**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет СПО (название факультета)

Направление подготовки (специальность) 09.02.07 Информационные системы и программирование

**О Т Ч Е Т**

**о производственной практике (по профилю специальности)**

**по ПМ.11 «Разработка, администрирование и защита баз данных»**

(наименование практики)

Тема задания: Разработка прототипа автоматизированной информационной системы для архива документов

Обучающийся \_\_\_\_\_\_Корсунов В.А.\_\_\_\_\_\_ Группа \_\_\_Y2433\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.) (номер группы)

Руководитель практики от организации: Ильяшенко Александр Сергеевич, начальник отдела НИОКР ООО «Центр реактивного программирования»

(Фамилия И.О., должность и место работы)

Руководитель практики от факультета: Говорова М.М., преподаватель  
Ефимова Т.Н., преподаватель

(Фамилия И.О., должность и место работы)

Ответственный за практику от университета: Королёв В.В., заместитель директора факультета СПО

(Фамилия И.О., должность)

Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи членов комиссии\_\_\_\_\_\_\_\_\_\_\_( )

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Говорова М.М.)

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Ефимова Т.Н.)

(подпись)

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2020

Содержание

ВЕДЕНИЕ 3

1 проектирование и программная реализация 5

1.1 Методологии проектирования системы 5

1.2 Архитектура системы 7

1.3 Обзор программных средств 8

1.4 Моделирование функционального поведения системы и базы данных 9

1.5 Программная реализация 10

заключение 16

список источников 17

Приложение а 19

Приложение б 31

введение

Целью практики является совершенствование навыков разработки информационных систем, повышение компетенции в сфере программирования.

Местом прохождения практики является ООО «Центр реактивного программирования».

Целью проекта является создание прототипа автоматизированной информационной системы (далее – АИС) для архива документов, которая будет предназначена для:

* сбора статистики добавленных архивистами данных;
* управления данными о документах, фондах и описях;
* хранения информации о внесенных документах.

Заказчик имеет потребность в создании ПО для ведения архивного делопроизводства.

Объектом рассмотрения являются документы, получаемые от клиентов архива. Когда клиенты обращаются в архив, они передают большое количество печатной/электронной документации, с целью ее организации, сортировки и дальнейшего хранения. Верхним уровнем организации является фонд, в котором хранятся описи, которые в свою очередь уже содержат документы, различаемые по типу, срокам хранения и другим параметрам.

Клиентская база архива увеличивается с большой скоростью, вследствие чего, с каждым днем, становится все труднее вести учет всех данных в бумажном варианте. С целью ускорения процесса ведения архивной документации и оптимизации работы сотрудников архива, будет разработана АИС для хранения всей информации об архивных единицах.

В процессе прохождения практики по профилю специальности поставлены следующие задачи:

1. Знакомство с особенностями разработки программного обеспечения.
2. Приобретение навыков изучения предметной области, поиска объектов, требований к ним, а также установление отношений между сущностями, в рамках проектируемой БД.
3. Изучение требований к оформлению и составлению документации.
4. Проектирование и реализация БД.
5. Разработка программного продукта с подключаемой базой данных.
6. Получение опыта защиты проекта.

Практическая значимость реализуемой системы заключается в том, что разрабатываемый программный продукт в разы упростит основные операции, выполняемые сотрудниками архива, а также позволит собирать необходимую статистику о проведенной работе.

1. проектирование и программная реализация
   1. Методологии проектирования системы

[Методология](https://studopedia.ru/9_174185_metodologiya.html)**проектирования** информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (далее - ЖЦ) информационной системы (далее – ИС), представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Цель создания методологии построения информационных систем заключается в регламентации процесса проектирования ИС и обеспечении управления этим процессом с тем, чтобы гарантировать выполнение требований, как к самой ИС, так и к характеристикам процесса разработки.

Внедрение методологии должно приводить к снижению сложности процесса создания ИС за счет полного и точного описания этого процесса, а также применения современных методов и технологий создания ИС на всем жизненном цикле ИС - от замысла до реализации.

Системы, которые осуществляют хранение и обработку информации называют информационно-вычислительными системами. В информационную систему данные поступают от источника информации. Эти данные отправляются на хранение либо претерпевают в системе некоторую обработку и затем передаются потребителю.

Под проектированием ИС понимается процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в соответствии с ГОСТом в проект ИС. С этой точки зрения проектирование ИС сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла ИС: планирования и анализа требований, технического и рабочего проектирования, внедрения и эксплуатации ИС.

На этапе проектирования формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа требований к ИС. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных. Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Для построения моделей информационных систем используется методология UML (Unified Modelling Language). В методологии UML функции системы представляются на диаграмме прецедентов (диаграммы вариантов использования, Use Case диаграммы), где изображены акторы и действия, которые они могут выполнять в системе.

Для построения модели поведения системы используется методология DFD (Data Flow Diagram). Главная цель DFD - показать, как каждая работа преобразует свои входные данные в выходные, а также выявить отношения между этими работами. При создании DFD используются понятия:

* Процесс (англ. Process), т.е. функция или последовательность 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»). Здесь нет строгой системы требований, как, например, в IDEF0 или BPMN, где нотации имеют жестко определенный синтаксис, так как они могут быть исполняемыми. Но все же определенных правил стоит придерживаться, чтобы не вносить путаницу при чтении DFD другими людьми.
* Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных. Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.
* Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе. Поступившие данные перед обработкой и результат после обработки, а также промежуточные значения должны где-то храниться. Это и есть базы данных, таблицы или любой другой вариант организации и хранения данных. Здесь будут храниться данные о клиентах, заявки клиентов, расходные накладные и любые другие данные, которые поступили в систему или являются результатом обработки процессов.
* Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме.
* Сущность (англ. Entity). Это абстрактный объект, имеющий в конкретном контексте независимое существование, а связь – это ассоциация сущностей. Атрибут – свойство сущности или связи.

