



Bachelorarbeit

AUTOMATISIERBARER ALGORITHMUS ZUR BEWERTUNG VON CROWDSOURCING-NUTZERN UND DEREN SUBMISSIONS

Tommy Kubica
Mat.-Nr.: 3680797

Betreut durch:
Dipl.-Inf. Tenshi Hara
und:
Dr.-Ing. Thomas Springer
Eingereicht am 30. August 2013

DANKSAGUNG

Ich bedanke mich bei Herrn Dr. Thomas Springer, durch den die Vergabe der Bachelorarbeit erst ermöglicht wurde und als mein Nebenbetreuer während der Arbeit aktiv war.

Einen besonderen Dank möchte ich Herrn Tenshi Hara aussprechen, der mir als Hauptbetreuer jederzeit mit Rat und Tat zur Seite stand und mir bei der Erstellung der Arbeit sehr geholfen hat. Zudem möchte ich mich bei Herrn Professor Alexander Schill bedanken, der als zuständiger Hochschullehrer agierte.

Ein weiterer Dank geht einerseits an Herrn Andreas Weigand, der mir als Ansprechpartner des ZIHs jederzeit für Fragen bereitstand, sowie an Herrn Dr. Dietbert Gütter, der einen interessanten Vorschlag einer expliziten Methode in den Raum geworfen hat.

Zu guter Letzt möchte ich mich bei meinen Eltern bedanken, die jederzeit mein Studium unterstützten und mir diese Arbeit erst ermöglichten.

ABSTRACT

Crowdsourcing ist ein Gebiet der Informatik, das ständig an Wichtigkeit gewinnt. Nicht nur bei der Erhebung von Daten, auch bei der Aktualisierung von bereits vorhandenen Daten, findet sich dadurch eine Möglichkeit, die durch ihre Einfachheit, Kostengünstigkeit und Qualität beeindruckt. Dabei zieht sich Crowdsourcing durch verschiedenste Gebiete, von der Erstellung von Werbespots beim Super Bowl bis hin zur Unterstützung bei überlebenswichtigen Hilfseinsätzen infolge von Naturkatastrophen. Um die Qualität von Crowdsourcing zu gewährleisten, wurde in dieser Arbeit ein Algorithmus entworfen, mit dem Submissions von Crowdsourcing-Benutzern, sowie die Benutzer selbst, bewertet werden können. Das ist ein wichtiger Schritt, beispielsweise im MapBiquitous-Projekt der TU Dresden, bei dem durch Crowdsourcing Daten über verschiedene Positionen auf Karten im Outdoor- und Indoorbereich erhoben werden sollen. Der Bewertungsalgorithmus wurde auf Grundlage von bereits vorhandenen, sehr gut funktionalen, Bewertungssystemen entwickelt, von denen essentielle Aspekte übernommen wurden. Das entwickelte Konzept besteht aus den fünf großen Bestandteilen: Alterungsprozess, Datenbankprüfung, Datenbestandvergleich, Warteschlangenvergleich und Neuberechnung der Benutzerbewertung, die zusammen den eigentlichen Bewertungsalgorithmus schaffen. Das Konzept wurde beispielhaft umgesetzt und die Funktionalität und Effizienz in der folgenden Evaluation gezeigt.



AUFGABENSTELLUNG FÜR DIE BACHELOR-ARBEIT

THEMA: Automatisierbarer Algorithmus zur Bewertung von Crowdsourcing-Nutzern und deren Submissions

Name, Vorname:	Kubica, Tommy	Studiengang:	Medien-Inf. (Bachelor)
Matrikel-Nummer:	3680797	Projekt/Schwerpunkt:	MapBiquitous/Mobile
Erster Gutachter:	Dipl.-Inf. Tenshi Hara	Zweiter Gutachter:	Dr.-Ing. Thomas Springer
Beginn am:	07.06.2013	Einzureichen bis:	30.08.2013

ZIELSTELLUNG

In vielen Anwendungsgebieten gestaltet sich die Erfassung von Informationen als herausfordernd, insbesondere wenn die zu Datenerfassung und Verwaltung einen nicht unerheblichen Zeitaufwand bedeuten oder Quasiechtzeitanforderungen bestehen. Als Lösungsmöglichkeit hat sich in den letzten Jahren das Crowdsourcing angeboten. Dabei erledigt eine dynamische Gruppe variabler Größe (Crowd) die Aufgabe der Informationsaggregation.

Am Lehrstuhl Rechnernetze wurde eine Crowdsourcing-basierte Proxy-Struktur integriert, welche Nutzer-basierte Eingaben zwischen Clients (Crowd) und Servern (Crowdfunder) prokuriert. Die crowdgesourceten Daten (Submissions) sind derzeit an die einzelnen Nutzer gebunden, aber eine Bewertung der Submissions/Nutzer erfolgt derzeit nicht, ist aber mit Rücksicht auf sinnlose oder gar mutwillig falsche Submissions, sowie die automatisierte Ergänzung und/oder Korrektur der Basisdatensätze wünschenswert.

Ziel der Belegarbeit ist die Untersuchung von Möglichkeiten, Submissions an sich, und über die Submissions die einzelnen Nutzer, in Bezug auf die Qualität ihrer Beiträge algorithmisch zu bewerten.

Auf Basis des/r konzipierten Algorithmus/en soll die Umsetzung eines neuen Einsatzszenarios ermöglicht werden, nämlich die Erfassung und Mitteilung der Besucherfülle einer Mensa auf dem Campus der TU Dresden. Des Weiteren soll eine einfache, nachvollziehbare Möglichkeit für Crowdfunder geschaffen werden, Nutzern den Zugang zum Crowdsourcing zu verweigern.

SCHWERPUNKTE

- Recherche verwandter Arbeiten zum Crowdsourcing und zu Bewertungsverfahren
- Schaffung oder Übernahme eines (automatisierbaren) Bewertungsalgorithmus
- Prototypische Umsetzung des Konzepts
- Erarbeitung einer Evaluationsmethodik und Evaluation der Ergebnisse

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill
(verantwortlicher Hochschullehrer)

ERKLÄRUNG

Ich erkläre, dass ich die vorliegende Arbeit selbständig, unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Dresden, 30. August 2013

INHALTSVERZEICHNIS

1	Einleitung	1
1.1	Motivation	1
1.2	Zielstellung der Arbeit	2
1.3	Aufbau der Arbeit	2
2	Grundlagen und Vorüberlegungen	5
2.1	Crowdsourcing	5
2.1.1	Verwendete Definition	6
2.1.2	Arten von Crowdsourcing	7
2.2	Ansätze zum Crowdsourcing-Anwendungsfall	8
3	Bewertungssysteme	11
3.1	Sinn und Zweck von Bewertungen	11
3.2	Art des Bewertungsverfahrens	11
3.3	Existierende Bewertungsverfahren	12
3.3.1	YouTube	12
3.3.2	eBay	14
3.3.3	Stack Overflow	16
3.4	Zusammenfassung	18

4	Diskussion möglicher Lösungsansätze	19
4.1	MAC-Ansatz	19
4.2	IP-Ansatz	21
4.3	WLAN-Ansatz	22
4.4	Akustischer Ansatz	23
4.5	Expliziter Ansatz	24
4.6	Zusammenfassung	25
5	Konzeption des Algorithmus	27
5.1	Anforderungsanalyse	27
5.2	Entwicklung eines Konzepts	29
5.3	Zusammenfassung	40
6	Implementierung	41
6.1	Probleme und Schwierigkeiten	41
6.2	Umsetzung	43
6.3	Verwendete Technologien und Programme	45
6.4	Zusammenfassung	45
7	Evaluation	47
7.1	Überprüfung der Anforderungsanalyse	47
7.1.1	Der Algorithmus an sich	48
7.1.2	Festlegung von Werten durch Crowdfunder	49
7.1.3	Möglichkeiten des Crowdfunders	49
7.2	Simulation des Anwendungsfalls	50
7.2.1	Vorüberlegungen und Vorbereitungen	50
7.2.2	Durchführung	54
7.2.3	Auswertung	55
7.3	Test unter Laborbedingungen	60
8	Zusammenfassung und Ausblick	63
A	Anhang	i

ABBILDUNGSVERZEICHNIS

2.1	Grundriss der Alten Mensa	8
3.1	YouTube - Beispiel einer positiven Bewertung eines Videos	12
3.2	YouTube - Beispiel einer negativen Bewertung eines Videos	13
3.3	YouTube - Die Kommentarfunktion	13
3.4	YouTube - Die Bewertungsfunktion von Kommentaren	13
3.5	eBay - Beispiel von Benutzerbewertungen	14
3.6	eBay - Detailansicht eines Benutzers	15
3.7	Stack Overflow - Beispiel einer gestellten Frage mit Bewertung	16
3.8	Stack Overflow - Beispiel einer Top-Antwort	16
3.9	Stack Overflow - Beispiel eines Benutzers und Möglichkeiten Ansehen zu bekommen	17
4.1	MAC-Ansatz - Beispiel einer Umsetzung unter Windows	20
4.2	IP-Ansatz - Beispielapp Fing-Network Tools	21
4.3	WLAN-Ansatz - Beispiel eines erzeugten Graphs	22
4.4	Akustischer Ansatz - Mögliche Zusammenhänge zwischen der Lautstärke und der Besucherzahl	23
4.5	Expliziter Ansatz - Beispiel einer möglichen Umsetzung	24
5.1	Konzeption- Zusammenhang zwischen positiver Submissionbewertung und Benutzerbewertung	30

