

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА ИНФОРМАЦИОННОГО СОПРОВОЖДЕНИЯ
ДЕЯТЕЛЬНОСТИ СОТРУДНИКОВ
(НА ПРИМЕРЕ ООО «ИНЛАЙТ»)**

Выпускная квалификационная работа
обучающегося по направлению подготовки
09.03.03 Прикладная информатика
очной формы обучения, группы 07001404
Морозова Фёдора

Научный руководитель
ст. преподаватель
Болгова Е.В.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитическая часть	5
1.1 Техничко-экономическая характеристика предметной области.....	5
1.1.1 Характеристика предприятия, включая организационную структуру управления предприятием.....	5
1.1.2 Краткая характеристика отдела разработки.....	7
1.2 Постановка задачи	8
1.2.1 Цель и назначение автоматизированного варианта решения задачи.....	8
1.2.2 Общая характеристика организации решения задачи на ЭВМ.....	9
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	11
1.4 Анализ существующих разработок.....	16
2 Проектная часть.....	20
2.1 Обоснование проектных решений	20
2.1.1 Обоснование проектных решений по техническому обеспечению	20
2.1.2 Обоснование проектных решений по информационному обеспечению	22
2.1.3 Обоснование проектных решений по программному обеспечению.....	24
2.1.4 Обоснование проектных решений по технологическому обеспечению.....	25
2.1.5 Обоснование выбора программных средств.....	26
2.2 Характеристика базы данных.....	28
2.2.1 Характеристика инфологической модели БД.....	28
2.2.2 Характеристика даталогической модели БД.....	31
3 Программная реализация	35
3.1 Описание дерева функций и сценария диалога	35
3.2 Организация технологии сбора, передачи, обработки и выдачи информации	37
3.3 Описание контрольного примера реализации проекта	38
3.4 Обоснование эффективности экономической разработки	50
3.4.1 Выбор и обоснование показателей эффективности и методики их расчёта.....	50
3.4.2 Расчёт показателей эффективности.....	50
ЗАКЛЮЧЕНИЕ	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	56
ПРИЛОЖЕНИЯ.....	59

ВВЕДЕНИЕ

По данным аналитического портала TAdviser объём российского рынка услуг в сфере информационных технологий (ИТ-услуг) составляет свыше 250 млрд руб. В 2017 году больше 90 млрд руб. составила сфера оказания услуг по разработке программного обеспечения специализированными организациями (ИТ-аутсорсинг), что составляет порядка 1/3 от всего объёма данного рынка [25]. Среди многочисленного списка организаций, оказывающих все виды услуги по разработке программного обеспечения, находится и ООО «ИНЛАЙТ». Сотрудниками данной организации являются высококвалифицированные инженеры-программисты, менеджеры, дизайнеры и проектировщики интерфейсов.

На данный момент все процессы по проектному управлению в ООО «ИНЛАЙТ» происходят в ручном режиме. Отсутствует формализованная отчётность сотрудников о выполненных ими задачах, а также не проводится анализ деятельности сотрудников на основе данных отчётов. На обработку информации генеральным директором ООО «ИНЛАЙТ» затрачиваются существенные временные ресурсы.

Объектом исследования в данной работе является ООО «ИНЛАЙТ».

Предмет исследования – деятельность сотрудников отдела разработки ООО «ИНЛАЙТ».

Целью данной выпускной квалификационной работы является разработка информационного сопровождения деятельности сотрудников отдела разработки ООО «ИНЛАЙТ».

Задачи, которые необходимо решить в рамках ВКР для достижения поставленной цели:

- изучить предметную область;
- выполнить сбор и анализ данных;
- привести обоснование выбора основных проектных решений;
- выполнить проектирование системы;

- разработать информационное сопровождение деятельности сотрудников;

- провести тестирование разработанной системы;

- выполнить обоснование экономической эффективности проекта.

Техническое задание на разработку информационного сопровождения деятельности сотрудников ООО «ИНЛАЙТ» представлено в приложении А.

Во введении приведена актуальность выбранной темы исследования, определены цель, задачи, объект и предмет исследования.

В первом разделе представлена технико-экономическая характеристика предметной области, содержащая краткую характеристику ООО «ИНЛАЙТ» и видов его деятельности. Приведена модель «Как есть» бизнес-процессов компании в нотации IDEF0. Обоснована необходимость решения задачи с использованием информационных технологий. Приведена модель «Как должно быть» бизнес-процессов компании в нотации IDEF0. Проведён подробный анализ существующих разработок и обоснование выбора технологии проектирования информационного сопровождения.

Второй раздел посвящён обоснованию и выбору проектных решений по техническому, информационному, программному и технологическому обеспечению. Приводится характеристика базы данных.

В третьем разделе описан разработанный программный продукт, дерево функций, сценарий диалога, процесс организации технологии сбора, передачи, обработки и выдачи информации. Представлен контрольный пример реализации проекта, а также приведён расчёт экономических затрат на разработку и эксплуатацию системы.

В заключении представлены выводы по проведённому исследованию.

Работа состоит из 54 страниц, которые содержат 28 рисунков, 8 таблиц, 2 формулы и 3 приложения.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия, включая организационную структуру управления предприятием

Организация ООО «ИНЛАЙТ» зарегистрирована 18 октября 2006 года по адресу 115612, г Москва, Южный административный округ, район Братеево, ул Борисовские Пруды, д 24/2. Основным государственным регистрационным номером организации (ОГРН) является: 1067758221740, идентификационный номер налогоплательщика (ИНН): 7724594030, код причины постановки на учет (КПП): 772401001. Генеральным директором организации является Кутасин Максим Леонидович.

Основным видом деятельности организации согласно Общероссийскому классификатору видов экономической деятельности (ОКВЭД) является: 62.01 «Разработка компьютерного программного обеспечения».

ООО «ИНЛАЙТ» тесно сотрудничает с коммерческими, государственными и бюджетными организациями на территории города Москвы и Московской области. Сотрудничество заключается в разработке B2B и B2C программных продуктов.

Целью деятельности организации ООО «ИНЛАЙТ», как и любого другого коммерческого предприятия, является максимизация прибыли от выполнения основной деятельности.

На сегодняшний день в организации можно выделить следующие структурные единицы:

- генеральный директор;
- отдел продаж;
- отдел разработки;

- отдел сопровождения.

На рисунке 1.1 представлена организационная структура управления ООО «ИНЛАЙТ».



Рисунок 1.1 – Организационная структура компании

Каждый отдел организации состоит из руководителя соответствующего отдела и сотрудников, которые подчиняются ему. В свою очередь все руководители отделов подчинены генеральному директору.

Отдел продаж – группа менеджеров, которые являются связующим звеном между клиентом и ООО «ИНЛАЙТ». В обязанности любого менеджера по работе с клиентами входит: планирование, организация и контроль работы отдела.

Отдел разработки – группа сотрудников, которые непосредственно реализуют задачи, поставленные менеджерами отдела продаж.

Отдел сопровождения – группа сотрудников, обеспечивающая техническую поддержку на всём протяжении эксплуатации программных продуктов клиентами организации.

Выполнение данной выпускной квалификационной работы предусмотрено в рамках отдела разработки, так как конкретно этот отдел занимается реализацией технических заданий, поступающих из отдела продаж.

1.1.2 Краткая характеристика отдела разработки

Для эффективного осуществления основного вида деятельности организации был сформирован отдел разработки.

Функциями отдела разработки являются:

- осуществление разработки технических проектов программного обеспечения, построение архитектуры программных систем, определение технических и программных требований к программному обеспечению;
- разработка и развитие, при необходимости, визуального интерфейса программного обеспечения.
- анализ поступающей информации о сбоях в работе программного обеспечения, связанных с ошибками в программном обеспечении и принятие меры к их оперативному устранению;
- оказание консультативной помощи отделу сопровождения по вопросам функционирования используемых технологий;
- осуществление разработки нового программного обеспечения, согласно получаемым из отдела продаж техническим заданиям;
- осуществление предварительного тестирования программного обеспечения на правильность реализации алгоритмов и соответствие техническому заданию;
- осуществление передачи отделу сопровождения новых версий программного обеспечения для осуществления последующего тестирования и сдачи в эксплуатацию;
- анализ полученных от отдела сопровождения заключения о результатах тестирования;
- осуществление разработки документации по установке программного обеспечения и руководства администратора.

1.2 Постановка задачи

1.2.1 Цель и назначение автоматизированного варианта решения задачи

По завершению процесса ознакомления с деятельностью ООО «ИНЛАЙТ» была сформулирована задача автоматизации взаимодействия сотрудников отдела разработки с менеджерами отдела продаж для повышения эффективности работы ООО «ИНЛАЙТ», контроля сроков выполнения технических заданий, сокращения сроков формирования отчётов работы программистов-инженеров за месяц, формирования рекомендаций по управлению сотрудниками.

Основные функции, которые должна выполнять автоматизированная система управления проектами:

- добавление данных о проектах и отдельных задачах;
- редактирование соответствующей информации;
- оперативное получение актуальных данных с использованием функции поиска;
- ведение истории обсуждения задач;
- контроль своевременного выполнения задач;
- формирование и печать отчета эффективности деятельности сотрудников отдела за месяц.

Целью автоматизации взаимодействия сотрудников отдела разработки с менеджерами отдела продаж является увеличение производительности компании.

С помощью автоматизации взаимодействия можно будет вести историю реализации технического задания клиента, историю изменения статусов задач, получать аналитическую информацию за необходимый период времени. Все это позволит принимать верные управленческие решения, снизить издержки,

уменьшить временные затраты, тем самым увеличить прибыль предприятия.

1.2.2 Общая характеристика организации решения задачи на ЭВМ

Разрабатываемое информационное сопровождение должно обеспечивать учёт и контроль выполнения задач сотрудниками ООО «ИНЛАЙТ».

Для этого создаваемое информационное сопровождение должно:

- обеспечивать ввод данных, связанных с задачами клиентов компании;
- осуществлять обработку введённых данных;
- создавать отчёты и сопутствующие документы;
- иметь информативную систему помощи по использованию информационного сопровождения.

Информационное сопровождение должно быть доступно с различными правами доступа генеральному директору организации, начальнику отдела и сотрудникам отдела разработки. Заданные характеристики функционирования должны обеспечиваться при условиях, которые определяются конкретным носителем данных, на котором они хранятся.

В качестве организации архитектуры аппаратной платформы решения задачи была выбрана архитектура клиент-сервер.

Технология клиент-сервер – способ взаимодействия программных компонентов, образующих единую систему. Как видно из самого названия, существует некий клиентский процесс, требующий определённых ресурсов, а также серверный процесс, который эти ресурсы предоставляет. Совсем необязательно, чтобы они находились на одном компьютере. Обычно принято размещать сервер на одном узле локальной или глобальной сети, а клиентов – на других узлах.

В контексте базы данных клиент управляет пользовательским интерфейсом и логикой приложения, действуя как рабочая станция, на которой

выполняются приложения баз данных. Клиент принимает от пользователя запрос, проверяет синтаксис и генерирует запрос к базе данных на языке SQL или другом языке базы данных, соответствующем логике приложения. Затем передаёт сообщение серверу, ожидает поступления ответа и форматирует полученные данные для представления их пользователю. Сервер принимает и обрабатывает запросы к базе данных, после чего отправляет полученные результаты обратно клиенту. Такая обработка включает проверку полномочий клиента, обеспечение требований целостности, а также выполнение запроса и обновление данных. Помимо этого, поддерживается управление параллельностью и восстановлением.

Клиент-серверная архитектура обладает следующими преимуществами:

- снижает стоимость аппаратного обеспечения (относительно производительный компьютер с объёмным устройством для хранения информации необходим лишь серверу для управления и хранения базой данных);
- обеспечивает более широкий доступ к существующим базам данных;
- сокращает расходы на коммуникацию. Только часть операций выполняется приложением на клиентских компьютерах, которые отправляют через сеть короткие запросы к базе данных. Эти решения позволяют существенно снизить объем пересылаемых данных через сеть;
- повышает общую производительность системы: поскольку клиенты и сервер находятся на разных компьютерах, их процессоры способны выполнять приложения параллельно;
- повышает уровень непротиворечивости данных. Сервер может самостоятельно управлять проверкой целостности данных, поскольку лишь он определяет и проверяет все ограничения. При этом каждому приложению не придётся выполнять собственную проверку.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

Для детального анализа деятельности организации необходимо использовать мощный инструмент для моделирования, а также для реорганизации сложных бизнес-процессов и документирования, в данной выпускной квалификационной работе будет использоваться AllFusion Process Modeler 7. [23]

AllFusion Process Modeler 7 – это совокупность средств и инструментов моделирования потоков работ (IDEF3), функций (IDEF0), данных (DFD). Данный программный продукт позволяет явно документировать все необходимые аспекты любых бизнес-процессов: действия, которые необходимо предпринять, способы их осуществления и контроля, требующиеся для этого ресурсы, а также визуализировать получаемые от этих действий результаты. [16]

Эффективность ИТ-решений в бизнесе отчасти достигается путём предоставления возможности проектировщикам моделей и аналитикам соотносить коммерческие инициативы и задачи с бизнес-интересами, процессами и требованиями информационных архитектур. С этим продуктивно справляется AllFusion Process Modeler 7. [2]

Модель, которая представляет систему в форме набора действий с иерархией, является Контекстной диаграммой. Следует отметить, что каждое действие в иерархии непосредственно преобразует некоторый объект или набор объектов.

На рисунке 1.2 представлена контекстная диаграмма ООО «ИНЛАЙТ» «Как есть».

Из данной диаграммы очевидно, что все действия (заявки клиентов, денежные средства, договора, программные продукты, сопроводительная документация) осуществляются исключительно вручную, что порождает высокие временные затраты на обработку данных.

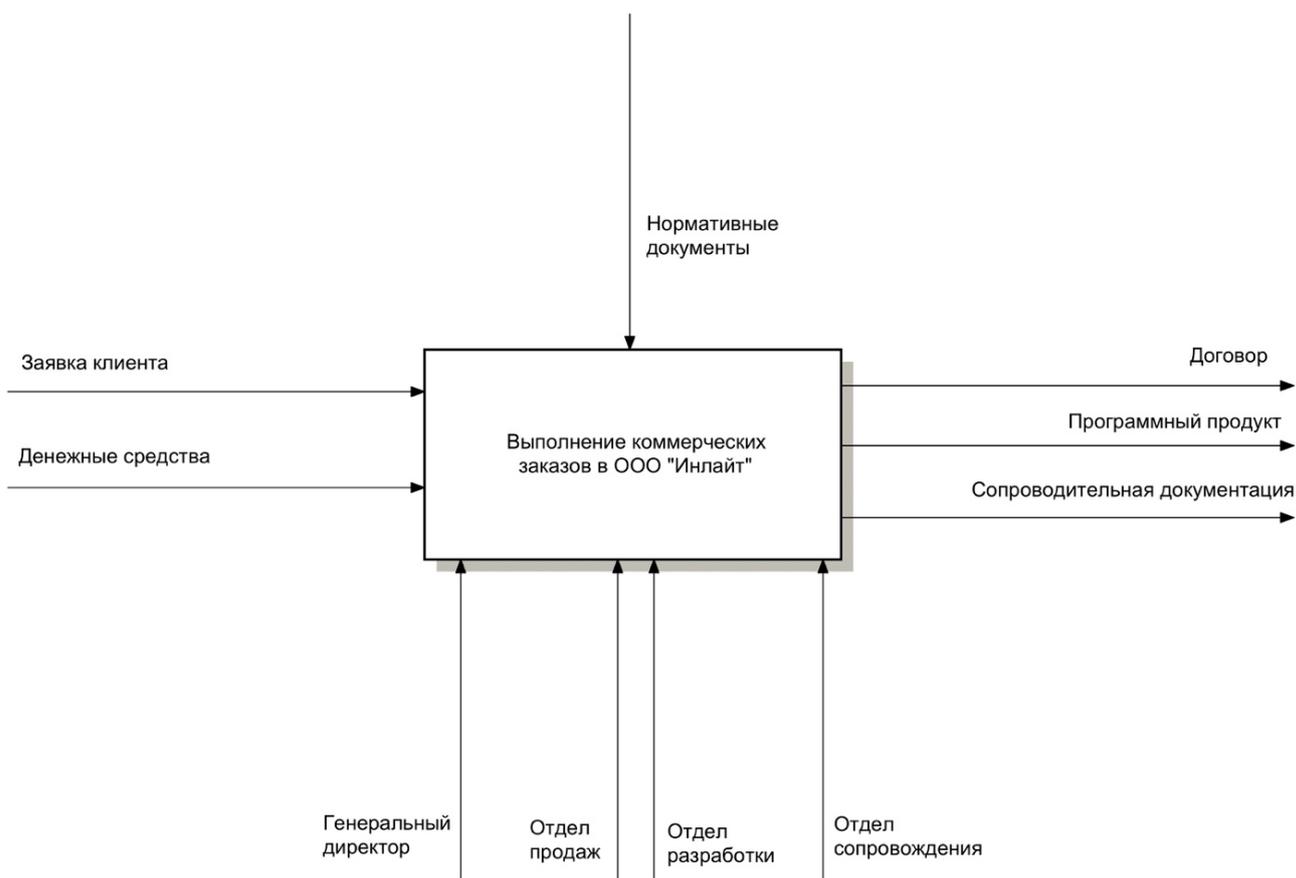


Рисунок 1.2 – Контекстная диаграмма ООО «ИНЛАЙТ» «Как есть»

При создании модели, для начала, необходимо изобразить действие контекста – самый высокий уровень. Непосредственное описание системы сформировано в наименовании действия и представлено одним глаголом в сочетании с существительным в обобщающем виде, которое поясняет общий взгляд на цель деятельности системы. [27]

Исходя из декомпозиции контекстной диаграммы (см. рисунок 1.3) можно выделить четыре основных процесса в деятельности организации при выполнении коммерческих заказов:

- анализ и согласование заявки клиента;
- оформление договора;
- выполнение заказа;
- сопровождение проекта.

