



**ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО
КОМПЮТЪРНО МОДЕЛИРАНЕ И
КОМПЮТЪРНИ СИСТЕМИ**

„Акад. Благовест Сендов“

ДИПЛОМНА РАБОТА

Тема:

**Разработка на система за управление на
киносалон**

Изготвил:

Кемал Антонов Иванов

**СПЕЦИАЛНОСТ ОТ ПРОФЕСЯ: „Системно
Програмиране“**

ПРОФЕСИЯ: „Системен Програмист“

ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ:

„Компютърни науки“

Ръководител: инж. Вяра Дукова

Варна 2024 г.

1.Съдържание

1.Съдържание.....	3
2.Увод.....	4
2.1 Съществуващи решения.....	6
2.2 Цел на дипломната работа и задачи.....	7
2.3 Задачи за реализиране.....	8
3.Избор на технологии.....	9
3.1 C#.....	9
3.2 WPF (Windows Presentation Foundation):.....	10
3.3 MySQL:.....	11
3.4 GitHub:.....	12
4.Проектиране и реализация на приложението.....	13
4.1 Описание на приложението.....	13
4.2 Описание на кода по страници.....	15
4.2.1 Идентични елементи по страниците.....	15
4.2.2 Описание на Main Page.....	18
4.2.3 Описание на Login Window.....	22
4.2.4 Описание на Select Tickets.....	23
4.2.5 Описание на Select Seats.....	31
4.2.6 Описание на Manage Halls.....	36
4.2.7 Описание на Schedule Panel.....	38
5.Бази данни.....	40
5.1 Схема.....	40
5.2 Връзки.....	41
5.2.1 Users и Tickets.....	41
5.2.2 Movies и Screenings.....	41
5.2.3 Halls и Screenings.....	41
5.2.4 Screenings и Tickets.....	41
5.2.5 Halls и TicketTypes.....	41
6. Потребителски интерфейс.....	42
6.1 Клиентско приложение.....	42
6.1.1 Login страница.....	42
6.1.2 Главна страница.....	43
6.1.3 Избор на места.....	44
6.2 Администраторско приложение.....	45
7. Бъдещо развитие на приложението.....	47
8. Заключение.....	49
9. Използвана литература.....	51

2. Увод

Със съчетанието на вълнуващи филми, съраунд звук и уникална атмосфера, киноиндустрията продължава да привлича хора от всички възрасти и културни среди. За много хора по целия свят, посещението в киносалона е не просто изживяване, а истинско приключение, което предоставя възможност за емоционално вълнение и откриване на нови светове.

С развитието на технологиите киносалоните са се трансформирали в центрове на развлеченията, предлагайки не само висококачествени филмови прожекции, но и цялостни преживявания за своите посетители. Модерните киносалони включват последни технологии за възпроизвеждане на видео и аудио, които гарантират възможно най-реалистично изживяване за зрителите.

В допълнение към това, киносалоните предлагат различни допълнителни услуги и удобства, които подобряват престоя на посетителите. Това включва комфортни седалки, 4DX®, ресторантски услуги, барове с напитки и закуски, както и луксозни VIP зони и ексклузивни прожекции. Всичко това цели да осигури най-доброто възможно изживяване за зрителите и да ги върже във вълнуващия свят на киното.

Въпреки значителното развитие в киноиндустрията и нейната еволюция като форма на забавление и културно обогатяване, мнозина киносалони все още се сблъскват с предизвикателства, свързани с управлението на техните операции. Липсата на централизирана и ефективна система за управление може да доведе до редица проблеми, които засягат както клиентите, така и бизнеса в цялост.

Неефективното разпределение на ресурсите може да доведе до загуби от продажби и недостиг на рентабилност за киносалоните. Недоволството на клиентите от лошо организирани събития може да доведе до намаляване на тяхната лоялност и доверие към марката на киносалона.

2.1 Съществуващи решения

Съществуват множество решения за проблема повечето под формата на уеб сайт най-популярните от които са <https://www.kinoarena.com/bg/> и <https://www.cinemacity.bg/> там могат да се намерят всички актуални филми които са на прожекция.

Главните недостатъци на тези сайтове са:

- Ограничен достъп до информацията: Въпреки че сайтовете предоставят списък с актуални филми и прожекции, информацията за конкретни прожекции, като свободни места и цени, често е ограничена. Потребителите често трябва да извършват няколко клика, за да достигнат до нужната информация.
- Липса на персонализация: Сайтовете не предлагат персонализирани препоръки или функционалности, които да улеснят потребителите в търсенето и избора на филми. Липсата на персонализация може да доведе до по-трудно намиране на желаните филми и прожекции.
- Ограничена функционалност за резервации и плащания: Въпреки че сайтовете позволяват регистрация и резервации за билети, процесът е често ограничен и неинтуитивен. Някои от тях могат да представляват проблеми при плащането или при добавянето на допълнителни услуги към резервацията.
- Липса на социални функционалности: Някои потребители може да търсят социални функции като споделяне на преживявания, коментари и оценки за филмите. Този аспект липсва в голяма степен или е ограничен във вече съществуващите решения. Изброените недостатъци подчертават възможността за разработване на по-интегрирана и усъвършенствана система за управление на киносалон, която да адресира тези проблеми и да предложи по-добър потребителски опит.

2.2 Цел на дипломната работа и задачи

Целта на представената дипломна работа е да се намерят тези проблеми чрез разработка и внедряване на система за управление на киносалон. Тази система ще обхване всички основни аспекти на операциите на киносалона, включително управление на програмата, билети, резервации и анализ на данни. Чрез автоматизация и оптимизация на тези процеси, системата ще подобри ефективността на работата на киносалона и ще осигури по-добро обслужване за клиентите.

Разработването на тази система със съвременни технологии е ключът към нейния успех и адаптирането към нуждите на киносалоните в днешно време. Тя трябва да бъде лесна за използване и гъвкава, за да може да се адаптира към различните модели на бизнес и специфичните изисквания на киносалоните. В крайна сметка, разработката на тази система има за цел да подобри както оперативните процеси, така и клиентското изживяване в съвременните киносалони.

След разработката на системата за управление на киносалон, процесът няма да приключи, а ще продължи с редица важни етапи. Освен основните стъпки като анализ, проектиране, разработка и тестване, следва да се отдели внимание и на интеграцията на новата система с вече съществуващите системи и услуги. Това е от съществено значение, тъй като плавният преход и минималните прекъсвания в ежедневноста дейност на киносалоните са от ключово значение за успеха на проекта.

След изпълнението на всички етапи от разработката и успешната интеграция, се очаква разработената система да допринесе за значително подобрене на ефективността и конкурентоспособността на киносалоните. Предвижда се, че тя ще осигури по-добро обслужване на клиентите, като предостави лесно за използване софтуерно приложение,

което да оптимизира работата на персонала и да подобри общото клиентско изживяване.

В същото време, разработената система ще създаде възможности за иновации и развитие в индустрията на кино и развлеченията. Чрез постоянно подобрене и интеграция на нови функции и технологии, киносалоните ще могат да се адаптират към променящите се потребителски изисквания и да останат конкурентни на пазара.

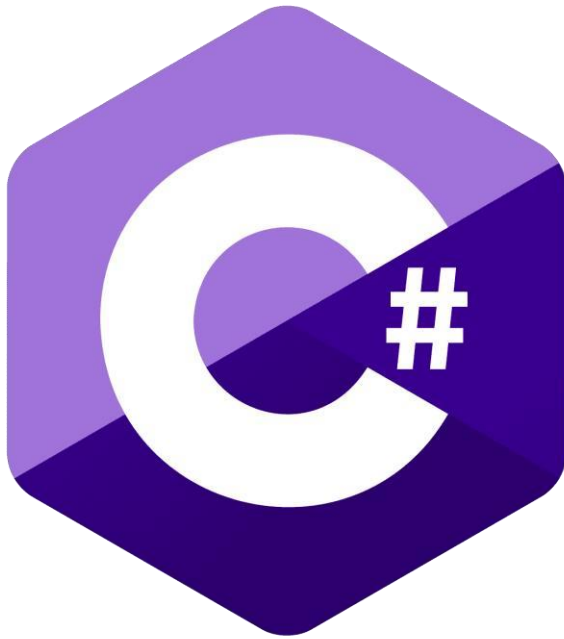
В дипломната работа ще бъдат представени детайлни анализи, проекти и реализации на разработената система, които ще обяснят как новата технология може да се интегрира в ежедневната дейност на киносалоните. Също така, ще бъде направено изследване на възможностите за бъдещо развитие и подобрения, което ще осигури дългосрочна успешност и иновативно развитие на индустрията.

2.3 Задачи за реализиране

- Максимално лесен и изчистен дизайн
- Оптимизиран код
- Подобрени функционалности за резервации и плащания:

3.Избор на технологии

3.1 C#



Фигура 1

Език за програмиране с широко приложение: C# е мощен и гъвкав език за програмиране, който се използва за разработване на разнообразни приложения, включително десктоп, уеб и мобилни приложения. Поддръжка на обектно-ориентирано програмиране: C# поддържа на основни принципи на обектно-ориентираното програмиране като инкапсулация, наследяване и полиморфизъм, C# позволява по-лесно създаване на гъвкави и поддържаеми приложения. Интеграция с .NET Framework: C# е част от .NET Framework, което позволява лесна интеграция с други технологии и библиотеки, с което се осигурява широка функционалност и гъвкавост на разработваните приложения.

3.2 WPF (Windows Presentation Foundation):



Фигура 2

Мощна платформа за разработване на десктоп приложения: WPF предоставя широк спектър от инструменти и възможности за създаване на модерни и атрактивни десктоп приложения за Windows операционни системи. Декларативно програмиране с XAML: WPF използва XAML (eXtensible Application Markup Language) за декларативно дефиниране на потребителски интерфейси, което улеснява разработката и поддръжката на приложенията. Гъвкавост и адаптивност: WPF предоставя гъвкави възможности за стилизиране и адаптиране на потребителския интерфейс, позволявайки създаването на различни дизайни и изживявания за потребителите.