5.2	Konzeption - Zusammenhang zwischen negativer Submissionbewertung und Benutzerbewertung	31
5.3	Konzeption - Zusammenhang zwischen Benutzerbewertung und der vergangenen Zeit seit der letzten Submission	31
5.4	Konzept - Graph eines Sperrungsverlaufs	32
5.5	Algorithmus - Gesamtübersicht	33
5.6	Algorithmus - Gesamtübersicht	35
5.7	Algorithmus - Datenbankvergleich	36
5.8	Algorithmus - Datenbestandsvergleich	37
5.9	Algorithmus - Warteschlangenvergleich	39
5.10	Algorithmus - Benutzerneuberechnung	40
6.1	Implementierung - Veränderung am Algorithmus	42
7.1	Evaluation - Vergleich von normalem Ergebnis mit dem Ergebnis vom Blindtest . .	56
7.2	Evaluation - Vergleich der Messergebnisse mit dem Ergebnis des Fragebogens . .	56
7.3	Evaluation - Auswertung des Fragebogens mit Beispielwerten I	57
7.4	Evaluation - Auswertung des Fragebogens mit Beispielwerten II	58

TABELLENVERZEICHNIS

4.1	Zusammenfassung der Vor- und Nachteile möglicher Ansätze	25
6.1	Sound-Ansatz - Aufzeigen der Abweichungen bei einer Messung	43
6.2	Implementierung - Entscheidungstabelle	45
7.1	Evaluation - Prüfung der Anforderungsanalyse I	48
7.2	Evaluation - Prüfung der Anforderungsanalyse II	49
7.3	Evaluation - Prüfung der Anforderungsanalyse II	49
7.4	Evaluation - Verteilung der Intervalle	51
7.5	Evaluation - Berechnung der arithmetischen Mittel für die Intervalle	51
7.6	Evaluation - Ergebnis der Verteilung	52
7.7	Evaluation - Besucheranzahl der einzelnen Intervalle	53
7.8	Evaluation - Verteilung der Fragen im Blindtest	54
7.9	Evaluation - Ergebnis des Fragebogens	55
7.10	Evaluation - Einfluss des Motivationsfaktors	58
7.11	Evaluation - Einfluss der positiven Bewertung	59
7.12	Evaluation - Einfluss der negativen Bewertung	59
7.13	Evaluation - Einfluss des Prozentsatzes	59
7.14	Evaluation - Einfluss des Prozentsatzes	60
7.15	Evaluation - Test unter Laborbedingungen I	60

7.16	Evaluation - Test unter Laborbedingungen II	61
7.17	Evaluation - Test unter Laborbedingungen III	61
A.1	Anhang - Tabelle vom WLAN-Management I	iii
A.2	Anhang - Tabelle vom WLAN-Management II	v
A.3	Anhang - Verteilungstabelle	vi

1 EINLEITUNG

1.1 MOTIVATION

Der in den letzten Jahren stattgefundenene Käufertrend zu immer weiter entwickelten, so genannten Smartphones, hat ebenfalls eine Wende in der Art der Anwendungen mit sich gebracht. Moderne Anwendungen machen sich eine Vielzahl von eingebauten Sensoren zu Nutze, wodurch Benutzer beispielsweise in der Lage sind, von ihrer Position abhängige Anwendungen zu nutzen (Location Based Services).

Da bei diesen Anwendungen die Aktualität der Daten essentiell ist, ist es nötig, eine effektive Methode zu nutzen, um die Daten, die von der Anwendung genutzt werden, stets aktuell zu halten. Eine der fortschrittlichsten Methoden ist die Einbeziehung der Benutzer, wodurch Daten aktualisiert und für andere Benutzer verfügbar gemacht werden können. Dabei reden wir von Crowdsourcing. Am Lehrstuhl Rechnernetze der TU Dresden wurde bereits eine Crowdsourcing-basierte Proxy-Struktur entworfen, die Nutzerbasierte Eingaben zwischen Clients und Servern prokurirt.

Diese Methode der Datenerhebung und -aktualisierung ist funktional, solange alle Benutzer wahrheitsgetreue Beiträge tätigen. Um das zu gewährleisten muss jeder Benutzer, der ein Teil der Crowd ist und an der Aktualisierung und Erhebung der Daten beteiligt ist, vom System bewertet werden. Bewertungsverfahren sind dazu da, Qualität zu fördern und sinnlose oder gar falsche Beiträge zu entfernen. Dadurch ist es möglich, Benutzern stets qualitativ hochwertige Ergebnisse zu liefern, wonach jede Anwendung strebt. Verschiedenste Dienste im Internet benutzen bereits Bewertungsverfahren für Benutzer, Beiträge oder auch zur Einschätzung des Transaktionsverhaltens, wodurch sie Sicherheit und Qualität gewährleisten und damit teilweise zum Vorreiter in ihrer Branche geworden sind. Denken wir beispielsweise an eBay als größtes Online-Auktionshaus der Welt, YouTube als größtes Online-Videoportal oder auch Stack Overflow als eine der bekanntesten und qualitativ hochwertigsten Frage- und Antwortseiten.

Bewertungsverfahren werden somit für Benutzer entwickelt, die den Dienst, den sie anfordern, optimal benutzen wollen und helfen bei der Unterscheidung von Benutzern mit positiven und negativen Absichten. Ziel dieser Arbeit ist es, eine Möglichkeit zu entwickeln, Crowdsourcing-Nutzer und ihre Beiträge, die wir im Folgenden als Submissions bezeichnen, algorithmisch zu bewerten, um so eine Grundlage für weitere Arbeiten zu schaffen, die dieses Konzept im MapBiquitous-Projekt der TU Dresden integrieren.

1.2 ZIELSTELLUNG DER ARBEIT

Wie bereits in der Motivation angedeutet, ist der zentrale Punkt dieser Arbeit der Benutzer mit seinen Submissions. Ziel ist es, einen automatisierbaren Algorithmus zur Bewertung von Crowdsourcing-Nutzern und deren Submissions in Bezug auf die Qualität ihrer Beiträge zu entwerfen. Damit sollen die crowdgesourceten Daten (Submissions) auf sinnlose oder gar mutwillig falsche Submissions untersucht werden, sowie der bestehende Basisdatensatz automatisch ergänzt und/oder korrigiert werden.

Dazu ist eine umfassende Recherche verwandter Arbeiten zum Thema Crowdsourcing nötig, ebenso zu eventuell hilfreichen Bewertungsverfahren. Anschließend soll es zur Schaffung oder Übernahme eines Bewertungsalgorithmus kommen. Letztendlich soll der entwickelte oder übernommene Bewertungsalgorithmus prototypisch umgesetzt und anschließend evaluiert werden.

Die Funktionalität des konzipierten Algorithmus wird bei der Umsetzung eines neuen Einsatzszenarios getestet, wobei durch Crowdsourcing die Besucherfülle einer Mensa der TU Dresden erfasst und mitgeteilt wird. Zudem soll für den Crowdfunder eine Möglichkeit geschaffen werden, Nutzern den Zugang zum Crowdsourcing zu verweigern.

Ein großes Ziel der Arbeit wird sein, zu gewährleisten, dass der entwickelte Algorithmus einfach nachvollziehbar, sowie leicht erweiterbar ist. Damit soll die Möglichkeit für weitere Arbeiten gegeben werden, an dieser Arbeit anzusetzen und das entwickelte Konzept in das MapBiquitous-Projekt der TU Dresden einzubringen, wobei durch Crowdsourcing verschiedene Daten erhoben werden sollen.

Die Schwerpunkte der Arbeit noch einmal stichpunktartig zusammengefasst:

- Recherche verwandter Arbeiten zum Crowdsourcing und zu Bewertungsverfahren,
- Schaffung oder Übernahme eines (automatisierbaren) Bewertungsalgorithmus,
- Entwicklung eines einfach nachvollziehbaren, leicht erweiterbaren Gesamtkonzepts,
- Prototypische Umsetzung des Konzepts und
- Erarbeitung einer Evaluationsmethodik und Evaluation der Ergebnisse.

