = models.CharField('Название проекта', max_length=255)
```

```
author = models.CharField('ФИО автора', max_length=255)
```

```
head = models.CharField('ФИО руководителя', max_length=255)
```

```
site_url = models.URLField('Ссылка на раздел на сайте, посвященный проекту')
```

```
term = models.CharField('Плановый срок реализации', max_length=255, choices=(
```

```
('year_1', '1 год'),
```

```
('year_2', '2 года'),
```

```
('year_3', '3 года'),
```

```
), null=True)
```

```
actuality = models.TextField('Актуальность', null=True, blank=True, default="")
```

```
objective = models.CharField('Цель', max_length=255, null=True, blank=True, default="")
```

```
tasks = models.TextField('Задачи', null=True, blank=True, default="")
```

```
innovations = models.TextField('Краткое обоснование инновационности проекта', null=True, blank=True, default="")
```

```
products = models.TextField('Ожидаемые продукты проекта', null=True, blank=True, default="")
```

```
innovation_content = models.TextField('На разработку какого иного инновационного контента направлен проект ', null=True, blank=True, default="")
```

```
approbation_and_dissemination = models.TextField('Способы апробации и диссеминации результатов', null=True, blank=True, default="")
```

```
risks = models.TextField('Риски и пути преодоления', null=True, blank=True, default="")
```

```
self_evaluation_mechanism = models.TextField('Механизм самооценки ',
null=True, blank=True, default="")
```

```
class Meta:
verbose_name = 'Проект'
verbose_name_plural = 'Проекты'
```

```
@property
def held_events_count(self):
return Event.objects.filter(project=self, status='held').count()
```

```
@property
def total_events_count(self):
return Event.objects.filter(project=self).count()
```

```
def __str__(self):
return self.name
```

```
class ProjectReport(models.Model):
project = models.ForeignKey(Project,
related_name='reports',
verbose_name='Проект',
on_delete=models.CASCADE)
date = models.DateField('Дата отчета')
year = models.IntegerField('Отчетный год')
type = models.CharField('Тип отчета', choices=(
('annual', 'Ежегодный'),
('final', 'Итоговый'),
), default='annual', max_length=255)
law_support = models.TextField('Нормативно-правовое обеспечение',
null=True, blank=True, default="")
staffing = models.TextField('Кадровое обеспечение', null=True,
blank=True, default="")
information_support = models.TextField('Информационное обеспечение ',
null=True, blank=True, default="")
project_experience = models.TextField('Опыт успешно реализованных
проектов', null=True, blank=True, default="")
conformity = models.TextField(
'Описание соответствия содержания изначально запланированного
проекта и полученных результатов ', null=True, blank=True, default="")
current_product_actuality = models.TextField('Описание текущей
актуальности продуктов', null=True, blank=True, default="")
```

```

products = models.TextField('Продукты ', null=True, blank=True,
default='')
qc_methods = models.TextField('Описание методов и критериев
мониторинга качества проекта ', null=True, blank=True, default='')
self_evaluation_results = models.TextField('Результаты самооценки,
мониторинга ', null=True, blank=True, default='')
results = models.TextField('Перечень достигнутых результатов',
null=True, blank=True, default='')
conclusions = models.TextField('Выводы по результатам деятельности',
null=True, blank=True, default='')

```

```

@property
def events(self):
if self.type == 'annual':
return Event.objects.filter(project=self.project, status='held',
date__year=self.year).order_by('date')
return Event.objects.filter(project=self.project, status='held').order_by('date')

```

```

class Meta:
verbose_name = 'Отчет'
verbose_name_plural = 'Отчеты'

```

```

class UniversityDepartmentYearPlan(models.Model):
university_department = models.ForeignKey(UniversityDepartment,
related_name='year_plans',
verbose_name='Университетский округ',
on_delete=models.SET_NULL,
null=True)
year = models.IntegerField('Год')

```

```

@property
def events(self):
return Event.objects.filter(date__year=self.year).order_by('date')

```

```

class Meta:
verbose_name = 'План мероприятий университетского округа'
verbose_name_plural = 'Планы мероприятий университетского округа'

```

```

def __str__(self):
return str(self.year)

```

```

class UniversityDepartmentReport(models.Model):

```

```

university_department = models.ForeignKey(UniversityDepartment,
related_name='reports',
verbose_name='Университетский округ',
on_delete=models.SET_NULL,
null=True)
date = models.DateField('Дата отчета')
year = models.IntegerField('Год')
type = models.CharField('Тип отчета', choices=(
('annual', 'Ежегодный'),
('final', 'Итоговый'),
), default='annual', max_length=255)
conclusions = models.TextField('Выводы', null=True, blank=True)

```

```

@property
def events(self):
    if self.type == 'annual':
        return
    Event.objects.filter(date__year=self.year).order_by('date').order_by('project__center_of_innovation_experience', 'project', 'date')
    return Event.objects.filter(date__year__lte=self.year,
date__year__gte=self.year-
2).order_by('project__center_of_innovation_experience', 'project', 'date')

```

```

@property
def reports(self):
    if self.type == 'annual':
        return ProjectReport.objects.filter(year=self.year).order_by('date')
    return ProjectReport.objects.filter(year__lte=self.year, year__gte=self.year-
2).order_by('date')

```

```

class Meta:
    verbose_name = 'Отчет университетского округа'
    verbose_name_plural = 'Отчеты университетского округа'

    def __str__(self):
        return str(self.year) + ' ' + self.get_type_display()

```

