

Номер зачетной книжки 88107134
Производственная практика зачтена с оценкой

(цифрой) (прописью)

(подпись руководителя практики от БГУИР)
. .2021

Место прохождения практики: ЗАО «Кьюликс Системс», г. Минск
Сроки прохождения практики: с 27.10.2021 по 23.11.2021

Руководитель практики от
предприятия: В.П.Липлянина
(подпись руководителя)
М . П .

Студент группы 881074
 _____ А.В.Третьякова
 (подпись студента)
 Руководитель практики от БГУИР
 Д.В. Горбачев – старший
 преподаватель кафедры ИСиТ ИИТ
 БГУИР, м.т.н.

Минск 2021

СОДЕРЖАНИЕ

Введение	3
1 Анализ литературы по теме дипломного проекта	4
1.1 Описание предметной области	4
1.2 Обзор существующих аналогов.....	6
1.3 Постановка цели и задач	12
1.4 Входные данные	12
1.5 Выходные данные	13
2 Моделирование предметной области.....	14
2.1 Функциональные требования.....	14
2.2 Описание функциональности программного обеспечения	14
2.3 Разработка информационной модели	16
2.4 Разработка модели взаимодействия пользователя с интерфейсом.....	18
2.5 Разработка технических требований к программному обеспечению....	19
3 Проектирование программного обеспечения.....	20
3.1 Обоснование выбора среды разработки	20
3.2 Разработка структурной схемы программного обеспечения	23
3.3 Разработка физической модели данных	24
Заключение	29
Список использованных источников	30

ВВЕДЕНИЕ

Актуальность учета компьютерной и другой техники в организации возрастает при наличии большого количества компьютеров, офисной техники и другого оборудования.

Автоматизация сфер деятельности организации, будь то сбор данных или управление различными технологическими процессами, позволяет ускорить работу, сделать ее более точной и эффективной, избежать потерь нужной информации, ошибок персонала, дублирования документов. Для эффективности работы по учету техники необходимо проводить мероприятия по обновлению материально-технической и информационной базы, внедрению современных методов управления с использованием автоматизированных систем, созданию базы данных о технике организации, его своевременному пополнению, оперативному представлению необходимой информации пользователям.

Целью дипломного проекта является разработка веб-приложения для учета мобильных устройств ЗАО «Кьюликс Системс».

Задачи дипломного проекта:

- изучить аналоги программного обеспечения по учету компьютерной либо любой другой техники в организации;
- спроектировать информационную структуру разрабатываемого программного обеспечения;
- разработать схему базы данных и пользовательский интерфейс для доступа к данным;
- провести расчет технико-экономических показателей разрабатываемого программного обеспечения.

Объектом дипломного проекта является процесс учета мобильных устройств ЗАО «Кьюликс Системс».

Предметом дипломного проекта является автоматизация учета мобильных устройств ЗАО «Кьюликс Системс».

Практическая значимость дипломного проекта состоит в том, что использование веб-приложения сократит трудозатраты пользователей за счет снижения трудоемкости выполнения «ручных» операций и сокращения времени на устранение инцидентов (сбой в работе устройства, нежелательное обновление операционной системы).

1 АНАЛИЗ ЛИТЕРАТУРЫ ПО ТЕМЕ ДИПЛОМНОГО ПРОЕКТА

1.1 Описание предметной области

Каждая организация приходит к тому, что требуется навести полный порядок в инфраструктуре организации, получить контроль над перемещениями техники, спланировать сервисные работы, заказы и закупки, списание техники. Необходимо иметь возможность в любой момент централизованно получать полную информацию о состоянии подотчетного оборудования.

Учет – это количественное отражение событий и фактов, используемое для управления обществом на различных организационных уровнях. Учет представляет собой сбор, обработку, классификацию, систематизацию и отражение информации в специальных регистрах на каких-либо носителях. В экономике используются различные виды учета: бухгалтерский, оперативно-технический, статистический и другие. Общая цель всех видов учета – упорядочение большого количества информации для эффективного использования в управленческих решениях, формирования отчетности и сохранение информации для архива [3].

Оперативная информация используется для принятия управленческих решений, составления и анализа финансовой отчетности организации, долгосрочного и текущего планирования, исполнения налоговых обязательств. Отличительной особенностью оперативной информации является ее конфиденциальность. Оперативная информация должна быть доступна только для сотрудников организации. Управленческий (оперативный) учет – это внутреннее дело каждой организации. Для ведения этого учета могут быть использованы любые формы и регистры, разработанные самой организацией [4].

Все виды учета являются составной частью интегрированной управленческой информационной системы. При создании такой системы необходимо ответить на следующие вопросы:

- какая информация нужна (информация различается по степени детализации и стилю представления. Степень детализации оперативной информации зависит от иерархии управления. Детализация и стиль представления оперативной информации зависят от получателя. Высшему управленческому персоналу требуется обобщенная оперативная информация. Она должна быть наиболее интегрированной и с более длительным периодом представления. Обобщенная оперативная информация может представляться ежемесячно);

- кому нужна эта информация (оперативная информация предназначена для внутреннего пользования. Состав пользователей информации определяется структурой организации, которая зависит от вида деятельности. Пользователей информации можно разделить на две

составляющие: управленческий персонал и непосредственно исполнители);

- почему нужна эта информация (оперативная информация необходима для принятия оперативных управленческих решений управленческим персоналом, реализации управленческих решений специалистами организации, определения отклонений достигнутых результатов от прогнозируемых, разработки мероприятий по выявленным отклонениям);

- когда она необходима (информация должна быть достоверной и достаточной для принятия управленческих решений);

- где получить информацию (информацию различают как первичную и вторичную. Источник информации, определяемый схемой документооборота, утвержденной руководством организации, можно отнести к вторичной информации. Внешними источниками информации могут быть поставщики и покупатели, конкуренты, реклама, государственные органы и др. Таким образом, наблюдение за внешней средой позволяет собрать первичную информацию, необходимую для бизнеса) [4].

Проектирование программного обеспечения должно предусматривать использование информационных технологий и способы обмена информацией.

Требования к качеству информации. Основными принципами учета оперативной информации являются качественные характеристики этой информации и принципы ее учета. Основное требование к учетной информации – это ее полезность для принятия решений различными группами пользователей. Полезность информации определяется специальными характеристиками:

- ясность;
- уместность;
- содержательность;
- существенность;
- своевременность;
- полнота отражения;
- логичность;
- достоверность [4].

Таким образом, качественные характеристики оперативной учетной информации определяют полезность оперативной отчетности для пользователей. В нахождении оптимального сочетания всех характеристик проявляется профессионализм специалистов по контролю и экономистов-аналитиков. Соблюдение всех вышеперечисленных требований к качеству информации должно сочетаться с ограничением затрат и выгоды, которое состоит в том, что выгода от информации должна быть больше, чем затраты на ее получение. Выполнение требований к качеству оперативной учетной информации должно дать в результате правдивую, полную, полезную оперативную отчетность о деятельности организации.

1.2 Обзор существующих аналогов

1.2.1 Программное обеспечение «Hardware Inspector»

Программное обеспечение (ПО) «Hardware Inspector» предназначено для инвентарного учета компьютеров и другой техники в организациях. Ориентировано на выполнение работ, связанных с учетом и планированием компьютерного (и иного) аппаратного обеспечения. Оно позволяет всегда быть в курсе всей информации о компьютерном парке компании, получать разнообразные отчеты, планировать его обслуживание, ремонт и обновление [5].