1.3 AUFBAU DER ARBEIT

Kapitel 2 soll sich mit den Grundlagen zum Thema beschäftigen, um einen kurzen Einblick in die Materie des Crowdsourcings zu geben und die Umsetzung des zuvor genannten Anwendungsfalles zu diskutieren. Dabei werden verschiedene implizite Ansätze, sowie ein expliziter Ansatz, vorgestellt, um Stärken und Schwächen verschiedener Methoden aufzuzeigen. Kapitel 3 beschäftigt sich mit verschiedenen bereits vorhandenen Methoden zur Bewertung einiger sehr bekannter Beispiele. Dabei soll nach Ideen Ausschau gehalten werden, die essentiell für den zu entwickelnden Algorithmus sein können. Im abschließenden Teil des Kapitels sollen die gesammelten Ideen zusammengefasst und bewertet werden, in welcher Art und Weise sie für den Algorithmus von Vorteil sein könnten. In Kapitel 4 werden die aus Kapitel 2 vorgestellten Ansätze zum Crowdsourcing-Anwendungsfall näher betrachtet und auf ihre Umsetzbarkeit geprüft. Dabei wird jeder Ansatz kurz vorgestellt, Vorteile und Probleme genannt, sowie eine Schlussfolgerung daraus gezogen. Ziel des Kapitels ist der Vergleich der verschiedensten Ansätze und die Auswahl eines Ansatzes, der später umgesetzt werden soll. In Kapitel 5 findet die Konzeption statt, wobei zuerst die aus Kapitel 3 gesammelten Ideen wiederaufgegriffen werden und eine Anforderungsanalyse durchgeführt und anschließend ein Gesamtkonzept des Algorithmus einwickelt wird. Im folgenden Kapitel werden die Implementierung des Algorithmus realisiert, die aufgetretenen Probleme

und Schwierigkeiten genannt und einige essentielle Stellen, die den Schwerpunkt des Algorithmus bilden, vorgestellt. Abschließend wird die Allgemeingültigkeit des Algorithmus diskutiert, um die Übertragbarkeit auf andere Crowdsourcing-Anwendungsfälle zu zeigen. In Kapitel 7 findet die Evaluation statt. Dabei wird der Algorithmus am zuvor ausgedachten Anwendungsfall getestet und bewertet. Das nächste Kapitel beschäftigt sich mit einer Zusammenfassung der Arbeit, sowie einem Ausblick auf weitere noch zu erledigende Arbeiten.

2 GRUNDLAGEN UND VORÜBERLEGUNGEN

Im folgenden Abschnitt soll eine Einführung in die Materie des Crowdsourcings gegeben werden. Dabei werden die zwei verschiedenen Crowdsourcing-Arten, implizites und explizites Crowdsourcing, sowie verschiedene Ansätze der Umsetzung des Anwendungsfalles „Bestimmung des Füllstandes der Mensa“ vorgestellt. Hierbei wird auf die Kernpunkte der Methode eingegangen, sowie auf die mögliche Art des zu verwendenden Crowdsourcings. Abschließend erfolgt ein Ausblick, der einen Einblick in die folgende Arbeit geben wird.

2.1 CROWDSOURCING

Crowdsourcing ist ein Begriff der Informatik, der erst in den letzten Jahren immer mehr in den Vordergrund der Aufmerksamkeit gerückt ist. Das Projekt SETI@home¹ zählt zu einem der ersten Crowdsourcing-Projekte, wodurch das Konzept erstmals in großem Rahmen angewandt wurde. Da der University of California Berkeley die nötige Rechenzeit fehlte, um die Masse an Daten, die sie von ihrem Radioteleskop in Arecebo gesammelt haben, zu verarbeiten, wurde nach einer Lösung für dieses Problem gesucht. Dabei wurde SETI@home ins Leben gerufen, wobei Leute sich Arbeitspakete herunterladen und die Leerlaufzeiten ihrer CPU sinnvoll nutzen konnten, indem sie Berechnungen durchführen ließen und die Ergebnisse anschließend zurückschickten. Dabei beschreibt Crowdsourcing die Auslagerung von Betriebsmitteln, in diesem Fall des Betriebsmittels Rechenzeit.

Ein anderes aktuelleres Beispiel, bei dem Crowdsourcing angewendet und so einer breiteren Öffentlichkeit vorgestellt wurde, war der Oil Spill im Golf von Mexiko im Jahr 2010. Laut [Zis11] war die Betreiberfirma BP ratlos, wie sie das Leck stopfen sollte, woraufhin die Öffentlichkeit aufgefordert wurde, ihre Lösungsvorschläge zuzusenden. Dabei gingen 43.000 konkrete Vorschläge aus über 100 Ländern ein, wie das Problem zu lösen sei. Dies konnte zwar nicht dazu beitragen, dass das Leck frühzeitig geschlossen wurde, aber es stellt ein gutes Beispiel dar, bei dem das Konzept, ein Problem von einer Crowd lösen zu lassen, im großen Stil angewendet wurde.

Um zu zeigen, dass durch Crowdsourcing auch Lösungen gefunden werden und die hohe Qualität, die durch Crowdsourcing entstehen kann, zu beweisen, wird eines der bekanntesten Sportereignisse betrachtet. Der Super Bowl² sorgt jährlich in den Vereinigten Staaten für die höchsten Einschaltquoten des Jahres. Ebenso teuer sind die Werbeplätze beim Super Bowl. Ein Artikel von

¹ <http://setiathome.berkeley.edu/> (aufgerufen am 22.08.2013)

² Das Finale der US-amerikanischen American-Football-Profiliga National Football League (NFL).

socialnetworkstrategien.de [Ros10] berichtet dabei über den Einsatz von Crowdsourcing zur Entwicklung von Werbespots und die Qualität dieser Beiträge. Doritos³ hat im Jahr 2010 erneut seine Spots über Crowdsourcing gesucht, anstatt sie von einer Agentur produzieren zu lassen. Dabei lockte man mit hohen Geldsummen für die Gewinner und noch höheren Geldsummen, wenn der Spot im USA Today Ad Meter einen der ersten drei Plätze belegte. Letztendlich wurden dabei vier Siegerspots entwickelt, die allesamt als „verdammt gut“ vom Autor eingeschätzt wurden. Einer der Doritos-Spots landete letztendlich sogar auf Platz 2 im Ad Meter von USA Today, wodurch die Macher des Spots zusätzlich 600.000 Dollar erhielten. „Mit Crowdsourcing kann Gutes entstehen! Wenn man es richtig anpackt“ [Ros10].

2.1.1 Verwendete Definition

Da für Crowdsourcing noch keine einheitliche Definition vorhanden ist, wird nach einer Methode gesucht, Crowdsourcing so zu definieren, dass alle wichtigen Aspekte abgedeckt sind und die Definition als allgemeingültig angesehen werden kann. Dabei ist es wichtig, dass eine Vielzahl an Definitionen betrachtet wird und anschließend eine einheitliche Definition entwickelt wird, die alle Kernpunkte der Definitionen enthält. Enrique Estellés-Arolas und Fernando González-Ladrón-de-Guevara haben in ihrer Arbeit „Towards an integrated crowdsourcing definition“ [EG12] 40 verschiedene Crowdsourcing-Definitionen analysiert und die Gemeinsamkeiten aller Definitionen zusammengefasst. Die in der Arbeit gefundene Definition wurde in Gerd Bombachs Belegarbeit [Bom12] übersetzt und wird hier übernommen:

Definition 2.1 (Crowdsourcing)

Unter Crowdsourcing versteht man die Aufforderung eines Individuums, einer Institution, einer Non-Profit-Organisation oder einer Firma an eine Gruppe von Individuen mit unterschiedlichen Fähigkeiten an einer Aktivität teilzunehmen und freiwillig eine Aufgabe zu erfüllen, deren Erledigung beiden Parteien einen gegenseitigen Nutzen bringt. Die Aufgabe kann unterschiedlich komplex sein und für die Erledigung Vorwissen der Individuen erfordern. Die Teilnehmer müssen für die Erledigung der Aufgabe Arbeit, Geld, Wissen oder Erfahrungen einbringen, wofür sie eine Bedürfnisbefriedigung erhalten, sei es ökonomischer, sozialer, wertschätzender Natur oder auch durch die Weiterentwicklung ihrer eigenen Fähigkeiten. Der Auftraggeber erhält im Gegenzug die Arbeitsleistung der Teilnehmer. Die Ergebnisse der Aufgabenerledigung können je nach Problemstellung sehr unterschiedlich aussehen.

Crowdsourcing ist also nichts anderes, als die verteilte Bearbeitung einer Aufgabe durch eine Gruppe variabler Größe und Fähigkeiten. Diese so genannte „Crowd“ ist je nach Aufgabe sehr unterschiedlich in ihrer Zusammenstellung von Beitragenden. Die Organisation oder das Individuum, was sich die Crowd zu Nutze macht, bezeichnet man auch als Crowdsourcer.

Zur Erfüllung der Aufgaben müssen die Crowd-Teilnehmer je nach Aufgabe Arbeit, Geld, Wissen oder Erfahrung einbringen, um diese erfolgreich zu lösen. Als Belohnung gibt es verschiedenste Möglichkeiten. Hierbei möchte ich noch einmal auf die Belegarbeit von Gerd Bombach [Bom12] verweisen, der in seiner Arbeit verschiedene Motivationsanreize näher beschreibt. Der Auftraggeber hingegen erhält im Gegenzug die Arbeitsleistung der Teilnehmer und die damit verbundenen Ergebnisse.

³ Ein Hersteller von Tortilla-Chips.

2.1.2 Arten von Crowdsourcing

Im Folgenden wird auf die Diplomarbeit von Tenshi Hara [Har12] verwiesen, der eine Taxonomie für das Thema Crowdsourcing entwickelt hat. In dieser Arbeit wird lediglich ein kurzer Einblick in die vom Autor später verwendeten Begrifflichkeiten gegeben, die aus Gerd Bombachs Übersetzung [Bom12] von Haras Arbeit sinngemäß übernommen wurden.

