Examples - Simple Music Portal

The following example shows a very simple music portal web application that allows users to buy albums which then can be downloaded as archive files containing MP3s. The following list gives a short informal description of the use cases and requirements.

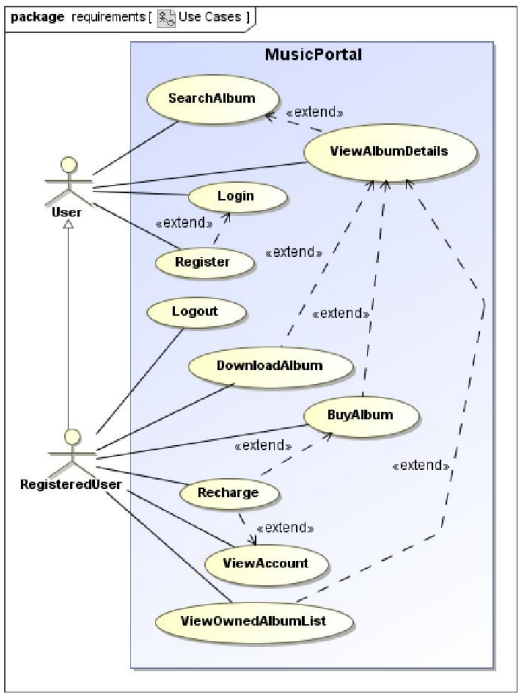
- A distinction is made between users and registered users. Only registered users can buy or download albums. A user becomes a registered user by logging in. Unregistered users can register with a username that has not been taken by another user and a freely chosen password.
- Every user can search for albums by their name. Other search methods are not offered. The search result is presented as a list of matching albums that provides links to a detail page for each album. The album detail pages show the title of the album, the name of the artist, the list of songs and the album's price. If the user has already bought the album then a download link is shown. Otherwise, there will be a link for buying the album.
- Only full albums can be downloaded.
- In this simplified example, each album has only one artist. This restriction is done to reduce the complexity of the navigation and presentation models. It would be easy though to add support for multiple artists by adding an index in the navigation model and an anchored collection in the presentation model, respectively.
- Each registered user has a credit account that is used to buy albums. The credit account can be recharged by credit card payment. To do this the user has to enter her credit card data and the amount to recharge with. This data is validated and the user has to confirm the transaction before the credit card is charged and the user's credit account is recharged. The details of credit card handling are not modelled in this example.
- If a user is logged in, she can navigate to an account page that shows the user's credits and the list of albums she has bought in the past.
- The links for logging in or out, for registering and to the user's account page are always shown. This also holds for the album search box.

UWE models

This example was modelled with UWE Profile - v1.9 defined in the Magic Draw 16.8 CASE tool and is available as mdzip and emf.

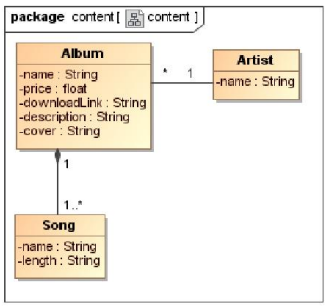
Requirements Analysis

The requirements in UWE are modelled by a use case diagram. They form a rough version of the functions described above.



