Grupp 2

Tasks Affecting Final Grade

It is NOT required that all functionality of the product is implemented. Instead it is important to choose which features to implement. Code what is needed to have all architectural decisions implemented, NOT MORE.

1 Grade Limits

- Grade E
  - Pass all tasks that are marked X in the Mandatory column.
- Grade D
  - Pass all tasks that are marked X in the Mandatory column and also give thorough motivations for the chosen architecture.
  - Report all tasks no later than Mars 10, 2017
- Grade C
  - Meet requirements for grade D
  - Achieve a higher grade score of at least 6 (out of 20).
- Grade B
  - Meet requirements for grade D
  - Achieve a higher grade score of at least 11 (out of 20).
- Grade A
  - Meet requirements for grade D
  - Achieve a higher grade score of at least 16 (out of 20).

Task	Mandatory	Higher Grade Score	Signature
Robustness			
Flexible, easily understood code, this *must* be motivated. It *must* also be motivated which patterns, code conventions and frameworks are used and how they are used. Alternatively, *motivate* why they are not used and why the code is flexible and easily understood without them.	X		
Usage of the architectural patterns Layer and MVC. *Motivate*!	X		
Javadoc comment for all public definitions.	X		
Security			
Authorization	X		
Authentication	X		
Logging, motivate *what* is logged, *how* it is logged and *where* the log is.		1	
Transactions			
Motivate *when* and *how* transactions begin and end.	X		
O/R Mapping			
Motivate how primary keys are created.	X		
Management of relations, that is how relations in the database are translated into relations in the program. *Motivate* why the chosen solution is appropriate.	X		
Motivate the mapping between entities in the model and the database.	X		
Error Handling			
Explain how the runtime errors (e.g. threw exceptions) are handled. Motivate why the chosen solution is appropriate. It should be clear where different errors are handled, if they are presented to the user (and if so, how) and whether and how they are logged.	X		

Task	Mandatory	Higher Grade Score	Signature
Internationalization and localization			
Internationalization and localization of web pages.		1	
Internationalization and localization of database.		2	
Process			
Justify the choice of functionality that has been implemented. It should be clear what requirements are solved by the written code. All existing code should solve a requirement from this list. *Write no code just to add more functionality.*	X		
Frequent integrations. Code developed by different group members should *often* be tested together and integrated in the group's shared repository.	X		
All code and documentation are stored *only* in a single repository of a version control system. This must work throughout the project.	X		
Use a build tool, for example Maven. Ant is not accepted.	X		
Packaging of the final product in a format suitable for delivery.	X		
Use one more development tool(s), besides a version control system and a build tool. The tool must be used throughout the project, it is not enough just to get it started.		1	

Task	Mandatory	Higher Grade Score	Signature
Testing			
Automated in-container unit testing, using for example *Arquillian* **or** mocked-container unit testing, using for example *Mockito*. Choose one framework, you can not get points for both arquillian and mockito *The tests must be extensive, it is not enough just to get them going.*		Arquillian: 3 **or** Mockito: 2	Måste ~~vara~~ uppfyllas kravet
Acceptance testing, for example using *Selenium*. The tests must assert that execution is correct, it is not sufficient just to record an interaction with the application. *The tests must be extensive, it is not enough just to get them going.*		1	
Reporting			
All group members hand in an individual report at each of the three report meetings. It shall contain about 400 words and answer the questions What have you done? Exactly which parts of the code have yo written? **Everyone must participate in both coding and documentation.** What have you learned? How is the group's work organized? Any comments about the course?	X		
First report No later than February 10, 2017 (book reporting time in Doodle). You shall have code connecting all layers. An HTTP request shall result in execution of presentation layer, business logic and database. It shall be production quality code that can be left in the final product. You shall have a clear picture of what is done and what problems there are. *The solution shall be explained and motivated.*	X	1 Given if the report is accepted before due date.	

Task	Mandatory	Higher Grade Score	Signature
Reporting, Continued			
Second report No later than February 24, 2017 (book reporting time in Doodle). You shall have substantially more code than at the first report (Feb 10, 2017). It shall be production quality code that can be left in the final product. You shall have a clear picture of what is done and what problems there are. *The solution shall be explained and motivated.*	X	1 Given if the report is accepted before due date.	
Final report Performed no later than Mars 10, 2017 (book reporting time in Doodle). The group gives a detailed report to the supervisor. *The solution shall be explained and motivated.*	X	1 Given if the report is accepted before due date.	
Architecture Document following guidelines specified in lecture notes from lecture one. You shall hand in a new version of the architecture document, reflecting the current state of the application, at each report meeting (Feb 10 and February 24, 2017). The final version shall be handed in at the final report meeting (Mars 10, 2017) and also emailed to Leif Lindbäck, `leifl@kth.se`.	X	2 Given if the document has high quality.	

Task	Mandatory	Higher Grade Score	Signature
Other Functionality			
Validation of data entered in HTML forms. Validation shall take place in *all* layers on server.	X		
Views consist of different parts, e.g. header, footer, menu and main content.	X	1	
A standalone client communicating with the server using web services as specified in section 4.5 in the project task. Use for example JAX-RS to implement web services.		2	
PDF documents can be generated for job applications.		1	
Other Non-Functional Requirements			
Meet one or two non-functional requirement from the following list, explanations for the different requirements are found in the lecture notes from lecture one. It must be clear how you have proved that the requirement(s) are met. • Availability • Reliability • Response time • Capacity • Scalability • Non-repudiation		1 point if one requirement is met 2 points if two requirements are met Meeting more than two requirements does not give any extra points.	
Your own choice			
You can define another requirement, not mentioned in this list. You must discuss the requirement with the teacher, and agree on how many higher grade points it is worth.			1 Docker