```

class YearPlan(models.Model):
    project = models.ForeignKey(Project,
related_name='plan',
verbose_name='Проект',
on_delete=models.CASCADE)
    year = models.IntegerField('Год')

```

```

current_objective = models.CharField('Цель на год', max_length=255)
current_tasks = models.CharField('Задачи по реализации проекта на год',
max_length=255)
terms = models.CharField('Срок реализации', max_length=255, null=True)

```

```

@property
def events(self):
return Event.objects.filter(project=self.project,
date__year=self.year).order_by('date')

```

```

class Meta:
verbose_name = 'План реализации'
verbose_name_plural = 'Планы реализации'

```

```

def __str__(self):
return self.project.name + " " + str(self.year)

```

```

class Event(models.Model):
project = models.ForeignKey(Project,
related_name='events',
verbose_name='Проект',
on_delete=models.CASCADE)
name = models.CharField('Название мероприятия', max_length=255)
description = models.TextField('Описание', max_length=255, null=True,
blank=True, default='')
date = models.DateField('Дата')
objective = models.CharField('Цель мероприятия', max_length=255,
null=True, default='')
result = models.TextField('Результат', null=True, blank=True, default='')
result_claims = models.CharField('Требования к результату',
max_length=255, null=True, blank=True, default='')
place = models.CharField('Место проведения', max_length=255)
participants_category = models.CharField('Категория участников',
max_length=255, null=True, blank=True, default='')
responsible = models.CharField('Ответственный за мероприятие',
max_length=255)
status = models.CharField('Статус', max_length=255, choices=(
('planned', 'Мероприятие запланировано'),
('held', 'Мероприятие проведено'),
('cancelled', 'Мероприятие отменено'),
), default='planned')

```

```

class Meta:

```

```

verbose_name = 'Мероприятие'
verbose_name_plural = 'Мероприятия'

def __str__(self):
    return "%s - %s" % (self.name, str(self.date))

class YearPlanEvent(models.Model):
    plan = models.ForeignKey(YearPlan, blank=True, null=True,
        related_name='planned_events',
        on_delete=models.CASCADE)
    event = models.ForeignKey(Event, verbose_name='Мероприятие',
        on_delete=models.CASCADE)

class EventFile(models.Model):
    event = models.ForeignKey(Event, related_name='files',
        verbose_name='Мероприятие', on_delete=models.CASCADE)
    description = models.CharField('Описание', max_length=255)
    file = models.FileField('Файл')

class Meta:
    verbose_name = 'Подтверждающие документы'
    verbose_name_plural = 'Подтверждающий документ'

from django.contrib.auth.models import User
from django.forms import Form

import common.models

from django.contrib.auth.forms import AuthenticationForm
from django import forms

from common.constants import ADMIN_CIO, SCIENTIFIC_DIRECTOR

class AuthForm(AuthenticationForm):
    username = forms.CharField(label=u"Логин", max_length=254,
        widget=forms.TextInput(
            attrs={'placeholder': u'Имя пользователя', 'class': 'form-control input-sm'}))
    password = forms.CharField(label=u"Пароль",
        widget=forms.PasswordInput(
            attrs={'placeholder': u'Пароль', 'class': 'form-control input-sm'}))

```

```

class CIOForm(forms.ModelForm):
    university_department = forms.ModelChoiceField(
        queryset=common.models.UniversityDepartment.objects.all(),
        widget=forms.HiddenInput(attrs={'class': 'form-control input-sm'}),
        initial=common.models.UniversityDepartment.objects.all().first()
    )
    administrator = forms.ModelChoiceField(
        queryset=User.objects.filter(profile__role=ADMIN_CIO),
        widget=forms.Select(attrs={'class': 'form-control input-sm'}),
        required=False,
        label='Администратор'
    )
    scientific_director = forms.ModelChoiceField(
        queryset=User.objects.filter(profile__role=SCIENTIFIC_DIRECTOR),
        widget=forms.Select(attrs={'class': 'form-control input-sm'}),
        required=False,
        label='Научный руководитель'
    )

    def clean_administrator(self):
        administrator = self.cleaned_data.get('administrator', None)
        if administrator:
            if administrator.profile.cio_admin and
administrator.profile.cio_admin != self.instance:
                raise forms.ValidationError('У данного администратора уже
назначен ЦИО')
            return administrator

    def clean_scientific_director(self):
        scientific_director = self.cleaned_data.get('scientific_director', None)
        if scientific_director:
            if scientific_director.profile.cio_scientific_head and
scientific_director.profile.cio_scientific_head != self.instance:
                raise forms.ValidationError('У данного научного руководителя
уже назначен ЦИО')
            return scientific_director

class Meta:
    model = common.models.CenterOfInnovationExperience
    fields = '__all__'
    widgets = {

```

```

        'name': forms.TextInput(attrs={'class': 'form-control input-sm'}),
        'head': forms.TextInput(attrs={'class': 'form-control input-sm'}),
        'address': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'phone': forms.TextInput(attrs={'class': 'form-control input-sm'}),
        'email': forms.TextInput(attrs={'class': 'form-control input-sm'}),
        'website': forms.TextInput(attrs={'class': 'form-control input-sm'}),
    }

```

```

class ProjectRequiredDataForm(forms.ModelForm):
    def __init__(self, *args, **kwargs):
        cio_pk = kwargs.pop('cio_pk', None)
        super().__init__(*args, **kwargs)
        if cio_pk:
            self.fields['center_of_innovation_experience'].initial = cio_pk

    class Meta:
        model = common.models.Project
        fields = ['center_of_innovation_experience', 'name', 'author', 'head',
'site_url', 'term']
        widgets = {
            'center_of_innovation_experience': forms.HiddenInput(),
            'name': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'author': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'head': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'site_url': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'term': forms.Select(attrs={'class': 'form-control input-sm'}),
        }

```

