|  |
| --- |
| Advanced Programming, Advanced software engineering |
| Music matching voor Dare2Date |
| Software architectuur document |

|  |
| --- |
| Hanna Hermsen 474296 Jeroen Jansen 483264  14-5-2014  Versie 0.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Naam | Datum | Wijziging |
| V0.1 | Jeroen & Hanna | 8-5-2014 | Opzet gemaakt, smart use cases en niet-functionele requirements toegevoegd |
| V0.2 | Jeroen & Hanna | 13-5-2014 | Organogram, flow chart en sequence diagrams toegevoegd. |
| V0.3 | Jeroen & Hanna | 14-5-2014 | Class diagram, package diagram en deployment diagram toegevoegd |

Inhoudsopgave

[1. Inleiding 4](#_Toc387850695)

[2. Architecturale eisen 5](#_Toc387850696)

[2.1. Niet functionele eisen 5](#_Toc387850697)

[2.2. Use case view (functionele requirements) 5](#_Toc387850698)

[3. Logical view 6](#_Toc387850699)

[3.1. Lagen 6](#_Toc387850700)

[3.2. Deelsystemen 7](#_Toc387850701)

[3.2.1 Service sequence diagram 7](#_Toc387850702)

[8](#_Toc387850703)

[3.2.2 Deelsysteem sequence diagrammen 9](#_Toc387850704)

[3.3 Use Case Realizations 10](#_Toc387850705)

[3.3.1 Flow diagram matching service 11](#_Toc387850706)

[3.4 Class diagram 11](#_Toc387850707)

[4 Implementation view 12](#_Toc387850708)

[4.1 Package structuur 12](#_Toc387850709)

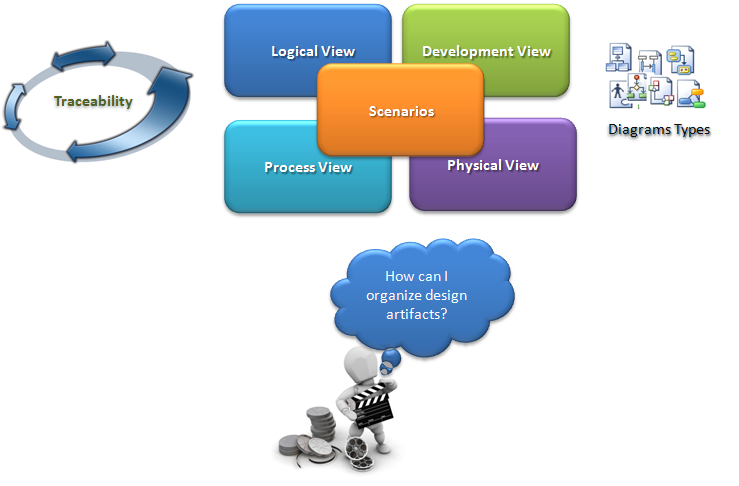
[4.2 Package diagram 12](#_Toc387850710)

[5 Deployment view 13](#_Toc387850711)

# Inleiding

Voor u ligt het Software Architectuur Document (SAD) van Jeroen Jansen en Hanna Hermsen.  
Het bevat een uitgebreide architecturale kijk op de music matching functionaliteit die wij bouwen voor de dating website Dare2Date.   
In principe is het document bedoelt voor Rody Middelkoop, docent van het vak Advanced Software Engineering van de minor Advanced Programming die wij nu volgen. Desalniettemin is dit document ook geschikt voor andere technische stakeholders van Dare2Date.

Dare2Date is een (nu nog fictieve) website waar 30+'ers hun ideale partner kunnen ontmoeten. Als toevoeging op het matching gedeelte zullen wij een music matching functionaliteit ontwerpen en implementeren die gebruik maakt van last.fm. Dit is een website die bijhoudt wat voor muziek iemand luistert en bevat ook sociale aspecten zoals een berichtenservice en groepen. Ook kan men elkaar uitnodigen voor evenementen (concerten/festivals) en deze joinen op last.fm.



