Daniel Varela Sanchez
Guillermo Martin-Coello Juarez
Javier Lopez de la Nieta Lujan

daniel.varelas@estudiante.uam.es
guillermo.martin-coello@estudiante.uam.es
javier.lopezdelanieta@estudiante.uam.es

TEST PLAN AND TEST RESULTS

Application: Theathecption

DATE: 18/05/2021

1. TEST CASE: BUY A TICKET

1.1 Use Case: Buy a ticket

Primary Actor : Registered User.
Stakeholders and Goals:
Registered User: buy a ticket for a chosen play.
Preconditions:
The user has selected a play by searching it after having logged in.
Success guarantee (Post-conditions):
The user buys a ticket for the selected play, and a pdf with all the information and an encrypted and unique code for each ticket. The capacity of the theater gets reduced by the number of tickets bought and the seats selected are now occupied so they cannot be selected for other users.

Main Success Scenario: (Scenario to be tested)

1. The user selects the date for the performance.
2. The user selects "purchase ticket".
3. The user chooses a standing zone for the performance.
4. The user chooses the number of tickets to buy.
5. The user selects to buy the tickets by credit card (by pass)
6. The user enters the credit card number and the CCV.
7. The system connects with the bank account and proceeds with the payment.
8. The system redirects the user to the "manage tickets" window where he can download the pdf.

Extensions (Alternative paths):

3a. The user selects a seating zone for the play.
3a.1. The user selects the seats for the play.
3a.2. The user selects to buy the tickets by credit card.
3a.3. The user enters the credit card number and the CCV.
3a.4. The system connects with the bank account and proceeds with the payment.
3a.5. The system redirects the user to the "manage tickets" window where he can download the pdf.

3a.2b The user selects to buy the tickets by pass.
3a.2b.1 The user selects the pass.
3a.3b.2 The system redirects the user to the "manage tickets" window where he can download the pdf.

3b The user selects a composed zone.
3b.a.1. The user selects the standing subzone.
3b.a.2. The user selects the number of tickets to buy.
3b.a.3. The user selects to buy the tickets by credit card.
3b.a.4.The user enters the credit card number and the CCV.
3b.a.5.The system connects with the bank account and proceeds with the payment.
3b.a.6.The system redirects the user to the "manage tickets" window where he can download the pdf.

3b.a.3a The user selects to buy the tickets by pass.
3b.a.3a.1 The user selects the pass.
3b.a.3a.2 The system redirects the user to the "manage tickets" window where he can download the pdf.

3b.b.1.The user selects the seating subzone
3b.b.2.The user selects the seats for the play.
3b.b.3.The user selects to buy the tickets by credit card
3b.b.4.The user enters the credit card number and the CCV.
3b.b.5.The system connects with the bank account and proceeds with the payment.
3b.b.6.The system redirects the user to the "manage tickets" window where he can download the pdf.

3b.b.3a The user selects to buy the tickets by pass.
3b.b.3a.1 The user selects the pass.
3b.b.3a.2 The system redirects the user to the "manage tickets" window where he can download the pdf.

6a The user selects to buy the tickets by pass.
6a.1 The user selects the pass.
6a.2 The system redirects the user to the "manage tickets" window where he can download the pdf.

Special Requirements:

A database to store all the information of the event and the available tickets.
A database to store the information of the users that have bought a ticket.

Technology and Data Variations List:

Graphical interface.
Possibility of downloading a pdf that contains the ticket bought.
Connection with a bank website to process the payments.

Frequency:

For this application we don't have to take into account the possibility of different users trying to access .

Open Issues:

Can a seating subzone have more subzones inside of it? Should it change the price of the ticket?
Can a standing zone have subzones inside of it? Should it change the price?

1.2. Test case design (including expected inputs and outputs)**:**

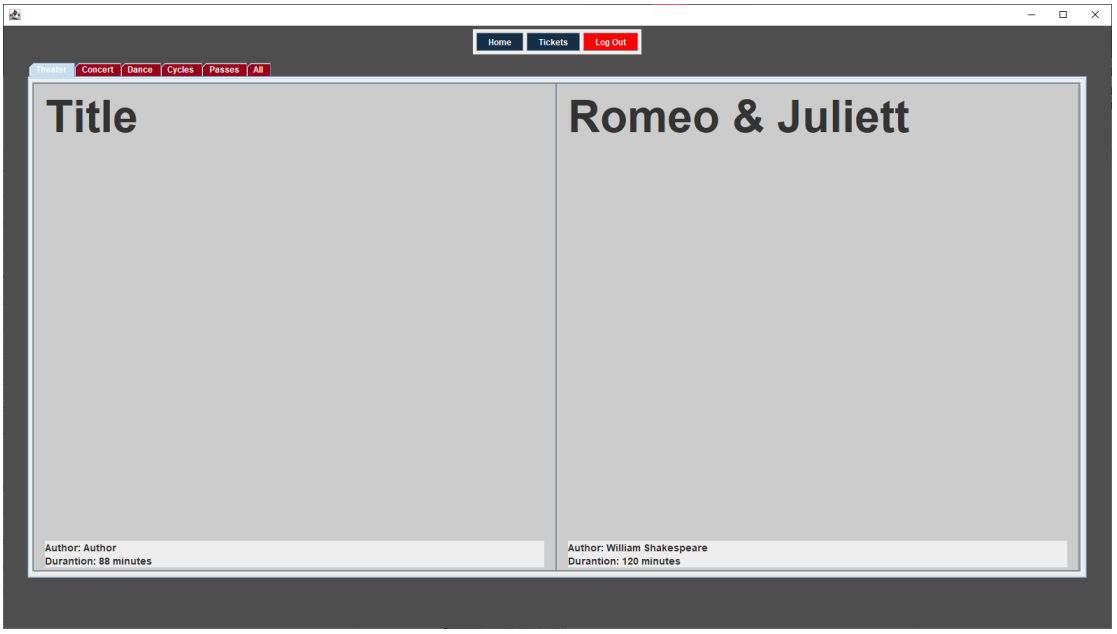
Preconditions:

The user has selected a play by searching it after having logged in.
And event with a performance and available tickets exist.

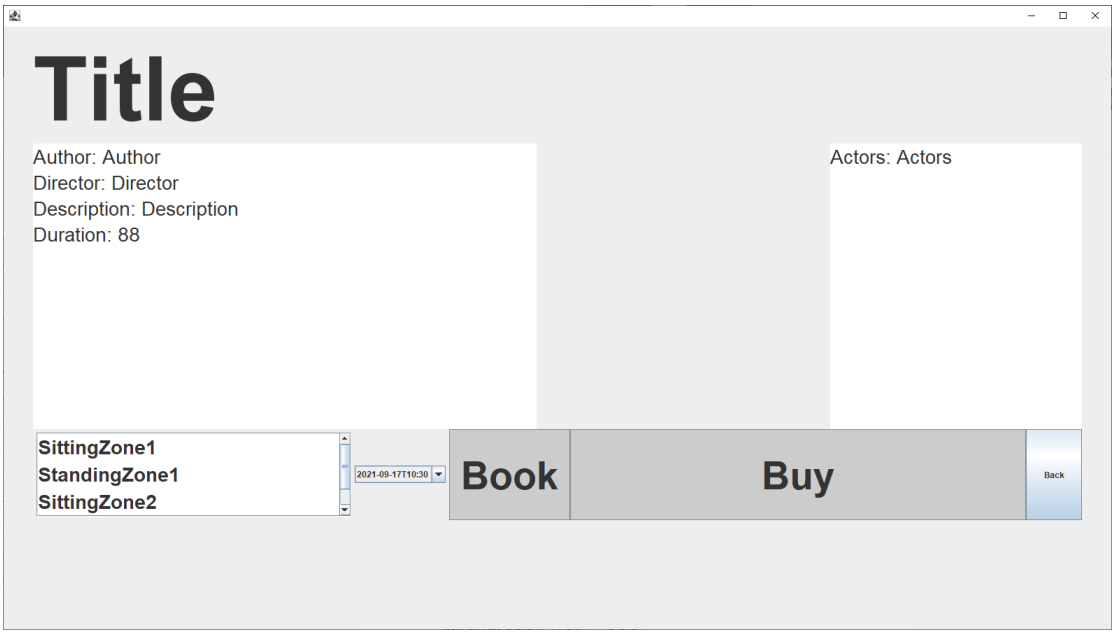
Scenario:

1. The clicks on the chosen event.
2. The user selects a zone.
3. The user chooses a performance and a standing zone for the performance.
4. The user selects to buy the tickets by credit card.
5. The user enters the credit card number.
6. The system connects with the bank account and proceeds with the payment.
7. The system redirects the user to the "manage tickets" window where he can download the pdf.

