Lab for Software Engineering

Cinema Management Application

Ifrat Jahan (3098878)
Jennifer Maxisch (3106694)
Georgios Adamos (3093306)
Thomas Klimek (3067855)
Melvin van der Linde (3106762)

November 22, 2022

University of Duisburg-Essen · Faculty of Engineering Working group Software Engineering – Prof. Dr. M. Heisel
Date: Do. 12-14- Supervisor: Marcel Schweikert

Contents

List of Figures

1 Analysis

1.1 A1

1.1.1 Requirements & Domain-Knowledge

Requirements

R1 Customers can create an account by providing an e-mail address and a password. If an e-mail address which is already associated with an account is provided, account creation fails.

R2 Customers can log in by providing their e-mail address and their password.

R3 A logged in customer can log out.

R4 A customer can browse available showings, ascendingly sorted by date.

R5 A logged in customer can book tickets by selecting the showing from the browsing list and selecting the desired seats. A showing can only be booked up to 15 minutes before it starts.

R6 Staff can add new showings to the database by providing the required data.

R7 Once a showing starts it is marked as "archived".

R8 Archived showings are visible to staff, but not to customers.

R9 Staff can cancel showings. When a show is cancelled all customers who booked tickets for it are notified via e-mail and the showing is then deleted.

R10 Showings which took place a year ago or longer are automatically removed from the database.

R11 When a showing is deleted its associated bookings are also deleted.

Facts

F1 A showing consists of the title of the movie, its duration, the date date, the hall number and unique ID.

F2 A hall consists of a number of rows, a number of seats per row and a unique hall number.

F3 Only one person at a time can sit in a seat.

Assumptions

A1 A web application is a good choice for implementing the desired functionality and all customers are able to use it.

A2 Customers only provide e-mail addresses they can access.

A3 Customers will stay up to date with the list of available showings.

A4 Every booking is paid via an external service.

A5 Staff will only add showings which take place in the future.