Модель «сущность – связь» (англ. Entity – Relationship, ER) позволяет описать логическую структуру базы данных. Известны несколько нотаций ER-моделей, наиболее популярной является IDEF1.

* 1. Архитектура системы

Для реализации Системы была выбрана архитектура «клиент-сервер».

Клиент-серверная архитектура реализует многопользовательский режим работы и является распределенной, когда клиенты и серверы располагаются на разных узлах локальной или глобальной вычислительной сети.

Существует несколько видов этой архитектуры:

* + - Архитектура "файл-сервер";
    - Двухуровневая архитектура "клиент-сервер";
    - Трехуровневая архитектура "клиент-сервер";
    - Многоуровневая архитектура "клиент-сервер".

Файл-серверная архитектура представляет наиболее простой случай распределенной обработки данных, согласно которой на сервере располагаются только файлы данных, а на клиентской части находятся приложения пользователей вместе с СУБД. Файл-сервер представляет собой достаточно мощную по производительности и оперативной памяти ПЭВМ, являющуюся центральным узлом локальной сети. Файл-сервер в среде сетевой операционной системы организует доступ к файлам, полностью эквивалентным файлам операционной системы и расположенным во внешней памяти файл-сервера.

Двухуровневая клиент-серверная архитектура основана на использовании только сервера базы данных (DB-сервера), когда клиентская часть содержит уровень представления данных, а на сервере находится база данных вместе с СУБД и прикладными программами.

Трехуровневая клиент серверная архитектура позволяет помещать прикладные программы на отдельные серверы приложений, с которыми через API-интерфейс (Application Program Interface) устанавливается связь клиентских рабочих станций. Работа клиентской части приложения сводится к вызову необходимых функций сервера приложения, которые называются «сервисами». Прикладные программы в свою очередь обращаются к серверу базы данных с помощью SQL запросов.

Многоуровневая архитектура «Клиент-сервер» создается для территориально-распределенных предприятий. Для нее в общем случае характерны отношения «многие ко многим» между клиентскими рабочими станциями и серверами приложений, между серверами приложений и серверами баз данных.

* 1. Обзор программных средств

Существует множество CASE-средств предназначенных для проектирования и документирования информационных систем и баз данных.

Для создания DFD-диаграммы используется CASE-средство, предназначенное для моделирования, анализа, документирования и оптимизации бизнес-процессов с использованием трех стандартных методологий: IDEF0 (функциональное моделирование), DFD (моделирование потоков данных) и IDEF3 (моделирование потоков работ) - AllFusion ERwin Process Modeler.

Данное программное обеспечение имеет низкий порог вхождения, большое количество обучающей документации, а также возможность работы по уровням и декомпозиции каждого процесса.

Для построения логической модели базы данных использовался MySQl WorkBench. Это инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных [MySQL](https://ru.wikipedia.org/wiki/MySQL" \o "MySQL). Продукт позволяет создавать модели в нотации Мартина, а также базы данных на основании этих моделей.

Drawio предоставляет собой инструмент, позволяющий создавать блок-схемы, сетевые диаграммы, интеллект-карты, отношения сущностей, программные блоки, UML, макеты и т. д. В списке функционала имеется возможность отслеживать и восстанавливать изменения, импортировать и экспортировать файлы различных форматов, делиться своими работами, также автоматически публиковать их.

Для проектирования системы были использованы следующие программные средства:

1. AllFusion ERwin Process Modeler.
2. MySQL WorkBench.
3. Drawio.
   1. Моделирование функционального поведения системы и базы данных

Для анализа системы были построены функциональная модель системы и модель поведения пользователей. В ходе функционального моделирования были выделены следующие основные процессы: добавление абонентов, городов, звонков, пополнений баланса; редактирование и удаление информации о абонентах и городах; просмотр количества и общей длительности звонков: всех, в регионы, в города, в определенный период. Построена диаграмма потоков данных, которая представлена в приложении Б.

Поведение пользователей определило двух актеров: администратора и архивиста. Диаграмма вариантов использования показана на рисунке П4 в приложении Б.

В ходе работы были выбраны следующие методы моделирования функционального поведения Системы и БД:

* моделирование диаграммы потоков данных и отображение процессов «вход-выход»;
* построение инфологической модели в нотации Мартина;
* создание реляционной базы данных по инфологической модели;
* нормализация моделей баз данных;

В ходе анализа данных были проанализированы объекты системы связи между ними и процессы, в которых участвуют объекты на предмет свойств этих связей и объектов. Были выделены основные объекты: абонент, город, пополнения баланса, звонки, определены их свойства.

* 1. Программная реализация

На основе сформированного технического задания разработано приложение для настольных компьютеров. Проведено детальное проектирование модулей системы и структур баз данных, после чего была проведена программная реализация, на основе средств, заявленных в техническом задании. Состав программы представлен в таблице 1.

Таблица 1 – Структура приложения

|  |  |  |
| --- | --- | --- |
| Интерфейс | Назначение | Функции |
| 1 | 2 | 3 |
| Auth\_interface | Интерфейс авторизации в систему. | Authorize(login, password) – авторизация в системе; |
| Operator\_interface | Интерфейс оператора, в котором он может проводит манипуляции над данными в таблице. | GetTableAbonent(); - получение таблицы с информацией об абонентах;  GetTableCity(); - получение таблицы с информацией о городах;  GetTableCall(); - получение таблицы с информацией о звонках;  GetTableReplenishment(); - получение таблицы с информацией об абонентах;  EditTableAbonent(Abonent); - редактирование таблицы с информацией об абонентах;  EditTableCity(City); - редактирование таблицы с информацией о городах;  DeleteAbonent(Abonent); - удаление информации об абоненте из таблицы;  DeleteCity(City); - удаление информации о городе из таблицы; |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Manager\_interface | Интерфейс менеджера, в котором он просмотреть статистику звонков. | GetCallsAll(); - получение статистики всех звонков;  GetCallsPeriod(); - получение статистики звонков за определенный период;  GetCallsZone(); - получение статистики звонков в определенный регион;  GetCallsCity(); - получение статистики звонков в определенный город; |

На рисунках 1-9 продемонстрированы скриншоты корректного выполнения данных функций авторизации, просмотра таблиц, редактирования таблиц, удаления информации из таблиц, просмотра статистики звонков.

