
nappydevelopment

**Nappy, the ingenious
Software Requirements Specification**

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

Revision History

Date	Version	Description	Author
12/10/15	1.0	Erste Version des Dokuments	Mehmet Ali Incekara, Marc Mahler und Marvin Zerulla
13/10/15	1.1	Überarbeitung des Inhalts	Mehmet Ali Incekara,
16/10/15	1.2	Anfügen „Overall UCD“ und „User Interface Mockups“	Mehmet Ali Incekara, Marvin Zerulla und Manuel Bothner
23/10/15	1.3	Add 3 UC	Mehmet Ali Incekara
25/10/15	1.4	Update UC and Mockup Link	Mehmet Ali Incekara
29/10/15	1.5	Add UseCase: View Statistic and Change Settings	Mehmet Ali Incekara
05/11/15	1.6	Add ClassDiagram	Mehmet Ali Incekara
12/11/15	1.7	Add SAD	Mehmet Ali Incekara
28/11/15	1.8	Add Gantt-Chart	Mehmet Ali Incekara
18/12/15	1.9	Korrekturlesen für Midterm-Abgabe	Mehmet Ali Incekara

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, Acronyms, and Abbreviations	5
1.4	Document – References	5
2.	Overall Description	6
2.1	Product perspective	6
2.2	Product functions	6
2.3	User characteristics	6
2.4	Constraints	6
2.5	Assumptions and dependencies	6
3.	Specific Requirements	7
3.1	Functionality	7
3.1.1	Spiel starten	7
3.1.2	Statistik	7
3.1.3	Einstellungen	7
3.1.4	Beenden	7
3.1.5	Spielanleitung	7
3.1.6	Wiki & Add Character	7
3.1.7	Info	7
3.2	Usability	8
3.2.1	Training Time	8
3.2.2	Hardware Requirements	8
3.2.3	Software Requirements	8
3.3	Reliability	8
3.3.1	Availability	8
3.3.2	Mean Time Between Failures	8
3.3.3	Mean Time To Repair	8
3.3.4	Accuracy	8
3.3.5	„Maximum Bugs or Defect Rate“ and „Bugs or Defect Rate“	8
3.4	Performance	8
3.4.1	Response Time	8
3.4.2	Throughput	9
3.4.3	Capacity	9
3.4.4	Degradation modes	9
3.4.5	Resource utilization	9
3.5	Supportability	9
3.6	Design Constraint	9
3.7	On-line User Documentation and Help System Requirements	9
3.8	Purchased Components	9
3.9	Interfaces	9
3.9.1	User Interfaces	9
3.9.2	Hardware Interfaces	9
3.9.3	Software Interfaces	9
3.9.4	Communications Interfaces	9
3.10	Licensing Requirements	9
3.11	Legal, Copyright, and Other Notices	9
3.12	Applicable Standards	9

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

4. Supporting Information

10

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

Software Requirements Specification

1. Introduction

Das Projekt "Nappy, the ingenious" hat das Ziel eine Desktop-Applikation zu entwickeln. Es ist eine Art Quizspiel, in dem der Actor gegen „Nappy“ (das System) spielen muss.

Vision: <https://nappydevelopment.wordpress.com/2015/09/30/a-little-bit-more-about-our-project-vision/>

1.1 Purpose

Das SRS soll einen Überblick über die Anforderungen und Dokumente von unserem Projekt bieten.

1.2 Scope

Dieses Dokument dient zur internen Nutzung und soll als Richtlinien für die Entwicklung des Spiels gelten.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification
Actor	Spieler, Nutzer
Single-User	Die Anwendung unterstützt nur einen Anwender (nicht mehrere gleichzeitig)
UCD	Use Case Diagram
SAD	Software Architecture Document

1.4 Document – References

Blog	https://nappydevelopment.wordpress.com/
GitHub-Quellcode	https://github.com/nappydevelopment/Nappy-the-ingenious
GitHub-Docs	https://github.com/nappydevelopment/docs
Jira	http://193.196.7.27:8080/secure/RapidBoard.jspa?projectKey=NAP&rapidView=8&view=planning
Mockups	https://github.com/nappydevelopment/docs/blob/master/pdfs/Mockups.pdf
UseCase Gamemode 1	https://github.com/nappydevelopment/docs/blob/master/pdfs/1%20UseCase_Gamemode%201.pdf
UseCase View Wiki	https://github.com/nappydevelopment/docs/blob/master/pdfs/2%20UseCase_View_Wiki.pdf
UseCase View Instruction	https://github.com/nappydevelopment/docs/blob/master/pdfs/3%20UseCase_View_Instructions.pdf
UseCase Change Settings	https://github.com/nappydevelopment/docs/blob/master/pdfs/4%20UseCase_Change_Settings.pdf
UseCase View Info	https://github.com/nappydevelopment/docs/blob/master/pdfs/5%20UseCase_View_Info.pdf
Class Diagram	https://github.com/nappydevelopment/docs/blob/master/svg/Class%20Diagram.svg
SAD	https://github.com/nappydevelopment/docs/blob/master/pdfs/Software%20Architecture%20Document.pdf
Gantt-Chart until Midterm	https://github.com/nappydevelopment/docs/blob/master/SE_GanttChart.mpp https://github.com/nappydevelopment/docs/blob/master/pdfs/SE_GanttChart_UntilMidterm.pdf
Midterm – Präsentation	https://github.com/nappydevelopment/docs/blob/master/Midterm-PPP.pptx

