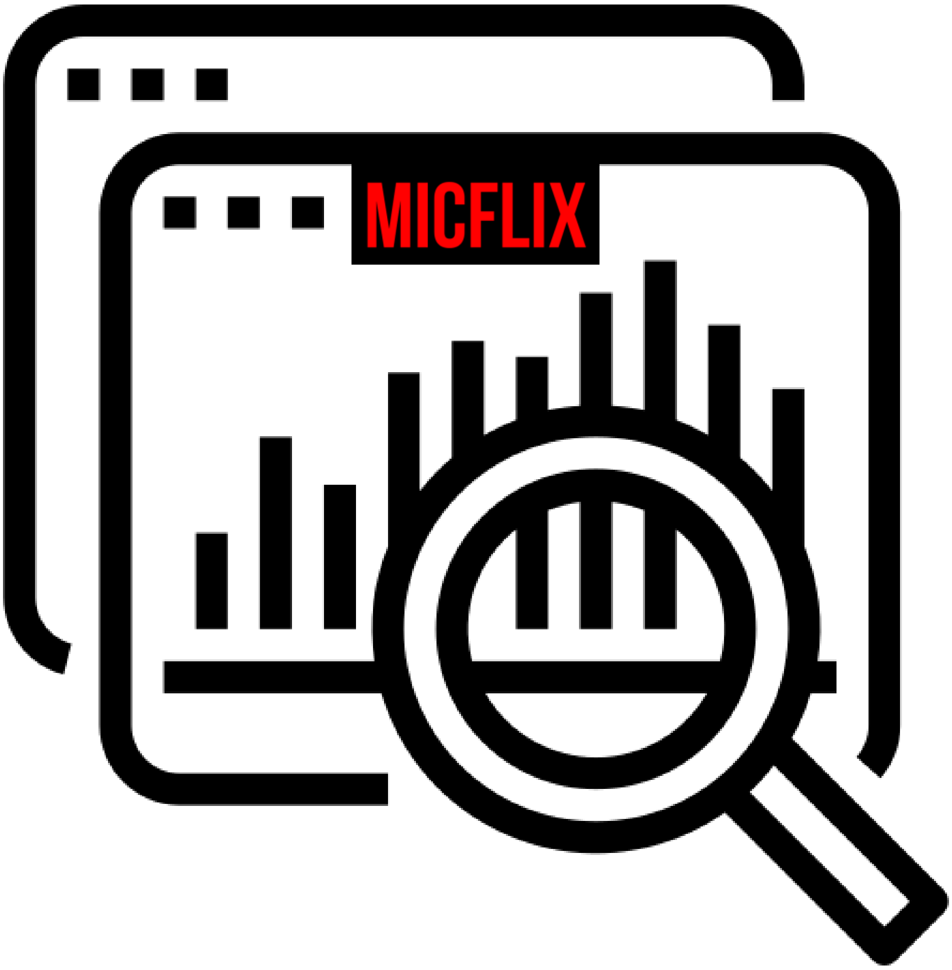
ANALYSIS – MICFLIX

FONTYS SEMESTER 6 SOFTWARE PROJECT



Data free icon: Flaticon.com

Date	**07-03-2022**
Version	**0.4**
State	**In progress**
Author	**Mickey Krekels**
Class	**RB04**

VERSION HISTORY

Version	Date	Author(s)	Changes	State
0.1	04-03-2022	Mickey Krekels	Added the main structure of the document.	In progress
0.2	07-04-2022	Mickey Krekels	Added the Deliverables & Non-deliverables chapter	In progress
0.3	08-03-2022	Mickey Krekels	Finished the Deliverables & Non-deliverables chapter	In progress
0.4	08-03-2022	Mickey Krekels	Finished the chapter use cases	In progress

PREFACE

This project is made for the Fontys semester-6 personal assignment. The goal is to make a streaming service clone with a strong inspiration from the platform Netflix. This clone will be called Micflix and the final version of the project will be of enterprise quality.

This analysis document serves as the main requirements description of the project. Each of the functional requirements will be coupled to a use case, where the overall steps and problems will be described. The non-functional requirements will be categorized based on their type and for each, there will be a requirement added. At the end of the analysis, there will be a final conclusion.

TABLE OF CONTENT

1 REQUIREMENTS

1.1 DELIVERABLES & NON-DELIVERABLES

These tables show all the deliverables & non-deliverables of this project, all the requirements will be prioritized using the MoSCoW technique (1).