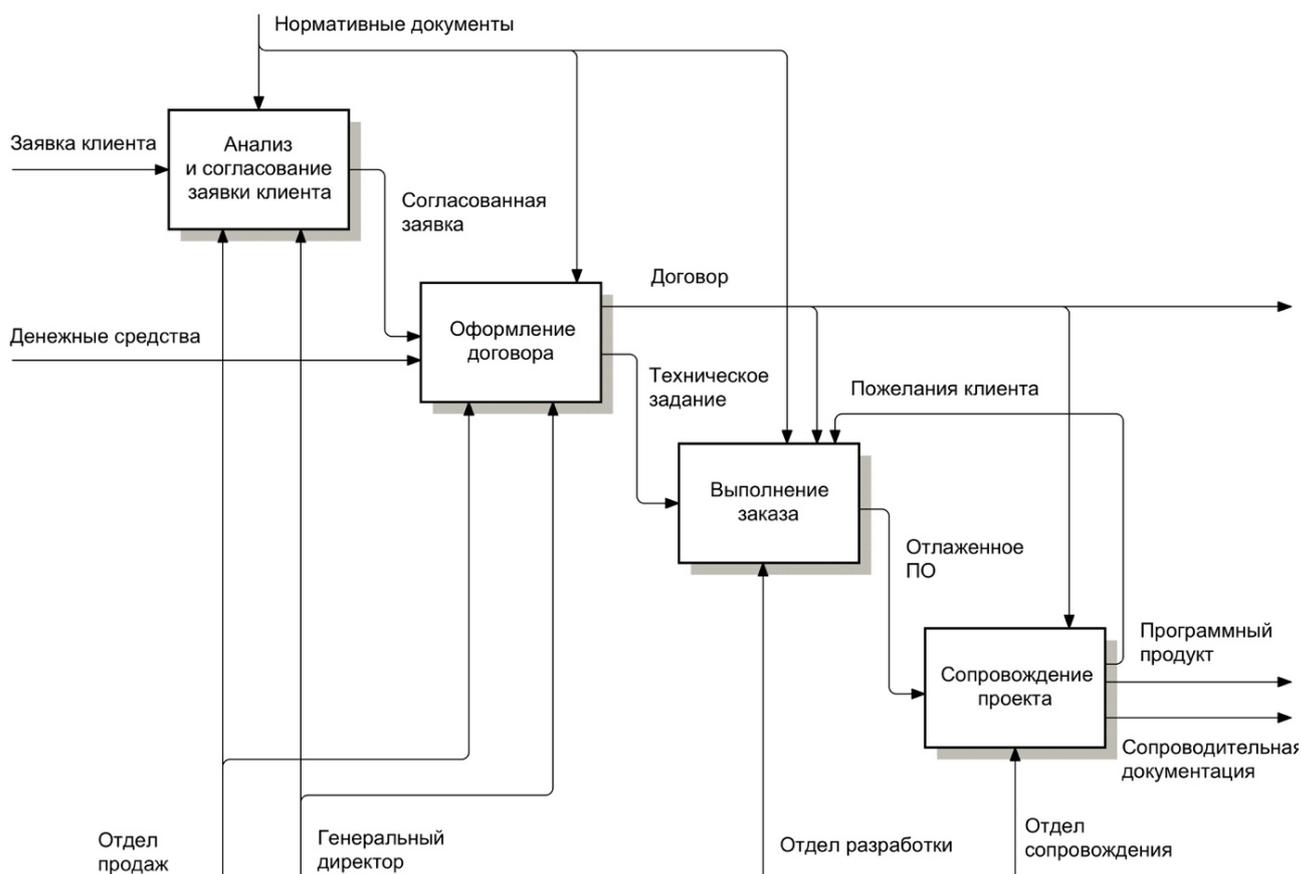


Рисунок 1.3 – Декомпозиция процесса «Выполнение коммерческих заказов»

Исследуемый в данной выпускной квалификационной работе отдел разработки непосредственно участвует в деятельности организации на этапе «Выполнение заказа». Рассмотрим диаграмму декомпозиции данного процесса на рисунке 1.4. Процесс состоит из трёх основных составляющих:

- формализации задач;
- разработки (выполнение задач);
- тестирования выполненных задач.

Процессы «Тестирование» и «Разработка» связаны обратной связью, так как отрицательные результаты тестирования разработанного ПО вынуждают сотрудников отдела разработки устранить неисправности при доработке каждой задачи.

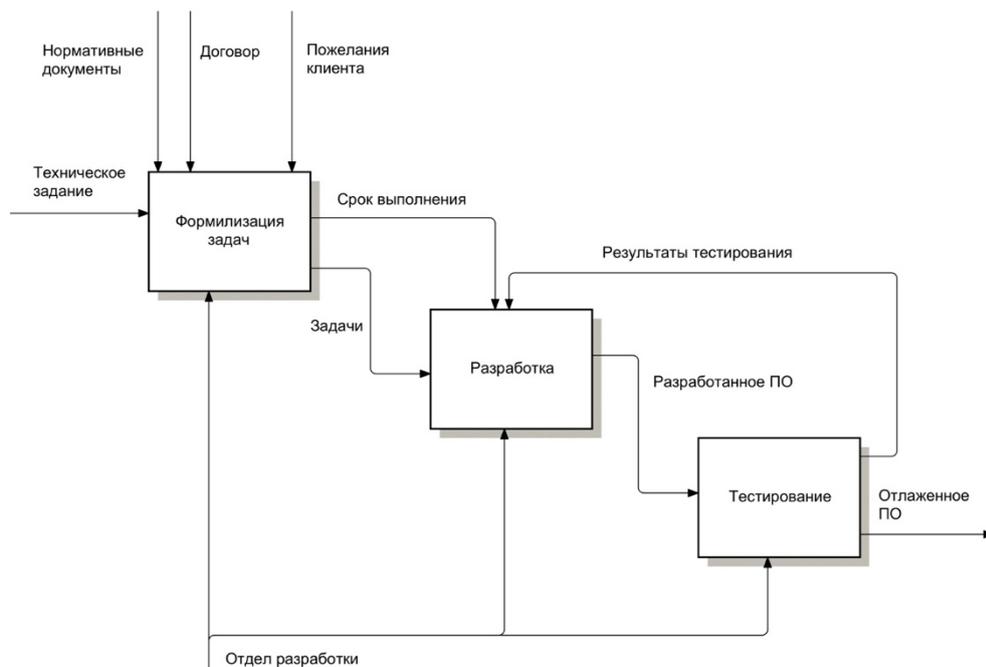


Рисунок 1.4 – Декомпозиция процесса «Выполнение заказа»

Исходя из диаграммы на рисунке 1.4. можно сделать вывод о том, что в организации не существует какой-либо системы учёта деятельности сотрудников отдела разработки, что отрицательно влияет на анализ эффективности каждого отдельного сотрудника и отдела разработки в целом.

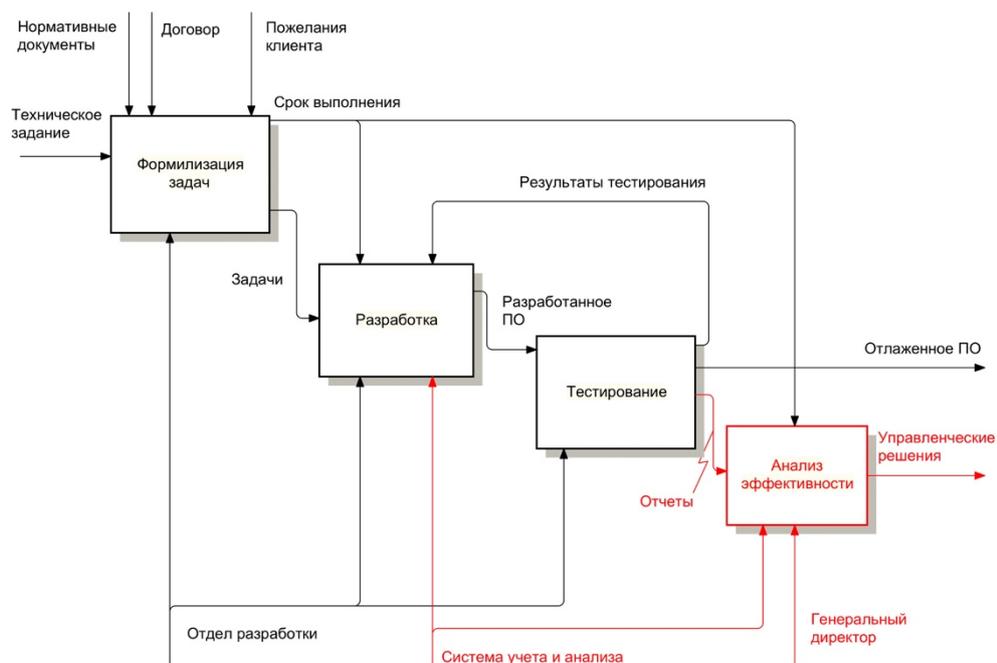


Рисунок 1.5 – Декомпозиция процесса «Выполнение заказа» «Как должно быть»

Для решения данной задачи в бизнес-процессы организации был внесён ряд изменений и сформирована диаграмма «Как должно быть». Среди изменений можно выделить добавление новых механизмов «Информационное сопровождение деятельности» и «Генеральный директор» в процесс «Выполнение заказа» (см. рисунок 1.5). Информационное сопровождение деятельности сотрудников необходимо для автоматического учёта их действий и последующего анализа с формированием статистических отчётов.

Рассмотрим новый процесс «Анализ эффективности» на рисунке 1.6.

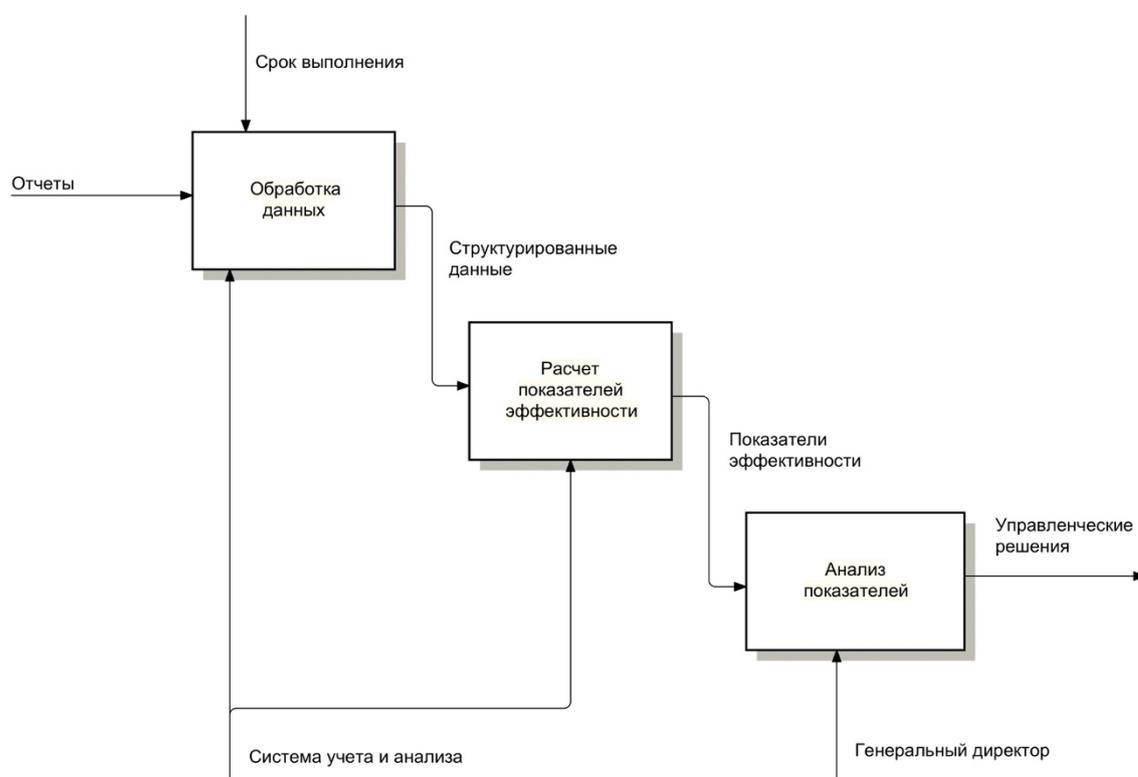


Рисунок 1.6 – Декомпозиция процесса «Анализ эффективности»

Данный процесс, состоит из трёх основных действий: «Обработка данных», «Расчет показателей эффективности», «Анализ показателей». В ходе выполнения бизнес-процесса информационное сопровождение формирует структурированные статистические данные, после анализа которых генеральный директор принимает соответствующие управленческие решения.

1.4 Анализ существующих разработок

Для выбора оптимального варианта разработки собственного продукта или использования готового, необходимо провести обзор самых популярных информационных систем, которые помогают вести проектный учёт деятельности сотрудников. В рамках выпускной квалификационной работы был проведен анализ следующих программ: «Битрикс24», «Мегаплан», «ПланФикс», «JIRA», а также выявлены преимущества и недостатки данных систем.

«Битрикс24» — это сервис, объединяющий в себе огромный набор функциональных возможностей. Пользователи могут опробовать в нём систему управления задачами, проектами, документами, социальную сеть, чат, видеозвонки, хранилище документов, календарь, почту, управление взаимоотношениями с клиентами. Помимо этого, система включает инструменты телефонии, HR, сервис учёта рабочего времени и многое другое. [28]

Преимущества:

- возможность планирования графиков общения с ключевыми заказчиками, в удобное им время и по комфортным для них каналам;
- опрос клиентов (можно собирать сведения о личных пристрастиях и хобби);
- возможность назначать скидки;
- бесплатная рассылка кейсов, методичек, советов;
- фиксация активности и получение обратной связи в системе управления заявками.

Недостатки:

- регламенты обработки заявок, оформления документации, подготовки отчетности реализуются только при помощи шаблонов «Битрикс24»;
- в бесплатной облачной версии «Проект» число пользователей CRM не может превышать 12 человек;
- формирование коммерческих предложений, отслеживание процесса

выполнения заказов и заключения сделок, ведение каталога товаров и услуг, выставление счетов на оплату возможны только в версии на коммерческой основе.

«Мегаплан» — это система управления бизнесом. Неважно, большой бизнес или маленький и в чём его суть. Важно, что бизнес работает, и им надо управлять. Уникальность «Мегаплан» в том, что это CRM система и управление проектами в одной программе. «Мегаплан» помогает раздавать поручения и контролировать сотрудников. Получать отчёты по выполненным задачам можно вне офиса. Сервис сохраняет историю отношений с клиентами: записи звонков, переписку, счета. [29]

Преимущества:

- у сотрудника единый список задач;
- сотрудник отчитывается о выполнении задачи в комментариях;
- менеджерам доступна актуальная информация и статусы по клиентам;
- напомнит о запланированных звонках и пригласит на встречу;
- менеджеры могут выставить счёт, и он синхронизируется с данными в 1С.

Недостатки:

- малофункциональное мобильное приложение;
- не пробрасываются данные при создании связанных сделок, какие неожиданные моменты в сценариях — в отдельные поля нельзя вставлять переменные;
- артефакты уже несуществующих функции — системы лояльности, работы с календарём;
- систематические перебои при регулярном обновлении продукта.

«ПланФикс» — платформа для создания систем управления. Она позволяет настроить и контролировать различные бизнес-процессы компании без программирования. [22]

Преимущества:

- настраиваемая структура компании и возможность задавать сотрудникам роли, предоставляя им доступ только к определенным проектам, задачам и данным;
- возможность создавать уникальное рабочее пространство для каждой должности или сотрудника компании с логотипом, цветовой схемой и разделами меню;
- возможность выстраивать структуру проектов любой вложенности, ставить задачи и следить за их выполнением с помощью удобных напоминаний;
- возможность добавлять чек-листы, разбивая задачу на несколько шагов;
- шаблоны позволяют создавать сделки, запросы в техподдержку.
- возможность обрабатывать электронные письма, звонки, заказы с сайта, комментарии из социальных сетей.

Недостатки:

- отсутствие мобильной версии или мобильного приложения;
- ограниченный во времени пробный период продукта;
- систематические сбои в работе визуального интерфейса;
- отсутствие возможности скрыть неиспользуемые настройки и параметры.

«JIRA» — коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами. Это инструмент для всех сотрудников в команде и руководителей проектов. «JIRA» помогает группе разработчиков делиться информацией и быстро вовлекать новых сотрудников в проекты и задачи, фиксировать и отслеживать ошибки пользователей, обеспечивать соблюдение выполнения работы точно в срок. [13]

Преимущества:

- отслеживание и работа с проектами — просмотр процесса работы разработчиков над проектом, какие возникают ошибки в его использовании клиентами, быстрое их устранение;

- решение вопросов — документы, почта, общение — всё в одном месте, решение каждого вопроса происходит быстро и легко;
- планирование рабочего процесса — построение рабочего процесса эффективнее и удобнее;
- отслеживание эффективности — постановка задач, приоритетов, помощь команде в выделении важного в работе и отслеживание выполнения задач;
- совместная работа — обмен информацией по проекту, совместное решение вопросов и обращение за помощью к коллегам;
- видимость-онлайн — новости, почта, просмотр работы в режиме реального времени в одном месте;
- интеграция с различными разработками и дополнениями Altassian и другими разработчиками.

Недостатки:

- сложность в тонкой настройке под нужды конкретной организации;
- высокая стоимость за каждого пользователя (сотрудника) внутри одной организации;
- не соответствует федеральному закону № 152-ФЗ;
- не входит в Единый реестр российских программ.

В первом разделе проведено описание технико-экономической характеристики ООО «ИНЛАЙТ», а также её деятельности и текущего состояния бизнес-процессов, выявлена необходимость в разработке информационного сопровождения деятельности сотрудников разработки. Генеральным директором было принято решение о разработке собственного информационного сопровождения.

2 Проектная часть

2.1 Обоснование проектных решений

2.1.1 Обоснование проектных решений по техническому обеспечению

Техническим обеспечением называется комплекс технических средств, которые предназначены для функционирования информационного сопровождения, а также необходимая документация для этих средств и технологических процессов. [5]

В комплекс технических средств входит:

- электронно-вычислительные машины (ЭВМ) различных модификаций;
- устройства ввода-вывода и обработки информации;
- устройства передачи данных по различным линиям связи;
- расходные и эксплуатационные материалы и т.д.

Для проектируемого информационного сопровождения требуется аппаратное обеспечение, среди которого можно выделить:

- персональные компьютеры для рабочих мест сотрудников;
- локальная вычислительная сеть;
- веб-сервер.

В настоящее время сотрудники отдела разработки используют компьютеры, конфигурационные характеристики которых соответствует всем требованиям, предъявляемым к информационному сопровождению (см. таблицу 2.1). Все компьютеры соединены в общую локальную вычислительную сеть с доступом в глобальный Интернет.

Таблица 2.1 – Минимальные требования к аппаратному обеспечению

Характеристика	Значение, не менее
Частота центрального процессора	1.5 ГГц
Объем оперативной памяти	4 Гб
Разрешение экрана	1366x768

Сервер баз данных обслуживает базу данных и отвечает за сохранность данных и их целостность, и, кроме того, при сеансе доступа клиента к информации обеспечивает операции ввода-вывода. Специфика сервера БД заключается в том, что данные, в большинстве случаев, обрабатываются с использованием механизма транзакций, т.е. система запрашивает данные небольшого объема, производит над ними операцию, и после этого сохраняет [18]. Эта особенность накладывает определенные требования на аппаратное обеспечение сервера баз данных, а именно:

- значительный объем ОЗУ для операций кэширования наиболее часто используемых участков базы данных;
- скоростная и высокопроизводительная дисковая система, способная обрабатывать очень большое число мелких запросов за секунду (IOPS - inputs/outputs per second);
- большая вычислительная мощность для выполнения операций по обработке информации.

Основное требование к системной плате сервера – она должна уметь вести в режиме реального времени контроль и мониторинг всех компонентов сервера, и в случае отказа любой из составляющих, не прерывая работы, выполнить автоматическое резервирование компонента и оповещение системного администратора о возникшей ошибке. Серверу базы данных необходим значительный объем ОЗУ. Работа с ОЗУ на несколько порядков быстрее, чем обмен данными с жесткими дисками, следовательно, чем большим объемом ОЗУ будет оборудован сервер базы данных, тем лучше, однако, есть одно условие – операционная система и само приложение сервера базы данных должны быть способны адресовать такое количество памяти и работать с ним.

