МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра компьютерных систем

Игнатенко Василий Павлович

ПРОЕКТ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА РАБОТЫ ОБОРУДОВАНИЯ С РАЗМЕЩЕНИЕМ В ОБЛАЧНЫХ СЕРВИСАХ

БАКАЛАВРСКАЯ РАБОТА
по основной образовательной программе подготовки бакалавров
по направлению 09.03.02 – Информационные системы и технологии

г. Владивосток
2018

Студент гр. Б8418 _____
(подпись)
«_____»_____ 2018 г.

Научный руководитель к.ф.-м.н., доцент
(должность, ученое звание)
_____ Е.В. Пустовалов
(подпись) (и.о.ф)
«_____»_____ 2018 г.

Защищена в ГАК с оценкой _____

Секретарь ГАК

_____ _____
подпись И.О.Фамилия

«_____» _____ 2018 г.

«Допустить к защите»

Заведующий кафедрой _к.ф.-м.н., доцент___
(ученое звание)

_____ __Е.В. Пустовалов____
(подпись) (и. о.ф)

«_____»_____ 2018 г

Аннотация

Выпускная квалификационная работа на тему: «Проект автоматизированной системы учёта работы оборудования с размещением в облачных сервисах».

Автор: Игнатенко Василий Павлович, студент 4 курса.

Выпускная квалификационная работа 39 с., 3 ч., 14 табл., 4 рис., 13 источников.

КЛЮЧЕВЫЕ СЛОВА

Проект, расписание, .

Цель дипломной работы: создать проект автоматизированной системы учёта работы оборудования

Цель дипломной работы может быть достигнута путем решения следующих задач:

1. Проведение анализа предметной области;
2. Проектирование структуры базы данных (инфологическое и логическое моделирование);
3. Разработка проекта приложения.

Дипломная работа состоит из введения, трех глав и заключения.

Во введении обосновывается актуальность выбранной темы, формулируется цель, и ставятся основные задачи работы.

В первой главе выполняется обзор облачных и WEB сервисов, методы и этапы разработки проекта, технологии реализации.

Во второй главе проводится анализ предметной области и производится выбор методов разработки приложения и базы данных.

В третьей главе представлена реализация решения поставленных задач.

Заключение содержит выводы, основанные на полученных результатах.

Оглавление

Введение

Правильное распределение времени пользования оборудованием в исследовательских лабораториях является важным фактором для своевременного выполнения научно-исследовательских работ. Существует несколько способов решения этой проблемы. Первый вариант — использовать для составления графика пользования оборудованием бумажные носители. Плюсы такого подхода: простое решение, отсутствие необходимости навыков работы с какими-либо программными средствами. Минусы: необходимость заполнения таблицы всеми сотрудниками вручную, невозможность изменения данных (либо переделывание всей таблицы), хранение большого количества бумаги для отчётности.

Второй вариант — использовать средства офисных приложений. Например, вести таблицу в MS Excel, либо другом табличном редакторе. Плюсы: всё ещё простое решение, требующее только навыков в использовании табличного редактора, возможность использования с любого компьютера (использование онлайн редактора). Минусы: необходимость вводить все данные вручную, необходимость контроля вводимых данных, трудности при обновлении и отслеживании внесённых изменений.

Третий вариант — воспользоваться программой для автоматического учёта работы оборудования. Преимущества этого решения: автоматическое заполнение таблицы с расписанием, возможность отслеживать все изменения. Хранение сведений в формате базы данных значительно облегчает построение отчётов о проведённых работах. Доступ к системе с любого компьютера позволяет оперативно резервировать оборудование, либо, при необходимости, изменять или удалять записи. Трудностью этого подхода является необходимость покупки или разработки такого решения и внедрение его.

Целью данной выпускной квалификационной работы является разработка проекта автоматизированной системы учёта работы оборудования. Для упрощения обслуживания и поддержки системы её можно разместить в облачных сервисах, где данные заботы ложатся на плечи облачного провайдера.

В соответствии с поставленной целью в работе определены следующие задачи:

4. Проведение анализа предметной области;

5. Проектирование структуры базы данных (инфологическое и логическое моделирование);

6. Разработка проекта приложения.

Актуальность разработки автоматизированной системы учёта оборудования обусловлена тем, что в процессе функционирования исследовательской лаборатории сотрудникам приходится тратить время на составление расписание и согласование его с другими сотрудниками, что можно ускорить в рамках автоматизированной системы, а также обеспечить оперативную работу с отчётными документами.

1

1.1 Облачные сервисы и виды их предоставления

1.1.1 Понятие облачных вычислений

Под облачными вычислениями понимают подход оперативного предоставления удаленных IT-ресурсов (сетевая, аппаратная и программная инфраструктура) по первому запросу клиента, в необходимом объеме и при минимальном участии поставщика услуг [1].

Неотъемлемыми требованиями, предъявляемыми к вычислительным облакам, являются:

- доступность из любой точки мира при наличии интернета;
- непрерывность и надежность доступа к сервисам и данным;
- высокая пропускная способность и простота администрирования;
- возможность быстрого масштабирования объема IT-ресурсов;
- автоматический учет потребляемых мощностей;
- стоимость, зависящая от степени использования.

1.1.2 Характеристики и возможности облачных хранилищ

Основными характеристиками облачных хранилищ являются:

1. Объём памяти;
2. Скорость обмена данными между серверами клиента и облачным хранилищем;
3. Степень защищенности передаваемых и хранимых данных;
4. Возможность интеграции и совместимости облачного хранилища с различными программными аппаратными платформами и БД;
5. Тип организации облачного хранилища:
6. файловое хранилище (file storage option) — подходит для мультимедийных массивов или личных каталогов пользователей, крупных репозиториев контента, среды разработки,
7. объектное хранилище (object storage option) — подходит для импорта неструктурированных данных (например, медиафайлов и веб-материалов) из

других хранилищ с целью аналитики, миграции данных, резервного копирования, архивации,

8. блочное хранилище (block storage option) — подходит для обработки массивов финансовой и иной информации, требующей сложных вычислений, для размещения ERP-систем, а также для любых часто перезаписываемых данных,

9. агрегатное хранилище (aggregate storage option) — вариант блочного хранилища, подходит для компактного хранения данных, которые не требуют масштабных перезаписей;

10. Дополнительные функции:

11. автоматическая синхронизация данных, в том числе посредством WebDAV,

12. совместная работа над документами несколькими пользователями и автоматическое создание версий файлов,

13. разграничение прав/уровней доступа к файлам и папкам,

14. сохранение удаленных файлов и максимальные сроки их восстановления,

15. встроенные программные клиенты, модули и сервисы: мастер создания и редактирования документов, медиаплеер, CRM и ERP-модули, функции аналитики, почтовый клиент,

16. вычисления: виртуальные контейнеры, развертывание web-приложений, балансировка (load balancing) и автомасштабирование нагрузки (auto scaling), система управления событиями, структурирование и систематизация больших массивов разобщенной информации,

17. брендирование интерфейса,

18. техническая поддержка 24/7: по e-mail, телефону, в режиме реального времени по чату и пр.,

19. Гарантии отказоустойчивости и сохранности данных, подкрепленные финансовыми обязательствами — страхование ответственности провайдера перед клиентом.

В зависимости от комбинации и объема вышеперечисленных характеристик поставщики облачных сервисов формируют для клиентов свои стандартизированные или персонифицированные предложения.

1.1.3 Варианты развёртывания облачных хранилищ

В зависимости от нужд и финансовых возможностей компании облачные вычисления могут использоваться в различных вариантах развертывания:

– частные облака;
– публичные облака;
– гибридные облака.

Частное облако.

Пользованием частным облаком представляет собой аренду сервиса и инфраструктуры в условиях частной сети и полностью выделенного виртуального ресурса. Плюсами использования частного облака являются:

– возможность собственноручно управлять облачной средой с высокой доступностью к физическим ресурсам и настройкам виртуальной платформы;
– более высокая степень защиты данных за счет одиночного потребления сервиса в этой сети;
– более высокая производительность;
– эластичная и быстрая масштабируемость ресурсов.

Минусами частного облака считают существенные затраты на внедрение и обеспечение функционирования ресурса после запуска проекта. Кроме этого, даже у частного облака есть ограничение объема. Если на определенном этапе его ресурсов будет недостаточно, то возникнут новые траты. Еще один существенный минус — это риск утраты работоспособности ресурса. Поскольку управление инфраструктурой облака идет изнутри (т.е. силами самого клиента), то этот риск значительно выше, чем при работе с публичным облаком.