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

2. Overall Description

Es wird grundsätzlich zwischen zwei Spielmodi unterschieden.

Im ersten Spielmodus denkt der Actor an eine Figur aus der Fernsehserie „The Simpsons“. Nappy versucht anhand von Fragen, welche der Actor mit „ja“, „nein“ oder „ich weiß nicht“ beantworten kann, die Figur zu erraten.

Im zweiten Spielmodus denkt Nappy an eine Figur und der Spieler muss anhand einer Liste von vorgegebenen Fragen die Figur erraten, indem Nappy auf die Fragen antwortet.

Derjenige mit weniger Versuchen gewinnt das Duell.

2.1 Product perspective

Das Quizspiel soll informativ sein und immer neue Inhalte bieten können. Das im Spiel integrierte Wiki bietet eine kompakte Übersicht über alle spielbaren Charakteren. Eine Funktion im Wiki soll es dem Actor ermöglichen neue Charakter hinzuzufügen, so dass die Anzahl der spielbaren Charaktere und somit die Vielfalt stetig wachsen kann.

2.2 Product functions

Semester 3:

- UseCase Gamemode 1
- UseCase View Wiki
- UseCase View Instruction
- UseCase Change Settings (Language)
- UseCase View Info

Semester 4 (geplant, aber noch nicht verbindlich):

- UseCase Gamemode 2
- UseCase Add Character
- UseCase Statistics
- UseCase Change Settings (übrigen)
- Ggf. weitere (optional)

2.3 User characteristics

Der Actor sollte sich für die Fernsehserie „The Simpsons“ interessieren um die Fragen beantworten zu können. Des Weiteren sollte er Spaß an Quizspielen haben.

2.4 Constraints

Es wird für das Spielen gegen Nappy keine Internetverbindung vorausgesetzt. Das Spiel ist komplett offline auf dem eigenen Rechner spielbar (ermöglicht durch eine embedded Datenbank).

2.5 Assumptions and dependencies

- IDE: IntelliJ und Eclipse
- Versionskontrolle: GitHub
- Scrum: JIRA
- Programmiersprache: Java 1.8 mit JavaFX
- Datenbank: H2 database
- Test: SikuliX, JUnit
- Build: Gradle

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

3. Specific Requirements

3.1 Functionality

3.1.1 Spiel starten

Der Actor kann das Spiel aus dem Hauptbildschirm heraus starten.

UseCase Gamemode 1:

https://github.com/nappydevelopment/docs/blob/master/pdfs/1%20UseCase_Gamemode%201.pdf

UseCase Gamemode 2:

(tbd, nächstes Semester)

Ein aktives Spiel kann jedes Zeit abgebrochen werden.

Hinweis: Abgebrochene Spiele werden nicht in die Statistik einbezogen.

3.1.2 Statistik

UseCase View Statistic:

(tbd, nächstes Semester)

3.1.3 Einstellungen

UseCase Change Settings (Language)

https://github.com/nappydevelopment/docs/blob/master/pdfs/4%20UseCase_Change_Settings.pdf

UseCase Change Settings (verbliebenden)

(tbd, nächstes Semester)

3.1.4 Beenden

Diese Funktion befindet sich in dem Menu: Spiel. Es beendet die ganze Applikation und auch ggf. das aktuell laufende Spiel.

3.1.5 Spielanleitung

Diese Funktion befindet sich auf dem Startbildschirm und in dem Menu: Hilfe. Es öffnet sich ein neues Fenster mit einer Spielanleitung.

Use Case View Instruction:

https://github.com/nappydevelopment/docs/blob/master/pdfs/3%20UseCase_View_Instructions.pdf

3.1.6 Wiki & Add Character

UseCase View Wiki:

https://github.com/nappydevelopment/docs/blob/master/pdfs/2%20UseCase_View_Wiki.pdf

UseCase Add Character:

(tbd, nächstes Semester)

3.1.7 Info

UseCase View Info:

https://github.com/nappydevelopment/docs/blob/master/pdfs/5%20UseCase_View_Info.pdf

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

3.2 Usability

3.2.1 Training Time

Dieses Spiel benötigt wenig Zeit, um sich mit dem Client des Spiels vertraut zu machen. Die Hauptfunktionen sind auf dem Hauptbildschirm angeordnet und weitere zusätzliche Funktionen über die beiden Menus erreichbar. Alle Buttons und Kennzahlen sind mit „sprechenden“ Icons ergänzt.