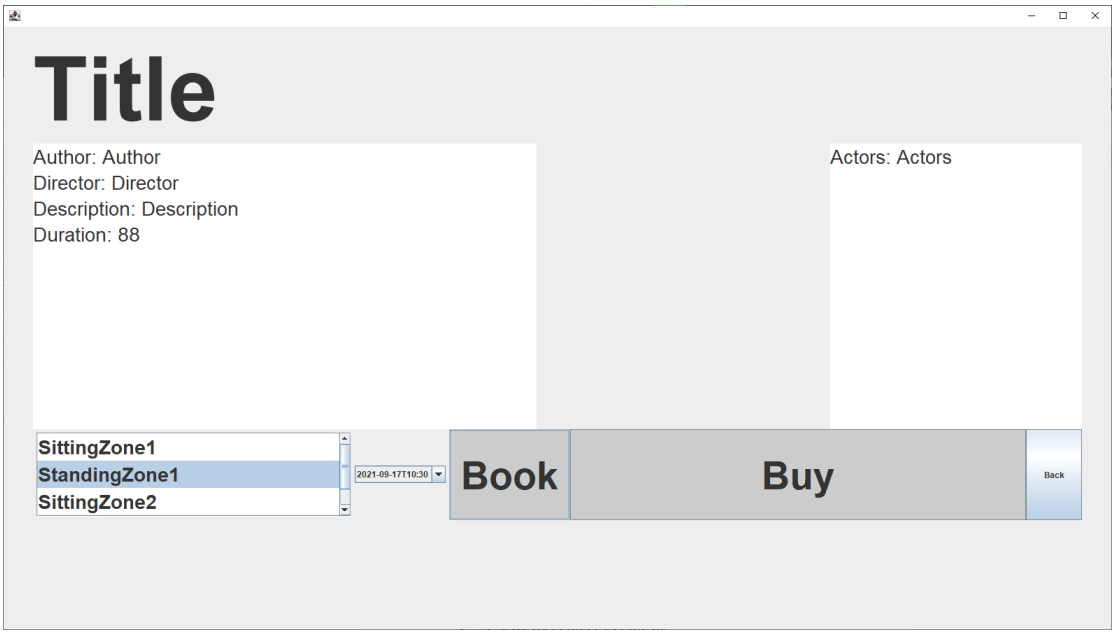
1.3. Test execution result



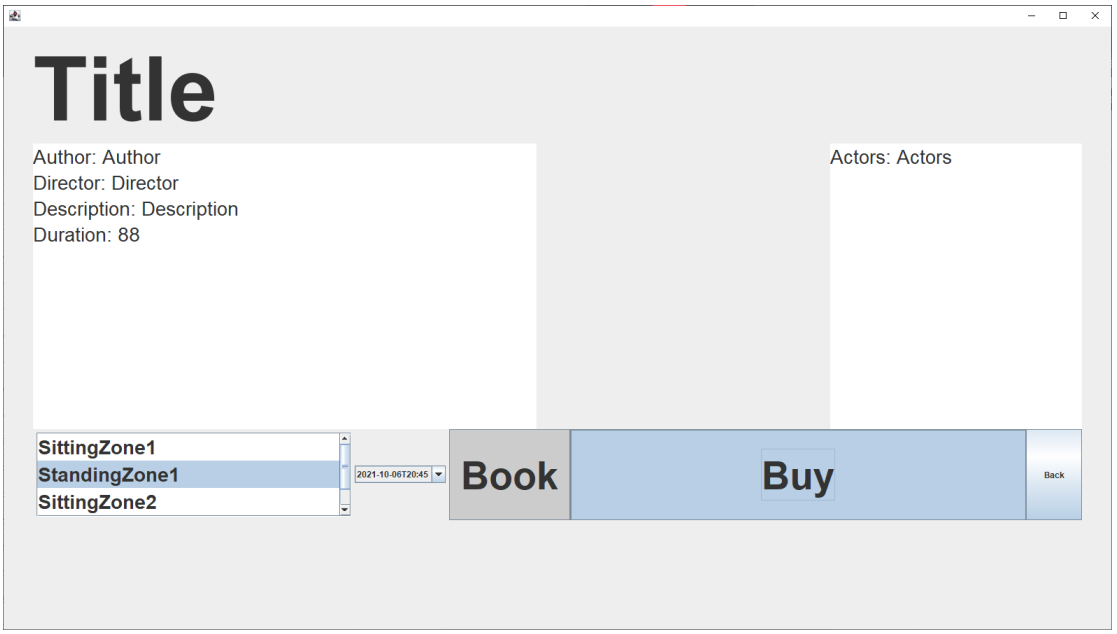
1. The user has to select an event to buy a pass of. In this case we are going to buy a ticket from
"Title". We forgot to include this simple step in our use case scenario.



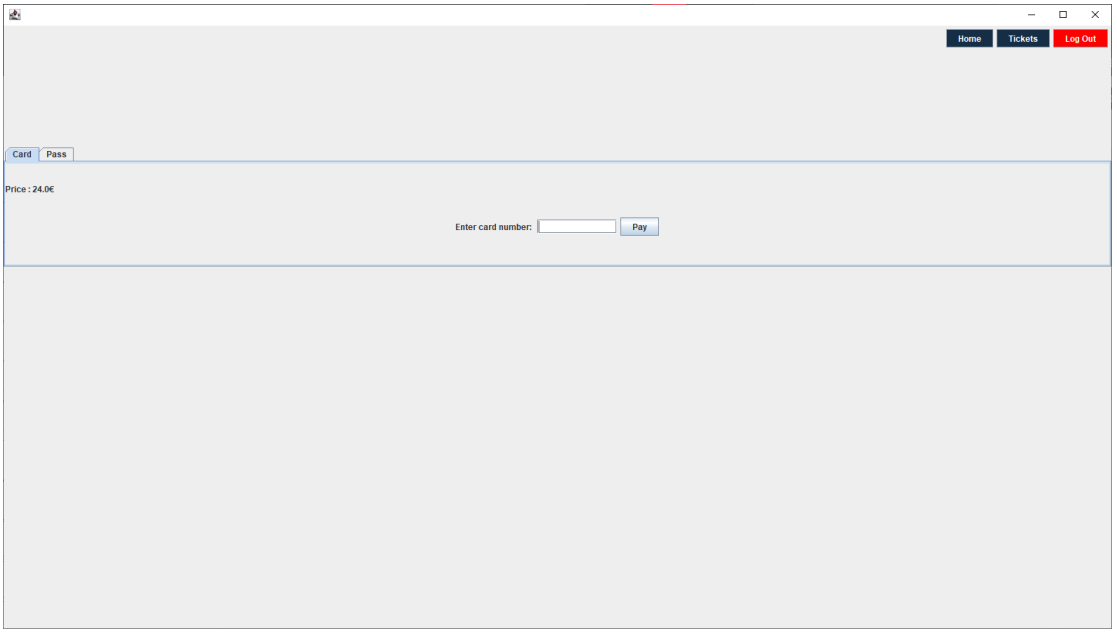
2. The user selects the zone to buy the ticket from. When we made the use case, we didn't know what the easiest design implementation would be. We finally decided to choose the zone to buy the ticket from, and then press the buy button.

