Министерство образования Иркутской области

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.03.22.191.10 ПЗ |  |

ИНФОРМАЦИОННАЯ СИСТЕМА «АВИАПЕРЕЛЁТЫ»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.А. Евдокимов) |

Иркутск 2022

# Содержание

[Введение 3](#_Toc122033024)

[1 Описание предметной области 4](#_Toc122033025)

[2 Анализ программных продуктов, используемых при разработке ИС 5](#_Toc122033026)

[3 Техническое задание 8](#_Toc122033027)

[4 Проектирование ИС 9](#_Toc122033028)

[4.1 Структурная схема ИС 9](#_Toc122033029)

[4.2 Функциональная схема ИС 11](#_Toc122033030)

[4.3 Проектирование базы данных ИС 13](#_Toc122033031)

[4.4 Проектирование интерфейса 18](#_Toc122033032)

[5 Разработка ИС 22](#_Toc122033033)

[5.1 Разработка интерфейса ИС 22](#_Toc122033034)

[5.2 Разработка базы данных ИС 26](#_Toc122033035)

[5.3 Разработка ИС 28](#_Toc122033036)

[6 Документирование программного продукта 33](#_Toc122033037)

[6.1 Руководство пользователя ИС 33](#_Toc122033038)

[Список используемых источников 39](#_Toc122033039)

[Приложение А – Техническое задание 40](#_Toc122033040)

[Приложение Б – Листинг авторизации 44](#_Toc122033050)

[Приложение В – Листинг добавления данных в таблицу самолетов 50](#_Toc122033051)

[Приложение Г – Листинг вывод, обновление и удаление данных в таблице самолетов 52](#_Toc122033052)

# Введение

В настоящее время автоматизация коснулась практически всех областей человеческой деятельности. Естественно, что на преимущества автоматизации первыми откликнулись и стали их использовать в своей деятельности коммерческие предприятия. Одним из бизнес-процессов, который подлежит автоматизации в первую очередь – это различного вида учеты, особенно это касается учета продаж.

Согласно статистики все больше людей покупают авиабилеты через интернет. Сегодня, как минимум каждый 8 пассажир России осуществляет покупку билета именно посредством всемирной паутины.

Хотя число таких людей растет из года в год, все же нам еще далеко до показателей развитых стран. Так, в Европе и США уже давно основная доля продаж (и не только авиабилетов) приходится именно на онлайн.

Актуальность темы – огромное повышение спроса на перелет между странами.

Целью данного курсового проекта является проектирование и разработка информационной системы, осуществляющей продажи авиабилетов

Разработка подобной системы весьма актуальна на данный момент. В современном мире самолеты являются не только самым быстрым видом транспорта, но и самым безопасным, в связи с этим авиаперелеты пользуются весьма высокой популярностью.

Вследствие этого продаваемые на рейсы билеты востребованы и с высокой вероятностью найдут своего покупателя, при условии, что авиакомпания обеспечила клиенту полноценный доступ к нужной ему информации.

# Описание предметной области

Авиакомпания занимается перевозкой пассажиров. Также она устанавливает маршруты полетов. Рейсы двигаются к установленному маршруту согласно расписанию. Продажа билета пользователю осуществляется при отправке запроса на бронирование, при условии, что до сих пор есть свободные места. Приобретя билет, пользователь предоставляет информацию о себе и становится пассажиром. Совершеннолетние пассажиры обязаны иметь занесенные в БД паспортные данные. Несовершеннолетние обязаны иметь занесенные в БД данные из свидетельства о рождении. Администраторы системы могут изменять данные о рейсах и пассажирах. Система создаётся для обслуживания следующих групп пользователей:

* пользователи, приобретающие билеты на рейсы и осуществляющие их поиск;
* администраторы, осуществляющие контроль за пользователями и функциональностью системы.

Абсолютно каждая авиакомпания использует определенную систему дистрибуции. Наиболее используемая это GDS (глобальные дистрибьюторские системы). Доступ к GDS является платным, поэтому в роли дистрибутивной системы для разрабатываемого продукта будет выступать БД, созданная PhpMyAdmin.

В БД должна храниться информация:

* о рейсах;
* о сотрудниках;
* о пассажирах;
* о самолетах.

Разрабатываемая информационная система предназначена для продажи авиабилетов и упрощения доступа к нужной информации. Наличие данной разработки улучшает организационную работу авиаперевозчика за счёт отсутствия бумажной документации, поиск и систематизация которой занимали бы очень большое количество времени.

# Анализ программных продуктов, используемых при разработке ИС

Для разработки информационной системы будут использоваться следующие программные продукты: MySQL, PhpMyAdmin, Visual Studio.

SQL – это язык запросов, который применяют, чтобы работать с базами данных, структурированных особым образом. Главные задачи SQL — составлять запросы так, чтобы находить среди большого объёма информации ту, что нужна для конкретных целей, сортировать её, структурировать и представлять в наиболее простом и понятном виде.

PhpMyAdmin – это веб-приложение для управления базами данных MySQL и MariaDB с использованием графического пользовательского интерфейса (GUI). Он предоставляет пользователям простой способ создавать, редактировать, изменять и удалять таблицы базы данных.

Visual Studio – это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

Преимущества:

* постоянно расширяющийся магазин дополнений, предоставляющий надстройки для расширения возможностей взаимодействия с IDE;
* встроенная технология автозавершения кода IntelliSense;
* возможности для командной работы: управление доступами к проекту, полностью настраиваемые параметры редактора;
* мультиязычность;
* индивидуальная настройка рабочей панели;
* есть поддержка split screen.

Недостатки:

* отсутствие возможности работы на Linux ОС;
* это достаточно тяжеловесная среда. Если у вас нет необходимости разрабатывать консольные приложения, то можно рассмотреть более подходящий редактор кода.

Язык C# – разработанный компанией Майкрософт, один из самых популярных современных языков программирования. Он востребован на рынке разработки в различных странах, C# применяют при работе с программами для ПК, создании сложных веб-сервисов или мобильных приложений.

Преимущества:

* подлинная объектная ориентированность;
* компонентно-ориентированное программирование;
* безопасный (по сравнению с языками C и C++) код;
* унифицированная система типизации;
* поддержка событийно-ориентированного программирования;
* объединение лучших современных языков программирования: Java.

Минусы:

* система требовательна к железу — сказывается то, что IDE базируется на движке Electron;
* для небольших проектов излишне функциональна;
* медленно запускает проекты;
* платные варианты стоят от 45 до 250 долларов в месяц — да, только по подписке.

Кроме того, для выбора инструментов разработки были произведены сравнения, результат которых представлены в таблицах 1, 2, 3.

Таблица 1 – Сравнение IDE для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название IDE | Visual Studio | Eclipse | IntelliJ IDEA |
| Бесплатность | -  (бесплатная студенческая  лицензия) | + | -  (бесплатная студенческая  лицензия) |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Автосохранение | + | + | + |
| Автодополнение | + | **-** | + |
| Интеграция с системами управлениями версиями(git) | + | **-** | - |
| Поиск по коду | + | + | + |

Таблица 2 – Сравнение языков программирования для разработки программного продукта

|  |  |  |
| --- | --- | --- |
| Название языка программирования | С# | Java |
| Простота обучаемости | + | **-** |
| Читабельность | + | **-** |
| Динамическая типизация | + | + |
| Интеграция баз данных | + | + |
| Объектно-ориентированные возможности | + | + |

Таким образом, выбранные средства разработки программного обеспечения являются лучшим вариантом в своей среде разработки, как по функционалу разработки, так и по программным решениям.

# Техническое задание

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 34.602-89.

Согласно ГОСТ 34.602-89 техническое задание должно включать следующие разделы:

1. общие сведения;
2. назначение и цели создания системы;
3. требования к системе в целом;
   1. требования к структуре и функционированию системы;
   2. требования к надежности;
   3. требования к безопасности;
   4. требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
4. требования к документированию;
5. состав и содержание работ по созданию системы.

Техническое задание на разработку информационной системы представлено в Приложение А.

# Проектирование ИС

## Структурная схема ИС

Проектирование информационной системы происходит при помощи CASE средств, которые позволяют за короткий срок создавать схемы и реляционные модели программ.

На рисунке 1 изображена Use Case View, которая показывает структурную схему ИС «Авиаперелеты» для клиентов и сотрудников.

Она отображает действия, выполняемые сотрудником и клиентом. «Сотрудник» и «Клиент» являются – актерами. Авторизация, Просмотр информации о ближайших рейсах, Просмотр информации о клиентах, Принятие брони и Регистрация клиента на рейс – действия сотрудника. Регистрация, Авторизация, Заказ, отмена и просмотр информация о билете, – действия «Клиента».

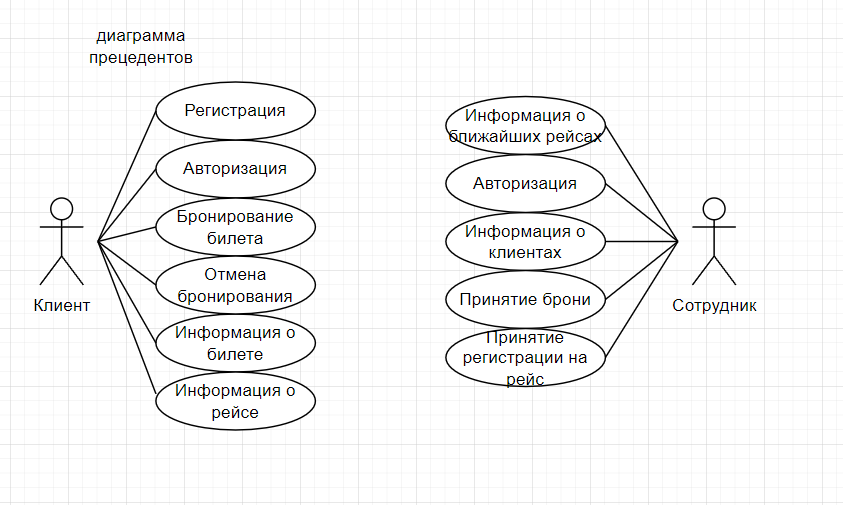


Рисунок 1 – Структурная схема ИС Авиаперелеты

На рисунке 2 представлена диаграмма деятельности. Она содержит 3 роли. Процесс начинается с клиента, который заказывает билет. Далее пользователь смотрит её и либо отменяет брони, либо регистрируется на рейс. Приложение оповещает сотрудника о новом заказе и передает ему данные клиента. Сотрудник проверяет данные клиента и одобряет, либо отколняет бронирование. После этого клиент получает всю информацию о предстоящем рейсе.