Исходя из описанных характеристик, частное облако может быть интересно представителям крупного бизнеса, которым важно сохранить максимальную безопасность и конфиденциальность.

Публичное облако.

Представляет собой аренду виртуальных сервисов и инфраструктуры поверх Глобальной сети. Поскольку затраты на обеспечение работы «облака» делятся на всех пользователей, то этот вариант хранилища является самым экономичным. Начать работу в публичном облаке может любое частное лицо или компания. Для этого необходимо лишь устройство с выходом в интернет. Подключение ресурсов и доступа происходит в течение нескольких минут. Вместе с этим, среди минусов публичного облака выделяют:

– отсутствие возможности клиента повлиять на работоспособность ресурса;

– зависимость работы в «облаке» от качества интернет-соединения;

– более низкая, по сравнению с частным облаком, защищенность данных.

Публичное облако оптимально подходит компаниям малого и среднего бизнеса, которые не имеют возможности содержать собственный IT-отдел. При этом сотрудники такой компании коллективно используют одни и те же приложения, например, электронную почту. Актуально публичное облако и для тех организаций, которые временно нуждаются в тестировании программы или приложения, для размещения интернет-проектов, аппликаций и как вспомогательный ресурс для реализации различных бизнес-задач.

Гибридное облако.

Современный вариант, объединяющий в одну сеть локальные ресурсы и публичный облачный сервис. Совокупность возможностей двух технологий превращает его в удобный и гибкий инструмент для различных целей. Гибридное облако позволяет клиенту не отказываться от работоспособного оборудования и сохранить за собой право самостоятельно управлять наиболее важными процессами.

Такая конфигурация является компромиссом для тех, кто сомневается в безопасности облачных сервисов: компании могут постепенно выводить из собственного управления наименее значимые элементы и одновременно оценивать преимущества облачного решения.

Гибридное облако удобно компаниям, нагрузка на IT-сервисы которых возникает волнообразно, например, при ведении сезонного бизнеса, а также при потребности одноразово протестировать программу или обработать большое количество данных. Кроме того, для бизнеса с капиталоемкой собственной IT-инфраструктурой гибридное облако можно назвать обязательным этапом при переходе к публичному облаку.

Среди недостатков гибридного облака следует отметить:

– снижение конфиденциальности по сравнению с возможностями частного облака;

– потребность в дополнительных тратах на развертывание технологии при запуске проекта;

– оставление части функций по управлению ресурсами в руках клиента, что может от него потребовать привлечения новых опытных IT-сотрудников или повышения квалификации существующих.

При выборе публичного или гибридного облака, компании предстоит выбор варианта организации сервиса. Доступны следующие варианты:

– IaaS (Infrastructure as a Service) – инфраструктура как услуга;

– PaaS (Platform as a Service) – платформа как услуга;

– SaaS (Software as a Service) – ПО как услуга.

IaaS – предоставление вычислительных ресурсов по запросу, на которых заказчик имеет возможность развернуть и запустить произвольное программное обеспечение, включающее в себя операционные системы и приложения. В рамках данной модели заказчик не управляет и не контролирует лежащую в основе физическую инфраструктуру, но имеет контроль над операционными системами и развернутыми приложениями.

PaaS – предоставление облачной платформы для развертывания программного обеспечения, созданного на базе языков программирования и инструментов, поддерживаемых облачным провайдером. Заказчик не имеет возможности управлять облачной инфраструктурой (сетевое и серверное

оборудование, СХД, операционными системами), но имеет контроль над развернутыми приложениями и, возможно, настройками окружающей среды.

SaaS – предоставление в пользование заказчику приложений, развернутых на облачной инфраструктуре провайдера. Приложения могут быть доступны с различных клиентских устройств посредством тонкого клиента, терминального клиента или браузера. Заказчик не контролирует параметры работы и настройки приложений. Весь сервис предоставляется под ключ [2].

1.2 Обзор сервисов WEB

В зависимости от требований и возможностей компания может выбрать способ размещения своих Web-сервисов несколькими путями:

- собственная инфраструктура;
- выделенный сервер;
- виртуальный хостинг (shared hosting);
- виртуальный выделенный сервер(VPS).

1.2.1 Виртуальный хостинг

В случае, когда не требуется или нет возможности содержать и обслуживать собственную инфраструктуру выбор падает на выведение Web-сервисов в «облака». Самым простым с точки зрения обслуживания является виртуальный хостинг. В этом варианте заказчику предоставляется раздел сервера, на котором расположены множество других веб-сайтов. Этот вариант также является и самым дешевым. В то же время подходит только для небольших проектов из-за неравномерной нагрузки. Виртуальный хостинг описывается следующими параметрами:

- размер дискового пространства;
- кол-во месячного трафика;
- кол-во сайтов, которые можно разместить в рамках хостинга как одной услуги;
- кол-во баз данных и кол-во места под базы данных;
- кол-во почтовых ящиков и FTP-аккаунтов;

– свободные ресурсы CPU и оперативной памяти [3].

1.2.2 Виртуальный выделенный сервер

Если нагрузка на веб-приложение компании превышает способности серверов виртуального хостинга, следует отдать предпочтение VPS-хостингу. VPS (Virtual Private Server) хостинг представляет собой виртуальный частный сервер, который находится на мощном физическом сервере. Физический сервер разделен на несколько виртуальных и таким образом достигается изолированность виртуальных серверов друг от друга. Каждый виртуальный сервер не влияет на производительность остальных, так как получает выделенные ресурсы. При использовании VPS заказчик получает полный контроль над сервером, как если бы это был физический сервер. Также в случае VPS имеется возможность выбрать управляемый или неуправляемый. При неуправляемом VPS весь контроль над сервером находится у заказчика. При выборе управляемого VPS эти функции берет на себя провайдер VPS [4].

На выбор провайдера VPS влияет множество факторов:

– операционная система сервера;
– технология виртуализации;
– надежность;
– избыточность;
– масштабируемость;
– квота на полосу пропускания;
– поддержка заказчиков;
– местонахождение VPS;
– дополнительные IP-адреса;
– стоимость [5].

Большинство провайдеров предлагают популярные Windows и Linux. ОС Linux как Open Source стоит дешевле Windows. Хостинг на базе Linux вполне дружественен к пользователю и поддерживает широкий спектр приложений. Во многих случаях это хороший выбор. Однако есть приложения, которые в Linux

либо вовсе не поддерживаются, либо лучше поддерживаются в Windows. Если нужно использовать такое ПО, например, как ASP или ASP.NET, то выбор – VPS на базе Windows. Сервер под Windows часто требуется для разработки на .NET или для развертывания приложений Microsoft и других приложений под эту платформу.

Технология виртуализации — один из важных факторов выбора виртуального сервера: программная (виртуализацию на уровне ядра операционной системы) или аппаратная (полная изоляция, управляемая гипервизором). В первом случае используется одно общее ядро системы и выбранный провайдером тип ОС: для Linux server — CentOS, Debian, Ubuntu и т.п., для Windows VPS – та же версия Windows. Полная виртуализация – это физическое разделение ядра, когда каждый участник может установить требуемую ОС.

Избыточность обычно предполагает резервирование ресурсов, особенно в ЦОД. Например, при отказе основного электропитания начинают работать ИБП и дизель-генераторы. Если проблемы у интернет-провайдера, то должны быть альтернативные каналы связи. Если один физический сервер будет перегружен, то должен быть предусмотрен резервный, и т.д. Масштабируемость, в свою очередь, означает способность справляться с резким ростом нагрузки на сервер, обычно за счет резервных ресурсов. Все это означает увеличение времени бесперебойной работы и стабильно высокую производительность.

Большинство провайдеров VPS ограничивают полосу пропускания для виртуального сервера и могут взимать отдельную плату за дополнительную.

Независимо от эффективности работы хостинг-провайдера и предлагаемой функциональности всегда возникают какие-то проблемы. На этот случай необходима удобная и действенная поддержка. Если хостер не в состоянии обеспечить поддержку в режиме 24/7, он просто не стоит денег. Когда сайт будет долго простаивать, это может привести к оттоку посетителей, а может быть и к серьезным финансовым потерям.