Implizites Crowdsourcing

Hara [Har12] definiert in seiner Taxonomie implizites Crowdsourcing als Typ des unbewussten, indirekten Crowdsourcing. Dabei handelt es sich um unbewusstes Crowdsourcing, was nur schwach mit den benötigten Daten in Zusammenhang steht. Neben der eigentlichen Tätigkeit wird eine Aufgabe als Nebeneffekt erfüllt, wobei eine Benutzer-Interaktion jedoch nur zur Erfassung der Submission notwendig ist.

Beispiel

Der Benutzer einer App, die Öffnungszeiten prüft, will herausfinden, ob das Geschäft x noch geöffnet hat. Dazu prüft die App nach einer Anfrage des Benutzers, ob andere Benutzer gerade im Geschäft x sind. Je nach Ergebnis der Prüfung wird eine Entscheidung getroffen, ob Geschäft x noch geöffnet ist. Es handelt sich hierbei um implizites Feedback, welches beim impliziten Crowdsourcing zur Datenerhebung genutzt wird.

Explizites Crowdsourcing

Explizites Crowdsourcing wird laut Hara [Har12] definiert als bewusstes, direktes Crowdsourcing. Dabei handelt es sich um bewusstes Crowdsourcing, was direkt mit den benötigten Daten in Zusammenhang steht. Der Benutzer muss aktiv Daten in ein System einbringen, wobei er das Bewusstsein hat, am Crowdsourcing-Prozess teilzunehmen und die Möglichkeit hat, zu bestimmen, ob und wann er teilnimmt.

Beispiel

Der Benutzer einer App, die Öffnungszeiten prüft, will herausfinden, ob das Geschäft x noch geöffnet hat. Zur Prüfung kann die App beispielsweise andere Benutzer fragen, ob sie die Öffnungszeiten des Geschäfts kennen oder die Benutzer senden die Öffnungszeiten regelmäßig an das System. Dadurch wird explizites Feedback genutzt, was beim expliziten Crowdsourcing bei der Datenerhebung Anwendung findet.

2.2 ANSÄTZE ZUM CROWDSOURCING-ANWENDUNGSFALL

Im Folgenden sollen Ansätze vorgestellt werden, die in der späteren Arbeit untersucht werden. Dabei handelt es sich um verschiedene Verfahren, die das Grundproblem lösen sollen, die im Moment an einem Ort befindlichen Personen nach bestimmten Kriterien zu erfassen und danach auf die Gesamtzahl dieser zu schließen. Als Anwendungsort sollte eine Mensa auf dem Campus der TU Dresden ausgewählt werden. Man einigte sich darauf, dass es sich hierbei um die Alte Mensa der TU Dresden handeln wird, die sich allein durch ihren Aufbau als sehr interessant herausstellt. Wie in Abbildung 2.1 zu sehen, ist die Alte Mensa in einer sehr speziellen Art und Weise entworfen worden. Dabei befindet sich in der Mitte der Mensa die Essensausgabe in einer Art Rechteck, welches ringsherum von den Besuchern begehbar ist. Im nächst größeren Rechteck befinden sich an jeder Ecke zwei Kassen. Die Mensa an sich besitzt vier Essenssäle (M06-M09) unterschiedlicher Größe. Dabei wird aus Gründen der einfacheren Auswertung davon ausgegangen, dass eine homogene Verteilung auf die Essenssäle stattfindet, da die Positionierung mit WLAN sowieso nicht so genau auf die Räume geschehen kann, dass man Inhomogenitäten abbilden könnte. Bestandteil der Mensa ist ebenfalls eine Cafeteria, sowie verschiedene Außenbereiche.

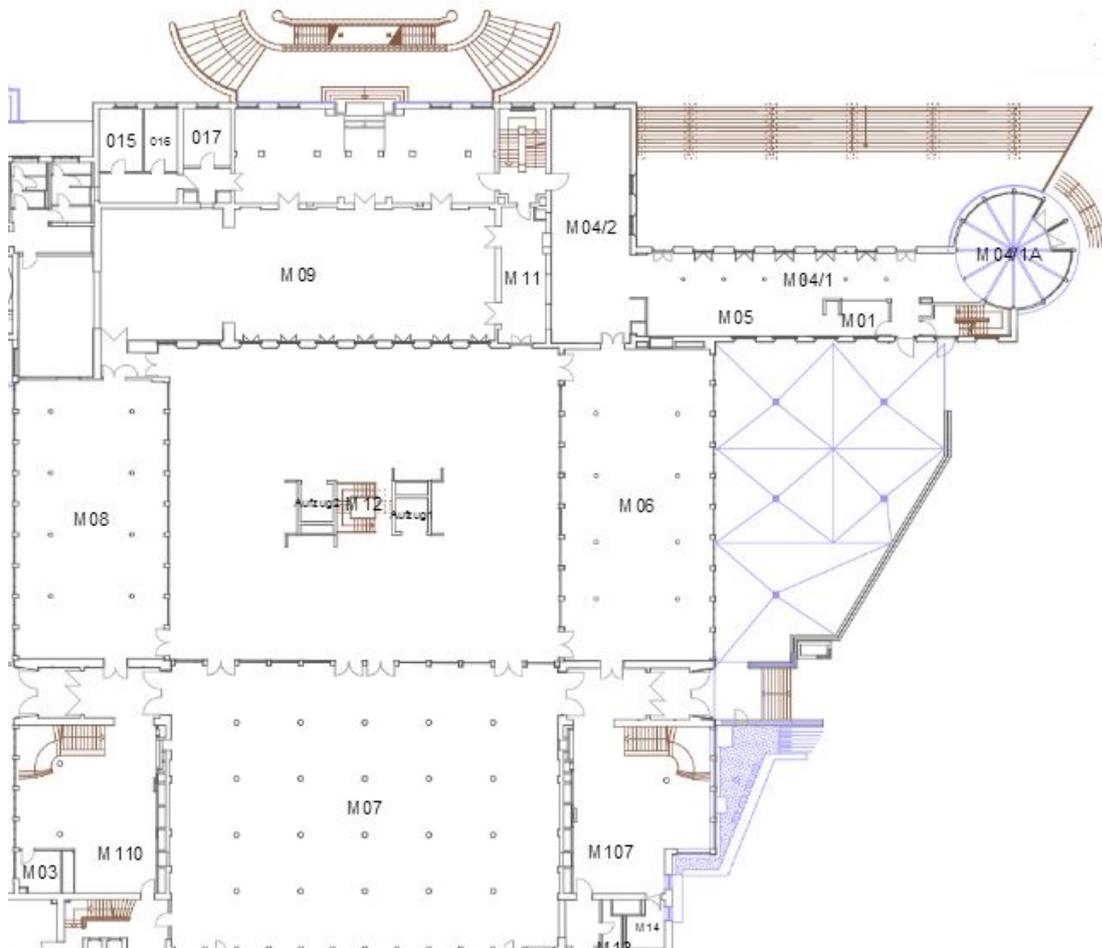


Abbildung 2.1: Der Grundriss zeigt den Aufbau der Mensa mit ihren verschiedenen Räumen. <http://navigator.tu-dresden.de/etplan/m13/00>

Kommentar

Ein besonderer Dank geht an dieser Stelle einerseits an Herrn Tenshi Hara, von dem die Idee eines MAC-Ansatzes stammt, sowie an Herrn Andreas Weigand, der die Idee des WLAN-Ansatzes bereitgestellt hat.

Bei den aufgeführten Ansätzen handelt es sich keineswegs um eine vollständige Liste vorhandener Möglichkeiten, die zur Lösung des Grundproblems herangezogen werden können, sondern lediglich um ausgewählte Ansätze, die im Anwendungsfall in Betracht gezogen werden sollen. Zur Lösung wurde auf Arten der Verifikation, die beispielsweise mit einem Positionierungsansatz möglich wäre, verzichtet, da reine Crowdsourcing-Ansätze untersucht werden sollten.

MAC-Ansatz

Der MAC-Ansatz befindet sich auf der zweiten Ebene (OSI-2) der Netzwerkkommunikation. Jedes Gerät, auf dem WLAN aktiviert ist, sendet sogenannte „Beacons“ (dt. Baken) aus, um Access Points zu entdecken. Die Zielstellung dieser Methode ist eine Art „AirCrack“. Die von den Geräten ausgesendeten Beacons sollen mitgehört werden und sofern ihre Signalstärke eine bestimmte Grenze überschreitet, ebenfalls auch gezählt werden. Mithilfe der Grenze der Signalstärke kann ein Radius untersucht werden, in dem nach Geräten Ausschau gehalten werden soll. Somit kann idealisiert ein Bereich einer Kugel geprüft werden, die sich um das Messgerät aufspannt und dessen Mittelpunkt das Gerät selbst ist. Unterscheidbar sind dabei die empfangenen Beacons durch ihre eindeutige MAC-Adresse, die für jedes Gerät einmalig auf der Welt vergeben wird. Es ist mit dieser Methode somit möglich, unterschiedliche Geräte, auf denen WLAN aktiviert ist, in einer bestimmten Entfernung vom eigenen Gerät mitzuhören.