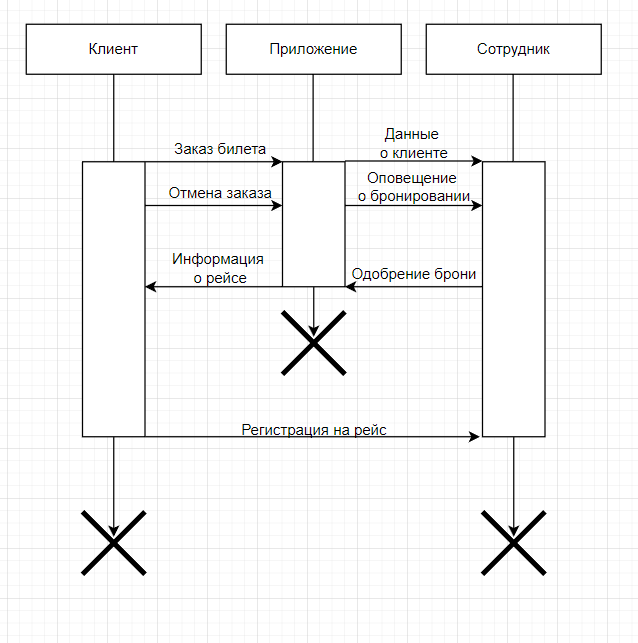


Рисунок 2 – Диаграмма деятельности

Таким образом, представленная диаграмма демонстрирует взаимодействие клиентов и сотрудников по средствам информационной системы.

На рисунке 3 представлена диаграмма компонентов. В общей сложности 10 элементов, которые служат для формирования билета, оплаты, выбора страны перелета, регистрации клиентов на рейс и изменения рейсов.

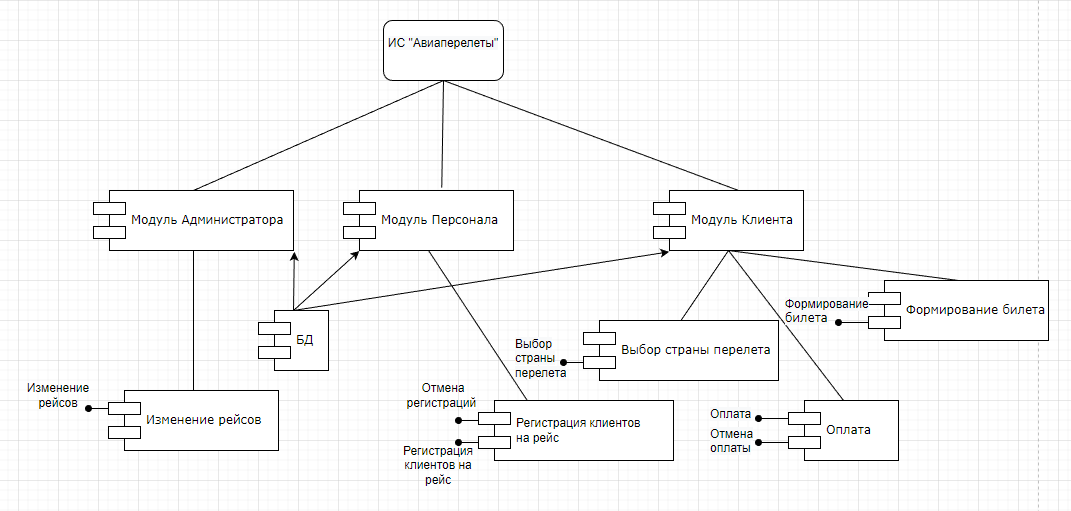


Рисунок 3 – Диаграмма компонентов

На рисунке 4 представлена диаграмма развёртывания. Она показывает, что, чтобы пользоваться программным продуктом, необходим web-сервер, на котором размещаются ИС и БД, клиентское приложение для заказа билетов и приложение для персонала находящееся на стойках регистрации, которое регистрирует клиента на рейс.

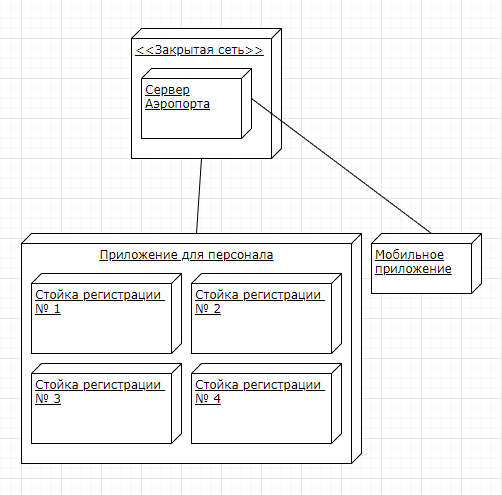


Рисунок 4 – Диаграмма развертывания

## Функциональная схема ИС

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На контекстной диаграмме на рисунке 5 показаны входные данные, управление механизм, выходные данные и функция.

В центре диаграммы показана функция «Информационная система Авиаперелеты». Входные данные: «Бронирование билета». Управление: «Алгоритм работы». Механизм «Приложение». Выходные данные «Перелет».

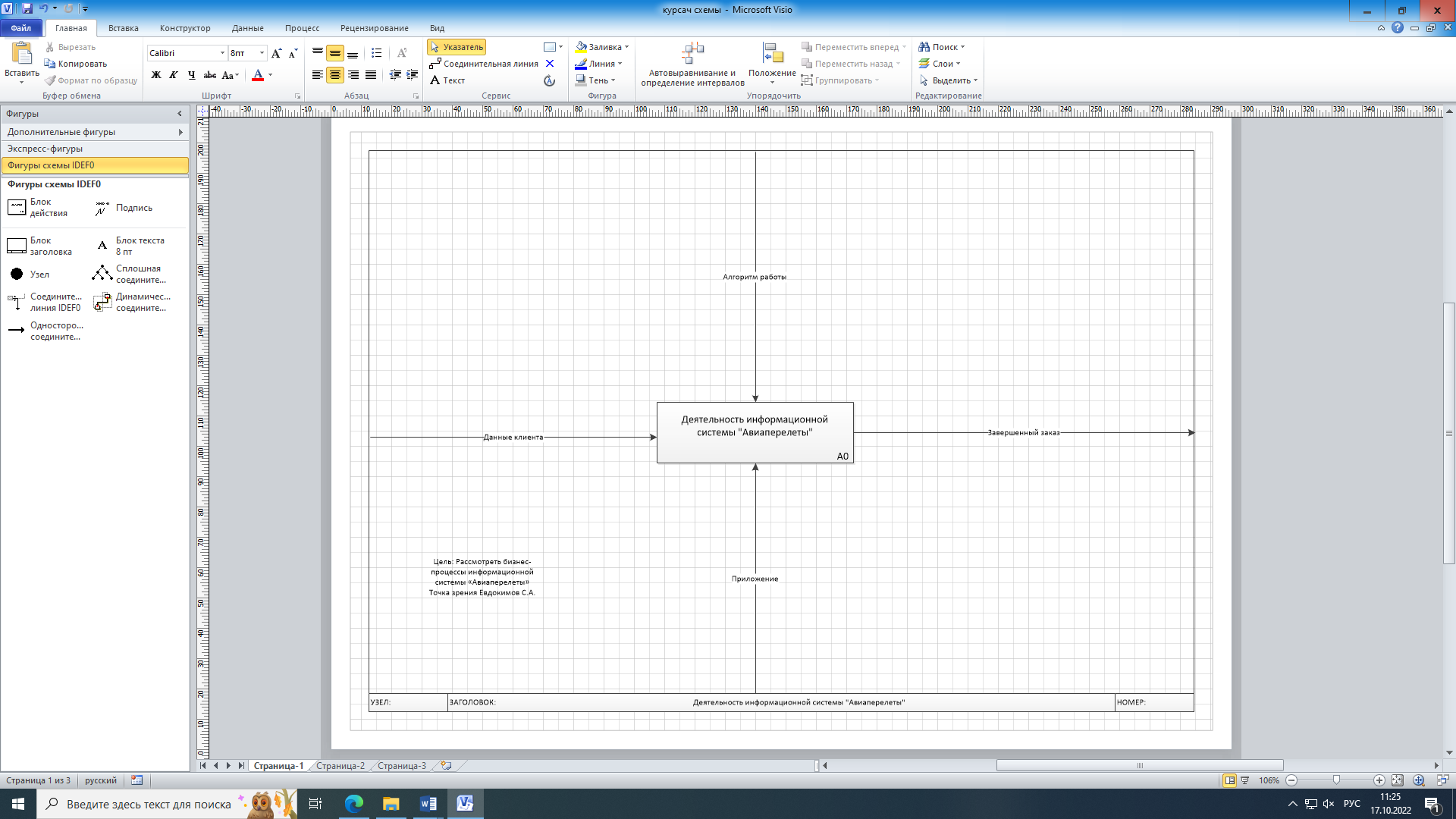


Рисунок 5 – Контекстная диаграмма

Диаграмма декомпозиции – это разбиение функции, т.е. «Деятельность информационной системы «Авиаперелеты»».

На рисунке 6 показана диаграмма декомпозиций, которая расписывает функцию. На этой диаграмме показаны такие функции как: «Регистрация», «Бронирование билета», «Принятие брони», «Посадка на рейс. А также на этой диаграмме присутствуют входные данные, выходные данные, управления и механизмы.