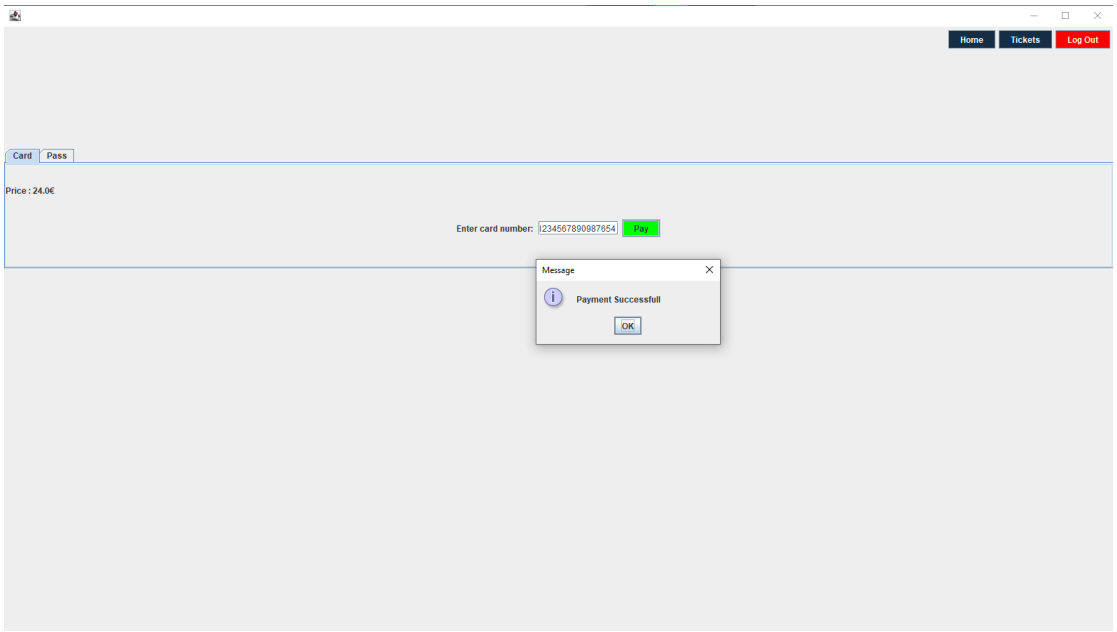


3. The user has selected the StandingZone1 and the desired date for the performance. When we made the use case, we didn't realize that the user also had to choose the date of the performance.

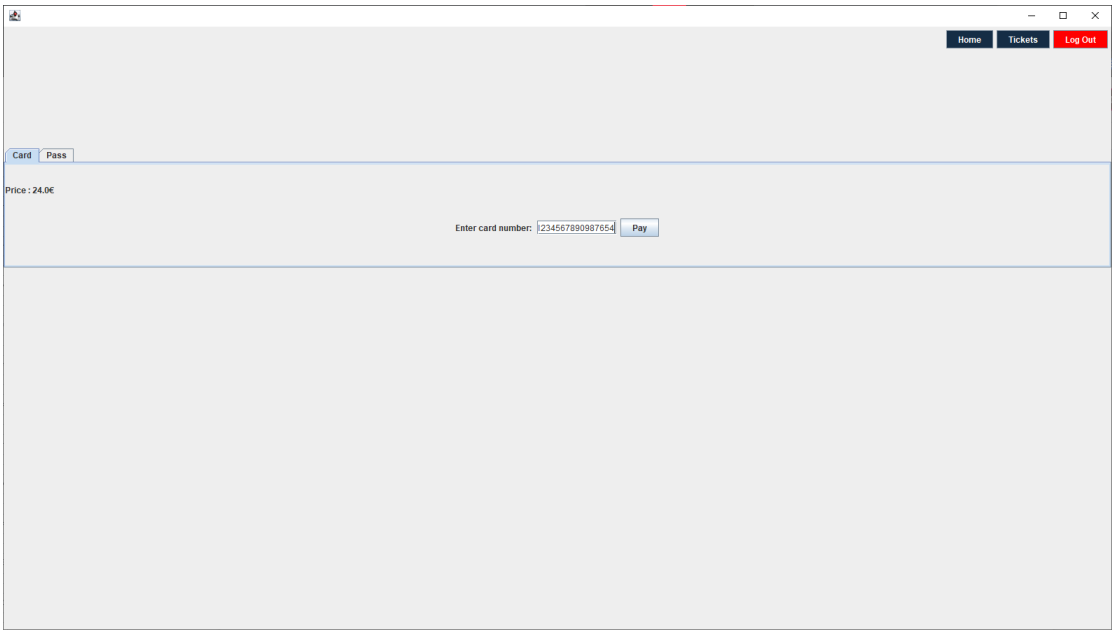


4. In this picture we click the buy button to buy the ticket for the selected date of the performance and the selected zone.

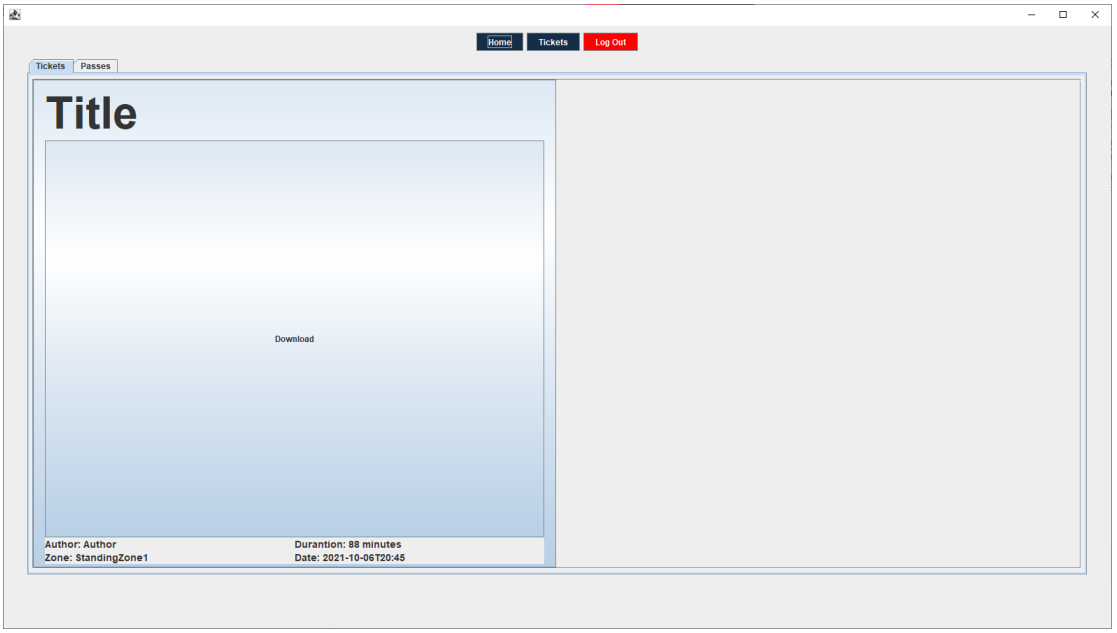




5. Here the user selects the option of paying the ticket by credit card.



6. The user writes his credit card number to do the payment.



7. Now the program redirects the user to the manage tickets window where he can download the ticket and where he can see the information about the performance.

