2. Разработка АИС «Издательство»

2.1 Анализ предметной области

Издательство

|  |  |
| --- | --- |
| Функциональная область | Процесс |
| Кадры | Зарплатная политика |
| Бухгалтерский учет | Учет основных средств |
|  | Учет торговых операций |
|  | Учет оплаты труда |
| Финансы | Управление финансовыми ресурсами |
| Торговля | Управление заказами |

Таблица 5 – анализ предметной области

Запишем основные действия в виде следующей таблицы.

|  |
| --- |
| Прием заказа |
| Выписка счета |
| Оплата товара |
| Проверка оплаты |
| Заказ права на печать |
| Проверка прав на печать |

Таблица 6 – основные действия

Исследуем динамику процесса товарооборота с рассмотрением участников операций, используемой информации и документов.

К торговому менеджеру обращается клиент, желающий выпустить в печать свою книгу. Может появиться лично или позвонить.

Менеджер собирает всю информацию, которая нужна для отправки данной книги на печать (будет ли интересно это аудитории, окупится ли в итоге данное действие, какую плату запросить и т.д.).

Далее после согласования каких-либо рабочих моментов происходит оплата. После оплаты согласовывается дата начала печати и выпуска книги в свет.

2.2 Техническое задание:

2.2.1 Требования к функциональным характеристикам

2.2.1.1 Состав выполняемых функций

Разрабатываемое ПО должно обеспечивать:

* сбор и анализ информации о заказах;
* сбор и анализ отзывов;
* возможность видеть отзыв на ту или иную книгу в издательстве;
* возможность увидеть какой работник ответственен за издательство;
* выдачу рекомендаций по дальнейшей работе;
* отображение заказов всех гостей;

В конце каждого месяца система должна архивировать данные.

2.2.1.2. Организация входных и выходных данных

Исходные данные в систему поступают в виде информации от менеджера, который сообщает что клиент подал заявку на издательство своей книги. Эти значе­ния отображаются на компьютере менеджера и оператора. После анализа по­ступившей информации оператор сообщает работнику, чтоб он проверил возможность осуществления издания данного материала (хватит ресурсов ли, окупится ли и т.д.).

Основной режим использования системы - ежедневная работа.

2.2.2 Требования к надежности

Для обеспечения надежности необходимо проверять кор­ректность введенных данных.

2.2.2.3.Условия эксплуатации и требования к составу и параметрам технических средств

Для работы системы должен быть выделен ответственный менеджер.

Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

2.2.4 Требования к информационной и программной совмес­тимости

Программа должна работать на платформах Windows 8/10/11.

2.2.5 Требования к транспортировке и хранению. Программа поставляется на дисковом носителе информации или отправляется на электронную почту.

Программная документация поставляется в электронном и пе­чатном виде.

2.2.6 Специальные требования

Интерфейс должен быть понятен для людей с средним уровнем знаний компьютера.

Требования к программной документации

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой Системы Программной Документации (ЕСПД); руководство пользовате­ля, руководство администратора, описание применения.

Технико-экономические показатели

Эффективность системы определяется удобством и скоростью работы данного ПО.

Порядок контроля и приемки

После передачи Исполнителем отдельного функционального модуля программы Заказчику, последний имеет право тестиро­вать модуль в течение 5 дней. После тестирования Заказчик дол­жен принять работу по данному этапу или в письменном виде изложить причину отказа от принятия. В случае обоснованного отказа Исполнитель обязуется доработать модуль

2.3 ER-диаграмма сущность-связь

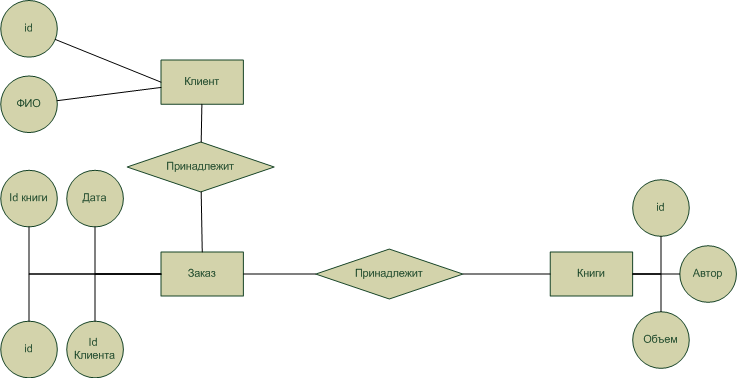


Рисунок 9 – ER-диаграмма сущность-связь

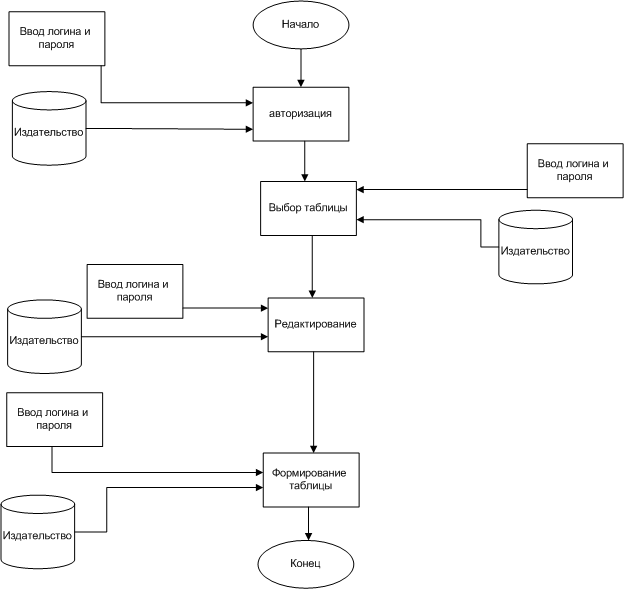
2.4 Функциональная система АИС

Рисунок 10 – Функциональная система АИС

Код АИС

package com.company;  
  
import java.sql.\*;  
import java.util.Scanner;  
public class Main {  
 static String username = "root";  
 static String password = "123456789";  
 static String url = "jdbc:mysql://127.0.0.1:3306/mydb";  
 static Scanner scan = new Scanner(System.in);  
 public static void main(String[] args) {  
  