Программное обеспечение «Hardware Inspector» решает задачи автоматизации инвентарного учета компьютерной техники и комплектующих, с возможностью хранения всей истории перемещений и обслуживания. Механизм ревизий рабочих мест предохраняет компьютеры и комплектующие от хищения и подмены. Детальный контроль за параметрами конфигурации компьютера обеспечивает свободу и оперативность действий по планированию модернизации и перераспределения устройств [5].

Функционал «Hardware Inspector» не ограничивается одним лишь учетом компьютеров – система позволяет автоматизировать почти все аспекты деятельности ИТ-подразделения. Набор из более чем 30 встроенных отчетов позволит оценить эффективность его работы и предоставить нужные отчеты вышестоящему руководству.

Основные возможности программного обеспечения «Hardware Inspector»:

- учет комплектующих;
 - отслеживание истории перемещения устройств, их ремонта, профилактики и инвентаризации;
 - возможность не только ручного ввода данных, но и импорта информации из отчетов программ анализа конфигурации компьютеров AIDA32, EVEREST, ASTRA И ASTRA32;
 - учет лицензий на программное обеспечение;
 - мониторинг инсталляции пользователями запрещенного ПО;
 - мониторинг изменений, происходящих в конфигурациях компьютеров (аудит рабочих мест);
 - учет заявок от пользователей;
 - инвентаризация устройств с использованием штрих-кодов;
 - гибкое разграничение доступа к данным;
 - мощные механизмы поиска устройств, лицензий и прочего;
 - большой набор настраиваемых отчетов, экспорт в различные форматы;
 - поддержка многопользовательской работы с базой данных в сети [5].
- Уникальность ПО «Hardware Inspector» заключается в возможности

вести учет не просто текущего состояния параметров компьютера, а всей истории жизни каждого устройства.

Механизм поиска поможет найти интересующее устройство (группу устройств) по заданному критерию. В фильтре поиска могут участвовать такие параметры устройств, как тип, модель, производитель, продавец, инвентарный номер, серийный номер, произвольная строка поиска и так далее (смотрите рисунок 1).

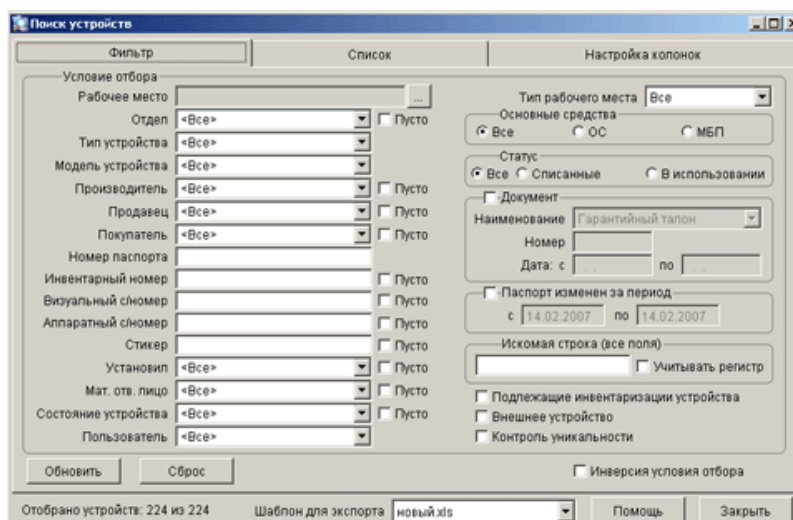


Рисунок 1 – Поиск устройства в «Hardware Inspector»

Достоинствами данной системы являются полный учет комплектующих, а не просто описание параметров рабочих станций; сопровождение системы после покупки и регулярное обновление системы (смотрите справочник типов и моделей устройств на рисунке 2).

Имеет гибкую систему разграничения прав доступа, большой набор встроенных отчетов, а также механизм создания отчетов, требуемых компании.

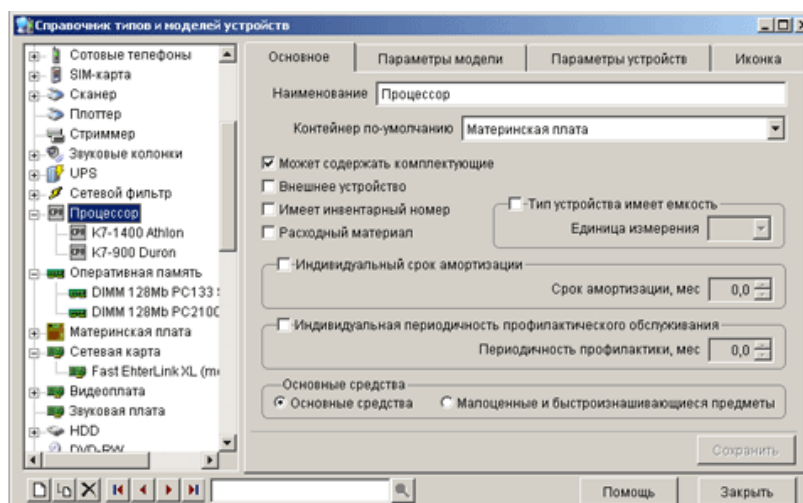


Рисунок 2 – Справочник типов и моделей устройств в «Hardware Inspector»

Данная система имеет закрытую архитектуру, в связи с чем отсутствует возможность вносить изменения в функционал системы. Такой недостаток не позволяет в полной мере внедрить систему в организацию, так же функционал, имеющийся в «Hardware Inspector», излишен для нужд ЗАО «Кьюликс Системс».

1.2.2 Программное обеспечение «Инвентаризация сети и учета и компьютеров»

Программное обеспечение «Инвентаризации сети и учет компьютеров» является разработкой компании Clear Apps.

Основная задача, которую выполняет система – это инвентаризация компьютерных сетей. В инвентаризацию сети входят учёт установленного программного обеспечения, а также регистрация всего аппаратного парка сети [6].

Данная система позволяет вести учёт всей компьютерной техники и программного обеспечения сети. Эта процедура просто необходима при планировании обновления аппаратного обеспечения, добавлении новых рабочих мест или закупки дополнительных аппаратных или программных ресурсов (смотрите главную страницу ПО на рисунке 3) [6].



Рисунок 3 – Главная страница ПЛ «Инвентаризации сети и учет компьютеров»

Система также предназначена для контроля использования аппаратных средств и лицензий ПО. Инвентаризация сети необходима для контроля соответствия установленного программного и аппаратного обеспечения с заданным стандартным набором для каждого рабочего места.

Данное программное обеспечение имеет систему отчётности выполнения плановых мероприятий по всей ИТ-инфраструктуре

компьютерной сети.

Имеется возможность добавлять в отчёты по инвентаризации компьютеры, которые не подключены к сети. Для этого в системе предусмотрен специальный модуль, обрабатывающий все данные, которые можно будет импортировать в интерфейс программы.

Для данной системы можно выделить следующие достоинства:

- автоматизация - сканирование сети осуществляется автоматически с высокой скоростью;
- наглядность – группировка собранной информации производится как для всей сети в целом, так и для каждого компьютера в отдельности;
- гибкость – имеется возможность добавлять специальные поля для каждого компьютера;
- удобство – собранные данные можно распечатать или экспортировать в нужный формат;
- отчёты – генератор отчётов и предустановленные шаблоны максимально ускорят процедуру построения отчётов по сети [6].

