

# Use Cases: *Basic interface methods*

David Bialik, Julian Fleischer, Hagen Mahnke,  
Konrad Reiche, André Zoufahl

March 4, 2011

	<b>User logs in</b>
Actors	Any user
Scope	GUI
Precondition	The user is not logged in, thus known to the system as an anonymous user.
Postcondition	-
Postcondition on Success	The user is logged in and a welcome-page is shown. From that point on, every page informs the user about his current login status (i.e. that she is logged in) and displays his user name.
Basic Course of Events	<ol style="list-style-type: none"><li>1. The user navigates to the login-form.</li><li>2. The user enters his login-credentials, i.e. his username (or email-address) and his password</li><li>3. The user submits the login-form.</li></ol>
Alternative Paths	-
Open Questions	-
Solved issues	-
Implementation Notes	<ul style="list-style-type: none"><li>• A login form may be present on all pages, whilst not logged in. It may for example be reachable via a menu.</li></ul>
Implementation Status	<ul style="list-style-type: none"><li>• Implemented.</li></ul>

	<b>User edits his own password</b>
Actors	Any user
Scope	GUI
Precondition	The user is logged in.
Postcondition	When the action is performed the user is presented another page which informs him about the success of the change of this password. The user is still logged in.
Postcondition on Success	The password is stored in the database, such that on next login only the new password will be accepted.
Basic Course of Events	<ol style="list-style-type: none"> <li>1. The user opens his personal preferences (a link should be available in the main navigation).</li> <li>2. Within the preferences there is a single form, that asks the user for his current password and the new password (which should be entered twice).</li> <li>3. The user submits the form.</li> </ol>
Alternative Paths	-
Open Questions	-
Solved issues	-
Implementation Notes	-
Implementation Status	-

  

	<b>User edits his profile</b>
Actors	Any user
Scope	
Precondition	
Postcondition	
Postcondition on Success	
Basic Course of Events	-
Alternative Paths	-
Open Questions	-
Solved issues	<p><b>Q:</b> Some attributes (like "Matrikel-Number") may not be edited by a user himself. Do we have a mechanism for that?</p> <p><b>A:</b> No, we do not. The data model should be updated by an attribute "user-editable".</p>
Implementation Notes	-
Implementation Status	-

	<b>User displays the lectures he currently is enrolled in.</b>
Actors	Any registered user
Scope	
Precondition	The user is logged in.
Postcondition	Anyway a nicely formatted page is shown (on failure, an error explaining what went wrong).
Postcondition on Success	The page is displayed. No state is changed.
Basic Course of Events	<ol style="list-style-type: none"> <li>1. The user clicks on a link, e.g. "my lectures"</li> <li>2. A list of the lectures the user currently is enrolled in is to be displayed. The user should be able to click on them for a detail view (see appropriate use case). People having special rights should special options they are allowed to do with these entities.</li> </ol>
Alternative Paths	-
Open Questions	<b>Q:</b> Do we have a "leave" mechanism? If so, a "leave" link should also be displayed.
Solved issues	<b>Q:</b> Do we have the information about what is a running Academic Term? <b>A:</b> Yes, the information can be deduced from the start/end-times and the current date.
Implementation Notes	-
Implementation Status	-

  

	<b>User views his timetable</b>
Actors	Student, Lecturer, Tutor, GUI, Database, Any registered user
Scope	GUI
Precondition	The system is running and the user is logged in.
Postcondition	The system is still running and the user is still logged in.
Postcondition on Success	The users timetable is displayed.
Basic Course of Events	<ol style="list-style-type: none"> <li>1. User clicks on MyCourses.</li> <li>2. The GUI requests the database for the students courses which the student is enrolled in.</li> </ol>
Alternative Paths	<b>2a.</b> The GUI requests the database for the lecturers courses which the lecturer gives. <b>2b.</b> The GUI requests the database for the tutor courses which the tutor is enrolled in and the courses which the tutor gives.
Open Questions	-
Solved issues	-
Implementation Notes	-
Implementation Status	-