Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗработка приложения для автоматизации работы библиотеки

Отчёт по производственной практике

ПМ.01Разработка модулей программного обеспечения для компьютерных систем

НАТКиГ.720200.010.000

Разработал:

студент группы ПР-20.102к

Герман Р.А.

2023

**Содержание**

[Введение 3](#_Toc136994067)

[1 Описание предметной области 4](#_Toc136994068)

[2 Анализ программных ресурсов, необходимых в работе 5](#_Toc136994069)

[3 UML-диаграммы 6](#_Toc136994070)

[3.1 Диаграмма прецедентов 6](#_Toc136994071)

[3.2 Диаграмма спецификаций 7](#_Toc136994072)

[4 Диаграмма потока данных 8](#_Toc136994073)

[5 ER диаграмма 9](#_Toc136994074)

[6 Описание разрабатываемых процедур и функций проекта 10](#_Toc136994075)

[Заключение 12](#_Toc136994076)

[Библиография 13](#_Toc136994077)

Введение

В современном информационном обществе автоматизация играет ключевую роль в повышении эффективности работы различных организаций. Одной из таких организаций является библиотека, которая служит важным источником знаний и информации для общества. Однако управление библиотекой может стать сложной задачей, особенно при увеличении объема книжного фонда и численности пользователей

В рамках производственной практики была поставлена задача разработки программного решения для автоматизации работы библиотеки. Целью данного проекта является создание программы, которая позволит эффективно управлять сотрудниками, книгами, каталогом книг и информацией о читателях, а также обеспечить надежную систему авторизации.

Основными функциональными требованиями к программе будут управление информацией о сотрудниках, авторизация пользователей, управление книгами и каталогом книг, а также информацией о читателях. Администратор должен иметь возможность добавлять, редактировать и удалять информацию о сотрудниках, а главный библиотекарь должен иметь возможность управлять информацией об авторах, жанрах книг и самих книгах. Библиотекарь, в свою очередь, должен иметь возможность управлять информацией о читателях и выдавать им книги.

Таким образом, разработка программного решения для автоматизации работы библиотеки будет способствовать оптимизации процессов управления, повышению эффективности и обеспечению более удобного доступа к информации для пользователей.

# Описание предметной области

В современном информационном обществе библиотека играет важную роль в предоставлении доступа к знаниям и информации. Она является центром, где люди могут получить доступ к разнообразным источникам информации, искать ответы на свои вопросы, углублять свои знания и развиваться. Однако с увеличением объема книжного фонда и численности пользователей управление библиотекой может стать сложной задачей.

Для эффективной организации работы и обеспечения удобства пользователей необходимо разработать систему управления библиотекой. В данной системе взаимодействуют несколько действующих лиц. Администратор отвечает за общее управление библиотекой, контроль доступа пользователей и настройку системы. Главный библиотекарь управляет каталогом книг, информацией об авторах и жанрах, а также организует выставки и мероприятия. Библиотекарь обслуживает читателей, регистрирует новых читателей, выдает и принимает книги.

В системе управления библиотекой присутствуют различные объекты и их свойства. Сотрудники библиотеки имеют такие свойства, как имя, фамилия, должность, контактные данные и данные для авторизации в системе. Книги включают информацию о названии, авторе, издательстве, годе издания, жанре, ISBN, количестве экземпляров и доступности. Читатели имеют свойства, включающие имя, фамилию, дату рождения, контактные данные и адрес проживания.

Процессы взаимодействия в системе управления библиотекой включают регистрацию сотрудников, управление каталогом книг, регистрацию читателей, выдачу и возврат книг, аутентификацию и авторизацию пользователей. При регистрации сотрудников администратор добавляет информацию о новых сотрудниках и назначает им данные для авторизации. Главный библиотекарь отвечает за управление каталогом книг, добавляя новые книги и редактируя информацию об авторах и жанрах. Библиотекарь регистрирует новых читателей и осуществляет выдачу и возврат книг. Пользователи системы вводят логин и пароль для аутентификации, после чего система проверяет эти данные и предоставляет соответствующие права доступа.