Недостатком системы можно выделить сложность ее внедрения: для внесения данных о компьютере, на него необходимо установить специальный модуль, сканирующий компьютер. Лишь после этого компьютер будет внесен в базу данных.

Кроме того, данный модуль проработан недостаточно хорошо, вследствие чего некоторые данные об оборудовании приходится вносить вручную. Для нужд ЗАО «Кьюликс Системс» данная система является слишком громоздкой и трудозатратой в плане переработки для собственных нужд.

1.2.3 Программное обеспечение «Учет компьютеров»

Программное обеспечение предназначено для учета компьютеров, оргтехники и других объектов в организации, разработана компанией ООО «Простой софт».

ПО позволяет фиксировать производимые ремонты, замены расходных материалов и частей, профилактические работы, установленное программное обеспечение. Все объекты учета можно закреплять за сотрудниками с разбиением на отделы и филиалы (рабочее окно ПО «Учет компьютеров» смотрите на рисунке 4) [6].

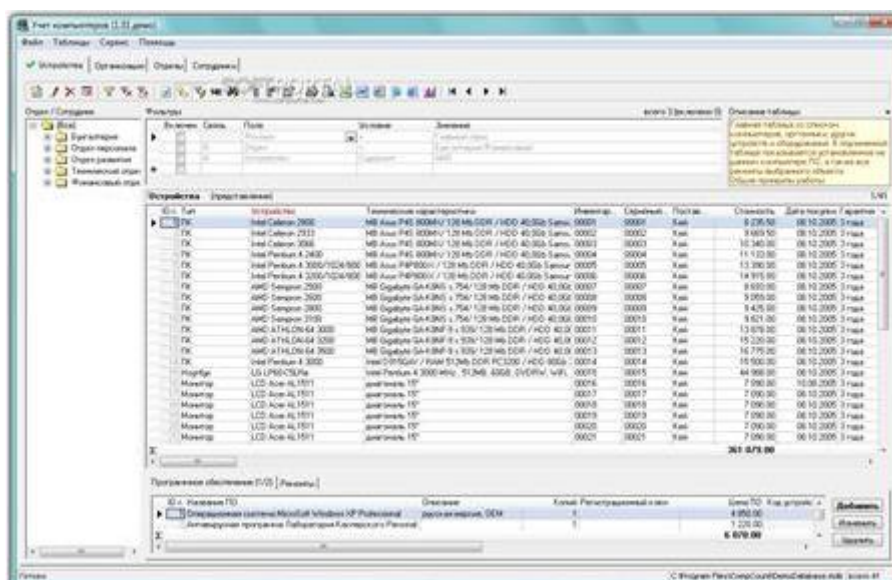


Рисунок 4 – Рабочее окно программы «Учет компьютеров»

ПО необходимо системным администраторам, руководителям ИТ – отделов, владельцам компьютерных магазинов, держателям домашних сетей, обладателям большого количества комплектующих и компьютерной техники.

Основные функции:

- ведение базы компьютеров и оргтехники: в базе данных содержится информация обо всех компьютерах и оргтехнике организации;
- предусмотрены удобные способы сортировки и фильтрации данных;
- поддерживается экспорт таблиц в MS Excel, MS Word или текстовый формат CSV, а также импорт из других источников данных;
- фиксация ремонтов и апгрейдов, а также контроль гарантийных сроков;
- учет установленного программного обеспечения;
- закрепление объектов за ответственными сотрудниками: все объекты учета можно закреплять за ответственными сотрудниками с разбиением на отделы и филиалы [6].

Данная система позволяет не только вести учет оборудования, но и составлять бюджет отдела: спланировать потребность в необходимом количестве расходных материалов на определённый период, комплектующих для резерва на случай возможных отказов и тому подобное.

Многопользовательский режим работы позволяет распределить зоны ответственности: кто-то будет заносить в базу информацию о новом поступившем оборудовании, кто-то о поставщиках и прочих деловых партнерах. В результате обеспечивается согласованная работа сотрудников различных подразделений. Права на просмотр и изменение информации в системе настраиваются администратором в зависимости от функциональных обязанностей каждого пользователя.

Преимущества системы можно выделить следующие возможности и

функции:

- возможность изменения структуры данных – добавления таблиц, полей, связей, что очень важно при внедрении системы на предприятие;
- возможность создания шаблонов отчетов (счетов, актов и других документов);
- используется генерация документов Word на основе шаблонов (счетов, актов и других документов);
- поддерживается многопользовательский режим;
- настраиваемый пользовательский интерфейс [6].

Но при этом, анализируя отзывы пользователей, было выявлено, что при большом компьютерном парке работа базы данных и системы в целом начинает работать медленно, возникают различные ошибки и прерывания в работе системы. Для нужд ЗАО «Кьюликс Системс» данная система является слишком громоздкой и трудозатратой в плане переработки для собственных нужд.

1.2.4 Вывод

Таким образом, проанализировав данные разработки, можно сделать следующие выводы.

Все системы используют автоматическое сканирование компьютеров для получения информации обо всех компьютерах сети. Для этого могут быть использованы как сторонние приложения (Everest, ASTRA и тому подобные), запускаемые на рабочих местах удаленно с помощью скриптов, так и встроенные модули системы.

Аналогичным образом происходит инвентаризация программного обеспечения, установленного на компьютерах сети предприятия. Данная возможность также характерна для всех рассматриваемых систем и является неотъемлемой частью системы учета компьютеров.

Все системы поддерживают интеграцию с различными базами данных, а также возможность импорта отчетов во все распространенные форматы файлов. Кроме того, в базах данных рассмотренных систем предусмотрена возможность привязки каждого компьютера к конкретному сотруднику и отделу.

Обязательной характеристикой оборудования при инвентаризации является информация о гарантийном сроке, дате поступления и поставщике оборудования.

Также неотъемлемой частью всех систем учета является генерация разнообразных отчетов. При этом имеются не только встроенные, шаблонные, отчеты, но и у большинства систем имеется возможность создания собственных, индивидуальных, отчетов.

Анализируя выделенные у данных систем достоинства и недостатки, можно сказать, что нет необходимости в разработке или же в приобретении

сторонних громоздких систем для учета мобильных устройств для нужд ЗАО «Кьюликс Системс», так как основная цель разрабатываемой системы – это учет местоположения девайса и сокращение времени реакции на какие-либо инциденты, связанные с девайсом, за счет того, что все отметки будут вестись в одном месте. Так же нет необходимости в предоставлении отчетности руководству о перемещениях мобильных устройств между сотрудниками.

1.3 Постановка цели и задач

Как показал анализ программных решений в области учета компьютерной техники, они обладают следующими недостатками:

- закрытая архитектура, в связи с чем отсутствует возможность вносить изменения в функционал системы;
- сложность внедрения и переработки имеющегося функционала ПО для нужд ЗАО «Кьюликс Системс»;
- громоздкость, то есть имеющийся функционал не нужен для учета мобильных устройств в ЗАО «Кьюликс Системс»;
- высокая стоимость.

Целью дипломного проекта является разработка веб-приложения для учета мобильных устройств ЗАО «Кьюликс Системс».

