

Week	Date	Activities, Topics, and Assessments
1	8/28	<p><b><u>Lecture 1: Introduction and Requirements Analysis (1)</u></b>  Introduction of instructor, course, students;  Overview of the ENPM613 class  Recap: SW Engineering overview; Models; OO SW development/programming</p>
2	9/4	<p><i>Project groups formation; Introduction of Project topic, activities, deliverables</i></p> <p><b><u>Lecture2:</u></b>  <b><u>Requirements analysis</u></b>  SW Requirements recap; Requirements analysis; Modeling in SW engineering; Domain/conceptual modeling; UML class diagram; Modeling the sw context with UML class diagram.  Modeling Use cases - UML Use case diagram; Description of Use cases; Textual (template-based); Use case description- graphical; UML Activity diagram</p> <p><i>Class exercise: Start developing the UCD for the project.</i></p> <p><b>Project management 101</b></p>
3	9/11	<p><b><u>Lecture 3: Software Quality attributes as design drivers</u></b>  SW Quality; Non-functional requirements; Quality scenarios; Utility tree  Security in the SDLC; Abuse Cases;</p> <p><i>Class exercise: Develop one abuse case and its description for the class project software. Start developing the utility tree for the class project software.</i></p>
4	9/18	<p><b><u>Lecture 4: User Centered design</u></b>  Concepts, user centric design process. User modeling (persona), interaction and UI modeling (wireframes, story boards), prototypes. Usability.</p> <p><i>Class exercise 1. Develop a persona for the project software typical user</i>  <i>Class exercise2: Develop a paper prototype for a scenario of the project software (and present it to class for "user representative" feedback)</i></p>

5	9/25	<p><a href="#">Brief Project presentation: User and Ux interaction models (10 mins/team)</a></p> <p><b>Lecture 5: Software engineering design</b>  Introduction to Software Design; Architecture and Detailed design;  SW Architecture - Definition, History, Stakeholders, Principles  Design approaches and strategies: functional decomposition</p> <p><i>Class exercise: start sketching the high level/preliminary architecture</i></p>
6	10/2	<p><b>Lecture 6: Software architecture development</b>  Design approaches and strategies: attribute driven design; Quality attributes mechanisms/tactics and styles;  Reuse; Designing with components and frameworks</p> <p><i>Class exercise: identify one or two quality attributes-related mechanisms/tactics for the class project</i></p>
7	10/9	<p><b>Lecture 7: Software architecture documentation</b>  Documenting/capturing architecture: Views;  UML component diagram, package diagram, deployment diagram, state diagram;  Design decisions and their documentation</p> <p><i>Class exercise: identify needed views and start developing one view for the class project</i></p>
8	10/16	<p><a href="#">Project presentation: Software preliminary architecture (15 mins/team)</a></p> <p><b>Lecture 8: Software architecture evaluation, evolution, and other considerations</b>  Architecture analysis/evaluation (scenario based);  Finalizing architecture: SAD; SAD review  Architecture reuse: SW Architecture for Product Lines  Architecture/design evolution, issues in maintenance of SW Architecture and Detailed Design;  Tools for architecture recovery, evaluation, and checking compliance with implementation  The role of Software Architect</p>
9	10/23	<b>MIDTERM EXAM - 2.5 hours</b>
10	10/30	<a href="#">Project presentation: Software architecture (completed)</a>

11	11/6	<b>Lecture 9: Software Detailed Design</b> Design principles and practices; SOLID Design patterns; Design documentation; Software implementation, testing, and deployment
12	11/13	<a href="#">Project presentation:</a> Software design
13	11/20	<a href="#">Project presentation:</a> Software design
14	11/27	<b>Thanksgiving break</b>
15	12/4	<a href="#">Final Project presentation:</a> Implementation demo
16	12/11	<b>FINAL EXAM</b> - Comprehensive assessment of all class material - 2.5 hours