Figuur 1: 4 + 1 Model

Bij het ontwerpen maken we gebruik van het 4 + 1 model, waarmee verschillende aspecten van het systeem worden toegelicht. Dit model stelt de verschillende belanghebbenden in staat vanuit hun eigen perspectief de invloed van de gekozen architectuur te bepalen. De Process View (communicatie van processen) is niet als los hoofdstuk uitgewerkt maar ondergebracht bij de hoofdstukken 3.3 en 5.

# Architecturale eisen

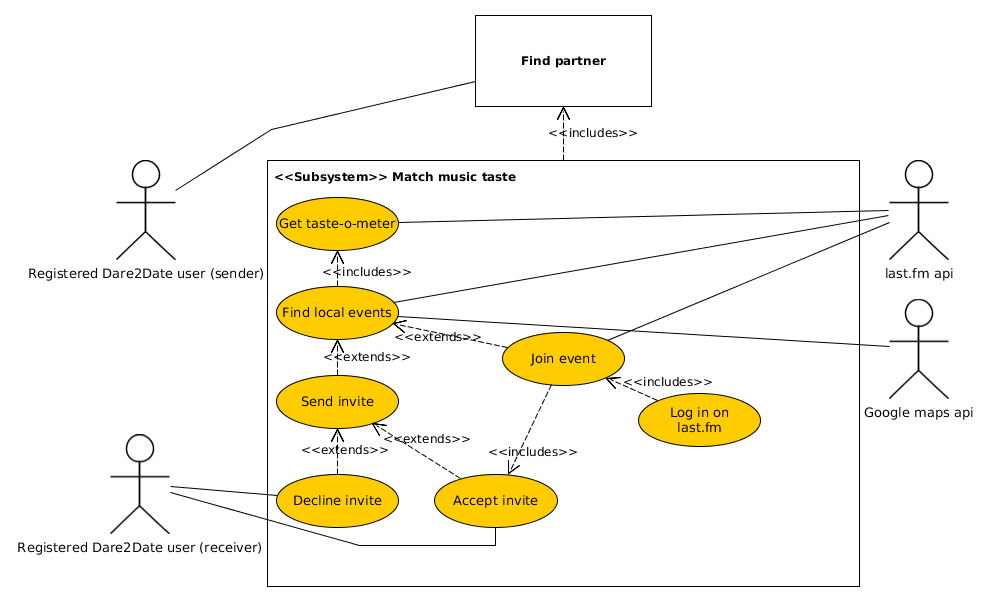
Deze sectie beschrijft de software eisen die voor het ontwikkelen van de software architectuur van belang zijn.

## Niet functionele eisen

|  |  |
| --- | --- |
| Naam | Architecturele relevantie |
| Systeem onafhankelijkheid | Er zal ontwikkeld worden in Java. |
| Ontwikkeltools | Voor de ontwikkeling van de software zal er gebruik worden gemaakt van de Eclipse IDE. |
| Versiebeheer | Voor versiebeheer van documenten en code zal er gebruik worden gemaakt van een git repository. |

Tabel 2.1.1: de niet-functionele eisen

## Use case view (functionele requirements)

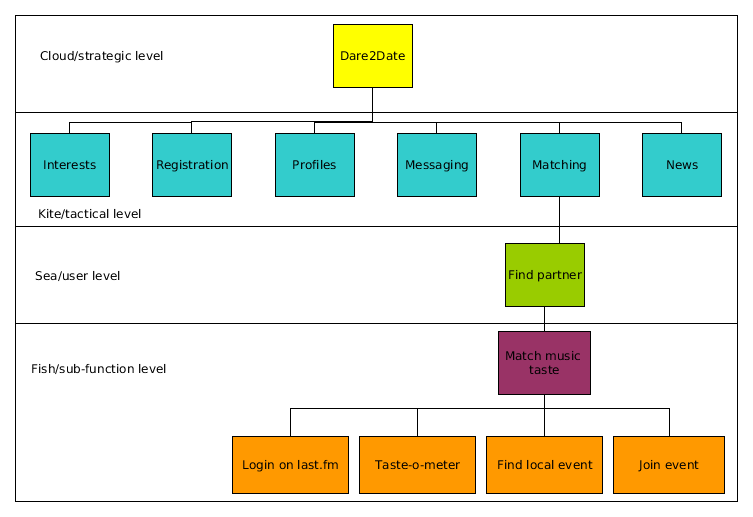


Figuur 2.2.1: use case matching

# Logical view

Hier wordt de architectureel significante logische opbouw van het systeem beschreven. Denk hierbij aan de decompositie in lagen en deelsystemen. Ook de manier waarop Use Cases, rekening houdend met deze logische decompositie, technisch worden vertaald naar Use Case Realizations worden hier beschreven.