3.3 MySQL:



Фигура 3

Мощна и надеждна релационна база данни: MySQL е една от най-популярните релационни бази данни, която предлага широки възможности за управление на данните и висока надеждност. Скалируемост и производителност: MySQL предоставя висока производителност и скалируемост, което я прави подходяща за различни видове приложения, от малки уебсайтове до големи корпоративни системи. Безплатна и отворена за обществеността: MySQL е безплатна за използване и разпространение, като също така е отворена за обществеността, което я прави предпочитан избор за много разработчици и организации.

3.4 GitHub:



Фигура 4

Уеб-базирана платформа за управление на кода: GitHub предоставя веб-базиран интерфейс за управление на кода, който позволява на разработчиците да сътрудничат, управляват и споделят своите проекти. Версионизиране и контрол на изходния код: GitHub осигурява инструменти за контрол на версиите, които позволяват на разработчиците да проследяват и управляват промените в своите проекти, като също така осигуряват възможност за създаване на клонове и сливане на промени. Интеграция с други инструменти: GitHub предлага интеграция с различни инструменти и услуги, като например CI/CD системи, уебсайти за документация и други, което улеснява интегрирането и автоматизацията на разработката.

4.Проектиране и реализация на приложението

4.1 Описание на приложението

Приложението е създадено да се използва глобално, и затова интерфейса е на английски език. Състои се от 4 основни менюта за потребители и администратори отделно:

- Login Window
- Main Page
- Select Seats
- Select Tickets
- Manage Halls – за администратори
- Schedule Panel – за администратори

Всяко меню извършва дадена цел на приложението.

Менюто Login Window съдържа един прозорец с елементи за регистрация и влизане в акаунта в софтуера в зависимост от избора на потребителя елементите за регистрация и влизане се скиват и показват чрез панел който се плъзга по самия прозорец.

В началната страница има шаблон за зареждане на филмите които са на прожекция добавени от администратора към базата данни има бутони с часове за прожекции до всеки филм от където потребителите могат да закупят билет

В менюто Select Tickets има данни за видовете билети които могат да изберат потребителите като студентски, детски, за възрастни над 60 години и нормален. Срещу всеки вид има брой на билети който може да се увеличава с + и – бутони

В менюто Select Seats има схема на залата с динамично добавени седалки в зависимост от капацитета на залата. Потребителя може да

избира толкова места колкото билети е избрал в предишния прозорец. При натискане на бутон „Pay“ се добавя в базата данни билет с всичката нужна информация за прожекцията, филма и потребителя.

В менюто Manage Halls администратора може да добавя зали в базата данни като въвежда име, тип и капацитет на залата спрямо този капацитет автоматично се създава схемата за седалките на определена зала.

В менюто Manage Schedule администратора може да добавя прожекции за определен филм, въвеждайки дата, час и зала на прожекцията чрез тези прожекции в Главната страница на потребителите се показват филмите които имат прожекция на определената дата.

4.2 Описание на кода по страници

4.2.1 Идентични елементи по страниците

Всеки прозорец има едни и същи компоненти като Header и Slider Panel.Ще предоставя кода за Header и Slider Panel по-долу за да предотвратя повторения при обяснение за всяка страница

```
<!--Header Elements -->
<Grid Background="Black" x:Name="HeaderElements" Height="90"
VerticalAlignment="Top" Margin="0,2,0,0" Grid.RowSpan="2">
    <Button HorizontalAlignment="Left" Panel.ZIndex="0" Height="20"
x:Name="CloseButton" Margin="823 ,17,0,0"
        VerticalAlignment="Top" Width="21" BorderBrush="{x:Null}"
Foreground="{x:Null}" Click="CloseButton_Click">
        <Button.Background>
            <ImageBrush ImageSource="/Resources/Images/bigger red
button.png" Stretch="UniformToFill"/>
        </Button.Background>
    </Button>
    <Button HorizontalAlignment="Left" x:Name="MaximizeButton"
Panel.ZIndex="0" Height="20" Margin="798,17,0,0"
        VerticalAlignment="Top" Width="21" BorderBrush="{x:Null}"
Foreground="{x:Null}" Click="MaximizeButton_Click" >
        <Button.Background>
            <ImageBrush ImageSource="/Resources/Images/bigger yellow
button.png" Stretch="UniformToFill"/>
        </Button.Background>
    </Button>
    <Button HorizontalAlignment="Left" Height="20" x:Name="MinimizeButton"
Panel.ZIndex="0" Margin="773,17,0,0"
        VerticalAlignment="Top" Width="21" BorderBrush="{x:Null}"
Foreground="{x:Null}" Click="MinimizeButton_Click">
        <Button.Background>
            <ImageBrush ImageSource="/Resources/Images/bigger green
button.png" Stretch="UniformToFill"/>
        </Button.Background>
    </Button>
    <Button Content="Button" HorizontalAlignment="Left" Height="45"
        VerticalAlignment="Center" Width="45" BorderBrush="{x:Null}"
Foreground="{x:Null}" x:Name="hamburgerButton"
        Click="hamburgerButton_Click_1">
```

```

        <Button.Background>
            <ImageBrush ImageSource="Resources/Images/hamburger button.png"
Stretch="UniformToFill"/>
        </Button.Background>
    </Button>
    <Ellipse HorizontalAlignment="Left" Height="90" Margin="366,0,0,0"
        VerticalAlignment="Center" Width="90" Fill="White">
        <Ellipse.OpacityMask>
            <ImageBrush ImageSource="/Resources/Images/F logo.png"/>
        </Ellipse.OpacityMask>
    </Ellipse>
</Grid>
<!--Slider-->

<Grid Name="SlideMenu" Width="200" Height="600" Margin="-201,2,851,-2"
Grid.RowSpan="2" Background="White" Panel.ZIndex="1" >

    <Grid.RowDefinitions>
        <RowDefinition Height="90"/>
        <RowDefinition Height="90"/>
        <RowDefinition Height="90"/>
    </Grid.RowDefinitions>

    <Button Grid.Row="0" Margin="7,0,0,23" HorizontalAlignment="Left"
VerticalAlignment="Center" Height="57"
        Width="57" Foreground="{x:Null}" BorderBrush="{x:Null}" >
        <Button.Background>
            <ImageBrush ImageSource="/Resources/Images/pp.png"
Stretch="UniformToFill"/>
        </Button.Background>
    </Button>
    <TextBlock VerticalAlignment="Center" HorizontalAlignment="Left"
Height="20" TextWrapping="Wrap" Text=""
        Width="89" x:Name="ProfileName" FontWeight="Bold"
FontSize="14" Margin="70,0,0,0"/>
    <Button Name="LogOutButton" Click="LogOutButton_Click" Margin="0,0,7,0"
HorizontalAlignment="Right" Height="26"
        VerticalAlignment="Center" Width="26" BorderBrush="{x:Null}">
        <Button.Background>
            <ImageBrush ImageSource="/Resources/Images/log out.png"/>
        </Button.Background>
    </Button>

```



```
<Border BorderThickness="0,0,0,2" BorderBrush="DimGray"/>

<Button Grid.Row="1" Background="White" Name="ManageHallsButton"
Foreground="Black" Content="Manage Halls"
        HorizontalContentAlignment="Left"                Padding="8"
BorderThickness="0,0,0,2" BorderBrush="DimGray" FontWeight="Heavy"

</Grid>
```

4.2.2 Описание на Main Page

MainPage.xaml и MainPage.xaml.cs съдържат код, който управлява главната страница на приложението. Този код дефинира визуални елементи като бутони, текстови блокове и списъци в XAML файл и логиката за техните действия в C# файл.

Файлът MainPage.xaml определя визуалния интерфейс на главната страница, включвайки различни елементи и стилове за тях. Тук се задават външни атрибути като разположение, размери, цветове и др.

MainPage.xaml.cs съдържа код за обработка на събития и логика за взаимодействие с потребителския интерфейс. Тук се изпълняват операции като извличане на данни от уеб сървъри и база данни, добавяне на нови филми и планиране на прожекции. Този код също така управлява навигацията между различните страници в приложението и обработва действията на потребителите като кликания на бутони и др. Общо взето, тези файлове представляват сърцето на главната страница на приложението, като предоставят възможност за визуализация на данни и взаимодействие с потребителите.

```

namespace Flickett
{
    public partial class MainPage : Window
    {
        private const string ApiKey = "186d8bc2a505456d116a65559e5d0788";
        private string userRole;

        public MainPage(string role = "") ...

        public MainPage(string role, string username) ...

        private async void LoadMovies()...

        public class MovieViewModel...

        private async Task<MovieViewModel> GetMovieDetails(HttpClient client, dynamic movieData)...

        private async Task AddMovieToDatabase(MovieViewModel movie)...
        private async void AddMovieButton_Click(object sender, RoutedEventArgs e)...
        private void MovieSchedule_Click(object sender, RoutedEventArgs e)...
        private void LoadMoviesFromDb()...
        private void ScheduleTimeButtons_Click(object sender, RoutedEventArgs e)...
        public static T FindParent<T>(DependencyObject child) ...

        private void MoveSliderMenuToRight()...
        private void MoveSliderMenuToLeft()...
        private void MoveSliderMenuToLeft()...

        private void hamburgerButton_Click_1(object sender, RoutedEventArgs e)...
        private void Window_MouseDown(object sender, MouseButtonEventArgs e)...
        private bool IsMouseOverUIElement(UIElement element, Point mousePosition)...
        private void MinimizeButton_Click(object sender, RoutedEventArgs e)...
        private void CloseButton_Click(object sender, RoutedEventArgs e)...
        private void MaximizeButton_Click(object sender, RoutedEventArgs e)...
        private void MoviePoster_Click(object sender, RoutedEventArgs e)...
        private void LogOutButton_Click(object sender, RoutedEventArgs e)...
    }
}

