



Linnæus University
Sweden

Report

Requirements Document



Author: Xingrong Zong
Supervisor: [Supervisor]
Semester: VT21
Course code: 2DV608



Table of contents

1	Introduction.....	3
1.1	<i>Purpose of the requirements document</i>	3
2	Scope of the product	4
2.1	<i>Stakeholders</i>	4
2.2	<i>Requirements</i>	5
2.3	<i>Requirements Analysis</i>	6
2.4	<i>Requirements Classification.....</i>	7
2.5	<i>Risk Assessment of Requirements.....</i>	9
2.6	<i>Systematic Validation of Requirements.....</i>	11
2.6.1	Are the requirements <i>complete</i> – does the checker know of any missing requirements or is there any information missing from individual requirement descriptions?.....	11
2.6.2	Are the requirements <i>consistent</i> – do the descriptions of different requirements include contradictions?	11
2.6.3	Are the requirements <i>comprehensible</i> – can readers of the documents understand what the requirements mean?	11
2.6.4	Are the requirements <i>ambiguous</i> – are there different possible interpretations of the requirements?	11
2.6.5	Is the requirements document <i>structured</i> – are the descriptions of requirements organized so that related requirements are grouped? ..	11
2.6.6	Are the requirements <i>traceable</i> – do the requirements include links to related requirements and to the reasons why these requirements have been included?.....	12
2.6.7	Does the requirements document as a whole, or do the individual requirements conform to defined <i>standards</i> ?	12
2.7	<i>Test Cases for Requirements</i>	13
2.7.1	TC 2.2.1.....	13
2.7.2	TC 2.2.2.....	13
2.7.3	TC 2.2.3.....	13
2.7.4	TC 2.2.4.....	13
2.7.5	TC 2.2.5.....	13
2.7.6	TC 2.2.6.....	13
2.7.7	TC 2.2.7.....	13
2.7.8	TC 2.2.8.....	13
3	Definitions, acronyms, and abbreviations	14
4	References.....	15
5	Overview of the document	16



1 Introduction

This requirements document is a report about designing and developing an online video streaming service for Netflix.

1.1 Purpose of the requirements document

The requirements document is an official statement of the system requirements for customers, end-users, and software developers.

How to use the requirements document as a target audience:

- System customers specify the requirements and read them to check that they meet their needs. Customers specify changes to the requirements.
- Managers use the requirements document to plan a bid for the system and to plan the system development process.
- System engineers use the requirements to understand what system is to be developed,
- System maintenance engineers use the requirements to understand the system and the relationships between its parts.
- System testers use the requirements to develop validation tests for the system.
- System end-users read the document to understand what the system provides and how the system works.



2 Scope of the product

2.1 Stakeholders

- End-users of the system – users who use Netflix to watch videos.
- Managers and others involved in the organizational processes – product manager / development manager, legal and compliance teams.
- Engineers responsible for the system development and maintenance – software developers and testers.
- Customers of the organization who will use the system to provide some services.
- External bodies – regulators or safety certification authorities.



2.2 Requirements

Unique Identifier	Requirements	Type of Requirements	Priority
2.2.1	The system should enable the user to watch a video.	Functional	Essential
2.2.2	The system should enable the user to choose the quality of a video.	Functional	Useful
2.2.3	The system should enable the user to choose the playback speed of a video.	Functional	Useful
2.2.4	The system must provide some facility for authenticating the identity of a system user.	Functional	Essential
2.2.5	The authentication process should be completed in 5 seconds or less.	Non-functional	Useful
2.2.6	The system should enable to provide a section with only children-friendly videos.	Non-functional	Useful
2.2.7	The system should be easy to use.	Non-functional	Desirable
2.2.8	The system should be easy to search videos.	Non-functional	Desirable



2.3 Requirements Analysis

The following is a checklist for Requirements Analysis:

1. Premature Design – Does the requirements include **premature** design or implementation **information**?
2. Combined Requirements – Could the description of a requirement be **broken down** into several different requirements?
3. Unnecessary Requirements – Is the requirement ‘**gold plating**’? That is, a cosmetic addition to the system which is not really necessary.
4. Use of Non-standard Hardware – Does the requirement mean that **non-standard** hardware or software must be used?
5. Conformance with Business Goals – Is the requirement **consistent** with the business **goals** defined in the introduction to the requirements document?
6. Requirements Ambiguity – Is the requirement **ambiguous**, i.e., could it be read in different ways by different people?
7. Requirements Realism – Is the requirement **realistic** given the technology which will be used to implement the system?
8. Requirements Testability – Is the requirement **testable**, that is, is it stated in such a way that test engineers can derive a test which can show if the system meets that requirement?

Checklist/Reqs	1	2	3	4	5	6	7	8
2.2.1	x				x		x	x
2.2.2	x						x	x
2.2.3	x						x	x
2.2.4	x						x	x
2.2.5	x						x	x
2.2.6	x							x
2.2.7	x	x				x		
2.2.8	x	x				x		



2.4 Requirements Classification

Requirements	Classification
2.2.1	<ul style="list-style-type: none">• System• User Interface• Database
2.2.2	<ul style="list-style-type: none">• System• User Interface• Communications
2.2.3	<ul style="list-style-type: none">• System• User Interface• Communications
2.2.4	<ul style="list-style-type: none">• System• User Interface• Database• Communications• Security
2.2.5	<ul style="list-style-type: none">• System• User Interface• Database• Communications• Security
2.2.6	<ul style="list-style-type: none">• System• User Interface• Database• Communications• Security
2.2.7	<ul style="list-style-type: none">• User Interface