Чем ближе сервер находится к целевой аудитории, тем эффективнее будет доступ к нему пользователей и выше шансы подняться в рейтингах поисковиков.

Дополнительные IP-адреса могут потребоваться в нескольких ситуациях:

– установка SSL-сертификата;

– присваивание выделенного IP каждому сайту на виртуальном сервере (иначе они автоматически получат IP-адрес сервера VPS);

– разные IP для разных каналов (веб-сайта, мобильные приложения и пр.);

– разные IP для разных сервисов (CMS, база данных и др.);

– назначение нескольких IP одному сайту, например, имеющему домены на разных языках.

Стоимость главным образом составляется из конфигурации сервера, но увеличивается при необходимости дополнительных услуг (например, управляемый сервер).

Аналогом VPS сервера является сервис Amazon VPC (Virtual Private Cloud), позволяющий выделить логически изолированный раздел облака AWS (Amazon Web Services). Пользователь полностью контролирует свою среду виртуальной сети, в том числе может выбирать собственные диапазоны IP-адресов, создавать подсети, а также настраивать таблицы маршрутизации и сетевые шлюзы. Сетевую конфигурацию для Amazon VPC можно легко настроить по своему усмотрению. Например, для веб-серверов можно создать публичную подсеть с выходом в Интернет, а внутренние системы, такие как базы данных или сервера приложений, расположить в частной подсети без доступа к Интернету [6].

Выделенный сервер предоставляет максимальный контроль над собой и лучшую производительность, но требует квалифицированного персонала и имеет большую стоимость по сравнению с другими вариантами.

1.3 Методы разработки проекта

В настоящее время используется большое количество подходов, которые позволяют, так или иначе, создавать модели бизнес-процессов предприятий. Как бы то ни было, их использование гарантирует стандартизированный подход к описанию, позволяет накапливать опыт и практические навыки и на протяжении длительного времени обеспечивать понимание созданных моделей другими

сотрудниками. Важнейшими из подходов являются структурный (функциональный) и объектно-ориентированный [7].

1.3.1 Структурный (функциональный) подход

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. В качестве средств структурного анализа и проектирования, наиболее распространены следующие нотации:

а) Нотация IDEF0 — Наиболее популярная нотация моделирования бизнес-процессов, основанная на методологии структурного анализа SADT (Structured Analysis and Design Technique). Методология IDEF0 — это методология моделирования, позволяющая создать функциональную модель, отображающую структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции. Бизнес-процессы в нотации IDEF0 представляются в форме прямоугольника, а стрелки отражают связь с другими процессами и внешней средой. Особенностью нотации является возможность декомпозировать процессы на подпроцессы и, таким образом, строить иерархические модели бизнес-процессов. Нотация IDEF0 используется для создания верхнего уровня модели бизнес-процессов. Построение IDEF0-диаграммы верхнего уровня обеспечивает наиболее общее или абстрактное описание объекта моделирования [8];

б) Нотация DFD — предназначена для моделирования информационный систем с точки зрения хранения, обработки и передачи данных. Применяется этот вид нотации в случае, когда требуется описание системы как хранилища данных. Нотация DFD состоит из нескольких элементов: процесс, внешние сущности, хранилище данных, поток данных. Процесс — функция или оследовательность

действий, которые нужно предпринять, чтобы данные были обработаны. Это может быть создание заказа, регистрация клиента и т.д. В названиях процессов принято использовать глаголы, т.е. «Создать клиента» (а не «создание клиента») или «обработать заказ» (а не «проведение заказа»). Внешние сущности — это любые объекты, которые не входят в саму систему, но являются для нее источником или получателями каких-либо данных из системы после обработки. Хранилище данных — внутреннее хранилище данных для процессов в системе. Здесь хранятся поступившие данные перед обработкой и результат после обработки, а также промежуточные значения. Поток данных в нотации отображается в виде стрелок, которые показывают, какие данные входит, а какие исходят из того или иного блока на диаграмме [9].

в) Нотация IDEF3 — нотация описания и моделирования бизнес-процессов, которая отображает причинно-следственные связи между событиями, функциями. Основная цель нотации IDEF3 — дать аналитикам возможность описать ситуацию, когда процессы (действия) выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе. Нотация IDEF3 состоит из: действий, связей, соединений (перекрестков) и объектов ссылки (указателей). В IDEF3 действия изображаются прямоугольниками с прямыми углами. Действия имеют имя, выраженное отглагольным существительным или глаголом, одиночным или в составе фразы с другим именем существительным, обычно отображающим основной выход (результат) работы, например, "Создание файла". Нотация IDEF3 позволяет декомпозировать (детализировать) действие многократно, т.е. включить в одну модель альтернативные описания процессов. Действия имеют входы и выходы, но не поддерживают управления и механизмы, как функции в нотации IDEF0. Связи показывают существенные взаимоотношения между действиями. Все связи в IDEF3 однонаправлены, могут начинаться и заканчиваться на любой стороне блока. Обычно диа-граммы IDEF3 стараются построить так, чтобы связи были направлены слева направо, сверху вниз. Перекрестки (соединения) используются для отображения логики взаимодействия связей при слиянии и разветвлении.

Различают перекрестки для слияния и разветвления стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления. Различают синхронные и асинхронные соединения. Используются для изображения соответственно синхронных действий, т.е. начинающихся и заканчивающихся одновременно, и для изображения асинхронных действий. Объект ссылки в IDEF3 выражает некую идею, концепцию или данные, которые нельзя описать стрелкой, перекрестком или действием. Объекты ссылки должны быть связаны с действиями или перекрестками линиями [10].

г) Нотация IDEF1X — является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы. Основными составляющими модели являются сущность, связь и атрибут. Сущностью называют любой действительный или представляемый объект, данные о котором должны сохраняться и быть доступны. Экземпляр сущности – это конкретный представитель данной сущности. Все экземпляры сущности должны быть различимы между собой. Сущности должны иметь свойства, которые будут уникальными для каждого экземпляра этой сущности. Множество из одного или нескольких атрибутов, которые определяют уникальность каждого экземпляра сущности, называют идентификатором (ключом). Каждая сущность должна иметь хоть один идентификатор. Атрибут – поименованная характеристика сущности, определяющая его свойства и принимающая одно значение множества. Каждому атрибуту присваивается имя, которое обозначает его смысл и значение. Сами атрибуты отображаются внутри сущности в виде списка их имен. Связь – это графически изображаемая ассоциация, которую чаще всего устанавливают между двумя сущностями, но бывают случаи рекурсивной связи, когда связь существует между сущностью и ей же самой. В любой связи выделяются два конца, на каждом из которых указывается имя связи [11].

1.3.2 Объектно-ориентированный подход

Принципиальное различие между структурным и объектно-ориентированным подходом (ООП) заключается в способе декомпозиции системы. ООП использует объектную декомпозицию, при этом статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщений между объектами.

Объект в ООП — это абстракция реальной или воображаемой сущности с четко выраженными концептуальными границами, индивидуальностью (идентичностью), состоянием и поведением. Индивидуальность – это свойство объекта, с помощью которого его можно отличить от других.

Для концептуальной группировки однотипных объектов в объектно-ориентированном подходе используется понятие «класс». Класс – это множество объектов, имеющих общую структуру и поведение. Таким образом, класс – это шаблон, на основе которого генерируются (создаются) однотипные объекты. В качестве синонима понятия «объект» часто употребляют понятие «экземпляр класса». Каждый класс и, соответственно, объект характеризуются строго определенным набором атрибутов и методов. Текущие значения атрибутов четко определяют текущее состояние объекта. Набор методов и их алгоритмическая реализация определяют поведение объекта (класса объектов).

Основными принципами ООП являются наследование, инкапсуляция и полиморфизм. Наследование – принцип, в соответствии с которым знание об общей категории разрешается применять для более узкой. Применительно к классам это означает, что дочерний класс (узкая категория) полностью включает в себя (наследует) все атрибуты и методы, определенные в родительском классе (общей категории). При этом в дочернем классе могут быть определены дополнительные атрибуты и методы. Инкапсуляция — принцип, в соответствии с которым содержание внутреннего устройства элементов системы должно быть скрыто друг от друга. Этот принцип предписывает обмен данными между объектами системы только в минимально необходимом объеме, ограничение доступа к атрибутам и методам объектов (классов) со стороны других объектов

(классов) и полное скрытие алгоритмической реализации методов от других объектов (классов). Полиморфизм – принцип построения элементов модели так, чтобы они могли принимать различные внешние формы или функциональность (поведение) в зависимости от обстоятельств.