```

class ProjectGeneralDataForm(forms.ModelForm):

    class Meta:
        model = common.models.Project
        fields = ['actuality', 'objective', 'tasks', 'innovations', 'products',
'innovation_content',
                'approbation_and_dissemination', 'risks',
'self_evaluation_mechanism']
        widgets = {
            'actuality': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
            'objective': forms.TextInput(attrs={'class': 'form-control input-sm'}),

```

```

        'tasks': forms.Textarea(attrs={'class': 'form-control input-sm', 'rows':
2}),
        'innovations': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'products': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'innovation_content': forms.Textarea(attrs={'class': 'form-control
input-sm', 'rows': 2}),
        'approbation_and_dissemination': forms.Textarea(attrs={'class':
'form-control input-sm', 'rows': 2}),
        'risks': forms.Textarea(attrs={'class': 'form-control input-sm', 'rows':
2}),
        'self_evaluation_mechanism': forms.Textarea(attrs={'class': 'form-
control input-sm', 'rows': 2}),
    }

```

```

class EventRequiredDataForm(forms.ModelForm):

```

```

    def __init__(self, *args, **kwargs):
        project_pk = kwargs.pop('project_pk', None)
        super().__init__(*args, **kwargs)
        if project_pk:
            self.fields['project'].initial = project_pk

```

```

    class Meta:

```

```

        model = common.models.Event
        fields = ['project', 'name', 'date', 'objective', 'result_claims', 'responsible',
'description', 'place',
                'participants_category']
        widgets = {
            'project': forms.HiddenInput(),
            'name': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'date': forms.DateInput(attrs={'class': 'form-control input-sm
datepicker'}),
            'objective': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'result_claims': forms.TextInput(attrs={'class': 'form-control input-
sm'}),
            'responsible': forms.TextInput(attrs={'class': 'form-control input-
sm'}),
            'description': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
            'place': forms.TextInput(attrs={'class': 'form-control input-sm'}),
            'participants_category': forms.TextInput(attrs={'class': 'form-control
input-sm'})

```

```
}
```

```
class EventStatusForm(forms.ModelForm):
    class Meta:
        model = common.models.Event
        fields = ['status']
        widgets = {
            'status': forms.Select(attrs={'class': 'form-control input-sm'}),
        }
```

```
class EventResultForm(forms.ModelForm):

    class Meta:
        model = common.models.Event
        fields = ['result']
        widgets = {
            'result': forms.Textarea(attrs={'class': 'form-control input-sm', 'rows':
2}),
        }
```

```
class EventFileForm(forms.ModelForm):
    def __init__(self, *args, **kwargs):
        event_pk = kwargs.pop('event_pk', None)
        super().__init__(*args, **kwargs)
        if event_pk:
            self.fields['event'].initial = event_pk

    class Meta:
        model = common.models.EventFile
        fields = '__all__'
        widgets = {
            'event': forms.HiddenInput(),
            'description': forms.TextInput(attrs={'class': 'form-control input-
sm'}),
            'file': forms.FileInput(attrs={'class': 'form-control input-sm'}),
        }
```

```
class ProjectReportRequiredForm(forms.ModelForm):
    def __init__(self, *args, **kwargs):
        project_pk = kwargs.pop('project_pk', None)
```

```

    super().__init__(*args, **kwargs)
    if project_pk:
        self.fields['project'].initial = project_pk

    class Meta:
        model = common.models.ProjectReport
        fields = ['project', 'date', 'year', 'type']
        widgets = {
            'project': forms.HiddenInput,
            'date': forms.DateInput(attrs={'class': 'form-control input-sm',
            datepicker'})),
            'type': forms.Select(attrs={'class': 'form-control input-sm'}),
            'year': forms.NumberInput(attrs={'class': 'form-control input-sm'}),
        }

```

```

class ProjectReportResourcesForm(forms.ModelForm):

```

```

    class Meta:
        model = common.models.ProjectReport
        fields = ['law_support', 'staffing', 'information_support',
        'project_experience']
        widgets = {
            'law_support': forms.Textarea(attrs={'class': 'form-control input-sm',
            'rows': 2}),
            'staffing': forms.Textarea(attrs={'class': 'form-control input-sm',
            'rows': 2}),
            'information_support': forms.Textarea(attrs={'class': 'form-control
            input-sm', 'rows': 2}),
            'project_experience': forms.Textarea(attrs={'class': 'form-control
            input-sm', 'rows': 2}),
        }

```

```

class ProjectReportRealizationForm(forms.ModelForm):

```

```

    class Meta:
        model = common.models.ProjectReport
        fields = ['conformity', 'current_product_actuality', 'products',
        'qc_methods',
            'self_evaluation_results']
        widgets = {

```

```

        'conformity': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'current_product_actuality': forms.Textarea(attrs={'class': 'form-
control input-sm', 'rows': 2}),
        'products': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'qc_methods': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        'self_evaluation_results': forms.Textarea(attrs={'class': 'form-control
input-sm', 'rows': 2}),

    }

```

```

class ProjectReportResultsForm(forms.ModelForm):

```

```

    class Meta:
        model = common.models.ProjectReport
        fields = ['results']
        widgets = {
            'results': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        }

```

```

class ProjectReportConclusionForm(forms.ModelForm):

```

```

    class Meta:
        model = common.models.ProjectReport
        fields = ['conclusions']
        widgets = {
            'conclusions': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        }

```

