**Movie Booking App**

GIT repository: <https://github.com/MziKay3/SMA-proiect>

**What?**

The Movie Booking App is a Android application designed to simplify the process of reserving movie tickets. It addresses the functionality of selecting films, showtimes, and seats, providing an easy solution for users to browse schedules, pick their preferred seats, and confirm bookings effortlessly.

**Why?**

The motivation behind the Movie Booking App lies in enhancing the convenience and efficiency of reserving movie tickets, which remains a popular activity but often involves outdated, and non-intuitive systems. While similar apps exist, this project focuses on combining an intuitive seat-selection interface, customizable showtime options, and integrated ticket confirmation to elevate the user experience.

By implementing this app, users can:

1. Avoid long queues or miscommunication at physical ticket counters.
2. Enjoy an accessible and easy to use interface tailored for mobile use.
3. Gain greater control over movie preferences, dates, and premium seat options, improving satisfaction.
4. **BookMyShow**

**Platform:** Google Play & App Store  
**Overview:** BookMyShow is a popular app for booking movies, events, plays, and other activities. It integrates ticket reservations with reviews, trailers, and multiple payment methods.

**Pros:**

* Offers a wide range of options beyond movie tickets, making it a versatile entertainment hub.
* Integration with digital wallets and payment gateways ensures quick, seamless transactions.
* Provides notifications for upcoming releases, discounts, and recommendations based on viewing history.

**Cons:**

* The interface can feel crowded due to the variety of activities supported, making it less focused for users primarily interested in movies.
* No highly interactive or innovative seat-selection feature; the layout is functional but not visually engaging.
* Overwhelming options may distract users who prefer a simplified, streamlined experience.

1. **Cinepolis**

**Platform:** Google Play & App Store  
**Overview:** Cinepolis is the official app for booking tickets at Cinepolis cinemas. It focuses exclusively on movie ticket reservations for a specific chain.

**Pros:**

* Straightforward and focused entirely on movies.
* Real-time updates on ticket availability and offers for members.
* Integrated loyalty programs provide benefits for frequent users.

**Cons:**

* Limited to Cinepolis theaters, reducing its utility for users who visit other chains.
* Feature set is minimal, with basic seat selection and schedule browsing.

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **BookMyShow** | **Cinepolis** |
| ***Store link*** | Google Play / App Store | Google Play |
| ***Store grade*** | 4.4 / 5 | 4.3 / 5 |
| ***Nr. installs*** | 100M+ | 10M+ |
| ***Nr. ratings*** | 6M+ | 300K |
| ***Ads/ in-app purchases*** | x | x |
| *Login/user* | x |  |
| *Update in real-time* | x | x |
| *Notifications* | x | x |
| *Seat visualization* | Basic | Minimal |

### ****Description of Use Cases (Functionalities)****

### 1. *****Select Movie:*****

* **Functionality**: The app provides a list of available movies in a dropdown spinner. When a user selects a movie, the available screening dates for that movie appear in another spinner, enabling them to pick a date.
* **Implementation**:
  + **UI Components**: Spinner for movie selection (movieSpinner), Spinner for date selection (dateSpinner).
  + **Flow**: When a movie is selected, an AdapterView.OnItemSelectedListener for movieSpinner dynamically updates the available dates using dateSpinner.
* **Technology**: Spinner widgets for dropdown lists.



Movie selected

No movie selected

**2.** ***Select Date and Time:***

* **Functionality**: After selecting a date, available time slots are shown to the user. The user selects a time via a list of radio buttons.
* **Implementation**:
  + **UI Components**: RadioGroup for time selection (timeRadioGroup), dynamically filled with time options based on the selected date.
  + **Flow**: When a date is selected, the dateSpinner triggers the population of available times.
* **Technology**: Dynamic population of RadioButton inside RadioGroup based on the date selected.