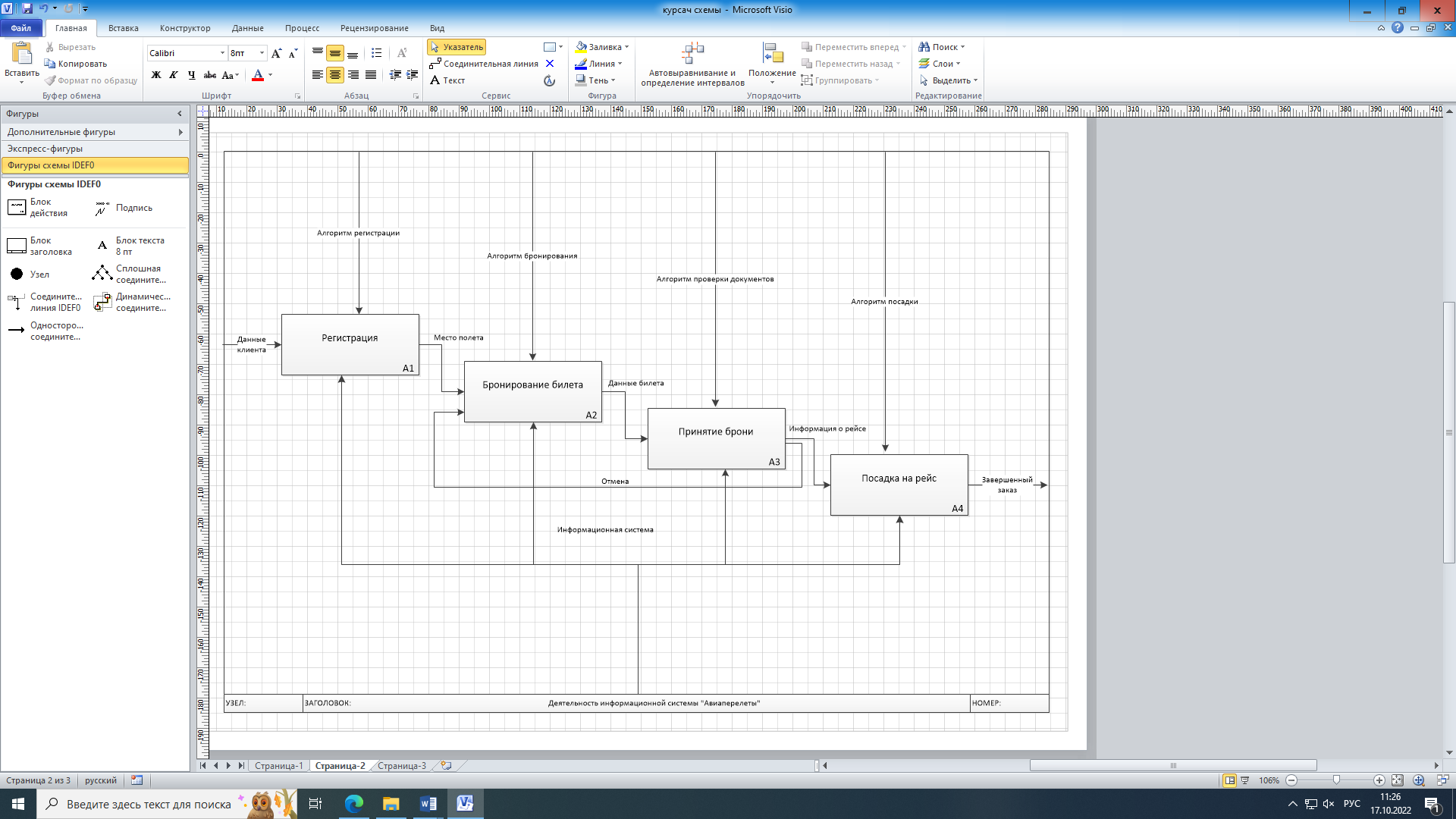


Рисунок 6 – Диаграмма декомпозиции

Таким образом, представленные диаграммы демонстрируют взаимодействие между основными функциями информационной системы «Авиаперелеты» и механизмами, влияющими на выполнение той или иной функции.

## Проектирование базы данных ИС

Проектирование информационной системы происходит при помощи CASE средств, которые позволяют за короткий срок создавать схемы и реляционные модели программ.

Концептуальное проектирование – построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных.

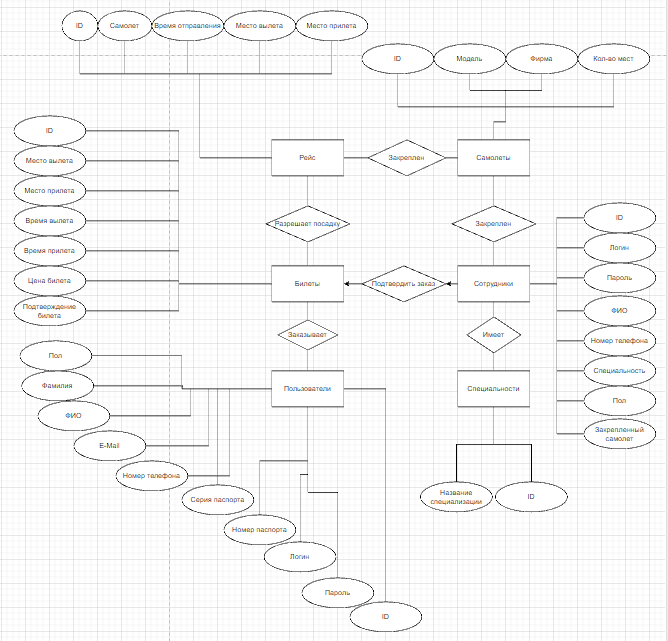


Рисунок 7 – Инфологическая модель базы данных

На инфологической модели базы данных выше схематично отображены сущности системы, их атрибуты и связи между ними. Так, в прямоугольник отображены сущности, такие как: Пользователь, Билеты, Самолеты и Специальности. В овалах отображены атрибуты сущностей, например: ID, логин, пароль, ФИО и т.д. Робами изображены связи между сущностями, например пользователь заказывает билет.

Далее происходит преобразование концептуальной модели в логическую модель, по формальным правилам.

На даталогической модели базы данных (рисунок) отображены сущности приложения, а также первичные и внешние ключи, связывающие сущности между собой. Так, сущность «user» содержит в качестве внешнего ключа поле «id\_user», которое содержит первичный ключ пользователя.

Для построения даталогической модели использовался инструмент «MySQL Workbench».

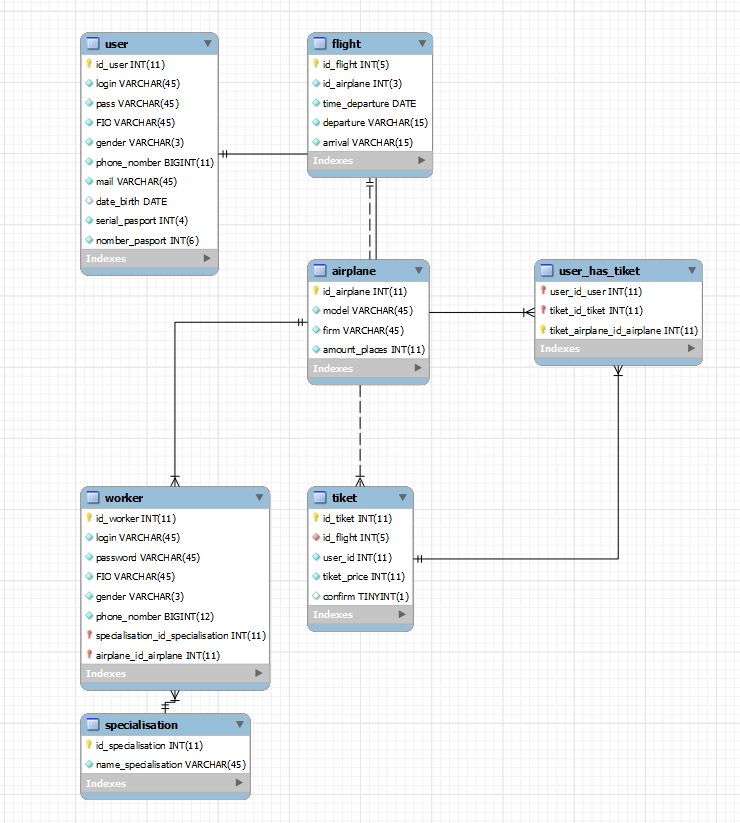


Рисунок 8 – Даталогическая модель базы данных

Таблица 3 – «airplane»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Id\_airplane | Int | Уникальный ключ |
| model | VarChar | Модель самолета |
| firm | VarChar | Фирма производителя |
| amount\_places | Int | Количество мест |

Таблица 4 – «flight»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id\_flight | Int | Уникальный ключ |
| id\_airplane | int | Ключ закрепленного самолета |
| time\_departure | date | Время вылета |
| departure | VarChar | Место вылета |
| arrival | VarChar | Место прилета |

Таблица 5 – «specialisation»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id\_specialisation | Int | Уникальный ключ |
| name\_specialisation | VarChar | Название специальности |

Таблица 6 – «tiket»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id\_tiket | Int | Уникальный ключ |
| id\_flight | int | Ключ рейса |
| user\_id | int | Ключ пользователя |
| tiket\_price | int | Цена билета |
| confirm | tinyint | Подтверждение билета |

Таблица 7 – «user»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id\_user | Int | Уникальный ключ |
| login | VarСhar | Логин |
| pass | VarСhar | Пароль |
| FIO | VarСhar | ФИО пользователя |
| gender | VarСhar | Пол |
| phone\_nomber | BigInt | Номер телефона |
| mail | VarСhar | Электронная почта |
| date\_birth | date | Дата рождения |
| serial\_pasport | Int | Серия паспорта |
| nomber\_pasport | Int | Номер паспорта |

Таблица 8 – «worker»

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id\_worker | Int | Уникальный ключ |
| login | VarChar | Логин |
| password | VarChar | Пароль |
| FIO | VarChar | ФИО рабочего |
| gender | VarChar | Пол |
| phone\_nomber | BigInt | Номер телефона |
| specialisation\_id\_specialisation | Int | Ключ специализации рабочего |
| airplane\_id\_airplane | Int | Ключ закрепленного за сотрудником самолета |

После завершения разработки прототипа базы данных информационной системы «Авиаперелеты» получиться готовая схема, по которой будет разрабатываться база данных информационной системы.

На рисунке 9 представлена диаграмма классов. Она содержит 5 классов, среди них пользователь, рабочий, специальность, билет и самолет.