## Lagen



Figuur 3.1.1: organogram van de business structuur Dare2Date

## 

Hierboven is te zien welke service er wordt uitgewerkt voor de dating site Dare2Date. De service die zal worden aangeboden is een matching service op basis van muzieksmaak. Wanneer er een match is gevonden worden aankomende evenementen weergegeven van artiesten waar beide personen naar luisteren (allemaal in de buurt). Er kan dan voor gekozen worden om de match uit te nodigen voorhet evenement en op basis van de reactie van de match het evenement zelf te joinen (of niet). Wanneer aan deze evenementen wordt deelgenomen wordt dit geüpdate op hun last.fm account.

## Deelsystemen

Hier wordt de decompositie van het systeem in deelsystemen de samenhang hiertussen beschreven met behulp van sequence diagrammen.

### 3.2.1 Service sequence diagram

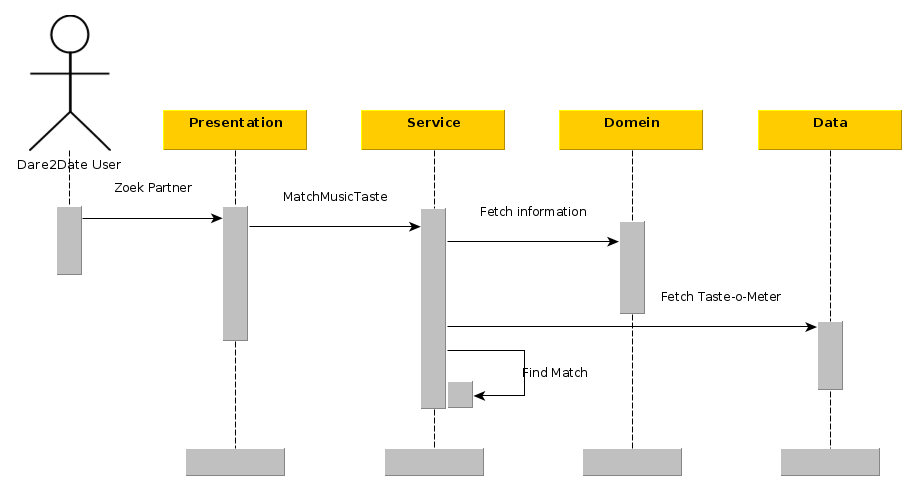
Op de volgende pagina staat het service sequence diagram. Hierin staat de communicatie tussen de services beschreven.

### 

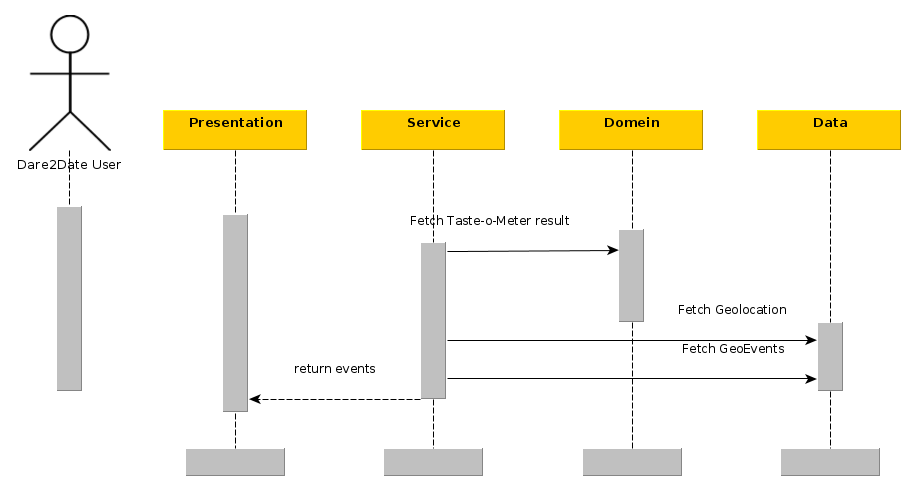
Figuur 3.2.1.1: Service sequence diagram