ПО должно обеспечивать выполнение следующих функций:

- авторизация сотрудника;
- добавление, редактирование и удаление информации о сотрудниках администратором;
- добавление, редактирование и удаление информации о мобильных устройствах администратором;
- изменение статуса мобильного устройства («Связан», «В ящике»);
- просмотр информации о мобильных устройствах;
- просмотр информации о сотрудниках (доступно только администратору);
- комментирование состояния мобильного устройства (необходимо для быстрого информирования о состоянии мобильного устройства (сбой в работе, нежелательное обновление операционной системы (ОС))).

1.4 Входные данные

К входной информации будут относиться все вводимые сотрудником и администратором данные.

Администратор при добавлении и редактировании информации о сотрудниках и мобильных устройствах будет указывать о них соответствующую информацию, также при авторизации сотрудник будет указывать свою персональную информацию (логин и пароль).

Так же сотрудник может оставлять комментарии по поводу состояния мобильного устройства.

Входной информацией будут являться:

- данные для добавления, редактирования и удаления информации о сотрудниках;
- данные для добавления, редактирования и удаления информации о мобильных устройствах;
- данные для авторизации в системе;
- комментарии по статусу мобильного устройства.

1.5 Выходные данные

В качестве выходной информации будут выступать данные после добавления и сохранения отредактированной информации о сотрудниках и мобильных устройствах.

Система будет отдавать в качестве выходной информации список сотрудников (доступно только для администратора) и список мобильных устройств.

При изменении статуса мобильного устройства будет назначаться соответствующее мобильное устройство на сотрудника, который его взял.

К выходной информации также относятся системные данные: сессии и токены авторизации, ошибки системы.

К выходной информации относится:

- статус мобильного устройства;
- сессия и JWT;
- данные о сотрудниках;
- данные о мобильных устройствах;
- ошибки.

2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1 Функциональные требования

Функциональным назначением разрабатываемого веб-приложения является предоставление возможности вести учет информации о мобильных устройствах ЗАО «Кьюликс Системс».

В качестве пользователя программного обеспечения может выступать сотрудник ЗАО «Кьюликс системс», который добавлен администратором в данную систему и имеет доступ к персональному компьютеру с доступом к сети интернет. Для использования программного обеспечения не требуется специальная подготовка или обучение пользователей.

Предполагается возможность одновременной эксплуатации разрабатываемого решения широким кругом пользователей.

Исходя из предполагаемого использования, можно заключить, что проектируемое программное обеспечение должно реализовывать следующие группы функций:

- авторизация в системе;
- добавление, редактирование и удаление информации о сотрудниках администратором;
- добавление, редактирование и удаление информации о мобильных устройствах администратором;
- изменение статуса мобильного устройства («Связан», «В ящике»);
- просмотр информации о мобильных устройствах;
- просмотр информации о сотрудниках (доступно только администратору);
- комментирование состояния мобильного устройства (необходимо для быстрого информирования о состоянии мобильного устройства (сбой в работе, нежелательное обновление операционной системы (ОС))).

2.2 Описание функциональности программного обеспечения

Диаграмма вариантов использования – это наиболее общее представление функционального назначения системы.

Диаграмма вариантов использования призвана ответить на главный вопрос моделирования: «что делает система во внешнем мире?». На диаграмме применяются два типа основных сущностей: варианты использования и действующие лица, между которыми устанавливаются следующие основные типы отношений:

- ассоциация между действующим лицом и вариантом использования;
 - обобщение между действующими лицами;
 - обобщение между вариантами использования;
 - зависимости (различных типов) между вариантами использования
- [7].

В системе можно выделить следующие действующие лица:

- держатель (сотрудник);
- администратор.

На рисунке 5 представлена общая диаграмма вариантов использования.

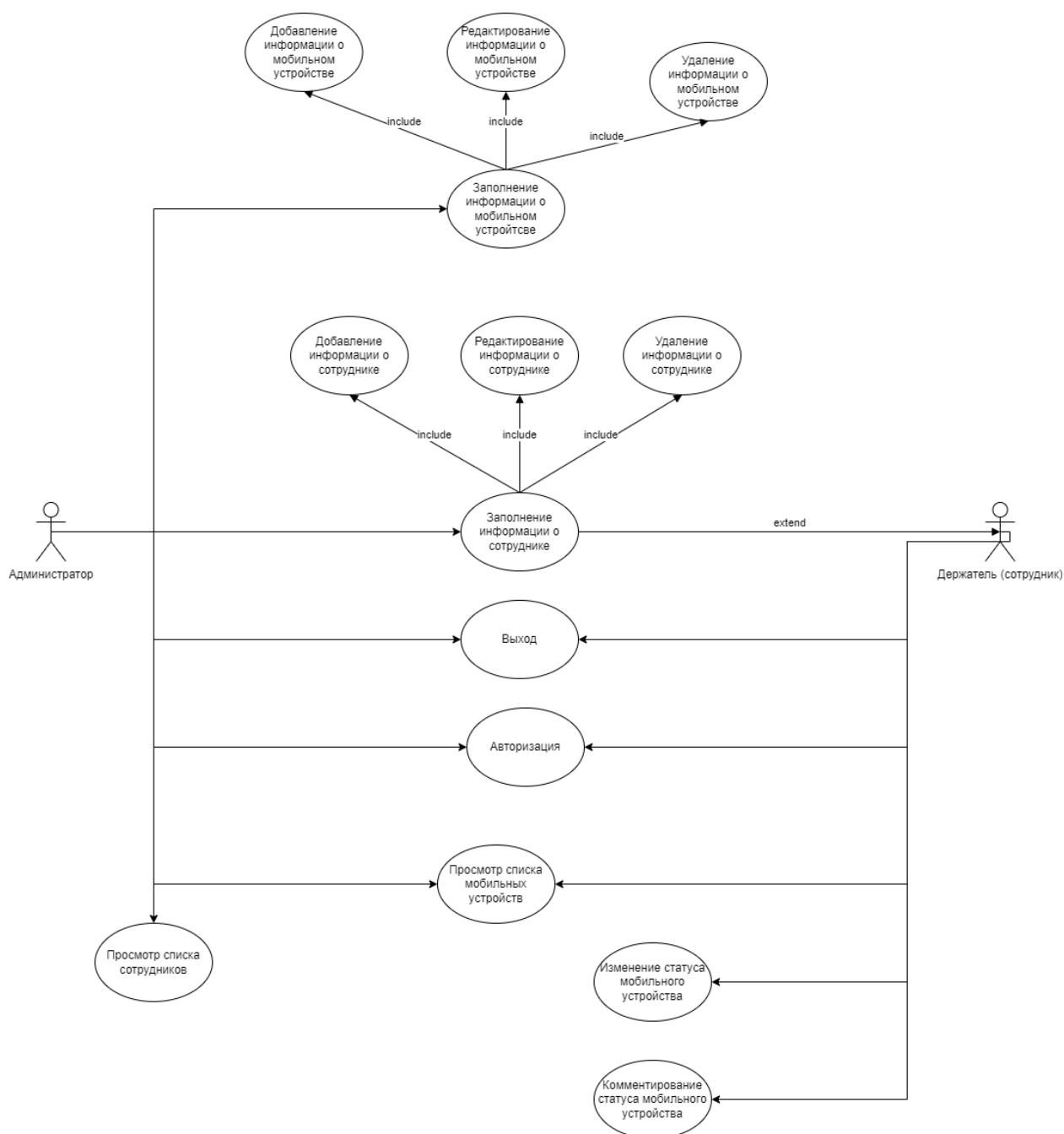


Рисунок 5 – Диаграмма вариантов использования

Учет мобильных устройств ЗАО «Кьюликс Системс» представляет собой процесс регистрации устройства, которое необходимо взять для работы, сотрудником в веб-приложении.