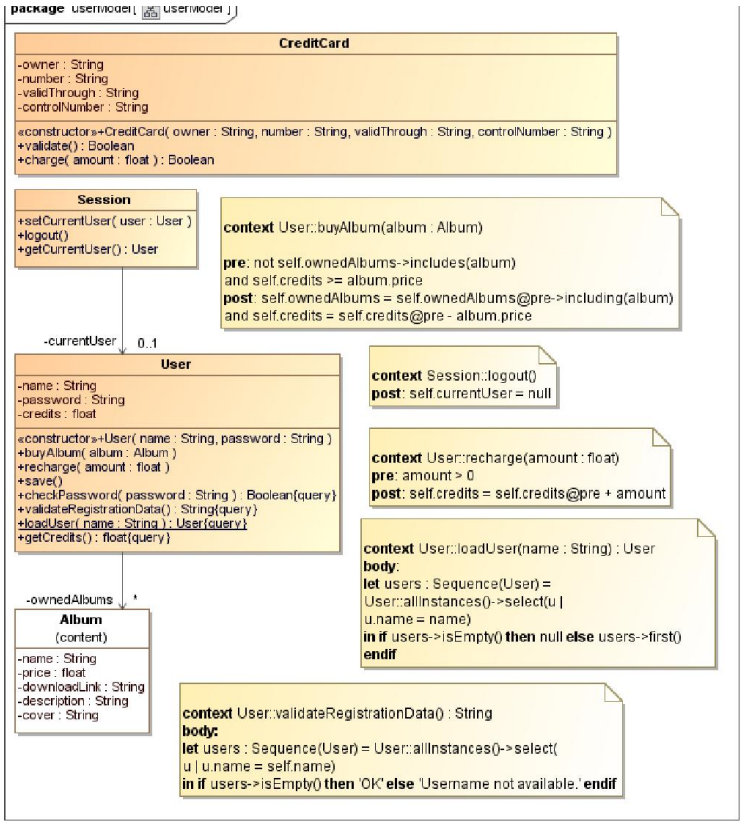
Content

The content model visualizes the domain relevant information for the Web system that mainly comprises the content of the Web application. In our example the information is provided by the classes Album, Artist and Song. A UML class diagram and UML plain classes are used to model the content.



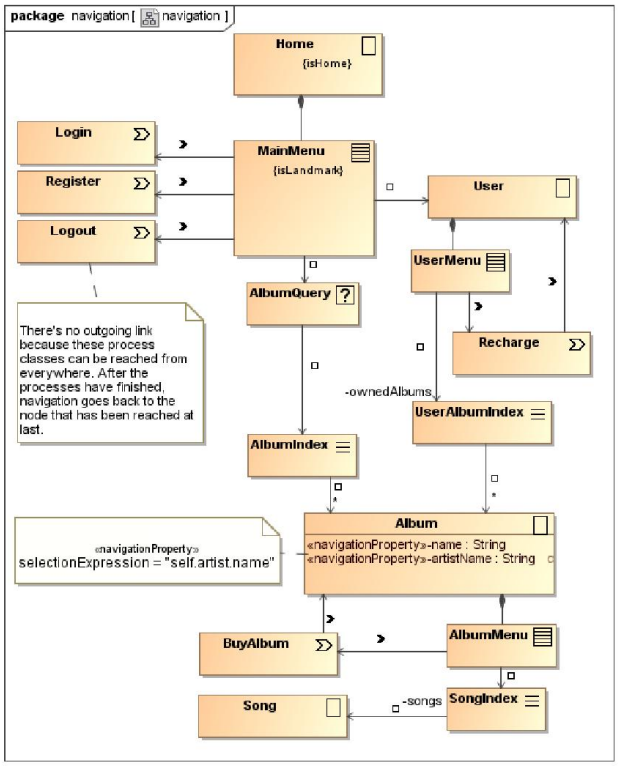
User Model

While the content model defines the data content of the application, the user model serves two different purposes. On the one hand, it contains classes that define what information is stored in the context of a session. In this example, we see that a session can have one current user who can have a collection of owned albums. On the other hand, the classes in the user model provide operations that can be used in the business processes.