Index	Functional requirements	Must	Should	Could	Won't
FR-1	User is able to register an account	X			
FR-2	User is able to login to his account	X			
FR-3	User can update their user data			X	
FR-4	User can view a selection of movies based on the date added		X		
FR-5	User can view a selection of movies based on last watched	X			
FR-6	User can pause the selected movie		X		
FR-7	User can change the resolution of the selected movie			X	
FR-8	User can download a selected movie				X
FR-9	User gets a notification on their application when a new movie is available.			X	
FR-10	User gets a selection of movie recommendations based on previously watched movies			X	
FR-11	User can filter movies based on name	X			
FR-12	User can filter movies based on genre	X			

Index	Non-Functional requirements	Must	Should	Could	Won't
NFR-1	The solution should be runnable in a Kubernetes cluster		X		
NFR-2	The solution must have a scalable architecture	X			
NFR-3	The solution has a mobile version				X
NFR-4	The solution must be GDPR compliant	X			
NFR-5	The solution must a CI/CD pipeline	X			
NFR-6	The solution must be secure				

1.2 REQUIREMENT USECASES

1.2.1 Login User

Use Case Name:	Login user
Index:	FR-1
Summary:	To view the available videos on the Micflix streaming platform, a user **must** be able to login into the system.
Basic Flow:	1. The user wants to login 2. The system requests the username and password 3. The User provides the username and password 4. The system compares the username and password with the hashed and secured data in the database. 5. The system starts a login session and provided the user with a secured login token
Alternative Flows:	1. if the username is invalid, the user goes back to step 2. 2. if the password is invalid, the user goes back to step 2.
Preconditions:	The user is registered.
Postconditions:	The user is login in on the platform

1.2.2 Register User

Use Case Name:	Register user
Index:	FR-2
Summary:	To login onto the Micflix streaming platform, a user **must** be able to register on the system.
Basic Flow:	1. The user wants to register. 2. The system requests a username, password and mail address. 3. The user enters a username and password. 4. The system checks that the username does not duplicate any existing registered usernames. 5. The system reroutes the user to the login screen
Alternative Flows:	1. If the username duplicates an existing username the system displays a message and the user goes back to step 2.
Preconditions:	None
Postconditions:	The user is now registered on the streaming platform.

1.2.3 Update User data

Use Case Name:	User can update their user data
Index:	FR-3
Summary:	A user wants to update their personal user data(username, password, email)
Basic Flow:	1. User wants to update their account data 2. The system sends a randomized code to the user 3. User copies the randomized code from their mail 4. User enters the randomized code into the system 5. System validates code 6. System sends user to account update page 7. User changes account data 8. System updates users account data
Alternative Flows:	1. If the randomized code is not equal to the one from the mail, user goes back to step 1.
Preconditions:	The user is registered.
Postconditions:	The user account data is updated

1.2.4 Selection of movies based on the date added

Use Case Name:	Selection of movies based on the date added
Index:	FR-4
Summary:	A user must be able to see a selection based on newly added movies
Basic Flow:	1. User wants to see a selection of newly added movies 2. User selects newly added 3. System provides all recently added movies
Alternative Flows:	None
Preconditions:	The user is registered.
Postconditions:	User gets a selection of recently added movies

1.2.5 Selection of movies based on last watched

Use Case Name:	Selection of movies based on last watched
Index:	FR-5
Summary:	A user must be able to continue their previously watched movie, therefor a selection of previously watched movies must be visible for the user.
Basic Flow:	1. User wants to see a selection of previously watched movies 2. User selects previously watched 3. System provides all previously watched movies
Alternative Flows:	1. If there is no record of previously watched movies, user gets rerouted to the main page.
Preconditions:	The user is registered. User must have a record of previously watched movies
Postconditions:	User gets a selection of previously watched movies

1.2.6 Pause video

Use Case Name:	Pause video
Index:	FR-6
Summary:	A user must be able to pause their selected movie
Basic Flow:	1. User wants to pause their selected movie 2. User clicks on the pause button 3. System saves current timestamp of the movie duration
Alternative Flows:	None
Preconditions:	The user is registered. Movie must be selected
Postconditions:	The selected video is paused

1.2.7 Select resolution

Use Case Name:	Select resolution Scenario: user selects the required resolution of the video.
Index:	FR-7
Summary:	A user must be able to change the resolution selected movie
Basic Flow:	1. User wants to change the resolution of the video 2. User clicks the resolution dropdown menu 3. User selects their preferred resolution 4. System saves current timestamp of the movie duration 5. System retrieves the data selected resolution version of the movie 1. System reloads movie from the point of the timestamp
Alternative Flows:	None
Preconditions:	The user is registered. Movie must be selected
Postconditions:	The resolution is changed and the movie is continued from the timestamp