IP-Ansatz

Der IP-Ansatz befindet sich auf der dritten Ebene (OSI-3) der Netzwerkkommunikation. Ziel des Ansatzes ist es, sich die Verteilung der IP-Adressen eines Netzwerkes zu Nutze zu machen. Je nach Netzwerkart besteht die Möglichkeit IP-Adressen auf verschiedenste Arten zu vergeben. Grundannahme des Ansatzes ist eine standortbasierte Vergabe. Je nach Gebäude bzw. naheliegenden Access Point wird eine IP-Adresse vergeben, die Aufschluss darüber gibt, an welchem Standort sich der Benutzer gerade ungefähr befindet. Durch ein Scannen des interessanten IP-Bereichs kann so darauf geschlossen werden, wie viele angemeldete Benutzer sich gerade in der Umgebung eines Bereiches befinden. Dabei ist das Ergebnis nicht wie im MAC-Ansatz von der Position abhängig, sondern lediglich von der Wahl des IP-Bereichs und kann genau dann erhoben werden, wenn sich das Gerät im Netzwerk befindet. Rückgabe sind verbundene Geräte mit diesem Netzwerk. Durch unterschiedliche IP-Adressen kann auf unterschiedliche Geräte geschlossen werden, da jedes nur genau einmal mit einem Netzwerk verbunden sein kann.

WLAN-Ansatz

Der WLAN-Ansatz beruht auf einer WLAN-Infrastruktur, die bei sehr großen drahtlosen Netzwerken oft umgesetzt ist. Ein weit verbreitetes Anwendungsgebiet solcher Infrastrukturen sind beispielsweise Campuse, auf denen eine komplette Vernetzung verfügbar sein muss, ebenso aber auch Sicherheitsaspekte im Vordergrund stehen zur Vermeidung von Missbrauch. Um Aufschluss über verschiedenste Daten zu bekommen, werden beispielsweise Infrastrukturen wie die Cisco Prime-Infrastruktur verwendet. Damit ist ein WLAN-Management möglich, wobei verschiedenste Daten abgerufen werden können. Je nach Qualität der Einbindung von Gebäuden kann unter anderem auch Aufschluss darüber gegeben werden, wie viele Geräte innerhalb eines Gebäudes verbunden sind. Dabei können sowohl verbundene, als auch authentifizierte Geräte abgefragt werden, wobei nach Gebrauch entschieden wird, welche der Informationen benötigt werden. Zur Erhebung der Daten muss der Benutzer lediglich Zugriff auf die Cisco Prime-Infrastruktur besitzen.

Kommentar

Ein besonderer Dank geht an dieser Stelle an Herrn Dr. Dietbert Gütter, der den Vorschlag einer Realisierung auf akustischer Ebene hatte.

Akustischer Ansatz

Der akustische Ansatz besteht aus der Annahme, dass je nach Füllstand eines Gebäudes auch die Lautstärke in diesem zunimmt. Je nach Art des Gebäudes kann dies sehr verschieden sein. Der funktionale Zusammenhang ist somit je nach Anwendungsfeld zu bestimmen. Zur Erhebung der Daten ist ähnlich wie im MAC-Ansatz keine weitere Infrastruktur im Gebäude nötig. Die Lautstärke kann mittels Schalldruck relativ bestimmt werden. Zur Messung ist lediglich das Gerät notwendig, das sich innerhalb des Gebäudes befinden muss. Die Messergebnisse sind dabei von der Position, sowie von der Art des Gerätes abhängig.

Expliziter Ansatz

Bei den vorangegangenen Ansätzen handelt es sich um implizite Crowdsourcing-Ansätze. Neben der Möglichkeit der automatischen Erfassung durch implizites Crowdsourcing ist es ebenfalls möglich, den Nutzer explizit in die Erfassung mit einzubeziehen.

Bei der expliziten Methode ist ebenfalls keine Infrastruktur innerhalb des Gebäudes notwendig. Der Ansatz wertet die Einschätzung der Benutzer direkt aus, die Eingaben ihrem Empfinden nach tätigen müssen. Da sie sich unmittelbar am Ort des Geschehens befinden, können sie den Füllstand sehr genau einschätzen. Zur Abgabe einer Submission muss sich der Benutzer im Gebäude befinden, oder sich gerade darin befunden haben, um Kenntnis darüber zu erlangen, wie hoch der Füllstand des Gebäudes ist.

Die Umsetzbarkeit der genannten Ansätze soll im Kapitel 4 genauer überprüft und bewertet werden. Eine Auswahl der zu verwendenden Methode für den Anwendungsfall des Füllstandes der Mensa findet sich ebenfalls dort. Im folgenden Kapitel soll näher auf Bewertungsverfahren eingegangen werden, um einen Einblick in verschiedene existente Verfahren zu geben.

3 BEWERTUNGSSYSTEME

Mit diesem Kapitel soll eine Einführung in das Thema der Bewertung von Nutzern und ihren Submissions gegeben werden. Zuerst soll geklärt werden, wozu Bewertungen durchgeführt werden und welche Art von Problemen ohne diese auftreten können. Dabei wird besonders auf Probleme durch beim Crowdsourcing erfasste Daten eingegangen. Danach wird diskutiert, wieso es wichtig ist, die Bewertung von Benutzern und Submissions geeignet miteinander zu verbinden. Anschließend werden drei verschiedene bekannte Seiten vorgestellt und die dortigen Bewertungsverfahren vorgestellt. Letztendlich werden die für die spätere Arbeit wichtigen Aspekte zusammengefasst.

3.1 SINN UND ZWECK VON BEWERTUNGEN

Crowdsourcing ist ein Verfahren, durch das sehr effizient Daten erhoben und aktualisiert werden können. Die Möglichkeit wird dadurch geboten, dass ein Individuum oder eine Gruppe von Leuten zur Datenerhebung/ bzw. -aktualisierung mit einbezogen wird. Da der Crowdsourcer, der Daten über die Crowd erhält, keine Sicherheit der Korrektheit der Daten besitzt, ist es nötig, diese Daten geeignet zu bewerten. Dabei sollen Daten genau dann aussortiert werden, wenn es sich um ungewollt falsch oder gar mutwillig falsch erhobene Daten handelt. Dazu wird des Öfteren der Nutzer, von dem die Submissions stammen, intern bewertet, wodurch schon vor einer Submission eine Tendenz der Richtigkeit dieser bekannt ist.

Eine Bewertung muss genau dann stattfinden, wenn die Qualität der Anwendung von der Korrektheit der erhobenen Daten abhängt. Keinem Benutzer einer Anwendung hilft es weiter, wenn er falsche Daten angezeigt bekommt. Da in heutiger Zeit immer mehr Leute anderen Leuten Schaden zufügen wollen und so bewusst falsche Submissions liefern, ist es notwendig, diese algorithmisch auszusortieren und gegebenenfalls für weitere Submissions zu sperren.

3.2 ART DES BEWERTUNGSVERFAHRENS

Bei einem Bewertungssystem ist es wichtig zu wissen, was genau bewertet werden soll und in welcher Art und Weise dies stattfindet. Aufgabenstellung der Bachelorarbeit ist die Entwicklung eines Algorithmus zur Bewertung von Crowdsourcing-Nutzern und deren Submissions. Dabei ist es von Vorteil, dass die Bewertungen beider Aspekte geeignet miteinander verbunden werden,

um die Benutzerbewertung in die Qualität der Submissions mit einzubeziehen. Je nach Rang des Benutzers sollen seine Submissions höher gewertet werden, wodurch sichergestellt wird, dass das optimale Ergebnis ausgewählt wird.

Die Bewertungsverfahren an sich laufen getrennt voneinander ab. Bei der Submissionbewertung wird lediglich die letzte Benutzerbewertung mit einbezogen. Je nach Ergebnis des Algorithmus wird eine Submission in einer Art positiv oder negativ bewertet, wobei es verschiedene Stufen geben wird. Wichtig dabei zu erwähnen ist die Trennung zwischen Benutzerbewertung und Bewertung der Submission, da das Ergebnis der Submissionbewertung nicht 1 zu 1 auf den Benutzer übernommen wird. Je nach Rang des Benutzers ist es notwendig, ihn geeignet zu bewerten. Dabei ist es von Vorteil mit einem Faktor zu arbeiten, womit die Benutzerbewertung im Maximalfall gegen 0 konvergiert. Damit soll sichergestellt werden, dass es eine Art Höchstbewertung eines Benutzers gibt. Weitere Aspekte des Bewertungsverfahrens werden genauer im Konzeptionsteil der Arbeit besprochen.

3.3 EXISTIERENDE BEWERTUNGSVERFAHREN

Nicht nur beim Crowdsourcing finden wir das Phänomen vor, dass Benutzer andere Benutzer gezielt täuschen oder betrügen wollen. Im folgenden Abschnitt werden drei bekannte Beispiele vorgestellt, bei denen Bewertungen von Beiträgen und Benutzern im Vordergrund stehen, durch welche die Seiten zum Vorreiter auf ihrem Gebiet geworden sind, was man an der Bekanntheit der Beispiele sehen wird.