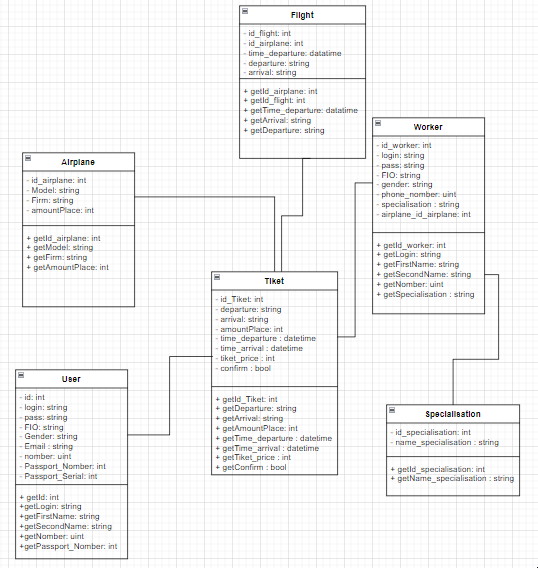


Рисунок 9 – Диаграмма классов

На рисунке 10 представлена диаграмма потоков данных. В центре всего пользователь, который может забронировать билет на рейс, после чего рабочий проверяет данные и регистрирует его на рейс. Для соответствующих действий в ИС предусмотрены соответствующие БД.

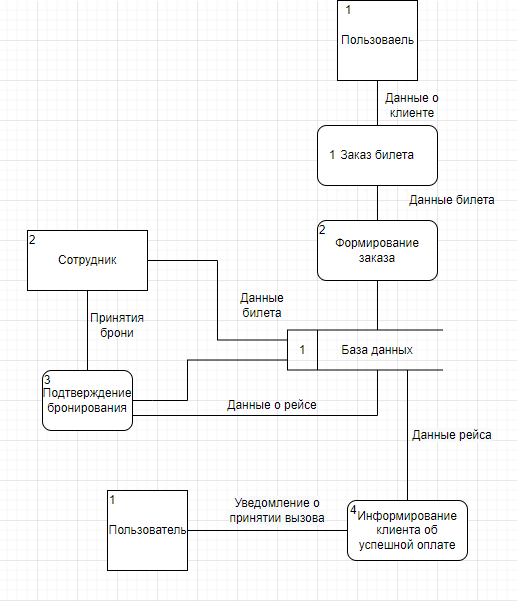


Рисунок 10 – Диаграмма потоков данных

## Проектирование интерфейса

Для разработки пользовательского интерфейса был выбран инструмент proto.io – браузерный инструмент для создания прототипов программ.

В результате проектирование интерфейса будущей информационной системы были спроектированы прототипы трёх форм: Форма авторизации (рисунок 11), Форма регистрации (рисунок 12), Форма Заказа билета, состоящая из: выбора места вылета, выбора места прилета и времени вылета (рисунки 13, 14, и 15).

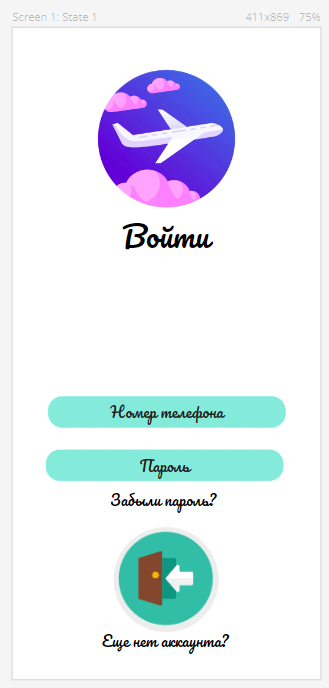


Рисунок 11 – Форма Авторизация



Рисунок 12 – Форма Регистрация

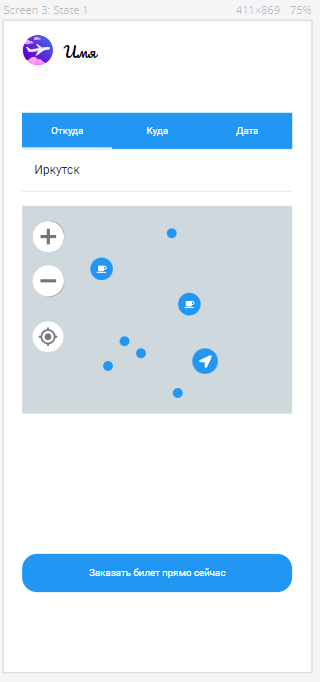


Рисунок 13 – Форма Заказа билета. Выбора место отправления

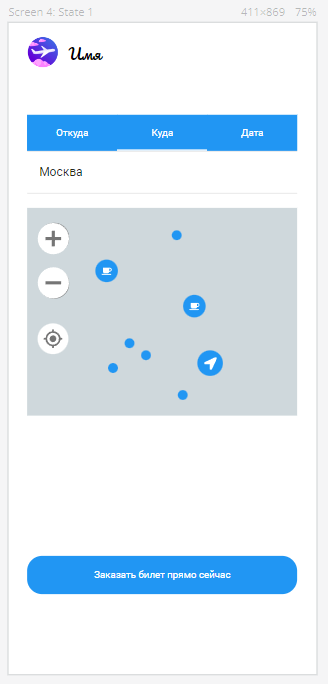


Рисунок 14 – Форма Заказа билета. Выбора места прилета

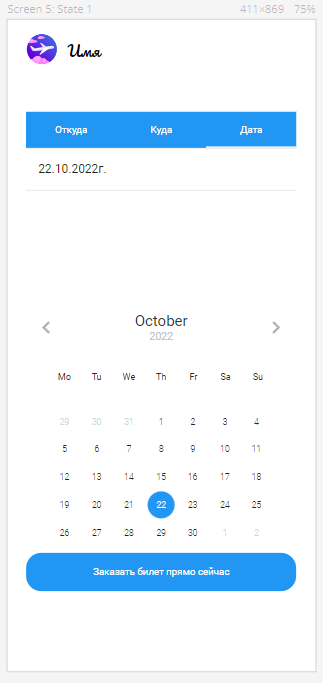


Рисунок 15 – Форма Заказа билета. Выбор времени отправления

# Разработка ИС

## Разработка интерфейса ИС

Разработка удобного пользовательского интерфейса - это один из важнейших этапов в процессе создания информационной системы.

В разрабатываемой информационной системе были поставлены и выполнены следующие задачи:

* авторизация и регистрация;
* просмотр данных;
* манипулирование данных.

Вся структура работы представляет собой набор кода формата \*.cs с возможность переходов между формами по средствам вызова и скрытия этих самых форм, начиная с экрана авторизации и переходя последовательно к другим экранам.

Для разработки дизайна использовалась библиотека Guna – она добавляет модифицированные компоненты стандартного набора Windows Form, что позволяет создать более приятный визуальный вид программы.

Компоненты используемые при разработки интерфейса:

Guna2TextBox – представляет собой графический элемент пользовательского интерфейса, который позволяет использовать простой ввод текста.

Guna2ImageButton – элемент кнопки с изображением вместо текста, клик по которой вызывает срабатывание события.

Guna2Button – обычная кнопка, клик по которой вызывает срабатывание события.

Guna2DataGridView – это элемент управления, позволяющий отображать данные в табличном формате, в настраиваемой сетке.

Guna2Panel – позволят объединять элементы в группы для облегченного управления ими.

Label – это простая текстовая метка, предназначенная для вывода текста.

GMapControl – элемент библиотеки GMap, позволяющий использовать карту мира из различных источников.

DataTimePicker – элемент управления, который позволяет выбрать дату и время и отобразить их в указанном формате.

Пользовательский элемент управления – позволяет инкапсулировать пользовательский интерфейс в повторно используемые пакеты, которые можно легко подключать к различным проектам через Panel.

Все поля ввода и формы обладают проверкой заполнения данных, таким образом в случае не заполнения данных пользователь получит сообщение об ошибки.

На рисунке 16 изображена страница авторизации, с помощью которой пользователь может войти в доступную ему систему заказа или принятия билетов, если это работник аэропорта, либо восстановить пароль написав номер телефона, который был указан при регистрации, после чего ему на почту придет код подтверждения смены пароля. В системе предусмотрена защита данных, таких как пароли, которая шифруется методом хеширования «MD5».



Рисунок 16 – Форма Авторизации

На форме регистрации, изображённой на рисунке 17, пользователь может создать новый аккаунт заполнив небольшой список информации о себе.



Рисунок 17 – Форма регистрации

На рисунке 18 изображена форма личного кабинета в которую пользователь попадает после авторизации, где он может: заказать билет, увидеть информацию об аккаунте и его ранее заказанные билеты.

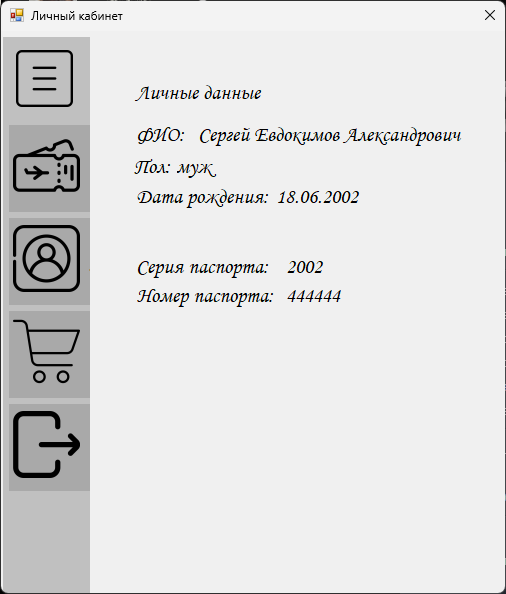


Рисунок 18 – Форма личного кабинета

На рисунке 19 изображена формы панели администратора в которой содержится вся информация об клиентах, рейсах, самолетах аэропорта, должностях и рабочих. Администратор так же может добавлять, изменять и удалять информацию при необходимости.