1.2.8 Download a selected movie

Use Case Name:	Download a selected movie
Index:	FR-8
Summary:	A user wants to download a movie for watching offline.
Basic Flow:	1. User wants to download the selected movie 2. User clicks the download button 3. System download the selected movie on the selected resolution 4. System adds the download movies to the downloaded selection
Alternative Flows:	1. If the internet is not available, the System cancels the download and the user is redirected to step 1 with a message no internet available.
Preconditions:	The user is registered. The user must have internet access
Postconditions:	The movie is downloaded on their local device

1.2.9 Notification when a new movie is available.

Use Case Name:	Notification when a new movie is available.
Index:	FR-9
Summary:	When a new movie is available, the system lets the user know with a notification.
Basic Flow:	1. System sends notification that there is a new movie available 2. User clicks on the notification 3. System directs the user to newly added movie
Alternative Flows:	None
Preconditions:	The user is registered.
Postconditions:	User is informed or redirected about the newly added movie

1.2.10 Movie recommendations based on previously watched movies

Use Case Name:	Movie recommendations based on previously watched movies
Index:	FR-10
Summary:	When a user has watched movies with a type of genre, a selection Is made available with recommendations for movies with that genre type.
Basic Flow:	1. User views a set of movies with a similar genre 2. System provides a set of recommendations in a selection view
Alternative Flows:	None
Preconditions:	The user is registered. The user viewed a set of movies with a similar genre
Postconditions:	User has a selection view added with recommendations

1.2.11 Filter movies based on genre

Use Case Name:	Filter movies based on genre
Index:	FR-11
Summary:	A user must be able to filter movies based on genre types
Basic Flow:	1. User wants to see a selection of movies with a genre type 2. User types the genre in the search bar 2. System provides all movies with the genre type
Alternative Flows:	1. If the genre is not recognized by the system, user gets redirected to step 2 2. If the user enters nothing and hits enter the system redirected to step 2
Preconditions:	The user is registered.
Postconditions:	User is shown a selection of movies with the selected genre type

1.2.12 Filter movies based on name

Use Case Name:	Filter movies based on name
Index:	FR-12
Summary:	A user must be able to filter movies based on name
Basic Flow:	1. User wants to see a selection of movies with a specific name 2. User types the name in the search bar 3. System provides all movies with the specific name
Alternative Flows:	1. If the name is not recognized by the system, user gets redirected to step 2 2. If the user enters nothing and hits enter the system redirected to step 2
Preconditions:	The user is registered.
Postconditions:	User is shown a selection of movies with the specific name

2 SECURITY BY DESIGN

The phrase "Security-by-Design" refers to an approach in which security is a priority across the whole software development lifecycle. The security-by-design technique suggests that security-related activities be considered at every stage of software development.

2.1 OWASP

The OWASP Foundation is a non-profit organization dedicated to improving software security. The Foundation is the source for developers and technologists to safeguard the web through community based open-source software projects.

In this chapter of the document, the OWASP top 10 security issues are going to be explored for potential pitfalls within the Micflix project. The context for each of these risks will be explained with the information provided by the “OWASP Top Ten” webpage (2) .

2.1.1 Broken Access Control

2.1.1.1 Is the project vulnerable?

2.1.1.2 How to prevent?

2.1.2 Cryptographic Failures

2.1.2.1 Is the project vulnerable?

2.1.2.2 How to prevent?

2.1.3 Injection

2.1.3.1 Is the project vulnerable?

2.1.3.2 How to prevent?

2.1.4 Insecure Design

2.1.4.1 Is the project vulnerable?

2.1.4.2 How to prevent?

2.1.5 Security Misconfiguration

2.1.5.1 Is the project vulnerable?

2.1.5.2 How to prevent?

2.1.6 Vulnerable and Outdated Components

2.1.6.1 Is the project vulnerable?

2.1.6.2 How to prevent?

2.1.7 Identification and Authentication Failures

2.1.7.1 Is the project vulnerable?

2.1.7.2 How to prevent?

2.1.8 Software and Data Integrity Failures

2.1.8.1 Is the project vulnerable?

2.1.8.2 How to prevent?

2.1.9 Security Logging and Monitoring Failures

2.1.9.1 Is the project vulnerable?

2.1.9.2 How to prevent?

2.1.10 Server-Side Request Forgery

2.1.10.1 Is the project vulnerable?

2.1.10.2 How to prevent?

Owasp vulnerability table

In this table, the project's overall OWAP vulnerability risks will be categorized by the following labels: **Excellent security**, **Good security,** **Average security**, **Poor security**, and **Bad security**.