1.1.2 Contextdiagram

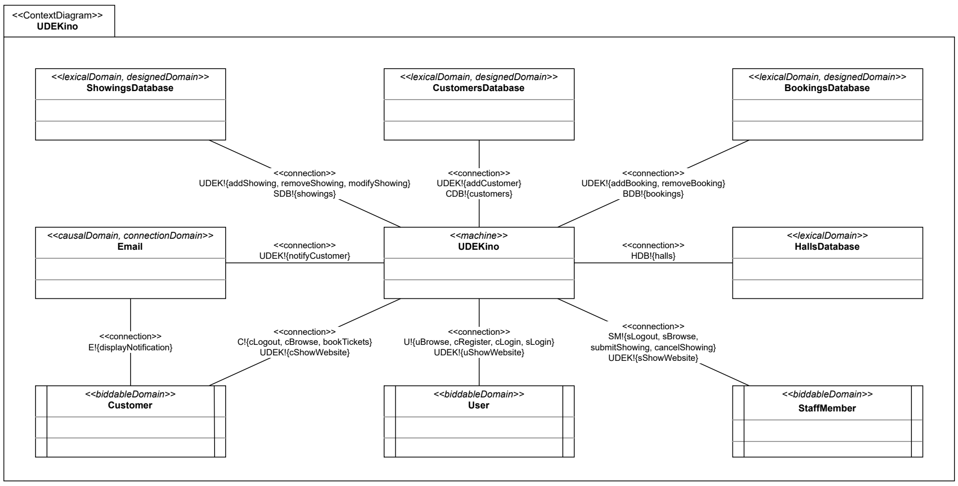

Figure 1.1: Contextdiagram

1.2 A2

We can derive the following problem diagrams

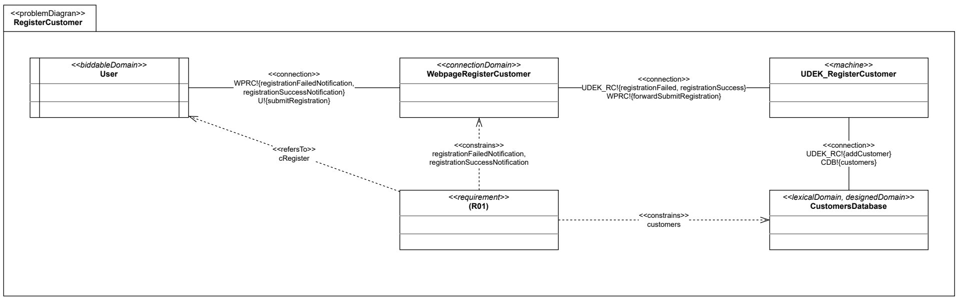

Figure 1.2: Problem diagram for R1

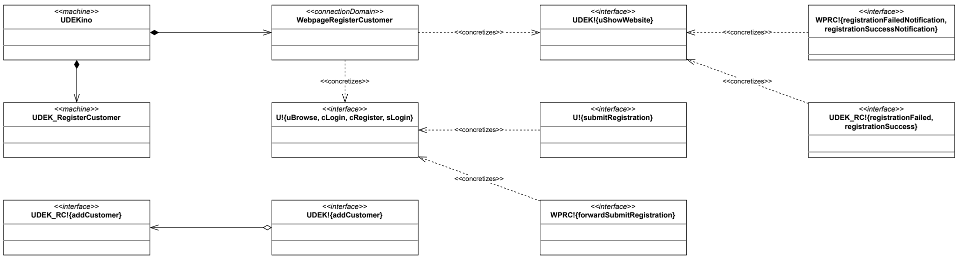

Figure 1.3: Mapping diagram for R1

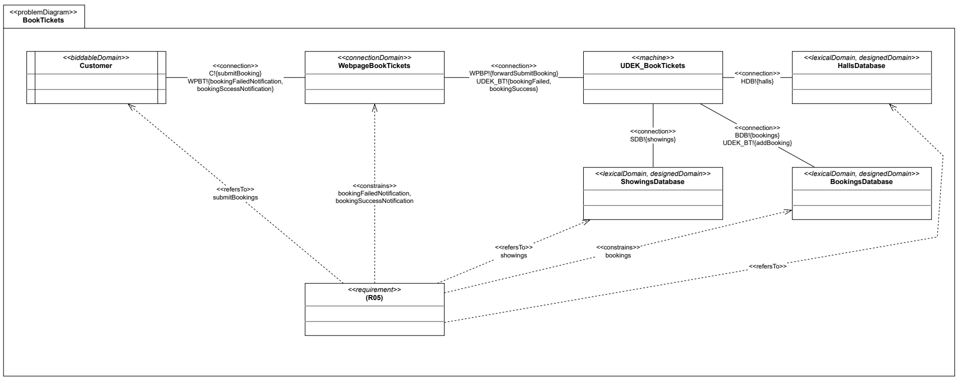

Figure 1.4: Problem diagram for R5

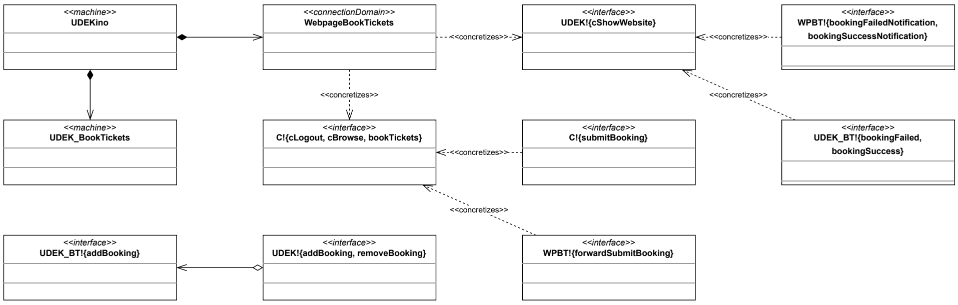

Figure 1.5: Mapping diagram for R5

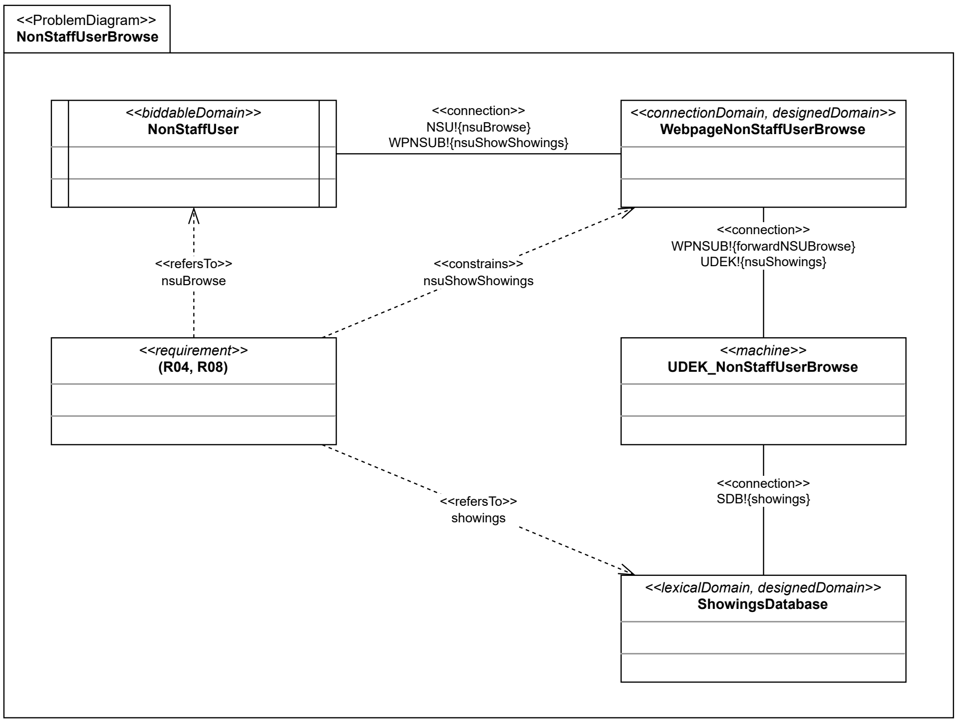

Figure 1.6: Problem diagram for R4 / R8

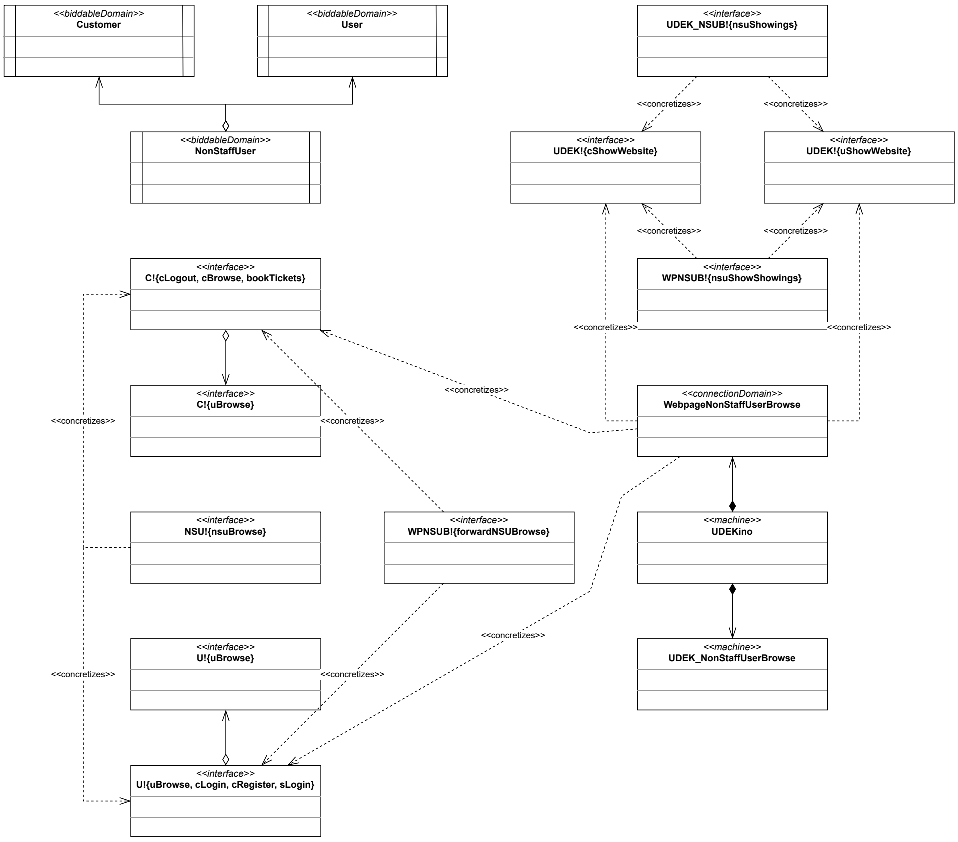

Figure 1.7: Mapping diagram for R4 / R8

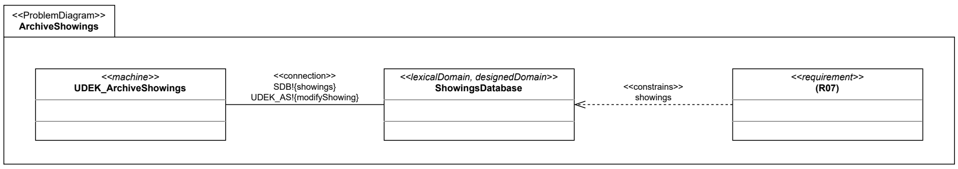

Figure 1.8: Problem diagram for R7

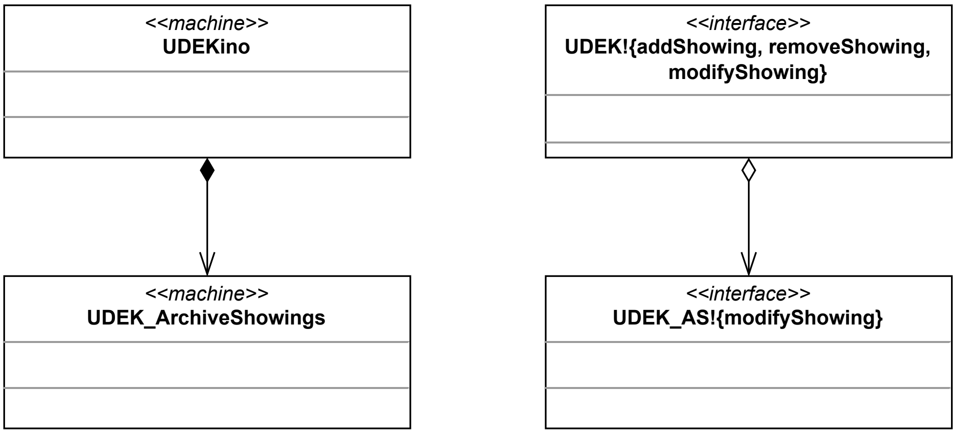