Основные термины, используемые при работе с программным обеспечением (ПО):

Веб-приложение – клиент-серверное приложение, в котором клиент

взаимодействует с веб-сервером при помощи браузера, которое выполняет функцию контроля пользования мобильными устройствами [1].

Мобильное устройство – переносное устройство, принадлежащее организации, с операционной системой IOS или Android.

Администратор – сотрудник ЗАО «Кьюликс Системс», который ведет учет мобильных устройств.

Держатель (сотрудник) – пользователь мобильного устройства в текущий момент времени.

Ящик – физическое общедоступное место хранения устройств, не связанных с держателем.

Штриховой код (штрихкод) – графическая информация, наносимая на поверхность, маркировку или упаковку изделий, предоставляющая возможность считывания её техническими средствами – последовательность чёрных и белых полос, либо других геометрических фигур, является идентифицирующим кодом веб-приложения, отсканировав который происходит переход к веб-приложению [2].

Короткое имя держателя – короткое имя сотрудника ЗАО «Кьюликс Системс», указанное в названии компьютера.

Статус – статус мобильного устройства в веб-приложении. Может иметь два активных состояния:

- «Связан» (связан с держателем);
- «В ящике» (находится в ящике).

После добавления администратором сотрудника в систему, он может авторизоваться в ней.

После авторизации сотруднику доступен следующий функционал:

- просмотр списка мобильных устройств;
- изменение статуса мобильных устройств;
- комментирование статуса мобильных устройств;
- выход из системы.

В системе присутствует роль «Администратор», которая включает в себя возможности «Держателя», а также другие возможности такие как:

- добавление, редактирование и удаление информации о сотруднике;
- добавление, редактирование и удаление информации о мобильных устройствах;
- просмотр информации о сотрудниках.

2.3 Разработка информационной модели

Информационный объект – это описание некоторой сущности предметной области реального объекта, процесса, явления или события. Информационный объект образуется совокупностью логически взаимосвязанных реквизитов, представляющих качественные и количественные характеристики сущности.

Проанализировав возможные пользовательские действия была спроектирована информационная модель базы данных (БД), представленная на рисунке 6.

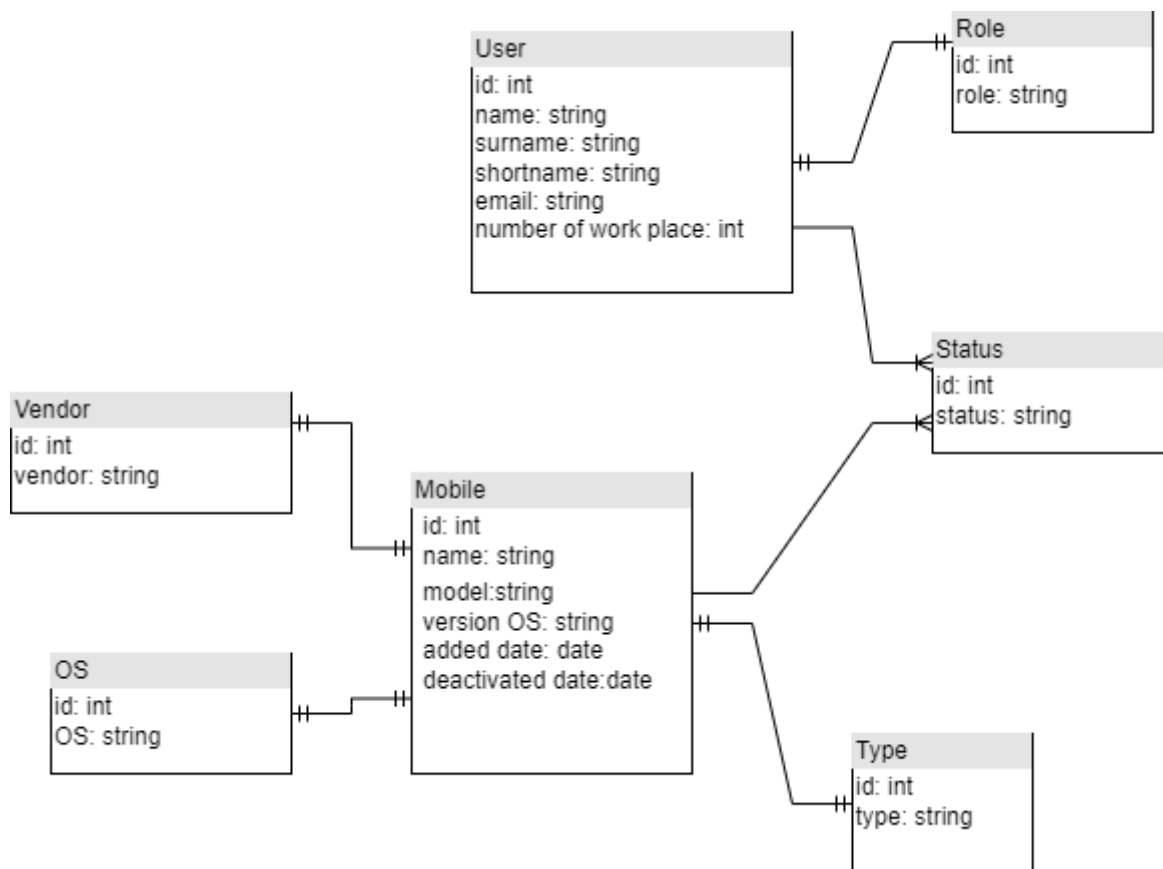


Рисунок 6 – Информационная модель БД

В процессе анализа предметной области были выделены следующие типы сущностей:

- **User** – сущность, хранящая пользовательские учетные записи с требуемой пользовательской информацией. Атрибуты: id – идентификатор сотрудника; name – имя сотрудника; surname – фамилия сотрудника; shortname – сокращенное имя сотрудника; email – почтовый адрес сотрудника; number of work place – номер рабочего места сотрудника;

- **Role** – сущность для хранения ролей в системе. Атрибуты: id – идентификатор роли; role – наименование роли;

- **Mobile** – сущность для хранения информации о мобильных устройствах. Атрибуты: id – идентификатор мобильного устройства; name – название мобильного устройства; model – модель мобильного устройства; version OS – версия операционной системы; added date – дата покупки мобильного устройства; deactivated date – дата деактивации мобильного устройства;

- **Status** – сущность для хранения статуса мобильного устройства. Атрибуты: id – идентификатор статуса мобильного устройства; status – статус мобильного устройства;

– Vendor – сущность для хранения информации о вендоре мобильного устройства. Атрибуты: id – идентификатор вендора мобильного устройства; vendor – наименование вендора мобильного устройства;

– OS – сущность для хранения информации об операционной системе мобильного устройства. Атрибуты: id – идентификатор операционной системы мобильного устройства; OS – наименование операционной системы мобильного устройства;

– Type – сущность для хранения информации о типе устройства. Атрибуты: id – идентификатор типа устройства; type – тип устройства.