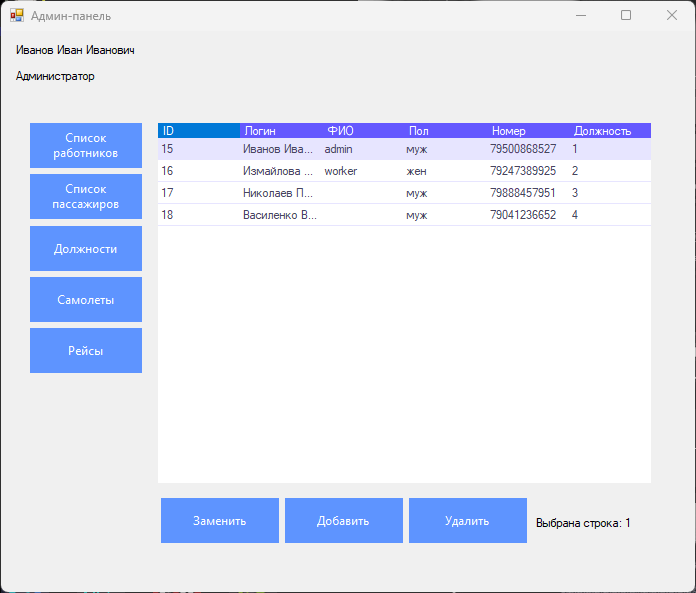


Рисунок 19 – Форма панели администратора

На рисунке 20 изображена форма панели рабочего, с помощью которой работники аэропорта принимают или отклоняют билеты на стойках регистрации, так же сотрудник может сохранять копию необходимых билетов для дальнейшей распечатки.

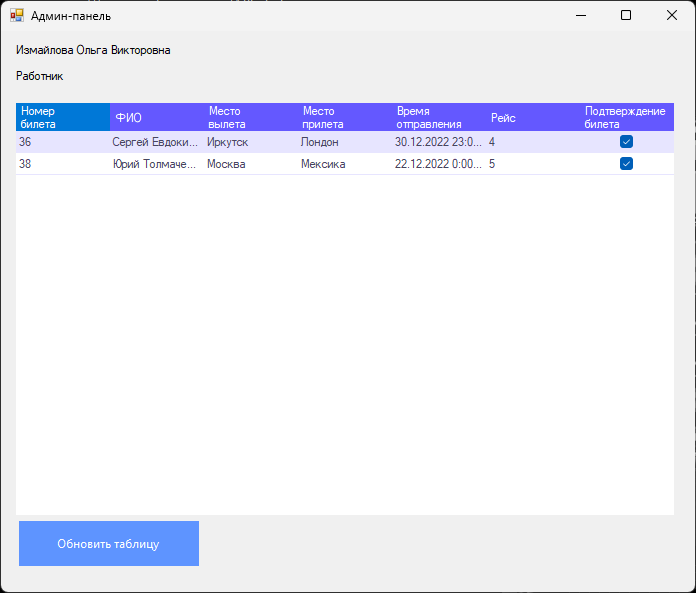


Рисунок 20 – Форма панели рабочего

## Разработка базы данных ИС

База данных – эта часть информационной системы, без которой она не может существовать. Поэтому ее проектирование и разработка влияют на качество самой информационной системы.

Разработка базы данных информационной системы реализовывалась в СУБД MySQL.

База данных информационной системы «Авиаперелеты» состоит из 6 таблиц.

Структуры таблиц соответствуют схеме базы данных из пункта 4.3.

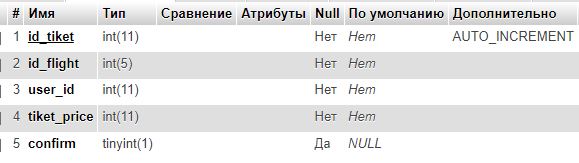


Рисунок 21 – Структура таблицы tiket

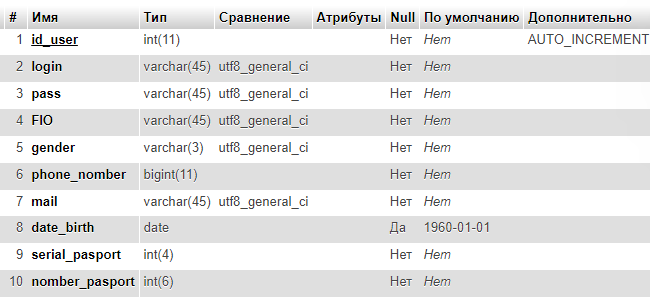


Рисунок 22 – Структура таблицы user



Рисунок 23 – Структура таблицы worker

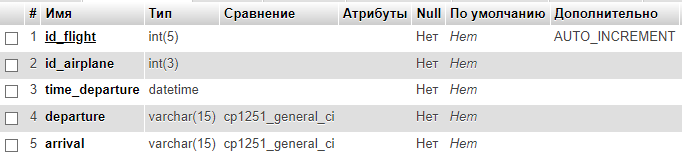


Рисунок 24 – Структура таблицы flight

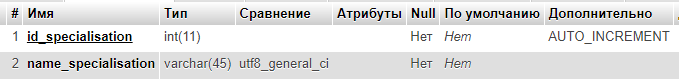


Рисунок 25 – Структура таблицы specialisation

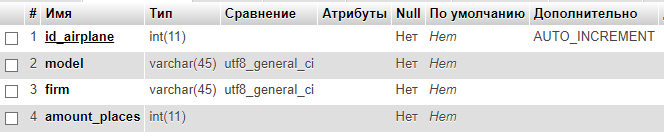


Рисунок 26 – Структура таблицы airplane

После создания всех таблиц и связей в базе данных, она готова к работе.

## Разработка ИС

Подключение к базе данных MySql осуществляется с помощью хостинга ru-hoster.com, данные для подключения хранятся в конфигурационном файле Connect.cs, показанном на рисунке 27.

Переменная conn содержит набор конфигураций подключений к базам данных в виде словаря.

Конфигурация каждого подключения состоит из ряда параметров. Параметр server указывает на хостинг в котором хранится БД, uid – логин для подключения, port ставится по умолчанию, pwd – пароль для подключения, database – название базы данных, convert zero datetime – позволяет использовать DataTime с нулевыми значениями, в результате чего он вернет наименьшее значение DataTime.MinValue.



Рисунок 27 – Подключение к БД

Для авторизации в программу используется функция «User()», показанная на рисунке 28, предназначена для авторизации пользователей и работает с любым устройством, а так же функция «AdminP()», показанная на рисунке 29, предназначена для авторизации сотрудников с защитой от входа в программу со сторонних машин.

Полный код авторизации представлен в Приложении Б.

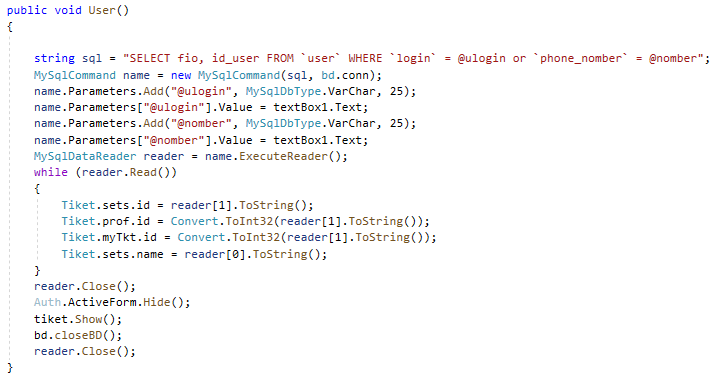


Рисунок 28 – Функция «User()»

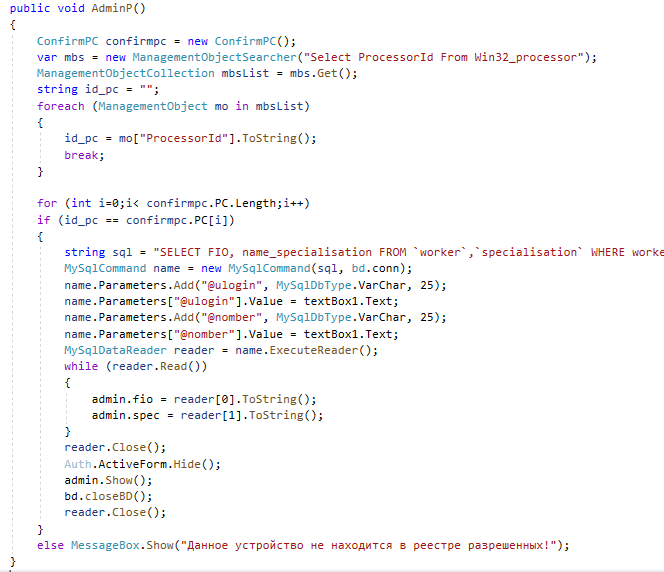


Рисунок 29 – Функция «AdminP()»

При авторизации или регистрации пароль всегда шифруется методом «MD5», показанным на рисунке 30.

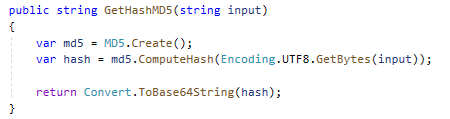


Рисунок 30 – Функция «GetHashMD5()»

Для вывода таблиц из БД используется функция «LoadData()», изображенная на рисунке 31, для вывода необходимой таблицы меняется только SQL запрос и расположение коллекции DataGridView

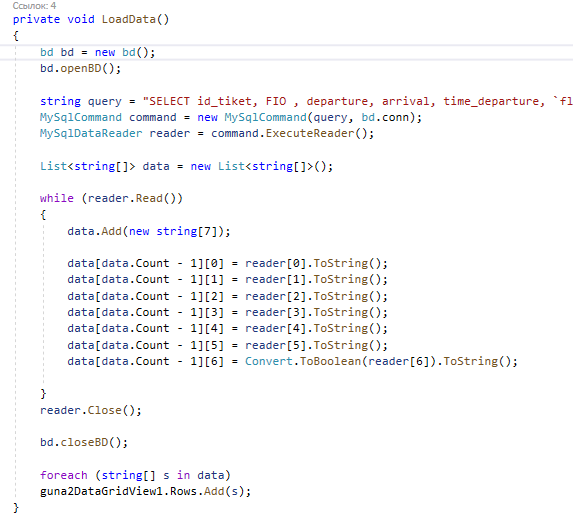


Рисунок 31 – Функция «LoadData()»

Добавление сотрудников производится через отдельную форму билета, содержащую поля для ввода информации (рисунок 32). Полный код манипуляций с таблицами представлен в Приложении В.