Таким образом, суть объектно-ориентированного подхода к анализу и проектированию информационных систем заключается в декомпозиции системы на классы, которые соответствуют однотипным объектам предметной области, и построении из них иерархии в виде ориентированного графа с использованием отношений композиции и наследования.

В качестве средства анализа и проектирования в ООП используется нотация UML (Unified Modeling Language) — графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем. Основное назначение UML – предоставить, с одной стороны, достаточно формальное, с другой стороны, достаточно удобное, и, с третьей стороны, достаточно универсальное средство, позволяющее до некоторой степени снизить риск расхождений в толковании спецификаций. В качестве основных графических элементов были выбраны следующие пять: фигура, линия, значок, текст, рамка.

Фигуры в UML используются двумерные (т.е. их можно нарисовать на плоскости) и замкнутые (т.е. есть внутренняя и внешняя части). Фигуры могут менять свои размеры и форму, сохраняя при этом свои интуитивные отличительные признаки. Внутри фигур могут помещаться другие элементы нотации: тексты, линии, значки и даже другие фигуры. Единственное требование: должно быть однозначно понятно, что элемент нотации находится внутри фигуры, в частности, его изображение не должно пересекать границу фигуры.

Линии в UML одномерные. Линии всегда присоединяются своими концами к фигурам или значкам, они не могут быть нарисованы сами по себе. В UML используется только два стиля линий, которые трудно спутать: сплошные и пунктирные линии. К линиям могут быть пририсованы различные дополнительные элементы: стрелки на концах, тексты и т.д. Единственное требование: должно быть

ясно, что дополнительный элемент относится именно к данной линии. Пересечение линий ничего не значит, но не рекомендуется из-за затруднения восприятия.

Значки в UML похожи на фигуры тем, что они двумерные, а отличаются тем, что не имеют внутренности, в которую можно что-то поместить, и, как правило, не меняют свою форму и размеры.

Тексты в UML – это, последовательности различимых символов некоторого алфавита. Алфавит не фиксирован – он только должен быть понятен читателю модели. Гарнитура, размер и цвет шрифта не имеют значения, а вот начертание шрифта имеет: в UML различаются прямые, курсивные и подчеркнутые тексты.

Рамки появились в UML 2. Рамка – это частный случай фигуры, которая используется исключительно как контейнер для других фигур, линий, значков и текстов. Пустые рамки не применяются. Рамка имеет прямоугольную форму и, как правило, ярлычок в левом верхнем углу, в котором указывается тип и имя рамки.

Модель UML – это совокупность конечного множества конструкций языка, главные из которых – это сущности и отношения между ними. Рассматривая модель UML с наиболее общих позиций, можно сказать, что это граф, в котором вершины и ребра нагружены дополнительной информацией и могут иметь сложную внутреннюю структуру. Вершины этого графа называются сущностями, а ребра – отношениями.

Диаграмма – это графическое представление некоторой части графа модели. Диаграммы UML являются накладываемой на модель структурой, которая облегчает создание и использование модели. В версии UML 2.0 и выше определены 13 типов диаграмм:

- диаграмма использования;
- диаграмма классов;
- диаграмма объектов;
- диаграмма деятельности;
- диаграмма последовательности;
- диаграмма компонентов;

– диаграмма размещения;
– диаграмма внутренней структуры;
– диаграмма пакетов;
– диаграмма автомата;
– диаграмма коммуникации;
– обзорная диаграмма взаимодействия;
– диаграмма синхронизации.

Все диаграммы UML можно условно разбить на две группы, первая из которых – общие диаграммы. Общие диаграммы практически не зависят от предмета моделирования и могут применяться в любом программном проекте без оглядки на предметную область, область решений и т.д. К общим диаграммам относятся диаграммы использования, классов, автомата, деятельности, последовательности, коммуникации, компонентов, размещения. Остальные диаграммы являются специальными и характеризуются тем, что чаще всего служат для дополнения какой-либо общей диаграммы, например, являются ее частным случаем или же просто играют вспомогательную роль, уточняя некоторые детали [12].

1.4 Этапы проектирование базы данных

К основным этапам проектирования базы данных в ходе разработки автоматизированной информационной системы относят следующие этапы:

а) Инфологическое проектирование – построение концептуальной модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель строится без ориентации на конкретную базу данных и модель данных. Такая модель включает в себя:

1) Описание объектов предметной области и связей между ними
2) Описание ограничений целостности, т.е. требования к допустимым значениям данных и к связям между ними
3) Описание алгоритмических зависимостей между данными

б) Логическое проектирование – преобразование инфологической модели

на основе выбранной модели данных в логическую модель, не зависимую от особенностей СУБД, с помощью которой в дальнейшем произойдет физическая реализация базы данных. Обычно логическая модель представляется в виде набора таблиц с указанием ключевых полей и связей между таблицами.

в) Физическое проектирование – реализация логической модели с помощи конкретного СУБД. Такое проектирование включает в себя задачи, которые решаются в основном средствами СУБД и скрыты от разработчика БД, а именно: выбор методов управления дисковой памятью, методов сжатия данных, методов доступа к данным [13].

1.5 Обзор технологий реализации проекта

Реализация проекта разделяется на две основные части: разработку серверной (бэкенд) и клиентской (фронтенд) частей.

1.5.1 Технологии реализации Front-end

Фронтенд (от англ. Front-end) — та часть приложения, которая выполняется непосредственно в браузере и с которой взаимодействует пользователь. Набор языков для фронтенд-разработки достаточно стандартизирован — HTML для разметки страниц, CSS для внешнего вида и JavaScript (или библиотеки на его основе) для сценариев.

HTML (от англ. HyperText Markup Language — язык гипертекстовой разметки) — стандартизированный язык разметки документов во Всемирной паутине.

CSS (от англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

JavaScript — полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах.

1.5.2 Технологии реализации Back-end

Бэкенд (от англ. Back-end) — часть приложения, выполняющаяся на сервере. Обычно состоит из трёх частей: сервер, приложение и база данных. Сервером может являться любой компьютер от микрокомпьютера вроде RaspberryPi до высокопроизводительного сервера в составе облака. Серверная часть приложения может быть реализована на множестве языков в зависимости от предпочтения компании или разработчика. Самыми популярными языками являются: PHP, Ruby, Python, Java, Go, а также фреймворки на их основе. Например, Ruby on Rails, Django, Node.js. В качестве хранилища данных используются системы управления базами данных, такие как MySQL, PostgreSQL, MongoDB, MariaDB.

2 Анализ предметной области, определение цели, выбор проектных решений

Анализ предметной области – это начальный шаг этапа системного анализа, с которого начинается разработка программного продукта. Предметная область оказывает сильнейшее влияние на все аспекты проекта: взаимодействие с пользователем, требования к системе, используемая для хранения данных модель, реализация и многое другое. Анализ предметной области проводится для выделения ее сущностей, определения первоначальных требований к функциональности и определения границ проекта.

2.1 Описание предметной области

Система учета работы оборудования может быть использована на различных предприятиях, в лабораториях. В данной работе рассматривается разработка проекта системы учета работы оборудования для исследовательской лаборатории. В лаборатории проводятся исследования по научно-исследовательским работам (НИР), которые заказывают заказчики. Заказчики предоставляют образцы для исследований. Исследования проводятся на оборудовании. Для образцов и

оборудования есть определенные типы исследований. Для выполнения работы исследователь резервирует установку на выбранное количество часов.

2.2 Метод разработки проекта базы данных

2.2.1 Разработка инфологической модели базы данных

Инфологическая модель данных представляет собой описание предметной области с помощью математических формул, таблиц, графиков и других средств, понятных всем людям, кто связан с проектированием баз данных.

Цель инфологического моделирования заключается в наиболее естественном сборе и представлении данных, которые в будущем собираются хранить в создаваемой базе данных. Основными конструктивными элементами таких моделей являются таблицы (сущности), связи между ними и атрибуты (свойства сущностей).

Один из самых популярных подходов к построению инфологической модели — модель «сущность-связь» или ER-модель (Entity-Relationship). Основными понятиями ER-модели являются сущность, связь и атрибут. Для создания ER-модели в данной работе использовался сервис Lucidchart.