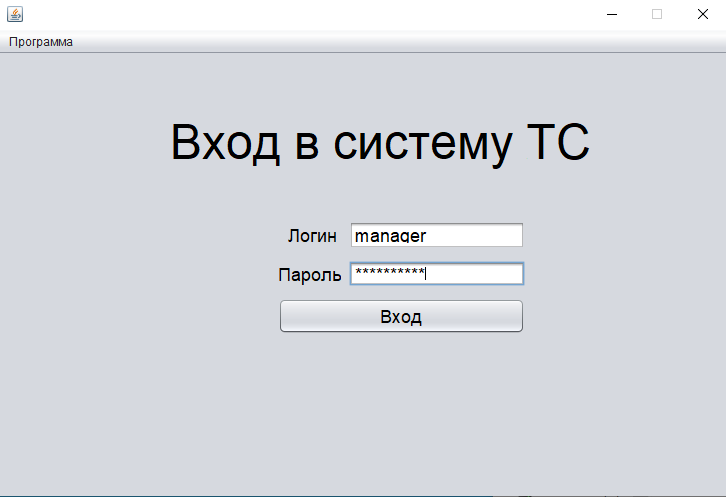


Рисунок 1 – Авторизация пользователя manager

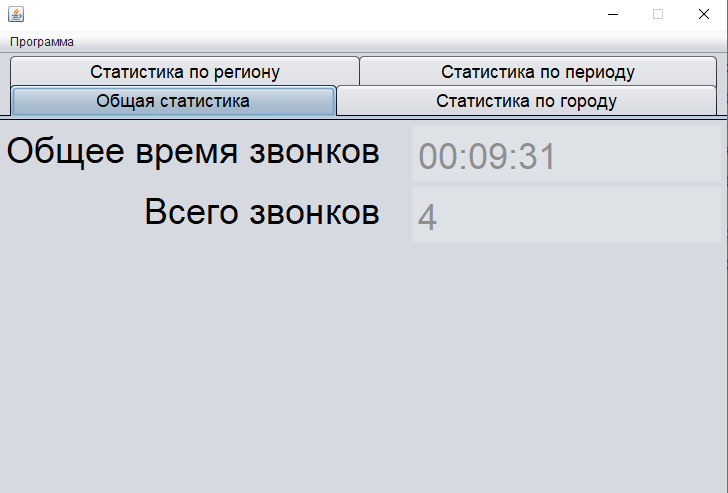


Рисунок 2 – Вывод статистики всех звонков

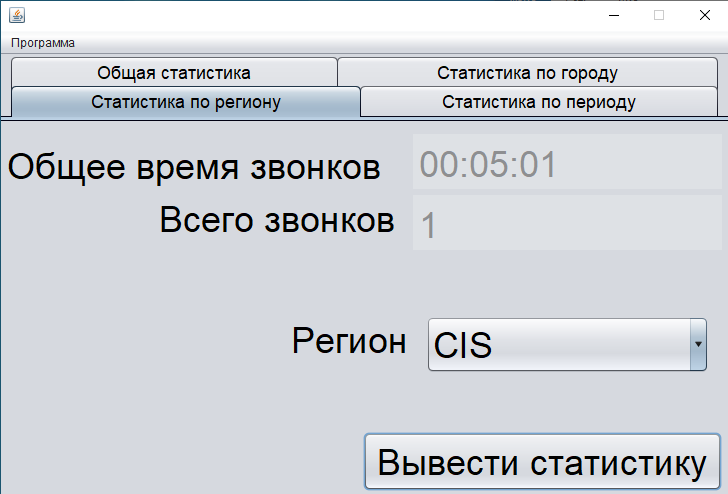


Рисунок 3 – Вывод статистики звонков для региона

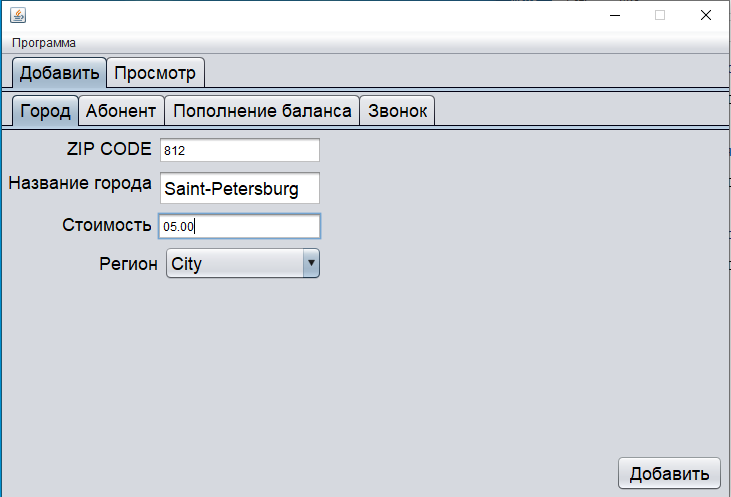


Рисунок 4 – Окно добавления города

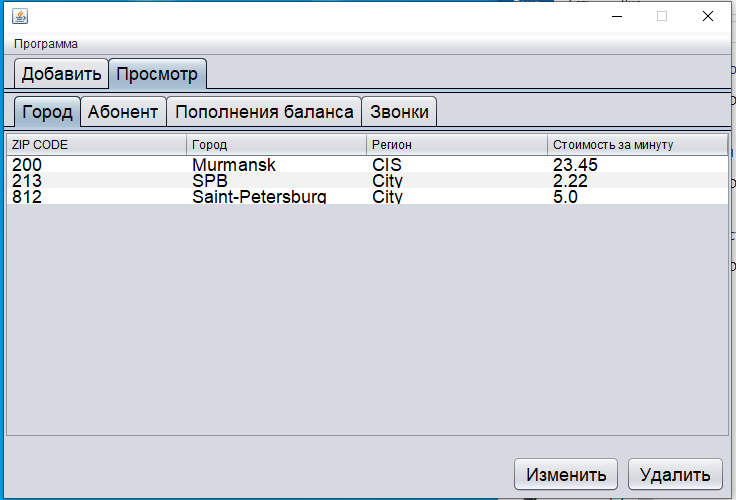


Рисунок 5 – Результат добавления города

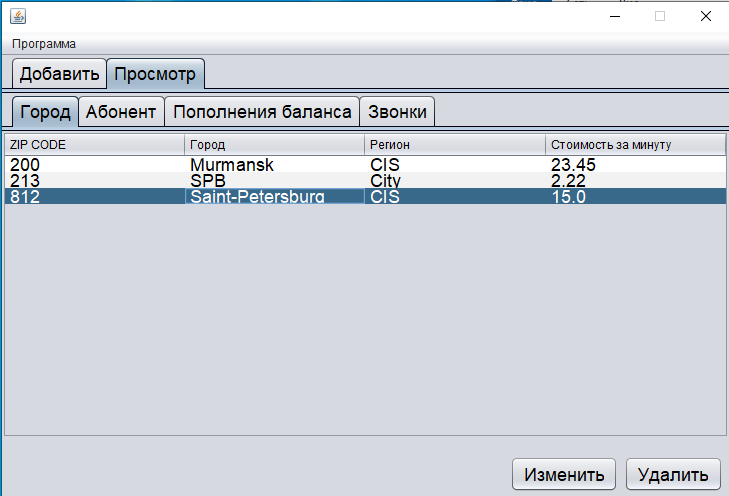
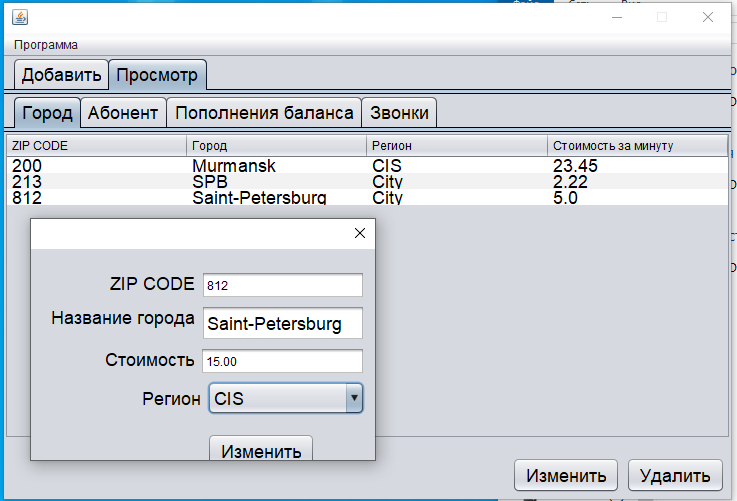


Рисунок 6 – Выбор поля для изменения/удаления города

Рисунок 7– Окно изменения города

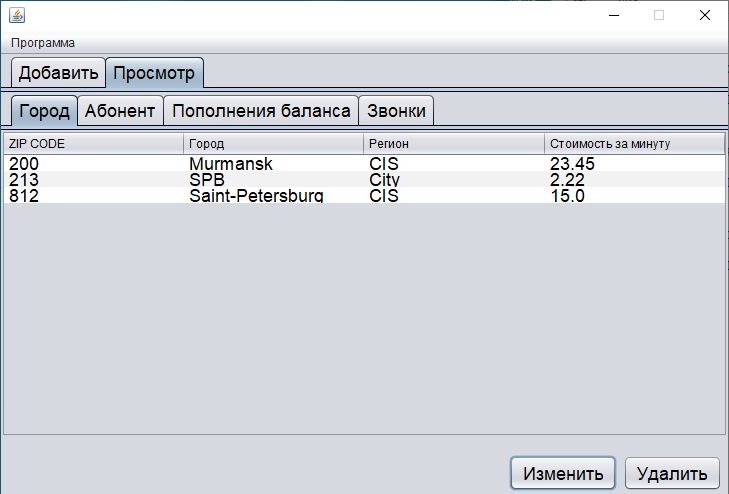


Рисунок 8 – Результат изменения города

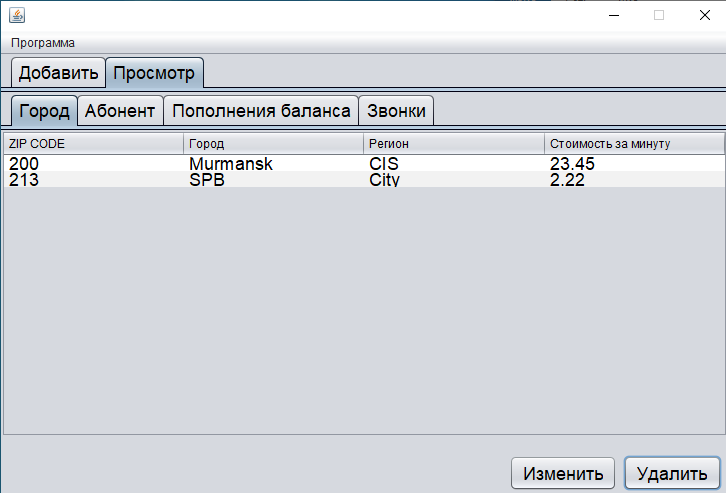


Рисунок 9 – Результат удаления города

заключение

Было определено место АИС в области управления данными ТС, изучен вопрос практической разработки АИС. Было уделено особое внимание стандартным функциям и процессам, которые должны быть реализованы. На основании изученного были разработаны начальные диаграммы процессов сбора и учета информации. На следующем этапе определялась платформа разработки АИС. На этапе разработки диаграммы были реализованы в модули программного продукта. Этап разработки логически завершается разработкой баз данных. После реализации модулей системы вся система, включая разработанные модули была перемещена на сервер. Следует отметить, что в системе разграничены права пользователей и рассматриваются две группы пользователей.