```

class YearPlanForm(forms.ModelForm):

```

```

    def __init__(self, *args, **kwargs):
        project_pk = kwargs.pop('project_pk', None)
        super().__init__(*args, **kwargs)
        if project_pk:
            self.fields['project'].initial = project_pk

```

```

class Meta:
    model = common.models.YearPlan
    fields = '__all__'
    widgets = {
        'project': forms.HiddenInput(),
        'year': forms.NumberInput(attrs={'class': 'form-control input-sm'}),
        'current_objective': forms.TextInput(attrs={'class': 'form-control
input-sm'}),
        'current_tasks': forms.TextInput(attrs={'class': 'form-control input-
sm'}),
        'terms': forms.TextInput(attrs={'class': 'form-control input-sm'}),
    }

```

```

class YearPlanEvent(forms.ModelForm):
    def __init__(self, project=None, **kwargs):
        super().__init__(**kwargs)
        if project:
            self.fields['event'].queryset =
common.models.Event.objects.filter(project=project)

```

```

class Meta:
    model = common.models.YearPlanEvent
    fields = ['event']
    widgets = {
        'event': forms.Select(attrs={'class': 'form-control input-sm'})
    }

```

```

class UniversityDepartmentYearPlanForm(forms.ModelForm):

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['university_department'].initial =
common.models.UniversityDepartment.objects.first()

```

```

class Meta:
    model = common.models.UniversityDepartmentYearPlan
    fields = '__all__'
    widgets = {
        'university_department': forms.HiddenInput(),
        'year': forms.NumberInput(attrs={'class': 'form-control input-sm'}),
    }

```

```

class UniversityDepartmentReportForm(forms.ModelForm):

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['university_department'].initial =
common.models.UniversityDepartment.objects.first()

    class Meta:
        model = common.models.UniversityDepartmentReport
        exclude = ['conclusions']
        widgets = {
            'university_department': forms.HiddenInput(),
            'date': forms.DateInput(attrs={'class': 'form-control input-sm
datepicker'})),
            'year': forms.NumberInput(attrs={'class': 'form-control input-sm'}),
            'type': forms.Select(attrs={'class': 'form-control input-sm'}),
        }

class UniversityDepartmentReportConclusionsForm(forms.ModelForm):
    class Meta:
        model = common.models.UniversityDepartmentReport
        fields = ['conclusions']
        widgets = {
            'conclusions': forms.Textarea(attrs={'class': 'form-control input-sm',
'rows': 2}),
        }

class EventsReportForm(Form):
    date_from = forms.DateField(label='Начало периода',
widget=forms.DateInput(attrs={'class': 'form-control input-sm datepicker'}))
    date_to = forms.DateField(label='Окончание периода',
widget=forms.DateInput(attrs={'class': 'form-control input-sm datepicker'}))
import json

import datetime

from django.urls import reverse

import common.models
from django.contrib.auth import authenticate, login
from django.http import HttpResponse, JsonResponse

```

```

from django.shortcuts import render, get_object_or_404, redirect

# Create your views here.
from django.views.generic import TemplateView, FormView, DetailView,
ListView, DeleteView, UpdateView, CreateView

from common import forms
from common.helpers import render_to_pdf
from common.mixins import MenuMixin, ModalFormMixin,
ListItemFormMixin, DeleteAjaxMixin

class Login(ModalFormMixin, FormView):
    template_name = 'common/forms/modal_form.html'
    form_class = forms.AuthForm
    success_url = '/'
    modal_form_title = u'Вход в систему'
    modal_form_submit_text = u'Вход'

    def post(self, request, *args, **kwargs):
        form = self.form_class(data=request.POST)
        if form.is_valid():
            username = request.POST.get('username', '')
            password = request.POST.get('password', '')
            user = authenticate(username=username, password=password)
            if user is not None:
                login(request, user)
                return HttpResponseRedirect(json.dumps({'redirect': '/'}),
content_type='application/json')
            return super(Login, self).post(request, *args, **kwargs)

class Home(MenuMixin, TemplateView):
    menu_page = 'home'
    template_name = 'common/pages/index.html'

class UniversityDepartmentPage(MenuMixin, DetailView):
    menu_page = 'university_department'
    template_name = 'common/pages/university_department.html'
    model = common.models.UniversityDepartment

    def get_object(self, queryset=None):
        return common.models.UniversityDepartment.objects.all().first()

```

```
class CIODashboardPage(MenuMixin, ListView):
    menu_page = 'cio_dashboard'
    template_name = 'common/pages/cio.html'
    model = common.models.CenterOfInnovationExperience
```

```
class CIODetailPage(MenuMixin, DetailView):
    menu_page = 'cio_dashboard'
    template_name = 'common/pages/cio_detail.html'
    model = common.models.CenterOfInnovationExperience
```

```
class CIOUpdate(ListItemFormMixin, UpdateView):
    model = common.models.CenterOfInnovationExperience
    form_class = forms.CIOForm
    list_template_name = 'common/lists/cio_list.html'
    detail_template_name = 'common/details/cio.html'
    modal_form_title = 'Редактировать ЦИО'
```

```
class ProjectDetailPage(MenuMixin, DetailView):
    menu_page = 'projects_dashboard'
    template_name = 'common/pages/project_detail.html'
    model = common.models.Project
```

```
class ProjectCreate(ListItemFormMixin, CreateView):
    model = common.models.Project
    form_class = forms.ProjectRequiredDataForm
    list_template_name = 'common/lists/projects_list.html'
    modal_form_title = 'Добавить проект'
```