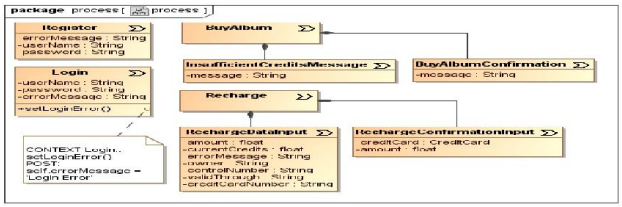
Navigation

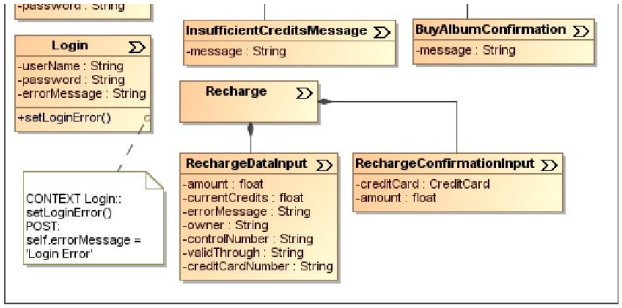
The navigation is strongly simplified and is mainly intended to demonstrate the use of the model elements than to show a realistic example.

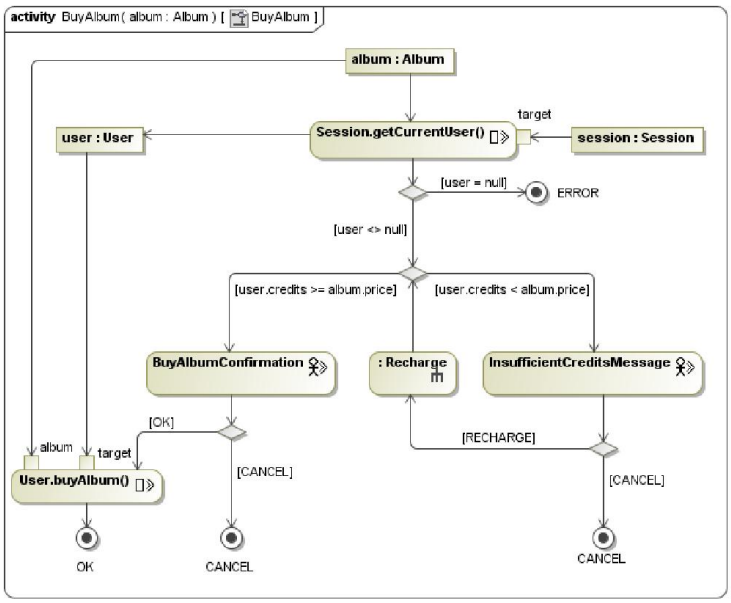


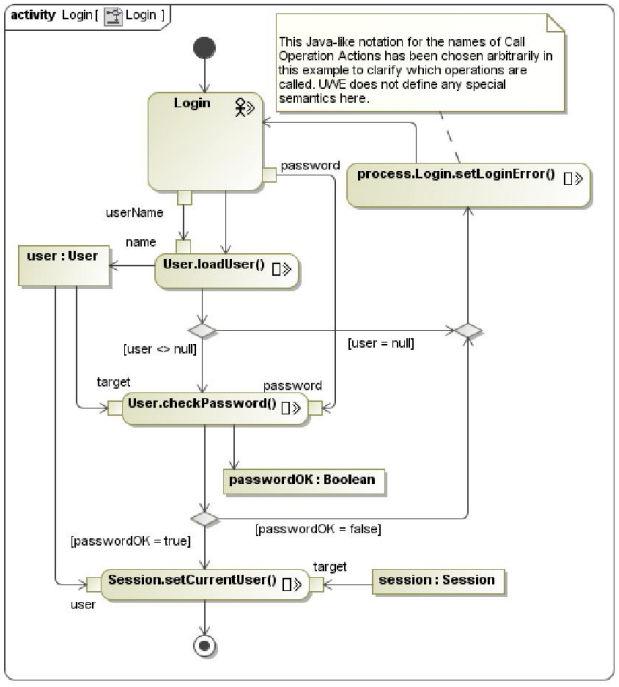
Processes

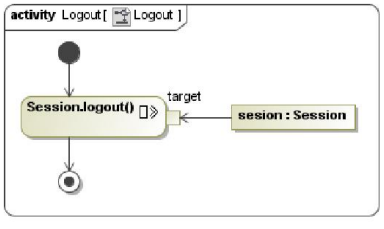
The example's process model consists of the process classes that are integrated in the navigation model, additional process classes to handle user input and activities that define the behaviour of the processes. The process model shows the relationship of main process class and process classes, which have associated the process flow.