3.3.1 YouTube

Bei YouTube handelt es sich um ein Online-Videoportal vom Anbieter Google, bei dem sich Benutzer rund um die Welt registrieren und Videos posten, sowie andere Videos bewerten oder kommentieren können. Gerade im Bereich des Veröffentlichens von Videos ist es wichtig, dass Videos, die Spam oder ähnlichen Inhalt zeigen, weniger Aufmerksamkeit bekommen, als Videos mit gutem Inhalt. Dies geschieht unter anderem durch ein Bewertungsverfahren unter jedem Video. Gefällt einem Benutzer ein Video, so kann er dies positiv mit einem Daumen nach oben bewerten, gefällt es ihm nicht oder stört ihn der Inhalt sogar, so kann er das Video negativ bewerten oder im Falle von nicht erlaubtem Inhalt sogar melden (siehe Abbildung 3.1).

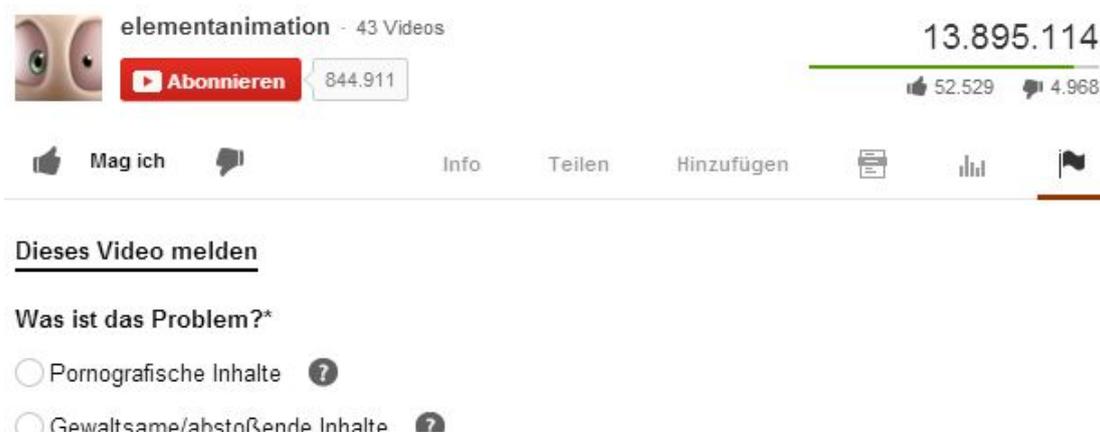


Abbildung 3.1: Beispiel einer sehr positiven Bewertung. Im ausgewählten Tab kann man als YouTube-Benutzer das Video nach Auswahl eines Kriteriums melden.

YouTube benutzt die Bewertung nicht nur zur Anzeige unter dem Video, sondern auch zum Ranking der Vielzahl an Videos hinsichtlich ihrer Qualität. Dabei will YouTube eine gewisse Qualität sicherstellen, wobei Videos, die andere Benutzer als negativ bewertet haben, weniger oft angezeigt werden, als gut bewertete Videos. YouTube nutzt dazu einen sehr ausgedehnten Rankingalgorithmus, dessen Einflussfaktoren sich von den Reaktionen (positiven/negativen Bewertungen, etc.), über Sharing, den Keywords, sowie der Anzahl an Views und weiteren Faktoren ziehen. Dabei möchte ich auf einen Artikel des tagseoblog.de verweisen, der sich genauer mit dem Ranking von YouTube auseinandersetzt und das Ganze grafisch aufbereitet [Mi12]. Die Tatsache, dass schlecht bewertete Videos trotzdem sehr viel Aufmerksamkeit bekommen, zeigt ein Musikvideo von Money Boy, wo dessen Bewertung und Klickzahl in Abbildung 3.2 zu sehen ist.

Money Boy - Dreh den Swag auf



Abbildung 3.2: Beispiel einer sehr negativen Bewertung.

Eine weitere sehr bekannte und beliebte Funktion von YouTube stellt die Kommentarfunktion dar, bei der ebenfalls eine Bewertung stattfindet. Jeder angemeldete Benutzer von YouTube hat die Möglichkeit unter Videos seiner Wahl Kommentare zu posten, die dann von anderen beantwortet werden können oder einfach als Kommentar zum Video dienen (siehe Abbildung 3.3).



Abbildung 3.3: Die Kommentarfunktion von YouTube.

Um die Qualität der Kommentare zu gewährleisten, findet sich auch hier die Daumen-Funktion wieder. Kommentare mit sehr positiven Feedback werden als sogenannte „Top-Comments“ weiter oben im Kommentarfeld angezeigt. Kommentare mit sehr negativem Feedback werden gesperrt. Ebenfalls können Kommentare, die unzulässigen Inhalt enthalten, auch von Nutzern als Spam markiert werden. In der heutigen Zeit hat diese Art von Kommentarbewertung eine Vielzahl an Vorteilen, beispielsweise das einfache Entfernen von unzulässigem Inhalt oder Spoilern, die den Ausgang des Videos vorhersagen (siehe Abbildung 4.5).



(a) Beispiel eines Top-Comment

(b) Beispiel von Spam (normalerweise ausgeblendet!)

Abbildung 3.4: Die Bewertungsfunktion von Kommentaren.

3.3.2 eBay

eBay ist das größte Online-Auktionshaus weltweit. Benutzer, verteilt um die halbe Welt, können hier Waren verschiedenster Kategorien verkaufen oder ersteigern. Da sich Jeder bei eBay anmelden kann, unabhängig von Beruf und ähnlichen Dingen, die Aufschluss über eine Person geben, ist es nötig, ein Bewertungsverfahren einzuführen. Dazu hat eBay eine umfassende Benutzerbewertung erstellt, die ebenfalls auf der Bewertung anderer Benutzer beruht. Kauft man bei eBay etwas, so wird man nach dem Kauf darauf hingewiesen, den Kauf zu bewerten, um so einen Einblick in die Transaktionsfähigkeit eines Benutzers zu bekommen. eBay unterscheidet dabei in positive Bewertung, neutrale Bewertung und negative Bewertung, wobei nur positive und negative Bewertungen zu einem Prozentsatz zusammengerechnet werden, der für die Anzahl positiver Bewertungen steht (siehe Abbildung 3.5).



Abbildung 3.5: Die Bewertung eines Benutzers wird bei jedem Artikel rechts oben angezeigt.

Wie in Abbildung 3.5 erkenntlich, gibt es verschiedene Arten von Verkäufern, die sich in der Anzahl ihre Transaktionen unterscheiden, gut zu vergleichen mit einem Klein- und Großanbieter derselben Ware. Der auf der Verkaufsseite angezeigte Bewertungswert wird jedoch gleich gebildet. Dieser Wert spiegelt das Transaktionsverhalten eines Benutzers in den letzten 12 Monaten wieder.

$$\frac{\text{PositiveBewertungen}}{\text{PositiveBewertungen} + \text{NegativeBewertungen}}$$

Die Formel für die Berechnung des Bewertungswertes. - <http://www.ebay.de>

Ebenfalls in Abbildung 3.5 erkenntlich sind die verschiedenen Bewertungssterne, sowie der Punktwert daneben. Dabei gibt es 12 verschiedene Sterne, die jeweils für eine Anzahl an Punkten stehen. Die Punkteanzahl bildet sich aus der Gesamtzahl aller positiven Bewertungen abzüglich der negativen Bewertungen. Neutrale Bewertungen werden ohne Zuwachs des Punktwertes gewertet. Neben der Bewertungsfunktion haben die Benutzer ebenfalls die Chance, ein Kommentar zur Bewertung hinzuzufügen, um späteren Käufern einen Einblick in die Art der Geschäfte des jeweiligen Verkäufers zu bieten. Diese Informationen können aufgerufen werden, indem im Angebot auf die Zahl der Bewertungspunkte geklickt wird. Es öffnet sich eine neue Übersichtsseite, wo jeder einsehen kann, wie die letzten Bewertungen eines Benutzers aussahen und einen Gesamtüberblick über diese erhält (siehe Abbildung 3.6). Um das Bewertungsverfahren abzurunden, können Benutzer ebenfalls als Käufer bewertet werden, wodurch auch Verkäufer einen Überblick über ihre Käufer bekommen können. Ebenfalls findet sich eine detaillierte Verkäuferbewertung auf dieser Seite, wodurch einzelne Informationen über die Leistungen eines Verkäufers sichtbar sind, die ebenfalls bewertet werden können. Die Gesamtheit dieser Informationen gibt Käufern und Verkäufern einen Einblick in das Transaktionsverhalten eines Benutzers und trägt damit zur Sicherheit des Geschäftskonzeptes bei. Benutzer, die sich vor dem Kauf oder Verkauf über ihren



steinmetz1976 (735 ☆)

Positive Bewertungen (der letzten 12 Monate): 99,2%
[Wie wird der Prozentsatz positiver Bewertungen berechnet?]

Mitglied seit: 10.09.07 in Deutschland

Mehr über dieses Mitglied

- [Mit Mitglied Kontakt aufnehmen](#)
- [Angebote aufrufen](#)
- [Bisherige Mitgliedsnamen aufrufen](#)
- [Zu meinen bevorzugten Verkäufern hinzufügen](#)
- [Meine eBay Welt aufrufen](#)
- [Testberichte & Ratgeber aufrufen](#)

Aktuelle Bewertungen (letzte 12 Monate) ?