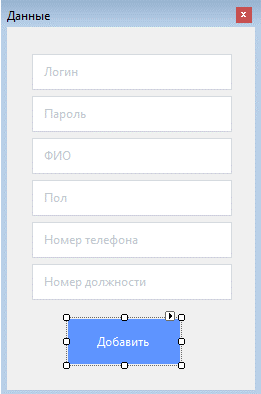


Рисунок 32 – Форма добавления новых сотрудников



Рисунок 33 – Код добавление новых сотрудников

Изменение информации о сотрудниках производится путем изменения полей таблицы DataGridView(рисунок 34), после чего по нажатию на кнопку «Замена» производится занесение выбранной строки в БД (рисунок 30).

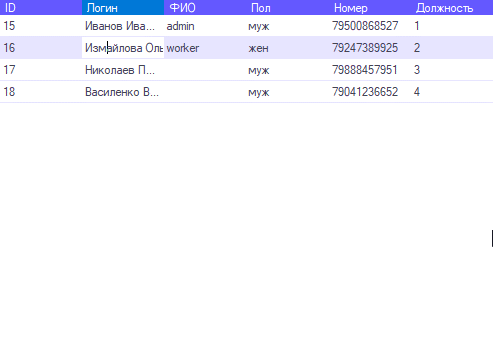


Рисунок 34 – Изменение таблицы информации

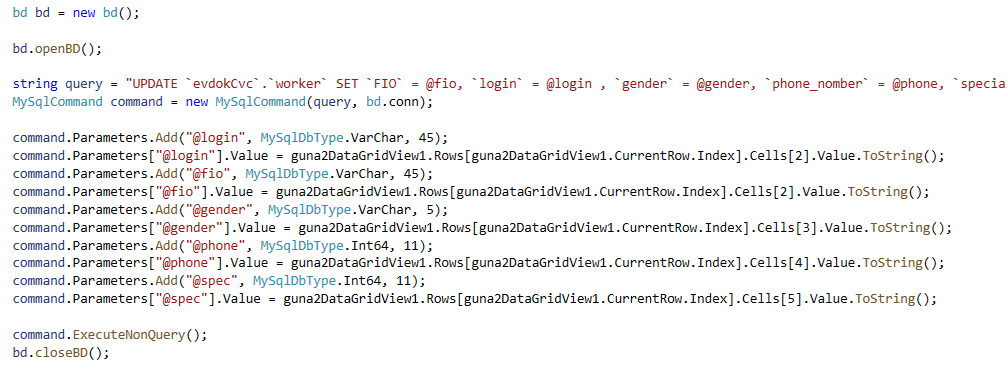


Рисунок 35 – Код изменения информации о сотрудниках

Удаление сотрудников происходит при нажатии на кнопку «Удалить», после чего администратору покажется диалоговое окно с подтверждением (рисунок 63).

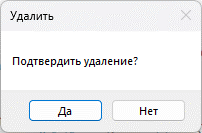


Рисунок 36 – Подтверждение удаления

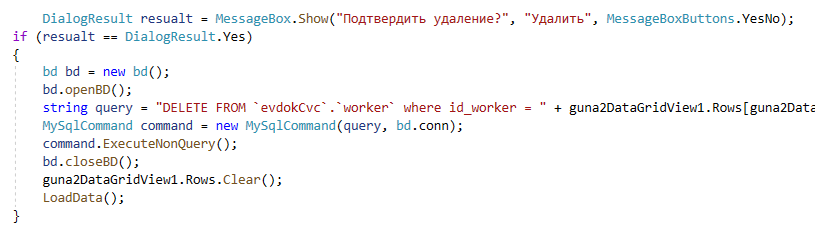


Рисунок 37 – Код удаление сотрудников

Функция «getCheckBox()» предназначена для подтверждения действительности билета (рисунок 38).



Рисунок 38 – Функция «getCheckBox()»

Функция «deleteTiket()», показанная на рисунке 39, предназначена для удаления билета.

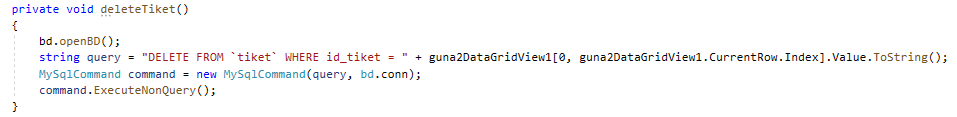


Рисунок 39 – Функция «deleteTiket()»

Функция «Print()», показанная на рисунке 40, предназначена для сохранения билета в формате .JPEG для дальнейшей распечатки.

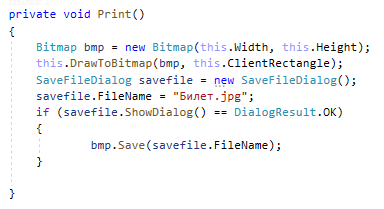


Рисунок 40 – Функция «Print()»

# Документирование программного продукта

## Руководство пользователя ИС

Для того чтобы пользователь смог использовать информационную систему «Авиаперелеты», сначала необходимо скачать файл распаковщик.

Следующим шагом является распаковка приложения, необходимо выбрать путь установки (рисунок 41).

Далее необходимо перейти распакованную папку и запустить приложение AirplaneTiket.exe

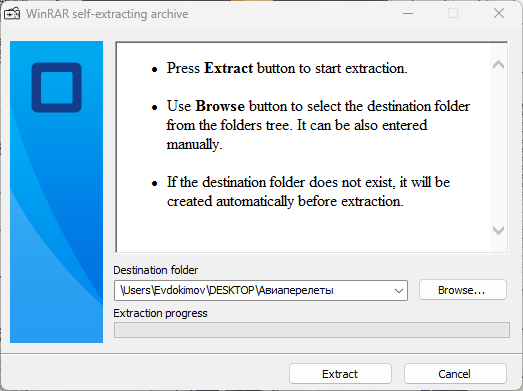


Рисунок 41 – Распаковка программы

При запуске приложения ссылке открывается главная страница, на которой пользователь должен авторизоваться либо зарегистрироваться.

После авторизации пользователь может перейти на страницу «Заказать» и просмотреть список возможных рейсов (рисунок 42).

После ознакомления со списком рейсов, пользователь должен заполнить данные своего паспорта во вкладке «Профиль», если не сделал этого ранее, это необходимо для идентификации пользователя на стойке регистрации (рисунок 43).

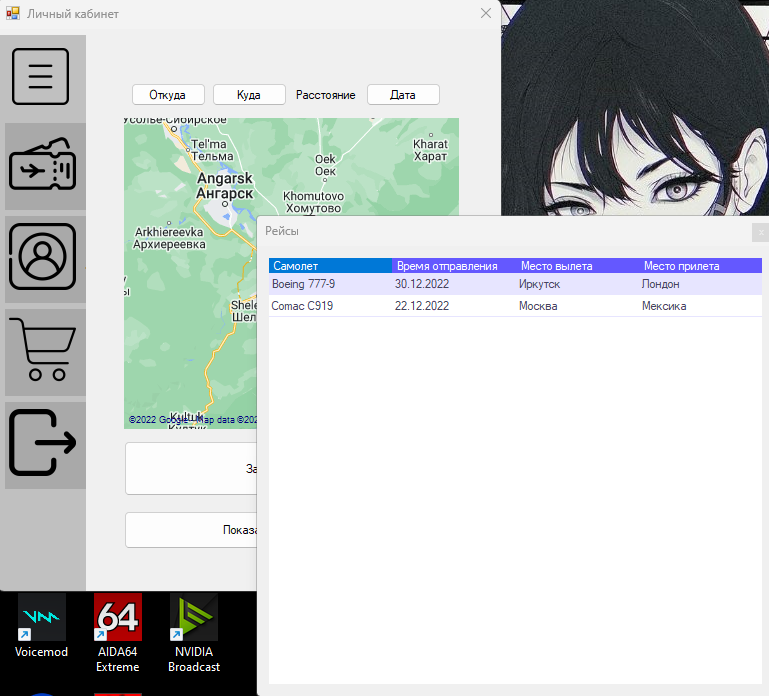


Рисунок 42 – Заказ билетов и текущие рейсы

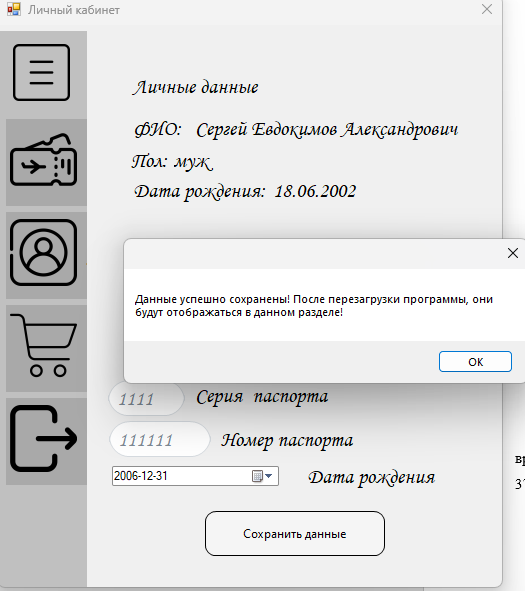


Рисунок 43 – Добавление данных паспорта

После выбора места перелета полете система подберет ближайшее время отправления и проинформирует, что заказ прошел успешно (рисунок 44). и перебросит пользователя на вкладку с его билетами (рисунок 45).

Если пользователь решил отказаться от полета, он может отменить заказ на странице «Мои билеты».

