CQRS Transaction Processing with Business Validation Rules

Your Name: ___	**GitHub User Name:** ___

> Place your name and your GitHub user name in the lines above.

Marking Summary

Evaluation Item	Item Weight	Earned
Rule Movie cannot begin before 11:00am	2	**TBA**
Rule Movie and cleanup cannot extend after 11:00pm	3	**TBA**
Rule Movie start time cannot overlap previous movie/cleanup end time	3	**TBA**
Error messages Distinct exceptions created for each possible error	2	**TBA**
Error messages All possible errors collected into a single thrown exception	1	**TBA**
Data Processing Add showtime rows with correct data	3	**TBA**
Penalties Data processing **NOT** completed as a single transaction	-3	
Total	*15*	**TBA**

Marking Rubric

Weight	Breakdown
1	1 = Proficient (requirement is met) 0 = Incomplete (requirement not met)
2	2 = Proficient (requirement is met) 1 = Limited (requirement is poorly met, minor errors) 0 = Incomplete (requirement not met, missing large portions)
3	3 = Proficient (requirement is met) 2 = Capable (requirement is adequately met, minor errors) 1 = Limited (requirement is poorly met, major errors) 0 = Incomplete (requirement not met, missing large portions)

Database

The physical database can be installed from the `.bak` file in the starter kit.

About OMST

> **Off Main Street Theatres** is a local movie theatre chain with multiple locations.

Overview

In this assessment, you will be demonstrating your understanding of CQRS using EntityFramework. In this assessment, you will be evaluated on the following:

- Implement a controller method for bulk processing a set of data records within a transaction.
- Implement logical validation on data within a BLL controller method transaction; collect all errors and throw them in an appropriate instance of the BusinessRuleException.

You have been supplied a starting solution for this assessment called AssessmentOMST in your assessment classroom repository. The repository contains the database in a .bak (backup file) called OMST_2018.bak. The starting solution has certain portions of the assessment pre-coded. **This code works and should not be altered.**

You are to complete each of the activities to create a successful solution to this assessment. You will need to use specified names in portions of the activities to integrate with the existing code. You many need to create local variables to use in your answer (these variables can be called whatever you wish). Use the following activity instructions to complete this assessment.

Submission:

Commit your work at the end of every Activity. Ensure you sync your local clone to the classroom on github before the end of class. The classroom access ends at the end of class.

Activity Setup:

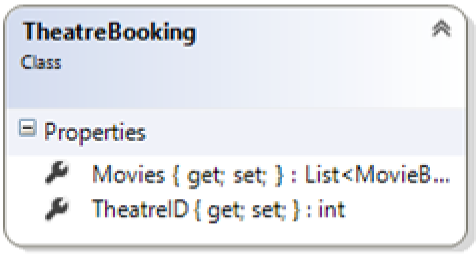
> **USE CHROME FOR TESTING** for a consistent experience with date and time input.

Restore the supplied SQL database. The database name is **OMST_2018**. The database contains data for testing your solution. Run the solution. You should be able to open all forms. The Movie Scheduler page should allow you to:

- **READ** the instructions and information on the web form.
- Select location and theatre plus enter the date and time data
- Press Prepare Theatres which **either** shows a message if the Theatre already has scheduled shows **or** a row of 4 movie scheduling slots. The test for already scheduled shows *has been coded for you* and is determine when the Prepare Theatres button is pressed.
- Press Schedule Movies to collect the scheduling data and sent it to the BLL method for processing (validation and adding records to the database in a transaction).

Activity 1: BLL method: ScheduleMovies.

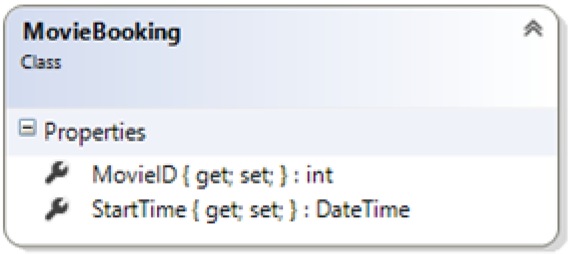
You have only one method to code. This method will receive a view model instance. Within this instance are the TheatreID being scheduled and a `List<MovieBooking>` which represents a particular movie and start time (`DateTime`). Bookings contains 4 movie booking rows representing the 4 movie time slots.