	<ul style="list-style-type: none">• Communications
2.2.8	<ul style="list-style-type: none">• User Interface• Database• Communications



2.5 Risk Assessment of Requirements

The following is a checklist of type of risks in implementing a requirement:

1. Performance Risks – May affect the overall performance of the system.
2. Safety and Security Risks – May cause problems in meeting overall system requirements for safety and security.
3. Process Risks – May require changes to the normal development process.
4. Implementation Technology Risks – May require the use of unfamiliar implementation technology.
5. Database Risks – May involve non-standard data which is not available in an existing system database.
6. Schedule Risks – May be technically difficult and may threaten the planned development schedule for the system.
7. External Risks – Involves external contractors.
8. Stability Risks – Requirement may be volatile and subject to evolution during the development process.

Requirements	Type of Risks	Risk Assessment
2.2.1	<ul style="list-style-type: none">• External Risks	<ul style="list-style-type: none">• High
2.2.2	<ul style="list-style-type: none">• Implementation Technology Risks• Performance Risks• Stability Risks	<ul style="list-style-type: none">• High• Medium• High
2.2.3	<ul style="list-style-type: none">• Implementation Technology Risks• Performance Risks• Stability Risks	<ul style="list-style-type: none">• High• Medium• High
2.2.4	<ul style="list-style-type: none">• Safety and Security Risks• Process Risks	<ul style="list-style-type: none">• High• Low



	<ul style="list-style-type: none">• Database Risk	<ul style="list-style-type: none">• Medium
2.2.5	<ul style="list-style-type: none">• Process Risks• Implementation Technology Risks• Schedule Risks	<ul style="list-style-type: none">• Low• Medium• Medium
2.2.6	<ul style="list-style-type: none">• Performance Risks• Process Risks• Implementation Technology Risks• Schedule Risks• Stability Risks	<ul style="list-style-type: none">• Medium• Medium• High• High• High
2.2.7	<ul style="list-style-type: none">• Performance Risks	<ul style="list-style-type: none">• Medium
2.2.8	<ul style="list-style-type: none">• Performance Risks• Process Risks• Implementation Technology Risks	<ul style="list-style-type: none">• Medium• Low• Medium



2.6 Systematic Validation of Requirements

The following is a checklist for Systematic Validation:

2.6.1 Are the requirements *complete* – does the checker know of any missing requirements or is there any information missing from individual requirement descriptions?

The requirements are incomplete. There is information missing from individual requirement description, such as what is the definition of “easy to use”.

2.6.2 Are the requirements *consistent* – do the descriptions of different requirements include contradictions?

The requirements are consistent, there is not any contradictions.

2.6.3 Are the requirements *comprehensible* – can readers of the documents understand what the requirements mean?

The requirements are mostly comprehensible. The readers of the documents can understand what the most requirements mean.

2.6.4 Are the requirements *ambiguous* – are there different possible interpretations of the requirements?

The requirements are mostly unambiguous. The ambiguous parts, such as “easy to use” and “easy to search”.

2.6.5 Is the requirements document *structured* – are the descriptions of requirements organized so that related requirements are grouped?

The requirements document is structured. Related requirements such as 2.2.2 and 2.2.3 are grouped, 2.2.4 and 2.2.5 are grouped.



2.6.6 Are the requirements *traceable* – do the requirements include links to related requirements and to the reasons why these requirements have been included?

The requirements are traceable. For example, requirement 2.2.1 is the main request in the introduction.

2.6.7 Does the requirements document as a whole, or do the individual requirements conform to defined *standards*?

The requirements document conforms to defined standards.



2.7 Test Cases for Requirements

2.7.1 TC 2.2.1

Main Scenario:

1. The user can play a video.

2.7.2 TC 2.2.2

Main Scenario:

1. The user can choose the quality of a video that is playing.

2.7.3 TC 2.2.3

Main Scenario:

1. The user can choose the play speed of a video that is playing.

2.7.4 TC 2.2.4

Main Scenario:

1. The user can login.

2.7.5 TC 2.2.5

Main Scenario:

1. The user can login within 5 seconds.

2.7.6 TC 2.2.6

Main Scenario:

1. The user can login within 5 seconds.

2.7.7 TC 2.2.7

Main Scenario:

1. The user feel the system is easy to use.

2.7.8 TC 2.2.8

Main Scenario:

1. The user feel the system is easy to search videos.



3 Definitions, acronyms, and abbreviations

- Functional requirements = FRs
- Non-functional requirements = NFRs
- Essential – it **must** be **included** in the system.
- Useful – the system will be **less effective** without it.
- Desirable – it is **not** a **core** system facility but makes the system **more attractive** to users.
- System – Requirements that affect the entire system such as performance or reliability requirements.
- User Interface – Requirements that are concerned with user interaction.
- Database – Requirements that are concerned with the data managed by the system.
- Communications – Requirements that are concerned with the external communication facilities in the system.
- Security – Requirements that are concerned with the security of the system or the user.



4 References

- Sommerville.I, Sawyer.P, “Requirements Engineering – A Good Practice Guide”
- Lectures from 2DV608 in Linneaus University



5 Overview of the document

Netflix should let end-users be able to watch online videos with navigation searching tool and categorizes sections. Videos should have qualities, playback speed and subtitle for users to choose. Netflix system should also provide a special mode for only providing children-friendly videos for the user who has kid. To be able to use Netflix, user needs to become a member and pays the membership monthly.