МИНИСТЕРСТВО ОБРАЗОВАНИЯ И МОЛОДЕЖНОЙ ПОЛИТИКИ СВЕРДЛОВСКОЙ ОБЛАСТИ
ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ КОЛЛЕДЖ ТРАНСПОРТНОГО СТРОИТЕЛЬСТВА»

Разработка и создание программы для кинотеатра

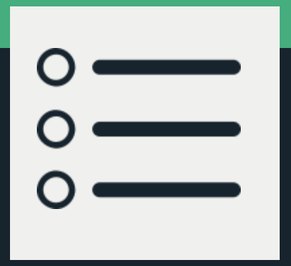
Работу выполнила: Корпушенко К.Р.
Руководитель: Мирошниченко Г.В.
Группа: Пр-31

О разработке

Программа предназначена для **организации деятельности кинотеатра**. Продукт позволяет **систематизировать рабочие процессы** кинотеатра.
Данная разработка - это **программный комплекс**, предназначенный для ведения учёта фильмов в прокате, билетов к данным фильмам, регистрации покупки и возврата билетов

Основной учет

Система



Прокат фильмов

Цели и задачи проекта

Целью является разработка программы для системы процессов продажи билетов в кинотеатре от создания проката фильмов до учета пользователей и продажи билетов

ЗАДАЧИ

- Изучить актуальную информацию по области данной задачи
- Подобрать средства для разработки программного продукта
- Подобрать наиболее удобный для пользователя дизайн
- Учесть требования к задаче и разработать подобранный интерфейс

Функциональные характеристики

ПРИВЛЕКАТЕЛЬНОСТЬ

Хороший интерфейс должен быть привлекательным, чтобы доставлять пользователю удовольствие при работе с программным продуктом.

ЛАКОНИЧНОСТЬ И ПРОСТОТА

Интерфейс не должен быть перегружен лишней информацией.

СИСТЕМАТИЗАЦИЯ

Упрощения и автоматизации операций, связанных с регистрацией, систематизацией, поиском и обработкой данных о пользователях

Используемые инструменты

Navigation

ROOM

API

Главное окно программы

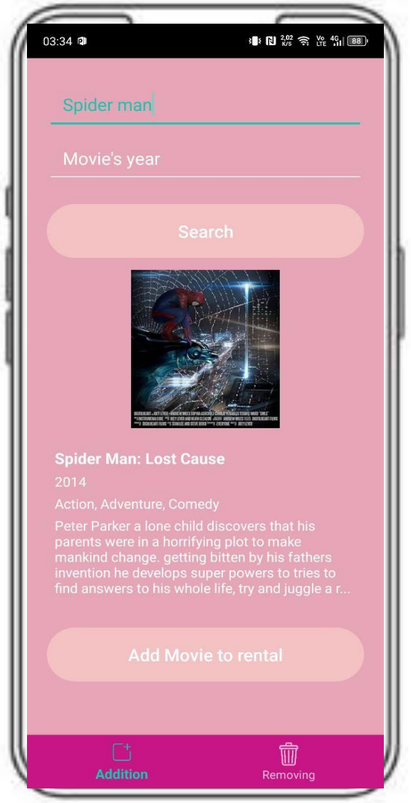
включает в себя:

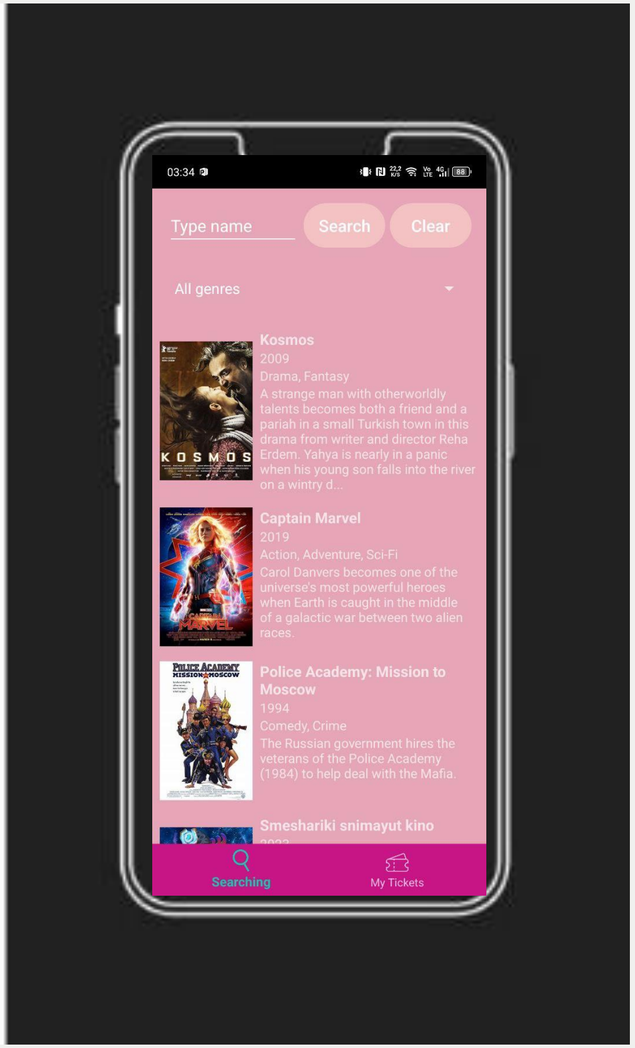
2 режима работы:

-для администратора

-для пользователя

Главным окном является активность, на которой расположены меню для навигации между экранами и контейнер для фрагментов экрана





Простота интерфейса способствует взаимодействию пользователя с информацией

Форма фильмы

Таблица «movies» - содержит информацию об имеющихся фильмах в прокате.
Возможно не только добавление фильмов, но и создание учета на основании таблицы.

```
@Entity(tableName = "movies")
data class Movie(
    @PrimaryKey(autoGenerate = true)
    val id: Long = 0,
    val title: String,
    val genre: String,
    val description: String,
    val poster: String,
    val year: String
)
```

Хранение информации о фильмах в базе кинотеатра

Форма пользователи

Таблица «users» - содержит информацию о пользователях:

- Логин
- Электронная почта
- Пароль
- Его роль в системе: администратор/пользователь

```
@Entity(tableName = "users")
data class User(
    @PrimaryKey(autoGenerate = true)
    val id: Long = 0,
    val username: String,
    val email: String,
    val password: String,
    val role: String // Роль: "User" или "Admin"
)
```

База данных с информацией о пользователях позволяет программе работать со спецификой его использования, например, для покупки билетов или для определения, кто работает в приложении и с какой целью

Форма билеты

включает в себя:

- Ряд
- Место
- Фильм, на который был выпущен билет
- Логин владельца (Поле может быть пустым, если билет ещё не был куплен)

```
@Entity(
    tableName = "tickets",
    foreignKeys = [
        ForeignKey(
            entity = Movie::class,
            parentColumns = ["id"],
            childColumns = ["movieId"],
            onDelete = ForeignKey.CASCADE
        )
    ],
    indices = [Index(value = ["movieId"])]
)

data class Ticket(
    @PrimaryKey(autoGenerate = true) val id: Long = 0,
    val row: Int,
    val seat: Int,
    val movieId: Long,
    var ownerUsername: String? = null // Логин владельца или null, если билет не куплен
)
```

База данных с информацией о билетах на фильм и информации о его владельце, что позволяет программе вести учет о продаже билетов

Перспективы разработки

Добавление большего функционала, синхронизация БД на сервере в интернете

Расширение функционала настройки интерфейса, Добавление интеграции платежа возврата средств

Оптимизация кода, редактирование базы данных