Figure 1.9: Mapping diagram for R7

Frames

- R1 fits to update 2
- R5 fits to update 2
- R4 / R8 fits to query 2
- R7 fits to simple transformation

1.3 A3

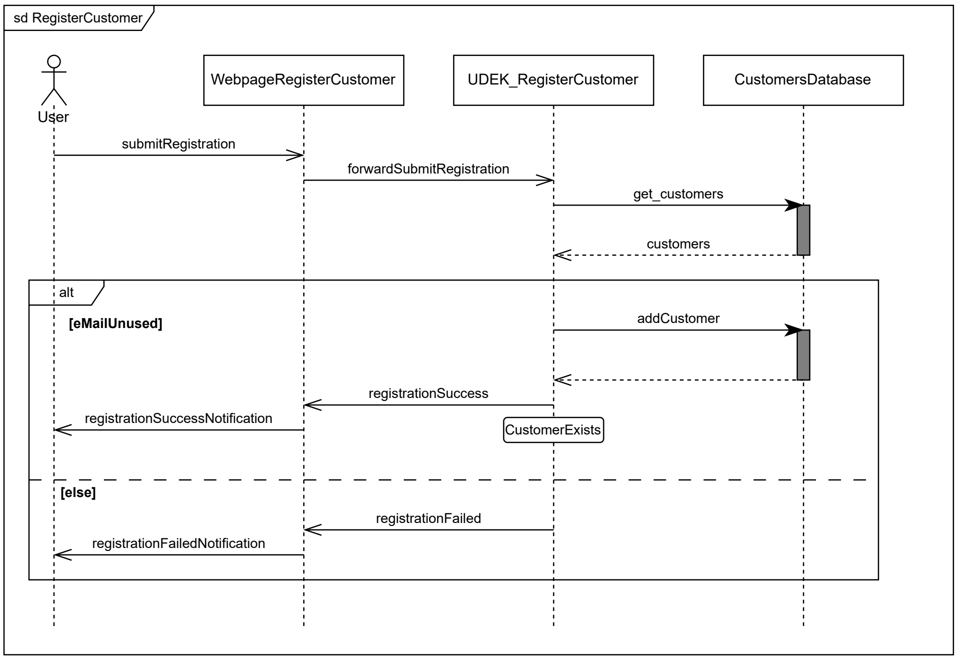

Figure 1.10: Sequence diagram for R1

S1a **WebpageRegisterCustomer**
When the WebpageRegisterCustomer recieves the command "submitRegistration", the command is forwarded to machine with "forwardSubmitRegistration". Results are recieved via commands "registrationFailed" or "registrationSuccess" and displayed to the User via "registrationFailedNotification" / "registrationSuccessNotification".

S1b **UDEK_RegisterCustomer**
When the machine receives the command "fowardSubmitRegistration" the availability of the e-mail address is checked against existing Customer accounts in the CustomersDatabase via "get_customers". If the e-mail address is available, a new Customer account is created with the data from the forwarded request and added to the CustomersDatabase via "addCustomers" and a confirmation is sent to the WebpageRegisterCustomer via "registrationSuccess". If the e-mail address is not available, account creation fails and a failure notification is sent to the WebpageRegisterCustomer via "registrationFailed".

S1c **CustomersDatabase**
When the database receives the command "get_customers", all Customer accounts are returned as the data "customers". When the database receives the command "addCustomer", the Customer account is added.

