

Onderdeel	Waar te vinden:	Weging
Secure Software		20%
De student heeft kennis en inzicht in een basisset aan maatregelen die genomen kunnen worden om de inbouw van kwetsbaarheden bij de bouw van software te voorkomen. De student kan afhankelijk van de situatie passende maatregelen selecteren om de basisveiligheid van software te maximaliseren.	Pagina 4 en 5: Risico- en beveiligingsanalyse; risico matrix, STRIDE-methode.	
De student toont aan mogelijke kwetsbaarheden in applicaties te kunnen herkennen, opsporen en tegenmaatregelen te treffen. Tevens kan de student software security toepassen in alle fasen van softwareontwikkeling.	Kwetsbaarheden: -form-login -> login.html -form-registration -> registration.html -form-upload -> dropdemo.html -SQL-queries -> data.sql, SecurityConfiguration.java, Demofile.java en alle repositories. -public class DemoFile -> Demofile.java Tegenmaatregelen: User.java: -javax.validation -javax.validation.constraints.Email -javax.validation.constraints.NotEmpty -hibernate.validator.constraints.Length SecurityConfiguration.java: -AuthenticationManagerBuilder -jdbcAuthentication() -HttpSecurity -authorizeRequests() WebMvcConfig.java: -BCryptPasswordEncoder() Demofile.java -java.io.Serializable pom.xml -hibernate-entitymanager -spring-boot-starter-security -spring-boot-security-test	
Ontwerp		30%
De student kan voor het ontwikkelen van een applicatie de requirements opstellen.	Pagina 6: Requirements.	
De student kan een duidelijk functioneel en technisch ontwerp maken volgens geldende richtlijnen.	Pagina 7 en 8: Use-case diagram en Klassendiagram	
De student toont aan een onderzoek uit te kunnen voeren naar de gewenste gebruikerservaring (UX) en user interface (UI) van de webapplicatie. Tevens kan de student de resultaten vertalen naar aanpassingen in het ontwerp.	Pagina 9 t/m 11: Rapportage UX onderzoek, Persona en User Stories. Online enquête: https://bit.ly/2u4uGZE	

Onderdeel	Waar te vinden	Weging
De student kan op basis van requirements, functioneel en technisch ontwerp een klikbaar prototype ontwerpen om het ontwerp, workflow en ervaring van gebruikers te verifiëren.	Pagina 12: Prototype. Klikbaar prototype: https://adobe.ly/2uUxJUN	
Front-end		20%
De student kan op basis van requirements, UX/UI wensen en geldende richtlijnen en technieken de front-end van de webapplicatie bouwen. Tevens toont de student aan opmaaktalen zoals HTML, CSS en Javascript te kunnen toepassen.	Pagina 13: Front-end. -HTML: home.html, login.html, registration.html, dropDemo.html, profile.html. -CSS: style.css, home.css, login.css, registration.css, hexagon.css -JavaScript/JQuery: /employee/home.html	
De student kan interactieve user interfaces componenten bouwen voor webapplicaties door middel van het React framework o.i.d.	Pagina 13: Bootstrap. Bootstrap: toegepast in elk html-bestand.	
Back-end		30%
De student kan de informatiebehoefte van een opdrachtgever vertalen naar een datamodel tevens toont de student aan SQL select instructie te kunnen maken, waarbij gebruik wordt gemaakt van 1 of meerdere tabellen in een (relationele) database.	Pagina 14: Datamodel. -SQL: data.sql, DemoFilesRepo.java, RoleRepository.java, User,Repository.java, SecurityConfiguration.java en Demofile.java. Models/entity classes: Demofile.java, Role.java, User.java	
De student kan volgens relevante technieken en richtlijnen de back-end van een webapplicatie bouwen en testen in Java of Python. Ook past de student principes van object oriëntatie toe bij het bouwen van de webapplicatie.	Pagina: 13 en 14: Webapplicatie, Gebruikte technieken en Unit testing.	
Totaal		100%