```
    def get_object_list(self):
        user_profile = self.request.user.profile
        cio_pk = self.kwargs.get('cio_pk', None) or user_profile.cio_admin or
        user_profile.cio_scientific_head
        return
    self.model.objects.filter(center_of_innovation_experience=cio_pk)
```

```
    def get_form_kwargs(self):
        kwargs = super().get_form_kwargs()
        kwargs.update({'cio_pk': self.kwargs.get('cio_pk', None)})
```

```
return kwargs
```

```
class ProjectUpdateRequiredData(ListItemFormMixin, UpdateView):
    model = common.models.Project
    form_class = forms.ProjectRequiredDataForm
    list_template_name = 'common/lists/projects_list.html'
    detail_template_name = 'common/details/project_required_data.html'
    modal_form_title = 'Редактировать проект'

    def get_object_list(self):
        cio_pk = self.get_object().center_of_innovation_experience
        return
self.model.objects.filter(center_of_innovation_experience=cio_pk)
```

```
class ProjectUpdateGeneralData(ListItemFormMixin, UpdateView):
    model = common.models.Project
    form_class = forms.ProjectGeneralDataForm
    list_template_name = 'common/lists/projects_list.html'
    detail_template_name = 'common/details/project_general_data.html'
    modal_form_title = 'Редактировать проект'
```

```
class ProjectDelete>DeleteAjaxMixin, DeleteView):
    model = common.models.Project
    list_template_name = 'common/lists/projects_list.html'

    def get_object_list(self, obj=None):
        return
self.model.objects.filter(center_of_innovation_experience=obj.center_of_innovation_experience)
```

```
class EventProjectDetailPage(MenuMixin, DetailView):
    menu_page = 'projects_dashboard'
    template_name = 'common/pages/event_detail.html'
    model = common.models.Event
```

```
class EventCreateForProject(ListItemFormMixin, CreateView):
    model = common.models.Event
    form_class = forms.EventRequiredDataForm
    list_template_name = 'common/lists/events_list_project.html'
```

```
modal_form_title = 'Добавить мероприятие'
```

```
def get_object_list(self):  
    return self.model.objects.filter(project=self.kwargs.get('project_pk'))
```

```
def get_form_kwargs(self):  
    kwargs = super().get_form_kwargs()  
    kwargs.update({'project_pk': self.kwargs.get('project_pk', None)})  
    return kwargs
```

```
class EventUpdateRequiredData(ListItemFormMixin, UpdateView):  
    model = common.models.Event  
    form_class = forms.EventRequiredDataForm  
    list_template_name = 'common/lists/events_list_project.html'  
    detail_template_name = 'common/details/event_required_data.html'  
    modal_form_title = 'Редактировать мероприятие'
```

```
def get_object_list(self):  
    return self.model.objects.filter(project=self.get_object().project)
```

```
class EventUpdateResult(ListItemFormMixin, UpdateView):  
    model = common.models.Event  
    form_class = forms.EventResultForm  
    detail_template_name = 'common/details/event_result.html'  
    modal_form_title = 'Редактировать мероприятие'
```

```
class EventUpdateStatus(ListItemFormMixin, UpdateView):  
    model = common.models.Event  
    form_class = forms.EventStatusForm  
    detail_template_name = 'common/details/event_required_data.html'  
    modal_form_title = 'Сменить статус мероприятия'
```

```
class EventDelete>DeleteAjaxMixin, DeleteView):  
    model = common.models.Event  
    list_template_name = 'common/lists/events_list_project.html'  
  
    def get_object_list(self, obj=None):  
        return self.model.objects.filter(project=obj.project)
```

```

class EventFileCreate(ListItemFormMixin, CreateView):
    model = common.models.EventFile
    form_class = forms.EventFileForm
    list_template_name = 'common/lists/event_files.html'
    modal_form_title = 'Добавить подтверждающий документ'

    def get_object_list(self):
        return self.model.objects.filter(event=self.kwargs.get('event_pk'))

    def get_form_kwargs(self):
        kwargs = super().get_form_kwargs()
        kwargs.update({'event_pk': self.kwargs.get('event_pk', None)})
        return kwargs

```

```

class EventFileUpdate(ListItemFormMixin, UpdateView):
    model = common.models.EventFile
    form_class = forms.EventFileForm
    list_template_name = 'common/lists/event_files.html'
    modal_form_title = 'Обновить подтверждающий документ'

    def get_object_list(self):
        return self.model.objects.filter(event=self.get_object().event)

```

```

class EventFileDelete>DeleteAjaxMixin, DeleteView):
    model = common.models.EventFile
    list_template_name = 'common/lists/event_files.html'

    def get_object_list(self, obj=None):
        return self.model.objects.filter(event=obj.event)

```

```

class ReportProjectDetailPage(MenuMixin, DetailView):
    menu_page = 'reports_dashboard'
    template_name = 'common/pages/report_detail.html'
    model = common.models.ProjectReport

```

```

class ReportCreateForProject(ListItemFormMixin, CreateView):
    model = common.models.ProjectReport
    form_class = forms.ProjectReportRequiredForm
    list_template_name = 'common/lists/reports_list_project.html'
    modal_form_title = 'Добавить отчет'

```

```

def get_object_list(self):
    return self.model.objects.filter(project=self.kwargs.get('project_pk'))

def get_form_kwargs(self):
    kwargs = super().get_form_kwargs()
    kwargs.update({'project_pk': self.kwargs.get('project_pk', None)})
    return kwargs

```