```

```

<!--Content Template-->
    <Grid Grid.Row="1" Width="850" Height="510" VerticalAlignment="Center"
    Background="Black" >

        <TextBlock Text="Enter Hall Details:" Foreground="White" Width="249"
    Height="64" HorizontalAlignment="Left" VerticalAlignment="Top" Padding="15"
    FontSize="25" Margin="42,9,0,0"/>

        <TextBlock x:Name="HallNamePlaceholder" FontSize="16"
    FontWeight="Light" Foreground="White" VerticalAlignment="Top"
    Margin="37,100,3,0" Panel.ZIndex="-1" Text="Hall Name"/>
        <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="200"
    Height="30"
    FontSize="20" HorizontalAlignment="Left"
    x:Name="HallNameBox" TextChanged="HallNameBox_TextChanged"
    VerticalContentAlignment="Center" BorderBrush="White"
    Margin="33,94,0,386" Foreground="White" Background="Transparent"
    SelectionBrush="{x:Null}" SelectionTextBrush="{x:Null}"/>

        <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="120"
    Height="30" TextChanged="CapacityBox_TextChanged"
    FontSize="20" HorizontalAlignment="Left" Name="CapacityBox"
    VerticalContentAlignment="Center" BorderBrush="White" Margin="36,148,0,332"
    Foreground="White" Background="Transparent" SelectionBrush="{x:Null}"
    SelectionTextBrush="{x:Null}"/>
        <TextBlock x:Name="CapacityPlaceholder" FontSize="16" FontWeight="Thin"
    Foreground="White" VerticalAlignment="Top"
    Margin="36,151,4,0" Panel.ZIndex="-1" Text="Enter Capacity"
    />

        <TextBlock x:Name="CategoryPlaceholder" FontSize="16" FontWeight="Thin"
    Foreground="White" VerticalAlignment="Top"
    Margin="38,213,2,0" Panel.ZIndex="-1" Text="Enter
    Category"/>
        <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="200"
    Height="30"
    FontSize="20" HorizontalAlignment="Left" x:Name="CategoryBox"
    TextChanged="CategoryBox_TextChanged"

```

```
                VerticalContentAlignment="Center"                BorderBrush="White"
Margin="36,210,0,270"        Foreground="White"        Background="Transparent"
SelectionBrush="{x:Null}" SelectionTextBrush="{x:Null}"/>
        <Button Width="100" Margin="37,274,713,205" Background="Transparent"
BorderBrush="White" Content="Add Hall" FontSize="12" Foreground="White"
Click="Button_Click"/>

</Grid>
```

4.2.3 Описание на Login Window

4.2.3.1 Front end

IgnWindow е прозорец за вход и регистрация в приложение, който включва валидация на потребителските данни и взаимодействие с база данни. Той предлага функции за вход на потребители, като проверява валидността на потребителско име и парола, както и за регистрация на нови потребители, като осигурява сигурно съхранение на паролите чрез хеширане преди запис в базата данни. Кодът включва логика за анимация при превключване между екраните за вход и регистрация, както и обработка на грешки при взаимодействие с базата данни.

4.2.3.2 Backend

Файлът представлява XAML код за дизайна на прозореца за вход и регистрация (IgnWindow). Прозорецът е стилизиран без рамка (`WindowStyle="None"`) с прозрачно фоново изображение (`AllowsTransparency="True"`) и стартира по центъра на екрана (`WindowStartupLocation="CenterScreen"`). Фона е черен (`Background="Black"`), а размерите са зададени на 500 пиксела височина и 800 пиксела ширина. В ресурсите на прозореца се дефинират стилове за бутоните и текстовите полета, като се използват различни цветове и изображения за фон и бордери. Също така, се задават стилове за `TextBox` и `PasswordBox` за въвеждане на потребителско име и парола. В главния `Grid` се разполагат различни елементи като бутони за минимизиране, максимизиране и затваряне на прозореца, текстови полета за въвеждане на потребителско име и парола, бутони за влизане и регистрация, както и текстови блокове за

показване на съобщения за грешки. В прозореца са включени и бутони за превключване между екраните за вход и регистрация.

4.2.4 Описание на Select Tickets

4.2.4.1 Backend

"SelectTickets" е WPF прозорец, който представлява интерфейс за избор на билети за конкретно филмово представление. В този прозорец потребителят може да избере брой билети за различни видове зрители (регулярни, деца, възрастни, студенти) чрез увеличаване и намаляване на съответните броячи. Има

```
public partial class SelectTickets : Window
{
    string connectionString = "server=localhost;uid=root;pwd=Antonow7;database=cinemadb;SslMk
    private MovieViewModel movie;
    private string screeningTime;
    private int totalQuantity = 0;

    public SelectTickets(MovieViewModel movie, string screeningTime)...

    private (int, int) GetScheduleDetails(string movieId, string screeningTime)...

    private string GetHallNameForHall(int hallId)...

    private (decimal, string) GetBasePriceAndHallNameForHall(int hallId)...

    int regularQuantity = 0;
    int kidsQuantity = 0;
    int adultQuantity = 0;
    int studentQuantity = 0;

    private void RegularTicketQuantityBoxPlus_Click(object sender, RoutedEventArgs e)...
    private void RegularTicketQuantityBoxMinus_Click(object sender, RoutedEventArgs e)...
    private void KidsTicketQuantityBoxPlus_Click(object sender, RoutedEventArgs e)...
    private void KidsTicketQuantityBoxMinus_Click(object sender, RoutedEventArgs e)...
    private void AdultTicketQuantityBoxPlus_Click(object sender, RoutedEventArgs e)...
    private void AdultTicketQuantityBoxMinus_Click(object sender, RoutedEventArgs e)...
    private void StudentTicketQuantityBoxPlus_Click(object sender, RoutedEventArgs e)...
    private void StudentTicketQuantityBoxMinus_Click(object sender, RoutedEventArgs e)...
    private void UpdateTotalQuantity()...
```

функционалност за преминаване към следващия етап - избор на места
- и за връщане към предишния етап - началната страница.

4.2.4.2 Front end

XAML файла дефинира интерфейса за избор на билети за филм. Включва елементи като текстови блокове за показване на информация за филма и цените на билетите, бутони за управление на броя на билетите и за навигация между етапите на приложението, както и рамки за разделяне на различните секции на интерфейса.

```
<!--Content Template-->
  <Grid Grid.Row="1" Width="850" Height="510" VerticalAlignment="Center"
  Background="Black" >

    <Grid>

      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="250"/>
        <ColumnDefinition/>
      </Grid.ColumnDefinitions>

      <Grid Grid.Column="0">
        <Grid.RowDefinitions>
          <RowDefinition Height="60"/>
          <RowDefinition Height="auto"/>
          <RowDefinition Height="auto"/>
        </Grid.RowDefinitions>

        <WrapPanel Grid.Row="0" >
          <TextBlock TextWrapping="Wrap" Text="{Binding Title}"
          Foreground="White" FontSize="20" Name="MovieNameTicket"/>
        </WrapPanel>
      </Grid>
    </Grid>
  </Grid>
```



```

        <TextBlock Grid.Row="1" Foreground="White" FontSize="15"
Name="ScheduleDateTime">
            <Run Text="{Binding ScreeningTime}"/>
        <Run Text=" | "/>
            <Run Text="{Binding ScreeningDate}"/>
        </TextBlock>

        <TextBlock Text="{Binding HallName}" Grid.Row="2"
Foreground="White" FontSize="15" Name="HallNameTicket"/>
    </Grid>

    <Grid Grid.Column="1 ">

        <Grid.RowDefinitions>
            <RowDefinition Height="70"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="50"/>
            <RowDefinition/>

        </Grid.RowDefinitions>

        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="230"/>
            <ColumnDefinition/>
            <ColumnDefinition/>

        </Grid.ColumnDefinitions>

        <TextBlock Grid.Row="0" Foreground="White" Text="Choose Ticket"
FontSize="35" FontWeight="DemiBold"/>
        <TextBlock Grid.Row="1" Foreground="White" Text="Type"
FontSize="20" FontWeight="Regular" VerticalAlignment="Bottom" Padding="15"/>
        <TextBlock Grid.Row="1" Grid.Column="1" Foreground="White"
Text="Price" FontSize="20" FontWeight="Regular" VerticalAlignment="Bottom"
Padding="15" HorizontalAlignment="Center"/>

```

```

        <TextBlock Grid.Row="1" Grid.Column="2" Foreground="White"
Text="Quantity"           FontSize="20"           FontWeight="Regular"
VerticalAlignment="Bottom" Padding="15" HorizontalAlignment="Center"/>

        <Border           Grid.Row="1"           BorderThickness="0,0,0,1"
BorderBrush="Orange"/>
        <Border Grid.Row="1" Grid.Column="1" BorderThickness="0,0,0,1"
BorderBrush="Orange"/>
        <Border Grid.Row="1" Grid.Column="2" BorderThickness="0,0,0,1"
BorderBrush="Orange"/>

        <TextBlock      Grid.Row="2" Foreground="White" Text="Regular
Ticket" FontSize="20"      FontWeight="Regular" VerticalAlignment="Bottom"
Padding="15"/>
        <Border           Grid.Row="2"           BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="2" Grid.Column="1" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="2" Grid.Column="2" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>

        <TextBlock Grid.Row="3" Foreground="White" Text="Kids Ticket"
FontSize="20" FontWeight="Regular" VerticalAlignment="Bottom" Padding="15"/>

        <Border           Grid.Row="3"           BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="3" Grid.Column="1" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="3" Grid.Column="2" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>

        <TextBlock      Grid.Row="4" Foreground="White" Text="Adult 60+
Ticket" FontSize="20"      FontWeight="Regular" VerticalAlignment="Bottom"
Padding="15"/>

        <Border           Grid.Row="4"           BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="4" Grid.Column="1" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>
        <Border Grid.Row="4" Grid.Column="2" BorderThickness="0,0,0,1"
BorderBrush="DimGray"/>

```

```

        <TextBlock      Grid.Row="5" Foreground="White" Text="Student
Ticket" FontSize="20"      FontWeight="Regular" VerticalAlignment="Bottom"
Padding="15"/>
        <Border      Grid.Row="5"      BorderThickness="0,0,0,1"
BorderBrush="Orange"/>
        <Border Grid.Row="5" Grid.Column="1" BorderThickness="0,0,0,1"
BorderBrush="Orange"/>
        <Border Grid.Row="5" Grid.Column="2" BorderThickness="0,0,0,1"
BorderBrush="Orange"/>

        <TextBlock      x:Name="RegularTicketPriceBox"      Grid.Row="2"
Grid.Column="1" Foreground="White" Text="{Binding basePrice}" FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>
        <TextBlock      x:Name="KidsTicketPriceBox"      Grid.Row="3"
Grid.Column="1" Foreground="White" Text="{Binding basePrice}" FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>
        <TextBlock      x:Name="AdultTicketPriceBox"      Grid.Row="4"
Grid.Column="1" Foreground="White" Text="{Binding basePrice}" FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>
        <TextBlock      x:Name="StudentTicketPriceBox"      Grid.Row="5"
Grid.Column="1" Foreground="White" Text="{Binding basePrice}" FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>

        <Grid Grid.Column="2" Grid.Row="2">
            <Grid.ColumnDefinitions>
                <ColumnDefinition/>
                <ColumnDefinition/>
                <ColumnDefinition/>
            </Grid.ColumnDefinitions>

            <TextBlock      x:Name="RegularTicketQuantityBox"
Grid.Column="1"      Foreground="White"      Text="0"      FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>