2. TEST CASE : CREATE EVENT

2.2 Use Case: Create an event

Primary Actor : Administrator
Stakeholders and Goals:
Administrator to create an event with all the necessary data.
Preconditions:
The user has logged in as administrator and has a new event to add.
Success guarantee (Post-conditions):
The new event is added and ready for the users to find.

Main Success Scenario: (Scenario to be tested)

1. The manager selects the "create event" option.
2. The manager sets a director.

3. The manager sets an author.
4. The manager sets a title.
5. The manager sets a description.
6. The manager sets the price of each zone.
7. The manager sets the theater act as the event type.
8. The manager sets the actors of the theater act.
9. The manager finishes the event creation and the system adds the new event to the database.

Extensions (Alternative paths):

7a The manager sets "concert" as the event type.
7a.1. The manager sets the music pieces.
7a.2. The manager sets the orchestra name.
7a.3. The manager finishes the event creation and the system adds the new event to the database.
7b. The manager sets "dance" as the event type.
7b.2.The manager sets a dancer.
7b.3. The manager sets an orchestra.
7b.4. The manager sets a conductor.
7b.5. The manager finishes the event creation and the system adds the new event to the database.

Special Requirements:

A database for all the events.
A database of the users for login.

Technology and Data Variations List:

Graphical interface.

Frequency:

For this application we don't have to take into account the possibility of different users trying to access because there is only one unique manager.

Open Issues:

The type of events are fixed to the existing ones. There is no way of creating a new event type.

2.2. Test case design (including expected inputs and outputs)**:**

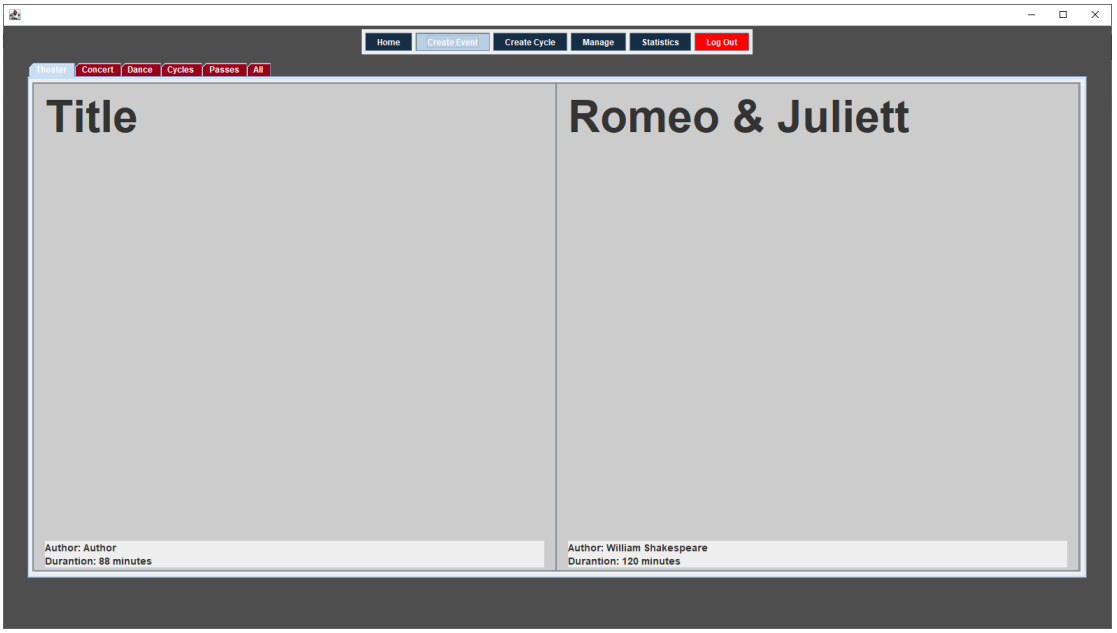
Preconditions:

The user has selected a play by searching it after having logged in.
And event with a performance and available tickets exist.

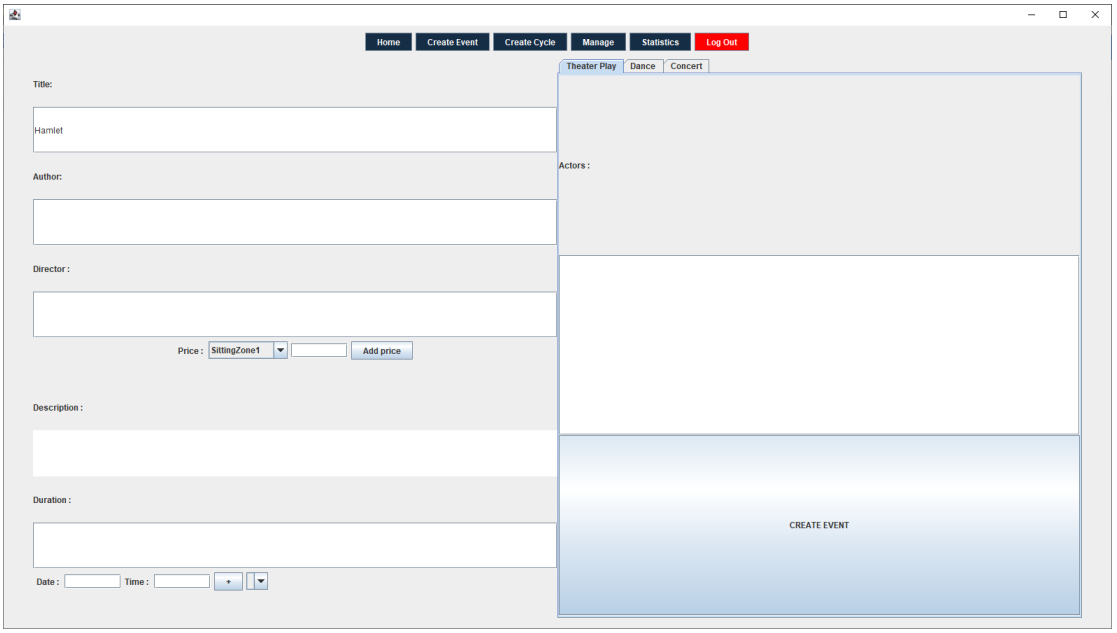
Scenario:

1. The manager selects the create button option.
2. The manager selects the Theater Play type.
3. The manager sets a title for the event.
4. The manager sets an author for the event.
5. The manager sets a Director for the event.
6. The manager sets a price for each of the zones of the event.
7. The manager sets a description for the event.
8. The manager sets the actors of the theater play.
9. The manager selects some performance dates for the event.
10. The manager clicks the create event button.