	1 Monat	6 Monate	12 Monate
Positiv	9	73	120
Neutral	0	1	1
Negativ	0	0	1

Detaillierte Verkäuferbewertungen (letzte 12 Monate) ?

Kriterien	Durchschnittliche Bewertung	Anzahl der Bewertungen
Artikel wie beschrieben	★★★★★	40
Kommunikation	★★★★★	40
Versandzeit	★★★★★	40
Versand- und Verpackungskosten	★★★★★	42

Bewertung als Verkäufer
Bewertung als Käufer
Alle Bewertungen
Für andere Mitglieder abgegebene Bewertung

271 Bewertung erhalten (angezeigt werden 1-25) Bearbeitete Bewertungen: 0 ?

Bewertungen	Vom Käufer/Preis	Datum/Uhrzeit
immer wieder gern Sony PlayStation 3 80 GB (PAL - CECH-L04),Neu Laser (Nr.350828786772)	daddydragon2011 (17 ☆) EUR 130,00	04.08.13 12:48 Artikel aufrufen
Top eBayer gerne wieder Asus CROSSHAIR Buch,CD,Blenda,Lufter (Nr.350833826569)	maxdakoermel01 (1367 ☆) EUR 1,00	29.07.13 20:50 Artikel aufrufen

Abbildung 3.6: Die Detailansicht der Bewertungen eines Benutzers.

Verkäufer/ bzw. Käufer informieren, können so vor Betrug geschützt werden. Es liegt damit an jedem Benutzer selbst, wie sicher er seine Transaktionen durchführen möchte. Neben all den positiven Aspekten, gibt es immer noch sehr viel Kritik an der Benutzerbewertung von eBay. Besonders bei einer Umstellung des Bewertungssystems 2008 kam es zu einem weit verbreiteten Protest gegen eBay. Artikel, wie der von t-online.de [Sch08] finden sich zur Genüge im Internet. Grund der Beschwerdewelle war, dass Verkäufer ab sofort Käufer nur noch positiv bewerten konnten, wodurch Rachebewertungen unterbunden werden sollten. Einige Verkäufer berichteten fortan jedoch von Erpressungsversuchen von Käufern, die Verkäufern mit einer negativen Bewertung drohten, falls diese den Preis nicht verringern würden. Je nach Anzahl der positiven und negativen Bewertungen schlägt sich eine negative Bewertung stark auf den Prozentanteil der positiven Bewertungen nieder, wodurch Verkäufer massiv geschädigt werden können. Trotz allem Protest gilt die Änderung der Bewertungen der Käufer bis heute.

3.3.3 Stack Overflow

Stack Overflow ist eine Website, auf der Benutzer Fragen zu verschiedensten Aspekten zum Thema Computerprogrammierung stellen können. Dabei handelt es sich um eine so genannte „Frage- und Antwortplattform“, mit der Benutzer anderen Benutzern behilflich sein können. Dabei kann jeder angemeldete Benutzer Fragen stellen oder auf gestellte Fragen antworten. Die Fragen und die Antworten können jedoch auch von nicht angemeldeten Benutzern gesehen werden, wodurch Antworten auf Programmierfragen heutzutage meist auf dieser Seite gefunden werden können. Auch hierbei ist es notwendig, die Fragen, sowie die Antworten, zu bewerten, wodurch einerseits Interesse der Benutzer gezeigt werden kann, welche dieselbe Frage wie der Beitragsersteller haben (siehe Abbildung 3.7), andererseits aber auch die Qualität der Antworten gezeigt wird, wenn eine Antwort viele positive Bewertungen bekommt (siehe Abbildung 3.8). Die Bewertung erfolgt mittels der Pfeile nach oben oder nach unten, wobei nach oben eine positive Bewertung darstellt. Zwischen den Pfeilen wird die aktuelle Bewertung der Frage bzw. Antwort angezeigt, die bei negativer Bewertung durch ein Minus als Vorzeichen gekennzeichnet ist. Ähnlich wie bei YouTube wird die Antwort mit den meisten positiven Bewertungen weiter oben angezeigt, wodurch Benutzer, die auf diese Seite stoßen, oft eine schnelle Antwort auf ihre Frage finden.

Android - basic gesture detection

I've been battling to get 'fling' gesture detection working on my Android application today. I've been looking at these sources:

- Detect Gestures - Tutorial
- SDK docs
- Calculator Code

Nothing has worked for me so far and I was hoping for some pointers.

What I have is a 'GridLayout' that contains 9 ImageViews. The source can be found here: [Romain Guys' GridLayout](#).

That file is take from Romain Guy's [Photostream application](#) and has only been slightly adapted.

tagged

- android × 367658
- listener × 1824
- gesture-recognition × 299

asked **4 years ago**
viewed **221096 times**
active **2 months ago**

Linked

- 6 [How to detect doubletap on a View?](#)

Abbildung 3.7: Beispiel einer Bewertung einer gestellten Frage. Rechts im Bild Tags, die der Frage zugeordnet wurden, sowie Links zu ähnlichen Fragen.

Thanks to [Code Shogun](#), whose code I adapted to my situation.

435 Let your activity implement OnClickListener as usual:

```
public class SelectFilterActivity extends Activity implements OnClickListener
{
    private static final int SWIPE_MIN_DISTANCE = 120;
    private static final int SWIPE_MAX_OFF_PATH = 250;
    private static final int SWIPE_THRESHOLD_VELOCITY = 200;
    private GestureDetector gestureDetector;
    View.OnTouchListener gestureListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        /* ... */

        // Gesture detection
        gestureDetector = new GestureDetector(this, new MyGestureDetector());
    }
}
```

Abbildung 3.8: Beispiel der Top-Antwort der in Abbildung 3.7 gestellten Frage. Benutzer entscheiden, ob ihnen die Antwort geholfen hat oder nicht. Der grüne Haken signalisiert, dass der Fragensteller die Antwort akzeptiert hat.

Die Seite stellt sich selbst nicht als Diskussionsforum vor, sondern vielmehr als eine Seite auf der man seine Antworten bekommt. Um die Qualität der Antworten aufrecht zu erhalten und Motivation für die Beantworter einer Frage zu geben, wird nach Fragen gestrebt, auf die ein Benutzer vorher selbst nach einer Antwort gesucht hat und seine gefundene Antwort der Seite präsentiert. Dabei ist es wichtig, dass keine Diskussionen entstehen sollen, sondern vielmehr Antworten gegeben werden. Das Diskussionsfeature ist deshalb nur sehr gering als Kommentarfunktion zu einer Frage oder Antwort vorhanden, wodurch Fragen erweitert und Antworten ausgeweitet werden können.

Ähnlich wie bei allen anderen Frage- und Antwortseiten ist der erste Schritt, bevor man eine Frage stellt, auf der Seite nach bereits gelösten Problemen zu suchen. Dies wird durch die Tags, wie in Abbildung 3.7 erkenntlich, getan. Zudem helfen diese, den Beantwortern Fragen zu finden, über die sie das nötige Wissen besitzen.

Neben der Bewertung der Fragen und Antworten haben auch Benutzer eine Bewertung. Für Fragen und Antworten kann eine Bewertung in Form von Ansehen erlangt werden, wie in Abbildung 3.9 zu sehen ist.



Abbildung 3.9: Beispiel eines Benutzers und Möglichkeiten Ansehen zu bekommen. - <http://stackoverflow.com/about>

Je nach Höhe der Ansehenszahl bekommt der Benutzer verschiedenste Privilegien zugesprochen⁴, wie beispielsweise ab einem Score von 15 das nach oben Voten und ab einem Score von 125 das nach unten Voten einer Frage oder Antwort. Dadurch wird für Benutzer ein Anreiz geschaffen, gute Antworten abzugeben, um so weiter aufzusteigen und letztendlich sogar Moderator der Seite zu werden, um Spam und ähnliches direkt zu entfernen.

Wie in Abbildung 3.9 erkenntlich, finden sich bei der Benutzeranzeige neben der Ansehenszahl auch bis zu drei andere Zahlen, sogenannte Abzeichen. Dabei handelt es sich um Bronze-, Silber- und Goldabzeichen, die auf verschiedenste Weise erhalten werden können⁵ und eine Art Belohnung für sehr gute Benutzer darstellen.

Durch sein umfassendes Bewertungssystem liefert Stack Overflow eine Antwortqualität, die bei anderen Seiten ihres gleichen sucht. Grund dieser Qualität ist jedoch lediglich eine gut durchdachte Benutzerbewertung, die sowohl Fragenden als auch Antwortenden positiv auffällt.

Neben Stack Overflow gibt es noch eine Vielzahl an Seiten, die alle nach demselben Schema aufgebaut sind und nicht nur auf Computerprogrammierung spezialisiert sind. Das Angebot von StackExchange enthält neben der Lösung für Computerprogrammierung auch Seiten für Fragen und Antworten über Mathematik, Serverfehler, Ubuntu, Fotografie, Spieleentwicklung, sowie sehr viele andere Gebiete⁶.

