De **Kwetter backend** ga je nu uitbreiden met een front-end op basis van Java Server Faces (JSF) enerzijds en Angular anderzijds. De Java Server Face technologie ga je toepassen om de beheerscherm(en) van de applicatie te ontwikkelen. Voor de gebruikersschermen (home, profiel etc) ga je een moderne en responsive UI ontwikkelen met Angular.

De user stories voor de front-end zijn uitgeschreven in de Kwetter Casus. Zie de opdrachtomschrijving in Canvas om de hyperlinks terug te vinden naar de bijkomende informatie.

[Overzicht front-end functionaliteiten 2](#_Toc505552645)

[Profiel pagina (Angular) 2](#_Toc505552646)

[Startpagina (Angular) 2](#_Toc505552647)

[Beheer (Java Server Faces) 2](#_Toc505552648)

[Realisatie 2](#_Toc505552649)

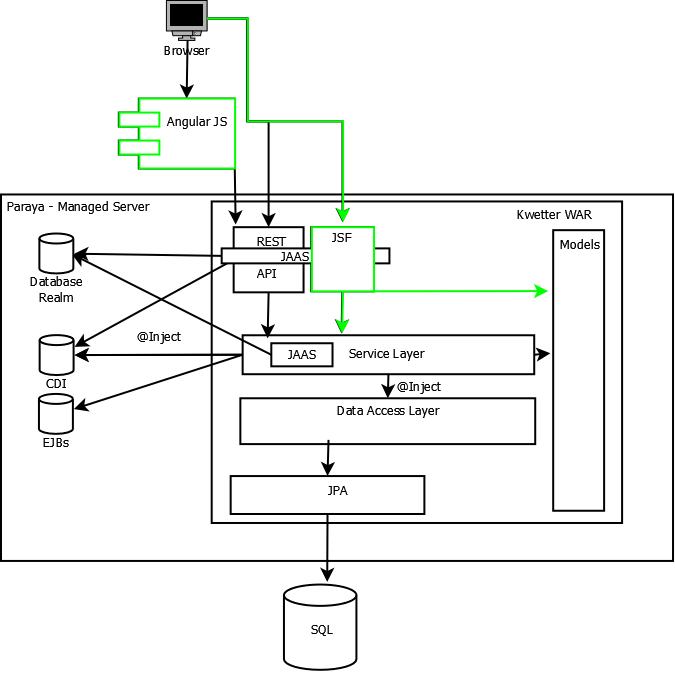
[GUI 1) JSF 2](#_Toc505552650)

[GUI 2) Angular 2](#_Toc505552651)

[Beoordeling 3](#_Toc505552652)

[Literatuurlijst 3](#_Toc505552653)

[Optionele functionaliteiten 3](#_Toc505552654)



Figuur Te realiseren componenten gebruikmakend van de backend als front-end

# Overzicht front-end functionaliteiten

Hierbij realiseer je de schermen zoals beschreven in figuur 1 en figuur 2 uit de Kwetter Casus met de onderstaande functionaliteiten.

## Profiel pagina (Angular)

* Profiel foto en profiel naam (S0)
* Profiel details (S1)
* Eigen tweets (S2)
* Followers (S3)
* Following (S4)

## Home(start)pagina (Angular)

Dit scherm is jouw persoonlijke pagina eens je ingelogd bent.

* Search
* What’s happening?
* Timeline
* Following/Followers
* Login / Logout

## Beheer (Java Server Faces)

Een administrator of een moderator heeft de mogelijkheid om de kwetters en gebruikers te beheren.

* Inloggen met moderator of administrator rol
* Verwijderen van berichten v/e gebruiker
* Overzicht van gebruikers (en rollen)
* Gebruikers rollen kan toekennen en gebruikers beheren.

