

Wer wird Millionär

Anforderungsdokument

Lei Chen

Marcel Khan

Jendrik Wenke

5. Dezember 2018

Inhaltsverzeichnis

1	Ziel und Zweck des Dokuments	2
1.1	Projektbeschreibung	2
1.1.1	Kurzbeschreibung des Projekts	2
1.1.2	Zweck des Projekts	2
1.1.3	Hintergrund, Problemstellung, Motivation für das Projekt	2
1.1.4	Ziele des Projekts	2
1.1.5	Erfolgskriterien	3
2	Anforderungen	3
2.1	Funktionale Anforderungen	3
2.2	Wartungsanforderungen	8
2.3	Performanceanforderungen	9
2.4	Umgebungsanforderungen	10
3	Anhang	11
3.1	Abkürzungen	11
3.2	Glossar	11
3.3	Literatur	11

Rev.	Datum	Autor	Bemerkungen	Status
1	17.01.2018	LC, MK, JW	Erste Version	fertiggestellt
2	21.11.2018	LC, MK, JW	Sprintziele für den 2. Sprint, neue Anforderungen A-006b und A-007b, Schönheitskorrekturen, Glossar	fertiggestellt
3	05.12.2018	LC, MK, JW	neue Anforderungen für den 3. Sprint	fertiggestellt



1 Ziel und Zweck des Dokuments

Dieses Dokument beschreibt die Anforderungen einer Software aus der Veranstaltung Teamprojekt. Es handelt sich hierbei um die Systemdefinition, die der Auftragnehmer (das Team) für den Auftraggeber (Prof. Höppner) erstellt, sodass der Auftraggeber versteht und validieren kann, was das System leisten wird.

1.1 Projektbeschreibung

1.1.1 Kurzbeschreibung des Projekts

Das Projekt „Wer wird Millionär“ widmet sich der Umsetzung einer Software, welche mit Hilfe einer strukturierten Wissenssammlung (Wikidata), Fragen und dazugehörige Antworten generiert.

1.1.2 Zweck des Projekts

Der Zweck des Projektes ist eine automatisierte Generierung von sinnvollen Fragen und Antworten. Damit werden Redakteure von Quiz-Sendungen oder -apps wie „Wer wird Millionär“ entlastet.

1.1.3 Hintergrund, Problemstellung, Motivation für das Projekt

Die Suche nach sinnvollen Fragen mit passenden Antwortsets ist sehr arbeits- und zeitintensiv und die Komplexität, neue Fragen zu finden, steigt stetig. Deshalb soll die Generierung automatisiert geschehen.

1.1.4 Ziele des Projekts

Das Ziel des Projekts ist die Entwicklung einer Software welche einen Redakteur für eine Quizshow oder ein Quizspiel beim Finden und Erstellen von neuen Quizfragen mit Antworten unterstützen soll. Diese Fragen sollen unterschiedlich sein und aus verschiedenen Themenbereichen kommen. Desweiteren sollen sich auch die Art der Fragen unterscheiden. Die Software soll mit der Konsole bedient werden. Das Ziel für den ersten Sprint ist erstmal, Wissensfragen zu generieren und passende Antworten. Danach werden wir nach Möglichkeiten suchen, um Trickfragen zu finden, und dabei schauen inwiefern es mit den Mitteln, die uns zur Verfügung stehen, überhaupt machbar ist. Die Ergebnisse lassen sich in jedem Fall auch später noch erweitern – zum Beispiel im Rahmen einer Vorlesung oder eines weiteren Teamprojekts. Auch soll die Software einfach erweiterbar sein, um zum Beispiel neue Fragearten zu generieren. Die Herausforderung liegt zum einen darin, aus den riesigen Datenmengen wichtige Informationen zu extrahieren und diese zu sinnvollen und beantwortbaren Fragen zu formen, wobei exakt eine der 4 Antworten richtig sein muss. Außerdem sollen die Fragen verschiedene Fragenschwierigkeit haben.



1.1.5 Erfolgskriterien

Das Projekt gilt als erfolgreich, wenn eine funktionierende Software entwickelt wurde, die alle Anforderungen erfüllt und alle Testfälle besteht. Außerdem gehört zu einer fertigen Software auch die Dokumentation.