Таким образом все задачи, поставленные на практику, решены. Цели практики достигнуты.

Список источников

1. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. (Переиздание (Ноябрь 1987 г.) с Изменением № 1, утвержденным в июле 1981 г (ИУС 7-81)) [Электронный ресурс] // RUGOST1.0 - разработка документации по ГОСТ 34, 19, РД-50: [сайт]. URL: http://www.rugost.com/index.php?option=com\_content&view=article&id=54:19201-78&catid=19 (дата обращения: 10.10.2019)
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы [Электронный ресурс] // RUGOST1.0 - разработка документации по ГОСТ 34, 19, РД-50: [сайт]. URL. http://www.rugost.com/index.php?option=com\_content&view=article&id=96:gost-34602-89&catid=22&Itemid=53 (дата обращения: 10.10.2019)
3. Ульман Д.Д., Уидом Д. Реляционные базы данных. Лори, 2014. 384c.
4. Рудаков А.В., Федорова Г.Н. Технология разработки программных продуктов. Практикум. А.В.: Издательский центр «Академия», 2010. 187c.
5. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Вильямс, 2017. 1440c.
6. Проектирование информационных систем [Электронный ресурс] // Национальный открытый университет ИНТУИТ [сайт], 2003-2020. URL. <https://www.intuit.ru/studies/courses/2195/55/info> (дата обращения: 26.12.2019).
7. Учимся проектированию Entity Relationship — диаграмм [Электронный ресурс] // Хабр [сайт], 16.02.2019. URL. <https://habr.com/ru/post/440556/> (дата обращения: 26.12.2019).
8. Проектирование хранилищ данных для приложений систем деловой осведомленности (Business Intelligence Systems) [Электронный ресурс] // Национальный открытый университет ИНТУИТ [сайт], 2003-2020. URL. <https://www.intuit.ru/studies/courses/599/455/info> (дата обращения: 26.12.2019).
9. Основы работы с базами данных [Электронный ресурс] // Национальный открытый университет ИНТУИТ [сайт], 2003-2020. URL. <https://www.intuit.ru/studies/courses/93/93/info> (дата обращения: 26.12.2019).
10. Нормализация отношений. Шесть нормальных форм [Электронный ресурс] // Хабр [сайт], 2.05.2015. URL. <https://habr.com/ru/post/254773/> (дата обращения: 26.12.2019).

ПРИЛОЖЕНИЕ А

Техническое задание

1. Назначение разработки

Автоматизированная информационная система (далее – АИС) предназначена для сбора информации и отслеживания статистики телефонной станции (далее – ТС).

Целевой аудиторией АИС являются операторы и менеджеры ТС.

1. цель и задачи

Целью проекта является разработка прототипа АИС для реализации хранения информации звонков клиентов ТС.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить предметную область.
2. Проанализировать требования к программному продукту (ПП).
3. Спроектировать и разработать БД, применимую в соответствующей предметной области.
4. Реализовать пользовательский интерфейс.
5. Реализовать бизнес-логику ресурса.
6. Провести отладку и тестирование программного продукта (ПП).
7. Требования к разработке
8. Требования к функциональным характеристикам

Система должна иметь базу данных для хранения информации и подсистемы ввода-вывода данных, предусматривающих интерфейсы для работы пользователя.

В системе необходимо выделить 2 категории пользователей:

* менеджер;
* оператор.

В системе должны содержаться следующие функциональные модули:

* модуль базы данных, хранящий информацию об абонентах, звонках, городах и пополнениях баланса;
* модуль пользовательского интерфейса для операторов ТС, предназначенный для:
  + - просмотра информации об абонентах, городах, звонках и пополнениях баланса;
    - редактирования/удаления информации об абонентах, городах;
    - внесения информации о звонках, пополнениях баланса, городов и новых абонентов;
* модуль интерфейса, предназначенный для менеджера для просмотра совершенных абонентам звонков и вывода статистики вызовов.

Общий функционал должен обеспечивать следующие возможности в режиме доступа офлайн:

* авторизация по учетным данным;
* редактирование/удаление личных данных пользователей;
* редактирование/удаление информации о городах;
* отображение информации о профиле пользователя;
* отображение информации о городе;
* отображение информации о пополнениях баланса пользователей;
* разграничение привилегий и прав для категорий пользователей;
* получение информации о звонках, для сбора статистики;
* получение информации о звонках, для оператора;
* внесении информации о новых абонентах и городах;
* внесение информации о новых звонках;
* внесение информации о пополнении баланса абонента;

1. Разграничения ответственности ролей при доступе

В системе должны быть определены роли пользователей менеджер, оператор. Распределение прав пользователей согласно ролям, в системе представлено в таблице А.1.

Таблица А.1 Распределение прав пользователей согласно ролям

|  |  |  |
| --- | --- | --- |
| Роль  Функции | Менеджер | Оператор |
| 1 | 2 | 3 |
| Получение информации о звонках | - | + |
| Авторизация | + | + |
| Сбор статистики звонков | + | - |
| Редактирование/удаление информации об абонентах/городах | - | + |
| Внесение информации о новых абонентах, городах, звонках и пополнениях баланса | - | + |
| Просмотр информации об абонентах, городах и пополнениях баланса | - | + |

1. Требования к обработке аварийных ситуаций

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях пользователю должны выдаваться соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной команде или некорректному вводу данных.

1. Требования к защите информации от несанкционированного доступа

Система должна поддерживать разграничение прав доступа с возможностью формирования групп пользователей и присвоение группе и каждому пользователю определенных полномочий на доступ к данным системы.

Обеспечение информационной безопасности подсистемы должно удовлетворять следующим требованиям:

* защита от несанкционированного доступа вне локальной сети;
* обеспечение разделения прав доступа между пользователями.