Система управления библиотекой позволяет эффективно организовать работу библиотеки, обеспечить удобство для пользователей и повысить доступность знаний и информации. Взаимодействие администратора, главного библиотекаря, библиотекаря и читателей в рамках определенных процессов обеспечивает эффективное функционирование библиотеки и удовлетворение потребностей пользователей.

# Анализ программных ресурсов, необходимых в работе

Для разработки программного решения по автоматизации работы библиотеки были использованы следующие программные средства:

Среда разработки Microsoft Visual Studio – Это интегрированная среда разработки (IDE), которая предоставляет все необходимые инструменты и функционал для создания приложений на различных платформах. В данном проекте использовалась версия Visual Studio, поддерживающая разработку на языке программирования C#.

Язык программирования C# выбран в качестве основного языка разработки для создания программного решения. C# является объектно-ориентированным языком программирования, разработанным Microsoft, и широко используется для создания Windows-приложений.

Графический интерфейс Windows Presentation Foundation (WPF) – WPF является технологией для создания графических пользовательских интерфейсов в Windows-приложениях. Он предоставляет богатые возможности для создания интерактивных и привлекательных пользовательских интерфейсов. WPF был использован для разработки пользовательского интерфейса программы управления библиотекой.

База данных Microsoft SQL Server – Была использована Microsoft SQL Server в качестве базы данных для хранения информации о сотрудниках, книгах и читателях. SQL Server является реляционной системой управления базами данных (СУБД), разработанной Microsoft. Он обеспечивает надежное хранение данных и эффективные средства работы с ними.

Взаимодействие с базой данных ADO.NET – является набором технологий для доступа к данным в приложениях .NET. Он предоставляет программистам возможность взаимодействовать с базами данных, включая Microsoft SQL Server. ADO.NET использовался для установления соединения с базой данных и выполнения операций чтения и записи данных.

# UML-диаграммы

## Диаграмма прецедентов

На рисунке 1 представлена диаграмма прецедентов.

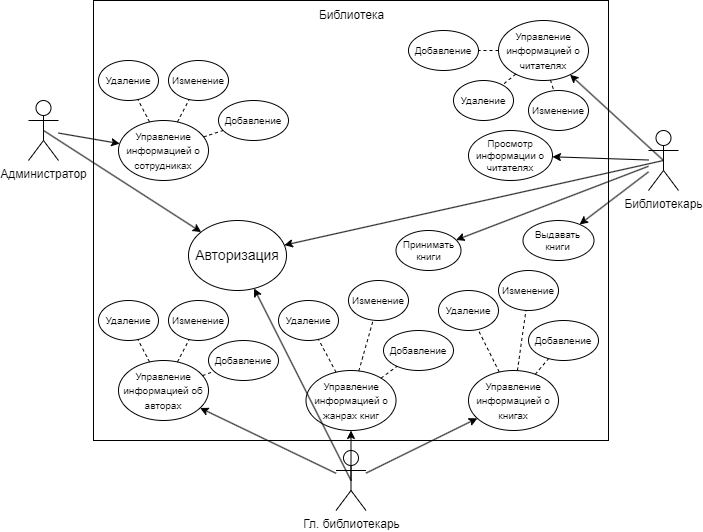


Рисунок 1 – Диаграмма прецедентов

Прецедент "Авторизация" позволяет пользователю войти в систему, вводя свой логин и пароль.

Прецедент "Управление информацией о сотрудниках": Администратор может добавлять, редактировать и удалять информацию о сотрудниках.

Прецедент "Управление информацией об авторах": Главный библиотекарь может просматривать, добавлять, редактировать и удалять информацию об авторах.

Прецедент "Управление информацией о жанрах книг": Главный библиотекарь может просматривать, добавлять, редактировать и удалять информацию о жанрах книг.

Прецедент "Управление информацией о книгах": Главный библиотекарь может просматривать, добавлять, редактировать и удалять информацию о книгах.

Прецедент "Управление информацией о читателях": Библиотекарь может просматривать, добавлять, редактировать и удалять информацию о читателях.

Прецедент "Просмотр информации о читателях": Библиотекарь может просматривать информацию о читателях.