```

class ReportUpdateRequiredData(ListItemFormMixin, UpdateView):
    model = common.models.ProjectReport
    form_class = forms.ProjectReportRequiredForm
    list_template_name = 'common/lists/reports_list_project.html'
    detail_template_name = 'common/details/report_required_data.html'
    modal_form_title = 'Редактировать отчет'

```

```

def get_object_list(self):
    return self.model.objects.filter(project=self.get_object().project)

```

```

class ReportUpdateResources(ListItemFormMixin, UpdateView):
    model = common.models.ProjectReport
    form_class = forms.ProjectReportResourcesForm
    detail_template_name = 'common/details/report_resources.html'
    modal_form_title = 'Ресурсное обеспечение'

```

```

class ReportUpdateRealization(ListItemFormMixin, UpdateView):
    model = common.models.ProjectReport
    form_class = forms.ProjectReportRealizationForm
    detail_template_name = 'common/details/report_realization.html'
    modal_form_title = 'Реализация проекта'

```

```

class ReportUpdateConclusion(ListItemFormMixin, UpdateView):
    model = common.models.ProjectReport
    form_class = forms.ProjectReportConclusionForm
    detail_template_name = 'common/details/report_conclusion.html'
    modal_form_title = 'Выводы по проекту'

```

```

class ReportPrint(DetailView):
    model = common.models.ProjectReport

```

```

template_name = 'common/reports/project_report.html'

def get(self, request, *args, **kwargs):
    data = {
        'object': self.get_object,
        'print_date': datetime.datetime.now()
    }
    pdf = render_to_pdf(self.template_name, data)
    return HttpResponse(pdf, content_type='application/pdf')

class ReportDelete(DeleteAjaxMixin, DeleteView):
    model = common.models.ProjectReport
    list_template_name = 'common/lists/reports_list_project.html'

    def get_object_list(self, obj=None):
        return self.model.objects.filter(project=obj.project)

class YearPlanDetailPage(MenuMixin, DetailView):
    menu_page = 'projects_dashboard'
    template_name = 'common/pages/year_plan_detail.html'
    model = common.models.YearPlan

class YearPlanCreateForProject(ListItemFormMixin, CreateView):
    model = common.models.YearPlan
    form_class = forms.YearPlanForm
    list_template_name = 'common/lists/year_plans_list_project.html'
    modal_form_title = 'Добавить пункт плана'

    def get_object_list(self):
        return self.model.objects.filter(project=self.kwargs.get('project_pk'))

    def get_form_kwargs(self):
        kwargs = super().get_form_kwargs()
        kwargs.update({'project_pk': self.kwargs.get('project_pk', None)})
        return kwargs

class YearPlanUpdateRequiredData(ListItemFormMixin, UpdateView):
    model = common.models.YearPlan
    form_class = forms.YearPlanForm
    list_template_name = 'common/lists/year_plans_list_project.html'

```

```
detail_template_name = 'common/details/year_plan_required_data.html'
modal_form_title = 'Редактировать пункт плана'
```

```
def get_object_list(self):
    return self.model.objects.filter(project=self.get_object().project)
```

```
class YearPlanPrint(DetailView):
    model = common.models.YearPlan
    template_name = 'common/reports/year_plan_report.html'

    def get(self, request, *args, **kwargs):
        data = {
            'object': self.get_object,
            'print_date': datetime.datetime.now()
        }
        pdf = render_to_pdf(self.template_name, data)
        return HttpResponse(pdf, content_type='application/pdf')
```

```
class YearPlanDelete(DeleteAjaxMixin, DeleteView):
    model = common.models.YearPlan
    list_template_name = 'common/lists/year_plans_list_project.html'

    def get_object_list(self, obj=None):
        return self.model.objects.filter(project=obj.project)
```

```
class YearPlanUniversityDepartmentDetailPage(MenuMixin, DetailView):
    menu_page = 'university_department'
    template_name = 'common/pages/year_plan__university_department_detail.html'
    model = common.models.UniversityDepartmentYearPlan
```

```
class YearPlanUniversityDepartmentCreate(ListItemFormMixin,
CreateView):
    model = common.models.UniversityDepartmentYearPlan
    form_class = forms.UniversityDepartmentYearPlanForm
    list_template_name = 'common/lists/year_plans_list_university_department.html'
    modal_form_title = 'Добавить план'
```

```

class YearPlanUniversityDepartmentUpdate(ListItemFormMixin,
UpdateView):
    model = common.models.UniversityDepartmentYearPlan
    form_class = forms.UniversityDepartmentYearPlanForm
    list_template_name =
'common/lists/year_plans_list_university_department.html'
    modal_form_title = 'Обновить план'

class YearPlanUniversityDepartmentPrint(DetailView):
    model = common.models.UniversityDepartmentYearPlan
    template_name =
'common/reports/year_plan_university_department_report.html'

    def get(self, request, *args, **kwargs):
        data = {
            'object': self.get_object,
            'print_date': datetime.datetime.now()
        }
        pdf = render_to_pdf(self.template_name, data)
        return HttpResponse(pdf, content_type='application/pdf')

class YearPlanUniversityDepartmentDelete(DeleteAjaxMixin, DeleteView):
    model = common.models.UniversityDepartmentYearPlan
    list_template_name =
'common/lists/year_plans_list_university_department.html'

class ReportUniversityDepartmentDetailPage(MenuMixin, DetailView):
    menu_page = 'university_department'
    template_name =
'common/pages/report_university_department_detail.html'
    model = common.models.UniversityDepartmentReport

class ReportUniversityDepartmentCreate(ListItemFormMixin, CreateView):
    model = common.models.UniversityDepartmentReport
    form_class = forms.UniversityDepartmentReportForm
    list_template_name =
'common/lists/reports_list_university_department.html'
    modal_form_title = 'Добавить отчет'

```