1. Требования к видам обеспечения
2. Требования к информационному обеспечению

Структура хранения данных Системы должна состоять из области постоянного хранения данных, которая должна строиться на основе реляционной модели данных.

1. Требования по применению систем управления базами данных

Для реализации подсистемы хранения данных должна использоваться СУБД MySQL в совокупности с интерфейсом управления базой данных – MySQL Workbench.

1. Требования к лингвистическому обеспечению

При реализации системы должны применяться следующие языки высокого уровня: SQL, Java. Также для реализации графического интерфейса должна быть использована библиотека Swing.

Ресурс должен быть реализован на русском языке.

1. Требования к техническому обеспечению

АИС необходимо реализовать на компьютерах локальной сети ТС, которые в свою очередь должен удовлетворять следующим критериям:

* Наличие JAVA SE(не ниже 12 версии);

1. Требования к технологиям и средствам разработки

Разрабатываемое ПО должно быть АИС с графическим интерфейсом. Для разработки ПО и графического интерфейса должна быть использована среда Apache NetBeans 11.2. Для разработки информационного обеспечения должна использоваться технология баз данных с интерфейсом управления MySQL Workbench.

Для разработки должны использоваться следующие программные средства:

* + Набор программирования на языке Java – Java Development Kit (не ниже 11 версии);
  + СУБД MySQL (не ниже версии 5.5)
  + Библиотека для создания графического интерфейса Swing для Java.

1. Описание входных и выходных данных

Входные и выходные данные Системы должны удовлетворять таблице А.2.

Таблица А.2 Входные и выходные данные модулей АИС

|  |  |  |
| --- | --- | --- |
| Функции | Входные данные | Выходные данные |
| 1 | 2 | 3 |
| Получение информации о звонках | - | Таблица звонков |
| Получение информации о городах | - | Таблица городов |
| Получение информации о пополнениях баланса | - | Таблица пополнений |
| Получение информации об абонентах | - | Таблица абонентов |
| Авторизация | Логин, пароль | Идентификатор пользователя |
| Сбор статистики звонков | Период, регион или город | Время звонков, их количество |
| Редактирование информации об абонентах | Новые данные об абоненте | Измененные данные абонента |
| Редактирование информации о городах | Новые данные о городе | Измененные данные города |
| Удаление информации об абонентах | Абонент | Ответ об успешности операции |
| Удаление информации о городах | Город | Ответ об успешности операции |
| Внесение информации о новых абонентах, | Данные нового абонента | Новый абонент |
| Внесение информации о новых городах | Данные нового города | Новый город |
| Внесение информации о новых звонках | Данные нового звонка | Новый звонок, обновленный баланс абонента |
| Внесение информации о новых пополнениях баланса | Данные нового пополнения | Новое пополнение баланса, обновленный баланс абонента |

1. Модель разработки
2. Функциональная модель

Система должна удовлетворять функциональной модели, представленной в приложении Б, на рисунках Б.1 – Б.2.

Для представления поведения системы и всех возможных вариантов действий была использована диаграмма вариантов использования (UML Use Case). В диаграмме вариантов использования выделены следующие актёры:

* Менеджер;
* Оператор.

Роли пользователей в системе должны удовлетворять модели, представленной на рисунке Б.4 в приложении Б.

1. Модель данных

База данных должна удовлетворять реляционной модели, представленной в таблицах в приложении Б и на рисунке Б.3.

1. Требования к эргономике, технической эстетике и структуре

Взаимодействие пользователей с Системой должно осуществляться посредством приложения с графическим интерфейсом. Интерфейс должен обеспечивать удобный доступ к основным функциям и операциям. Разрабатываемая АИС должна функционировать на русском языке.

Страницы пользовательского интерфейса должны проектироваться с учетом требований унификации:

* страницы должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;
* в разделах интерфейса для обозначения сходных операций должны использоваться сходные графические значки, кнопки и т.п. управляющие (навигационные) элементы;
* внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки и т.п.) должны реализовываться одинаково для однотипных элементов.

Страница администратора включает:

* интерфейс добавления, удаления и редактирования городов/абонентов;
* интерфейс добавления пополнений баланса и звонков;
* интерфейс просмотра данных об абонентах, городах, пополнениях баланса и звонках;

Страница пользователя категории оператор включает интерфейс просмотра статистики звонков.

1. Методы тестирования соответствия разработки предъявленным требованиям

Тестирование должно осуществляться на модульном, интеграционном уровнях.

На этапе разработки программного обеспечения должна применяться технология тестирования «черного ящика» с использованием следующих видов тестирования:

* переходов между состояниями;
* функциональной эквивалентности.

