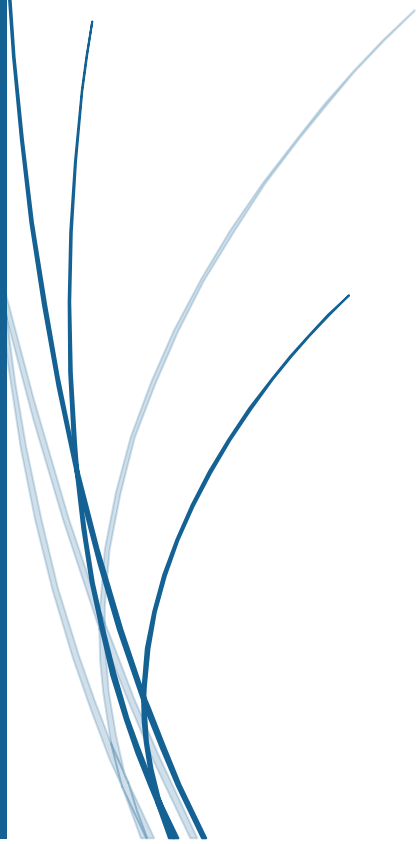
06/19/2022

Analysis

Kwetter-Case



Samir Zalmay
FONTYS HOGESCHOLEN

INHOUD

1. INTRODUCTION

In this semester, the design and implementation of Enterprise software are paramount. With this project, I will prove what I've learned from this. Enterprise software must be able to handle many users who simultaneously perform actions on the application. It must of course also be able to handle a lot of data transfer. Some other points that are important for this project are the safety, portability, and maintainability of the product. To be able to apply all this as effectively as possible, the plan of approach must be carefully considered. That is why I first start by drawing up an analysis document. This document is therefore going to create a picture of the product. All ideas and information found are brought together in a structured way in this document.

2. FUNCTIONAL REQUIREMENTS

In chapter 2 I usually explain the background and the vision of the project. However, since I will be working on the Kwetter-case, you can find the introduction here. As for this chapter, the functional requirements will be stated in the table down below. These functional requirements are deducted from the project-description in the Kwetter Case (see hyperlink above).

ID	UCID*	NAME	DESCRIPTION	MOSCOW	URGENCY /JIRA
F1	**UC01**	Register	A **user** must be able to register him/herself.	**MUST**	**HIGH**
F2	**UC02**	Login	A **user** must be able to log in to have access to certain areas of the website.	**MUST**	**HIGH**
F3	**UC03**	Seeing trends	A (**non-)user** must be able to see the current top 5 trends (applies **R1**** CH 4).	**MUST**	**HIGH**
F4	**UC04**	Seeing tweets	A (**non-)user** must be able to see the most recent tweets based on a trend (applies **R2**** CH 4.	**MUST**	**HIGH**
F5	**UC05**	Personal Info	A user must be able to see and modify his/her personal info (applies **R3&4****).	**MUST**	**HIGH**
F6	**UC06**	Personal Tweets	A **user** must be able to see/modify/delete his/her own tweets (applies **R4&5&6&7****).	**MUST**	**HIGH**
F7	**UC07**	Timeline	A **user** must be able to see the most recent tweets of his/herself and the other users being followed.	**MUST**	**HIGH**
F8	**UC08**	Like	A **user** must be able to heart a tweet from another user (applies **R8****).	**MUST**	**HIGH**
F9	**UC09**	Mentions	A **user** must be able to see the most recent tweets where his/her name is mentioned.	**MUST**	**HIGH**
F10	**UC10**	Create tweet	A **user** must be able to create a tweet (applies **R5&6&7****).	**MUST**	**HIGH**
F11	**UC11**	Followers	A **user** must be able to see which other users are following him/her.	**MUST**	**HIGH**
F12	**UC12**	(Un)follow	A **user** must be able to (un)follow another user.	**MUST**	**HIGH**
F13	**UC13**	Following	A **user** must be able to see which other users his/her followers/following users are following.	**MUST**	**HIGH**
F14	**UC14**				

*** Stands for Use Case ID. Will be referred to in Chapter 5 Use Cases.**
**** Stands for Rules and refers to the rules found in Chapter 3.**

3. RULES

In chapter 2 the functional requirements have been properly written. However, some of these requirements are large and need some rules applied to them within the function. The rules can be found down below and are pointed out in the table of the functional requirements in chapter 2.

ID	DESCRIPTION
R1	The top 5 trends are calculated based on at most one-week-old tweets.
R2	Tweets are at most one-week-old and contain a hashtag followed by the selected trend
R3	Biography is limited to 160 characters.
R4	For anything personal, the user can decide for who it is visible.
R5	Tweets are limited to 140 characters.
R6	You can mention another user by using the '@' annotation in front of the username.
R7	You can contribute to trends by using the '#' annotation in front of a word.
R8	Cannot heart more than 1 time per tweet per user.

f the data has been entered incorrectly, the actor will see a message and the actor will then be returned to step 1.

4. USE-CASES

In this chapter you can find the Use Cases for each requirement of Chapter 2.

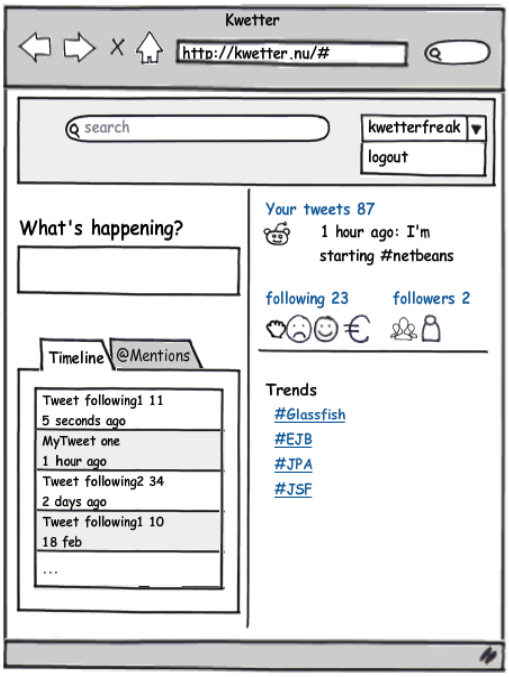
Name	**UC1:** Login
Summary	A user can login
Actor	User
Assumptions	
Description	1. The user enters his/her username and password. 2. The system checks whether the data is correct. [1] 3. The user is logged in.
Exception	[1] If the data has been entered incorrectly, the actor will see a message and the actor will then be returned to step 1.
Result	The user is logged in.

Naam	**UC2:**
Samenvatting	
Actoren	
Aannamen	
Beschrijving	
Uitzondering	
Resultaat	

Naam	**UC3:**
Samenvatting	
Actoren	
Aannamen	
Beschrijving	
Uitzondering	
Resultaat	

5. GLOBAL UI

Homepage



Profile page