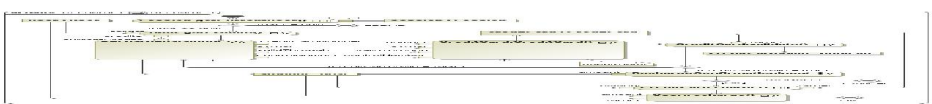


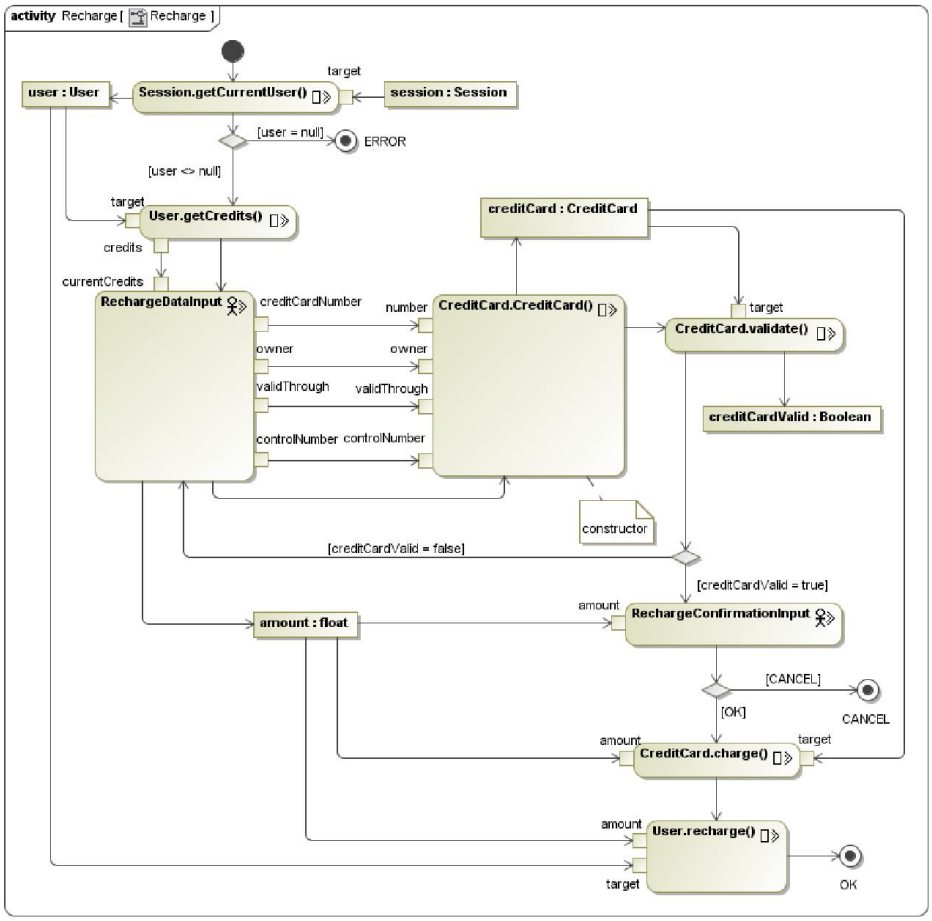

Login
-userName : String
-password : String
-errorMessage : String
+setLoginError()
CONTEXT Login::
setLoginError()
POST:
self.errorMessage =
'Login Error'
InsufficientCreditsMessage
-message : String
BuyAlbumConfirmation
-message : String
Recharge
RechargeDataInput
-amount : float
-currentCredits : float
-errorMessage : String
-owner : String
-controlNumber : String
-validThrough : String
-creditCardNumber : String
RechargeConfirmationInput
-creditCard : CreditCard
-amount : float