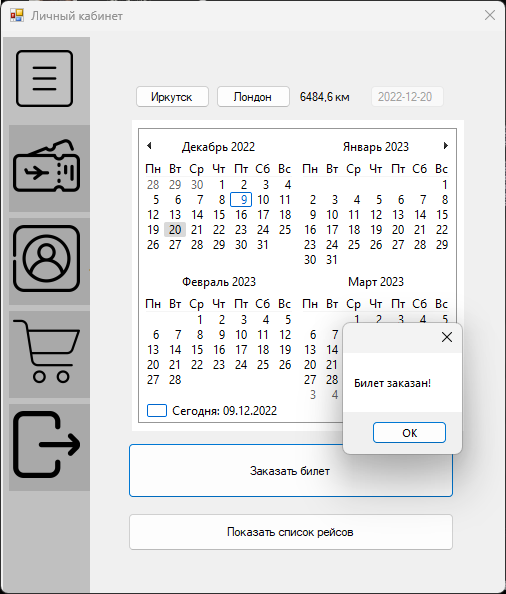


Рисунок 44 – Окно подтверждения заказа

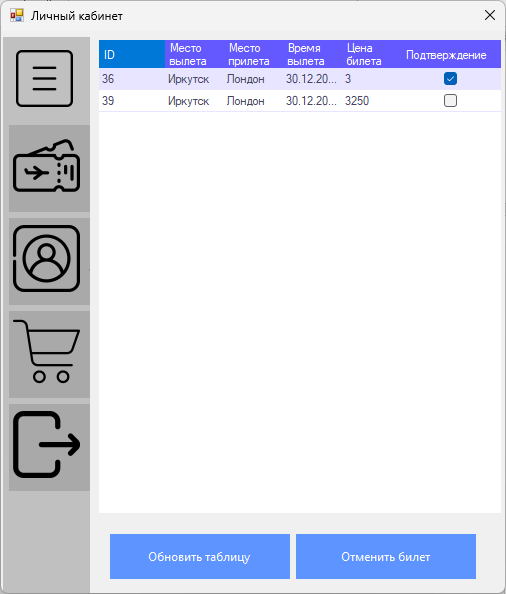


Рисунок 45 – Список билетов пользователя

Сотрудники могут авторизоваться только с заранее занесенных в систему в компьютеров, если они попытаться войти со стороннего устройства, последует уведомление о невозможности входа (рисунок 46).

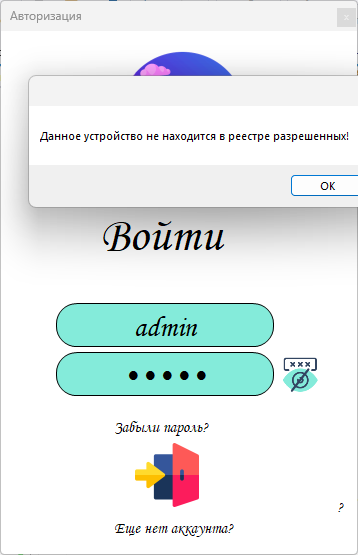


Рисунок 46 – Переход в панель управления

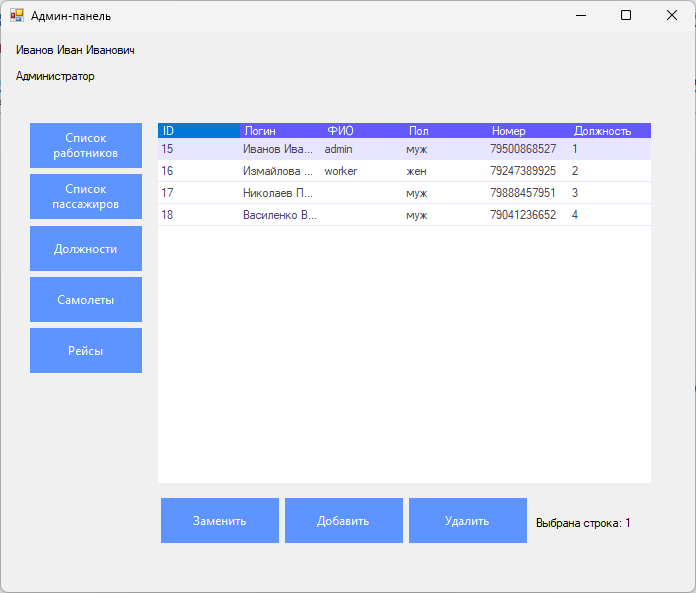
После успешного входа в панель управления, администратор может взаимодействовать с работниками, пассажирами, рейсами, самолетами и должностями. На рисунке 47 представлен общий вид панели управления.

Рисунок 47 – Панель управления**Заключение**

Информационная система «Авиаперелеты» имеет простой и понятный интерфейс, благодаря которому пользователь может с легкостью заказать авиабилет.

Поставленная цель создания курсового проекта в виде разработки информационной системы «Авиаперелеты» и реализации заказа билетов на наиболее подходящие рейсы. Также, была реализована панель управления, с помощью которой администратор может добавлять, редактировать и удалять составляющие ИС, а именно: пользователи, рейсы, самолеты, сотрудники, специализация,.

Для достижения цели были решены следующие задачи:

* изучение предметной области;
* анализ и обоснование выбора программных средств;
* анализ и обоснование выбора языка программирования;
* анализ и обоснование выбора среды разработки;
* проектирование ИС;
* проектирование базы данных;
* проектирование интерфейса;
* разработка интерфейса;
* разработка базы данных;
* разработка ИС;
* разработка руководства пользователя;
* написание ТЗ.

В дальнейшем ИС «Авиаперелеты» может развиваться путём расширения функционала, увеличения количества функционала и персонала, тем самым набирая большее количество клиентов.

Все поставленные цели и задачи курсового проекта были успешно выполнены.

Список используемых источников

1. Хабр – Работа с картами. GMAP C#. – URL: https://habr.com/ru/post/585734/ (дата обращения: 04.10.2022). – Текст: электронный.
2. GunaUI – Build an elegant UI for WinForms. – URL: https://gunaui.com/products/ui-winforms/ (дата обращения: 29.09.2022). – Текст: электронный.
3. Microsoft – CultureInfo Class. – URL: https://learn.microsoft.com/en-us/dotnet/api/System.Globalization.CultureInfo?view=net-6.0 (дата обращения: 27.11.2022). – Текст: электронный.
4. Microsoft – UserControl Class. – URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.controls.usercontrol?view=netframework-4.8 (дата обращения: 03.10.2022). – Текст: электронный.
5. StackOverflow – How to fast get Hardware-ID in C#. – URL: stackoverflow.com/questions/2333149/how-to-fast-get-hardware-id-in-с (дата обращения: 09.12.2022). – Текст: электронный.
6. С#Corner – Hashing Passwords In .NET Core With Tips. – URL: https://www.c-sharpcorner.com/article/hashing-passwords-in-net-core-with-tips/ (дата обращения: 05.10.2022). – Текст: электронный.
7. VsCode – MySQL и C# — работаем с базой данных. – URL: https://vscode.ru/prog-lessons/mysql-c-sharp.html (дата обращения: 19.10.2022). – Текст: электронный.
8. Хабр – Связи между таблицами базы данных MySQL. – URL: https://habr.com/ru/post/488054/ (дата обращения: 25.10.2022). – Текст: электронный.
9. TradeBenefit – Примеры SQL запросов к базе данных MySQL. – URL: http://tradebenefit.ru/primery-mysql-zaprosov (дата обращения: 21.10.2022). – Текст: электронный.
10. CyberForum – Скриншот формы. – URL: https://www.cyberforum.ru/windows-forms/thread2025297.html (дата обращения: 21.10.2022). – Текст: электронный.

# Приложение А – Техническое задание

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
|  |  |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА

«Авиаперелеты»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.А. Евдокимов) |

Иркутск 2022

# 1 Общие сведения

Наименование работы: информационная система «Авиаперелеты».

Исполнитель: студент иркутского авиационного техникума, группы ПКС-19-1, Евдокимов С.А.

Разработка информационной системы проходит в рамках курсового проекта по МДК.03.01 «Технология разработки программного обеспечения», на основании приказа №37-у от 12 сентября 2022 года.

Сроки разработки информационной системы с 21.09.2022 по 06.12.2022 года.

# 2 Назначение и цели создания системы

Назначение информационной системы «Авиаперелеты» заключается в возможности бронирования мест в самолетах на перелет, получении информации о пассажирах и сотрудниках. Для клиента необходимы такие функции как регистрация и авторизация в системе, бронирование места, просмотр своих рейсов, отмена заказа. Для работников –авторизация в системе, просмотр текущих рейсов, отмена брони, получение информации о клиенте, просмотр информации о рейсе.

3 **Требования к системе в целом**

## 3.1 Требования к структуре и функционированию системы

Функции информационной системы:

1. раздел «Бронирование»;
   1. бронирование билета;
   2. отображение информации о заказанном рейсе;
   3. отмена брони;
2. раздел «Рейсы»;
   1. отображение информации о рейсах которые отлетают в ближайшее время;
3. Раздел «Клиенты»;
   1. отображение информации о клиентах.

# 3.2 Требования к надежности

Для обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях таблицы.

**3.3 Требования к безопасности**

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа.

## 3.4 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Минимальные системные требования для рабочей станции:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 250мб;
4. Операционная система: Windows.

Минимальные системные требования для сервера:

1. Процессор с тактовой частотой 1200 MHz
2. Оперативная память 256 Мб или больше
3. Свободное место на жёстком диске от 540 Мб
4. Архитектура с разрядностью 32 бит или 64 бит (x86 или x64)
5. Операционная система Windows XP, Windows Vista, Windows 7, Windows 8

# 4 Требования к документированию

Основным документам, регламентирующими использование информационной системы является руководство пользователя.

Основным документам, регламентирующими разработку информационной системы является техническое задание.