TheatreBooking
Class

Properties

Movies { get; set; } : List<MovieB...
TheatreID { get; set; } : int



MovieBooking
Class

Properties

MovieID { get; set; } : int
StartTime { get; set; } : DateTime

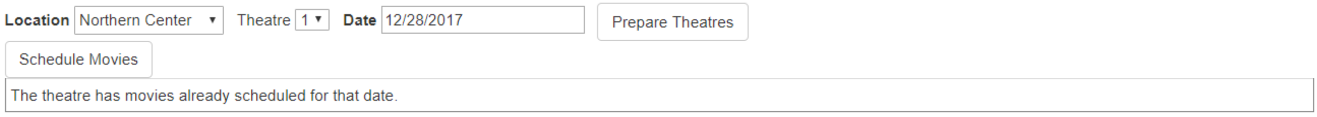
Business Rules

1. There is a 20 minute clean up period between movies in a theatre. The next movie cannot start within this 20 minutes buffer at the end of the previous movie.
2. No movie can start before 11:00 am.
3. No movie and clean up period can extend pass 11:00 pm.

A `List<Exception>` has been created for you to store messages about any movie that violates one or more of these rules. If a movie violates one or more of these rules, generate and store an error message identifying the movie and the start time. **All** movies must be validated for all rules. All violations must be returned. Check this list before committing your transaction. If any errors exist do not commit your transaction. Use the `BusinessRuleCollectionException()` to throw the collection of error messages.

Your logic will require you use DateTime objects. You may use the internet to query how to manipulate DateTime and TimeSpan data.

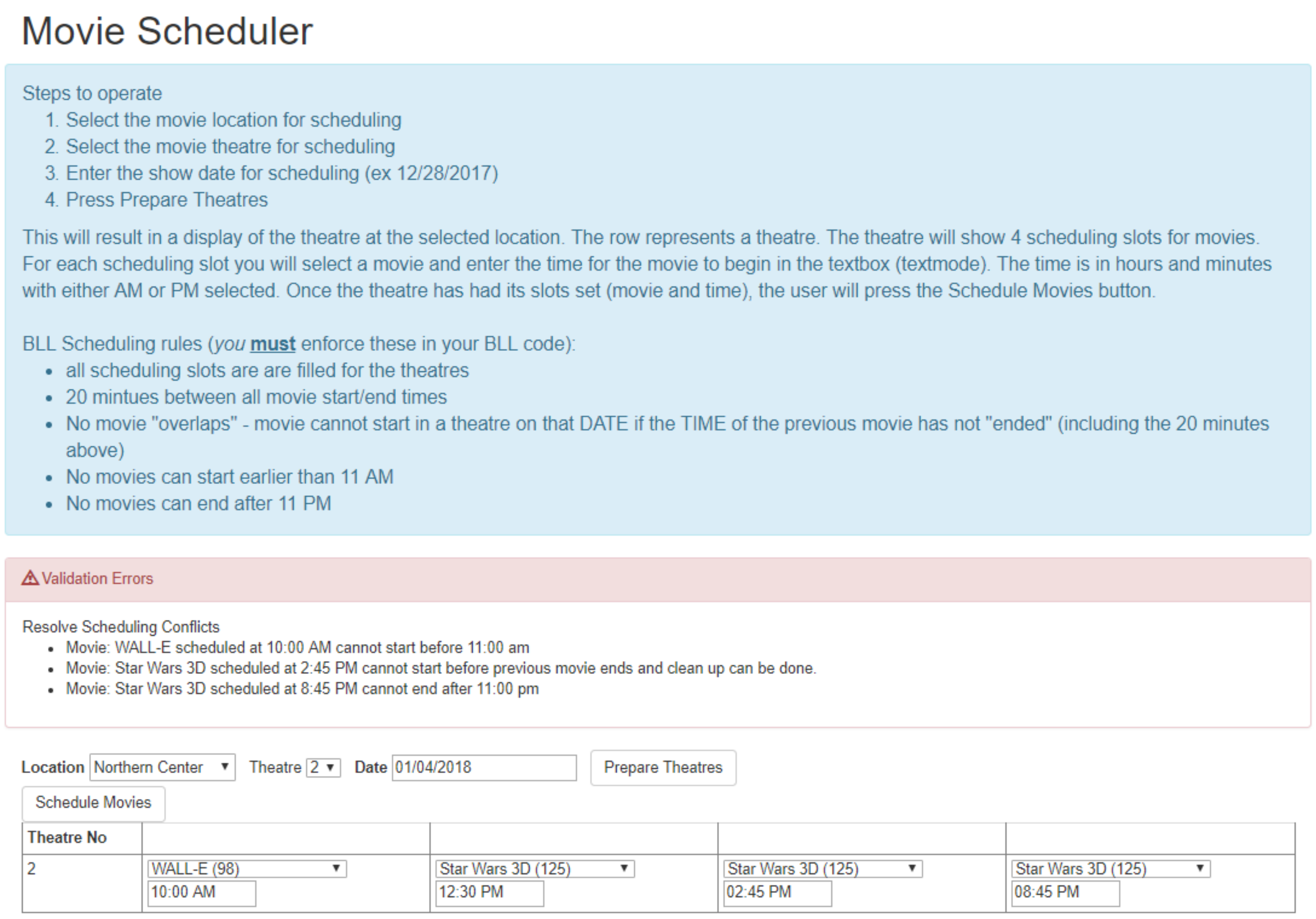
Sample of selecting a location, theatre and date that already has scheduled movies. If this message appears when Prepare Theatres is pressed then you will not be able to enter new show times.