2.1.2 Обоснование проектных решений по информационному обеспечению

Информационным обеспечением (ИО) называют совокупность проектных решений, которые определяют объём, тип и вид информации, циркулирующей в системе [29]. Информационное обеспечение включает в себя:

- набор справочных данных и спецификаций информации;
- специально автоматизировано для выполнения в автономном режиме обслуживания;
- большие объемы информации, находящиеся на различных носителях;
- системы документации;
- сотрудников, которые осуществляют контроль над системой и выполняют изменения, хранения и распространения информации.

Далее следует отметить требования к структуре и методам, которые определяют группирование данных в рамках информационных систем:

- отображаемые данные должны быть как можно более информативными по отношению к изучаемой предметной области, что означает в набор данных должна входить только необходимая информация;
- всей информации в системе следует храниться в реляционной базе данных, модель которой должна отображать несколько сущностей, объединённых уникальными полями, также таблицы должны быть приведены к нормальным формам;
- связь между различными модулями информационной системы должны находиться только внутри её.

Проектируемое информационное сопровождение должно быть разработана с использованием системы управления базами данных MySQL – это основное требование по использованию СУБД.

Перечислим основные требования к характеристикам языка описания базы данных:

- простота изучения;
- наглядность;
- встроенные в систему управления базами данных средства предупреждения и защиты от несанкционированного доступа;
- обеспечение необходимой степени независимости баз данных.

Перечислим основные требования к структуре передачи, сбора, представления и обработки данных в системе:

- ввод информации в проектируемую систему производится посредством ручного ввода с помощью экранных форм. Исходя из этого, должны быть спроектированы экранные формы для ввода информации, которые обеспечивают минимально возможную трудоёмкость ввода. Требуется обеспечить связь между вводимыми данными и нормативно- справочной информацией; [26]

- разрабатываемая система должна включать в себя необходимые средства для осуществления отбора информации в таблицах базы данных. Вывод информации в системе должен производиться с помощью набора экранных форм, таблиц, отчетов.

Требования к хранению, контролю, восстановлению и обновлению данных: для обеспечения корректировки и контроля данных вся необходимая информация из базы данных должна быть доступной для ручного изменения. [30]

Принципы организации информационного обеспечения:

- актуальность данных;
- минимально возможный объем хранения;
- достоверность и непротиворечивость данных;
- удобство и простота эксплуатации;
- гибкость, т.е. возможность расширения базы данных и адаптации ее к изменениям предметной области.

2.1.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение (ПО) состоит из совокупности программ, которые реализуют задачи и функции системы, а также обеспечивают стабильную работу комплексов различных технических средств [18].

Программное обеспечение включает в свой состав:

- специальные и общесистемные программы;
- инструктивно-методические пособия и материалы по использованию средств программного обеспечения;
- обслуживающий персонал, ответственный за разработку и сопровождение программного обеспечения в течение всего его жизненного цикла.

В общесистемное программное обеспечение входят программы, используемые большим количеством пользователей и предназначенные для обеспечения вычислительного процесса и решений наиболее часто встречающихся задач обработки информации. Общесистемное программное обеспечение позволяет расширить функциональные возможности компьютера, автоматизировать планирование очередности вычислительных задач, осуществлять контроль и управление обработкой данных, и автоматизировать работу программистов.

Системное программное обеспечение (СПО) – это совокупность программ, проектируемых при создании информационных систем для конкретного функционального назначения. Системное программное обеспечение включает пакеты прикладных программ (ППП) которые осуществляют организацию и обработку данных при решении конкретных прикладных задач. [1]

Для успешного функционирования информационного сопровождения на рабочем месте сотрудника отдела требуется наличие следующего программного обеспечения:

- операционная система Windows 7 или выше;
- веб-браузер Internet Explorer или аналогичный ему по функционалу;
- программа драйвер для работы с принтером.

2.1.4 Обоснование проектных решений по технологическому обеспечению

В зависимости от того насколько правильно и рационально будет разработан технологический процесс, настолько точно будет наблюдаться уменьшение трудовых и стоимостных затрат.

Технологический процесс обычно состоит из нескольких взаимосвязанных этапов. Конечной целью первого этапа будет сбор, регистрация и передача собранных данных для последующей обработки. Результатом первого этапа является составление документа. Конечной целью второго этапа является перенос собранных данных на машинные носители и первичное формирование информационной базы системы. Третий этап обычно включает в себя операции накопления, сортировки данных, их обработки и корректировки.

В ходе разработки при выборе варианта используемого технологического процесса необходимо учитывать следующие требования:

- решение задач в назначенные сроки;
- обеспечение актуальности, достоверности и целостности обрабатываемой информации;
- минимизация трудовых и стоимостных затрат в процессе обработки данных;
- возможности анализа и обработки данных на ЭВМ;
- возможность использования различных режимов при решении задач.

Исходя из указанных выше требований, наиболее целесообразным будет проектирование системы, которое децентрализует процесс решения поставленной в выпускной квалификационной работе задачи и, как следствие,

повысит производительность труда программиста и разработчика.

В процессах обработки данных необходимо использовать массивы, содержащие справочную информацию. Такой подход даст выигрыш в скорости выбора, сортировки и т.д. При этом требуется предусмотреть возможность предварительного просмотра полученных результатов запроса перед передачей и оформлением выходной информации. Практический опыт разработчиков показывает, что разработка сайтов с применением принципов построения модели, основанной на диалоге, обеспечивает наиболее гибкую связь системы с пользователем.

Диалоговый режим обладает рядом преимуществ: он удобен при работе с базой данных; обеспечивает защиту при несанкционированном доступе; обеспечивает непосредственное участие пользователя в процессе решения задачи; обеспечивает управляемость процессом; быстрый доступ, поиск и выдача информации в любой момент времени, выбор различных режимов работы; осуществление быстрого перехода от одной операции к другой. [3]

Выделяют несколько видов диалога: меню, управляющие команды, диалог на ограниченном естественном языке и запросы.

В данной работе наиболее оптимальным будет использование меню с одноуровневой структурой.

2.1.5 Обоснование выбора программных средств

Учитывая поставленные выше задачи и техническое задание, в качестве основного инструмента разработки информационного сопровождения использовался микро-фреймворк Flight на основе PHP.

Flight — это высокоэффективный, основанный на компонентной структуре PHP-фреймворк для разработки крупных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Фреймворк является

бесплатным программным обеспечением и распространяется по свободной лицензии MIT (разработана Массачусетским технологическим институтом). [9]

Основным достоинством Flight можно считать поддержку им объектно-ориентированного программирования (ООП). Следует отметить также высокую скорость работы и развитую техподдержку. Это и интегрированное использование Asynchronous Javascript and XML (AJAX), и встроенная поддержка интернационализации приложений, а также простой инструментарий для работы с базами данных. [24]

В Flight присутствует функционал, значительно повышающий разработку. При этом в нём нет ничего лишнего. Разумеется, требуется некоторое время, чтобы понять, как разрабатывать программное обеспечение на данном фреймворке.

Flight подходит для разработки веб-приложений с большим потоком трафика, поскольку обладает легковесностью и продвинутыми средствами кэширования. Фреймворк способен работать с разными системами управления базами данных, в частности с MySQL, SQLite, PostgreSQL, Oracle.

Кроме того, требуются следующие инструментальные программные средства разработки: язык PHP 7 и дополнительно его расширения, локальный сервер Apache, система управления базами данных MySQL.

MySQL – это сравнительно быстрая и небольшая система управления реляционной базой данных, которая основана на традициях Hughes Technologies Mini SQL (mSQL). Рассматриваемая система управления базами данных имеет средства для обеспечения целостности данных, контроля доступа к данным, а также управления контролируемым параллелизмом и, что немаловажно, восстановлением базы данных. [9]

PHP – является серверным скриптовым языком разработки сценариев наиболее общего назначения, который интенсивно используется для создания веб-приложений. На сегодняшний день он выделяется в качестве лидера среди языков программирования. Система управления базами данных MySQL и PHP 7 требуются для обеспечения функционирования информационного

сопровождения на отдельном локальном компьютере, а кроме того, для создания собственных таблиц базы данных и расширений для решения прикладных задач. Применение локального сервера Apache необходимо для реализации возможности локального тестирования информационного сопровождения и его предварительного просмотра перед размещением на веб-сервер. [14]

Указанные выше инструментальные средства разработки распространяются бесплатно и имеют свободную лицензию, поэтому их применение позволяет избежать лишних затрат на проектирование.

2.2 Характеристика базы данных

2.2.1 Характеристика инфологической модели БД

База данных предназначена для отображения реального мира, то есть его какой-либо части. В зависимости от цели, с которой создаётся информационная система вообще и база данных в частности, зависит полнота её описания. [30]

В базе данных отображается предметная область и для адекватного её отображения проектировщику базы данных необходимо иметь представление обо всех нюансах предметной области и умение адекватно их отобразить. Следовательно, при создании базы данных какой-либо предметной области необходима информация и представление о её функционировании и проектировщик базы данных на первом этапе должен выяснить данные вопросы. Предметная область должна иметь полное предварительное описание. Для описания предметной области можно использовать естественный язык, однако такое описание будет иметь определенные недостатки, такие как громоздкость описания и, во многих случаях, неоднозначность трактовки. Таким образом, при описании предметной области рекомендовано использовать формальные языковые средства.

Инфологическая модель (ИЛМ) – это описание предметной области,

которое использует специальные языковые средства для ее представления. При этом языковые средства являются независимыми от тех программных средств, которые будут использоваться далее.

При построении инфологической модели необходимо учитывать, что она должна быть пригодна для использования не только для системы управления базами данных, но и для работы с другими программными средствами при реализации информационной системы.

Инфологическая модель используется для представления информации о предметной области. Инфологическое проектирование представляет собой этап создания инфологической модели.

Требования, которые предъявляются к инфологической модели:

- непротиворечивость (трактовка модели не может быть неоднозначной);
- адекватное отображение;
- легкость использования всеми пользователями;
- легкость увеличения (возможность расширения, то есть гарантировать ввод новых данных при неизменности ранее определенных);
- гибкий язык (язык должен быть применим как при ручном, так и при автоматизированном проектировании).

Из самого названия инфологической модели следует основное к ней требование, которое заключается в адекватности отображения предметной области. Таким образом, язык, который используется для представления ИЛМ обязан обладать выразительными возможностями для представления различных явлений, которые имеются в предметной области. Инфологическая модель не должна быть противоречивой. Она должна быть отражением потребностей и взглядов всех, кто пользуется системой и представлять собой единое интегрированное описание предметной области. Не допускается категорически неоднозначная трактовка используемой модели.

Цель инфологического моделирования заключается в создании точного и полного отображение реального мира, которое в дальнейшем будет

использоваться в качестве источника информации для построения базы данных.

Для всех категорий пользователей инфологической модели необходимо легкое ее восприятие. Инфологическая модель представляет собой ядро системы проектирования. При проектировании автоматизированной системы обработки информации инфологическая модель должна содержать всю достаточную и необходимую информацию. Информация из инфологической модели корреспондирует со словарной системой и другими компонентами банка данных. [6]

На рисунке 2.1 приведена инфологическая модель данных, которая использована для решения задач, поставленных в данной выпускной квалификационной работе. Данная модель показывает основные сущности, ключевые поля и атрибуты, входящие в каждую сущность. Также показаны информационные связи и потоки информации, позволяющие решить поставленные задачи.

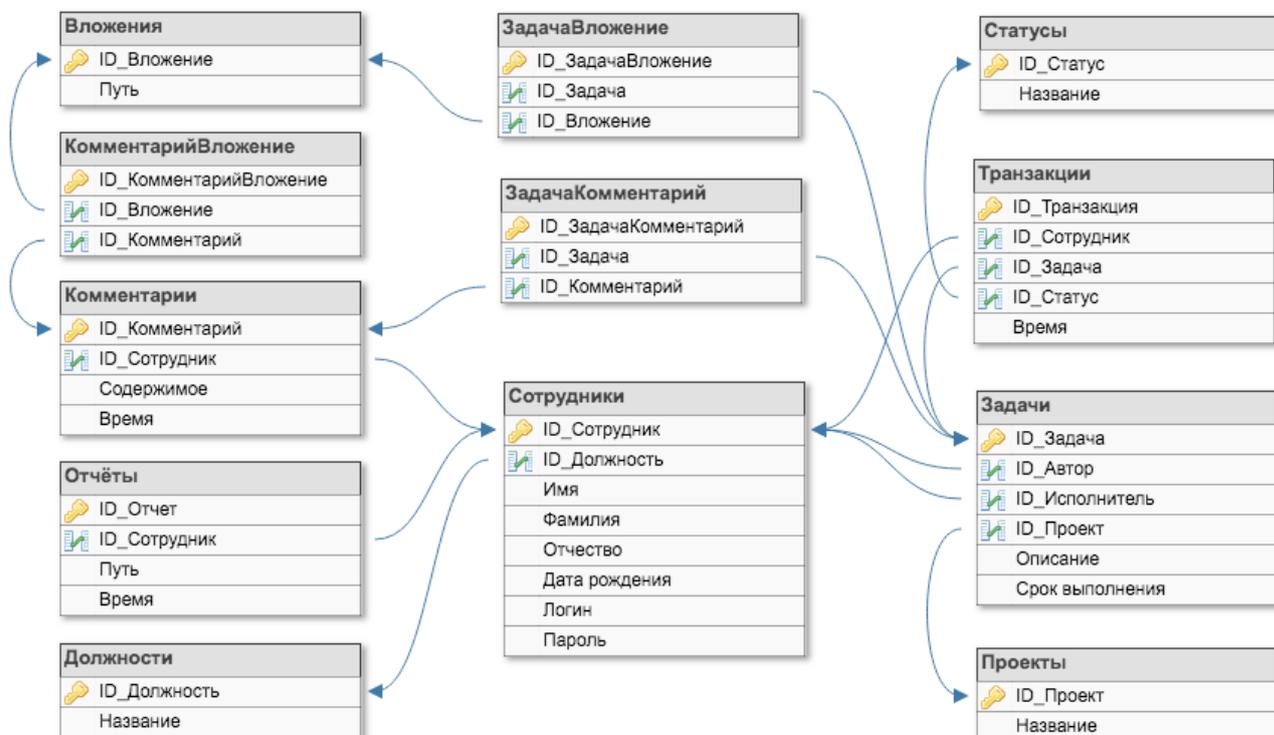


Рисунок 2.1 – Инфологическая модель данных

Модель данных, которая использована для решения задач, поставленных в данной выпускной квалификационной работе, содержит следующие сущности:

- Сотрудники;
- Задачи;
- Транзакции;
- Проекты;
- Статусы;
- Комментарии;
- Отчёты;
- Должности.

2.2.2 Характеристика даталогической модели БД

Даталогическое проектирование представляет собой организацию данных, выделенных на предыдущем этапе проектирования в форму, принятую в выбранной системе управления базами данных. Любая СУБД работает с допустимыми для нее логическими единицами данных, а также допускает использование определенных правил композиции логических структур более высокого уровня из составляющих информационных единиц более низкого уровня. Также, следует отметить, что многие системы управления базами данных накладывают количественные и иные ограничения на структуру базы данных. Поэтому, прежде чем приступить к построению даталогической модели, необходимо детально изучить особенности СУБД, определить факторы, влияющие на выбор проектного решения, ознакомиться с существующими методиками проектирования, а также провести анализ имеющихся средств автоматизации проектирования, возможности и целесообразности их использования. [26]

Несмотря на то, что даталогическое проектирование является

проектированием логической структуры базы данных, на него оказывают влияние возможности физической организации данных, предоставляемые конкретной системой управления базами данных. Поэтому знание особенностей физической организации данных является необходимым при проектировании логической структуры.

Логическая структура базы данных, а также сама заполненная данными база данных являются отображением реальной предметной области. Поэтому на выбор проектных решений самое непосредственное влияние оказывает специфика отображаемой предметной области, отраженная в инфологической модели. []

Конечным результатом даталогического проектирования является описание логической структуры базы данных на языке описания данных (ЯОД). Однако если проектирование выполняется «вручную», то для большей наглядности сначала необходимо построить схематическое графическое изображение структуры базы данных. При этом должно быть обеспечено однозначное соответствие между конструкциями языка описания данных и графическими обозначениями информационных единиц и связей между ними. Графическое представление используется и при автоматизированном проектировании структуры базы данных как интерфейсное средство общения с проектировщиком, и при документировании проекта.

Процесс проектирования логической структуры базы данных заключается в определении всех информационных единиц и связей между ними, указании их имён. В случаях, когда для информационных единиц возможно использование разных типов, необходимо определить их тип данных. Также необходимо для оптимизации работы базы данных задать количественные характеристики такие, как длину поля.

Даталогическая модель – это модель, которая отражает логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. Разработка даталогической модели осуществляется с учетом конкретной реализации системы управления базами данных, а также с

соблюдением специфики конкретной предметной области на основе ее инфологической модели.

На рисунке 2.2 представлена даталогическая модель базы данных, используемая для решения задач данной выпускной квалификационной работы.

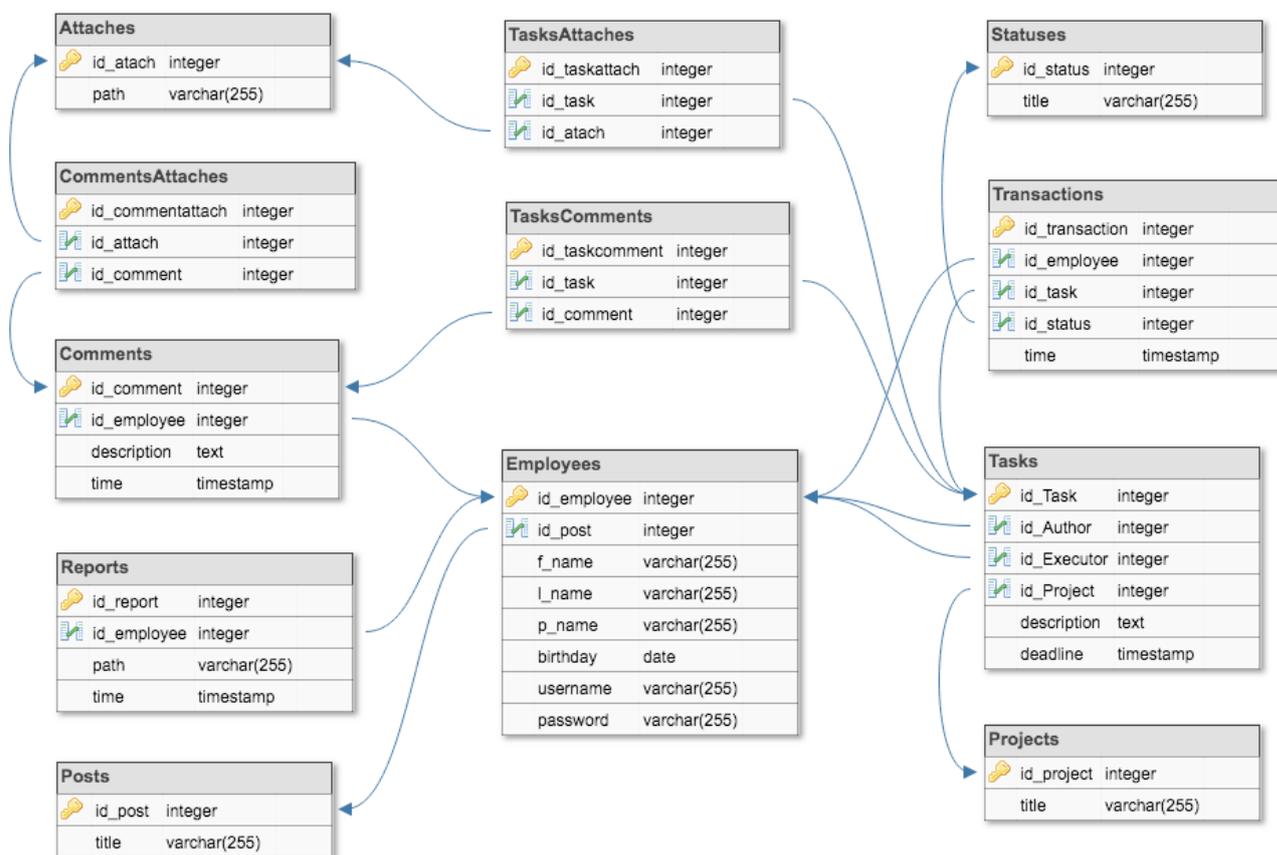


Рисунок 2.2 – Даталогическая модель базы данных

Табличное представление даталогической модели представлено в таблице 2.2.