Этапы процесса построения инфологической модели:

а) Определение сущностей;

б) Определение зависимостей между сущностями;

в) Задание первичных и альтернативных ключей;

г) Определение атрибутов сущностей;

д) Приведение модели к требуемому уровню нормальной формы.

2.2.2 Разработка логической модели

Исходными данными для диаграмм логической модели служат диаграммы концептуальной (инфологической) модели ИС. Преобразование локальной концептуальной модели данных в локальную логическую модель заключается в удалении из концептуальных моделей нежелательных элементов и преобразование полученных моделей в локальные логические модели. Логическая схема базы данных реализовывается с помощью реляционной модели данных. В реляционной базе данных каждая сущность (таблица) имеет первичный ключ, состоящий из

одного или нескольких атрибутов (полей), которые идентифицирует каждую запись в таблице.

2.3 Метод разработки проекта приложения

Для разработки проекта приложения системы учета работы оборудования был выбран функциональный подход, так как функциональные модели позволяют реализовать структурный подход к проектированию автоматизированных информационных систем (АИС) по принципу «сверху — вниз», когда каждый функциональный блок может быть декомпозирован на множество отдельных блоков и так далее, выполняя, таким образом, модульное проектирование АИС. Для функциональных моделей характерны процедурная строгость декомпозиции ИС и наглядность представления. Для составления функциональной модели было использовано средство Ramus, позволяющее создавать модели IDEF0 и DFD.

Исходя из функций, исполняемых лабораторией, целесообразно создание автоматизированной информационной системы, позволяющей автоматизировать резервирование оборудования. Основной функцией АИС является выполнение процесса резервирования оборудования сотрудником лаборатории.

3 Практический результат

3.1 Создание проекта базы данных

3.1.1 Разработка инфологической модели

В результате анализа предметной области были выделены следующие основные сущности:

а) «Пользователи» – список сотрудников лаборатории;

б) «Оборудование» – список оборудования в лаборатории;

в) «НИР» – детали научно-исследовательской работы;

г) «Заказчик НИР» – список заказчиков НИР;

д) «Образцы» – список образцов для исследований;

е) «Заказ НИР» – список заказанных НИР;

ж) «Типы исследований» – наименования типов исследований;

з) «Типы исследования на оборудовании» – типы исследований, которые можно проводить на данном оборудовании;

и) «Тип исследования для образца» – типы исследований, которые можно производить с данным образцом;

к) «Дни недели» – список рабочих дней недели;

л) «Рабочие часы» – список рабочих часов;

м) «Зарезервированное время» – время, зарезервированное сотрудником;

н) «Проведённые работы» – отчеты о проведенных работах.

На рисунке 1 представленная инфологическая модель.

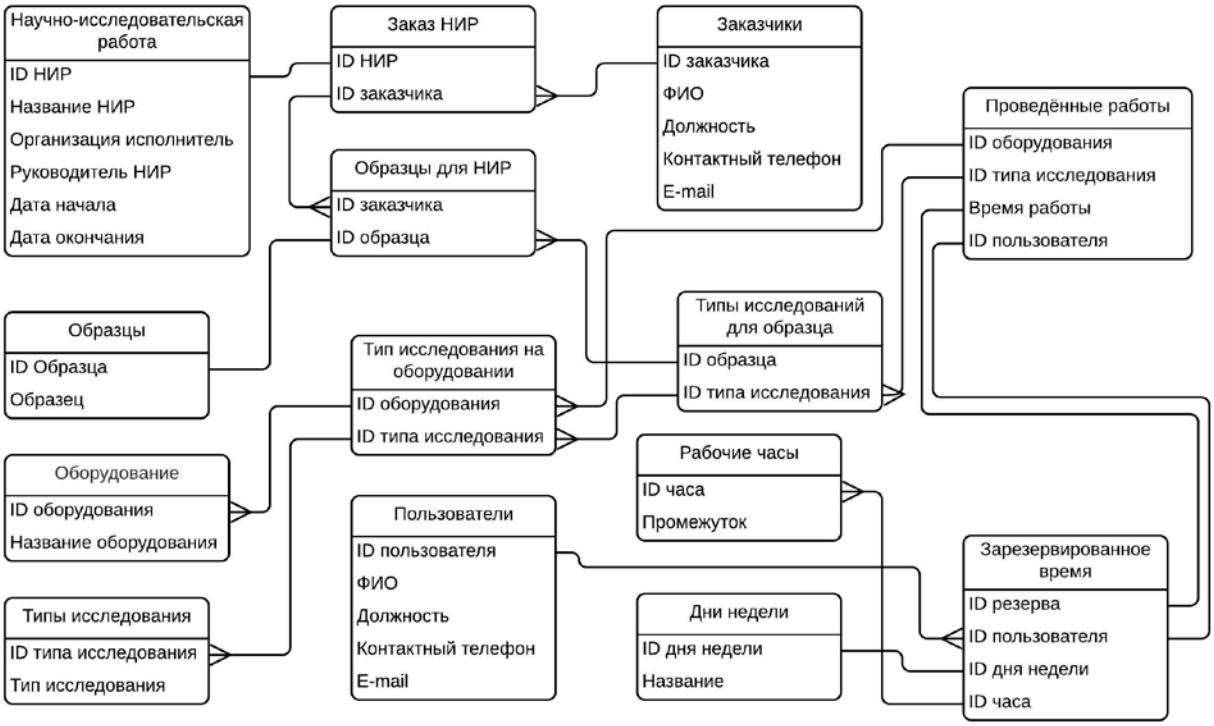


Рисунок 1 - Инфологическая модель

3.1.2 Разработка логической модели

Логическая схема базы данных представлена в виде таблиц 1.1 – 1.14.

Таблица 1.1 – Научно-исследовательская работа

№	Имя поля	Внешний ключ для	Тип
1	ID НИР		Счетчик
2	Название НИР		Текст(50)
3	Организация исполнитель		Текст(50)
4	Руководитель НИР		Текст(50)
5	Дата начала		Дата и время
6	Дата окончания		Дата и время

Таблица 1.2 – Заказ НИР

№	Имя поля	Внешний ключ для	Тип
1	ID НИР	Научно-исследовательская работа.ID НИР	Счетчик
2	ID заказчика	Заказчики.ID заказчика	Счетчик

Таблица 1.3 – Заказчики

№	Имя поля	Внешний ключ для	Тип
1	ID заказчика		Счетчик
2	ФИО		Текст(50)
3	Должность		Текст(50)
4	Контактный телефон		Текст(12)
5	E-mail		Текст(50)

Таблица 1.4 – Образцы

№	Имя поля	Внешний ключ для	Тип
1	ID образца		Счетчик
2	Образец		Текст(50)

Таблица 1.5 – Образцы для НИР

№	Имя поля	Внешний ключ для	Тип
1	ID заказчика	Заказ НИР.ID заказчика	Счетчик
2	ID образца	Образцы.ID образца	Счетчик

Таблица 1.6 – Оборудование

№	Имя поля	Внешний ключ для	Тип
1	ID оборудования		Счетчик
2	Название оборудования		Текст(50)

Таблица 1.7 – Типы исследования

№	Имя поля	Внешний ключ для	Тип
1	ID типа исследования		Счетчик
2	Тип исследования		Текст(50)

Таблица 1.8 – Тип исследования на оборудовании

№	Имя поля	Внешний ключ для	Тип
1	ID оборудования	Оборудование.ID оборудования	Счетчик
2	ID типа исследования	Типы исследования.ID типа исследования	Счетчик

Таблица 1.9 – Тип исследования для образца

№	Имя поля	Внешний ключ для	Тип
1	ID образца	Образцы для НИР.ID образца	Счетчик
2	ID типа исследования	Тип исследования на оборудовании.ID типа исследования	Счетчик

Таблица 1.10 – Пользователи

№	Имя поля	Внешний ключ для	Тип
1	ID заказчика		Счетчик
2	ФИО		Текст(50)
3	Должность		Текст(50)
4	Контактный телефон		Текст(12)
5	E-mail		Текст(50)

Таблица 1.11 – Рабочие часы

№	Имя поля	Внешний ключ для	Тип
1	ID часа		Счетчик
2	Промежуток		Текст(12)

Таблица 1.12 – Дни недели

№	Имя поля	Внешний ключ для	Тип
1	ID дня недели		Счетчик
2	Название		Текст(11)

Таблица 1.13 – Зарезервированное время

№	Имя поля	Внешний ключ для	Тип
1	ID резерва		Счетчик
2	ID пользователя	Пользователи.ID пользователя	Счетчик
3	ID дня недели	Дни недели.ID дня недели	Счетчик
4	ID часа	Рабочие часы.ID часа	Счетчик

Таблица 1.14 – Проведённые работы

№	Имя поля	Внешний ключ для	Тип
1	ID оборудования	Тип исследования на оборудовании.ID оборудования	Счетчик
2	ID типа исследования	Типы исследований для образца.ID типа исследования	Счетчик
3	ID пользователя	Зарезервированное время.ID пользователя	Счетчик
4	Время работы	Зарезервированное время.ID резерва	Счетчик

3.2 Создание проекта приложения

3.2.1 Функциональная модель IDEF0

В процессе разработки проекта было построено три уровня диаграммы:

- контекстная;
- функциональная декомпозиция;
- декомпозиция процессов.