```

```

        <Button Click="RegularTicketQuantityBoxPlus_Click"
x:Name="RegularTicketQuantityBoxPlus" FontSize="20" Grid.Column="0"
Content="-" HorizontalAlignment="Right" Background="Transparent"
Foreground="White" BorderThickness="0" Height="26" Width="19"/>
        <Button Click="RegularTicketQuantityBoxMinus_Click"
x:Name="RegularTicketQuantityBoxMinus" FontSize="19" Grid.Column="2"
Margin="0,0,10,0" Content="+" HorizontalAlignment="Left"
Background="Transparent" Foreground="White" BorderThickness="0" Height="26"
Width="19"/>

</Grid>

<Grid Grid.Column="2" Grid.Row="3">
    <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>

    <TextBlock x:Name="KidsTicketQuantityBox" Grid.Column="1"
Foreground="White" Text="0" FontSize="20" FontWeight="Regular"
VerticalAlignment="Bottom" HorizontalAlignment="Center" Padding="15"/>
    <Button Click="KidsTicketQuantityBoxPlus_Click"
x:Name="KidsTicketQuantityBoxPlus" FontSize="20" Grid.Column="0" Content=""
HorizontalAlignment="Right" Background="Transparent" Foreground="White"
BorderThickness="0" Height="26" Width="19"/>
    <Button Click="KidsTicketQuantityBoxMinus_Click"
x:Name="KidsTicketQuantityBoxMinus" FontSize="19" Grid.Column="2"
Margin="0,0,10,0" Content="+" HorizontalAlignment="Left"
Background="Transparent" Foreground="White" BorderThickness="0" Height="26"
Width="19"/>

</Grid>

<Grid Grid.Column="2" Grid.Row="4">
    <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
        <ColumnDefinition/>

```

```

</Grid.ColumnDefinitions>

        <TextBlock x:Name="AdultTicketQuantityBox" Grid.Column="1"
Foreground="White" Text="0" FontSize="20" FontWeight="Regular"
VerticalAlignment="Bottom" HorizontalAlignment="Center" Padding="15"/>
        <Button Click="AdultTicketQuantityBoxPlus_Click"
x:Name="AdultTicketQuantityBoxPlus" FontSize="20" Grid.Column="0" Content=""
HorizontalAlignment="Right" Background="Transparent" Foreground="White"
BorderThickness="0" Height="26" Width="19"/>
        <Button Click="AdultTicketQuantityBoxMinus_Click"
x:Name="AdultTicketQuantityBoxMinus" FontSize="19" Grid.Column="2"
Margin="0,0,10,0" Content="+" HorizontalAlignment="Left"
Background="Transparent" Foreground="White" BorderThickness="0" Height="26"
Width="19"/>

```

```

</Grid>

```

```

<Grid Grid.Column="2" Grid.Row="5">
    <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>

        <TextBlock x:Name="StudentTicketQuantityBox"
Grid.Column="1" Foreground="White" Text="0" FontSize="20"
FontWeight="Regular" VerticalAlignment="Bottom" HorizontalAlignment="Center"
Padding="15"/>
        <Button Click="StudentTicketQuantityBoxPlus_Click"
x:Name="StudentTicketQuantityBoxPlus" FontSize="20" Grid.Column="0"
Content="-" HorizontalAlignment="Right" Background="Transparent"
Foreground="White" BorderThickness="0" Height="26" Width="19"/>
        <Button Click="StudentTicketQuantityBoxMinus_Click"
x:Name="StudentTicketQuantityBoxMinus" FontSize="19" Grid.Column="2"
Margin="0,0,10,0" Content="+" HorizontalAlignment="Left"
Background="Transparent" Foreground="White" BorderThickness="0" Height="26"
Width="19"/>

```

```
</Grid>
```

```
        <Button      Click="SelectTicketBackButton_Click"
x:Name="SelectTicketBackButton"  Grid.Column="0"  Grid.Row="6"  Height="40"
Width="100"  Background="Black"  Foreground="White"  Content="Back"
FontSize="20"      BorderBrush="White"      VerticalAlignment="Top"
HorizontalAlignment="Left" Margin="0,10,0,0" BorderThickness="2"/>
        <Button      Click="SelectTicketNextButton_Click"
x:Name="SelectTicketNextButton"  Style="{DynamicResource  NextButton}"
Grid.Column="2 " Grid.Row="6" Height="40" Width="100" Background="Orange"
Foreground="black" Content="Next" FontSize="20" BorderBrush="White"
VerticalAlignment="Top" HorizontalAlignment="Right" Margin="0,10,10,0"
BorderThickness="0"/>
```

```
</Grid>
```

4.2.5 Описание на Select Seats

4.2.5.1 Front end

Този XAML файл определя потребителски интерфейс за някаква апликация, вероятно свързана с киното, тъй като виждаме седалки и екран за филми. Главен контейнер е Grid, с два реда. В първия ред има черен контейнер, вероятно представляващ кино екран. Във втория ред има вложен Grid с име "SeatGrid", който съдържа седалки и бутон за плащане. В SeatGrid има вложени Grid-ове. Първият такъв се явява схемата на залата, където са разположени седалките и екранът. След това са разположени текстови блокове с номера на седалките от горния и долния ред. Вътрешният Grid съдържа бутон за плащане ("PayButton"), който изглежда се намира в последната колона.

```
<!--Content Template-->
<Grid Grid.Row="1" Width="850" Height="510" VerticalAlignment="Center"
Background="Black" >

</Grid>

<Grid Grid.Row="2" x:Name="SeatGrid">

    <!--Movie Screen-->

    <Grid>

        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="60"/>
        </Grid.RowDefinitions>
    </Grid>
</Grid>
```

```

        </Grid.RowDefinitions>
        <Image Source="/screen 2.png" Stretch="Fill"
HorizontalAlignment="Center" RenderTransformOrigin="0.5,0.5"
Margin="3,9,0,11" Grid.Row="1" />

```

```

<!--Seats-->

```

```

<Grid Grid.Row="0" x:Name="SeatsLayout">

```

```

    <Grid.RowDefinitions>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

        <RowDefinition/>

```

```

    </Grid.RowDefinitions>

```

```

    <Grid.ColumnDefinitions>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

        <ColumnDefinition/>

```

```

    </Grid.ColumnDefinitions>

```



```

        <TextBlock Grid.Row="2"      Grid.Column="1" Foreground="White"
Text="9"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="3"      Grid.Column="1" Foreground="White"
Text="8"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="4"      Grid.Column="1" Foreground="White"
Text="7"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="5"      Grid.Column="1" Foreground="White"
Text="6"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="6"      Grid.Column="1" Foreground="White"
Text="5"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="7"      Grid.Column="1" Foreground="White"
Text="4"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="8"      Grid.Column="1" Foreground="White"
Text="3"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="9"      Grid.Column="1" Foreground="White"
Text="2"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="10"     Grid.Column="1" Foreground="White"
Text="1"          FontSize="25"      VerticalAlignment="Center"
HorizontalAlignment="Center"/>


        <TextBlock Grid.Row="2" Grid.Column="12" Foreground="White"
Text="9"          FontSize="25"          VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="3" Grid.Column="12" Foreground="White"
Text="8"          FontSize="25"          VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="4" Grid.Column="12" Foreground="White"
Text="7"          FontSize="25"          VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="5" Grid.Column="12" Foreground="White"
Text="6"          FontSize="25"          VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <TextBlock Grid.Row="6" Grid.Column="12" Foreground="White"

```

```

Text="5"                FontSize="25"                VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        <TextBlock Grid.Row="7" Grid.Column="12" Foreground="White"
Text="4"                FontSize="25"                VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        <TextBlock Grid.Row="8" Grid.Column="12" Foreground="White"
Text="3"                FontSize="25"                VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        <TextBlock Grid.Row="9" Grid.Column="12" Foreground="White"
Text="2"                FontSize="25"                VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        <TextBlock Grid.Row="10" Grid.Column="12" Foreground="White"
Text="1"                FontSize="25"                VerticalAlignment="Center"
HorizontalAlignment="Center"/>

        <Button Click="PayButton_Click" x:Name="PayButton"
Grid.Column="13" Grid.ColumnSpan="4" Background="Black" Foreground="White"
Content="Pay" FontSize="20" BorderThickness="0" Margin="0,5,10,0"/>

</Grid>

</Grid>

```

4.2.5.2 Backend

SelectSeats.cs е клас, който управлява взаимодействието между потребителския интерфейс (определен в XAML файл) и базата данни за закупуване на билети за кино. Съдържа конструктор за инициализация на прозореца, методи за връщане на детайли за програмата на филма и капацитета на залата, както и метод за добавяне на седалки към интерфейса. Включва и обработчици на събития за навигация и управление на прозореца, както и за закупуване на билети.

```

public partial class SelectSeats : Window
{

    string connectionString = "server=localhost;uid=root;pwd=Antonow7;database=cinemadb;SslMode=None;";

    private MovieViewModel movie;
    private string screeningTime;
    private int totalQuantity;
    private int screeningId;

    public SelectSeats(MovieViewModel movie, string screeningTime ,int totalQuantity) {...}

    private (int, int) GetScheduleDetails(string movieId, string screeningTime) {...}

    private int GetHallCapacity(string movieId, string screeningTime) {...}

    private void AddSeatsToGrid(int hallCapacity) {...}

    private int selectedSeatsCount = 0;

    private void SeatButton_Click(object sender, RoutedEventArgs e) {...}
}