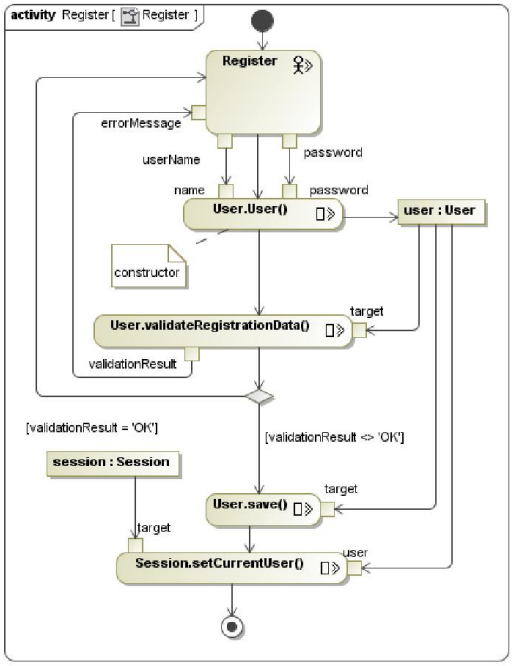

activity BuyAlbum(album : Album) [BuyAlbum]
album : Album
user : User
Session.getCurrentUser()
target
session : Session
[user = null]
ERROR
[user <> null]
[user.credits >= album.price]
[user.credits < album.price]
BuyAlbumConfirmation
: Recharge
InsufficientCreditsMessage
[OK]
[CANCEL]
[RECHARGE]
[CANCEL]
album
target
User.buyAlbum()
OK
CANCEL
CANCEL


activity Login [Login]
This Java-like notation for the names of Call Operation Actions has been chosen arbitrarily in this example to clarify which operations are called. UWE does not define any special semantics here.
Login
password
process.Login.setLoginError()
userName
name
user : User
User.loadUser()
[user <> null]
[user = null]
target
password
User.checkPassword()
passwordOK : Boolean
[passwordOK = false]
[passwordOK = true]
target
Session.setCurrentUser()
session : Session
user