Таблица 2.2 – Даталогическая модель базы данных

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип данных поля
1	2	3	4	5
Сотрудники	Employees	Код сотрудника	id_employee	integer
		Код должности	id_post	integer
		Фамилия	l_name	varchar(255)
		Имя	f_name	varchar(255)

Продолжение таблицы 2.2

1	2	3	4	5
		Отчество	p_name	varchar(255)
		Дата рождения	birthday	date
		Логин	username	varchar(255)
		Пароль	password	varchar(255)
Задачи	Tasks	Код задачи	id_task	integer
		Код автора	id_author	integer
		Код исполнителя	id_executor	integer
		Код проекта	id_project	integer
		Описание	description	text
		Срок выполнения	deadline	timestamp
Транзакции	Transactions	Код транзакции	id_transaction	integer
		Код сотрудника	id_employee	integer
		Код задачи	id_task	integer
		Код статуса	id_status	integer
		Время	time	timestamp
Проекты	Projects	Код проекта	id_project	integer
		Название	title	varchar(255)
Статусы	Statuses	Код статуса	id_status	integer
		Название	title	varchar(255)
Комментарии	Comments	Код комментария	id_comment	integer
		Код сотрудника	id_employee	integer
		Описание	description	text
		Время	time	timestamp
Отчёты	Reports	Код отчёта	id_report	integer
		Код сотрудника	id_employee	integer
		Путь	path	varchar(255)
		Время	time	timestamp
Должности	Posts	Код должности	id_post	integer
		Название	title	varchar(255)

Список SQL-запросов для создания БД представлен в приложении Б.

Во втором разделе проведено детальное обоснование и выбор различных проектных решений по техническому, информационному, программному и технологическому обеспечению. Сформулированы требования к информационному сопровождению и системе управления базами данных. Определены минимальные требования к аппаратному обеспечению компьютеров сотрудников отдел разработки.

3 Программная реализация

3.1 Описание дерева функций и сценария диалога

Основным способом описания структуры функций иерархических процессов, которые происходят в организации, является построение дерева функций. [6]

На рисунке 3.1 представлено дерево функций информационного сопровождения деятельности сотрудников, которое разделено на две составляющие «ветви»: основные и вспомогательные функции.

Среди основных функций расположены те, которые используются сотрудником (пользователем информационного сопровождения) непосредственно в ходе выполнения его основной проектной деятельности.

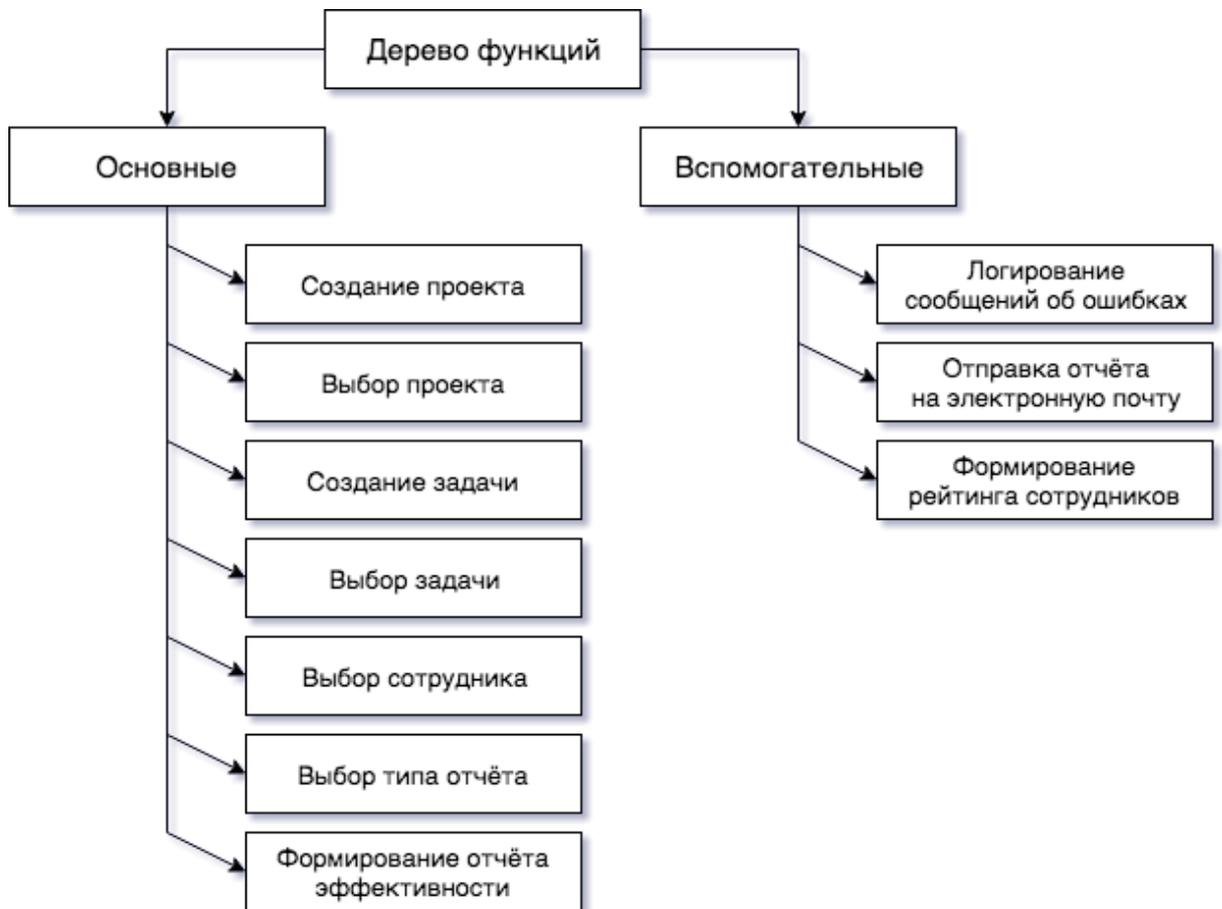


Рисунок 3.1 – Дерево функций

Вспомогательные функции – это те процессы, которые не являются необходимыми в основной деятельности сотрудников. Среди них можно выделить функцию «Отправки отчёта на электронную почту», которая доступна только генеральному директору организации в соответствующем разделе информационного сопровождения.

На рисунке 3.2 отражён сценарий диалога информационного сопровождение, который начинается с формы «Авторизации». Успешная авторизация в системе позволяет перейти к главному меню информационного сопровождения, которое состоит из пунктов перехода к основным разделам программного продукта. Такие разделы, как «Задачи», «Проекты», «Сотрудники» содержат в себе диалоговые формы создания, редактирования и удаления соответствующих записей в базе данных. Раздел «Отчёты» содержит в себе пункты перехода ко всем вида отчётных документов информационного сопровождения.

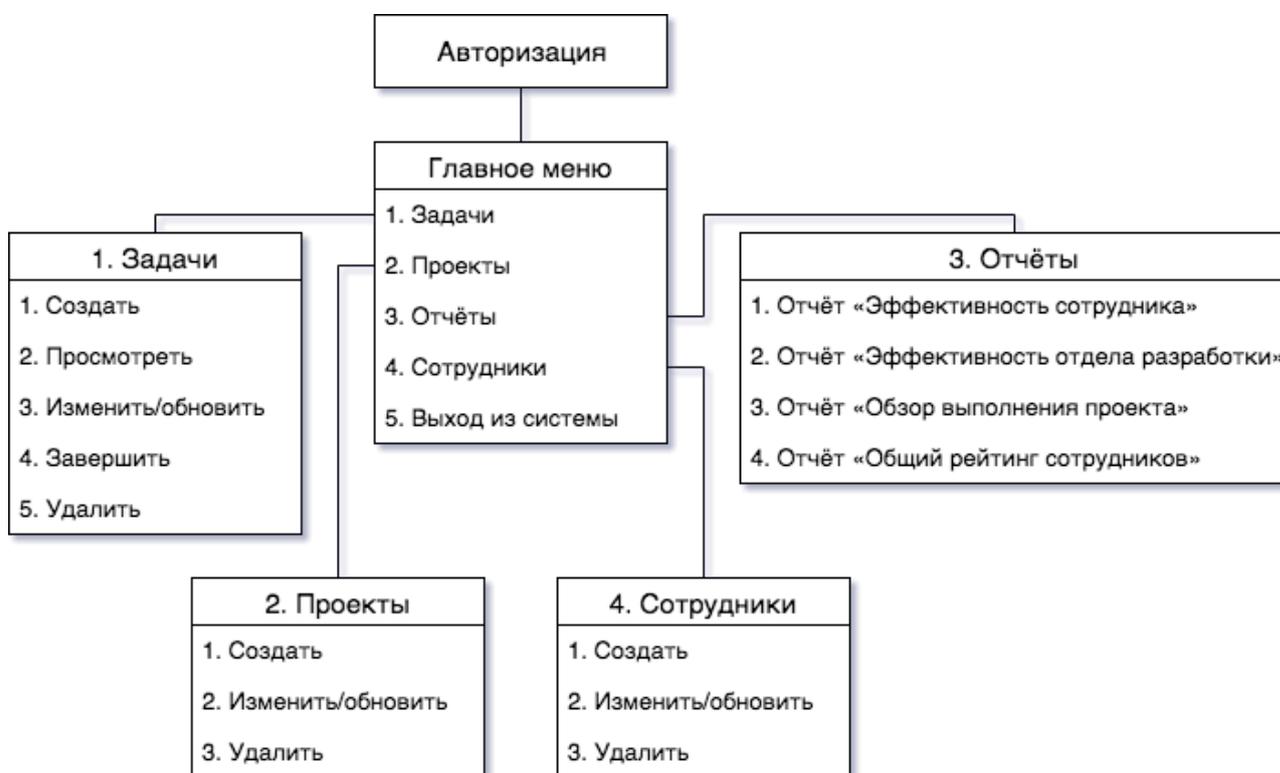


Рисунок 3.2 – Сценарий диалога

3.2 Организация технологии сбора, передачи, обработки и выдачи информации

Для сбора, передачи, обработки и выдачи информации в информационном сопровождении используются возможности языка программирования PHP, системы управления базами данных MySQL с применением языка запросов к данным SQL. [9]

В качестве шаблона проектирования была выбрана схема разделения данных «Модель-Представление-Контроллер» (MVC). Концепция шаблона заключается в разделении бизнес-логики и визуализации проекта. Название включает в себе все составные части: Модель (Model), Представление (View), Контроллер (Controller). Каждый компонент отвечает за свою задачу. [4]

Модель управляет поведением и данными приложения, отвечает на запросы информации о его состоянии (обычно из Представления) и отвечает на инструкции по изменению состояния (обычно с Контроллера).

Представление управляет отображением информации в интерфейсе программного продукта.

Контроллер интерпретирует входные данные мыши и клавиатуры от пользователя, информируя Модель и/или Представление с целью их изменения соответствующим образом.

Взаимодействия компонентов представлены на рисунке 3.3.

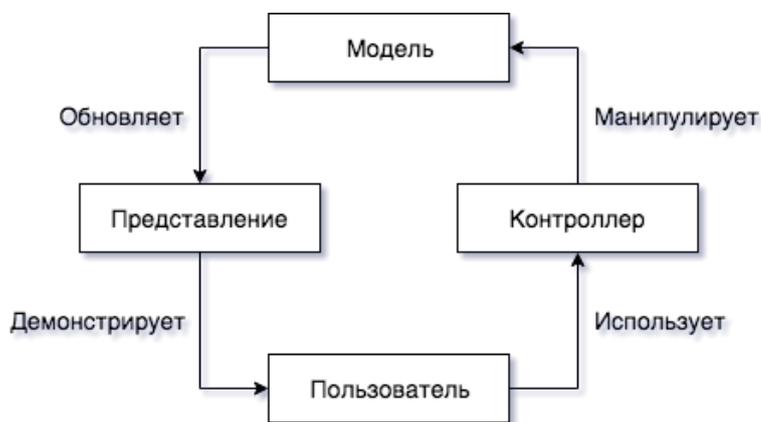


Рисунок 3.3 – Взаимодействие компонентов

3.3 Описание контрольного примера реализации проекта

Рассмотрим интерфейс информационного сопровождения и его основные функции. Каскадная таблица стилей визуального интерфейса информационного сопровождения представлена в приложении Б. Стандартным способом осуществляем запуск интернет-обозревателя. Чтобы запустить информационное сопровождение и, собственно, начать работу с ним необходимо перейти на соответствующий веб-ресурс в локальной сети организации. Для входа в система необходимо пройти аутентификацию, введя имя пользователя и пароль (см. рисунок 3.4).

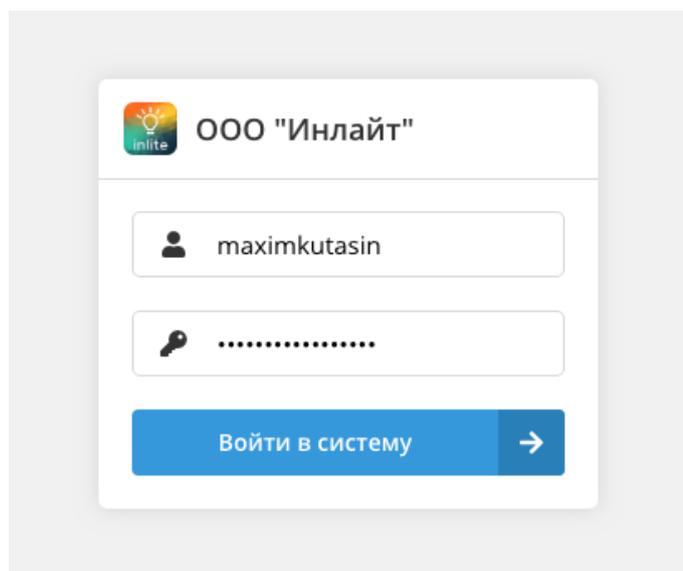


Рисунок 3.4 – Окно авторизации

В данном информационном сопровождении существует разграничение прав доступа для сотрудников и директора организации. Для сотрудников установлены частичные права записи и чтения, а для генерального директора – полные права доступа.

Рассмотрим информационное сопровождение, при входе директора организации. Войдем под пользователем «maximkutasin».

Интерфейс системы состоит из двух основных частей: боковая навигация

(см. рисунок 3.5) и область контента.

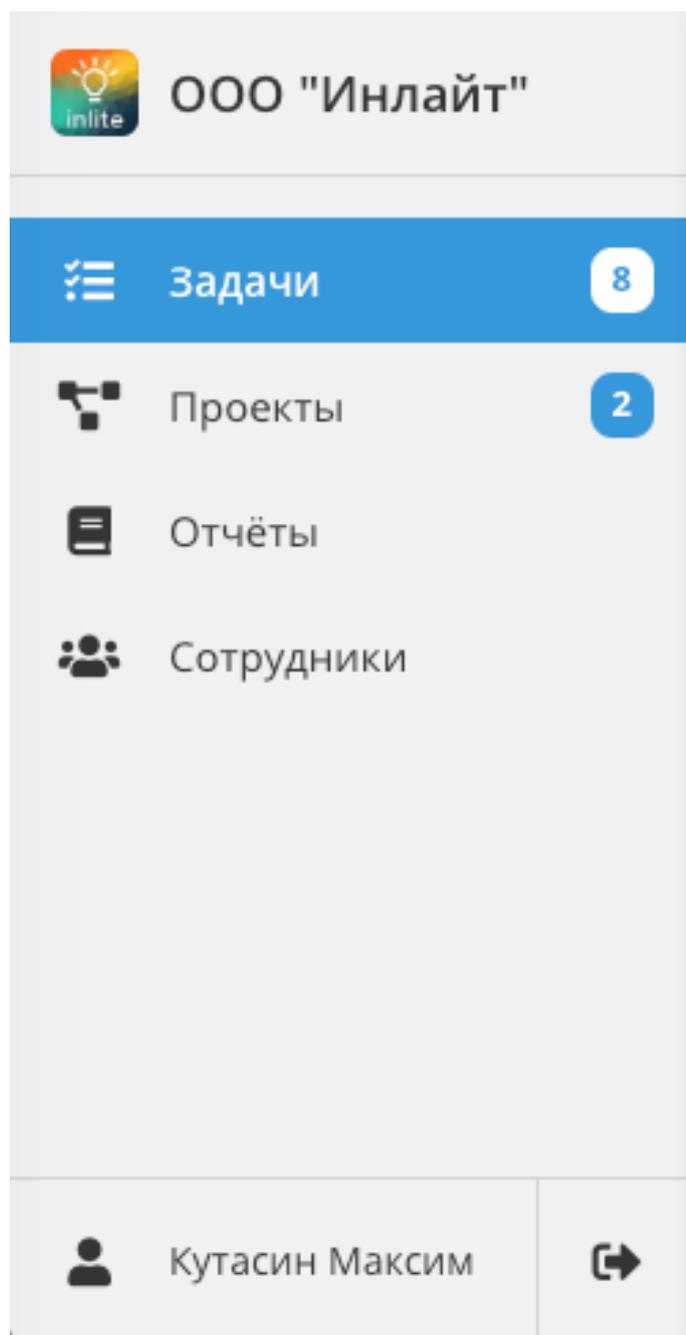


Рисунок 3.5 – Боковая навигация

Основная навигация позволяет пользователю системы быстро перейти к любому разделу информационного сопровождения. Среди них: «Задачи», «Проекты», «Отчёты», «Сотрудники». Рассмотрим каждый из них.

На рисунке 3.6 представлен интерфейс основного раздела системы - «Задачи».