Location Northern Center Theatre 1 Date 12/28/2017

The theatre has movies already scheduled for that date.

After calling the BLL method, the UI will display the business rule violations identified in the thrown `BusinessRuleCollectionException`.



Movie Scheduler

Steps to operate

1. Select the movie location for scheduling
2. Select the movie theatre for scheduling
3. Enter the show date for scheduling (ex 12/28/2017)
4. Press Prepare Theatres

This will result in a display of the theatre at the selected location. The row represents a theatre. The theatre will show 4 scheduling slots for movies. For each scheduling slot you will select a movie and enter the time for the movie to begin in the textbox (textmode). The time is in hours and minutes with either AM or PM selected. Once the theatre has had its slots set (movie and time), the user will press the Schedule Movies button.

BLL Scheduling rules (*you* **must** enforce these in your BLL code):

- all scheduling slots are filled for the theatres
- 20 mintues between all movie start/end times
- No movie "overlaps" - movie cannot start in a theatre on that DATE if the TIME of the previous movie has not "ended" (including the 20 minutes above)
- No movies can start earlier than 11 AM
- No movies can end after 11 PM

⚠ Validation Errors

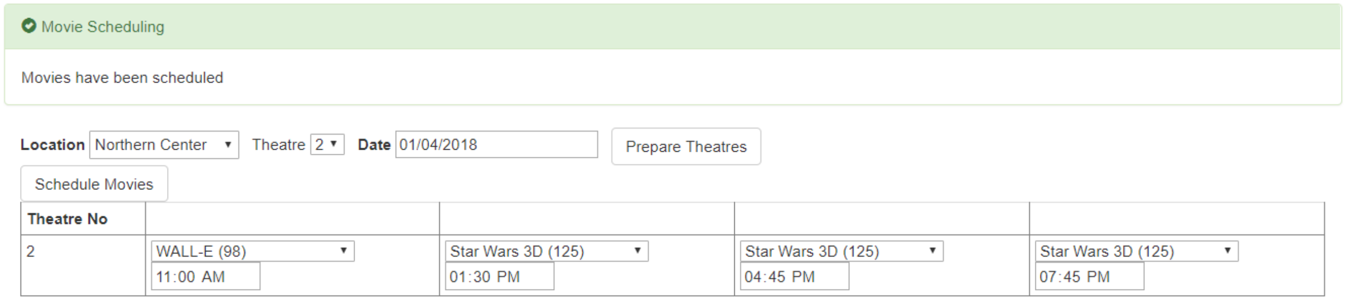
Resolve Scheduling Conflicts

- Movie: WALL-E scheduled at 10:00 AM cannot start before 11:00 am
- Movie: Star Wars 3D scheduled at 2:45 PM cannot start before previous movie ends and clean up can be done.
- Movie: Star Wars 3D scheduled at 8:45 PM cannot end after 11:00 pm

Location Northern Center ▾ Theatre 2 ▾ **Date** 01/04/2018

Theatre No				
2	WALL-E (98) ▾ 10:00 AM	Star Wars 3D (125) ▾ 12:30 PM	Star Wars 3D (125) ▾ 02:45 PM	Star Wars 3D (125) ▾ 08:45 PM

If there are no exceptions in your BLL, the UI will display a success message.