Прецедент "Выдавать книги": Библиотекарь может выдавать книги читателям, указывая дату выдачи и дату возврата.

Прецедент "Принимать книги": Библиотекарь может принимать возвращенные читателями книги и обновлять информацию о статусе книги (доступна/недоступна).

## Диаграмма спецификаций

В таблице 1 представлена спецификация прецедента «Авторизация».

Таблица 1— Спецификация для прецедента «Авторизация»

|  |  |
| --- | --- |
| Раздел | Описание |
| Краткое описание | Позволяет авторизоваться пользователю в программе |
| Действующие лица | Администратор, Главный библиотекарь, Библиотекарь |
| Предусловия | Необходимость авторизоваться в программе |
| Основной поток | 1. Система открывает окно авторизации  2. Пользователь вводит имя пользователя и пароль  3. Система сравнивает логин и пароль с имеющимися записями в базе  А1. Совпадений нет  3. Система открывает главный экран приложения |
| Альтернативный поток | А1. Совпадений нет  1.Система выводит сообщение «Неверно введен логин или пароль»  2. Пользователь подтверждает просмотр сообщения  3. Поток возвращается на 2 этап основного потока |
| Постусловие | Результатом является авторизация в систему |

В процессе работы над спецификациями прецедентов были определены основные прецеденты и актеры для системы, были указаны шаги, необходимые для выполнения прецедента.

# Диаграмма потока данных

Здесь необходимо дать краткое описание и представить на рисунке диаграмму. Диаграмма должна располагаться так, чтобы занимать большую часть страницы. Над рисунком должно быть обращение к нему, например, как представлено на рисунке 3.

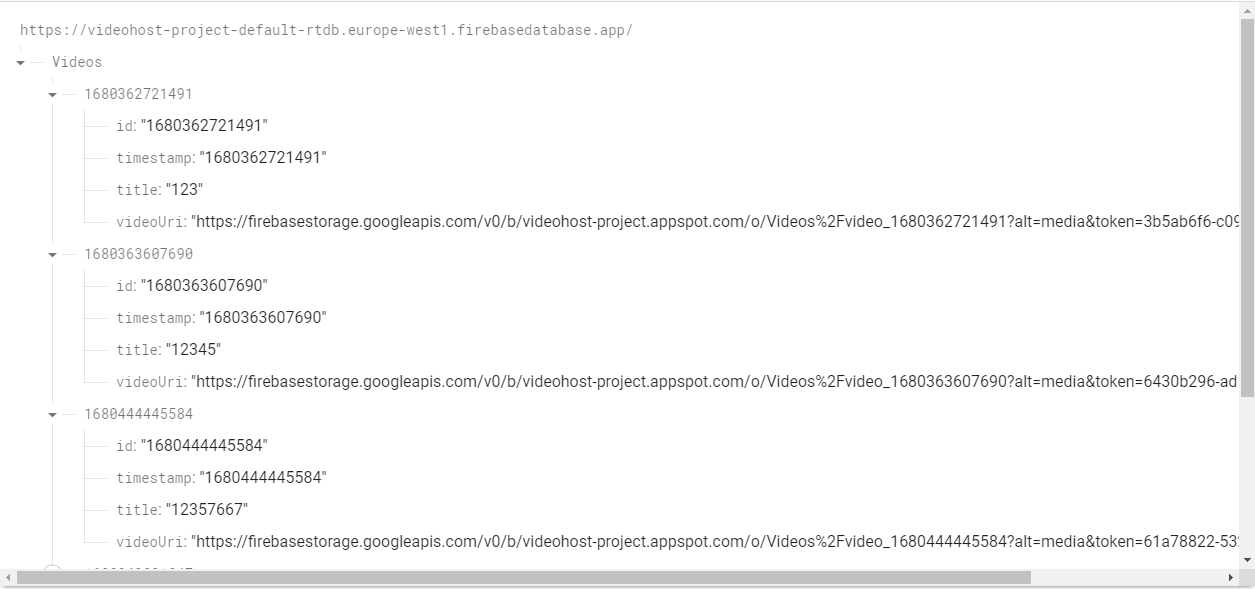


Рисунок 3 – Диаграмма потока данных

Краткий вывод или продолжение описания. После рисунка необходимо добавить 1 пустую строку.