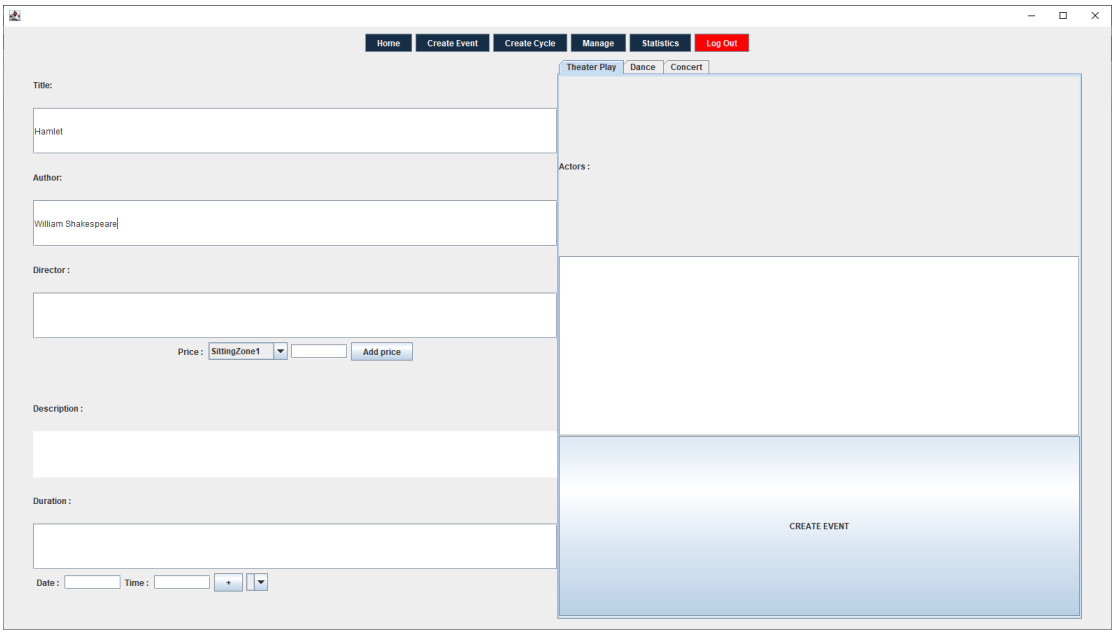
1.3. Test execution result



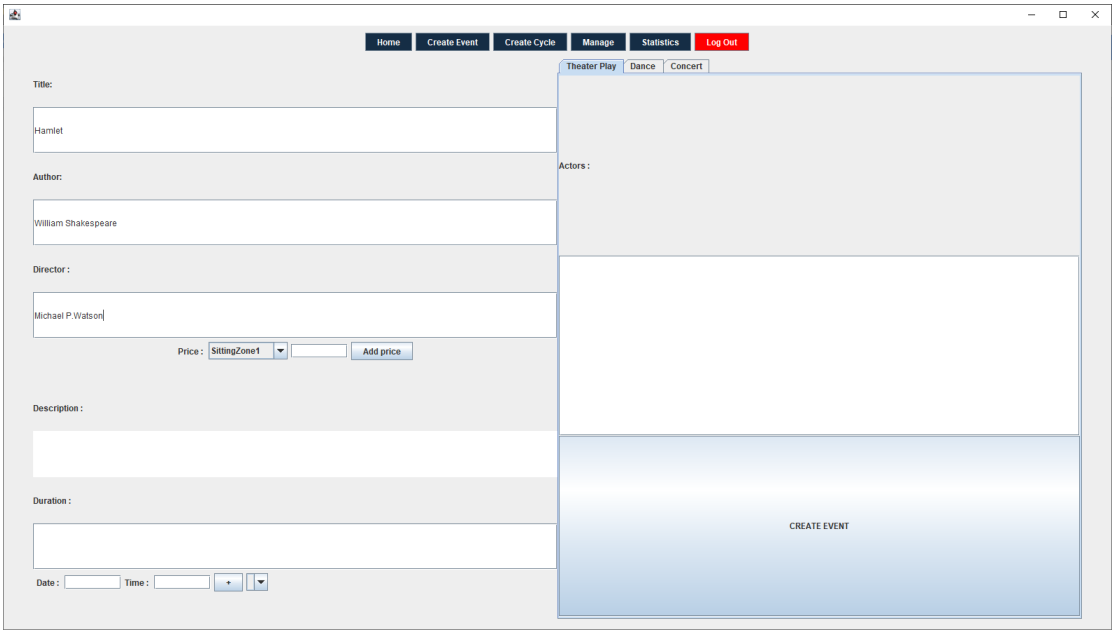
1. The manager selects the create event button. This is the same as the use case.



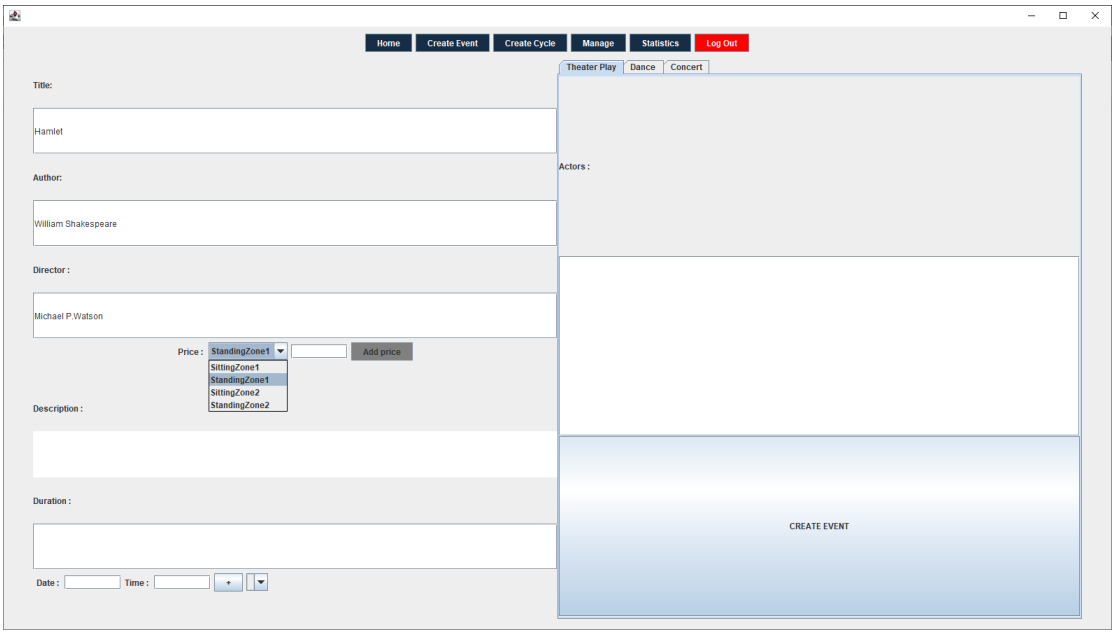
2. The manager sets a title. The order is different from the use case, but you can really insert the attributes of the event in the order you prefer.



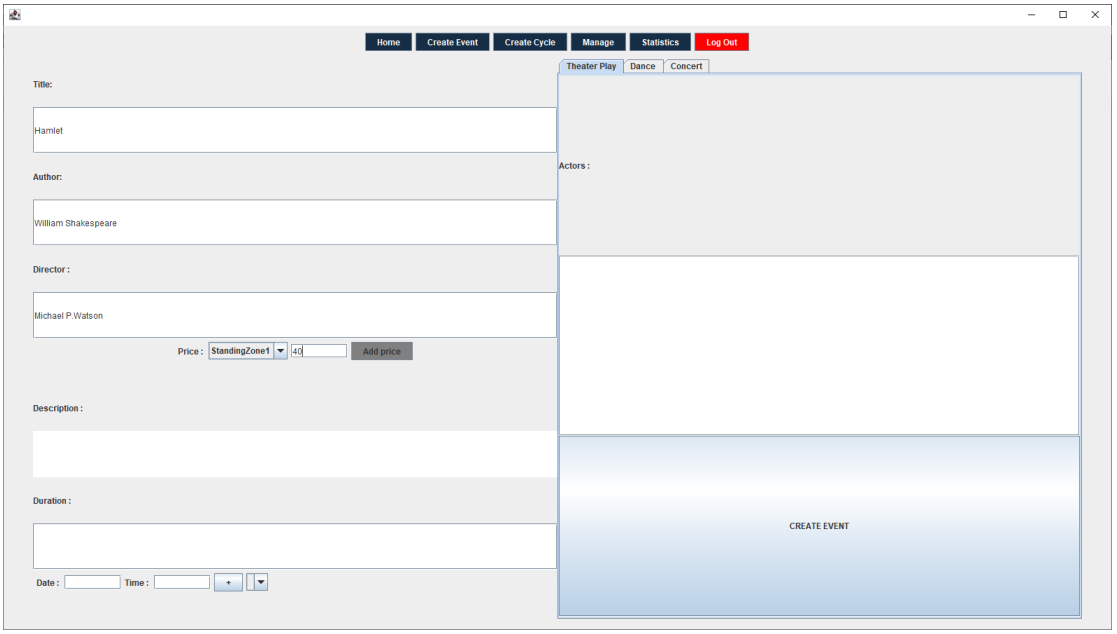
3. The manager sets an author of the event. As we said, the order is different as in the use case, but it doesn't really matter at all.