Das Spiel selbst benötigt ebenfalls keine Trainingszeit, da das Spiel ganz einfach über Mausinteraktionen zu steuern ist (Point-and-Click).

3.2.2 Hardware Requirements

Der Actor benötigt einen Computer (Bildschirm + Maus). Es wird nur zum Herunterladen des Spiels eine Internetverbindung benötigt.

3.2.3 Software Requirements

Es wird eine installierte Version von Java (JRE) (mind. Java 1.8) benötigt, um das Spiel starten zu können. Das Spiel kann auf jedem OS gestartet werden (Plattformunabhängig).

3.3 Reliability

3.3.1 Availability

Das Spiel arbeitet mit einer embedded Datenbank und braucht keine Verbindung zu irgendeinem Server nach dem Download. Deswegen muss das Spiel zu 100% (24h, 7Tage die Woche) spielbar sein, solange der Computer funktionsfähig ist.

Bugs können über die angegebene E-Mail schnell (innerhalb des Spieles) gesendet werden.

Aus diesem Grund fallen die nächsten Punkte (3.3.2, 3.3.3, 3.3.4) weg.

3.3.2 Mean Time Between Failures

not applicable (n/a)

3.3.3 Mean Time To Repair

not applicable (n/a)

3.3.4 Accuracy

not applicable (n/a)

3.3.5 „Maximum Bugs or Defect Rate“ and „Bugs or Defect Rate“

Die Punkte „Maximum Bugs or Defect Rate“ und „Bugs or Defect Rate“ wurden zu einem Punkt zusammengefasst, da eine direkte Unterscheidung nicht möglich war.

Alle Funktionen sollen zum Ende der Entwicklung ausführbar sein, so dass der Actor das Spiel ohne Einschränkungen spielen kann.

Es ist möglich, dass der Algorithmus zum erraten der Figur des Actors, beim Spielmodus 1 falsch liegt.

Bei falschen Eingaben oder zu vielen „Ich weiß nicht“-Eingaben wird der Algorithmus bei dieser Anzahl an möglichen Figuren auf keine bestimmte (100% zustimmende) kommen.

3.4 Performance

3.4.1 Response Time

Die Werte sind stark abhängig, wie Leistungsstark der Prozessor des Computers ist.

Zum aktuellen Zeitpunkt wurden folgende Zeiten festgelegt.

Start der Applikation:

Durchschnittlich: 3 bis 5 Sekunden

Maximum: 7 Sekunden

Interaktionen nach dem Start:

Durchschnittlich: <500 Milisekunde

Maximum: 1 Sekunde

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

3.4.2 Throughput

not applicable (n/a)

3.4.3 Capacity

„Nappy, the ingenious“ ist als eine Single-User Anwendung geplant, in der der Spieler gegen den Computer spielt.

3.4.4 Degradation modes

not applicable (n/a)

3.4.5 Resource utilization

Prozessor: max. 125MB

3.5 Supportability

not applicable (n/a)

3.6 Design Constraint

Die Anwendung ist eine Java Applikation.

Architektur, Class Diagramm, Datenbank Model kann aus unserem „Software Architecture Document“ entnommen werden.

3.7 On-line User Documentation and Help System Requirements

Grundsätzlich wird keine Anleitung benötigt, aber es wird eine Funktion geben, die eine Spielanleitung öffnet.

3.8 Purchased Components

not applicable (n/a)

3.9 Interfaces

3.9.1 User Interfaces

Mockups:

<https://github.com/nappydevelopment/docs/blob/master/pdfs/Mockups.pdf>

Screenshots:

(tbd, am Ende des 4. Semesters, nach Fertigstellung)

3.9.2 Hardware Interfaces

not applicable (n/a)

3.9.3 Software Interfaces

not applicable (n/a)

3.9.4 Communications Interfaces

not applicable (n/a)

3.10 Licensing Requirements

not applicable (n/a)

3.11 Legal, Copyright, and Other Notices

Der Code des Spiels ist öffentlich verfügbar und kann von jedem verwendet werden, der Interesse daran hat. Alle Simpsons Inhalte (Charaktere, etc.) gehören dem rechtmäßigen Besitzer Matt Groening.

GitHub: <https://github.com/nappydevelopment/Nappy-the-ingenuous>

3.12 Applicable Standards

Code Conventions by Oracle

Nappy, the ingenious	Version: 1.9
Software Requirements Specification	Date: 18/12/15

4. Supporting Information

not applicable (n/a)