2.4 Разработка модели взаимодействия пользователя с интерфейсом

Для архитектуры был выбран архитектурный шаблон MVC.

Шаблон архитектуры Model-View-Controller (MVC) разделяет приложение на три основных компонента: модель, представление и контроллер. Платформа ASP.NET MVC представляет собой альтернативу схеме веб-форм ASP.NET при создании веб-приложений. Платформа ASP.NET MVC является легковесной платформой отображения с широкими возможностями тестирования и, подобно приложениям на основе веб-форм, интегрирована с существующими функциями ASP.NET, например, с главными страницами и проверкой подлинности на основе членства. Платформа MVC определяется в сборке System.Web.Mvc. Схема шаблона разработки MVC представлен на рисунке 7.

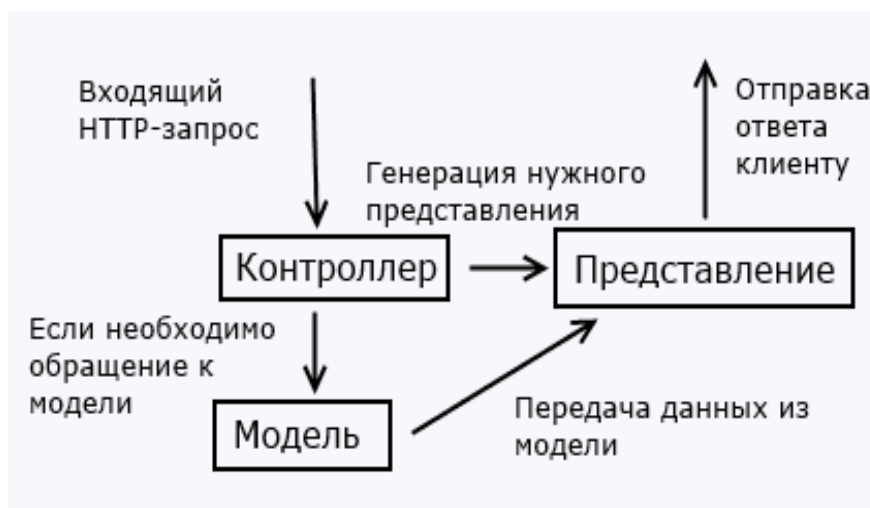


Рисунок 7 – Схема шаблона разработки Model View Controller

Шаблон разработки MVC позволяет создавать приложения, различные аспекты которых (логика ввода, бизнес-логика и логика интерфейса) разделены, но достаточно тесно взаимодействуют друг с другом. Эта схема указывает расположение каждого вида логики в приложении. Пользовательский интерфейс располагается в представлении. Логика ввода

располагается в контроллере. Бизнес-логика находится в модели. Это разделение позволяет работать со сложными структурами при создании приложения, так как обеспечивает одновременную реализацию только одного аспекта. Например, разработчик может сконцентрироваться на создании представления отдельно от бизнес-логики [8].

Связь между основными компонентами приложения MVC также облегчает параллельную разработку. Например, один разработчик может создавать представление, другой – логику контроллера, а третий – бизнес-логику модели.

Платформа ASP.NET MVC имеет следующие преимущества:

- облегчает управление сложными структурами путем разделения приложения на модель, представление и контроллер;
- не использует состояние просмотра и серверные формы. Это делает платформу MVC идеальной для разработчиков, которым необходим полный контроль над поведением приложения;
- использует схему основного контроллера, при которой запросы веб-приложения обрабатываются через один контроллер. Это позволяет создавать приложения, поддерживающие расширенную инфраструктуру маршрутизации;
- обеспечивает расширенную поддержку разработки на основе тестирования;
- хорошо подходит для веб-приложений, поддерживаемых крупными коллективами разработчиков, а также веб-разработчикам, которым необходим высокий уровень контроля над поведением приложения [8].

2.5 Разработка технических требований к программному обеспечению

Разрабатываемое программное обеспечение должно обеспечивать корректное функционирование при развертывании программных модулей на сервере со следующими техническими характеристиками:

- Intel 2.2 ГГц или более быстродействующий процессор;
- оперативная память 4 Гбайт или более;
- доступный объем дискового пространства 100 Гбайт.

Для нормального функционирования клиентской части программного обеспечения должны выполняться следующие технические требования:

- Pentium 2 ГГц или более быстродействующий процессор;
- оперативная память 2 Гбайт или более;
- 32- или 64-битная версия операционных систем Windows 10, 11;
- браузер Google Chrome версии 95.x.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Обоснование выбора среды разработки

В качестве среды разработки была выбрана: Microsoft Visual Studio 2019 Community.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server) [9].

Для написания данной системы была выбрана технология ASP.NET Core.

ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету. ASP.NET Core позволяет выполнять следующие задачи:

- создавать веб-приложения и службы, приложения IoT и серверные части для мобильных приложений;
- использовать избранные средства разработки в Windows, macOS и Linux;
- выполнять развертывания в облаке или локальной среде;
- работать в .NET Core или .NET Framework [10].

ASP.NET Core предоставляет следующие преимущества:

- единое решение для создания пользовательского веб-интерфейса и

веб-API;

- интеграция современных клиентских платформ и рабочих процессов разработки;
- облачная система конфигурации на основе среды;
- встроенное введение зависимостей;
- упрощенный высокопроизводительный модульный конвейер HTTP-запросов;
- возможность размещения в IIS, Nginx, Apache, Docker или в собственном процессе;
- параллельное управление версиями приложения, ориентированное на .NET Core;
- инструментарий, упрощающий процесс современной веб-разработки;
- возможность сборки и запуска в ОС Windows, macOS и Linux;
- открытый исходный код и ориентация на сообщество [10].

ASP.NET Core поставляется полностью в виде пакетов NuGet. Использование пакетов NuGet позволяет оптимизировать приложения для включения только необходимых зависимостей. На самом деле для приложений ASP.NET Core 2.x, ориентированных на .NET Core, требуется только один пакет NuGet. За счет небольшого размера контактной зоны приложения доступны такие преимущества, как более высокий уровень безопасности, минимальное обслуживание и улучшенная производительность [10].

C# – это изящный объектно-ориентированный язык со строгой типизацией, позволяющий разработчикам создавать различные безопасные и надежные приложения, работающие на платформе .NET Framework. C# можно использовать для создания клиентских приложений Windows, XML-веб-служб, распределенных компонентов, приложений клиент-сервер, приложений баз данных и т. д. Visual C# предоставляет развитый редактор кода, удобные конструкторы пользовательского интерфейса, интегрированный отладчик и многие другие средства, которые упрощают разработку приложений на языке C# для платформы .NET Framework.

Помимо этого, C# предлагает ряд инновационных языковых конструкций, упрощающих разработку программных компонентов. Инкапсулированные сигнатуры методов, именуемые делегатами, которые позволяют реализовать типобезопасные уведомления о событиях. Свойства, выполняющие функцию акцессоров для закрытых переменных-членов. Атрибуты, предоставляющие декларативные метаданные о типах во время выполнения. Внутри строчные комментарии для XML-документации. LINQ для создания запросов к различным источникам данных [10].

JavaScript – прототипно-ориентированный сценарный язык программирования. Является реализацией языка ECMAScript (стандарт ECMA- 262).

Основные архитектурные черты: динамическая типизация, слабая

типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, анонимные функции.