Movie date selected

No movie selected (no date available)



No movie selected (no time stamp available)

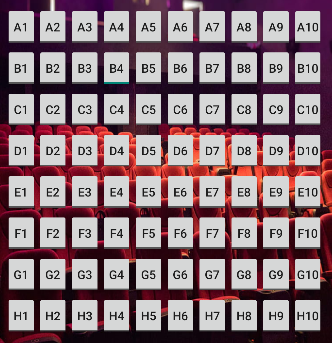
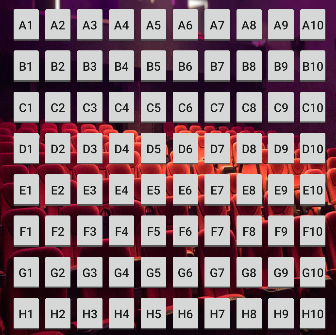
No time stamp selected



Time stamp selected

**3.** ***Select Seats:***

* **Functionality**: The user selects seats using a grid of buttons represented by ToggleButton inside a GridLayout.
* **Implementation**:
  + **UI Components**: GridLayout for seat layout (seatGrid), ToggleButton for each seat.
  + **Flow**: Seats are dynamically created, and users toggle between selected/unselected states. A ToggleButton.setOnCheckedChangeListener listens to the seat selections.
  + **Price Update**: As seats are selected, the app updates the total price, with premium seats costing more (rows "F", "G", "H").
* **Technology**: ToggleButton, GridLayout, dynamic view updates based on state changes.

B4 seat selected

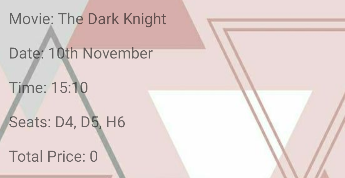
No seats selected

No valid selections (price is 0)

Price calculated after selections

**4. *Booking Confirmation:***

* **Functionality**: Once the user selects a movie, time, and seats, they can press a "Book Ticket" button. This triggers the creation of a summary page displaying the selected movie, time, seats, and total price.
* **Implementation**:
  + **UI Components**: Button for ticket booking (bookTicketButton), TextViews to display movie details.
  + **Flow**: Clicking the "Book Ticket" button creates an Intent to move to a new activity (TicketConfirmationActivity) that shows the summary of the booking.
* **Technology**: **Intents** for navigation and passing data between activities, **TextView** to show booking information.



Book ticket button

Ticket summary

**5.*****Return Button in Confirmation:***

* **Functionality**: A return button allows the user to go back to the previous screen and clear all selections.
* **Implementation**:
  + **UI Components**: Button for returning (returnButton).
* **Technology**: Calls finish() to close the current activity and go back to the main activity.



Return button

**6. *Deselect All option:***

* **Functionality**: Users can clear all selections with a "Deselect All" button, removing all selected seats, clearing movie selection, and resetting time options.
* **Implementation**:
  + **UI Components**: Button for deselection (deselectAllButton), resets the entire UI when clicked.



Deselect All button

**Technologies and Libraries:**

* + **Spinner**: Used for movie and date selections.
  + **RadioGroup & RadioButton**: Used for selecting available showtimes.
  + **GridLayout & ToggleButton**: Custom grid of seat selection (dynamic generation and interaction via buttons).
  + **Button**: For booking confirmation and deselection.
  + The state of the app (movie, date, time, seats, total price) is managed using simple variables and updated whenever any interaction happens.
  + **Intents**: Used to pass data between the MainActivity and TicketConfirmationActivity when the booking is confirmed.

**Android Specific Concepts:**

1. **Activity Lifecycle**:
   * onCreate() initializes the user interface and sets up listeners for UI components like spinners and buttons.
2. **Adapters & Listeners**:
   * **ArrayAdapter** is used for populating the spinners (movieSpinner, dateSpinner) with movie titles and available dates.
3. **GridLayout**:
   * The grid of seats is dynamically populated by creating ToggleButton for each seat and setting it within a GridLayout.
4. **Intent Communication**:
   * Data is passed to the confirmation screen via Intent.putExtra(). This includes movie, date, time, seats, and total price for display on the confirmation page.
5. **UI Customization**:
   * Custom styling like white text on spinners, and background color and layout adjustments are implemented for UI consistency.
6. **Event Handling**:
   * Listeners such as onCheckedChangeListener for ToggleButton (seats) and OnItemSelectedListener for spinners are used to update the app state and the UI dynamically.

**Conclusion:**The design and performance of the app came out as planned , even though it could use many improvements visually and design/functionality wise. I liked creating the design and thinking of every option I could use for a certain functionality so that it doesn’t get repetitive or boring for the user using the app. Unfortunately the only problem that I didn’t manage to solve was the local database , unfortunately I had to resort to storing the data directly into the code. Overall it was an interesting project that I enjoyed.