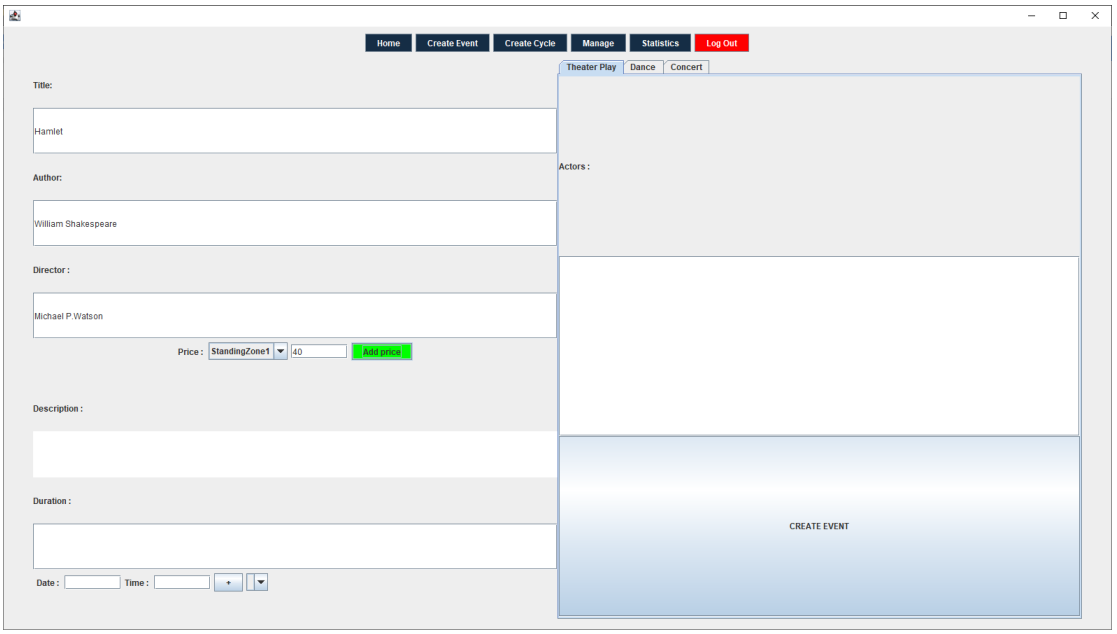
4. The manager sets the director of the event. Here the order is different too.



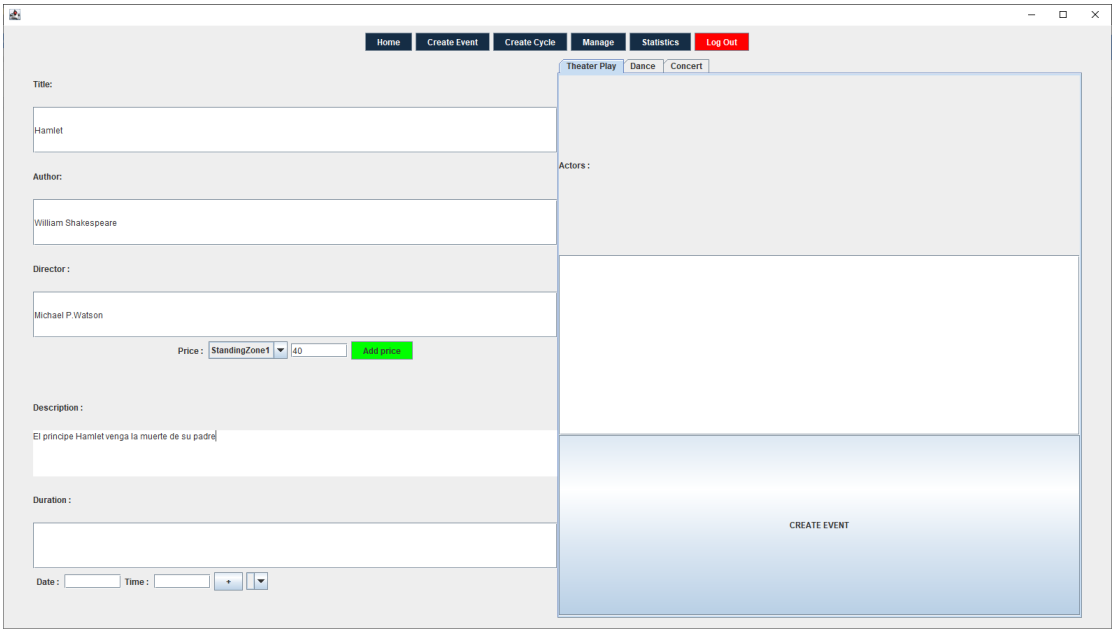
5. The manager sets a zone to add the price of the event. This is done for each zone as indicated in the use case but we do it in step 6.



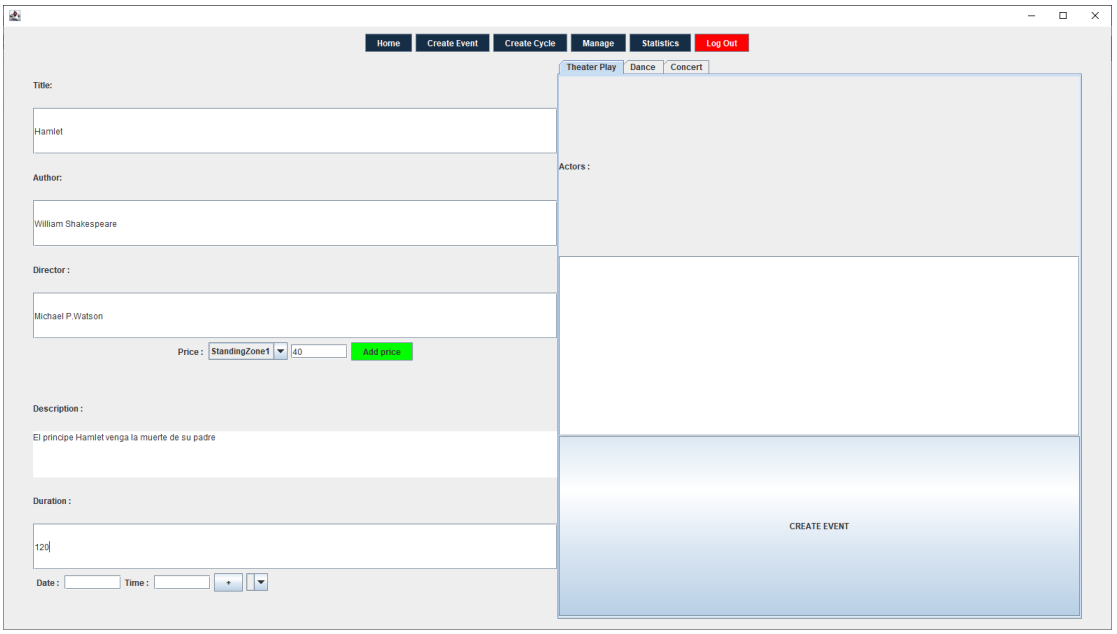
6. The manager writes the price for such zones.



