Project Name: Movie Reservation

Version: v1.0

Release Date: 12/12/2021

Author: Albert Christian G. Bangan

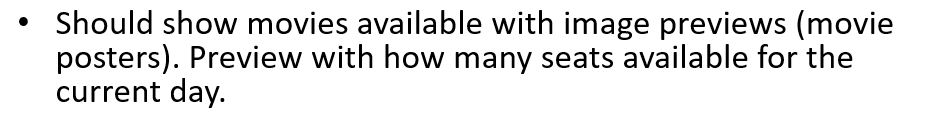
INITIAL RELEASE

---------------------------------------------------------------------

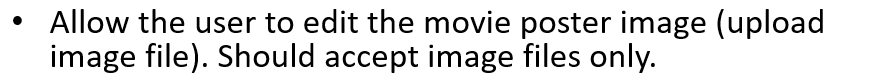
RELEASE NOTES

---------------------------------------------------------------------

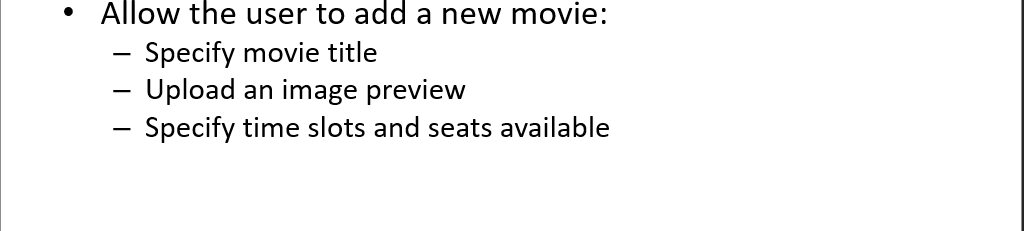
Progress:



* Done with show movies with image previews
* Have bug with available seats / slightly working



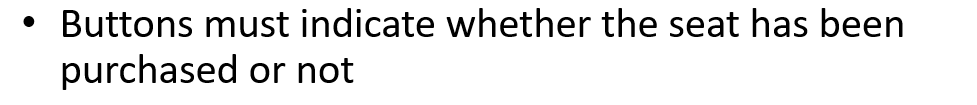
* Done



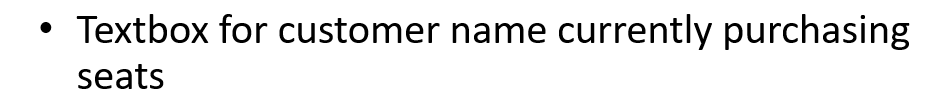
* Done



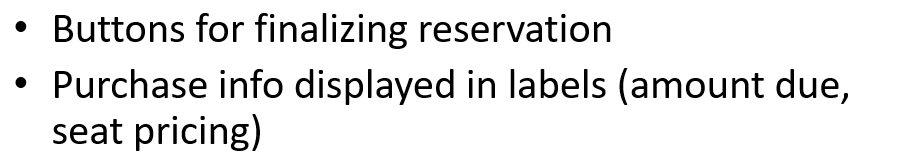
* Partially Done

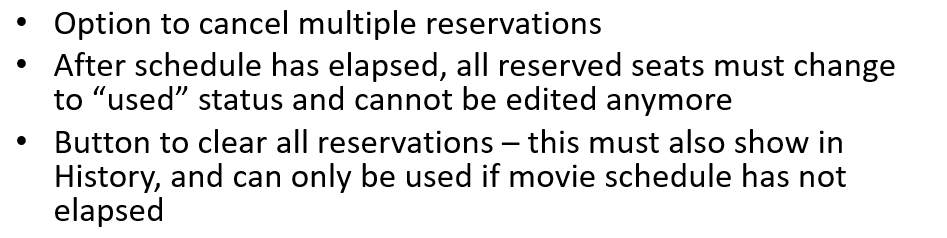


* Partially done (bought seats checkboxes are disabled)

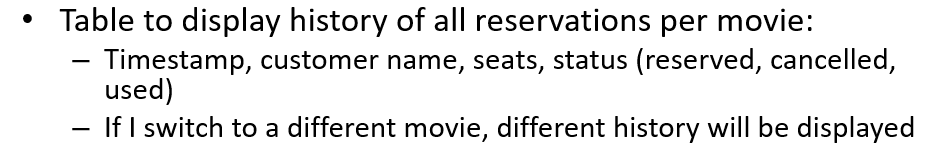


* Done





* Not Started



* Partially Done (Created SP)

**SYSTEM INFO:**

DATABASE-SIDE

Tables:

Logs

- ID INT PK NOT NULL

- Date DateTime NOT NULL

- Thread VARCHAR(255) NOT NULL

- Level VARCHAR(50) NOT NULL

- Logger VARCHAR(255) NOT NULL

- Message VARCHAR(4000) NOT NULL

- Exception VARCHAR(2000) NULLABLE

Movies

- MovieId PK INT NOT NULL

- Title VARCHAR(1000) NOT NULL

- Description VARCHAR(1000) NOT NULL

- Image\_path VARCHAR(1000) NOT NULL

- Date\_Added DateTime NOT NULL

Schedules

- ScheduleId INT PK NOT NULL

- ScheduleTime VARCHAR(255) NOT NULL

- MovieId FK

Keys

- SeatId INT PK NOT NULL

- SeatValue VARCHAR(2) NOT NULL

- Status VARCHAR(10) NOT NULL

- MovieId FK

Transactions

- TransactionId INT PK NOT NULL

- CustomerName VARCHAR(1000) NOT NULL

- TransactionDate DateTime NOT NULL

- ReservedSeats VARCHAR(2) NOT NULL

- MovieId FK

Stored Procedures:

spAvailableSeats

- a stored procedure that get the count of available seats per schedule per movie

Parameter/s: @MovieId int, @ScheduleTime varchar(10)

spEditImage

- a stored procedure that will edit movie image

Parameter/s: @newPath varchar(1000), @id int

spGetHistory

- a stored procedure that will get history per movie

Parameter/s: @MovieId int

spGetMovies

- a stored procedure that will return all movies

spGetSchedules

- a stored procedure that will return all schedules by movie

Parameter/s: @MovieId int

spGetSeatsById

- a stored procedure that will return seat per movie

Parameter/s: @MovieId int

spGetTransactionById

- A stored procedure that selects all transactions by id

Parameter/s: @MovieId int

spInsertMovie

- A stored procedure that creates movie data

Parameter/s:@Title varchar(1000), @Description varchar(1000), @Image\_path varchar(1000), @Date\_added DateTime, @Identity INT OUT

spInsertSchedules

- A stored procedure that creates schedule data

Parameter/s: @ScheduleTime varchar(20), @Identity int

spInsertSchedules

- A stored procedure that creates seats data

Parameter/s: @SeatValue varchar(2), @Identity int

spInsertTransactions

- A stored procedure that creates schedules data

Parameter/s: @CustomerName varchar(255), @TransactionDate DateTime, @ReservedSeats varchar(255), @MovieId int

.NET-SIDE

Models

- contains all models

Models > Repository

- Planned to make it repository pattern and create unit tests but I don't have enough time to implement

- Contains all data logic

Views

- Contains all UI

Controller

- Contains all controllers