$(A2) \wedge (S1a) \wedge (S1b) \wedge (S1c) \implies (R1)$

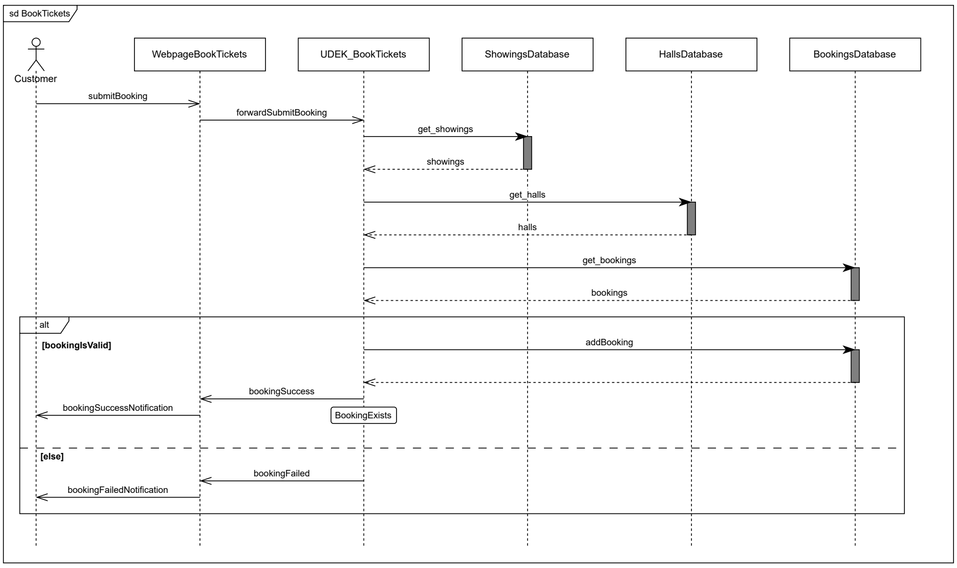

Figure 1.11: Sequence diagram for R5

S2a **WebpageBookTickets**
When the Webpage receives the command “submitBooking”, the command is forwarded to the machine with the command “forwardSubmitBooking”. Results are received via “bookingFailed” or “bookingSuccess” and displayed the the Customer via “bookingFailedNotification” / “bookingSuccessNotification”

S2b **UDEK_BookTickets** When the machine receives the command “forwardSubmitBooking”, the machine checks the availability of the desired showing and seats against the ShowingsDatabase, HallsDatabase and BookingsDatabase via “get_showings”, “get_halls” and “get_bookings”. If the desired showing and seats exist, the showing begins in more than 15 minutes and the seats are not already booked, the booking is added to the BookingsDatabase via “addBooking” and a success notification is sent to the WebpageBookTickets via “bookingSuccess”. Otherwise the booking fails and the Webpage is notified of the failure via “bookingFailed”.

S2c **ShowingsDatabase** When the database receives the command “get_showings”, all showings are returned as the data “showings”.

S2d **HallsDatabase** When the database receives the command “get_halls”, all halls are returned as the data “halls”.

S2c **BookingsDatabase** When the database receives the command “get_bookings”, all bookings are returned as the data “bookings”. When the database receives the command “addBooking”, the booking is added.

$$(F3) \wedge (S2a) \wedge (S2b) \wedge (S2c) \wedge (S2d) \wedge (S2e) \implies (R5)$$

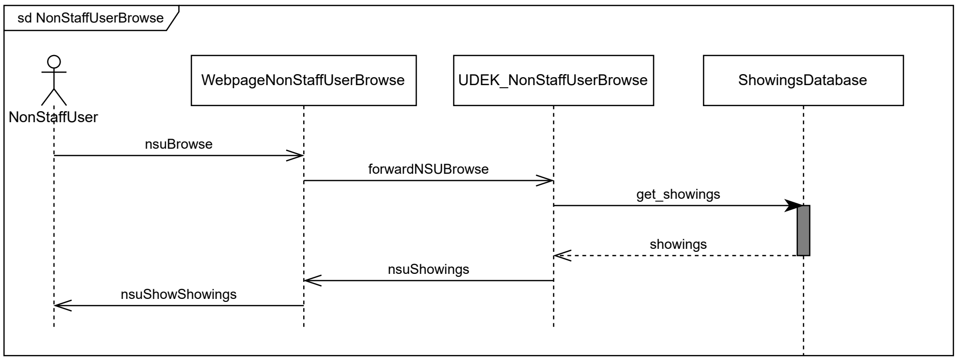

Figure 1.12: Sequence diagram for R4/R8

S3a **WebpageNonStaffUserBrowse** When the Webpage receives the command “nsuBrowse”, the command is forwarded to the machine with the command “forwardNSUBrowse”. Results are received via “nsuShowings” and displayed to NonStaffUser via “nsuShowShowings”.

S3b **UDEK_NonStaffUserBrowse** When the machine receives the command ”forwardNSUBrowse”, the machine gets all showings from the ShowingsDatabase via “get_showings”. All non-archived showings are send/transfered to the Webpage via “nsuShowings”.