# 5 Состав и содержание работ по созданию системы

В таблице 1 представлены плановые сроки начала и окончания работы по созданию информационной системы

Таблица 1 – Плановые сроки по созданию информационной системы

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Содержание этапов разработки | Сроки выполнения | |
| Начало | Окончание |
| 1 | Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 21.09.22 | 25.09.22 |
| 2 | Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 26.09.22 | 24.10.22 |
| 3 | Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 25.10.22 | 27.11.22 |
| 4 | Разработка (программирование) и отладка программного продукта | 28.11.22 | 05.12.22 |
| 5 | Составление программной документации (оформление руководства пользователя) | 06.12.22 | 16.12.22 |

**Приложение Б – Листинг авторизации**

using System;

using System.Text;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using System.Security.Cryptography;

using System.Diagnostics;

using System.Reflection;

using System.Management;

namespace AirplaneTiket

{

public partial class Authz : UserControl

{

bool hide = true;

Tiket tiket = new Tiket();

Admin admin = new Admin();

bool enter;

bd bd = new bd();

ConfirmPC confirmpc = new ConfirmPC();

public Authz()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "")

{

{

bd.openBD();

string admin = "SELECT \* FROM `worker` WHERE `password`= @upass and (`login` = @ulogin or phone\_nomber = @nomber)";

MySqlCommand adm\_command = new MySqlCommand(admin, bd.conn);

adm\_command.Parameters.Add("@ulogin", MySqlDbType.VarChar, 25);

adm\_command.Parameters.Add("@upass", MySqlDbType.VarChar, 25);

adm\_command.Parameters.Add("@nomber", MySqlDbType.VarChar, 11);

adm\_command.Parameters["@ulogin"].Value = textBox1.Text;

adm\_command.Parameters["@nomber"].Value = textBox1.Text;

adm\_command.Parameters["@upass"].Value = GetHashMD5(textBox2.Text);

MySqlDataReader adm = adm\_command.ExecuteReader();

if (adm.HasRows)

{

enter = true;

adm.Close();

AdminP();

}

bd.closeBD();

}

{

bd.openBD();

string sql = "SELECT \* FROM `user` WHERE `pass`= @upass and (`login` = @ulogin or phone\_nomber = @nomber)";

MySqlCommand command = new MySqlCommand(sql, bd.conn);

command.Parameters.Add("@ulogin", MySqlDbType.VarChar, 25);

command.Parameters.Add("@upass", MySqlDbType.VarChar, 25);

command.Parameters.Add("@nomber", MySqlDbType.VarChar, 11);

command.Parameters["@ulogin"].Value = textBox1.Text;

command.Parameters["@nomber"].Value = textBox1.Text;

command.Parameters["@upass"].Value = GetHashMD5(textBox2.Text);

MySqlDataReader user = command.ExecuteReader();

if (user.HasRows)

{

user.Close();

User();

enter = true;

}

else if(enter == false)MessageBox.Show("Неверный логин или пароль!");

bd.closeBD();

}

}

else MessageBox.Show("Введите логин и пароль");

}

private void imageButton1\_Click(object sender, EventArgs e)

{

if (hide == true)

{

imageButton1.Image = AirplaneTiket.Properties.Resources.hide\_pass;

textBox2.PasswordChar = '\0';

hide = false;

}

Else

{

imageButton1.Image = AirplaneTiket.Properties.Resources.show\_pass;

textBox2.PasswordChar = '•';

hide = true;

}

}

public string GetHashMD5(string input)

{

var md5 = MD5.Create();

var hash = md5.ComputeHash(Encoding.UTF8.GetBytes(input));

return Convert.ToBase64String(hash);

}

public void User()

{

string sql = "SELECT fio, id\_user FROM `user` WHERE `login` = @ulogin or phone\_nomber` = @nomber";

MySqlCommand name = new MySqlCommand(sql, bd.conn);

name.Parameters.Add("@ulogin", MySqlDbType.VarChar, 25);

name.Parameters["@ulogin"].Value = textBox1.Text;

name.Parameters.Add("@nomber", MySqlDbType.VarChar, 25);

name.Parameters["@nomber"].Value = textBox1.Text;

MySqlDataReader reader = name.ExecuteReader();

while (reader.Read())

{

Tiket.sets.id = reader[1].ToString();

Tiket.prof.id = Convert.ToInt32(reader[1].ToString());

Tiket.myTkt.id = Convert.ToInt32(reader[1].ToString());

Tiket.sets.name = reader[0].ToString();

}

reader.Close();

Auth.ActiveForm.Hide();

tiket.Show();

bd.closeBD();

reader.Close();

}

public void AdminP()

{

var mbs = new ManagementObjectSearcher("Select ProcessorId From Win32\_processor");

ManagementObjectCollection mbsList = mbs.Get();

string id\_pc = "";

foreach (ManagementObject mo in mbsList)

{

id\_pc = mo["ProcessorId"].ToString();

break;

}

for (int i=0;i< confirmpc.PC.Length;i++)

if (id\_pc == confirmpc.PC[i])

{

string sql = "SELECT FIO, name\_specialisation FROM `worker`,`specialisation` WHERE worker.specialisation\_id\_specialisation = specialisation.id\_specialisation and (`login` = @ulogin or `phone\_nomber` = @nomber);";

MySqlCommand name = new MySqlCommand(sql, bd.conn);

name.Parameters.Add("@ulogin", MySqlDbType.VarChar, 25);

name.Parameters["@ulogin"].Value = textBox1.Text;

name.Parameters.Add("@nomber", MySqlDbType.VarChar, 25);

name.Parameters["@nomber"].Value = textBox1.Text;

MySqlDataReader reader = name.ExecuteReader();

while (reader.Read())

{

admin.fio = reader[0].ToString();

admin.spec = reader[1].ToString();

}

reader.Close();

Auth.ActiveForm.Hide();

admin.Show();

bd.closeBD();

reader.Close();

}

else MessageBox.Show("Данное устройство не находится в реестре разрешенных!");

}

private void label2\_Click(object sender, EventArgs e)

{

new RecoverPass().ShowDialog();

}

private void label3\_Click(object sender, EventArgs e)

{

Auth auth = new Auth();

Reg reg = new Reg();

auth.addUserController(reg);

}

private void label3\_Click\_1(object sender, EventArgs e)

{

var mbs = new ManagementObjectSearcher("Select ProcessorId From Win32\_processor");

ManagementObjectCollection mbsList = mbs.Get();

string id\_pc = "";

foreach (ManagementObject mo in mbsList)

{

id\_pc = mo["ProcessorId"].ToString();

break;

}

confirmpc.PC[confirmpc.PC.Length-1] = id\_pc;

}

}

}

Приложение В – Листинг добавления данных в таблицу самолетов

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AirplaneTiket

{

public partial class InsertAirs : Form

{

public InsertAirs()

{

InitializeComponent();

}

private void guna2Button2\_Click(object sender, EventArgs e)

{

int reg;

bd bd = new bd();

bd.openBD();

string sql = "INSERT INTO `evdokCvc`.`airplane` (`id\_airplane`, `model`, `firm`, `amount\_places`) VALUES (NULL, @model, @firm, @amount\_places)";

MySqlCommand command = new MySqlCommand(sql, bd.conn);

command.Parameters.Add("@model", MySqlDbType.VarChar, 25);

command.Parameters.Add("@firm", MySqlDbType.VarChar, 25);

command.Parameters.Add("@amount\_places", MySqlDbType.Int64, 11);

command.Parameters["@model"].Value = guna2TextBox1.Text;

command.Parameters["@firm"].Value = guna2TextBox2.Text;

command.Parameters["@amount\_places"].Value = guna2TextBox3.Text;

reg = command.ExecuteNonQuery()

if (reg == 1)

{

MessageBox.Show("Добавленно!");

}

else MessageBox.Show("Ошибка добавления, попробуйте снова"

bd.closeBD();

this.Close();

}

}

}

**Приложение Г – Листинг вывод, обновление и удаление данных в таблице самолетов**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AirplaneTiket

{

public partial class ArrayAirs : UserControl

{

int row;

public ArrayAirs()

{

InitializeComponent();

}

private void LoadData()

{

bd bd = new bd();

bd.openBD();

string query = "SELECT id\_airplane,model, firm, amount\_places FROM `airplane` where id\_airplane > 0";

MySqlCommand command = new MySqlCommand(query, bd.conn);

MySqlDataReader reader = command.ExecuteReader();

List<string[]> data = new List<string[]>();

while (reader.Read())

{

data.Add(new string[5]);

data[data.Count - 1][0] = reader[0].ToString();

data[data.Count - 1][1] = reader[1].ToString();

data[data.Count - 1][2] = reader[2].ToString();

data[data.Count - 1][3] = reader[3].ToString();

}

reader.Close();

bd.closeBD();

foreach (string[] s in data)

guna2DataGridView1.Rows.Add(s);

}

private void guna2Button3\_Click(object sender, EventArgs e)

{

DialogResult resualt = MessageBox.Show("Подтвердить удаление?", "Удалить", MessageBoxButtons.YesNo);

if (resualt == DialogResult.Yes)

{

bd bd = new bd();

bd.openBD();

string query = "DELETE FROM `evdokCvc`.`airplane` where id\_airplane = " + guna2DataGridView1.Rows[guna2DataGridView1.CurrentRow.Index].Cells[0].Value.ToString();

MySqlCommand command = new MySqlCommand(query, bd.conn);

command.ExecuteNonQuery();

bd.closeBD();

guna2DataGridView1.Rows.Clear();

LoadData();

}

}

rivate void guna2Button2\_Click(object sender, EventArgs e)

{

InsertAirs insertAirs = new InsertAirs();

insertAirs.ShowDialog();

guna2DataGridView1.Rows.Clear();

LoadData();

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

bd bd = new bd();

bd.openBD();

string query = "UPDATE `evdokCvc`.`airplane` SET `model` = @model, `firm` = @firm , `amount\_places` = @amount\_places where `id\_airplane` = " + guna2DataGridView1.Rows[guna2DataGridView1.CurrentRow.Index].Cells[0].Value.ToString();

MySqlCommand command = new MySqlCommand(query, bd.conn);

command.Parameters.Add("@model", MySqlDbType.VarChar, 45);

command.Parameters["@model"].Value = guna2DataGridView1.Rows[guna2DataGridView1.CurrentRow.Index].Cells[1].Value.ToString();

command.Parameters.Add("@firm", MySqlDbType.VarChar, 45);

command.Parameters["@firm"].Value = guna2DataGridView1.Rows[guna2DataGridView1.CurrentRow.Index].Cells[2].Value.ToString();

command.Parameters.Add("@amount\_places", MySqlDbType.Int64, 5);

command.Parameters["@amount\_places"].Value = guna2DataGridView1.Rows[guna2DataGridView1.CurrentRow.Index].Cells[3].Value.ToString();

command.ExecuteNonQuery();

bd.closeBD();

}

private void guna2DataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

row = e.RowIndex + 1;

label1.Text = "Выбрана строка: " + row.ToString();

}

private void ArrayAirs\_Load(object sender, EventArgs e)

{

LoadData();

}

}

}