✓ Movie Scheduling

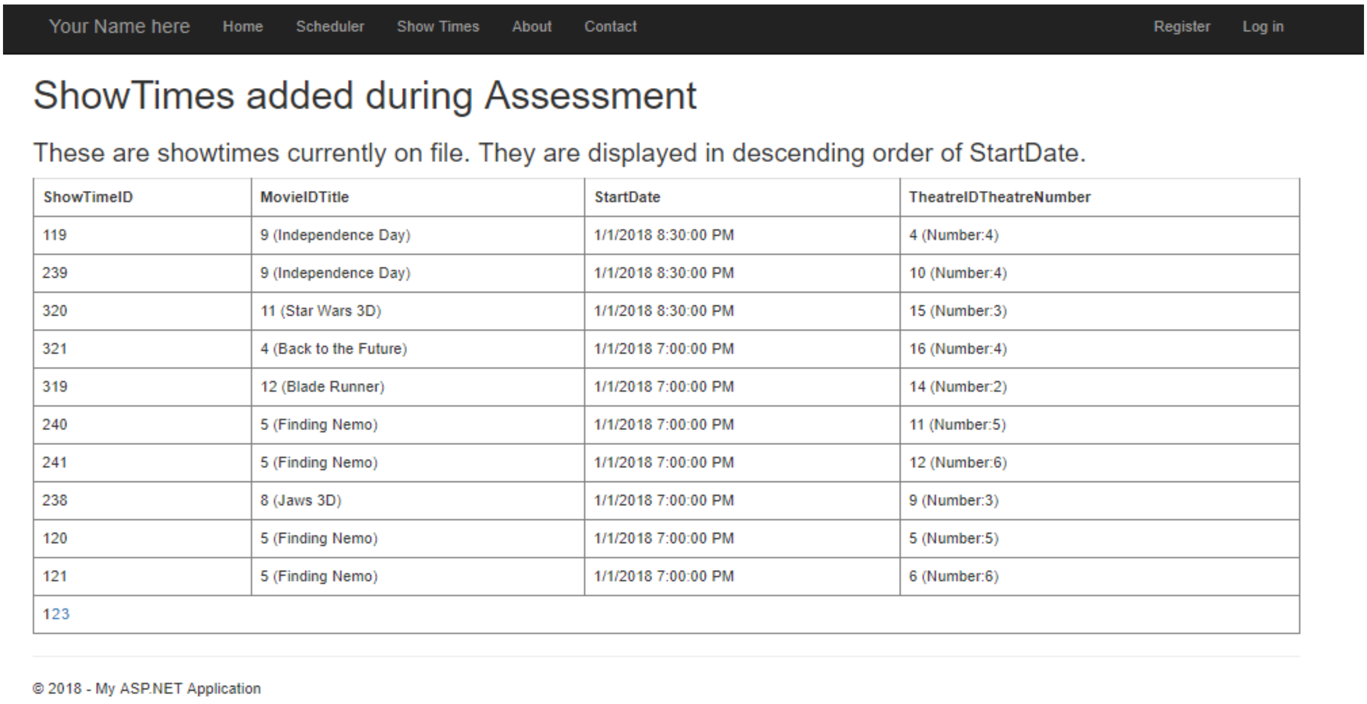
Movies have been scheduled

Location Northern Center ▾ Theatre 2 ▾ **Date** 01/04/2018

Theatre No				
2	WALL-E (98) ▾ 11:00 AM	Star Wars 3D (125) ▾ 01:30 PM	Star Wars 3D (125) ▾ 04:45 PM	Star Wars 3D (125) ▾ 07:45 PM

View New ShowTimes

You have been given a web form to view newly created ShowTimes. Use the menu item View New ShowTimes. The list will be in descending order of StartDate.