# Realisatie

## GUI 1) JSF

Je gaat de beheerscherm(en) realiseren in Java Server Faces. Creëer de benodigde JSF pagina’s, JSF templates en controller klassen om de beheerscherm(en) te realiseren. Voor het security aspecten moet je gebruik maken van Session-Bases-Security beschreven in de Support-Security opdracht

## GUI 2) Angular

Kwetter™ Headquarters heeft kennisgenomen van een veelbelovende alternatieve technologie voor het maken van websites: Angular. Om te onderzoeken wat deze techniek kan doen voor Kwetter wordt er door jou een Single Page Applicatie (SPA) prototype ontwikkeld met alle gebruikers functionaliteiten. Creëer het Angular project waarbij je gebruik maakt van je bestaande REST API om de gegevens synchroon en asynchroon op te halen. Voor de security aspecten kun je gebruik maken van Token-based-security voor REST authenticatie en authorisatie, zie beschrijving in de Security opdracht.

# Beoordeling

De student demonstreert de functionaliteit van de gerealiseerde applicaties waarbij hij 1) de scenario’s aantoonbaar getest heeft met unit testen en 2) een scenario van de docent via de GUI kan demonstreren.

De kwaliteit van de programmacode wordt meegenomen in de beoordeling.

1. Ontwerp voldoet aan SOLID principes en waar mogelijk zijn design patterns toegepast
   * Je kunt je code verdedigen op vlak van uitbreidbaarheid en onderhoudbaarheid.
2. Gebruik maken van de mogelijkheden van de enterprise web server voor object management en resource management
   * Je gebruikt geen SE oplossingen waar EE mogelijk is, en je hebt een bewuste keuze gemaakt tussen CDI en EJB annotaties.
3. Je GUI biedt de vereiste functionaliteiten aan, ook afhankelijk van soort ingelogde gebruiker
4. Je kunt gebruik maken van JSF, een Server-side component-oriented Java web framework waarbij je diens Architectuur en LifeCycle van de backing beans begrijpt en goed kunt toepassen
   * Je begrijpt welke mogelijkheden Primefaces biedt, en kunt deze in je JSF templates verwerken.
   * Je begrijpt welke authenticatie en authorisatiemogelijkheden JAAS aanbiedt voor JSF en kunt deze toepassen
5. Je kunt gebruik maken van Angular, een client-side javascript framework waarbij je diens Architectuur en pagina routering begrijpt en goed kunt toepassen
   * Je begrijpt welke mogelijkheden er zijn om van een REST api gebruik te maken, en je kunt dit toepassen.
   * Je begrijpt hoe de buildingblocks Modules, Components, templates, databinding, directives, services en dependency injection gebruikt kunnen worden en hebt dit kunnen toepassen.
   * Je hebt deep linking mogelijk gemaakt

# Literatuurlijst

Zie de opdrachtomschrijving van Front-end in Canvas om de hyperlinks terug te vinden naar de bijkomende informatie. Bestudeer eerst Servlets (voorkennis) en JavaServer Faces en vervolgens Angular.

# Optionele functionaliteiten

Bekijk de specifieke opdracht pagina’s in Canvas voor meer informatie over deze uitbreidingen.

**JSF: prime faces & meertaligheid**

In deze opdracht ga je je kwetter JSF front-end uitbreiden zodat de GUI verrijkt wordt met PrimeFaces en meertaligheid.

**Front-end/Backend: mentions en trends**

In deze opdracht ga je je kwetter backend en front-end uitbreiden om in berichten naar andere gebruikers te verwijzen (en die ontvangen daar dan een melding van), en trendy onderwerpen weer te geven op basis van gebruikte hashtags.

**GUI snelheid: asynchroon & caching (Hartje)**

In deze opdracht ga je een kwetter GUI naar keuze uitbreiden zodat de 'gevoelde' traagheid van interacties zo minimaal wordt, door een Like functionaliteit toe te voegen en **Front-end/Backend queries**: mentions en trends te realiseren met aandacht voor laadsnelheid.

**Angular: GUI testing**

In deze opdracht ga je de Angular kwetter GUI voorzien van minstens 3 geautomatiseerde testen die de werking van de belangrijkste functionaliteiten van je front-end valideren. Bestuur de mogelijkheden om Angular te [testen](https://angular.io/guide/testing).