

<div>Java Belt Rubric</div> <div>Auto Binary FailWeigh</div>			PASS		INCOMPLETE	
			Mastery	Proficient	Almost There	Struggling
			Really demonstrated mastery of skills and ability to synthesize concepts	Did the work as requested to the level needed in the field.	Minimal or few errors. A bit more time studying or tinkering may fix it.	Many errors or an indication of misunderstanding of concepts, or clear student copy/pasted code from outside code base. A bit more time with the instructor requested.
			100.00%	80.00%	50.00%	0.00%
Competency	Objective					
OOP	Correctly implement member variables, getters, setters and constructors as necessary.		More sophisticated model level validations or data types not included in the course material, ie. LocalTime type, uniqueness validation (1.0)	Implements models with getters, setters, generally uses appropriate data types, access modifiers and validation annotations. (0.8)	Functional use of getters, setters and constructors, but may be missing key annotations or fields. May use inappropriate access modifiers. (0.5)	Models have fatal errors, missing key elements, fails to compile due to entity creation errors. (0.0)
	Implement basic dependency injection, overloaded constructors and class methods. Use object instances to call instance methods, from within the same class or another class.	x1	May add additional logic to account for edge cases in the service layer. Includes a nullable many-to-one relationship, multiple 1:n or n:m using Lists. (1.0)	Appropriate use of dependency injection for service & controllers. Uses instance manipulation to call methods in other classes. Appropriate member variables and instance logic for one-to-many relationships using Lists. (0.8)	One-to-many relationships may be implemented, but are not implemented correctly as per the wireframe. (0.5)	No relationships present between classes. (0.0)
Routing, Rendering & Basic Backend Logic	Map requests and pass information using path variables and parameters. Handle POST requests, render JSPs. Implement RESTful design patterns.	x0.5	Uses full RESTful routing in addition to the proficiency requirements (0.5)	All the correct routes are in place and flow as expected from links or submission. However, may not have used strict RESTful routes for DELETE and PUT requests. (0.4)	Errors in paths or path variables, missing one controller method to connect a CRUD action, or one 404 response (0.25)	Uses path variables incorrectly, server does not run properly, multiple 404 responses (0.0)
	Session: Manipulate session to access user data across requests and perform basic authentication logic for CRUD functions.		User is unable to access any pages they shouldn't have access to. May also exhibit more sophisticated use of session, such as conditional template logic to prevent non-owning CRUD operations. (1.0)	Able to log in and log out, retaining the user's ID. Successfully retrieves and utilizes session id to display data correctly according to the wireframe. May implement unnecessary session variables. (0.8)	Logout does not clear the user ID from session. Or uses session inappropriately, ie. logged in user is displayed when it should be another user. (0.5)	ID is not saved or saved and accessed incorrectly in session. (0.0)
ORM & Database	Implement CRUD functionality using JPA queries and logic flow in the Service and Repositories.	x2	Demonstrates ANY of the following according to the wireframe: Querying capabilities for sorting, more complex filtering, or aggregation; has functional implementation of 3 or more classes; appropriately used a datatype not taught in the course material. (2.0)	CRUD functionality required for wireframe is completely implemented using the controller-service-repository modularization. (1.6)	One of the CRUD functionalities is incomplete or broken in the controller, service, or repository. For example, when attempting to edit, creates a new record. May be missing the service layer. (1.0)	Multiple CRUD functionalities are missing or broken in the controller, service or repository. (0.0)
	Create models and database relationships to perform CRUD queries using ORM (JPA Queries) correctly according to project specifications	2	Demonstrates ANY of the following: Many-to-many relationship. Multiple one-to-many relationships. Joining model. Able to reassign many-to-one relationships to another user. A self-join (2.0)	Successfully implements a one-to-many relationship with full CRUD functionality (1.6)	When attempting to edit loses one-to-many relationships or one-to-many relationships have other functional issues. (1.0)	Unable to query successfully for one-to-many relationships (0.0)
Full-Stack MVC	Include basic user authentication with password hashing and pattern validation by creating a login and registration feature	1	Full validations, all errors displayed properly. User is unable to see other pages when logged out. Implements Bcrypt or Spring Security (1.0)	Full validations, user cannot see the dashboard if logged out, but may be able to access other pages. Implements BCrypt hashing. (0.8)	One or at most 2 minor validations are missing such as uniqueness, but login still functional. Implements BCrypt hashing. (0.5)	Missing critical validations, produces an error page or does not implement hashing of passwords. (0.0)
	Build in pattern, logic and type validation on CRUD features.	1	Prevents non-owning users from altering data, and/or has more difficult or custom validations, such as date, time, uniqueness or pattern-matching (1.0)	Core validations are fully functional as per the wireframe on create and edit (0.8)	Validations work but do not appear on the page, or one of either create or edit does not have full validations (0.5)	Little or no validations, or critical errors on submission regarding validations (0.0)
	Demonstrate proficiency and good practice overall in code, design and presentation. Application meets at least "red belt" or minimum viable product requirements.	x0.5	Implements every CRUD, one to many and clickable buttons/links shown on wireframe. Clean and organized code. CSS and overall presentation is excellent. (0.5)	Fully functional, but may be missing minor elements that do not effect CRUD functionality such as navigational links, or other comparable minor features. (0.4)	May contain other back end or front-end implementation errors that significantly affect functionality. (0.25)	Overall, web app does not function sufficiently to assess features. (0.0)
Deployment on AWS	Black: Host web application on third party site.	b0	Followed instructions to deploy exam on AWS REQUIRED FOR BLACK BELT		Wasn't able to deploy the exam Only deducted if already has 9.5 or higher, will lose up to 0.51 points - whatever amount to put them below 9.5	
			0		-0.51	