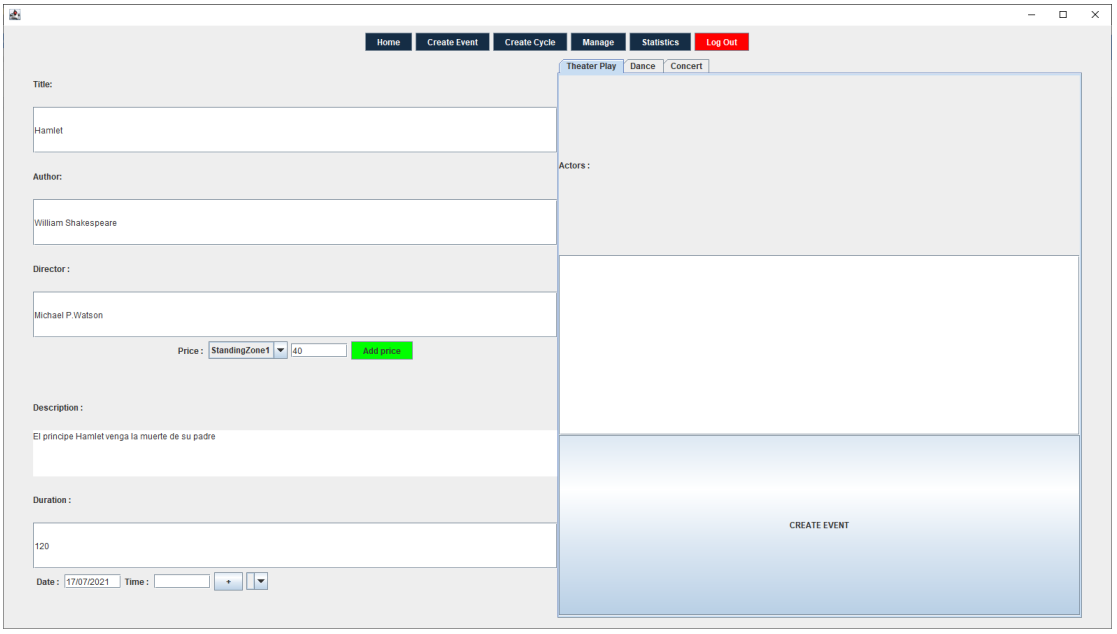
7. As we can see, the manager finally clicks the Add price button to set such price to the selected zone. If a price is already setted, the button will be shown in green indicating that a new price is setted, and if an invalid date is written the button will turn red and a message window will pop up.



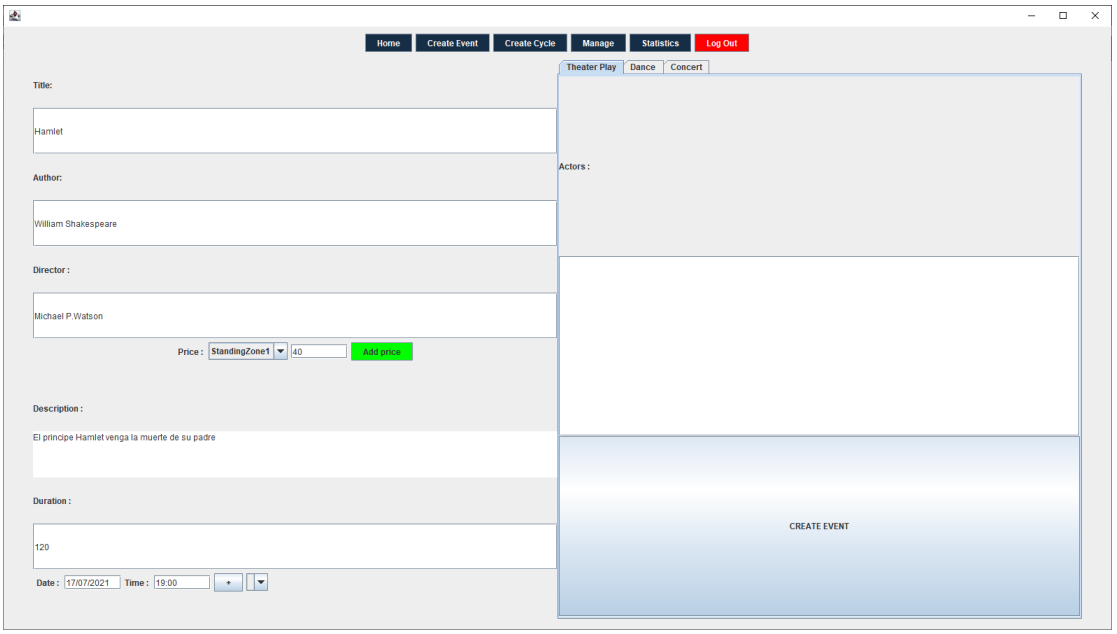
8. The manager sets a description for the event. The order here changes too.



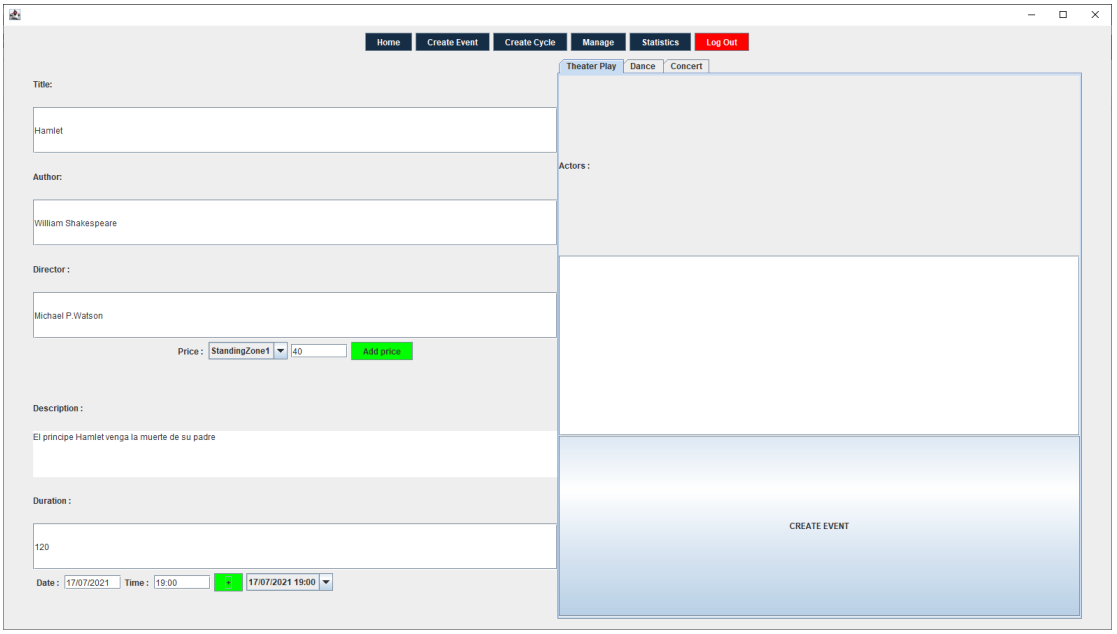
9. The manager sets a duration for the event. We didn't include this step in the use case.