 }  
 public static void vvod(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement add = conn.createStatement();  
 System.out.println("Введите автора книги");  
 String a = scan.next();  
 System.out.println("Введите название книги");  
 String b = scan.next();  
 System.out.println("Введите количество страниц книги");  
 int c = scan.nextInt();  
 System.out.println("Введите язык книги");  
 String d = scan.next();  
 int rows = add.executeUpdate("INSERT tabl(avtor, name, obyem, language) VALUES ('"+a + "'," +"'"+b+ "'," +"'" +c+ "',"+"'"+d+"')");  
 } catch (SQLException e) {  
 System.out.println("Connection failed...");  
 System.out.println(e);  
 }  
 }  
 public static void vivodtabl(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 ResultSet resultSet = statement.executeQuery("SELECT \* FROM tabl");  
 while(resultSet.next()){  
 int id = resultSet.getInt(1);  
 String avtor = resultSet.getString(2);  
 String name = resultSet.getString(3);  
 int obyem = resultSet.getInt(4);  
 String language = resultSet.getString(5);  
 System.out.println(id + " " + avtor+ " " + name+ " " + obyem+ " " +language +"\n");  
 }  
 }  
 catch (SQLException ex) {  
 System.out.println("Connection failed...");  
 System.out.println(ex);  
 }  
 }  
 public static void vivodpolanguage(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 System.out.println("Введите язык книги");  
 String z = scan.next();  
 ResultSet resultSet = statement.executeQuery("SELECT \* FROM tabl WHERE language='" +z+"'");  
 while(resultSet.next()){  
 int id = resultSet.getInt(1);  
 String avtor = resultSet.getString(2);  
 String name = resultSet.getString(3);  
 int obyem = resultSet.getInt(4);  
 String language = resultSet.getString(5);  
 System.out.println(id + " " + avtor+ " " + name+ " " + obyem+ " " +language +"\n");  
 }  
 }  
 catch (SQLException ex) {  
 System.out.println("Connection failed...");  
 System.out.println(ex);  
 }  
 }  
 public static void udalenie(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 System.out.println("Введите id строчки для удаления");  
 int b = scan.nextInt();  
 int rows = statement.executeUpdate("DELETE FROM tabl WHERE Id ="+b);  
 } catch (SQLException e) {  
 System.out.println("Connection failed...");  
 System.out.println(e);  
 }  
 }  
}

В коде программы есть несколько методов, метод vvod, чтоб добивать новую запись в таблицу, vivodtabl, позволяющий вывести данные присутствующие в базе данных, udalenie, позволяющий удалить запись из базы данных, vivodpolanguage, позволяющий вывести БД с выборкой по языку книги.

Вывод изначальной таблицы с помощью метода vivodtabl

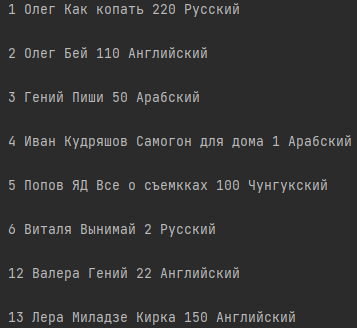


Рисунок 11 – Вывод изначальных данных в бд

Добавления строки в таблицу и последующий вывод таблицы с помощью методов vvod и vivodtabl

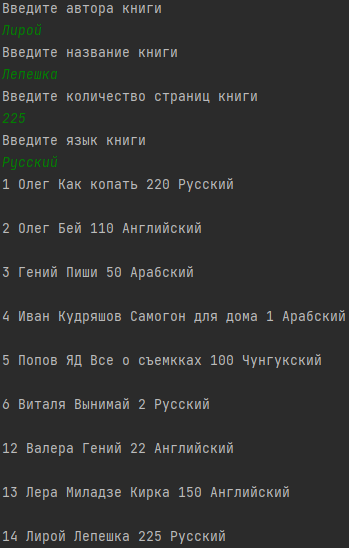


Рисунок 12 – добавление строки и последующий вывод данных их БД

Удаление строки по введеному id с помощью метода udalenie и последующий вывод БД методом vivodtabl

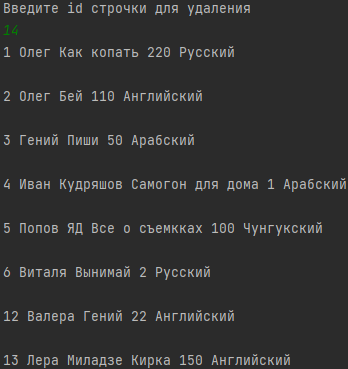


Рисунок 13 – удаление строки из БД