Figuur 3.2.1.1: service sequence diagram

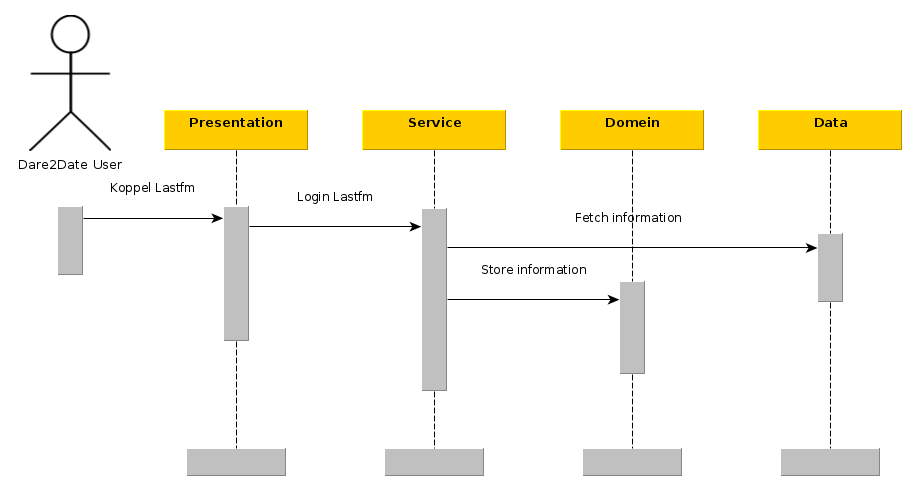
### 3.2.2 Deelsysteem sequence diagrammen



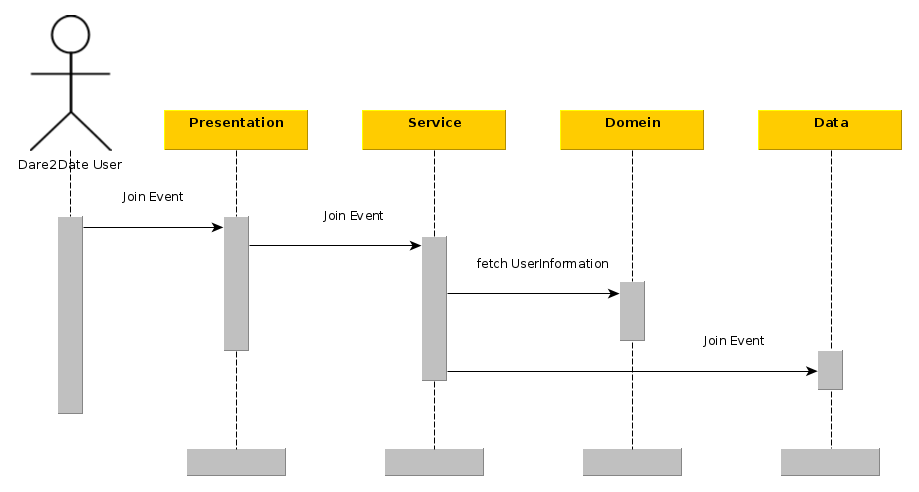
Figuur 3.2.2.1: match music taste deelsysteem