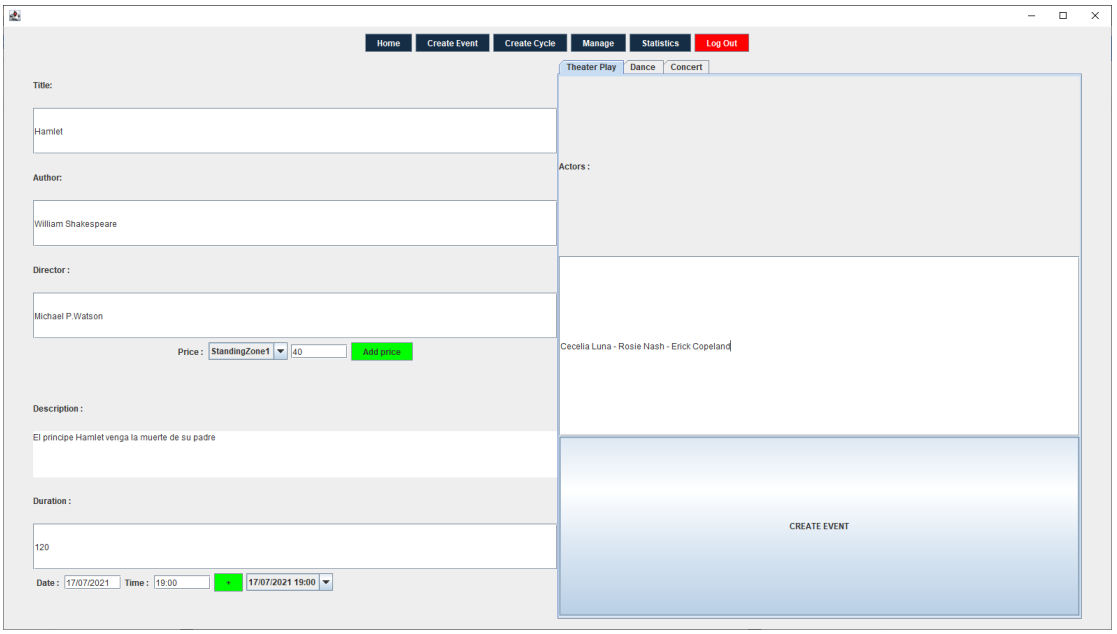
10. The user writes a date in dd/MM/yyyy format. As we can see, in the use case we don't set a date to create a performance.



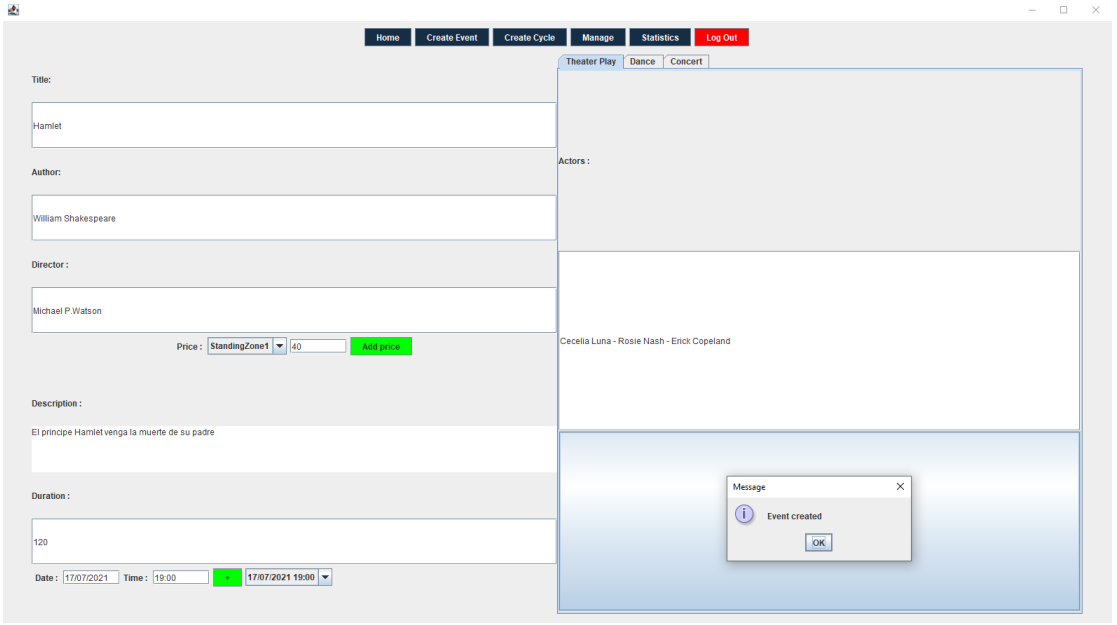
11. The user writes a time in HH:mm format. As said in point 10, in the use case we don't set a date nor time for performances of the event.



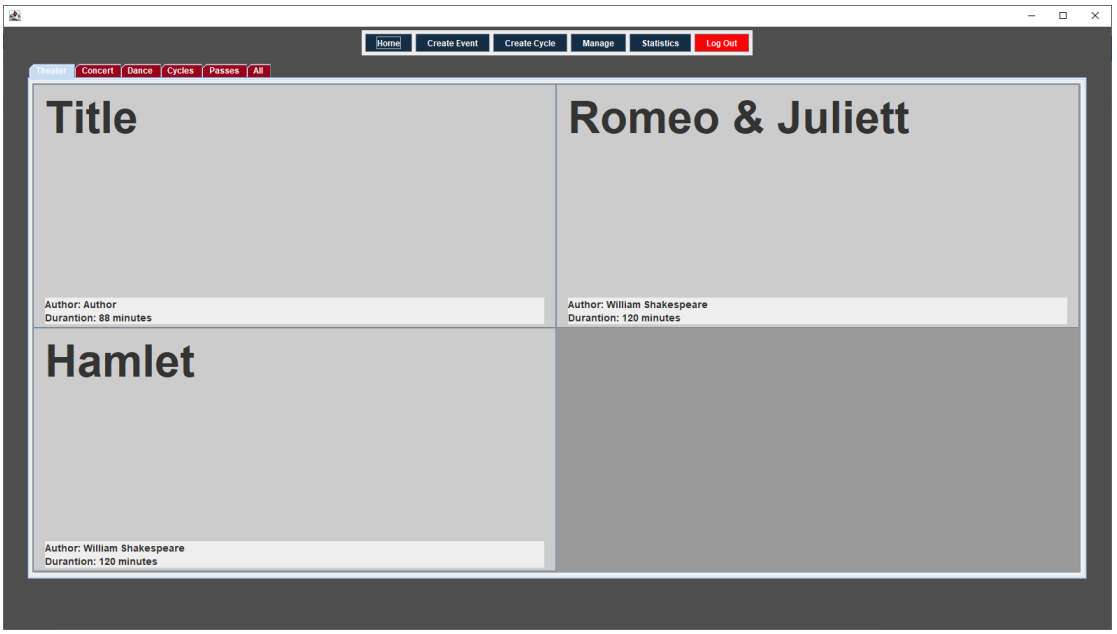
12. When clicking on the + button, if the date written is correct and future, the performance will be created for the event.



13. The user sets the authors of the play on a textfield.



14. When you click create event, it checks if all the fields are correct and if so, it creates the event. Now with the performances added, unlike the use case.



15. The app adds the event and brings you back to the main page where you can see the new theater play created.