На первом шаге проводится описание системы в целом и её взаимодействие с окружающим миром в виде контекстной диаграммы, представленной на рисунке 2.

Основные информационные потоки представлены в виде стрелок:

а) Входные потоки:

- заказы НИР — данные о НИР от заказчика, а также предоставленные образцы;
- оборудование — оборудование, находящееся в лаборатории.

б) Управляющие потоки:

- правила — необходимые условия, которые требуется соблюсти;
- приказы руководства — поставленная задача руководства.

в) Ресурсные потоки:

- сотрудники — сотрудники лаборатории, занимающиеся исследованиями;
- сервер — сервер, с помощью которого осуществляется функционирование приложения.

г) Выходной поток:

– отчёт — документ, в котором описаны детали проведенных работ, а также их результат.

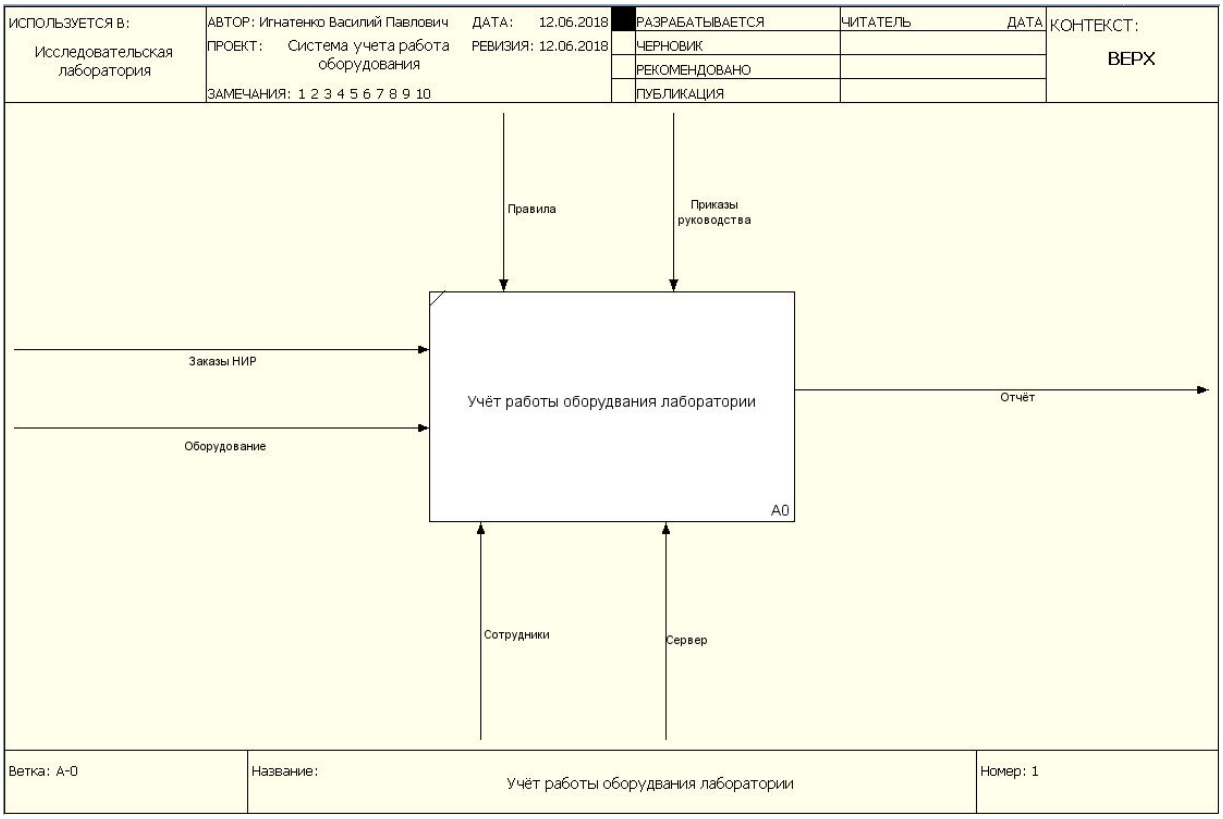


Рисунок 2 - Контекстная диаграмма

После построения контекстной диаграммы проводится её функциональная декомпозиция, которая представлена на рисунке 3. Вся система разбивается на подсистемы, и каждая из них описывается отдельно.

Процесс «Учёт работы оборудования лаборатории» был декомпозирован на следующие активности:

– регистрация запроса резервирования установки в системе;

– составление отчёта о произведённых исследованиях.

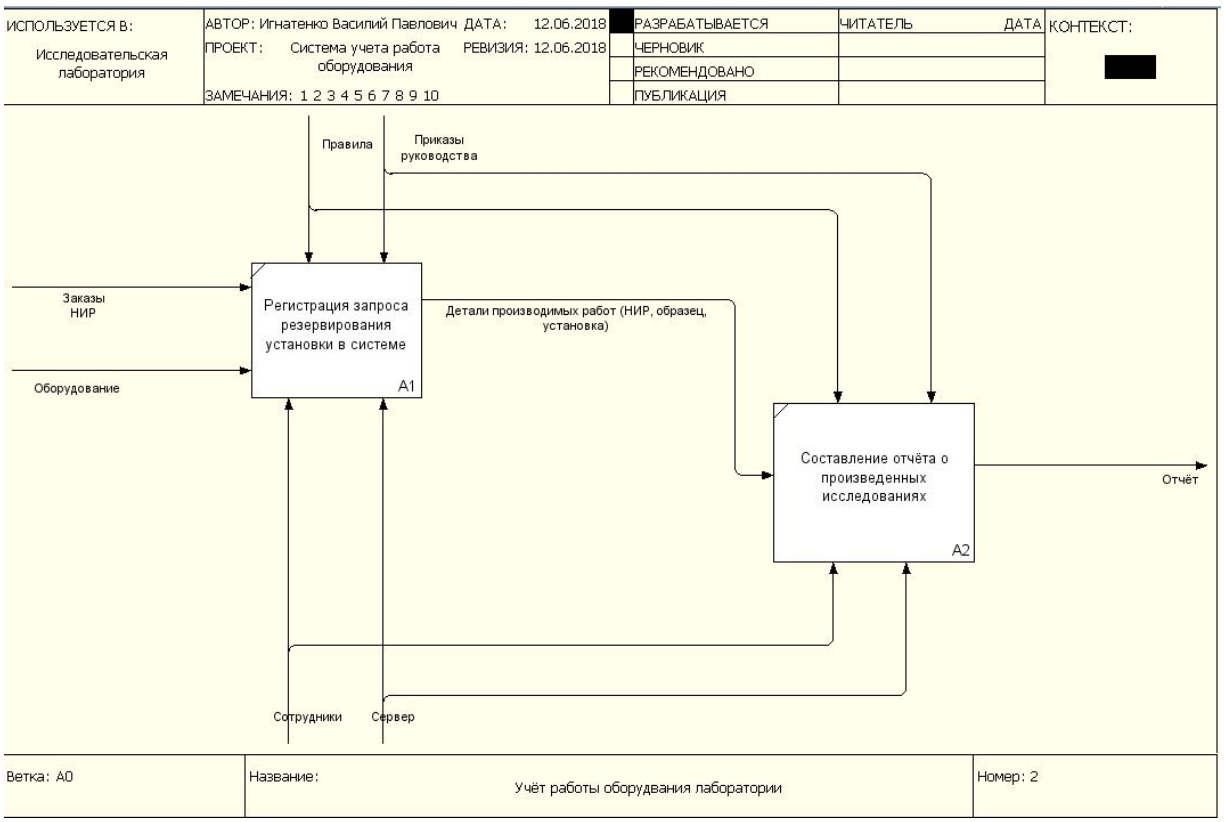


Рисунок 3 - Диаграмма декомпозиции 1-го уровня «Учёт работы оборудования лаборатории»

Регистрация запроса резервирования установки в системе — процесс записи в базу данных сведений о резервируемом сотрудником времени, НИР, по которой планируется исследование, установке, на которой планируется исследование, а также образцах, с которыми будут производиться работы. В данную активность входит шесть граничных стрелок (вход, механизм, управление) и выходит одна стрелка (связь по входу).