activity Logout [Logout]
target
Session.logout()
sesion : Session

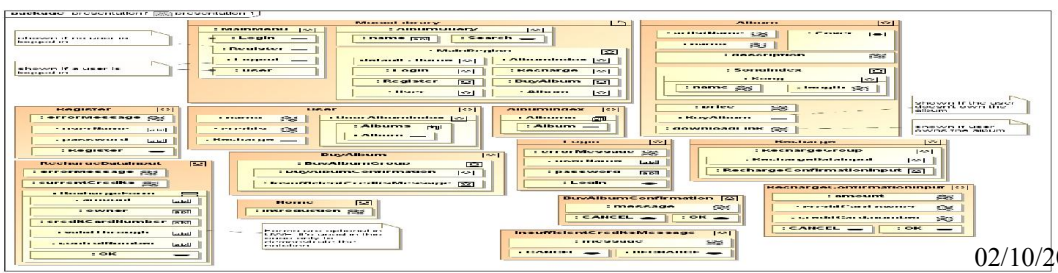


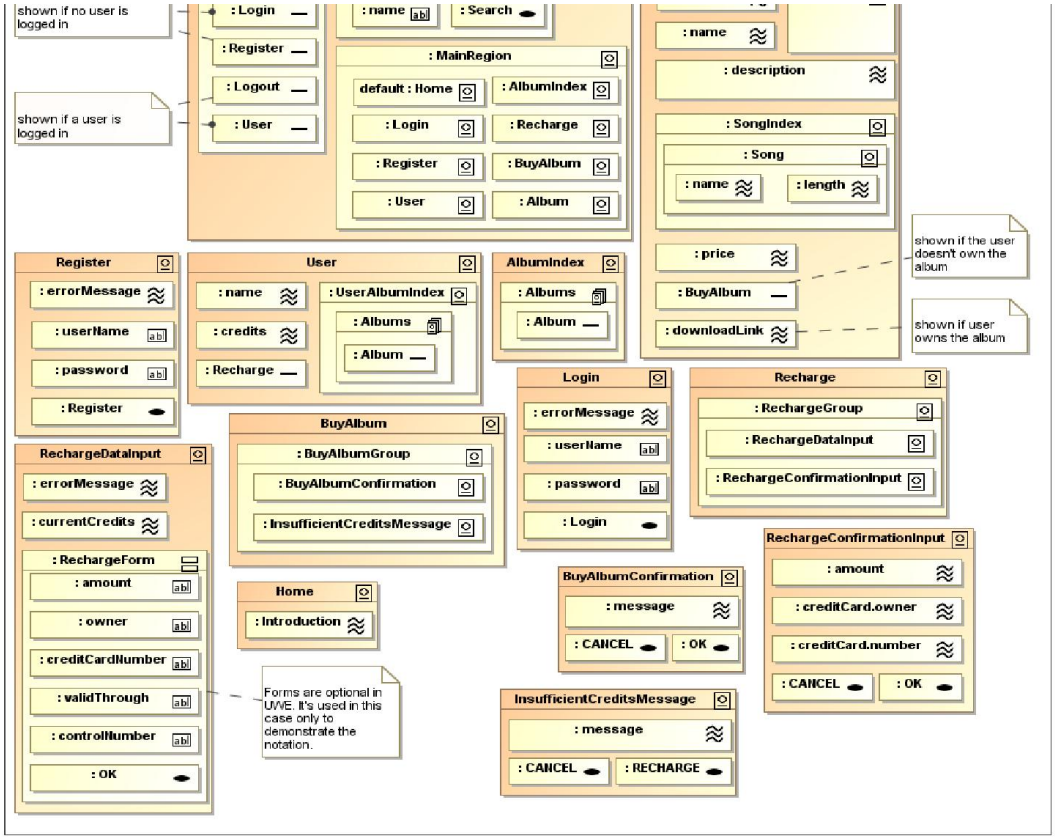




Presentation

The presentation model of the example is shown as a UML composite structure diagram. In this kind of diagram, properties that are contained by composition are shown as rectangles that are contained in the figure of the containing class. For example the presentation class MainMenu could be shown in the following two equivalent ways.




shown if no user is logged in
: Login
: Register
shown if a user is logged in
: Logout
: User
: name
: Search
: MainRegion
default : Home
: AlbumIndex
: Login
: Recharge
: Register
: BuyAlbum
: User
: Album
: name
: description
: SongIndex
: Song
: name
: length
: price
: BuyAlbum
: downloadLink
shown if the user doesn't own the album
shown if user owns the album
Register
: errorMessage
: userName
: password
: Register
User
: name
: credits
: Recharge
: UserAlbumIndex
: Albums
: Album
AlbumIndex
: Albums
: Album
Login
: errorMessage
: userName
: password
: Login
Recharge
: RechargeGroup
: RechargeDataInput
: RechargeConfirmationInput
BuyAlbum
: BuyAlbumGroup
: BuyAlbumConfirmation
: InsufficientCreditsMessage
RechargeDataInput
: errorMessage
: currentCredits
: RechargeForm
: amount
: owner
: creditCardNumber
: validThrough
: controlNumber
: OK
Home
: Introduction
Forms are optional in UWE. It's used in this case only to demonstrate the notation.
BuyAlbumConfirmation
: message
: CANCEL
: OK
RechargeConfirmationInput
: amount
: creditCard.owner
: creditCard.number
: CANCEL
: OK
InsufficientCreditsMessage
: message
: CANCEL
: RECHARGE