```

class ReportUniversityDepartmentUpdate(ListItemFormMixin,
UpdateView):
    model = common.models.UniversityDepartmentReport
    form_class = forms.UniversityDepartmentReportForm
    list_template_name = 'common/lists/reports_list_university_department.html'
    modal_form_title = 'Обновить отчет'

```

```

class ReportUniversityDepartmentUpdateConclusion(ListItemFormMixin,
UpdateView):
    model = common.models.UniversityDepartmentReport
    form_class = forms.UniversityDepartmentReportConclusionsForm
    detail_template_name = 'common/details/report_conclusion.html'
    modal_form_title = 'Выводы по проекту'

```

```

class ReportUniversityDepartmentPrint(DetailView):
    model = common.models.UniversityDepartmentReport
    template_name = 'common/reports/university_department_report.html'

    def get(self, request, *args, **kwargs):
        data = {
            'object': self.get_object,
            'print_date': datetime.datetime.now()
        }
        pdf = render_to_pdf(self.template_name, data)
        return HttpResponse(pdf, content_type='application/pdf')

```

```

class ReportUniversityDepartmentDelete>DeleteAjaxMixin, DeleteView):
    model = common.models.UniversityDepartmentReport
    list_template_name = 'common/lists/university_department_report.html'

```

```

class StatReportUniversityDepartmentPrint(DetailView):
    model = common.models.UniversityDepartmentReport
    template_name = 'common/reports/stat_report.html'

    def get(self, request, *args, **kwargs):
        data = {
            'events': self.get_object().events,
            'object': self.get_object(),
            'print_date': datetime.datetime.now()
        }

```

```

    }
    pdf = render_to_pdf(self.template_name, data)
    return HttpResponse(pdf, content_type='application/pdf')

class EventsReport(TemplateView):
    modal_form_submit_text = u'Просмотр отчета'
    modal_form_title = u'Отчет о мероприятиях ЦИО'
    template_name = 'common/reports/cio_events_report.html'

    def get(self, request, *args, **kwargs):
        params = request.GET
        df = params.get('date_from', None)
        dt = params.get('date_to', None)
        if df and dt:
            date_from = datetime.datetime.strptime(df, '%Y-%m-%d')
            date_to = datetime.datetime.strptime(dt, '%Y-%m-%d')
            data = {
                'date_from': date_from,
                'date_to': date_to,
                'events':
common.models.Event.objects.filter(date__gte=date_from,
date__lte=date_to).order_by('date',
'project__center_of_innovation_experience'),
                'print_date': datetime.datetime.now()
            }
            pdf = render_to_pdf(self.template_name, data)
            return HttpResponse(pdf, content_type='application/pdf')
        return render(request, 'common/forms/event_report_form.html', {'form':
forms.EventsReportForm(), 'form_title': 'Отчет по мероприятиям',
'submit_text': 'Печать отчета'})

    def post(self, request, *args, **kwargs):
        form = forms.EventsReportForm(request.POST)
        if form.is_valid():
            date_from = form.cleaned_data.get('date_from').strftime('%Y-%m-%d')
            date_to = form.cleaned_data.get('date_to').strftime('%Y-%m-%d')
            return JsonResponse({'redirect':
reverse('common:events_report_print') + '?date_from=%s&date_to=%s' %
(date_from, date_to), 'blank': True})
from django.conf.urls import url
from django.contrib.auth.views import login, logout, login_required

```

```

from common.forms import AuthForm
from . import views

app_name = 'common'
urlpatterns = [
    url(r'^login_page/$', login, {'authentication_form': AuthForm},
name="login"),
    url(r'^login/$', views.Login.as_view(), name="login"),
    url('logout/', logout, {'next_page': '/'}, name="logout"),

    url(r'^university_department/$',
login_required(views.UniversityDepartmentPage.as_view()),
name="university_department"),

    url(r'^cio/$', login_required(views.CIODashboardPage.as_view()),
name="cio_dashboard"),
    url(r'^cio/(?P<pk>[0-9]+)$',
login_required(views.CIODetailPage.as_view()), name="cio_detail"),
    url(r'^cio/(?P<pk>[0-9]+)/update$',
login_required(views.CIOUpdate.as_view()), name="cio_update"),

    url(r'^cio/(?P<cio_pk>[0-9]+)/projects/create$',
login_required(views.ProjectCreate.as_view()), name="project_create"),
    url(r'^projects/(?P<pk>[0-9]+)$',
login_required(views.ProjectDetailPage.as_view()), name='project_detail'),
    url(r'^projects/(?P<pk>[0-9]+)/update$',
login_required(views.ProjectUpdateRequiredData.as_view()),
name="project_update_required_data"),
    url(r'^projects/(?P<pk>[0-9]+)/update/general$',
login_required(views.ProjectUpdateGeneralData.as_view()),
name="project_update_general_data"),
    url(r'^projects/(?P<pk>[0-9]+)/delete$',
login_required(views.ProjectDelete.as_view()),
name="project_delete"),

    url(r'^projects/(?P<project_pk>[0-9]+)/events/create$',
login_required(views.EventCreateForProject.as_view()),
name='event_create'),
    url(r'^events/(?P<pk>[0-9]+)$',
login_required(views.EventProjectDetailPage.as_view()),
name='event_project_detail'),
    url(r'^events/(?P<pk>[0-9]+)/update/required',
login_required(views.EventUpdateRequiredData.as_view()),
name='event_update_required_data'),

```