```

4.2.6 Описание на Manage Halls

4.2.6.1 Front end

Файлът ManageHalls.xaml е част от потребителския интерфейс за управление на залите. В него се дефинират различни елементи, които позволяват на администраторите да въвеждат и редактират информация за залите. Включени са текстови полета за въвеждане на име, капацитет и категория на залата, както и бутон за добавяне на нова зала. Контролите са разположени във Grid контейнер с черен фон, който дава един стилизиран външен вид на интерфейса.

```
<!--Content Template-->
<Grid Grid.Row="1" Width="850" Height="510" VerticalAlignment="Center"
Background="Black" >

    <TextBlock Text="Enter Hall Details:" Foreground="White" Width="249"
Height="64" HorizontalAlignment="Left" VerticalAlignment="Top" Padding="15"
FontSize="25" Margin="42,9,0,0"/>

    <TextBlock x:Name="HallNamePlaceholder" FontSize="16"
FontWeight="Light" Foreground="White" VerticalAlignment="Top"
Margin="37,100,3,0" Panel.ZIndex="-1" Text="Hall Name"/>
    <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="200"
Height="30"
FontSize="20" HorizontalAlignment="Left"
x:Name="HallNameBox" TextChanged="HallNameBox_TextChanged"
VerticalContentAlignment="Center" BorderBrush="White"
Margin="33,94,0,386" Foreground="White" Background="Transparent"
SelectionBrush="{x:Null}" SelectionTextBrush="{x:Null}"/>

    <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="120"
Height="30" TextChanged="CapacityBox_TextChanged"
FontSize="20" HorizontalAlignment="Left" Name="CapacityBox"
VerticalContentAlignment="Center" BorderBrush="White" Margin="36,148,0,332"
Foreground="White" Background="Transparent" SelectionBrush="{x:Null}"
SelectionTextBrush="{x:Null}"/>

    <TextBlock x:Name="CapacityPlaceholder" FontSize="16" FontWeight="Thin"
```

```

Foreground="White" VerticalAlignment="Top"
        Margin="36,151,4,0" Panel.ZIndex="-1" Text="Enter Capacity"
    />

    <TextBlock x:Name="CategoryPlaceholder" FontSize="16" FontWeight="Thin"
Foreground="White" VerticalAlignment="Top"
        Margin="38,213,2,0" Panel.ZIndex="-1" Text="Enter
Category"/>
    <TextBox CaretBrush="White" BorderThickness="0,0,0,2" Width="200"
Height="30"
        FontSize="20" HorizontalAlignment="Left" x:Name="CategoryBox"
TextChanged="CategoryBox_TextChanged"
        VerticalContentAlignment="Center" BorderBrush="White"
Margin="36,210,0,270" Foreground="White" Background="Transparent"
SelectionBrush="{x:Null}" SelectionTextBrush="{x:Null}"/>
    <Button Width="100" Margin="37,274,713,205" Background="Transparent"
BorderBrush="White" Content="Add Hall" FontSize="12" Foreground="White"
Click="Button_Click"/>

</Grid>

```

4.2.6.2 Backend

Файлът ManageHalls.cs съдържа логика за управление на залите в приложението. Той включва инициализация на компонентите в конструктора, методи за анимация при преместване на менюто, обработчици на събития за натискане на бутони и преместване на прозореца с мишката, валидация на входните данни за залата (име, капацитет, категория) и изпращане на заявка за вмъкване на данни в базата данни при добавяне на нова зала.

4.2.7 Описание на Schedule Panel

4.2.7.1 Front end

SchedulePanel е част от потребителския интерфейс и предоставя полета и контроли за добавяне на график за филми. Включва поле със снимка на постера на филма, текстови блокове за преглед, заглавие и жанрове и продължителност, както и комбо кутии за избор на дата, час и зала. Освен това, има бутон за добавяне на графика.

```
<!--Content Template-->
<Grid Grid.Row="1" Width="850" Height="510"
VerticalAlignment="Center" Background="Black" >
    <Rectangle x:Name="PosterBox" Height="300"Width="200"
Margin="30,0,0,150" HorizontalAlignment="Left" Stroke="White"
StrokeThickness="2">
        <Rectangle.Fill>
            <ImageBrush ImageSource="{Binding PosterUrl}"
Stretch="UniformToFill"/>
        </Rectangle.Fill>
    </Rectangle>
    <TextBlock x:Name="OverViewTextBox" Margin="257,131,345,175"
FontWeight="Bold" TextWrapping="Wrap" Foreground="White"/>
    <TextBlock x:Name="TitleTextBlock" Margin="257,40,18,398"
FontWeight="Bold" Foreground="White" FontSize="22"/>
    <TextBlock x:Name="GenresAndDurationTextBox" Width="200"
Height="40" Margin="30,333,611,127" FontWeight="DemiBold"
Foreground="White"/>

    <DatePicker x:Name="datePicker" Margin="529,128,121,351"
Width="200" BorderThickness="0" SelectedDate="{x:Static sys:DateTime.Today}"
DisplayDateStart="{x:Static sys:DateTime.Today}"/>
    <ComboBox x:Name="hoursComboBox" Width="50" Height="30"
Margin="531,188,269,282"/>
    <ComboBox x:Name="minutesComboBox" Width="50" Height="30"
Margin="598,188,202,282"/>
    <ComboBox x:Name="SelectHall" Height="30"
Margin="676,188,51,282"/>
    <TextBlock HorizontalAlignment="Left" Height="23"
```

```

Margin="533,166,0,0"      TextWrapping="Wrap"      FontSize="14"Text="Hour:"
Foreground="White" VerticalAlignment="Top" Width="45"/>
    <TextBlock      HorizontalAlignment="Left"      Height="21"
Margin="600,166,0,0"      TextWrapping="Wrap"      FontSize="14" Text="Min:"
Foreground="White" VerticalAlignment="Top" Width="45"/>
    <TextBlock      HorizontalAlignment="Left"      Height="21"
Margin="678,166,0,0" TextWrapping="Wrap" FontSize="14"      Text="Hall:"
Foreground="White" VerticalAlignment="Top" Width="46"/>
    <Button      Width="100"      Margin="710,448,40,31"
Background="Transparent" BorderBrush="White" Content="Add Schedule"
FontSize="12"      Foreground="White"      Name="AddScheduleButton"
Click="AddScheduleButton_Click"/></Grid>

```

4.2.7.2 Backend

SchedulePanel е прозорец, който представя формата за добавяне на график за филм. Инициализира се с данни за филма и показва информация за него като заглавие, преглед и жанрове с продължителност. Потребителят може да избере дата, час и зала, след което да добави графика. Включва и функционалности като преместване на менюто, минимизиране, максимизиране, затваряне на прозореца и изход от акаунта. Той също така може да пренасочи потребителя към управлението на зали.

```

public partial class SchedulePanel : Window
{
    private MovieViewModel movieData;

    public SchedulePanel(MovieViewModel movieData) {...}

    public TimeSpan GetSelectedTime() {...}

    private void ShowHallData() {...}

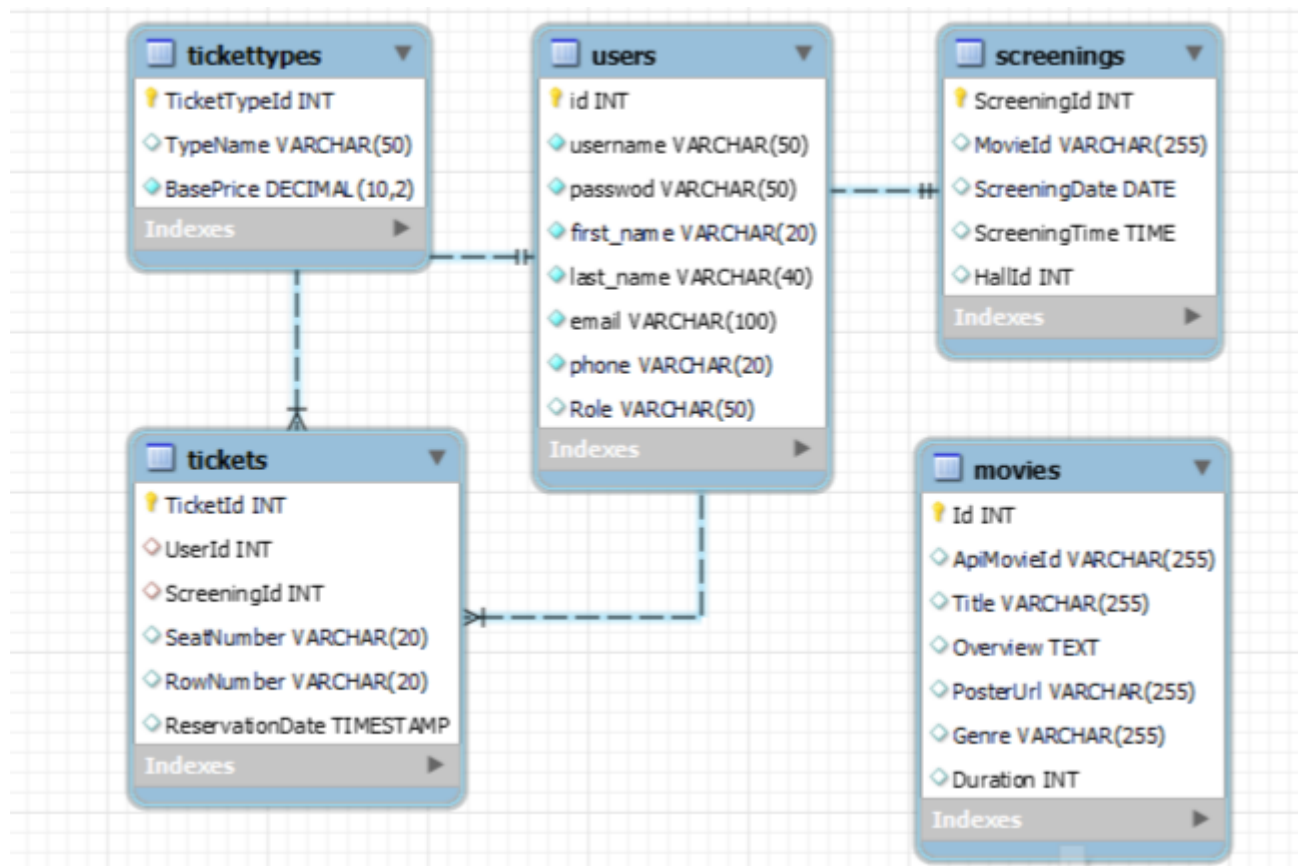
    private void AddScheduleButton_Click(object sender, RoutedEventArgs e) {...}

    private int GetHallId(string hallName) {...}
}

```

5.Бази данни

5.1 Схема



5.2 Връзки

5.2.1 Users и Tickets

Връзка: едно към много (един потребител може да има много билети).

