

# Use Cases: *Enrollment*

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

March 4, 2011

	<b>User enrolls in a course</b>
Actors	Student, GUI, Database
Scope	GUI
Precondition	The system is running and the user is logged in with the rights of a student.
Postcondition	The system is still running, the user is still logged in.
Postcondition on Success	The user is enrolled in the chosen courses.
Basic Course of Events	<ol style="list-style-type: none"><li>1. User open course panel.</li><li>2. User selects a course instance from a list of course instances.</li><li>3. User clicks on submit.</li><li>4. GUI displays details about the course instance data, e.g. main lecturer, lesson period, different course element instances</li><li>5. User selects varies course element instances.</li><li>6. User clicks on submit.</li><li>7. GUI inserts data into the database with the association between person to course instance and person to course element instance.</li></ol>
Alternative Paths	-
Open Questions	-
Solved issues	<p><b>Q:</b> Do we provide a grading system, e.g. checking if enough course element instances were chosen?</p> <p><b>A:</b> No, but the relational schema allows the implementation of checks. A complete grading system could be implemented in</p>
Implementation Notes	<ul style="list-style-type: none"><li>• The implementation is based on the relation CourseElementInstance and not on the ProposedScheduling.</li></ul>
Implementation Status	-