```

        url(r'^events/(?P<pk>[0-9]+)/update/status',
        login_required(views.EventUpdateStatus.as_view()),
        name='event_update_status'),
        url(r'^events/(?P<pk>[0-9]+)/update/result',
        login_required(views.EventUpdateResult.as_view()),
        name='event_update_result'),
        url(r'^events/(?P<pk>[0-9]+)/delete',
        login_required(views.EventDelete.as_view()), name='event_delete'),

        url(r'^events/(?P<event_pk>[0-9]+)/files/create',
        login_required(views.EventFileCreate.as_view()),
        name='event_file_create'),
        url(r'^events/files/(?P<pk>[0-9]+)/update',
        login_required(views.EventFileUpdate.as_view()),
        name='event_file_update'),
        url(r'^events/files/(?P<pk>[0-9]+)/delete',
        login_required(views.EventFileDelete.as_view()),
        name='event_file_delete'),

        url(r'^projects/(?P<project_pk>[0-9]+)/reports/create$',
        login_required(views.ReportCreateForProject.as_view()),
        name='report_create'),
        url(r'^reports/(?P<pk>[0-9]+)/$',
        login_required(views.ReportProjectDetailPage.as_view()),
        name='report_project_detail'),
        url(r'^reports/(?P<pk>[0-9]+)/update/required',
        login_required(views.ReportUpdateRequiredData.as_view()),
        name='report_update_required_data'),
        url(r'^reports/(?P<pk>[0-9]+)/update/resources',
        login_required(views.ReportUpdateResources.as_view()),
        name='report_update_resources'),
        url(r'^reports/(?P<pk>[0-9]+)/update/realization',
        login_required(views.ReportUpdateRealization.as_view()),
        name='report_update_realization'),
        url(r'^reports/(?P<pk>[0-9]+)/update/conclusion',
        login_required(views.ReportUpdateConclusion.as_view()),
        name='report_update_conclusion'),
        url(r'^reports/(?P<pk>[0-9]+)/delete',
        login_required(views.ReportDelete.as_view()), name='report_delete'),
        url(r'^reports/(?P<pk>[0-9]+)/print',
        login_required(views.ReportPrint.as_view()), name='report_print'),

```

```

        url(r'^projects/(?P<project_pk>[0-9]+)/year_plans/create$',
login_required(views.YearPlanCreateForProject.as_view()),
name='year_plan_create'),
        url(r'^year_plans/(?P<pk>[0-9]+)/$',
login_required(views.YearPlanDetailPage.as_view()),
name='year_plan_detail'),
        url(r'^year_plans/(?P<pk>[0-9]+)/update',
login_required(views.YearPlanUpdateRequiredData.as_view()),
name='year_plan_update'),
        url(r'^year_plans/(?P<pk>[0-9]+)/delete',
login_required(views.YearPlanDelete.as_view()), name='year_plan_delete'),
        url(r'^year_plans/(?P<pk>[0-9]+)/print',
login_required(views.YearPlanPrint.as_view()), name='year_plan_print'),
        url(r'^university_department/year_plans/create$',
login_required(views.YearPlanUniversityDepartmentCreate.as_view()),
name='university_department_year_plan_create'),
        url(r'^university_department/year_plans/(?P<pk>[0-9]+)/$',
login_required(views.YearPlanUniversityDepartmentDetailPage.as_view()),
name='university_department_year_plan_detail'),
        url(r'^university_department/year_plans/(?P<pk>[0-9]+)/update',
login_required(views.YearPlanUniversityDepartmentUpdate.as_view()),
name='university_department_year_plan_update'),
        url(r'^university_department/year_plans/(?P<pk>[0-9]+)/delete',
login_required(views.YearPlanUniversityDepartmentDelete.as_view()),
name='university_department_year_plan_delete'),
        url(r'^university_department/year_plans/(?P<pk>[0-9]+)/print',
login_required(views.YearPlanUniversityDepartmentPrint.as_view()),
name='university_department_year_plan_print'),
        url(r'^university_department/reports/create$',
login_required(views.ReportUniversityDepartmentCreate.as_view()),
name='university_department_report_create'),
        url(r'^university_department/reports/(?P<pk>[0-9]+)/$',
login_required(views.ReportUniversityDepartmentDetailPage.as_view()),
name='university_department_report_detail'),
        url(r'^university_department/reports/(?P<pk>[0-9]+)/update$',
login_required(views.ReportUniversityDepartmentUpdate.as_view()),
name='university_department_report_update'),
        url(r'^university_department/reports/(?P<pk>[0-
9]+)/update/conclusion',
login_required(views.ReportUniversityDepartmentUpdateConclusion.as_vie
w()), name='university_department_report_update_conclusion'),
        url(r'^university_department/reports/(?P<pk>[0-9]+)/delete',
login_required(views.ReportUniversityDepartmentDelete.as_view()),
name='university_department_report_delete'),

```

```

        url(r'^university_department/reports/(?P<pk>[0-9]+)/print',
login_required(views.ReportUniversityDepartmentPrint.as_view()),
name='university_department_report_print'),
        url(r'^university_department/reports/(?P<pk>[0-9]+)/stat/print',
login_required(views.StatReportUniversityDepartmentPrint.as_view()),
name='university_department_stat_report_print'),

        url(r'^reports/events', login_required(views.EventsReport.as_view()),
name='events_report_print'),
        url('', views.Home.as_view(), name="home"),
    ]

```