5.2.2 Movies и Screenings

Връзка: едно към много (един филм може да има много прожекции).

5.2.3 Halls и Screenings

Връзка: едно към много (една зала може да има много прожекции).

5.2.4 Screenings и Tickets

Връзка: едно към много (една прожекция може да има много билети).

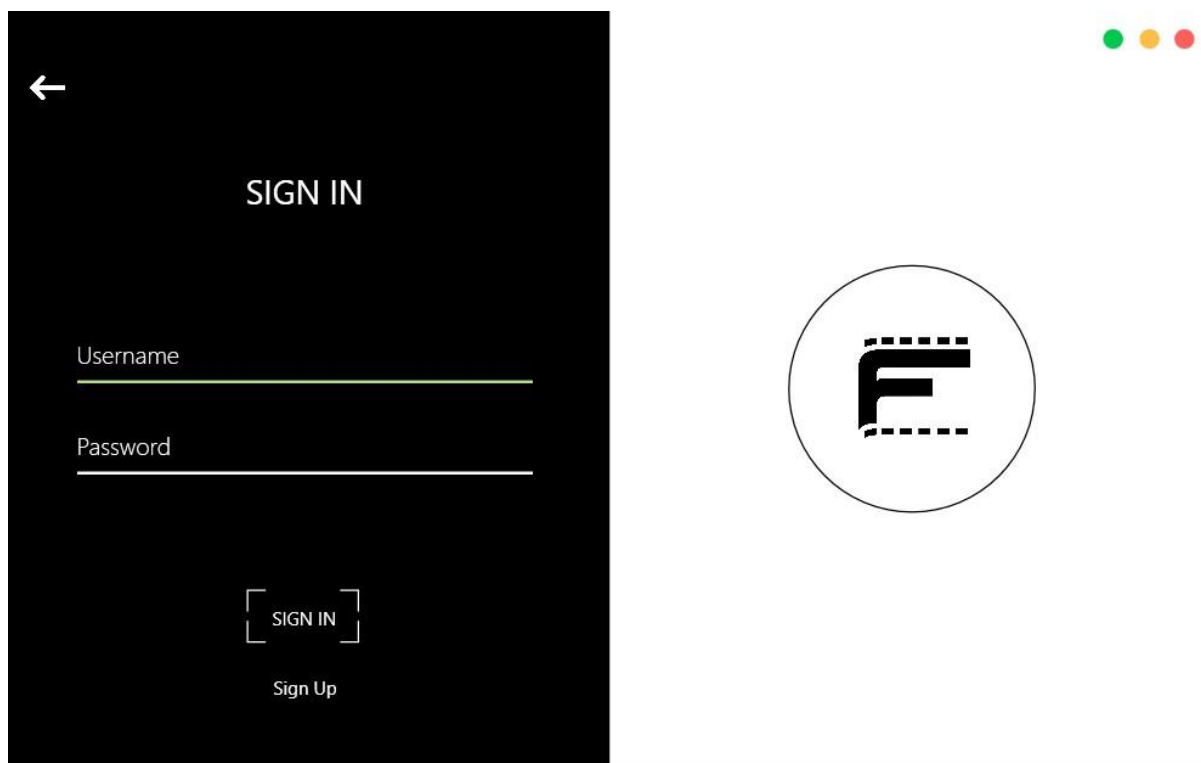
5.2.5 Halls и TicketTypes

Връзка: много към едно (много зали могат да имат един и същи тип билет).