S3c **ShowingsDatabase** When the database receives the command “get_showings”, all showings are returned data as “showings”

$(S3a) \land (S3b) \land (S3c) \implies (R4) \land (R8)$

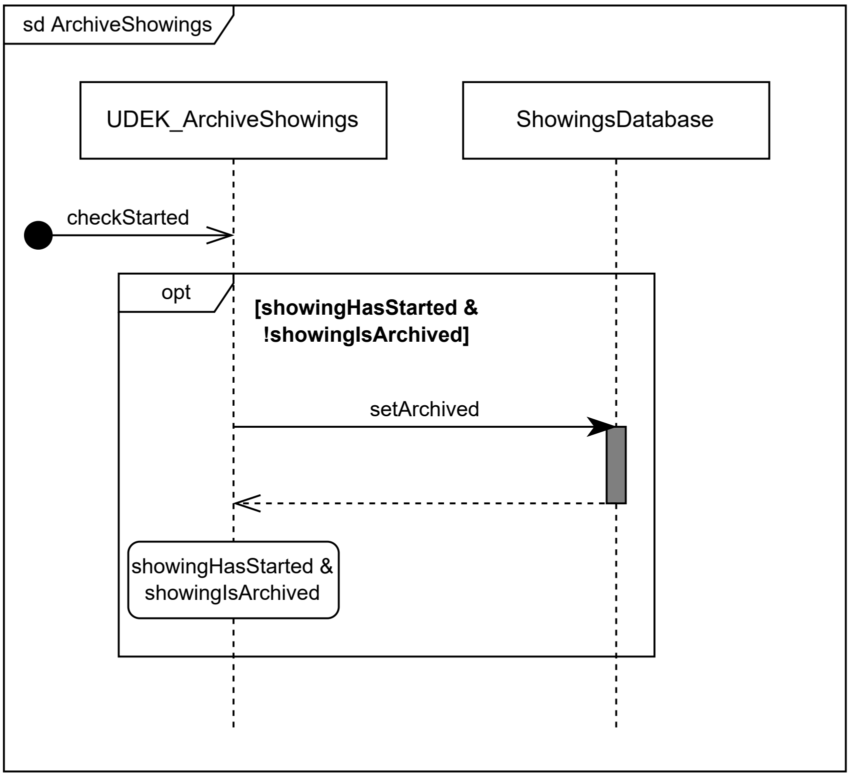

Figure 1.13: Sequence diagram for R7

S4a **UDEK_ArchiveShowings** When receiving the command “checkStarted”, all showings which have already started, and are not yet marked as archived, are marked as archived using the command “setArchived”.

S4b **ShowingsDatabase** When receiving the command “setArchived”, all showings which have already started, and are not yet marked as archived, are marked as archived.

$(S4a) \land (S4b) \implies (R7)$

1.4 A4

1.5 A5

A short OCL example:

```
context Person inv: self.alter >=0

pre alter >30
post alter=alter@pre+1
```

1.6 A6

Examples of a life-cycle using the math-environment:
$LC_{guest} = (Browse^{+}; [Book])^{*}$

2 Design

2.1 D1

2.2 D2

2.3 D3

2.4 D4

State diagrams with tikZ:

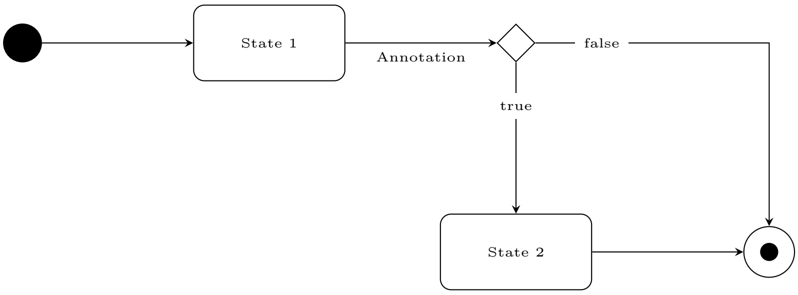

Figure 2.1: Zustandsdiagramm Person 1

3 Implementation & Testing

3.1 I

3.2 T1

3.3 T2

3.4 T3

4 Glossary

Table 4.1: Glossary

Name	Type	Description	Source
A			
addBooking	phenomenon	the machine adds a new booking to the bookings database	CD
addBooking	message	contains a showing ID and seats	SD R5
addCustomer	phenomenon	the machine adds a new customer to the customers database	CD
addCustomer	message	contains an e-mail address and a password	SD R1
addShowing	phenomenon	the machine adds a new showing to the customers database	CD
B			
BookingExists	state predicate	given booking exists within the BookingsDatabase	SD R5
bookingFailed	phenomenon	the machine notifies the webpage that a booking has failed	PD R5
bookingFailed	message	informs the WebpageBookTickets that the booking failed	SD R5
bookingFailedNotification	phenomenon	the webpage displays a notification to the customer that a booking has failed	PD R5
bookingFailedNotification	message	informs the user that the booking failed	SD R5
bookingIsValid	guard	showing with ID contained in request exists and starts in more than 15 minutes and the seats contained in the request exist in the showing's hall and are not already booked	SD R5
bookings	message	all bookings in the BookingsDatabase	PD R5
bookingSuccess	phenomenon	the machine notifies the webpage that a booking has succeeded	PD R5
bookingSuccess	message	informs the WebpageBookTickets that the booking was successful	SD R5
bookingSuccessNotification	phenomenon	the webpage displays a notification to the customer that a booking has succeeded	PD R5
bookingSuccessNotification	message	informs the Customer that the booking was successful	SD R5
BookingsDatabase	lexical domain, designed domain	a database containing the bookings made by customers	CD

Table 4.1: Glossary

Name	Type	Description	Source
BookingsDatabase	object	the database containing all bookings	SD R5
bookTickets	phenomenon	a customer books tickets for a showing	CD
C			
cBrowse	phenomenon	a customer browses available showings	CD
checkStarted	found message	a prompt for the UDEK_ArchiveShowings machine to mark all showings which have already started and are not marked as archived, as archived	SD R7
cLogin	phenomenon	a user attempts to log into a customer account	CD
cLogout	phenomenon	a customer attempts to log out	CD
cRegister	phenomenon	a user attempts to create customer account on UDEKino	CD
cShowWebsite	phenomenon	the machine shows a website to the customer	CD
Customer	biddable domain	a customer of UDEKino; a user who has logged into a customer account	CD
Customer	actor	a customer who wishes to book tickets	SD R5
CustomerExists	state predicate	the customer account with the given e-mail address and password exists within the CustomersDatabase	SD R1
customers	message	all customers in the CustomersDatabase	SD R1
CustomersDatabase	lexical domain, designed domain	a database containing customer data	CD
CustomersDatabase	object	the database of customer acccounts	SD R1
D			
displayNotification	phenomenon	the customer's e-mail client displays a notification e-mail to the customer	CD
E			
Email	causal domain, connection domain	an e-mail service offering to deliver e-mails	CD
eMailUnused	guard	the e-mail contained in the registration request is not contained in customers	SD R1
F			
forwardNSUBrowse	phenomenon	the website sends a request for a list of upcoming showings to the machine	PD R4 / R8

Table 4.1: Glossary

Name	**Type**	**Description**	**Source**
forwardNSUBrowse	message	a request for the machine to send a list of available, i.e., non-archived, showings	SD R4/8
forwardSubmitBooking	phenomenon	the webpage forwards a request to book tickets to the machine	PD R5
forwardSubmitBooking	message	contains the showing ID and the desired seats	SD R5
forwardSubmitRegistration	phenomenon	the webpage forwards a request to register a customer account to the machine	PD R1
forwardSubmitRegistration	message	a request from the WebpageRegisterCustomer to register a new customer account, containing an e-mail address and a password	SD R1
G			
get_bookings	message	contains all messages in the BookingsDatabase	SD R5
get_customers	message	returns all customer accounts in the CustomersDatabase	SD R1
get_halls	message	returns all halls in the HallsDatabase	SD R5
get_showings	message	returns all showings in the ShowingsDatabase	SD R5, 4/8, 7
H			
halls	message	all halls in the HallsDatabase	SD R5
HallsDatabase	lexical domain	a database containing the cinema halls, provided by the cinema operator	CD
HallsDatabase	object	the database containing the cinema halls	SD R5
I			
J			
K			
L			
M			
modifyShowing	phenomenon	the machine modifies a showing in the showings database	CD
N			
NonStaffUser	biddable domain	either of Customer or User	PD R4 / R8
NonStaffUser	actor	a user who is not logged in as staff and wishes to browse available showings	SD R4/8
notifyCustomer	phenomenon	the machine notifies the customer via e-mail	CD
nsuBrowse	phenomenon	either of cBrowse or uBrowse	PD R4 / R8