# ER диаграмма

На рисунке 2 расположена ER-диаграмма, которая включает в себя такие сущности как: Readers, Genres, Authors, Books, IssuedBooks, Employees, EmployeeTypes.

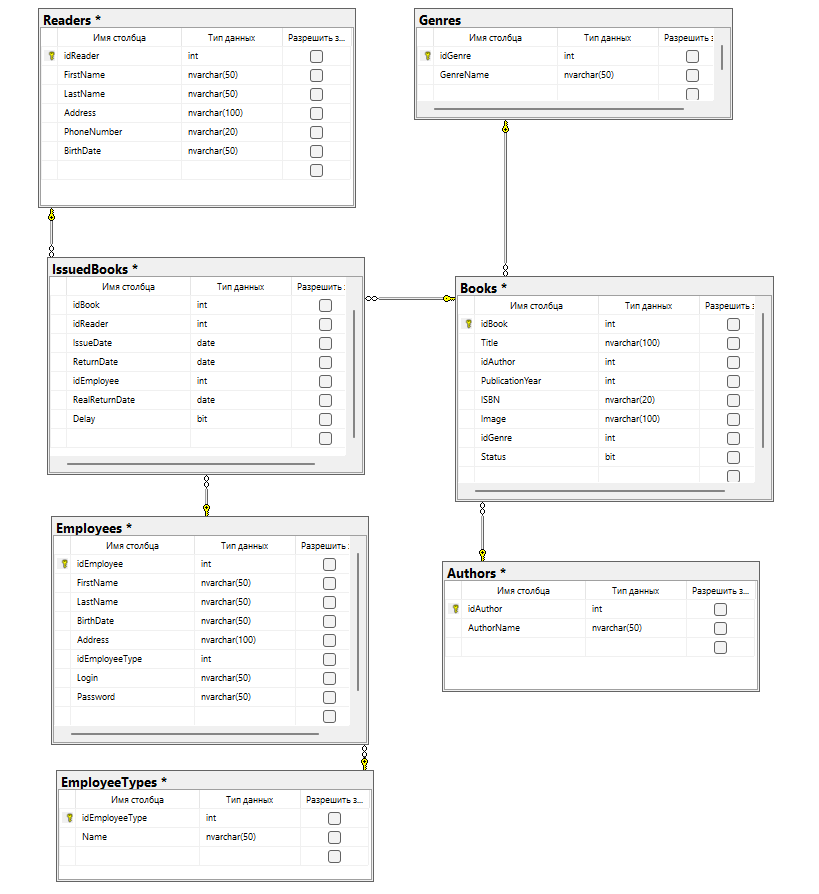


Рисунок 2 – ER Диаграмма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idAuthor | int | Yes | Auto increment |
|  | AuthorName | nvarchar(50) | No |  |

Таблица Authors

Таблица Books

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idBook | int | Yes | Auto increment |
|  | Title | nvarchar(100) | No |  |
| FK | idAuthor | int | No |  |
|  | PublicationYear | int | No |  |
|  | ISBN | nvarchar(20) | No |  |
|  | Image | nvarchar(100) | No |  |
| FK | idGenre | int | No |  |
|  | Status | bit | No |  |

Таблица Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idEmployee | int | Yes | Auto increment |
|  | FirstName | nvarchar(50) | No |  |
|  | LastName | nvarchar(50) | No |  |
|  | BirthDate | nvarchar(50) | No |  |
|  | Address | nvarchar(100) | No |  |
| FK | idEmployeeType | int | No |  |
|  | Login | nvarchar(50) | No |  |
|  | Password | nvarchar(50) | No |  |

Описать словарь данных. Перечислить все сущности ERD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idGenre | int | Yes | Auto increment |
|  | GenreName | nvarchar(50) | No |  |

Таблица EmployeeTypes

Таблица Genres

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idGenre | int | Yes | Auto increment |
|  | GenreName | nvarchar(50) | No |  |