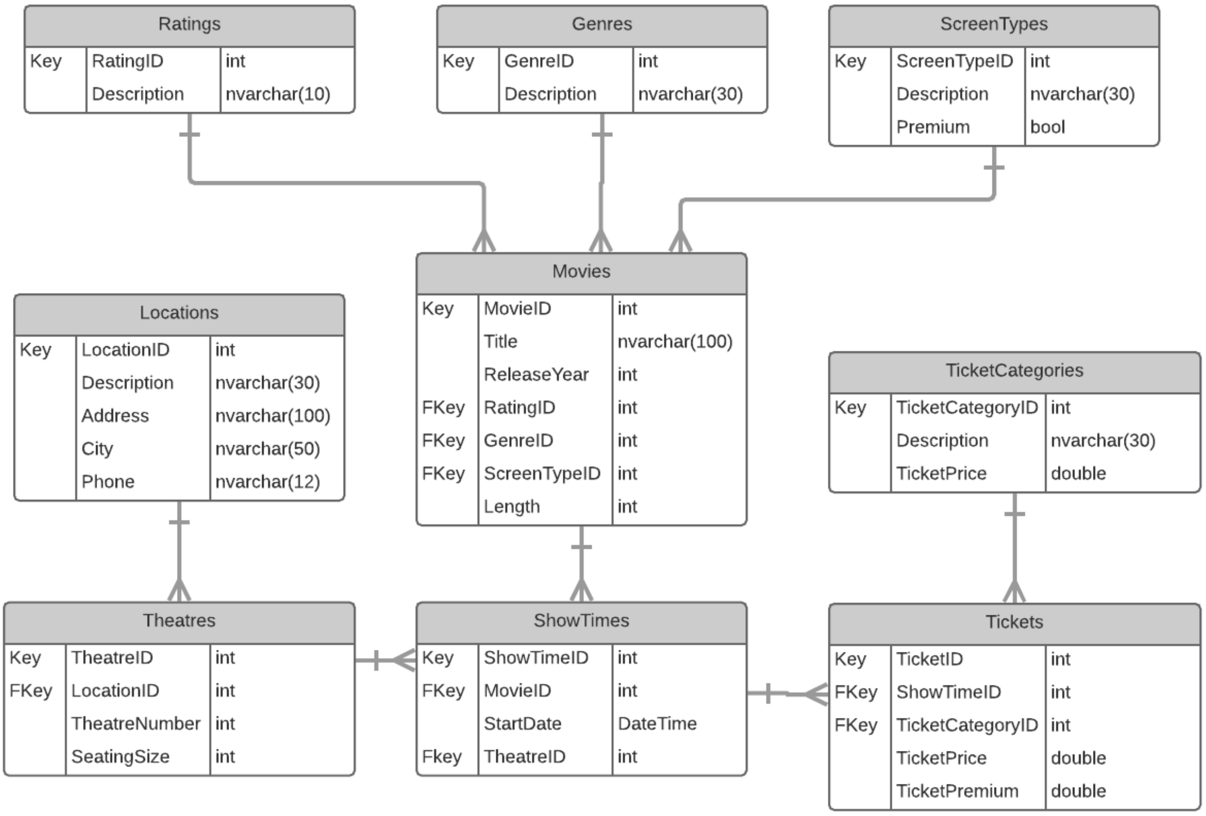


ShowTimes added during Assessment

These are showtimes currently on file. They are displayed in descending order of StartDate.

ShowTimeID	MovieIDTitle	StartDate	TheatreIDTheatreNumber
119	9 (Independence Day)	1/1/2018 8:30:00 PM	4 (Number:4)
239	9 (Independence Day)	1/1/2018 8:30:00 PM	10 (Number:4)
320	11 (Star Wars 3D)	1/1/2018 8:30:00 PM	15 (Number:3)
321	4 (Back to the Future)	1/1/2018 7:00:00 PM	16 (Number:4)
319	12 (Blade Runner)	1/1/2018 7:00:00 PM	14 (Number:2)
240	5 (Finding Nemo)	1/1/2018 7:00:00 PM	11 (Number:5)
241	5 (Finding Nemo)	1/1/2018 7:00:00 PM	12 (Number:6)
238	8 (Jaws 3D)	1/1/2018 7:00:00 PM	9 (Number:3)
120	5 (Finding Nemo)	1/1/2018 7:00:00 PM	5 (Number:5)
121	5 (Finding Nemo)	1/1/2018 7:00:00 PM	6 (Number:6)
123			

© 2018 - My ASP.NET Application



Ratings

Key	RatingID	int
	Description	nvarchar(10)

Genres

Key	GenreID	int
	Description	nvarchar(30)

ScreenTypes

Key	ScreenTypeID	int
	Description	nvarchar(30)
	Premium	bool

Movies

Key	MovieID	int
	Title	nvarchar(100)
	ReleaseYear	int
FKey	RatingID	int
FKey	GenreID	int
FKey	ScreenTypeID	int
	Length	int

Locations

Key	LocationID	int
	Description	nvarchar(30)
	Address	nvarchar(100)
	City	nvarchar(50)
	Phone	nvarchar(12)

TicketCategories

Key	TicketCategoryID	int
	Description	nvarchar(30)
	TicketPrice	double

Theatres

Key	TheatreID	int
FKey	LocationID	int
	TheatreNumber	int
	SeatingSize	int

ShowTimes

Key	ShowTimeID	int
FKey	MovieID	int
	StartDate	DateTime
Fkey	TheatreID	int

Tickets

Key	TicketID	int
FKey	ShowTimeID	int
FKey	TicketCategoryID	int
	TicketPrice	double
	TicketPremium	double

OMST ERD

All fields are required