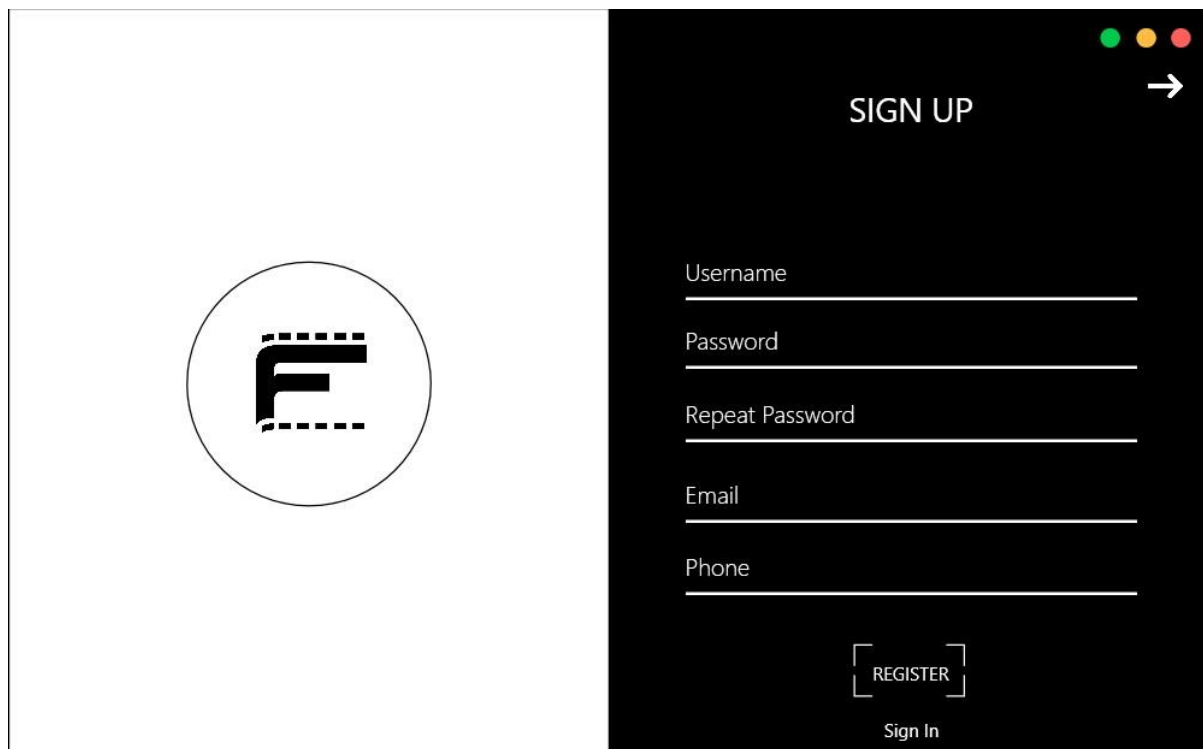
6. Потребителски интерфейс

6.1 Клиентско приложение

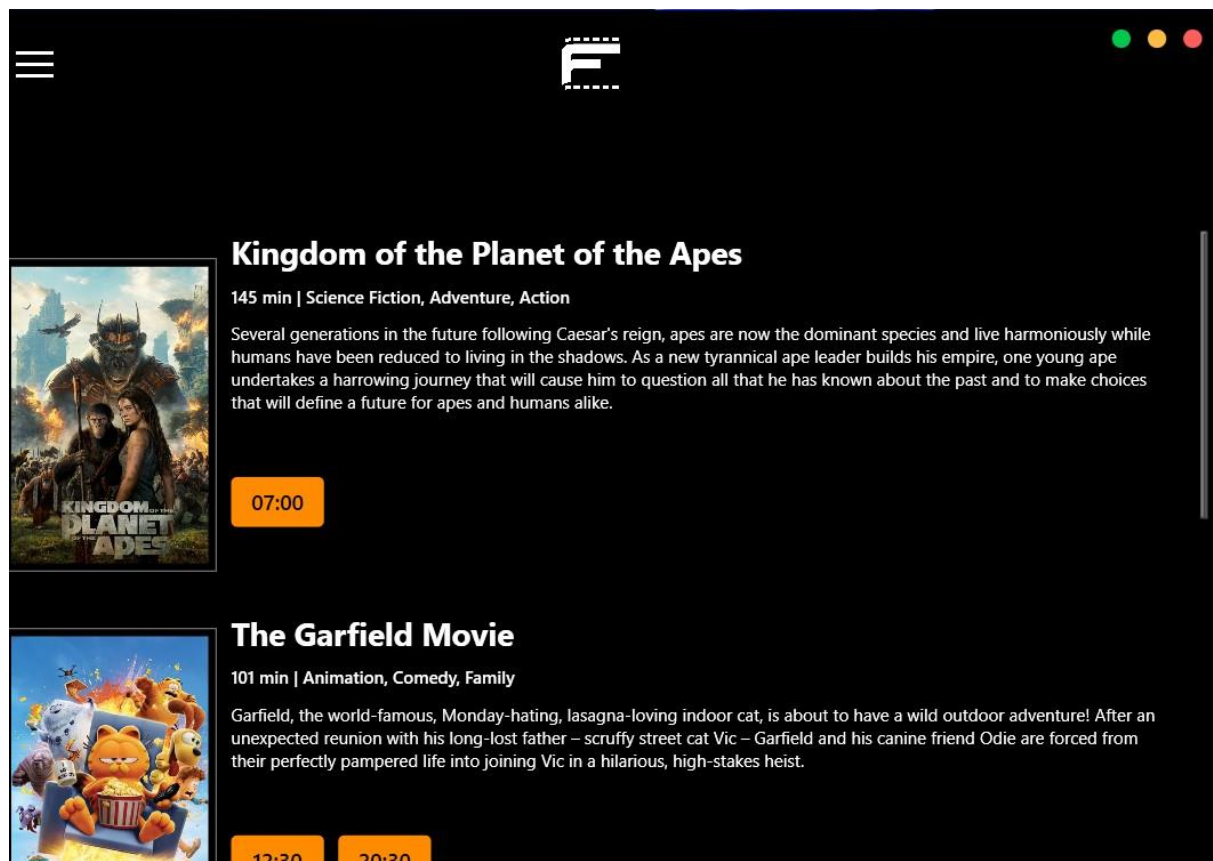
6.1.1 Login страница



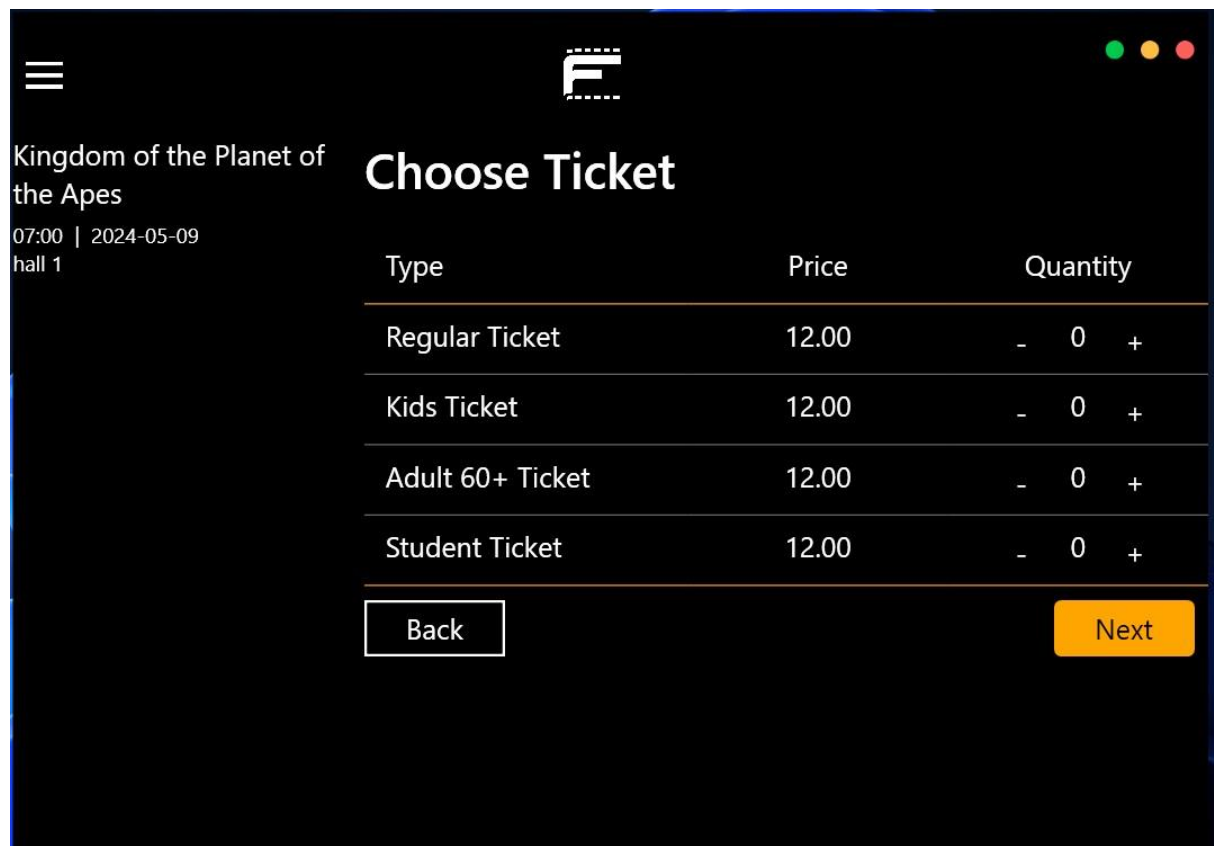
Фигура 6 (начална страница)



Фигура 7(начална страница)

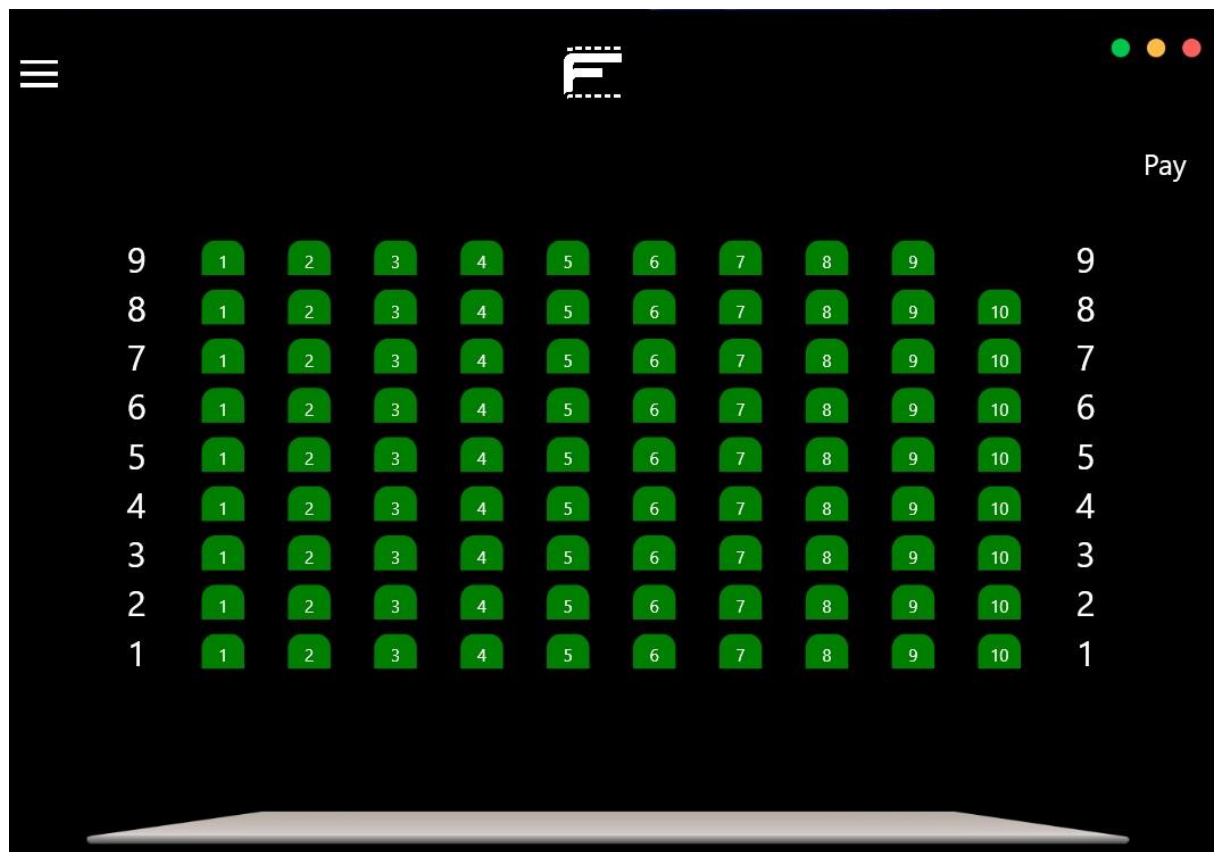


Фигура 8(главна страница)



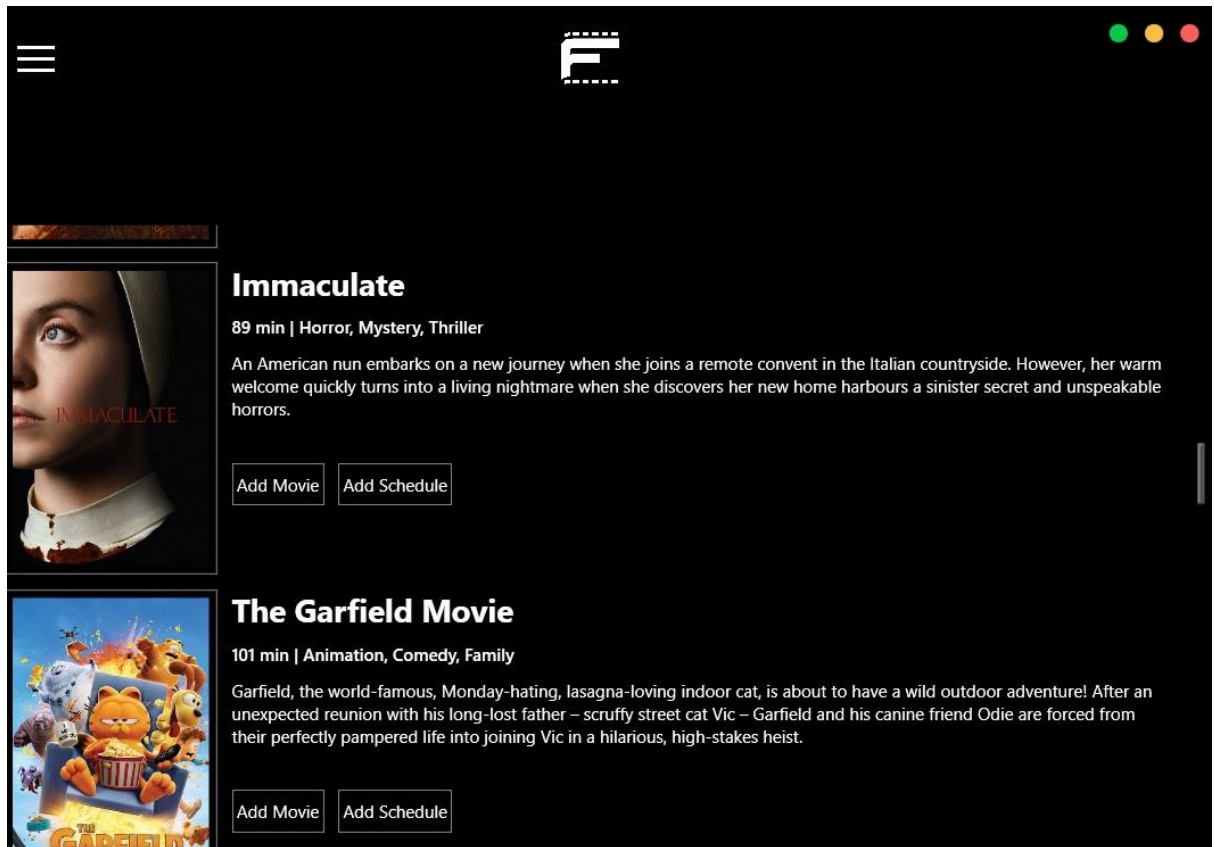
Фигура 9(избор на билети)