Задачи				+ Добавить задачу
#	Статус	Приоритет	Описание	Исполнитель
NP-173	ОЖИДАЕТ	Обычный	Fixing polygon in polygon algorithm (Algorithm is optimized for additional poin...	Завтраков Андрей
R1-172	В ПРОЦЕССЕ	! Высокий	Provide assesor area data instead of our one	Вирский Павел
RI-171	ГОТОВО	Обычный	Implement Tabs	Вирский Павел
NP-170	ВОПРОС	! Высокий	Copy parcel data to deals	Платонов Валерий
NP-169	ОЖИДАЕТ	Обычный	Add more data to parcels	Завтраков Андрей
R1-168	В ПРОЦЕССЕ	! Высокий	Split the ownership into private and public based on Urbyn rules [+]	Долматов Евгений
RI-167	ГОТОВО	Обычный	UI how to highlight a group of parcels on the map	Платонов Валерий
R1-166	ВОПРОС	! Высокий	Change the default BBL parcel format to 4-10235-0367 (zeros + dashes)	Завтраков Андрей

Рисунок 3.6 – Раздел «Задачи»

Раздел состоит из заголовка и кнопки «Добавить задачу», которая открывает диалоговое окно для заполнения данных новой задачи (см. рисунок 3.7). Так же в разделе представлена таблица с краткой информацией об актуальных задачах во всех проектах. Информация о каждой задаче содержит в себе сведения о статусе, приоритете, исполнителе и т.п.

Добавить задачу
✕

Проект:

Исполнитель:

Срок выполнения:

Приоритет:

Описание:

Отменить
Добавить

Рисунок 3.7 – Диалоговое окно «Добавить задачу»

Переход на страницу с подробной информацией о задаче происходит щелчком мыши по соответствующей строке в таблице.

На рисунке 3.8 представлена страница конкретной отдельной задачи. Интерфейс состоит из двух основных элементов: слева – список всей информации о задаче (статус, приоритет, назначенный исполнитель, автор задачи, даты создания и необходимого выполнения), справа – блок с комментариями сотрудников, которые они формируют в процессе обсуждения выбранной задачи.

К каждой задаче может быть прикреплено неограниченное количество вложений, что представлено в блоке «Вложения». Это могут быть любые файлы, которые имеют отношение к выполнению задачи. Кроме этого, подобные вложения могут быть в каждом отдельном комментарии, что обеспечивает версию файлов и историю их изменений.

The screenshot shows a task page titled "Задача NP-173" with a "Редактировать задачу" button. The left sidebar contains the following information:

- Статус:** ОЖИДАЕТ
- Приоритет:** Высокий
- Исполнитель:** Завтраков Андрей
- Автор:** Долматов Евгений
- Создана:** 14 мая 2018 г.
- Выполнить к:** 20 мая 2018 г.
- Описание:** Fixing polygon in polygon algorithm (Algorithm is optimized for additional point at the end and our model is not)
- Вложения:** catalog_podborki_01.psd, catalog_podborki_inner_01.psd, catalog_sale_01.psd, main_01.psd, search_results_01.psd, search_resultsemtyp_01.psd

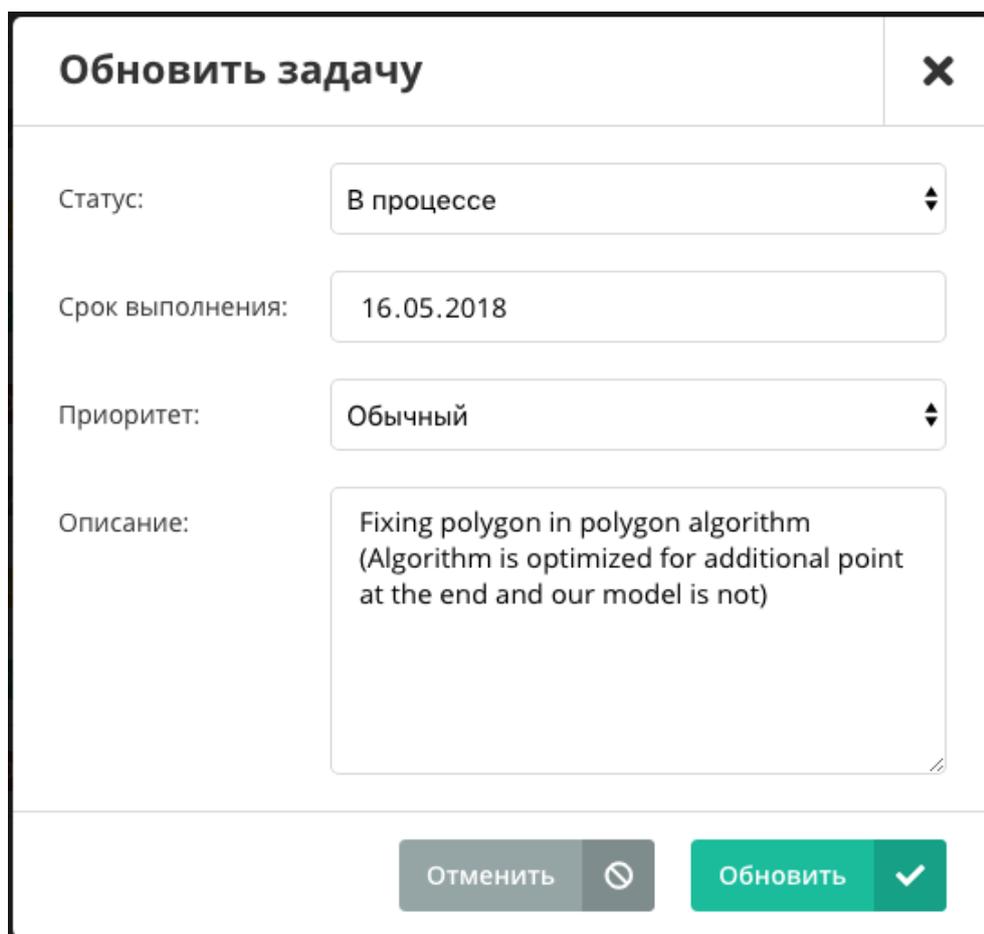
The right sidebar shows a comment thread:

- steve 00:01:** По идее весь флоу авторизации/регистрации готов технически, не хватает пары кнопок (но это не технически сложная задача). Не хватает логаута на экранах с ошибками или если аккаунт заблочен, не хватает переключателя организаций.
- steve 00:21:** А можно все в цеплин?
- minooor 00:21:** да, конечно. скоро внесу правки и обновлю
Attachments: catalog_podborki_01.psd
- steve 01:25:** а где интерком на лендинге?!
- minooor 03:10:** чеклист и эшкены на хоум завтра доделаю
- steve 03:17:** больше наглости в чеклисте и лучше наверное горизонтально сделать прогресс и на всю ширину (сдвинуть весь контент) и добавить крестик справа что бы можно было убрать

At the bottom, there is a text input field "Ваш комментарий..." and an "Отправить" button.

Рисунок 3.8 – Подробная информация о задаче

В правом верхнем углу интерфейса расположена кнопка «Редактировать задачу», которая вызывает диалоговое окно для внесения изменений в основную информацию о задаче (см. рисунок 3.9). Среди прочего в данном окне представлена возможность изменить описание, сменить исполнителя, обновить дату необходимого выполнения задачи и т.п.



Обновить задачу X

Статус: В процессе

Срок выполнения: 16.05.2018

Приоритет: Обычный

Описание: Fixing polygon in polygon algorithm
(Algorithm is optimized for additional point at the end and our model is not)

Отменить Отменить

Обновить Обновить

Рисунок 3.9 – Диалоговое окно «Обновить задачу»

Следующий, доступный для пользователя, раздел - «Проекты» представлен на рисунке 3.10. В данном разделе представлена сводка о текущих проектах, которые проектируются и разрабатываются сотрудниками отдела разработки организации. Информацию о проектах заполняет руководитель отдела или директор компании. Сотрудникам доступ к изменениям в данном разделе ограничен.

Проекты		+ Добавить проект			
#	Название	ОЖИДАЕТ	В ПРОЦЕССЕ	ГОТОВО	ВОПРОС
NP	НТВ-Плюс	2	0	0	1
R1	Радио 1 Подмосковья	0	2	0	1
RI	Риамо	0	0	2	0

Рисунок 3.10 – Раздел «Проекты»

В данном разделе, как и во всех остальных, в правом верхнем углу расположена кнопка для открытия диалогового окна добавления записей в раздел. Форма создания проекта и заполнения информации о нём представлена на рисунке 3.11. В ней представлены два текстовых поля, в которые необходимо внести идентификационный код проекта и его название.

Добавить проект
✕

Код:

Название:

Отменить
Добавить

Рисунок 3.11 – Диалоговое окно «Добавить проект»

На рисунке 3.12 представлен раздел «Сотрудники», который содержит всю необходимую информацию о каждом сотруднике отдела разработки

Сотрудники						+ Добавить сотрудника
#	Фамилия	Имя	Отчество	Должность	Дата рождения	
E-4	Завтраков	Андрей	Валерьевич	Программист	30 Июн 1982 г.	
E-3	Вирский	Павел	Иванович	Дизайнер	25 Июл 1987 г.	
E-2	Платонов	Валерий	Фёдорович	Программист	2 Фев 1992 г.	
E-1	Долматов	Евгений	Леонидович	Менеджер проекта	12 Апр 1974 г.	

Рисунок 3.12 – Раздел «Сотрудники»

Среди прочей информации представлены: ФИО сотрудника, должность с пиктографическим и цветовым выделением и дата рождения. Все возможные варианты оформления должности сотрудников представлены на рисунке 3.13.



Рисунок 3.13 – Варианты представления должности сотрудника

Диалоговое окно редактирования данных сотрудника оформлено в привычном уже ключе всех форм данного информационного сопровождения. Оно позволяет редактировать всю информацию о сотруднике и, к тому же, обновить его логин и пароль (см. рисунок 3.14)

Рисунок 3.14 – Диалоговое окно «Обновить данные сотрудника»

Последним рассмотренным разделом информационного сопровождения является раздел «Отчёты», который представлен на рисунке 3.15. Он представляет из себя таблицу со списком сформированных ранее отчётов о деятельности сотрудников отдела разработки и самого отдела, в частности.

Отчёты		
#	Дата создания	Тип
R-15	14 мая 2018 г.	Эффективность сотрудника "Завтраков Андрей" за Апрель 2018 г.
R-14	10 мая 2018 г.	Эффективность отдела разработки за Май 2018 г.
R-13	21 апреля 2018 г.	Обзор выполнения проекта "Радио 1 Подмосковья"
R-12	13 апреля 2018 г.	Общий рейтинг сотрудников
R-11	11 апреля 2018 г.	Обзор выполнения проекта "Риамо"
R-10	09 апреля 2018 г.	Эффективность сотрудника "Завтраков Андрей" за Март 2018 г.
R-9	07 апреля 2018 г.	Эффективность сотрудника "Платонов Валерий" за Март 2018 г.

Сформировать отчёт

Рисунок 3.15 – Раздел «Отчёты»

Доступ к чтению и записи в данном разделе ограничен для всех пользователей системы за исключением директора организации.

По нажатию на кнопку «Сформировать отчёт» открывается диалоговое окно с формой для создания отчёта (см. рисунок 3.16). В данной форме директору необходимо выбрать один из доступных типов отчёта: «Эффективность сотрудника», «Эффективность отдела разработки», «Обзор выполнения проекта», «Общий рейтинг сотрудников». В зависимости от выбранного типа будут предложены для ввода соответствующие данные. Рассмотрен случай при выборе типа «Эффективность сотрудника». Для формирования отчёта необходимо выбрать конкретного сотрудника организации и период отчётности.

Сформировать отчёт ✕

Тип:

Сотрудник:

Период:

Отправить отчёт на почту

Рисунок 3.16 – Диалоговое окно «Сформировать отчёт»

Так же существует возможность отправить сформированный отчёт на электронный почтовый адрес генерального директора организации. Для этого необходимо выбрать соответствующий пункт «Отправить отчёт на почту» в форме диалогового окна.

Рассмотрим каждый вариант сформированного отчёта на соответствующих примера.

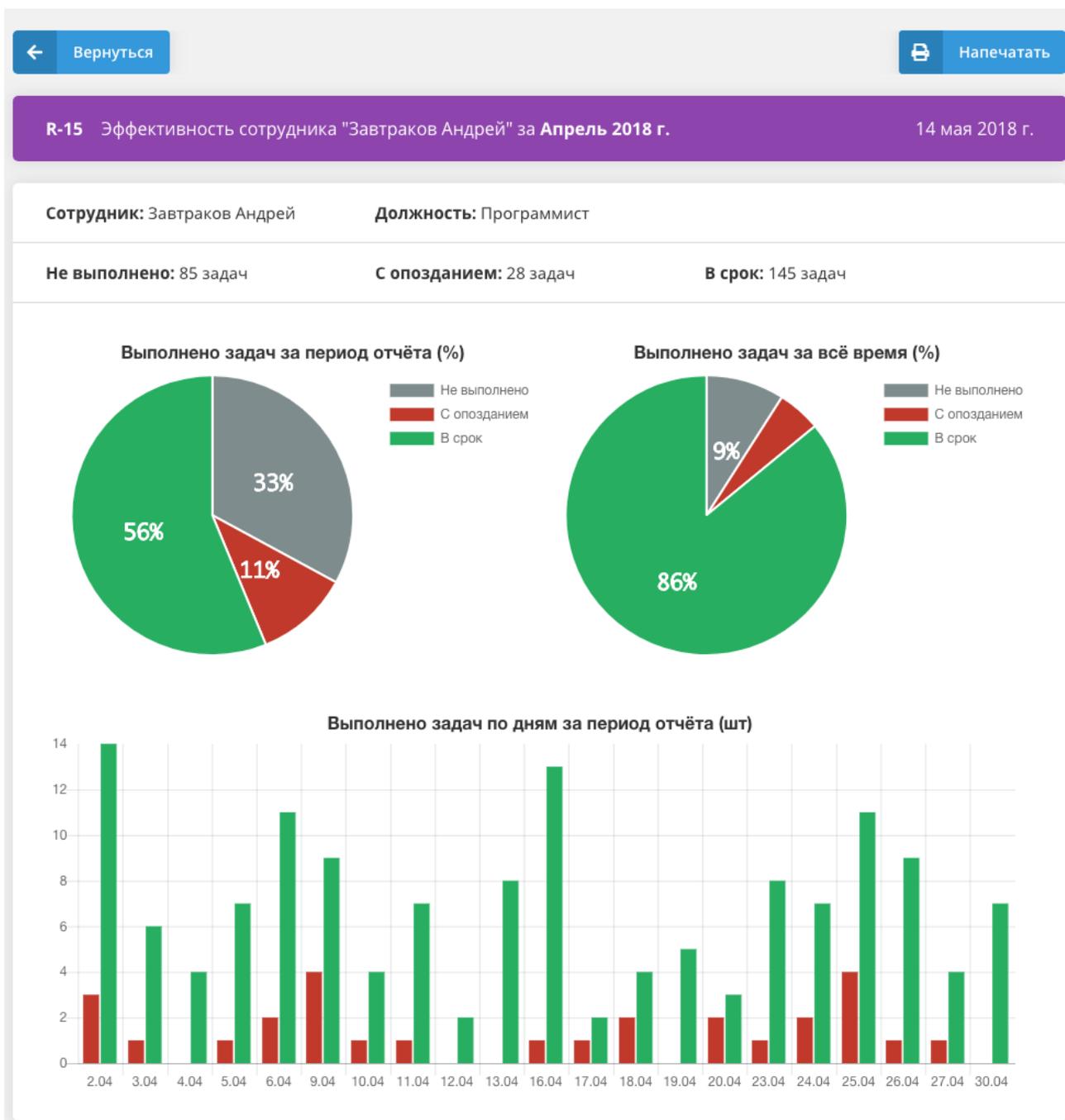


Рисунок 3.17 – Пример отчёта «Эффективность сотрудника»

На рисунке 3.17 представлен пример сформированного отчёта с анализом эффективности выбранного сотрудника. Отчёт состоит из трёх диаграмм различного типа (круговые и столбчатая). Круговые диаграммы демонстрируют в процентах относительное количество задач с различным статусом (не выполнено, выполнено с опозданием, выполнено в срок) за выбранный отчётный период и за всё время работы сотрудника. Столбчатая диаграмма позволяет рассмотреть статистику выполнения конкретных задач в каждый день сформированного отчёта, исключая выходные дни.