Достоинства JavaScript:

- ни один современный браузер не обходится без поддержки JavaScript;
- с использованием написанных на JavaScript плагинов и скриптов справится даже не специалист;
- полезные функциональные настройки;
- постоянно совершенствующийся язык;
- взаимодействие с приложением может осуществляться даже через текстовые редакторы – Microsoft Office и Open Office;
- перспектива использования языка в процессе обучения программированию и информатике [10].

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована [10].

Достоинства JQuery:

- скорость. Одной командой в JQuery можно сделать то, на что понадобится десятки строк простого JavaScript;
- понятность. Код и функции JQuery гораздо более удобны для чтения и понимания. Основная логика заключается в том, что сначала выбираем элементы, затем что-то с ними делаем. Поняв главное в JQuery, селекторы, мы можем делать впечатляющие вещи, и начинаем чувствовать всю магию на кончиках наших пальцев;
- кроссбраузерность;
- легкость отладки и тестирования. Данный пункт вытекает из предыдущего;

- возможности. Используя JQuery, можно подключать множество стандартных компонентов, называемых плагинами. Эти плагины можно по-своему оформлять, и они также будут работать во всех браузерах;
- поддержка. JQuery разрабатывается уже давно, разные версии JQuery совместимы друг с другом, над этим продуктом трудятся тысячи программистов.
- конкурентоспособность. Программист, использующий и знающий JQuery гораздо более высоко ценится на рынке труда [10].

3.2 Разработка структурной схемы программного обеспечения

В ходе моделирования было определено, что для разработки структурной схемы программного обеспечения будет использоваться архитектура MVC.

MVC представляет собой стандартный шаблон разработки, знакомый многим специалистам. Некоторые типы веб-приложений имеют преимущества при создании на платформе MVC. Для других может быть целесообразно использование традиционной схемы приложения ASP.NET, основанной на веб-формах и обратной передаче. В некоторых случаях возможно сочетание двух подходов: применение одной схемы не исключает использования другой [8].

В состав платформы MVC входят следующие компоненты.

Модели являются частями приложения, реализующими логику для домена данных приложения. Объекты моделей часто получают и сохраняют состояние модели в базе данных. Например, объект Product может получать информацию из базы данных, работать с ней, а затем записывать обновленные данные в таблицу Products базы данных SQL Server [8].

Представления служат для отображения пользовательского интерфейса приложения. Пользовательский интерфейс обычно создается на основе данных модели. Примером может служить представление для редактирования таблицы Products, которое содержит текстовые поля, раскрывающиеся списки и флажки, значения которых основаны на текущем состоянии объекта Product [8].

Контроллеры осуществляют взаимодействие с пользователем, работу с моделью, а также выбор представления, отображающего пользовательский интерфейс. В приложении MVC представления только отображают данные, а контроллер обрабатывает вводимые данные и отвечает на действия пользователя. Например, контроллер может обрабатывать строковые значения запроса и передавать их в модель, которая может использовать эти значения для отправки запроса в базу данных [8].

На рисунке 8 показана архитектура MVC приложения.

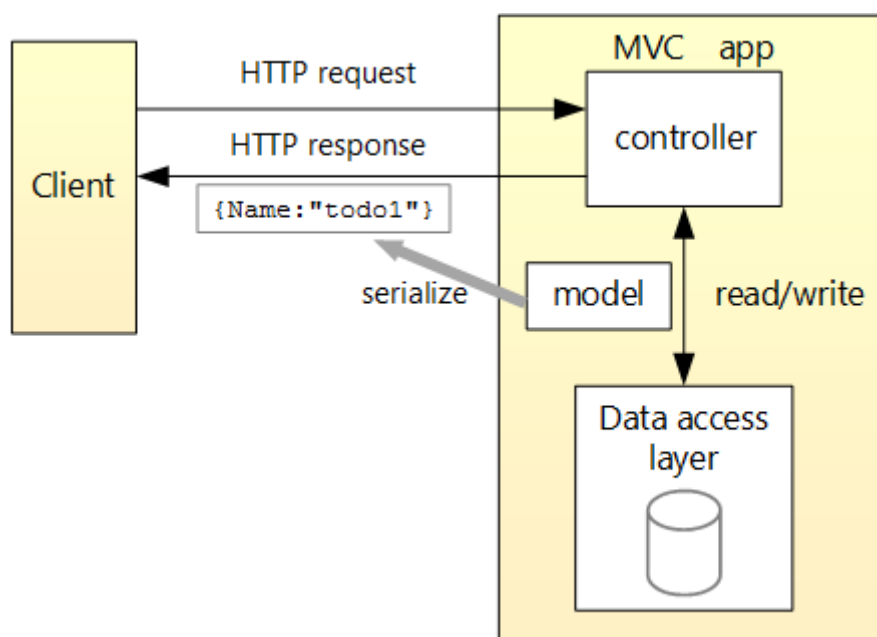


Рисунок 8 – архитектура MVC приложения

3.3 Разработка физической модели данных

На рисунке 9 представлена физическая модель базы данных, построенная на основе информационной модели базы данных. Данная модель жестко связана с конкретной системой управления базами данных. В данной модели представлены реальные типы данных, ограничения, реальные названия таблиц базы данных.

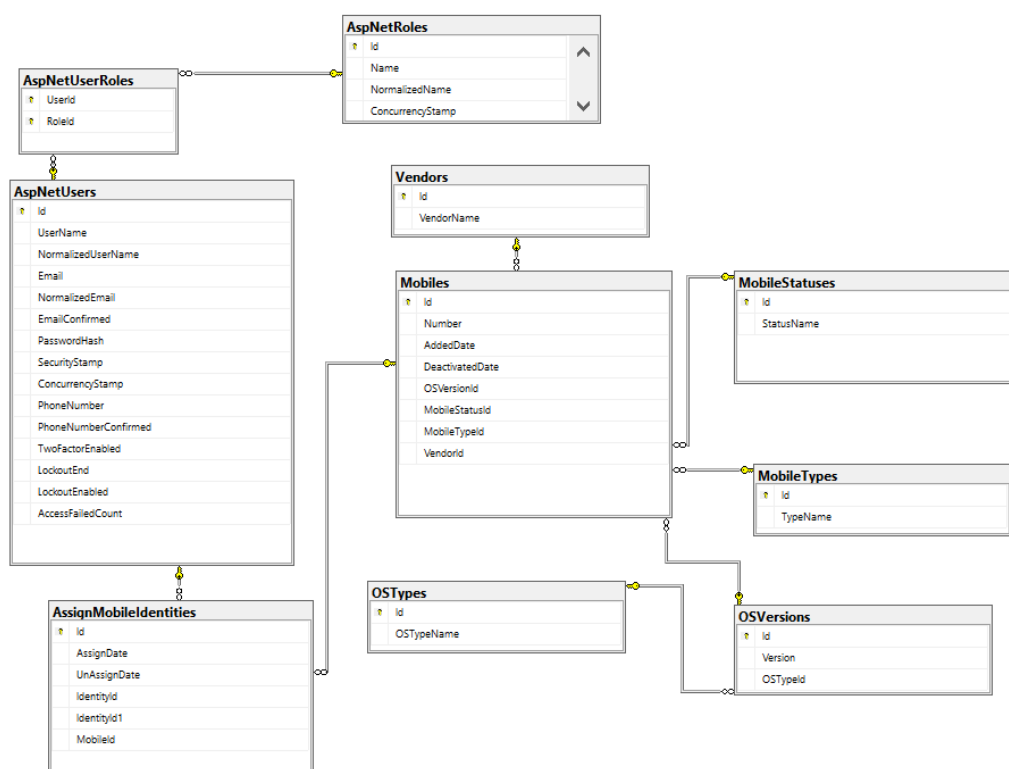


Рисунок 9 – Физическая модель БД

Формализованное описание объектов представлено в таблицах 1–10
Описание таблицы «AspNetUsers» представлено в таблице 1.