2 Anforderungen

2.1 Funktionale Anforderungen

ID	A-001
Anforderungstyp	Funktionale Anforderung
Anforderung	Das Programm soll Fragen und Antworten ausgeben.
Abnahmekriterium	Das Programm gibt Fragen und Antworten aus.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch (Kernfunktionalität)
Konflikte	–
ID	A-002
Anforderungstyp	Funktionale Anforderung
Anforderung	Das Programm soll zu jeder Frage eine richtige und 3 falsche Antworten ausgeben.
Abnahmekriterium	Manuelle Überprüfung, ob einige Ergebnisse stimmen.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch (Kernfunktionalität)
Konflikte	–

Team 1

Lei Chen Marcel Khan Jendrik Wenke



ID	A-003
Anforderungstyp	Funktionale Anforderung
Anforderung	Die Fragen und Antworten sollen auf deutsch sein, aber nicht grammatikalisch korrekt.
Abnahmekriterium	Die Sprache der ausgegeben Antworten wird manuell überprüft.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Normal
Priorität	Hoch
Konflikte	–
ID	A-004
Anforderungstyp	Funktionale Anforderung
Anforderung	Die Antworten auf eine Frage sollen einen hohes Ähnlichkeitsmaß besitzen.
Begründung	Ein Programm, welches Fragen generiert, deren Antworten keinerlei Ähnlichkeitsmaß besitzen, erfüllt nicht den Zweck, die Arbeit von Redakteuren zu erleichtern, da wir sonst einen sehr niedrigen Schwierigkeitsgrad haben, wenn die Antworten leicht durch Ausschlussverfahren zu erraten sind.
Abnahmekriterium	Manuelle Überprüfung, ob alle Antworten einiger Fragen theoretisch stimmen könnten.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch
Konflikte	–



ID	A-005
Anforderungstyp	Funktionale Anforderung
Anforderung	Zu einer Frage soll eine Fragenschwierigkeit zwischen 1 und 15 angegeben werden.
Begründung	Bei einer Sendung werden Fragen mit verschiedenen Fragenschwierigkeiten benötigt.
Abnahmekriterium	Manuelle Überprüfung, ob Fragen eine passende Fragenschwierigkeit zugeordnet ist. Dabei ist Schwierigkeit natürlich subjektiv und verschiedene Leute werden verschiedene Meinungen haben.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Normal
Priorität	Niedrig
Konflikte	–
ID	A-005b
Anforderungstyp	Funktionale Anforderung
Anforderung	Generierter Fragenkatalog soll möglichst gleichverteilte Fragenschwierigkeit haben.
Begründung	Bei einer Sendung werden Fragen mit verschiedenen Fragenschwierigkeiten benötigt. Wenn wir aber nur leichte oder nur schwere in unserem Fragenkatalog generieren, sind die Fragen nur bedingt nutzbar.
Abnahmekriterium	Manuelle Überprüfung, ob Fragen eine ausgeglichene Fragenschwierigkeit haben. Dabei ist Schwierigkeit natürlich subjektiv und verschiedene Leute werden verschiedene Meinungen haben.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Normal
Priorität	Niedrig
Konflikte	–

Team 1

Lei Chen Marcel Khan Jendrik Wenke



ID	A-006
Anforderungstyp	Funktionale Anforderung
Anforderung	Es muss verschiedene Arten von Fragen geben, und es soll möglich sein, nur bestimmte Arten zu generieren
Begründung	Wenn alle Fragen nach dem gleichen Schema generiert sind, ist die Software nicht hilfreich. Beispielsweise lassen sich alleine durch die Auswahl von 4 aus 16 Bundesländern eine große Zahl an Antwortsets generieren.
Abnahmekriterium	Manuelle Überprüfung, ob verschiedene Arten von Fragen generiert werden.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch
Konflikte	Wenn Generierung generalisiert wird, ist ein generieren von Spezifischen Arten von Fragen eventuell nicht mehr möglich.
ID	A-006b
Anforderungstyp	Funktionale Anforderung
Anforderung	Es muss mehr als 3 Arten von Fragen geben.
Begründung	Wenn es nur wenige Arten von Fragen gibt, dann können nur wenige für eine Sendung genutzt werden und die Fragen drohen sich zu wiederholen.
Abnahmekriterium	Manuelle Überprüfung, ob mehr als 3 Arten von Fragen generiert werden.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch
Konflikte	–

Team 1

Lei Chen Marcel Khan Jendrik Wenke



ID	A-006c
Anforderungstyp	Funktionale Anforderung
Anforderung	Generierte Fragen sollen nicht zu spezifisch auf Details ausgerichtet sein.
Begründung	Wenn Fragen generiert werden, die z.B nach Dingen wie Daten oder Straßen fragen, hat man nicht viel nützliches von den Fragenkatalogen.
Abnahmekriterium	Manuelle Stichpunktartige Überprüfung, ob Fragen nicht zu Spezifisch sind.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Groß
Priorität	Hoch
Konflikte	–
ID	A-007
Anforderungstyp	Funktionale Anforderung
Anforderung	Es sollte Trickfragen geben.
Begründung	Am Anfang der Sendung werden Fragen gestellt, die wenig Wissen erfordern, aber bei denen man z.B. wegen ähnlich klingender Wörter um die Ecke denken muss.
Abnahmekriterium	Das Programm sollte bei Restriktionen auf diese Arten von Fragen Ergebnisse liefern.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Klein
Priorität	Niedrig
Konflikte	–