Figuur 3.2.2.2: return events deelsysteem



Figuur 3.2.2.3: link account deelsysteem

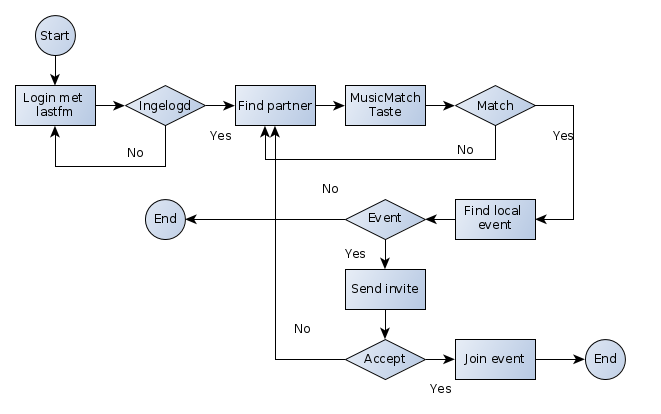


Figuur 3.2.2.4: join event deelsysteem

## 3.3 Use Case Realizations

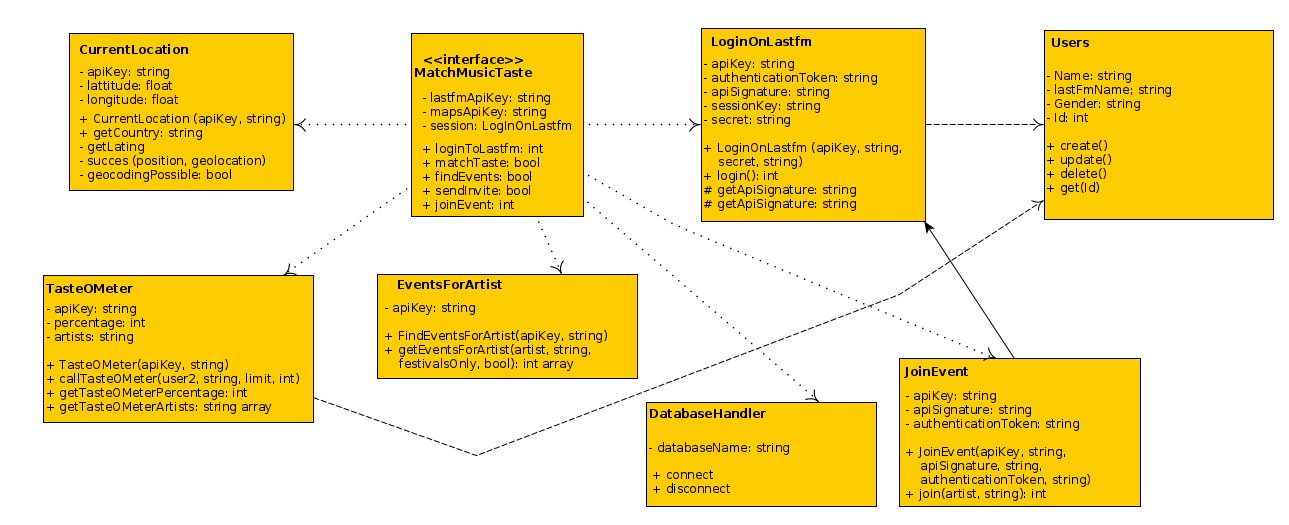
Hier wordt geillusteerd op welke wijze waarop de eerder genoemde Use Case Specifications technisch vertaald worden naar Use Case Realizations.

### 3.3.1 Flow diagram matching service



Figuur 3.3.1.1: flow chart MatchMusicTaste service Dare2Date

## 3.4 Class diagram

In het onderstaande class diagram wordt de invulling en samenhang van/tussen de classes weergegeven.

Figuur 3.4.1: class diagram

# 4 Implementation view

## 4.1 Package structuur

De Dare2Date MusicMatch service zal een aantal verschillende packages bevatten. Deze packages zijn als volgt:

nl.han.ica.daretwodate.services

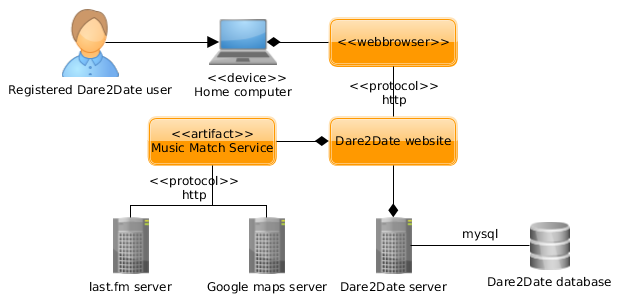
nl.han.ica.daretwodate.domains

nl.han.ica.daretwodate.data

nl.han.ica.daretwodate.config

# 5 Deployment view

Hieronder staat een deployment diagram die de fysieke opstelling laat zien.



Figuur 5.1: deployment diagram