Приложение б

Модель разработки

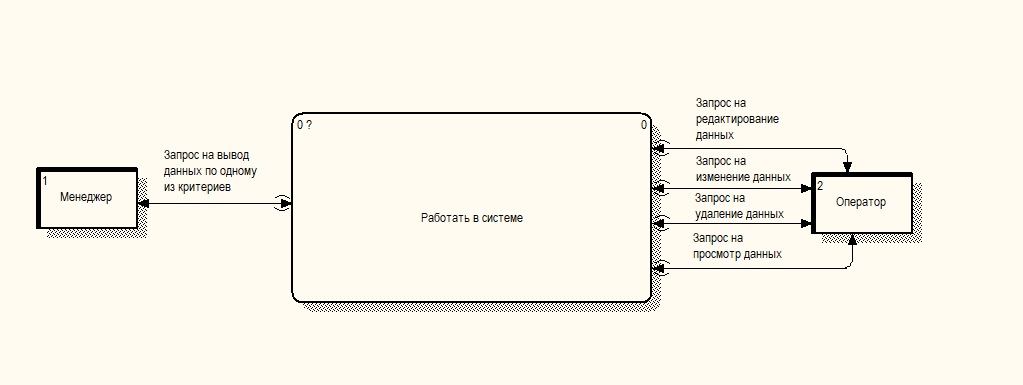


Рисунок Б.1 – Функциональная модель DFD: Работа в система, уровень А-0

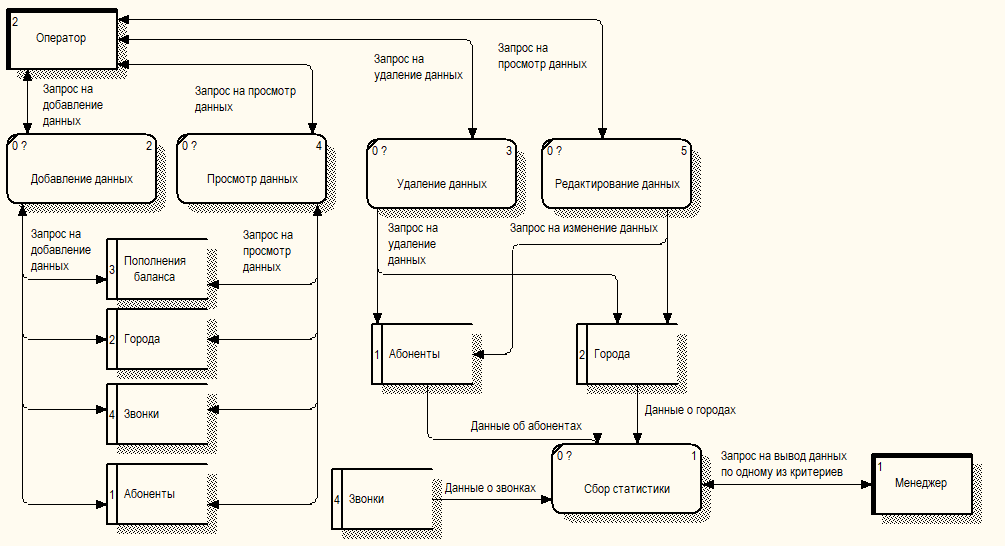


Рисунок Б.2 Функциональная модель DFD: детализация главного процесса, уровень А0

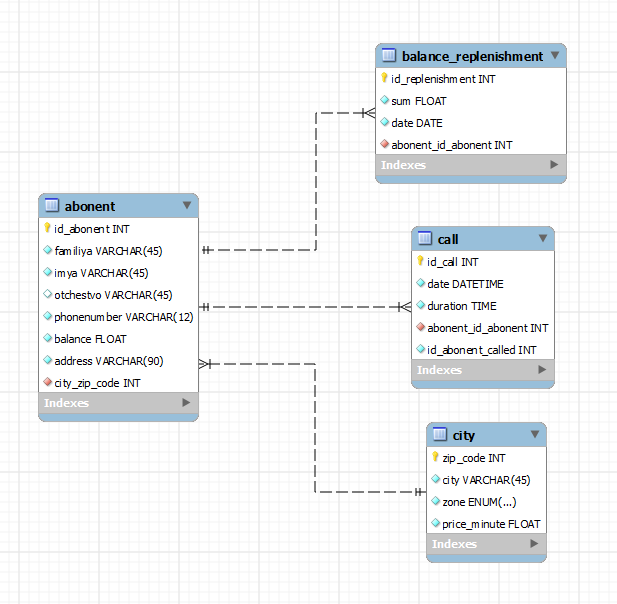


Рисунок Б.3 - Структура базы данных в нотации Мартина

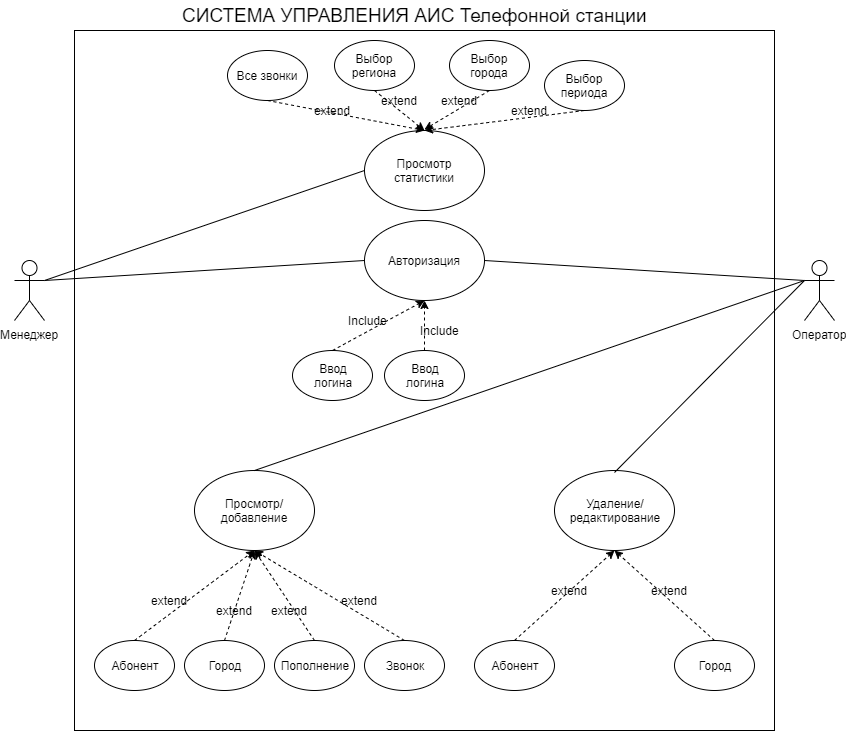


Рисунок Б.4 - Use case диаграмма, Актеры: Менеджер, оператор

Таблица Б.1 – Таблица абонентов (abonent)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
| id\_abonent | int | + | + | - | Автоинкремент | Идентификатор абонента |
| familiya | Varchar(45) | + | - | - | Буквы и “-“ | Фамилия абонента |
| imya | varchar (45) | + | - | - | Буквы и “-“ | Имя абонента |
| otchestvo | varchar (45) | - | - | - | Буквы и “-“ | Отчество абонента |
| city\_zip\_code | int | + | - | + |  | ZIP-код города, где живет абонент |
| balance | float | + | - | - | Значение по-умолчанию - 0 | Баланс абонента |
| phonenumber | Varchar(12) | + | - | - | Маска: +7-XXX-XXX-XX-XX | Номер телефона абонента |
| address | Varchar(90) | + | - | - |  | Адрес абонента |