Table 4.1: Glossary

Name	Type	Description	Source
nsuBrowse	message	a request for the WebpageNonStaffUserBrowse to display available showings	SD R4/8
nsuShowings	phenomenon	the machine sends a list of upcoming showings to be displayed by the website	PD R4 / R8
nsuShowings	message	contains available, i.e., non-archived, showings	SD R4/8
nsuShowShowings	phenomenon	the website displays a list of upcoming showings to the user	PD R4 / R8
nsuShowShowings	message	a rendition of available, i.e., non-archived, showings	SD R4/8
O			
P			
bookings	phenomenon	the bookings database provides the bookings data to the machine	CD
customers	phenomenon	the customers database provides the customers data to the machine	CD
halls	phenomenon	the halls database provides the halls data to the machine	CD
showings	phenomenon	the showings database provides the showings data to the machine	CD
Q			
R			
registrationFailed	phenomenon	the machine notifies the webpage that the registration has failed	PD R1
registrationFailed	message	informs the WebpageRegisterCustomer that account creation has failed	SD R1
registrationFailedNotification	phenomenon	the webpage displays a to the user that the registration has failed	PD R1
registrationFailedNotification	message	informs the user that account creation has succeeded	SD R1
registrationSuccess	phenomenon	the machine notifies the webpage that the registration has succeeded	PD R1
registrationSuccess	message	informs the WebpageRegisterCustomer that account registration has succeeded	SD R1
registrationSuccessNotification	phenomenon	the webpage displays a notification to the user that the registration has succeeded	PD R1
registrationSuccessNotification	message	informs the User that account creation has succeeded	SD R1
removeBooking	phenomenon	the machine removes a booking from the bookings database	CD

Table 4.1: Glossary

Name	**Type**	**Description**	**Source**
removeCustomer	phenomenon	the machine removes a customer from the customers database	CD
removeShowing	phenomenon	the machine removes a showing from the showings database	CD
S			
sBrowse	phenomenon	a staff member browses available showings	CD
sCancelShowing	phenomenon	a staff member attempts to cancel a showing	CD
setArchived	message	contains the ID of the showing which is to be marked as archived	SD R7
ShowingHasStarted	guard / state predicate	whether the showing in question has already started, i.e., its starting date and time lies in the past	SD R7
ShowingIsArchived	guard / state predicate	whether the showing in question is marked as archived"	SD R7
showings	message	contains all showings in the ShowingsDatabase	SD R5, 4/8, 7
ShowingsDatabase	lexical domain, designed domain	a database containing the cinema showings	CD
ShowingsDatabase	object	the database containing the showings	SD R5, 4/8, 7
sLogin	phenomenon	a user attempts to log in as a staff member	CD
sLogout	phenomenon	a staff member attempts to log out	CD
sShowWebsite	phenomenon	the machine shows a website to the staff member	CD
StaffMember	biddable domain	a member of cinema staff; a user who has logged in as staff	CD
submitBooking	phenomenon	the customer selects the tickets they wish to book and hits the submit button	PD R5
submitBookign	message	contains the showing ID and desired seats	SD R5
submitRegistration	phenomenon	the user submits a request to register a new customer account, containing an e-mail address and a password	PD R1
submitRegistration	message	a request from the user to register a new customer account, containing an e-mail address and a password	SD R1
submitShowing	phenomenon	a staff member submits a new showing to the machine for entry into the database	CD
T			
U			
uBrowse	phenomenon	a user browses available showings	CD

Table 4.1: Glossary

Name	**Type**	**Description**	**Source**
UDEKino	machine	the machine to be developed	CD
UDEK_ArchiveShowings	machine	the sub-machine responsible for automatically archiving showings once they have begun	PD R7
UDEK_ArchiveShowings	object	the sub-machine responsible for archiving showings which have already started	SD R7
UDEK_BookTickets	machine	the sub-machine responsible for customer booking tickets	PD R5
UDEK_BookTickets	object	the machine responsible for the booking of tickets	SD R5
UDEK_NonStaffUserBrowse	machine	the sub-machine responsible for registered and non-registered customers browsing upcoming showings	PD R4 / R8
UDEK_RegisterCustomer	machine	the sub-machine responsible for customer account registration	PD R1
UDEK_RegisterCustomer	object	the machine responsible for customer account registration	SD R1
User	biddable domain	a user of the application who is not logged in	CD
User	actor	a user of who wishes to register a new customer account	SD R??
uShowWebsite	phenomenon	the machine shows a website to the user	CD
V			
W			
WebpageBookTickets	connection domain, designed domain	a webpage via which a customer can book tickets	PD R5
WebpageBookTickets	object	the webpage for booking tickets	SD R5
WebpageNonStaffUserBrowse	connection domain, designed domain	a webpage via which a user can browse upcoming showings	PD R4 / R8
WebpageNonStaffUserBrowse	object	the webpage for NonStaffUsers to browse available showings	SD R4/8
WebpageRegisterCustomer	connection domain, designed domain	a webpage via which a user can register a new customer account	PD R1
WebpageRegisterCustomer	object	the webpage for registering a new customer account	SD R1
X			
Y			
Z			