Входные потоки:

– заказы НИР;

– оборудование.

Управляющие потоки:

– правила;

– приказы руководства.

Ресурсные потоки:

– сотрудники;

– сервер.

Связь по входу между «Регистрация запроса резервирования установки в системе» и «Составление отчёта о произведённых исследованиях» — детали производимых работ (НИР, образец, установка).

Составление отчёта о произведённых исследованиях — процесс составления отчёта о факте проведённых работ по НИР с предоставленными образцами на установке. В активность входит пять граничных (управление, механизм, выход) и одна внутренняя (связь по входу).

Входной поток:

– детали производимых работ (НИР, образец, установка).

Управляющие потоки:

– правила;

– приказы руководства.

Ресурсные потоки:

– сотрудники;

– сервер.

Выходной поток:

– отчёт.

После декомпозиции контекстной диаграммы проводится декомпозиция фрагментов системы на более мелкие, пока не будет достигнут нужный уровень подробности описания. На рисунке 4 представлена диаграмма, детализирующая регистрация запроса резервирования установки в системе.

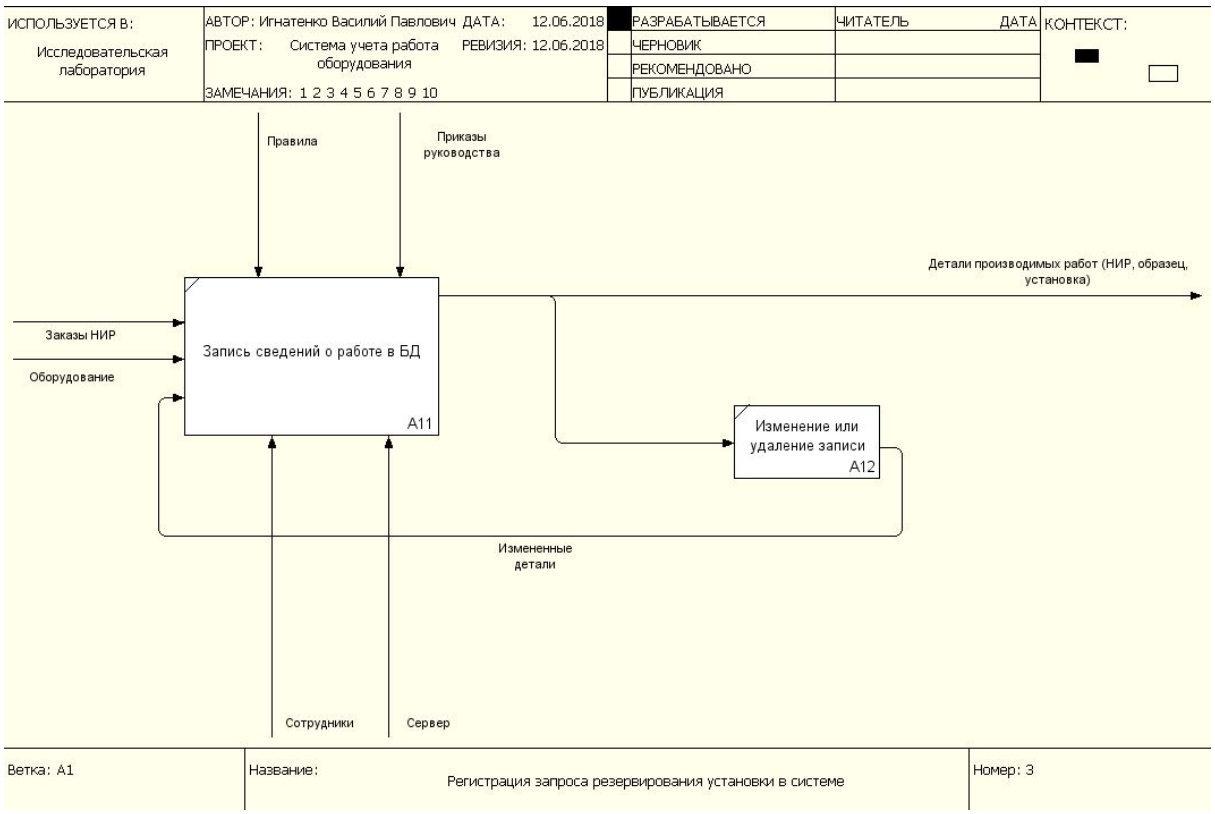


Рисунок 4 - Диаграмма декомпозиции 2-го уровня «Регистрация запроса резервирования установки в системе»

В результате детализации процесса «Регистрация запроса резервирования установки в системе» были выделены две основные функции:

– запись сведений о работе в БД;

– изменение или удаление записи.

Запись сведений о работе в БД — процесс внесения сведений о планируемом исследовании в базу данных. В данную активность входит семь граничных стрелок (управление, вход, механизм, выход) и одна внутренняя.

Входные потоки:

– заказы НИР;

– изменённые детали;

– оборудование.

Управляющие потоки:

– правила;

– приказы руководства.

Ресурсные потоки:

– сотрудники;

– сервер.

Выходной поток:

– детали производимых работ (НИР, образец, установка).

Изменение или удаление записи — процесс изменения записи о планируемом исследовании, либо её удалении. В активность входят две внутренние стрелки.

Входной поток:

– детали производимых работ (НИР, образец, установка).

Выходной поток:

– изменённые детали.

Заключение

В ходе выполнения работы были изучены методы проектирования баз данных и приложений. В ходе работы были выполнены следующие задачи:

1. Проведён анализ предметной области «исследовательская лаборатория», который помог определить требования к разрабатываемому проекту информационной системы.
2. По результатам анализа предметной области была составлена инфологическая и разработана логическая модели базы данных.
3. Разработана функциональная модель приложения.

Список литературы

1. Облачное хранилище: как выбрать, что такое облачный сервер, в каком облаке лучше хранить данные? [Электронный ресурс] // Комсомольская Правда. - 11 Октябрь 2017 г.. - Режим доступа: https://www.kp.ru/guide/oblachnoe-khranilishche.html

2. Центр компетенции: Облачные сервисы или что такое Iaas? Отличие от SaaS и PaaS [Электронный ресурс] // Облачный провайдер "ИТ-ГРАД". - 3 Сентябрь 2014 г.. - Режим доступа: http://www.it-grad.ru/competence-center/technical-blog/33/

3. Виртуальный хостинг [Электронный ресурс] // Википедия - свободная энциклопедия. - Режим доступа: https://ru.wikipedia.org/wiki/Виртуальный_хостинг

4. VDS/VPS: от хостинга до облаков [Электронный ресурс] // Хабрахабр. - 10 Август 2016 г.. - Режим доступа: https://habrahabr.ru/company/ruvds/blog/307494/

5. Что такое на самом деле VPS-хостинг и как выбрать надежного провайдера VPS [Электронный ресурс] // Хабрахабр. - 3 Октябрь 2016 г.. - Режим доступа: https://habrahabr.ru/company/ruvds/blog/311608/

6. Amazon VPC: вопросы и ответы [Электронный ресурс] // Amazon Web Services (AWS) - сервисы облачных вычислений. – Режим доступа: https://aws.amazon.com/ru/vpc/faqs/

7. Сравнительный анализ подходов к проектированию ИС [Электронный ресурс] // Институт вычислительных технологий СО РАН - Режим доступа: http://www.ict.nsc.ru/ws/YM2004/8666/index.htm