Index	OWASP RISKS	PROJECT VULNERABILITIES
2.1.1	**Broken Access Control**	
2.1.2	**Cryptographic Failures**	
2.1.3	**Injection**	
2.1.4	**Insecure Design**	
2.1.5	**Security Misconfiguration**	
2.1.6	**Vulnerable and Outdated Components**	
2.1.7	**Identification and Authentication Failures**	
2.1.8	**Software and Data Integrity Failures**	
2.1.9	**Security Logging and Monitoring Failures**	
2.1.10	**Server-Side Request Forgery**	

3 GDPR AND DATA COMPLEXITIES

3.1 GDPR

3.2 CAP THEOREM

The three letters in CAP stand for three desired features of distributed systems with replicated data: consistency among replicated copies, system availability for read and write operations, and partition tolerance.

The CAP theorem implies that in a distributed system with data replication, all three desirable features: consistency, availability, and partition tolerance cannot be guaranteed at the same time (see Figure 1 for a visual overview).

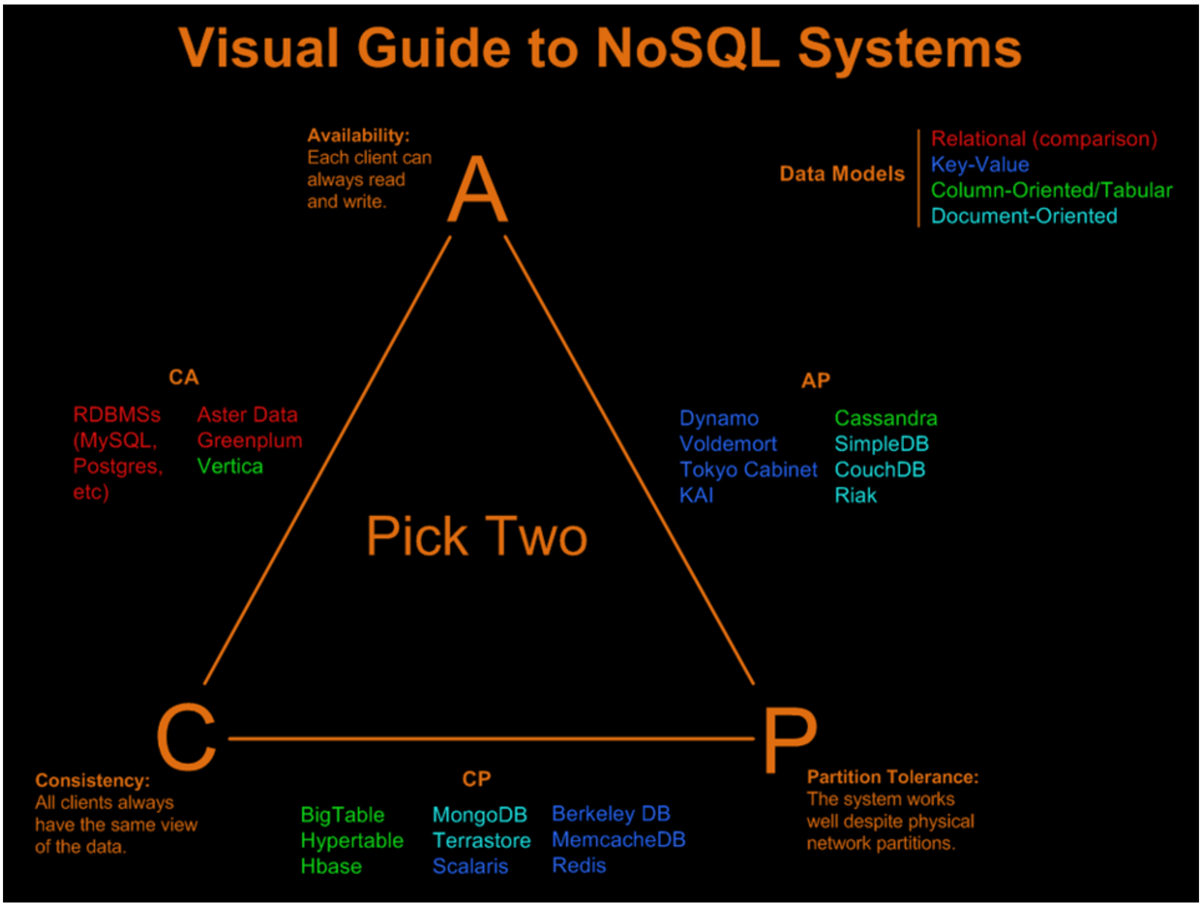


Figure 1 Fontys S-ESE6-CMK Module GDPR and Data Complexities- Theoretical Background on Canvas

4 CONCLUSION

5 BIBLIOGRAFIE

1. **Vliet, Hans van.** MoSCoW-methode. *wikipedia.* [Online] 2008. https://nl.wikipedia.org/wiki/MoSCoW-methode.

2. **OWASP Top Ten. *owasp.* [Online] 2022. https://owasp.org/www-project-top-ten/.**