Таблица IssuedBooks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idIssuedBook | int | Yes | Auto increment |
| FK | idBook | int | No |  |
| FK | idReader | int | No |  |
|  | IssueDate | date | No |  |
|  | ReturnDate | date | No |  |
| FK | idEmployee | int | No |  |
|  | RealReturnDate | date | No |  |
|  | Delay | bit | No |  |

Таблица Readers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | idReader | int | Yes | Auto increment |
|  | FirstName | nvarchar(50) | No |  |
|  | idAuthor | int | No |  |
|  | LastName | nvarchar(50) | No |  |
|  | Address | nvarchar(100) | No |  |
|  | PhoneNumber | nvarchar(20) | No |  |
|  | BirthDate | nvarchar(50) | No |  |

# Описание разрабатываемых процедур и функций проекта

Перечислить разработанные функции продукта.

Например:

В приложении разработаны следующие функции:

* регистрация и авторизация;
* переход между страницами с помощью навигационного меню;
* добавление видеоролика на канал;
* добавление видеоролика в избранное;
* просмотр видеороликов.

Описать 1-2 функции подробнее.

Например:

Авторизация в приложении реализована по средству запросов в базу данных. Ниже представлен код авторизации.

btnLog.setOnClickListener **{** when{  
 TextUtils.isEmpty(txtMail.*text*.toString().*trim* **{ it** <= ' ' **}**) -> {  
 Toast.makeText(  
 this@LoginActivity,  
 "Пожалуйcта, заполните поле - e-mail.",  
 Toast.*LENGTH\_SHORT* ).show()  
 }  
 TextUtils.isEmpty(txtPass.*text*.toString().*trim* **{ it** <= ' ' **}**) -> {  
 Toast.makeText(  
 this@LoginActivity,  
 "Пожалуйcта, заполните поле - Пароль.",  
 Toast.*LENGTH\_SHORT* ).show()  
 }  
 else -> {  
 val email: String = txtMail.*text*.toString().*trim* **{ it** <= ' ' **}** val password: String = txtPass.*text*.toString().*trim* **{ it** <= ' ' **}** // Входим как юзер.  
 FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener **{** task **->** //Если вход прошёл успешно  
 if (task.*isSuccessful*) {  
  
 Toast.makeText(  
 this@LoginActivity,  
 "Вы успешно авторизировались.",  
 Toast.*LENGTH\_SHORT* ).show()  
  
 val intent = Intent (this@LoginActivity, MainActivity::class.*java*)  
 intent.*flags* = Intent.*FLAG\_ACTIVITY\_NEW\_TASK* or Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* intent.putExtra("user\_id", FirebaseAuth.getInstance().*currentUser*!!.*uid*)  
 intent.putExtra("email\_id" , email)  
 startActivity(intent)  
 finish()  
 } else {  
 // Если что-то пошло не так и вылезла ошибка  
 Toast.makeText(  
 this@LoginActivity,  
 task.*exception*!!.message.*toString*(),  
 Toast.*LENGTH\_SHORT* ).show()  
 }  
 **}** }  
 }  
**}**

Сделать небольшой вывод по итогу разрабатываемых функций. В листинге кода должны присутствовать комментарии, после листинга кода необходимо добавить 1 пустую строку.

Заключение

Написать вывод по итогу разработки индивидуального задания по производственной практике. Не менее 0,8 страницы и не более 2 страниц.

Библиография

1. AndroidDevelopers [электронный ресурс]: ДокументацияAndroidStusio – https://developer.android.com/docs/
2. HelpCenter [электронныйресурс]: ДокументацияJava: https://docs.oracle.com/en/java/
3. Автостат[электронный ресурс]:Структура российских автовладельцев по возрасту и маркамhttps://www.autostat.ru/infographics/40710/
4. Firebase [электронныйресурс]: Документация Firebase - https://firebase.google.com/docs?hl=uk
5. Material Design [электронныйресурс]: BottomNavigation: https://material.io/components/bottom-navigation/android
6. Git[электронныйресурс]: ДокументацияGit: https://git-scm.com/
7. Figma[электронныйресурс]: Прототипирование проектовhttps://www.figma.com/

Перечислить источники, если происходило исследование или обращение к каким-либо источникам.