Таблица 1 – Описание таблицы «AspNetUsers»

Поле	Тип данных	Описание
Id	nvarchar(450)	Идентификатор сотрудника
UserName	nvarchar(256)	Имя сотрудника
NormalizedUserName	nvarchar(256)	Нормализованное имя сотрудника
Email	nvarchar(256)	Почтовый адрес сотрудника
NormalizedEmail	nvarchar(256)	Нормализованный почтовый адрес сотрудника
EmailConfirmed	bit	Флаг подтверждения почтового адреса сотрудника
PasswordHash	nvarchar(MAX)	Хеш пароля сотрудника
SecurityStamp	nvarchar(MAX)	Метка безопасности
ConcurrencyStamp	nvarchar(MAX)	Метка параллелизма
PhoneNumber	nvarchar(MAX)	Мобильный номер сотрудника
PhoneNumberConfirmed	bit	Флаг подтверждения мобильного номера сотрудника
TwoFactorEnabled	bit	Флаг двухфакторной аутентификации
LockoutEnd	datetimeoffset(7)	Дата окончания запрета авторизации сотрудника
LockoutEnabled	bit	Флаг запрета авторизации сотрудника
AccessFailedCount	int	Число попыток неудачного входа в систему

Описание таблицы «Mobiles» представлено в таблице 2.

Таблица 2 – Описание таблицы «Mobiles»

Поле	Тип данных	Описание
1	2	3
Id	int	Идентификатор мобильного устройства
Number	nvarchar(MAX)	Краткий идентификатор мобильного устройства
AddedDate	datetime2(7)	Дата покупки мобильного устройства
DeactivatedDate	datetime2(7)	Дата деактивации мобильного устройства

1	2	3
OSVersionId	int	Внешний ключ идентификатора версии операционной системы
MobileStatusId	int	Внешний ключ идентификатора статуса мобильного устройства
MobileTypeId	int	Внешний ключ типа мобильного устройства
VendorId	int	Внешний ключ вендора мобильного устройства

Описание таблицы «AspNetRoles» представлено в таблице 3.

Таблица 3 – Описание таблицы «AspNetRoles»

Поле	Тип данных	Описание
Id	nvarchar(450)	Идентификатор роли сотрудника
Name	nvarchar(256)	Наименование роли сотрудника
NormalizedName	nvarchar(256)	Нормализованное наименование роли сотрудника
ConcurrencyStamp	nvarchar(MAX)	Метка параллелизма

Описание таблицы «AspNetUserRoles» представлено в таблице 4.

Таблица 4 – Описание таблицы «AspNetUserRoles»

Поле	Тип данных	Описание
UserId	nvarchar(450)	Идентификатор сотрудника
RoleId	nvarchar(450)	Идентификатор роли сотрудника

Описание таблицы «Vendors» представлено в таблице 5.

Таблица 5 – Описание таблицы «Vendors»

Поле	Тип данных	Описание
Id	int	Идентификатор вендора
VendorName	nvarchar(MAX)	Наименование вендора

Описание таблицы «MobileStatuses» представлено в таблице 6.

Таблица 6 – Описание таблицы «MobileStatuses»

Поле	Тип данных	Описание
Id	int	Идентификатор статуса мобильного устройства
StatusName	nvarchar(MAX)	Наименование статуса мобильного устройства

Описание таблицы «MobileTypes» представлено в таблице 7.

Таблица 7 – Описание таблицы «MobileTypes»

Поле	Тип данных	Описание
Id	int	Идентификатор типа мобильного устройства
TypeName	nvarchar(MAX)	Наименование типа мобильного устройства

Описание таблицы «OSTypes» представлено в таблице 8.

Таблица 8 – Описание таблицы «OSTypes»

Поле	Тип данных	Описание
Id	int	Идентификатор типа операционной системы
OSTypeName	nvarchar(MAX)	Наименование типа операционной системы

Описание таблицы «OSVersions» представлено в таблице 9.

Таблица 9 – Описание таблицы «OSVersions»

Поле	Тип данных	Описание
Id	int	Идентификатор версии операционной системы
Version	nvarchar(MAX)	Версия операционной системы
OSTypeId	int	Внешний ключ идентификатора типа операционной системы

Описание таблицы «AssignMobileIdentities» представлено в таблице 10.

Таблица 10 – Описание таблицы «AssignMobileIdentities»

Поле	Тип данных	Описание
1	2	3
Id	int	Идентификатор

1	2	3
AssignDate	datetime2(7)	Дата назначения мобильного устройства на сотрудника
UnAssignDate	datetime2(7)	Дата снятия мобильного устройства с сотрудника
IdentityId	int	Внешний ключ идентификатора сотрудника
MobileId	int	Внешний ключ идентификатора мобильного устройства

ЗАКЛЮЧЕНИЕ

В ходе прохождения преддипломной практики были выполнены все поставленные цели. Была разобрана предметная область и проанализированы аналоги разрабатываемого программного обеспечения, определены их достоинства и недостатки. На основе анализа, была поставлены цели и задачи.

После анализа и определения целей и задач, а также постановки входных и выходных данных, была проведено моделирование предметной области. В ходе моделирования предметной были определены функциональные требования. Также была смоделирована информационная модель и модели взаимодействия пользователя с системой.

Завершив моделирование были спроектированы структурная схема программного обеспечения программы и физическая модель базы данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Веб-приложение [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5> Дата доступа: 02.11.2021
- [2] Штриховой код [Электронный ресурс]. – Электронный ресурс. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A8%D1%82%D1%80%D0%B8%D1%85%D0%BE%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%B4 Дата доступа: 02.11.2021
- [3] Виды учета [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://ta-aspect.by/project-materials/hozyajstvennyj-uchet> Дата доступа: 05.11.2021
- [4] Характеристика полной оперативной информации [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://infopedia.su/18x14bba.html> Дата доступа: 05.11.2021
- [5] ПО «Hardware Inspector» [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://www.hwinspector.com/> Дата доступа: 08.11.2021
- [6] ПО «Инвентаризация сети и учета и компьютеров» [Электронный ресурс]. – Электронный ресурс. – Режим доступа: https://studbooks.net/2226090/informatika/programmnyy_produkt_inventarizatsiya_seti_ucheta_kompyuterov Дата доступа: 09.11.2021
- [7] Диаграмма вариантов использования [Электронный ресурс]. – Электронный ресурс. – Режим доступа: http://book.uml3.ru/sec_1_5 Дата доступа: 11.11.2021
- [8] Общие сведения об ASP.NET MVC [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview> Дата доступа: 12.11.2021
- [9] Microsoft Visual Studio [Электронный ресурс]. – Электронный ресурс. – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio Дата доступа: 15.11.2021
- [10] Введение в язык C# и .NET Framework [Электронный ресурс]. – Электронный ресурс. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> Дата доступа: 16.11.2021