6.1.3 Избор на места

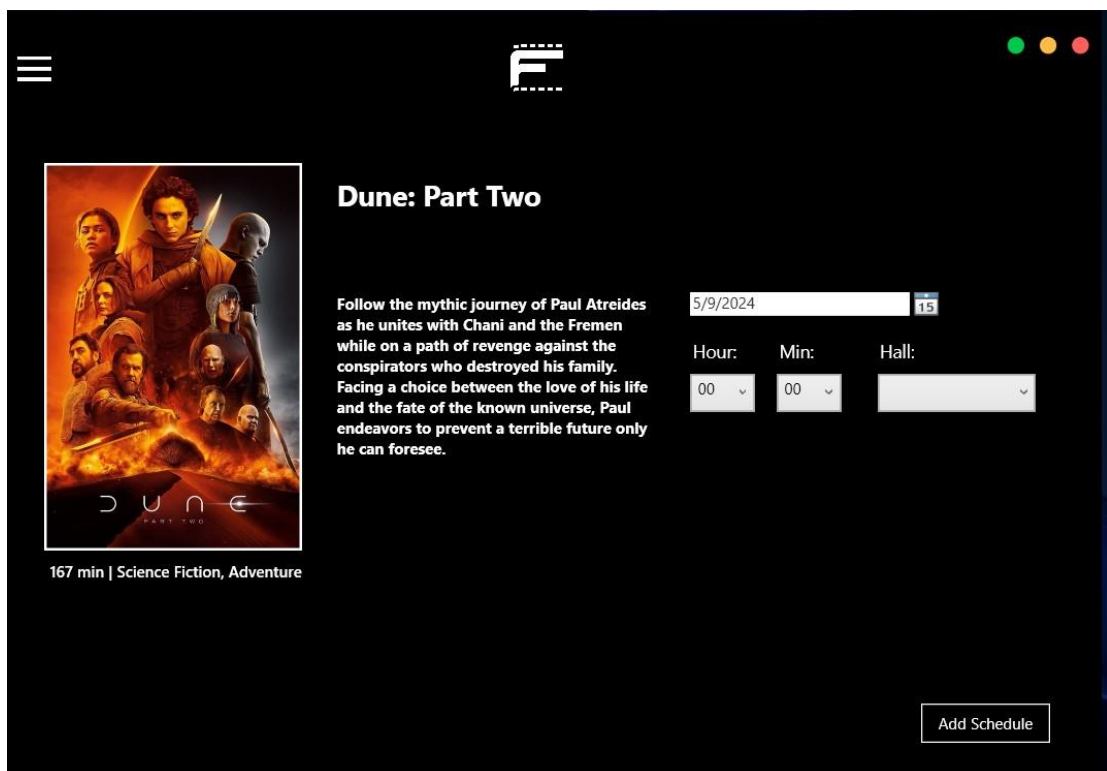


Фигура 10(избор на места)

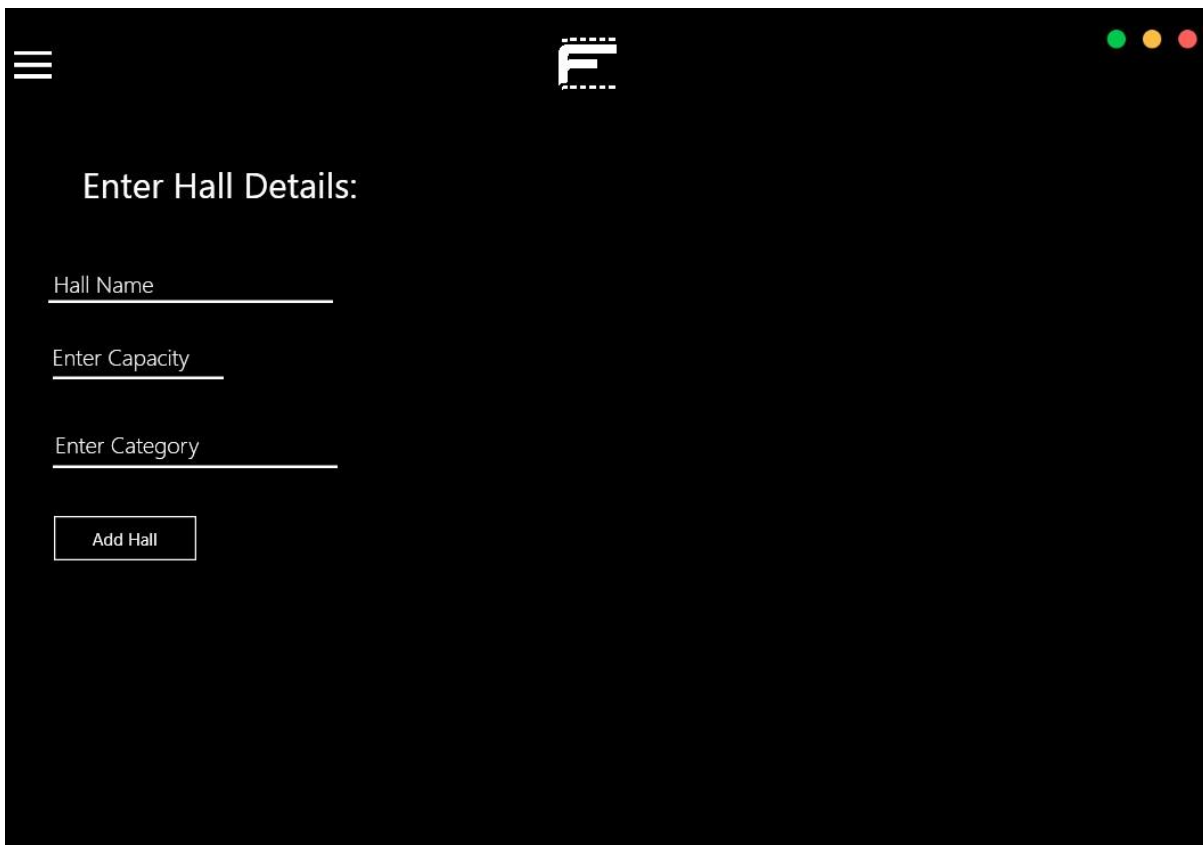
6.2 Администраторско приложение



Фигура 11(администраторска главна страница)



Фигура 12(добавяне на прожекция)



The screenshot shows a web application interface with a dark blue background. At the top left is a hamburger menu icon (three horizontal lines). At the top center is a logo consisting of a stylized 'F' with dashed lines. At the top right are three colored window control buttons (green, yellow, red). Below the header, the text 'Enter Hall Details:' is displayed in white. Underneath, there are three input fields, each with a white underline and a label above it: 'Hall Name', 'Enter Capacity', and 'Enter Category'. At the bottom left of the form area is a rectangular button with a white border and the text 'Add Hall' in white.

Enter Hall Details:

Hall Name

Enter Capacity

Enter Category

Add Hall

Фигура 13(добавяне на зала)

7. Бъдещо развитие на приложението

Бъдещото развитие на приложението за управление на киносалон може да включва множество подобрения и нови функционалности, които ще подобрят потребителския опит и ефективността на управлението на киносалона. Като:

- Управление на билети: Добавяне на функционалност за редактиране и изтриване на билети. Възможност за печат на билети или изпращане на електронни билети към клиентите.
- Интеграция с плащане: Интегриране на платежни шлюзове за онлайн плащания за резервации на билети. Поддръжка на различни методи за плащане като кредитни карти, PayPal и др.
- Анализ на данни и отчети: Добавяне на аналитични инструменти за проследяване на продажбите, популярността на филмите и други статистически данни. Генериране на отчети за финансовата производителност, запълненост на зали, анализ на публиката и други.
- Управление на филмови програми: Възможност за съставяне на програми за филми и събития с възможност за предварително планиране на графика. Автоматизация на процеса за добавяне и актуализиране на филми и графици.
- Интерактивност за потребителите: Възможност за коментари и оценки на филмите от страна на потребителите.
- Управление на персонала: Добавяне на модул за управление на персонала с възможности за програмиране на графици, заплащане и следене на производителността.
- Интеграция на социални медии: Интегриране на социални медии за споделяне на информация за предстоящи събития и филми. Мобилно приложение:

- Разработка на мобилно приложение за бърз достъп до програмата на киносалона, закупуване на билети и получаване на известия за специални оферти и събития.

8. Заключение

Дипломната работа представя процеса на разработка на система за управление на киносалон като ключов етап за модернизиране и подобряване на операциите в тази индустрия. Важността на такава система е непреходна, тъй като киносалоните са изправени пред изискванията на все по-изискващите и технологично напреднали потребители.

Системата за управление на киносалон играе решаваща роля в оптимизирането на бизнес процесите и подобряването на клиентското изживяване. Тя предоставя инструменти за ефективно управление на програмата, продажбата на билети, резервациите, инвентара и персонала. Освен това, тя предоставя възможности за анализ на данните, което позволява на киносалоните да вземат по-информирани решения и да адаптират стратегиите си спрямо нуждите на пазара.

Технологиите, използвани за създаването на тази система, са съобразени с най-новите тенденции в софтуерната индустрия. Включват се модерни програмни езици, бази данни и инструменти за разработка, които гарантират бързо и надеждно функциониране на приложението. В същото време, се изследват възможностите за интеграция с други системи и услуги, което осигурява съвместимост и синхронизация с вече съществуващите технологии в киносалоните.

Потенциалните ползи, които системата за управление на киносалон може да донесе за индустрията на киното, са значителни. Тя може да увеличи ефективността на операциите, да подобри обслужването на клиентите и да осигури по-голяма конкурентоспособност на бизнеса. Същевременно, тя може да създаде възможности за иновации и развитие в индустрията, като отговори на изменящите се нужди и предизвикателства в сферата на киноизкуството и развлекателната индустрия като цяло.

Процесът на разработка на системата за управление на киносалон е от съществено значение за модернизирането на индустрията на киното и предоставяне на по-качествени и иновативни услуги на клиентите. С тази дипломна работа се цели да се представи важността и потенциала на подобни технологични решения за развитието на киносалоните и индустрията на киното като цяло.

9. Използвана литература

- MySQL.com
- W3school
- Wikipedia-C#
- <https://learn.microsoft.com/en-us/C#/>
- GitHub Docs
- Windows Presentation Foundation documentation