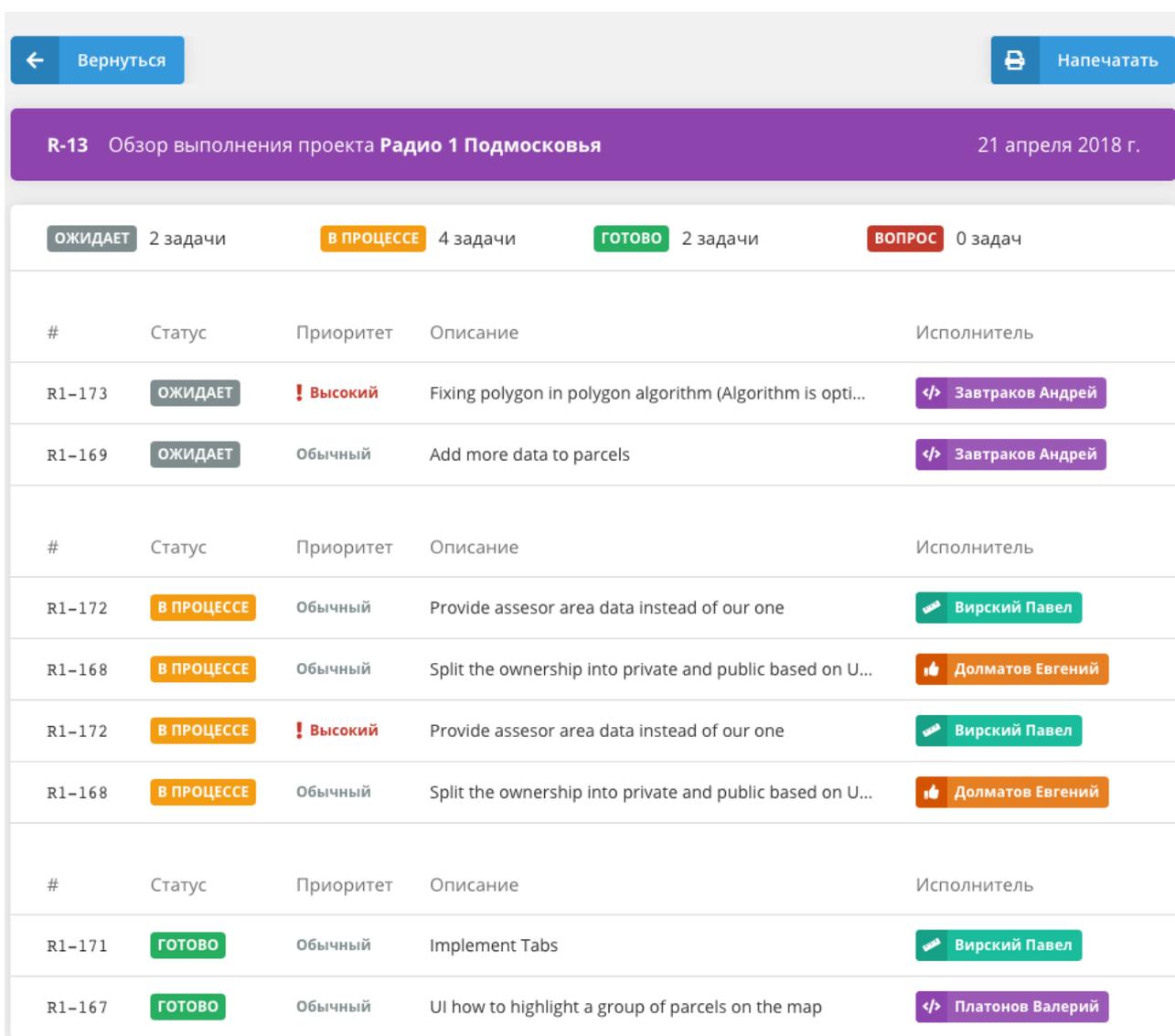
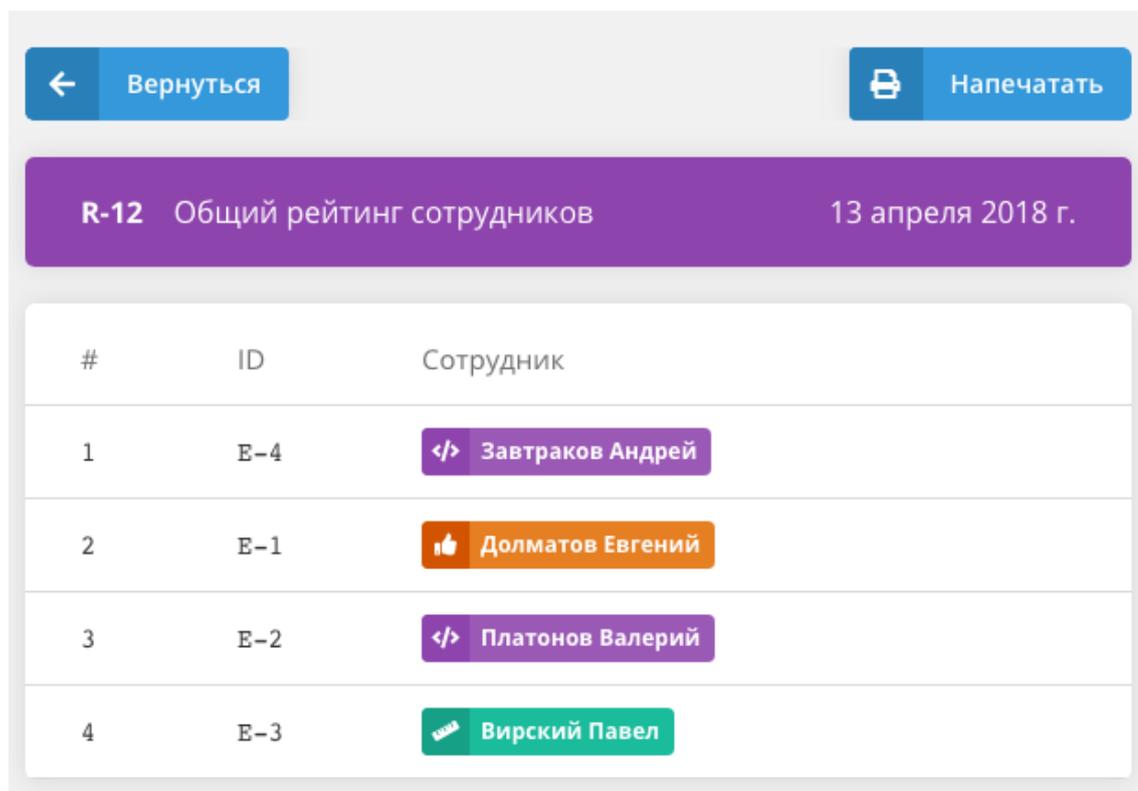


Рисунок 3.18 – Пример отчёта «Обзор выполнения проекта»

На рисунке 3.18 продемонстрирован пример отчёта «Обзор выполнения проекта». Суть отчёта заключается в формировании краткой сводки о текущем состоянии всех задач выбранного проекта. Задачи отсортированы по своему статусу («Ожидает», «В процессе», «Готово», «Вопрос»).



#	ID	Сотрудник
1	E-4	 Завтраков Андрей
2	E-1	 Долматов Евгений
3	E-2	 Платонов Валерий
4	E-3	 Вирский Павел

Рисунок 3.19 – Пример отчёта «Общий рейтинг сотрудников»

Отчёт «Общий рейтинг сотрудников», представленный на рисунке 3.19, является весьма формальным. Его суть сводится к сравнению выполненных задач за выбранный период или за всё время работы сотрудников.

На рисунке 3.20 представлен пример отчёт «Эффективность отдела разработки». В его основе лежит структура и логика отчёта «Эффективность сотрудника» за исключением того, что в качестве объекта используется не конкретный сотрудник, а весь отдел разработки в целом, включая всех сотрудников.

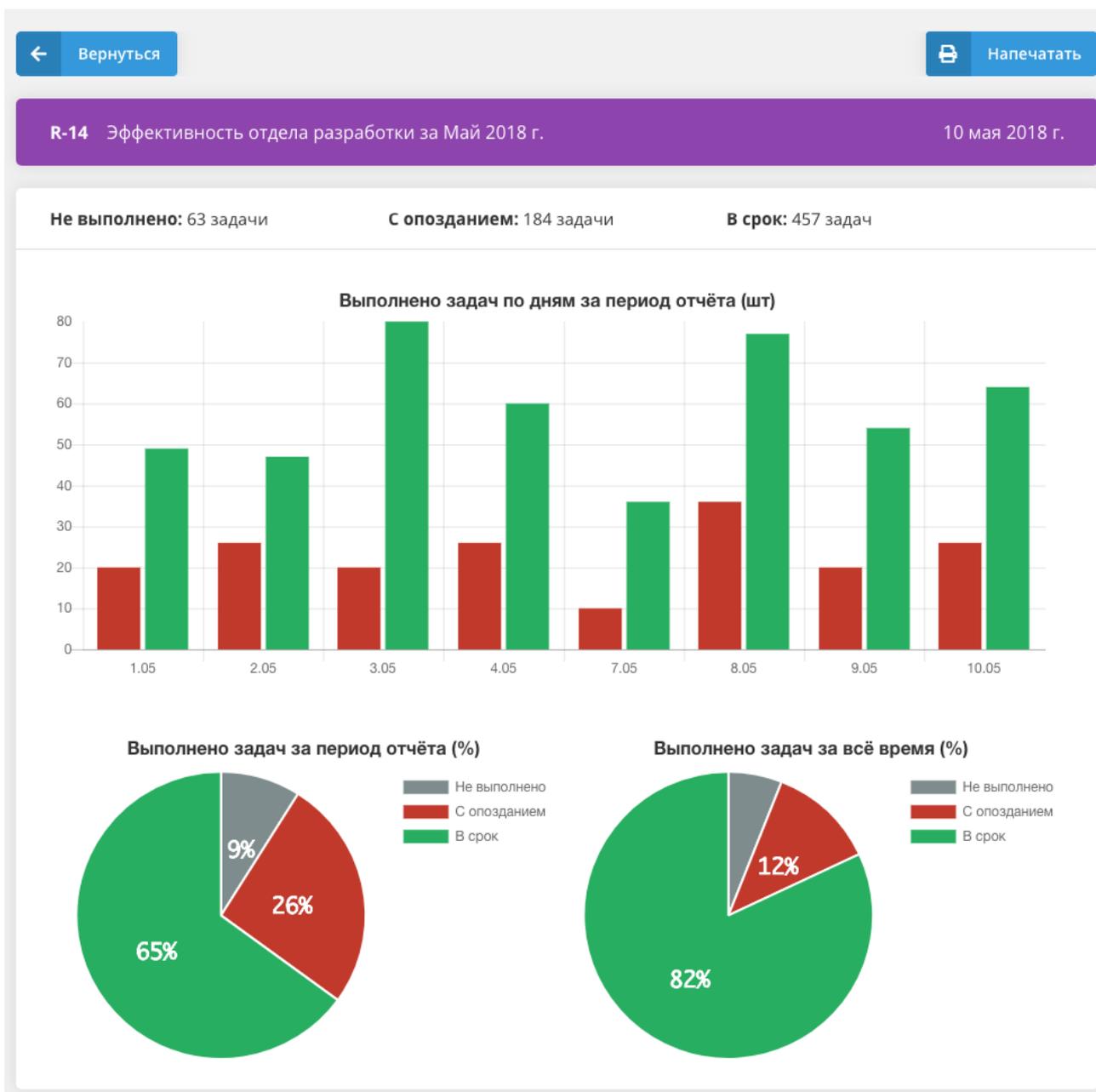


Рисунок 3.20 – Пример отчёта «Эффективность отдела разработки»

На этом описание контрольного примера реализации завершено. Были рассмотрены все разделы информационного сопровождения деятельности сотрудников отдела разработки, все диалоговые окна взаимодействия пользователя с системой, все варианты отчётов и т.п.

3.4 Обоснование эффективности экономической разработки

3.4.1 Выбор и обоснование показателей эффективности и методики их расчёта

Разработка информационного сопровождения ООО «ИНЛАЙТ» позволит снизить трудоемкость работ, повысить производительность труда персонала, в связи с уменьшением рутинной работы, сокращением избыточности хранимых данных, а следовательно, экономией объёма используемой памяти, уменьшением затрат на многократные операции обновления избыточных копий и устранением возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте, увеличением степени достоверности информации и увеличением скорости обработки информации; излишнее количество внутренних промежуточных документов и т.д., повторное внесение одной и той же информации в различные промежуточные документы.

Внедрение данного информационного сопровождения значительно облегчает работу сотрудников отдела разработки. Совершенствование же разработки при недостатке финансирования и нехватке кадров невозможно; угроза устаревания разработки является несущественной. Таким образом, разработанное информационное сопровождение решает все поставленные задачи для учёта деятельности сотрудников в ООО «ИНЛАЙТ». Для определения целесообразности разработки нового информационного сопровождения необходимо произвести расчёт показателей экономической эффективности.

3.4.2 Расчёт показателей эффективности

Экономическая эффективность - результативность системы, выражающаяся в отношении полезных конечных результатов ее

функционирования к затраченным ресурсам. При проектировании данной информационной системы были определены этапы разработки:

- анализ и изучение технического задания - 12 часов;
- анализ библиографических источников - 40 часов;
- проектирование информационного сопровождения - 25 часов;
- разработка информационного сопровождения - 50 часов;
- ввод в эксплуатацию и тестирование созданного информационного сопровождения - 40 часов;
- разработка инструкции и описания информационного сопровождения - 30 часов.

Определяем состав технических специалистов для осуществления работ по каждому этапу: инженер-программист и разработчик информационного сопровождения.

Расчет стоимости создания информационного сопровождения для ООО «ИНЛАЙТ» выполним по следующим статьям расходов:

- накладные расходы;
- прямые материальные затраты;
- заработная плата (фонд оплаты труда);
- отчисления на социальные нужды.

Расчет материальных затрат приведен в таблице 3.1.

Таблица 3.1 – Расчёт прямых материальных затрат

Материалы	Ед. измер.	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	лист	200	0.5	100
Картридж для принтера	шт.	1	400	400
Диск	шт.	5	10	50
Итого				550
Расходы на энергию	кв/ч	100	5.38	538
ИТОГО				1088

Прямые материальные затраты составили 1088 рублей. Для расчёта заработной платы (фонда оплаты труда) необходимо рассчитать заработную плату разработчиков информационного сопровождения, составить баланс рабочего времени. Данные сведём в таблице 3.2.

Таблица 3.2 – Баланс рабочего времени

П/п	Наименование показателей	ИТР
1	Число календарных дней в году	0.5
2	Число выходных и нерабочих дней в году	400
3	Число рабочих дней в году	10
4	Невыходы на работу:	
	а) по болезни	0
	б) очередной отпуск	0
5	Фактическое число рабочих дней в году	253
6	Продолжительность рабочего дня	8
7	Годовой фонд рабочего времени (час.)	2024

Часовую ставку заработной платы (Чс) определяем по формуле:

$$Чс = (З * П * К) / Ф \quad (3.1)$$

где Чс – месячная зарплата, руб.;

П – число месяцев в году, исключая отпуск;

К – коэффициент, учитывающий премии из фонда зарплаты;

Ф – фактический годовой фонд рабочего времени, час.

Примем среднемесячную зарплату инженера-программиста и разработчика информационного сопровождения равной:

$$З = 40\,000 \text{ руб.},$$

При $П = 11$, а $Ф = 2024$ час., получим $Чс = 217,39$ руб./час.

Размер основной заработной платы определяем исходя из времени, затрачиваемого на выполнение работ и стоимости часа работы исполнителя.

Основная заработная плата определяется по графику основных этапов работ. Дополнительная заработная плата может составить до 15% от основной. Для расчёта оплаты труда необходимо учитывать ряд особенностей, такие как анализ ТЗ, подбор литературы, разработка самого алгоритма и структуры программы, основная заработная плата, дополнительная заработная плата (премии) и т.д. Расчёт фонда оплаты труда приведён в таблице 3.3.

Таблица 3.3 – Расчёт фонда оплаты труда

Этапы разработки	Время (час)	Часовая ставка (руб.)	Сумма (руб.)
Анализ ТЗ	12	217,39	2608,68
Подбор, изучение литературы	40	217,39	8695,6
Разработка алгоритма и структуры программы	25	217,39	5434,75
Программирование	50	217,39	10869,5
Тестирование ПО	40	217,39	8695,6
Разработка инструкций ПО	30	217,39	6521,7
Основная заработная плата (итого)			42825,83
Дополнительная заработная плата			6423,87
ИТОГО			49249,70

Размер отчислений на социальные нужды определяется исходя из размера фонда оплаты труда.

Расчёт приведен в таблице 3.4.

Таблица 3.4 – Отчисления на социальные нужды

Отчисления	Доля от фонда оплаты труда (%)	Сумма (руб.)
В пенсионный фонд	20,6	10145,43
В фонд занятости	1,7	837,24
Медицинское страхование	2,6	1280,49
Социальное страхование	2,9	1428,24
ИТОГО	27,8	13691,4

Величина накладных расходов определяется в размере 80% от основной заработной платы и составит и определяется по формуле 3.2:

$$HP = 0.8 * 42825,83 = 34260,66 \quad (3.2)$$

Полную смету затрат на разработку системы приведём в таблице 3.5.

Таблица 3.5 – Расчёт затрат на разработку информационного сопровождения

Статья расхода	Сумма (руб.)
Фонд оплаты труда	49249,70
Отчисления на социальные нужды	13691,4
Материальные затраты	1088
Накладные расходы	34260,66
ИТОГО	98289,76

Расчёт ежемесячных затрат на эксплуатацию системы приведём в таблице 3.6.

Таблица 3.6 – Расчёт затрат на эксплуатацию системы

Статья расхода	Сумма (руб.)
Зарплата администратора БД	30000
Отчисления на социальные нужды	8340
Затраты на электроэнергию	538
ИТОГО	38878

Таким образом, проведён расчёт экономической эффективности проекта с учётом затрат на фонд оплаты труда, отчисления на социальные нужды и эксплуатацию систему в будущем.

В третьем разделе описан разработанный программный продукт, дерево функций, сценарий диалога, процесс организации технологии сбора, передачи, обработки и выдачи информации. Представлен контрольный пример реализации проекта, а также приведён расчёт экономических затрат на разработку и эксплуатацию системы.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы для учёта деятельности сотрудников отдела разработки было разработано информационное сопровождение деятельности сотрудников в ООО «ИНЛАЙТ», организующее динамическое обновление информации, а также накапливающее информацию в виде, позволяющем анализировать деятельность сотрудников за любой промежуток времени.

В процессе выполнения выпускной квалификационной работы был реализован ряд следующих задач:

- изучена предметная область;
- выполнен сбор и анализ данных;
- приведено обоснование выбора основных проектных решений;
- выполнено проектирование системы;
- разработано информационное сопровождение деятельности сотрудников;
- проведено тестирование разработанной системы;
- выполнено обоснование экономической эффективности проекта.

В процессе выполнения выпускной квалификационной работы было разработано информационное сопровождение деятельности сотрудников отдела разработки ООО «ИНЛАЙТ».

Был получен акт о внедрении информационного сопровождения (см. приложение В).

Внедрение данного сопровождения позволяет сотрудникам ООО «ИНЛАЙТ» рациональнее использовать рабочее время для эффективной организации процесса трудовой деятельности, обсуждения поставленных задач, не расходуя его на поиск требуемой информации в разных отделах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алиев, С.А. Быстрая разработка программного обеспечения [Текст] / С.А. Алиев. – М., 2013. – 336 с.
2. Андрейчиков, А.А. Системный анализ и синтез стратегических решений в инноватике. Концептуальное проектирование инновационных систем [Текст] / А.А. Андрейчиков, О.С. Андрейчикова. – М., 2013. – 402 с.
3. Антонова, А.С. Автоматизированные информационные системы, базы и банки данных. Вводный курс: Учебное пособие [Текст] / А.С. Антонова. – М.: Гелиос АРВ, 2014. – 368 с.
4. Астел, Д.С. Практическое руководство по экстремальному программированию [Текст] / Д.С. Астел. – М.: Издательский дом "Вильямс", 2011. – 320 с.
5. Баранова, Е. Н. Основы информатики и защиты информации [Текст] / Е.Н. Баранова. – М., 2013. – 192 с.
6. Баранова, Е.Н. Информационная безопасность и защита информации [Текст] / Е.Н. Баранова, А.Н. Бабаш. – М., 2014. – 256 с.
7. Билл Карвин, Программирование баз данных SQL. Типичные ошибки и их устранение [Текст] / Б. Карвин. – М., 2014. – 336 с.
8. Брайан, С.А. HTML5 и CSS3. Веб-разработка по стандартам нового поколения [Текст] / С.А. Брайан – М., 2014. – 320 с.
9. Бретт, М.К. PHP и MySQL. Исчерпывающее руководство [Текст] / М.К. Бретт. – М., 2014. – 544 с.
10. Брюс, Б.Б. Семь языков за семь недель. Практическое руководство по изучению языков программирования [Текст] / Б.Б. Брюс. – М., 2014. – 384 с.
11. Вигерс, Б.М. Разработка требований к программному обеспечению [Текст] / Б.М. Вингерс. – М., 2014. – 736 с.
12. Гашков, С.Б. Криптографические методы защиты информации [Текст] / С.Б. Гашков, Э.А. Применко, М.А. Черепнев. – М., Издательство: Академия, 2010. – 304 с.