8. Знакомство с нотацией IDEF0 и пример использования [Электронный ресурс] // Хабрахабр – 28 Февраль 2017 г.. - Режим доступа: https://habr.com/company/trinion/blog/322832/

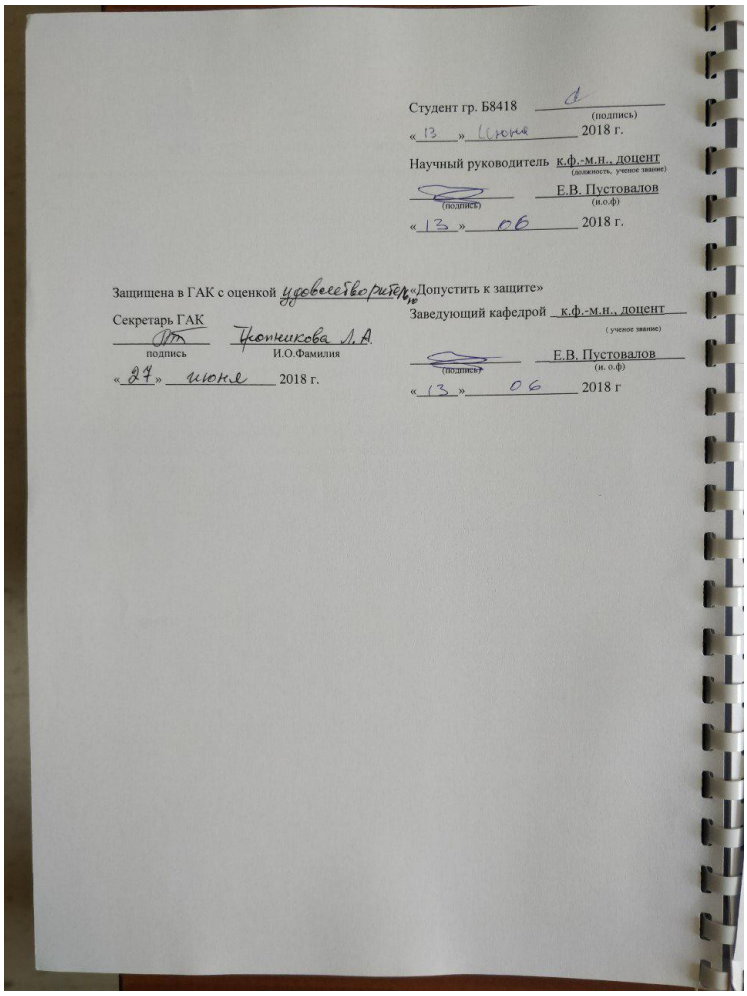
9. Что такое DFD (диаграммы потоков данных) [Электронный ресурс] // Хабрахабр – 13 Октябрь 2017 г.. - Режим доступа: https://habr.com/company/trinion/blog/340064/

10. Метод описания процессов IDEF3 [Электронный ресурс] // Электронная библиотека - Режим доступа: http://libraryno.ru/9-metod-opisaniya-processov-idef3-trpo/

11. Методология IDEF1 [Электронный ресурс] // Море аналитической информации - Режим доступа: http://citforum.ru/database/case/glava2_4_2.shtml

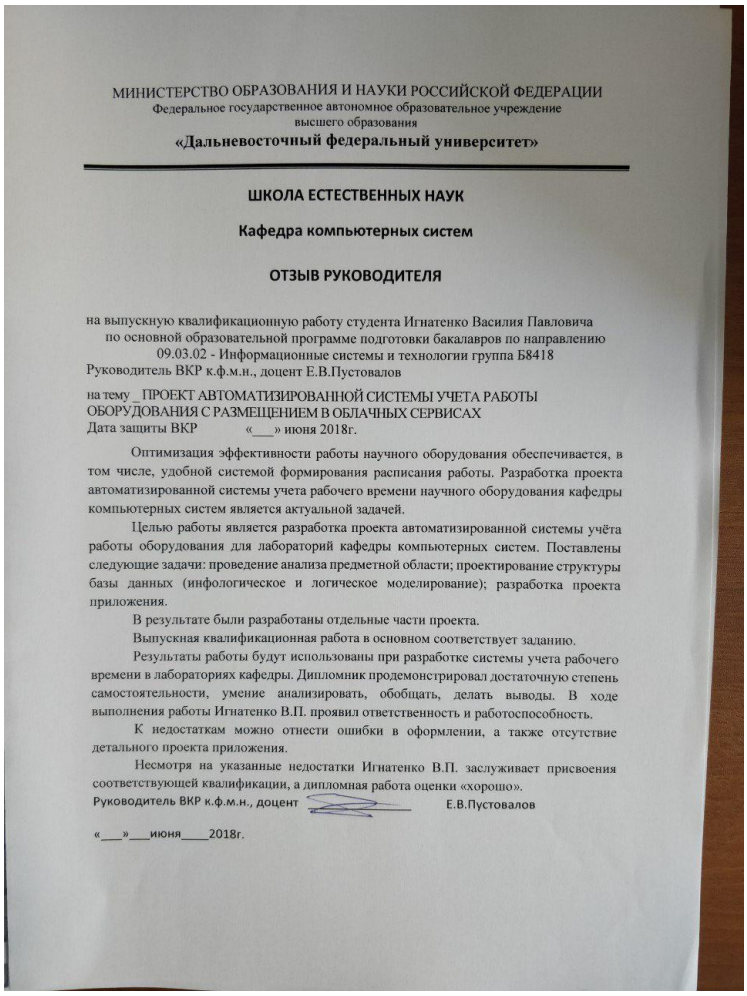
12. Новиков, Ф.А. Моделирования на UML [Электронный ресурс] // Новиков Ф.А., Иванов Д.Ю. – 3 Апрель 2013 г.. - Режим доступа: http://book.uml3.ru/

13. Основы проектирования баз данных [Электронный ресурс] // - Режим доступа: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema7/tema7_1



Студент гр. Б8418 _____ (подпись)
«13» июня 2018 г.
Научный руководитель к.ф.-м.н., доцент (должность, ученое звание)
_____ (подпись) Е.В. Пустовалов (и.о.ф)
«13» 06 2018 г.

Защищена в ГАК с оценкой удовлетворительно «Допустить к защите»
Секретарь ГАК
_____ подпись Пономарева Л.А. И.О.Фамилия
«24» июня 2018 г.
Заведующий кафедрой к.ф.-м.н., доцент (ученое звание)
_____ (подпись) Е.В. Пустовалов (и. о.ф)
«13» 06 2018 г



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра компьютерных систем

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу студента Игнатенко Василия Павловича
по основной образовательной программе подготовки бакалавров по направлению
09.03.02 - Информационные системы и технологии группа Б8418
Руководитель ВКР к.ф.м.н., доцент Е.В.Пустовалов

на тему _ПРОЕКТ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА РАБОТЫ ОБОРУДОВАНИЯ С РАЗМЕЩЕНИЕМ В ОБЛАЧНЫХ СЕРВИСАХ
Дата защиты ВКР «___» июня 2018г.

Оптимизация эффективности работы научного оборудования обеспечивается, в том числе, удобной системой формирования расписания работы. Разработка проекта автоматизированной системы учета рабочего времени научного оборудования кафедры компьютерных систем является актуальной задачей.

Целью работы является разработка проекта автоматизированной системы учёта работы оборудования для лабораторий кафедры компьютерных систем. Поставлены следующие задачи: проведение анализа предметной области; проектирование структуры базы данных (инфологическое и логическое моделирование); разработка проекта приложения.

В результате были разработаны отдельные части проекта.

Выпускная квалификационная работа в основном соответствует заданию.

Результаты работы будут использованы при разработке системы учета рабочего времени в лабораториях кафедры. Дипломник продемонстрировал достаточную степень самостоятельности, умение анализировать, обобщать, делать выводы. В ходе выполнения работы Игнатенко В.П. проявил ответственность и работоспособность.

К недостаткам можно отнести ошибки в оформлении, а также отсутствие детального проекта приложения.

Несмотря на указанные недостатки Игнатенко В.П. заслуживает присвоения соответствующей квалификации, а дипломная работа оценки «хорошо».

Руководитель ВКР к.ф.м.н., доцент _____ Е.В.Пустовалов

«___»___июня____2018г.