ID	A-007b
Anforderungstyp	Funktionale Anforderung
Anforderung	Die Software sollte ohne Internetverbindung funktionieren.
Begründung	Ein wichtiger Teil dieser Software ist der Umgang mit großen Datenmengen. Wenn die API von Wikidata genutzt wird, geht damit das Merkmal dieses Projektes verloren.
Abnahmekriterium	Das Programm funktioniert ohne Internetverbindung mit einem lokalen Dump.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Mittel
Priorität	Normal
Konflikte	–

2.2 Wartungsanforderungen

ID	A-008
Anforderungstyp	Wartungsanforderung
Anforderung	Es soll relativ einfach für Entwickler sein, neue Fragentypen in das bestehende Programm zu integrieren.
Begründung	Die Fragen von einem Typ werden irgendwann ausgeschöpft sein. Durch sinnvolle Schnittstellen innerhalb der Software ist diese leicht anzupassen.
Abnahmekriterium	Evaluation der Schnittstelle von allen Entwicklern, während das Programm entwickelt wird.
Anforderer	SM
Kundenzufriedenheit	Klein
Priorität	Niedrig
Konflikte	–

**2.3 Performanceanforderungen**

ID	A-009
Anforderungstyp	Performanceanforderung
Anforderung	Das Programm soll auf gängigen Arbeitsrechnern zumindest bei reduziertem Datensatz innerhalb von absehbarer Zeit Ergebnisse liefern.
Begründung	Wenn das Programm nicht in absehbarer Zeit Ergebnisse liefert, verringert sich der Nutzen, den man aus dem Programm ziehen kann stark. Das Ziel des Projektes ist es, Fragen einfacher und schneller generieren zu können als ein Mensch das könnte. Dieser Vorteil wäre dann nicht vorhanden.
Abnahmekriterium	Evaluation der Schnittstelle von allen Entwicklern, während das Programm entwickelt wird.
Anforderer	SM
Kundenzufriedenheit	Klein
Priorität	Niedrig
Konflikte	–
ID	A-009b
Anforderungstyp	Performanceanforderung
Anforderung	Das Programm soll mit gängiger Arbeitsspeichergröße laufen und Ergebnisse liefern.
Begründung	Wenn das Programm nicht in absehbarer Zeit Ergebnisse liefert, verringert sich der Nutzen, den man aus dem Programm ziehen kann stark. Desweiteren hilft es nichts, wenn ein Programm theoretisch viele Fragen generieren kann, aber nicht läuft, weil der Rechner nicht genug Arbeitsspeicher hat.
Abnahmekriterium	Evaluation der Schnittstelle von allen Entwicklern, während das Programm entwickelt wird.
Anforderer	SM
Kundenzufriedenheit	Groß
Priorität	Hoch
Konflikte	–



2.4 Umgebungsanforderungen

ID	A-010
Anforderungstyp	Umgebungsanforderung
Anforderung	Das Programm soll unter Linux laufen.
Begründung	Alle Teammitglieder und Herr Höppner verwenden Linux.
Abnahmekriterium	Das Team entwickelt und testet unter Linux. Es würde auffallen, wenn das Programm nicht läuft.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Normal
Priorität	Hoch
Konflikte	–
ID	A-011
Anforderungstyp	Umgebungsanforderung
Anforderung	Das Programm soll in Java geschrieben und mit Maven compilierbar sein.
Begründung	Die Library „Wikidata-Toolkit“ ist in Java geschrieben.
Abnahmekriterium	Das Team arbeitet mit Maven. Fehler würden auffallen.
Anforderer	Prof. Höppner
Kundenzufriedenheit	Normal
Priorität	Hoch
Konflikte	–



3 Anhang

3.1 Abkürzungen

SM Scrum Master. 8, 9

3.2 Glossar

Entity Eine Entity ist ein Objekt, Konzept oder Ereignis in Wikidata (frei übersetzt nach[Wik]). 11

Fragenschwierigkeit Eine natürliche Zahl zwischen 1 und 15 (einschließlich). Repräsentiert die Wahrscheinlichkeit, dass eine Frage mit dieser Schwierigkeit von einer zufälligen Person beantwortet werden kann. Dabei steht 1 für „sehr einfach“ und 15 für „sehr schwierig“.. 2, 5

Knoten Entity. Entitys bilden aus graphentheoretischer Sicht einen Graphen, bei dem Entitys die Knoten sind. 11

Ähnlichkeitsmaß Zwei Entitys gelten als ähnlicher, je weniger Schritte über „instance of“- und „subclass of“-Beziehungen benötigt werden, um zu einem gemeinsamen Knoten zu gelangen. 4

3.3 Literatur

[Wik] *Wikidata:Glossary*. <https://www.wikidata.org/wiki/Wikidata:Glossary#Item>. – Abgerufen am 16.11.2018