13. Голицына, О. Л. Информационные технологии [Текст] / О.Л. Голицына, И.И. Попов, Н.В. Максимов, Т.Л. Партыка. – М., 2014. – 608 с.
14. Головков, А.С. HTML5 и CSS3 для всех [Текст] / А.С. Головков. – М.: Эксмо-Пресс, 2012. – 368 с.
15. ГОСТ Р ИСО/МЭК 12207/99. Государственный стандарт РФ. Информационная технология. Процессы жизненного цикла информационных систем. Издание официальное [Текст]. – М., 2012. – 78 с.
16. Гусятников, В.Н. Стандартизация и разработка программных систем. [Текст] / В.Н. Гусятников. – М.: Финансы и статистика, 2012. – 288 с.
17. Джигалов, А.С. HTML5, CSS3 и JavaScript. Исчерпывающее руководство [Текст] / А.С. Джигалов. – М., 2014. – 528 с.
18. Довлатов, П.Р. Практическое руководство по доступу к данным [Текст] / П.Р. Довлатов. – СПб, 2013. – 304 с.
19. Дудукало, А.В. HTML и CSS. Разработка и дизайн веб-сайтов [Текст] / А.В. Дудукало. – М., 2013. – 480 с.
20. Дунаев В.А. HTML, скрипты и стили [Текст] / В.А. Дунаев. – М., 2015. – 824 с.
21. Дэвид Макфарланд Большая книга CSS3 [Текст] / Д. Макфарланд. – М., 2014. – 608 с.
22. Дэвид Паттерсон. Архитектура компьютера и проектирование компьютерных систем [Текст] / Д. Паттерсон. – М., 2014. – 784 с.
23. Емельянова, Н.З. Проектирование информационных систем [Текст] / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. – М., 2014. – 432 с.
24. Жадаев, А.Н. PHP для начинающих [Текст] / А.Н. Жадаев. – М., 2014. – 288 с.
25. ИТ-аутсорсинг (рынок России) [Электронный ресурс] – Режим доступа: [http://tadviser.ru/index.php/Статья:ИТ-аутсорсинг_\(рынок_России\)](http://tadviser.ru/index.php/Статья:ИТ-аутсорсинг_(рынок_России)) (дата обращения 20.12.2017 г.)
26. Когаловский, М.Р. Технология баз данных на персональных ЭВМ [Текст] / М.Р. Когаловский. – М.: Финансы и статистика, 2013. – 123 с.

27. Методология IDEF0 [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/IDEF0> (дата обращения 18.12.2017 г.)

28. Норенков, И.П. Автоматизированные информационные системы: Учебное пособие [Текст] / И.П. Норенков. – М.: МГТУ им. Баумана, 2013. – 342 с.

29. Российская государственная библиотека [Электронный ресурс] – Режим доступа: <http://www.rsl.ru/> (дата обращения 24.12.2017)

30. Федорова, Г.Н. Информационные системы: Учебник для студ. учреждений сред. проф. образования [Текст] / Г.Н. Федорова. – М.: ИЦ Академия, 2013. – 208 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А
Техническое задание

INIITE

«УТВЕРЖДАЮ»

Генеральный директор

ООО «ИНЛАЙТ»

/ М. Л. Кутасин

«13» декабря 2017 г.



ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку информационного сопровождения деятельности
сотрудников ООО «ИНЛАЙТ»

«СОГЛАСОВАНО»

Руководитель отдела разработки

ООО «ИНЛАЙТ»

/ А. В. Завтраков

«12» декабря 2017 г.

Исполнитель:

студент 4-го курса очной формы

обучения НИУ «БелГУ»

/ Морозов Ф.

МОСКВА 2017

1 Введение

Организация ООО «ИНЛАЙТ» зарегистрирована 18 октября 2006 года по адресу 115612, г Москва, Южный административный округ, район Братеево, ул Борисовские Пруды, д 24/2. ОГРН: 1067758221740, ИНН: 7724594030, КПП: 772401001.

Основным видом деятельности организации согласно Общероссийскому классификатору видов экономической деятельности (ОКВЭД) является: 62.01 «Разработка компьютерного программного обеспечения».

ООО «ИНЛАЙТ» тесно сотрудничает с коммерческими, государственными и бюджетными организациями на территории города Москвы и Московской области. Сотрудничество заключается в разработке B2B и B2C программных продуктов.

2 Основание для разработки

На данный момент все процессы по проектному управлению в ООО «ИНЛАЙТ» происходят в ручном режиме. Отсутствует формализованная отчётность сотрудников о выполненных ими задачах, а также не проводится анализ деятельности сотрудников на основе данных отчётов. На обработку информации генеральным директором ООО «ИНЛАЙТ» затрачиваются существенные временные ресурсы. В связи с этим необходимо разработать информационное сопровождение деятельности сотрудников отдела разработки.

3 Назначение разработки

В назначение разработки входят следующие цели:

- хранение информации о деятельности сотрудников отдела разработки;
- формирование отчётов эффективности деятельности сотрудников;

- организация комфортного функционирования для взаимодействия всех участников проектной деятельности;
- централизованное хранение обсуждения проектной деятельности.

4 Требования к программному изделию

4.1 Требования к функциональным характеристикам

Программный продукт должен включать в себя ввод, вывод, редактирование, удаление и хранение следующей информации:

- информация о задачах;
- информация о проектах;
- информация о сотрудниках;
- информация об отчётах.

Входной информацией является:

- описание и необходимый сроки выполнения задач;
- тип и период формирования отчётов эффективности;
- исследуемый объект отчётов.

Выходной информацией является:

- документ, содержащий отдельный отчёт эффективности в электронном виде;
- соответствующее электронное письмо.

4.2 Требования к программным средствам

В качестве основного инструмента разработки информационного сопровождения необходимо использовать микро-фреймворк Flight версии 1.3.6 на основе PHP версии 7.2.5 и систему управления базами данных MySQL версии 8.0.11.

4.3 Требования к надёжности

Информационное сопровождение должно:

- проводить контроль вводимой информации;
- блокировать некорректные действия пользователя при работе с интерфейсом программного продукта;
- обеспечивать целостность данных.

Один раз в неделю должно осуществляться резервное копирование базы данных информационного сопровождения. В случае сбоя в работе технических средств или иных исключительных ситуаций система должна быть восстановлена из последней актуальной резервной копии, а подразделение, обеспечивающее поддержку, должно заранее информировать всех пользователей о временной приостановке работы в системе.

4.4 Условия эксплуатации

Пользователя информационного сопровождения будут являться сотрудники отдела разработки. Интерфейс программного продукта должен быть максимально приближен к визуальному оформлению подобных систем. Ввод информации должен осуществляться в наиболее унифицированных формах.

4.5 Требования к маркировке и упаковке

Диски с эталонным и рабочим экземплярами программы должны иметь маркировку, состоящую из надписи "резерв" или "рабочая", даты последней перезаписи программы. Упаковка должна соответствовать условиям хранения диска. На упаковке должны быть указаны условия транспортирования и хранения диска.

5 Требования к программной документации

Программная документация должна содержать ряд документов (см. ГОСТ 19.101-77):

Программные документы:

- Спецификация (ГОСТ 19.202-78);
- Текст программы (ГОСТ 19.401-78);
- Описание программы (ГОСТ 19.402-78);
- Пояснительная записка (ГОСТ 19.404-79);

Эксплуатационные документы:

- Ведомость эксплуатационных документов (ГОСТ 19.507-79);
- Формуляр (ГОСТ 19.501-78);
- Описание применения (ГОСТ 19.502-78);
- Руководство системного программиста (ГОСТ 19.503-79);
- Руководство программиста (ГОСТ 19.504-79);
- Руководство оператора (ГОСТ 19.505-79);

Требования к перечисленным документам не отличаются от требований, определенных в Единой системе программной документации (ЕСПД).

6 Порядок контроля и приёмки

Приемка работы осуществляется по результатам приемо-сдаточных испытаний проводимых в присутствии представителей заказчика в соответствие с программой и методикой испытаний на тестовых данных, подготовленных заказчиком.

ПРИЛОЖЕНИЕ Б

Набор SQL-запросов для создания пустой базы данных

```
CREATE TABLE `Employees` (  
    `id_employee` INT NOT NULL AUTO_INCREMENT,  
    `id_post` INT NOT NULL,  
    `f_name` varchar(255) NOT NULL,  
    `l_name` varchar(255) NOT NULL,  
    `p_name` varchar(255),  
    `birthday` DATE NOT NULL,  
    `username` varchar(255) NOT NULL,  
    `password` varchar(255) NOT NULL,  
    PRIMARY KEY (`id_employee`)  
);  
  
CREATE TABLE `Posts` (  
    `id_post` INT NOT NULL AUTO_INCREMENT,  
    `title` varchar(255) NOT NULL,  
    PRIMARY KEY (`id_post`)  
);  
  
CREATE TABLE `Tasks` (  
    `id_Task` INT NOT NULL AUTO_INCREMENT,  
    `id_Author` INT NOT NULL,  
    `id_Executor` INT NOT NULL,  
    `id_Project` INT NOT NULL,  
    `description` TEXT NOT NULL,  
    `deadline` TIMESTAMP NOT NULL,  
    PRIMARY KEY (`id_Task`)  
);  
  
CREATE TABLE `TasksComments` (  
    `id_taskcomment` INT NOT NULL AUTO_INCREMENT,  
    `id_task` INT NOT NULL,  
    `id_comment` INT NOT NULL,  
    PRIMARY KEY (`id_taskcomment`)  
);  
  
CREATE TABLE `Comments` (  
    `id_comment` INT NOT NULL AUTO_INCREMENT,  
    `id_employee` INT NOT NULL,  
    `description` TEXT NOT NULL,  
    `time` TIMESTAMP NOT NULL,  
    PRIMARY KEY (`id_comment`)  
);  
  
CREATE TABLE `Transactions` (  
    `id_transaction` INT NOT NULL AUTO_INCREMENT,  
    `id_employee` INT NOT NULL,  
    `id_task` INT NOT NULL,
```

```

        `id_status` INT NOT NULL,
        `time` TIMESTAMP NOT NULL,
        PRIMARY KEY (`id_transaction`)
    );

```

```

CREATE TABLE `Statuses` (
    `id_status` INT NOT NULL AUTO_INCREMENT,
    `title` varchar(255) NOT NULL,
    PRIMARY KEY (`id_status`)
);

```

```

CREATE TABLE `TasksAttaches` (
    `id_taskattach` INT NOT NULL AUTO_INCREMENT,
    `id_task` INT NOT NULL,
    `id_atach` INT NOT NULL,
    PRIMARY KEY (`id_taskattach`)
);

```

```

CREATE TABLE `Attaches` (
    `id_atach` INT NOT NULL AUTO_INCREMENT,
    `path` varchar(255) NOT NULL,
    PRIMARY KEY (`id_atach`)
);

```

```

CREATE TABLE `CommentsAttaches` (
    `id_commentattach` INT NOT NULL AUTO_INCREMENT,
    `id_attach` INT NOT NULL,
    `id_comment` INT NOT NULL,
    PRIMARY KEY (`id_commentattach`)
);

```

```

CREATE TABLE `Reports` (
    `id_report` INT NOT NULL AUTO_INCREMENT,
    `id_employee` INT NOT NULL,
    `path` varchar(255) NOT NULL,
    `time` TIMESTAMP NOT NULL,
    PRIMARY KEY (`id_report`)
);

```

```

CREATE TABLE `Projects` (
    `id_project` INT NOT NULL,
    `title` varchar(255) NOT NULL,
    PRIMARY KEY (`id_project`)
);

```

```

ALTER TABLE `Employees` ADD CONSTRAINT `Employees_fk0` FOREIGN KEY
(`id_post`) REFERENCES `Posts`(`id_post`);
ALTER TABLE `Tasks` ADD CONSTRAINT `Tasks_fk0` FOREIGN KEY (`id_Author`)
REFERENCES `Employees`(`id_employee`);
ALTER TABLE `Tasks` ADD CONSTRAINT `Tasks_fk1` FOREIGN KEY
(`id_Executor`) REFERENCES `Employees`(`id_employee`);
ALTER TABLE `Tasks` ADD CONSTRAINT `Tasks_fk2` FOREIGN KEY (`id_Project`)

```

```

REFERENCES `Projects`(`id_project`);
    ALTER TABLE `TasksComments` ADD CONSTRAINT `TasksComments_fk0`
FOREIGN KEY (`id_task`) REFERENCES `Tasks`(`id_Task`);
    ALTER TABLE `TasksComments` ADD CONSTRAINT `TasksComments_fk1`
FOREIGN KEY (`id_comment`) REFERENCES `Comments`(`id_comment`);
    ALTER TABLE `Comments` ADD CONSTRAINT `Comments_fk0` FOREIGN KEY
(`id_employee`) REFERENCES `Employees`(`id_employee`);
    ALTER TABLE `Transactions` ADD CONSTRAINT `Transactions_fk0` FOREIGN KEY
(`id_employee`) REFERENCES `Employees`(`id_employee`);
    ALTER TABLE `Transactions` ADD CONSTRAINT `Transactions_fk1` FOREIGN KEY
(`id_task`) REFERENCES `Tasks`(`id_Task`);
    ALTER TABLE `Transactions` ADD CONSTRAINT `Transactions_fk2` FOREIGN KEY
(`id_status`) REFERENCES `Statuses`(`id_status`);
    ALTER TABLE `TasksAttaches` ADD CONSTRAINT `TasksAttaches_fk0` FOREIGN
KEY (`id_task`) REFERENCES `Tasks`(`id_Task`);
    ALTER TABLE `TasksAttaches` ADD CONSTRAINT `TasksAttaches_fk1` FOREIGN
KEY (`id_attach`) REFERENCES `Attaches`(`id_attach`);
    ALTER TABLE `CommentsAttaches` ADD CONSTRAINT `CommentsAttaches_fk0`
FOREIGN KEY (`id_attach`) REFERENCES `Attaches`(`id_attach`);
    ALTER TABLE `CommentsAttaches` ADD CONSTRAINT `CommentsAttaches_fk1`
FOREIGN KEY (`id_comment`) REFERENCES `Comments`(`id_comment`);
    ALTER TABLE `Reports` ADD CONSTRAINT `Reports_fk0` FOREIGN KEY
(`id_employee`) REFERENCES `Employees`(`id_employee`);

```

CSS-стили визуального интерфейса

```

html, body, body div, span, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr,
address, cite, code, del, dfn, em, img, ins, kbd, q, samp, small, strong, sub, sup, var, b, i, dl, dt, dd, ol,
ul, li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article, aside, figure,
footer, header, hgroup, menu, nav, section, time, mark, audio, video { margin: 0; padding: 0; border:
0; outline: 0; font-size: 100%; vertical-align: baseline; background: transparent; }
    article, aside, figure, footer, header, hgroup, nav, section { display: block; }
    header, section, footer { clear: both; }
    html { overflow-y: scroll; }
    ul { list-style: outside disc; margin: 15px 0 15px 30px; }
    ol { list-style: outside decimal; margin: 15px 0 15px 30px; }
    table { border-collapse: collapse; border-spacing: 0; }
    td, td img { vertical-align: top; }
    button, input, select, textarea { margin: 0; outline: none; }
    input::-moz-focus-inner, button::-moz-focus-inner { padding: 0; border: 0; }
    input, select { vertical-align: middle; }
    input[type=radio], input[type=checkbox] { margin: 1px 4px 4px 0; }
    input[type=text], input[type=password] { -webkit-border-radius: 0; border-radius: 0; -
webkit-appearance: none; }
    textarea { overflow: auto; resize: vertical; -webkit-border-radius: 0; border-radius: 0; -
webkit-appearance: none; }
    button, input[type=submit] { width: auto; overflow: visible; }
    .ie7 input[type=submit], .ie7 input[type=reset], .ie7 input[type=button] { -webkit-filter:

```

```

chroma(color=#000000); filter: chroma(color=#000000); }
  input[type=button], input[type=submit], button { cursor: pointer; -webkit-border-radius: 0;
border-radius: 0; -webkit-appearance: none; }
  blockquote, q { quotes: none; }
  blockquote:before, blockquote:after, q:before, q:after { content: ""; content: none; }
  del { text-decoration: line-through; }
  abbr[title], dfn[title] { border-bottom: 1px dotted #000; cursor: help; }
  small { font-size: 85%; }
  strong { font-weight: bold; }
  sub, sup { font-size: 75%; line-height: 0; position: relative; }
  sup { top: -0.5em; }
  sub { bottom: -0.25em; }
  hr { display: block; height: 1px; border: 0; border-top: 1px solid #ccc; margin: 1em 0;
padding: 0; font-size: 0; line-height: 0; }
  pre, code, kbd, samp { font-family: monospace, sans-serif; }
  pre { white-space: pre; white-space: pre-wrap; white-space: pre-line; word-wrap: break-
word; }
  @font-face { font-family: 'Open Sans'; font-style: normal; font-weight: 300; src: local('Open
Sans
  Light'),
  local('OpenSans-Light'),
url(https://fonts.gstatic.com/s/opensans/v15/mem5YaGs126MiZpBA-UN_r8OUuhs.ttf)
format('truetype'); }
  @font-face { font-family: 'Open Sans'; font-style: normal; font-weight: 400; src: local('Open
Sans
  Regular'),
  local('OpenSans-Regular'),
url(https://fonts.gstatic.com/s/opensans/v15/mem8YaGs126MiZpBA-UFVZ0e.ttf)
format('truetype'); }
  @font-face { font-family: 'Open Sans'; font-style: normal; font-weight: 600; src: local('Open
Sans
  SemiBold'),
  local('OpenSans-SemiBold'),
url(https://fonts.gstatic.com/s/opensans/v15/mem5YaGs126MiZpBA-UNirkOUuhs.ttf)
format('truetype'); }
  @font-face { font-family: 'Open Sans'; font-style: normal; font-weight: 700; src: local('Open
Sans
  Bold'),
  local('OpenSans-Bold'),
url(https://fonts.gstatic.com/s/opensans/v15/mem5YaGs126MiZpBA-UN7rgOUuhs.ttf)
format('truetype'); }
  html, body { overflow: hidden; width: 100vw; height: 100vh; }
  html { min-width: auto; width: auto; height: 100%; }
  body { font: normal normal 14px/20px 'Open Sans', arial, sans-serif; background: #F1F1F2;
color: #333333; -webkit-text-size-adjust: 100%; -webkit-font-smoothing: antialiased; -webkit-box-
sizing: border-box; box-sizing: border-box; min-width: 1200px; min-height: initial; position: relative;
top: auto; right: auto; bottom: auto; left: auto; }
  body input, body textarea, body button { font-family: 'Open Sans', arial, sans-serif; }
  body *, body *:before, body *:after { -webkit-box-sizing: border-box; box-sizing: border-
box; }
  .left { float: left; }
  .right { float: right; }
  .clearfix { *zoom: 1; }
  .clearfix:after { clear: both; }
  .clearfix:before, .clearfix:after { content: " "; display: table; }
  a, button { -webkit-transition: 300ms all ease; -o-transition: 300ms all ease; transition:
300ms all ease; }
  .btn { background: #3498DB; color: #ffffff; text-decoration: none; padding: 10px 15px; -
webkit-border-radius: 4px; border-radius: 4px; text-align: center; overflow: hidden; border: none;
display: inline-block; vertical-align: top; position: relative; top: auto; right: auto; bottom: auto; left:

```

```

auto; font-size: 14px; line-height: 20px; font-weight: 600; }
    .btn i { display: inline-block; vertical-align: top; font-size: 16px; line-height: inherit; }
    .btn.with-icon i, .btn.with-icon-right i { background: #2980B9; display: block; width: 40px;
height: 40px; font-size: 16px; line-height: 40px; }
    .btn:hover { background: #2980B9; }
    .btn.with-icon { padding-left: 55px; }
    .btn.with-icon i { position: absolute; top: 0; right: auto; bottom: auto; left: 0; }
    .btn.with-icon-right { padding-right: 55px; }
    .btn.with-icon-right i { position: absolute; top: 0; right: 0; bottom: auto; left: auto; }
    .btn.is-green { background: #1ABC9C; }
    .btn.is-green.with-icon i, .btn.is-green.with-icon-right i { background: #16A085; }
    .btn.is-green:hover { background: #16A085; }
    .btn.is-grey { background: #95A5A6; }
    .btn.is-grey.with-icon i, .btn.is-grey.with-icon-right i { background: #7F8C8D; }
    .btn.is-grey:hover { background: #7F8C8D; }
    .btn.is-block { display: block; }
    .container { padding: 0 30px; }
    .page { width: 100vw; height: 100vh; }
    .p-side { z-index: 1; position: absolute; top: 0; right: auto; bottom: 0; left: 0; width: 260px;
height: auto; }
    .p-logo { border-bottom: 1px solid rgba(0, 0, 0, 0.15); padding: 0 0 0 58px; margin: 0 0
15px; display: block; position: relative; top: auto; right: auto; bottom: auto; left: auto; font-size: 18px;
line-height: 60px; font-weight: 600; }
    .p-logo img { -webkit-border-radius: 6px; border-radius: 6px; margin: -16px 0 0; display:
block; position: absolute; top: 50%; right: auto; bottom: auto; left: 15px; width: 32px; height: 32px;
}
    .p-nav ul { list-style: none; margin: 0; padding: 0; }
    .p-nav li { display: block; }
    .p-nav a { padding: 0 0 0 58px; text-decoration: none; color: #333333; display: block; font-
size: 15px; line-height: 46px; position: relative; top: auto; right: auto; bottom: auto; left: auto; }
    .p-nav a:hover { background: rgba(0, 0, 0, 0.03); }
    .p-nav i { text-align: center; display: block; font-size: 18px; line-height: inherit; position:
absolute; top: 0; right: auto; bottom: auto; left: 0; width: 58px; height: auto; }
    .p-nav span { background: #3498DB; -webkit-border-radius: 8px; border-radius: 8px;
padding: 0 8px 2px; color: #ffffff; margin: -12px 0 0; display: block; position: absolute; top: 50%;
right: 25px; bottom: auto; left: auto; font-size: 13px; line-height: 22px; font-weight: 700; }
    .p-nav li.is-active a { background: #3498DB; color: #ffffff; font-weight: 600; cursor: default;
}
    .p-nav li.is-active a span { background: #ffffff; color: #3498DB; }
    .p-user { border-top: 1px solid rgba(0, 0, 0, 0.15); padding: 0 0 0 58px; display: block;
position: absolute; top: auto; right: 0; bottom: 0; left: 0; font-size: 14px; line-height: 60px; }
    .p-user i { text-align: center; display: block; font-size: 18px; line-height: inherit; position:
absolute; top: 0; right: auto; bottom: auto; left: 0; width: 58px; height: auto; }
    .p-logout { color: #333333; text-decoration: none; text-align: center; border-left: 1px solid
rgba(0, 0, 0, 0.15); display: block; width: 58px; height: 60px; position: absolute; top: auto; right:
10px; bottom: 0; left: auto; }
    .p-logout:hover { background: rgba(0, 0, 0, 0.03); }
    .p-logout i { display: block; font-size: 18px; line-height: 60px; }
    .p-box { z-index: 2; background: #ffffff; -webkit-border-radius: 5px 0 0 5px; border-radius:
5px 0 0 5px; -webkit-box-shadow: 0 2px 10px 0 rgba(0, 0, 0, 0.25); box-shadow: 0 2px 10px 0 rgba(0,
0, 0, 0.25); position: absolute; top: 0; right: 0; bottom: 0; left: 250px; }
    .p-header { border-bottom: 1px solid rgba(0, 0, 0, 0.3); position: relative; top: auto; right:

```

```

auto; bottom: auto; left: auto; }
.p-header--title { padding: 15px 0; font-size: 24px; line-height: 30px; font-weight: 700; }
.p-header--btn { position: absolute; top: 10px; right: 15px; bottom: auto; left: auto; }
.p-content { overflow: auto; width: 100%; height: -webkit-calc(100vh - 61px); height:
calc(100vh - 61px); }
.p-table { table-layout: fixed; width: 100%; height: auto; }
.p-table th { color: rgba(51, 51, 51, 0.7); border-bottom: 1px solid rgba(0, 0, 0, 0.15);
padding: 20px 15px 15px; text-align: left; font-size: 15px; line-height: 20px; font-weight: 400; }
.p-table th:first-child { padding-left: 30px; }
.p-table th:last-child { padding-right: 30px; }
.p-table tbody tr { cursor: pointer; }
.p-table tbody tr:hover { background: rgba(0, 0, 0, 0.03); }
.p-table td { padding: 15px 15px; border-bottom: 1px solid rgba(0, 0, 0, 0.15); font-size:
14px; line-height: 20px; font-weight: 400; }
.p-table td:first-child { padding-left: 30px; }
.p-table td:last-child { padding-right: 30px; }
.p-table th .l-status, .p-table td .l-status { margin: -1px 0; }
.p-table th .p-employee, .p-table td .p-employee { margin: -3px 0; }
.p-table.is-tasks th:nth-child(1) { width: 100px; height: auto; }
.p-table.is-tasks th:nth-child(2) { width: 120px; height: auto; }
.p-table.is-tasks th:nth-child(3) { width: 110px; height: auto; }
.p-table.is-tasks th:nth-child(4) { width: -webkit-calc(100% - 100px - 120px - 110px -
230px); width: calc(100% - 100px - 120px - 110px - 230px); height: auto; }
.p-table.is-tasks th:nth-child(5) { width: 230px; height: auto; }
.p-table.is-projects th:nth-child(1) { width: 100px; height: auto; }
.p-table.is-projects th:nth-child(2) { width: -webkit-calc(100% - 100px - 120px - 120px -
120px - 120px); width: calc(100% - 100px - 120px - 120px - 120px - 120px); height: auto; }
.p-table.is-projects th:nth-child(3) { width: 120px; height: auto; }
.p-table.is-projects th:nth-child(4) { width: 120px; height: auto; }
.p-table.is-projects th:nth-child(5) { width: 120px; height: auto; }
.p-table.is-projects th:nth-child(6) { width: 120px; height: auto; }
.p-table.is-reports th:nth-child(1) { width: 100px; height: auto; }
.p-table.is-reports th:nth-child(2) { width: 180px; height: auto; }
.p-table.is-reports th:nth-child(3) { width: -webkit-calc(100% - 100px - 180px); width:
calc(100% - 100px - 180px); height: auto; }
.p-table.is-employees th:nth-child(1) { width: 100px; height: auto; }
.p-table.is-employees th:nth-child(2) { width: 180px; height: auto; }
.p-table.is-employees th:nth-child(3) { width: 180px; height: auto; }
.p-table.is-employees th:nth-child(4) { width: 180px; height: auto; }
.p-table.is-employees th:nth-child(5) { width: 250px; height: auto; }
.p-table.is-employees th:nth-child(6) { width: -webkit-calc(100% - 100px - 180px - 180px -
180px - 250px); width: calc(100% - 100px - 180px - 180px - 180px - 250px); height: auto; }
.p-table.is-rating th:nth-child(1) { width: 100px; height: auto; }
.p-table.is-rating th:nth-child(2) { width: 100px; height: auto; }
.p-table.is-rating th:nth-child(3) { width: -webkit-calc(100% - 100px - 100px); width:
calc(100% - 100px - 100px); height: auto; }
.p-employee { color: #333333; text-decoration: none; white-space: nowrap; overflow:
hidden; -o-text-overflow: ellipsis; text-overflow: ellipsis; -webkit-border-radius: 4px; border-radius:
4px; color: #ffffff; padding: 3px 8px 3px 32px; display: inline-block; vertical-align: top; font-size:
12px; line-height: 20px; font-weight: 700; position: relative; top: auto; right: auto; bottom: auto; left:
auto; }
.p-employee i { text-align: center; display: block; position: absolute; top: 0; right: auto;

```

```

bottom: auto; left: 0; width: 26px; height: 26px; font-size: 12px; line-height: 26px; }
.p-employee.is-green { background: #1ABC9C; }
.p-employee.is-green i { background: #16A085; }
.p-employee.is-purple { background: #9B59B6; }
.p-employee.is-purple i { background: #8E44AD; }
.p-employee.is-orange { background: #E67E22; }
.p-employee.is-orange i { background: #D35400; }
.p-employee.is-grey { background: #95A5A6; }
.p-employee.is-grey i { background: #7F8C8D; }
.p-mono { white-space: nowrap; overflow: hidden; -o-text-overflow: ellipsis; text-overflow:
ellipsis; font-family: 'Courier', monospace; }
.t-ellipsis { white-space: nowrap; overflow: hidden; -o-text-overflow: ellipsis; text-
overflow: ellipsis; }
.l-status { -webkit-border-radius: 4px; border-radius: 4px; padding: 1px 6px; color: #ffffff;
text-transform: uppercase; white-space: nowrap; display: inline-block; vertical-align: top; }
.l-status:before { display: block; font-size: 12px; line-height: 20px; font-weight: 700; }
.l-status.is-todo { background: #7F8C8D; }
.l-status.is-todo:before { content: 'Ожидает'; }
.l-status.is-progress { background: #F39C12; }
.l-status.is-progress:before { content: 'В процессе'; }
.l-status.is-done { background: #27AE60; }
.l-status.is-done:before { content: 'Готово'; }
.l-status.is-needinfo { background: #C0392B; }
.l-status.is-needinfo:before { content: 'Вопрос'; }
.l-priority { font-size: 0; line-height: 0; }
.l-priority i { margin: 0 6px 0 0; display: inline-block; vertical-align: top; font-size: 14px
!important; line-height: 20px !important; }
.l-priority:after { display: inline-block; vertical-align: top; font-size: 12px; line-height: 20px;
font-weight: 700; }
.l-priority.is-low { color: #7F8C8D; }
.l-priority.is-low:after { content: 'Обычный'; }
.l-priority.is-high { color: #C0392B; }
.l-priority.is-high:after { content: 'Высокий'; }
.report { padding: 0 0 50px; overflow: auto; width: 100vw; height: 100vh; }
.report--in { margin: 20px auto; -webkit-border-radius: 5px; border-radius: 5px;
background: #ffffff; -webkit-box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1); box-shadow: 0 0 20px 0
rgba(0, 0, 0, 0.1); overflow: hidden; width: 960px; height: auto; }
.report--in.is-tools { background: none; -webkit-box-shadow: none; box-shadow: none;
*zoom: 1; }
.report--in.is-tools:after { clear: both; }
.report--in.is-tools:before, .report--in.is-tools:after { content: " "; display: table; }
.report--head { background: #8E44AD; color: #ffffff; padding: 15px 30px; font-size: 16px;
line-height: 30px; *zoom: 1; }
.report--head:after { clear: both; }
.report--head:before, .report--head:after { content: " "; display: table; }
.report--title { float: left; }
.report--date { float: right; }
.report--group { border-bottom: 1px solid rgba(0, 0, 0, 0.15); padding: 15px 30px; }
.report--body { padding: 25px 30px; }
.report--charts { margin: 0 0 -40px; font-size: 0; line-height: 0; }
.report--chart { margin: 0 0 40px; display: inline-block; vertical-align: top; width: 50%;
height: auto; }

```

```

.report--chart.is-full { width: 100%; height: auto; }
.report--chart canvas { display: block; }
.report--fields { font-size: 0; line-height: 0; }
.report--field { display: inline-block; vertical-align: top; width: 33.33333333%; height: auto;
font-size: 15px; line-height: 24px; }
.report--field.is-quattro { width: 25%; height: auto; }
.report--field.is-half { width: 50%; height: auto; }
.report--field .l-status { margin: 2px 10px 0 0; }
.report--btn { -webkit-box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1); box-shadow: 0 0 20px 0
rgba(0, 0, 0, 0.1); }
.report--btn.is-left { float: left; }
.report--btn.is-right { float: right; }
.report.is-small .report--in { width: 600px; height: auto; }
.auth { position: relative; top: auto; right: auto; bottom: auto; left: auto; width: 100vw;
height: 100vh; }
.auth--in { -webkit-border-radius: 5px; border-radius: 5px; background: #ffffff; -webkit-
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1); box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1); -webkit-
transform: translate(-50%, -50%); -ms-transform: translate(-50%, -50%); transform: translate(-50%,
-50%); position: absolute; top: 50%; right: auto; bottom: auto; left: 50%; width: 300px; height: auto;
}
.auth--body { padding: 4px 20px 20px; }
.auth--field { margin: 0 0 20px; position: relative; top: auto; right: auto; bottom: auto; left:
auto; }
.auth--field i { text-align: center; position: absolute; top: 0; right: auto; bottom: auto; left: 0;
width: 50px; height: auto; font-size: 16px; line-height: 40px; }
.auth--field input { -webkit-border-radius: 5px; border-radius: 5px; border: 1px solid rgba(0,
0, 0, 0.15); padding: 0 15px 0 50px; display: block; width: 100%; height: 40px; font-size: 15px; line-
height: 20px; }
.modals { z-index: 5000; display: none; position: fixed; top: 0; right: 0; bottom: 0; left: 0; }
.modals--overlay { z-index: 1; background: rgba(0, 0, 0, 0.9); position: absolute; top: 0;
right: 0; bottom: 0; left: 0; }
.modals--item { margin: 0 0 0 -270px; -webkit-border-radius: 5px; border-radius: 5px;
background: #ffffff; -webkit-box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.1); box-shadow: 0 0 20px 0
rgba(0, 0, 0, 0.1); z-index: 2; display: none; width: 540px; height: auto; position: absolute; top: 80px;
right: auto; bottom: auto; left: 50%; }
.modals--head { padding: 8px 25px 12px; border-bottom: 1px solid rgba(0, 0, 0, 0.15); font-
size: 22px; line-height: 40px; font-weight: 700; position: relative; top: auto; right: auto; bottom: auto;
left: auto; }
.modals--close { color: #333333; text-decoration: none; border-left: 1px solid rgba(0, 0, 0,
0.15); text-align: center; display: block; position: absolute; top: 0; right: 0; bottom: auto; left: auto;
width: 60px; height: 60px; }
.modals--close i { display: block; font-size: 24px; line-height: 60px; }
.modals--body { padding: 20px 25px; overflow: auto; max-width: initial; max-height: -
webkit-calc(100vh - 80px - 61px - 71px - 80px); max-height: calc(100vh - 80px - 61px - 71px -
80px); }
.modals--field { margin: 0 0 20px; padding: 0 0 0 150px; position: relative; top: auto; right:
auto; bottom: auto; left: auto; }
.modals--field:last-child { margin: 0; }
.modals--field input[type="text"], .modals--field input[type="password"], .modals--field
input[type="date"] { -webkit-border-radius: 5px; border-radius: 5px; border: 1px solid rgba(0, 0, 0,
0.15); padding: 0 15px; display: block; width: 100%; height: 40px; font-size: 15px; line-height: 20px;
}

```

```

.modals--field select { padding: 8px 15px 12px; border: 1px solid rgba(0, 0, 0, 0.15);
background: #ffffff; display: block; width: 100%; height: 40px; font-size: 15px; line-height: 20px; }
.modals--field textarea { -webkit-border-radius: 5px; border-radius: 5px; border: 1px solid
rgba(0, 0, 0, 0.15); padding: 8px 15px 12px; display: block; width: 100%; height: auto; min-width:
initial; min-height: 160px; font-size: 15px; line-height: 20px; }
.modals--label { position: absolute; top: 0; right: auto; bottom: auto; left: 0; font-size: 14px;
line-height: 40px; }
.modals--foot { padding: 15px 25px; border-top: 1px solid rgba(0, 0, 0, 0.15); *zoom: 1; }
.modals--foot:after { clear: both; }
.modals--foot:before, .modals--foot:after { content: " "; display: table; }
.modals--btn { float: right; margin: 0 0 20px; }
.task { width: auto; height: 100%; display: -webkit-box; display: -webkit-flex; display: -ms-
flexbox; display: flex; }
.task--info { padding: 20px 30px; width: 55%; height: auto; }
.task--field { margin: 0 0 15px; -webkit-box-align: center; -webkit-align-items: center; -ms-
flex-align: center; align-items: center; display: -webkit-box; display: -webkit-flex; display: -ms-
flexbox; display: flex; }
.task--key { font-weight: 700; width: 25%; height: auto; }
.task--value { width: 75%; height: auto; }
.task--description { margin: 30px 0; }
.task--description .task--key, .task--attachments .task--key, .task--description .task--value,
.task--attachments .task--value { width: auto; height: auto; }
.task--description .task--key, .task--attachments .task--key { margin: 0 0 5px; }
.task--attachments .task--value a { margin: 0 10px 10px 0; }
.task--comments { background: rgba(0, 0, 0, 0.05); -webkit-box-shadow: 0 0 20px 0 rgba(0,
0, 0, 0.15); box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.15); padding: 15px; -webkit-box-orient: vertical;
-webkit-box-direction: normal; -webkit-flex-direction: column; -ms-flex-direction: column; flex-
direction: column; width: 45%; height: auto; }
.task--list { background: #ffffff; -webkit-border-radius: 4px; border-radius: 4px; overflow:
auto; width: auto; height: -webkit-calc(100% - 85px); height: calc(100% - 85px); }
.task--comment { padding: 10px 15px 5px; border-top: 1px solid rgba(0, 0, 0, 0.15); }
.task--comment:first-child { border: none; }
.task--tools { margin: 0 0 5px; font-size: 13px; line-height: 20px; *zoom: 1; }
.task--tools:after { clear: both; }
.task--tools:before, .task--tools:after { content: " "; display: table; }
.task--author { float: left; }
.task--date { margin: 0 0 15px; float: left; opacity: .60; }
.task--text { margin: 0 0 10px; }
.task--form { padding: 15px 0 0; display: -webkit-box; display: -webkit-flex; display: -ms-
flexbox; display: flex; width: auto; height: 85px; }
.task--textarea { -webkit-box-flex: 1; -webkit-flex-grow: 1; -ms-flex-positive: 1; flex-grow:
1; padding: 0 10px 0 0; }
.task--textarea textarea { resize: none; -webkit-border-radius: 4px; border-radius: 4px;
border: 1px solid rgba(0, 0, 0, 0.3); padding: 8px 15px; display: block; width: 100%; height: 70px;
font-size: 14px; line-height: 20px; }

```

ПРИЛОЖЕНИЕ В

Акт о внедрении информационного сопровождения

INIITE

«УТВЕРЖДАЮ»

Генеральный директор



М. Л. Кутасин

АКТ

о внедрении результатов, полученных в ходе выполнения Выпускной квалификационной работы

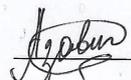
на тему «Разработка информационного сопровождения деятельности сотрудников организации (на примере ООО «ИНЛАЙТ»)), по направлению подготовки 09.03.03 «Прикладная информатика», по профилю «Прикладная информатика в экономике» студента Морозова Ф.

Мы, нижеподписавшиеся представители ООО «ИНЛАЙТ», находящегося по адресу: Россия, г. Москва, ул. Борисовские Пруды, д. 24/2 (далее: Общество): Генеральный директор – М. Л. Кутасин, руководитель отдела разработки – А. В. Завтраков, составили настоящий акт о том, что разработанное информационное сопровождение деятельности сотрудников организации, в рамках выпускной квалификационной работы студента Морозова Ф., было внедрено в деятельность Общества.

Генеральный директор ООО «ИНЛАЙТ»

 / М.Л. Кутасин

Руководитель отдела разработки

 / А.В. Завтраков