Таблица Б.2 - Таблица пополнений баланса (balance\_replenishment)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
| id\_replenishment | int | + | + | - | Автоинкремент | Идентификатор пополнения |
| sum | float | + | - | - | Не может быть меньше 0 | Сумма пополнения |
| date | date | + | - | - | Не может быть позже текущего дня | Дата пополнения баланса |
| abonent\_id\_abonent | int | + | - | + |  | Идентификатор абонента |

Таблица Б.3 - Таблица звонков(call)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
| id\_call | int | + | + | - | Автоинкремент | Идентификатор звонка |
| date | DateTime | + | - | - | Не может быть позже текущего дня и времени | Дата и время звонка |
| duration | time | + | - | - | Нет ограничений | Длительность звонка |
| abonent\_id\_abonent | int | + | - | + |  | Идентификатор вызывающего абонента |
| id\_abonent\_called | int | + | - | - | должен существовать в таблице abonent | Идентификатор вызываемого абонента |

Таблица Б.4 - Таблица городов (city)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
| Zip-code | int (15) | + | + | - | 3 символа | ZIP-код города |
| city | varchar (45) | + | - | - | Нет Ограничений | Название города |
| zone | ENUM('City', 'Republic', 'CIS', 'Foreign country') | + | - | - |  | Зона расположения города |
| price\_minute | float | + | - | - | Не может быть меньше нуля | Цена за минуту разговора |

**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет среднего профессионального образования**

**И Н Д И В И Д У А Л Ь Н О Е З А Д А Н И Е**

**на производственную практику (по профилю специальности)**

**(ПП.11.01)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Студент** | Разницын Глеб Александрович | | | | | **Группа №** | Y2436 |
|  | | | | | | | |
| **Руководитель**  **от организации** | | Ильяшенко Александр Сергеевич, начальник отдела НИОКР ООО «Центр реактивного программирования» | | | | | |
| **Руководитель от**  **факультета СПО** | | Говорова Марина Михайловна, преподаватель факультета СПО  Ефимова Татьяна Николаевна, преподаватель факультета СПО | | | | | |
|  | | | | | | | |
| **Тема задания:** | | | Разработка прототипа автоматизированной информационной системы для телефонной станции | | | | |
| **Сроки прохождения практики:** | | | | 09.09.2019 – 28.12.2019 | | | |
| **Место прохождения практики:** | | | | ООО «Центр реактивного программирования» | | | |
| **Должность практиканта:** | | | | – | | | |
| **1. Виды работ и требования к их выполнению:** | | | |  | | | |
| Производственная практика проводится аудиторно распределенно (понедельно в течение семестра) на базе ООО «Центр реактивного программирования». В ходе прохождения практики выполняются следующие виды работ:   1. Вводный инструктаж по технике безопасности и общим целям и задачам практики. 2. Анализ поставленной задачи: формирование по результатам анализа поставленной задачи исходных материалов для формирования технического задания на разработку. Изучение требований к составлению программной документации. 3. Выполнение индивидуального задания: формирование функциональные требований к разработке, выбор программных средств для реализации задачи, технологий тестирования, моделирование разработки, формирование разделов технического задания, программная реализация и тестирование разработки. 4. Формирование отчета по производственной практике, включающего техническое задание и техническое описание разработанного программного продукта. | | | | | | | |
| **2. Виды отчетных материалов и требования к их оформлению:** | | | | |  | | |
| По результатам прохождения практики составляется отчет, в котором представляется техническое задание на разработку и техническое описание разработанного программного продукта. Оформление отчета должно соответствовать Методическим рекомендациям по оформлению отчетов факультета СПО. | | | | | | | |

**3. ПЛАН-ГРАФИК**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа | Наименование этапа | Срок завершения этапа | Виды работ | Форма отчетности |
| 1 | Вводный инструктаж | 09.09.2019 – 14.09.2019 | Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами производственной практики | Журнал по технике безопасности |
| 2 | Анализ поставленной задачи | 16.09.2019 – 30.09.2019 | Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме производственной практики. | Отчет по практике:  индивидуальное задание  техническое задание  Дневник практики |
| 3 | Выполнение индивидуального задания | 01.10.2019 – 13.12.2019 | Определение требований к разработке. Выполнение поиска и мотивированного выбора инструментальных средств для решения задачи. Выбор технологий тестирования.  Моделирование функционального поведения системы и поведения на уровне пользователей с использованием CASE-средств. Создание концептуальной, логической и физической моделей базы данных (с использованием CASE-средств). Формирование технического задания. Разработка серверной части системы в инструментальной программной оболочке. Реализация и заполнение базы данных. Разработка клиентской части системы в инструментальной программной оболочке. Тестирование разработки. Установка и настройка параметров системы. Демонстрация системы. | Отчет по практике: техническое описание  Дневник практики |
| 4 | Подготовка отчетных материалов | 15.12.2019 – 20.12.2019 | Формирование отчета о прохождении практики. | Отчет по практике  Дневник практики |
| 5 | Защита результатов практики | 22.12.2019 – 28.12.2019 | Защита результатов практики в форме устного собеседования и представления результатов с руководителями практики и представителями работодателя. | Отчет по практике  Дневник практики |

**Председатель выпускающей комиссии**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Королёв В. В.

(подпись) (ФИО)

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

|  |  |  |
| --- | --- | --- |
| **Дата выдачи задания:** | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г. | |
|  | |  |
| **Руководитель от организации** | |  |
|  | | (подпись руководителя) |
| **Руководитель от факультета СПО** | |  |
|  | | (подпись руководителя) |
| **Задание принял к** | |  |
| **исполнению** | |  |
|  | | (подпись студента) |