⁴ <http://stackoverflow.com/help/privileges> (aufgerufen am 09.08.2013)

⁵ <http://stackoverflow.com/help/badges> (aufgerufen am 09.08.2013)

⁶ <http://stackexchange.com/sites> (aufgerufen am 09.08.2013)

3.4 ZUSAMMENFASSUNG

Die Betrachtung der Beispiele hat gezeigt, wie essentiell Bewertungsverfahren bei den verschiedenen Arten von Webseiten sind, bei denen Benutzer eine Rolle spielen. Die Bekanntheit dieser Internetseiten lässt darauf schließen, wie stark die Begeisterung der Benutzer mit der Qualität der Beiträge zusammenhängt. Einige Merkmale der genannten Beispiele sollen sich auch im zu entwickelnden Bewertungsalgorithmus wiederfinden. Bewertungen bei allen drei genannten Beispielen sind nur dann möglich, wenn der Benutzer der Seite angemeldet ist. Das ist ein wichtiger Bestandteil eines jeden Bewertungsalgorithmus, wodurch eine Vielzahl an falschen Bewertungen vermieden wird. Sobald man Benutzer in ein System einführt, sollte man bei der Bewertung dieses Verfahren bevorzugen. Ein weiterer Punkt, der positiv aufgefallen ist, ist bei eBay die 12-Monate Regel. Der Bewertungswert wird dabei nur für die letzten 12 Monate betrachtet, wodurch ein Benutzer einen Überblick über eine recht aktuelle Zahl bekommt. Das ist ebenfalls ein essentieller Punkt, da sonst schlecht bewertete Benutzer nie von ihren Bewertungen loskommen und positiv bewertete Benutzer immer positiv bewertet bleiben. Neben den positiven und negativen Bewertungen gibt es bei eBay neutrale Bewertungen, welche sich bei YouTube und Stack Overflow dadurch auszeichnen, dass man die Bewertung einfach weglässt und nicht erhebt. Im angestrebten Algorithmus muss es ebenfalls eine Möglichkeit geben, Submissions als neutral einzustufen, wenn diese nicht eindeutig eingeordnet werden können. Als letzten Aspekt müssen sehr positive Bewertungen auch eine Art Bonus bekommen, was sich bei YouTube dadurch auszeichnet, dass Videos höher gerankt werden, genau wie Kommentare oder die Top-Antwort bei Stack Overflow.

Zusammengefasst sind folgende Merkmale zu übertragen:

- Submissions können nur durch registrierte Benutzer erstellt werden,
- Es muss eine Möglichkeit geben, von negativen Bewertungen loszukommen,
- Neben positiven und negativen Bewertungen ist es sinnvoll neutrale bzw. andere Bewertungen einzuführen, falls weder das eine, noch das andere 100prozentig zutrifft und
- Von Benutzern mit guter Bewertung sollen Submissions leichter angenommen werden.

4 DISKUSSION MÖGLICHER LÖSUNGSANSÄTZE

Folgend sollen die in den Grundlagen besprochenen Ansätze näher betrachtet werden in Hinsicht darauf, wie die Daten für den gewählten Anwendungsfall erhoben werden können. Jeder der Ansätze wird dazu noch einmal kurz zusammengefasst und anschließend in einem Anwendungsfall, wie er in der Alten Mensa stattfinden würde, vorgestellt. Danach werden die Vorteile des besprochenen Ansatzes vorgestellt. Abschließend werden die Probleme besprochen, die allgemein im Ansatz vorhanden sind oder sich bei der Umsetzung bzw. bei der Anwendung im Anwendungsfall ergeben haben. Nachdem alle Ansätze besprochen wurden, kommt es zu einer Zusammenfassung und Auswahl einer dieser Ansätze, der im Anwendungsfall realisiert werden soll. Das ist ein wichtiger Schritt, bevor eine Benutzerbewertung stattfinden kann. Dabei ist es wichtig zu erwähnen, dass die ausgewählte Lösung keine allgemeingültige Lösung für andere ähnliche Anwendungsfälle darstellen wird, da die Art der Umsetzung sehr stark von der Infrastruktur der entsprechenden Gebäude abhängig ist. In unserem Anwendungsfall wurde, wie in den Grundlagen vorgestellt, die Alte Mensa der TU Dresden gewählt, deren Aufbau in Abbildung 2.1 ersichtlich ist.

4.1 MAC-ANSATZ

Beim MAC-Ansatz handelt es sich, wie entsprechend in den Grundlagen beschrieben, um eine Art „Mithören“ der sendenden Geräte. Dabei kann mit geringem Aufwand eine durchaus gute Variante realisiert werden, mit der Geräte in einem Umfeld gefunden werden können, auf denen WLAN aktiviert ist. Je nach Signalstärke dieser kann auf verschiedene Entfernungen geschlossen werden.

Anwendungsfall 1

Der Benutzer der App sitzt in der Alten Mensa der TU Dresden. Mittels der oben beschriebenen Methode prüft er an einer beliebigen Sitzposition, wie viele Leute WLAN aktiviert haben. Da es sich um eine studentische Mensa handelt, in der der Großteil der Besucher aus Studenten oder Mitarbeitern der TU Dresden besteht, kann man davon ausgehen, dass die Anzahl der gemessenen Geräte je nach Mensa-Füllstand sehr schwankt, jedoch aber selbst bei sehr geringem Füllstand der Mensa vorhanden ist. Die Anzahl der gemessenen Geräte ist somit ebenfalls als repräsentativ für den Mensa-Füllstand anzusehen. Die App gibt je nach Anzahl der gemessenen Geräte einen Füllstand der Mensa zurück.

Vorteile

Die Vorteile des Ansatzes sind eine recht einfache Implementierung mit der nötigen Schnittstelle, sowie die Möglichkeit der Bestimmung der Anzahl von Geräten in einem festgelegten Radius. Je nach Wahl der Minimalsignalstärke kann auf einen konstanten Radius geschlossen werden, wodurch die Messergebnisse für ein bestimmtes Umfeld bestimmt werden können. Zudem ist die Zahl der gemessenen Größe repräsentativ zur Bestimmung der Besucherfülle der Mensa, da von der Zahl der Geräte, auf denen WLAN aktiviert ist, recht genau auf die Zahl der Besucher geschlossen werden kann.

Probleme und Schlussfolgerung

Probleme im genannten Ansatz können dann auftreten, wenn der Benutzer an einer Randstelle im Raum sitzt und der Radius somit nur die Hälfte der in der Mitte gemessenen Daten abdeckt. Die App kann dies nicht erkennen und gibt als Füllstand nur die Hälfte des eigentlichen Wertes an. Je nach gewähltem Radius könnte dies eventuell ausgeglichen oder gar relativiert werden. Betrachten wir uns die Methode jedoch weiter im Detail, so fallen zwei weitere Probleme auf. Das erste dieser Probleme liegt in der Umsetzbarkeit der vorhandenen Methode auf dem zu verwendenden Android-Betriebssystem, da es sich später um eine Android-Anwendung handeln sollte. Es musste eine Methode gefunden werden, welche die erforderlichen Informationen bereitstellt. Weil es mit der Android API 1 lediglich möglich ist, von Access Points ausgesendete Beacons zu empfangen, wurde geschaut, ob es eventuell eine höhere API gibt, welche die geforderten Informationen bereitstellt. Weitere Möglichkeiten der Lösung wurden in diesem Fall nicht gefunden. Zu hohe APIs, wie beispielsweise API 14, wurden nicht betrachtet, da die App eine Vielzahl an Nutzern ansprechen können soll und das damit nicht mehr gegeben wäre. Weitere Betrachtungen der Methode und die Suche nach einer Lösung machten, wie zuvor schon erwähnt, auf ein weiteres Problem aufmerksam. Bei der beschriebenen Methode handelt es sich um eine Art „Air-Crack“. Dabei handelt es sich um einen Ansatz, bei dem von anderen Geräten gesendet Beacons mitgehört werden. Diese Beacons enthalten neben der Signalstärke ebenfalls Aussagen über das Gerät (siehe Abbildung 4.1). Da jede MAC-Adresse eindeutig einem Gerät zugewiesen ist und jedes Gerät auch einer Person zugeordnet werden kann, handelt es sich hierbei um die Erhebung von personenbezogenen Daten. Juristisch gesehen ist das in Deutschland strengstens verboten. Zudem würde es zum Ausschluss durch das ZIH⁷ der TU Dresden führen, da das Sniffen⁸ gegen die Nutzungsbedingungen verstößt und höchstens nach vorheriger Anmeldung für Forschungszwecke durchgeführt werden darf. Neben all diesen negativen Aspekten kommt noch hinzu, dass die Station-Liste von Abbildung 4.1 beim ZIH leer bleiben würde, da Sicherheitsvorkehrungen getroffen werden. Aufgrund der vielen aufgetretenen Probleme wurde der Ansatz verworfen. In anderen Ländern und mit anderen APIs handelt es sich jedoch um einen Ansatz, der Anwendung finden könnte.

```
C:\WINDOWS\system32\CMD.exe - airodump-ng --ivs --ch 9 -w dumpfile 127.0.0.1:12345
CH 9 II Elapsed: 1 min II 2007-11-27 21:09
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ES
00:1B:2F:0E:79:C4 0 10 153 0 0 11 54 WEP WEP n
00:01:36:0E:AF:54 0 93 801 5729 205 9 11 WEP WEP g

BSSID STATION PWR Rate Lost Packets Probes
00:01:36:0E:AF:54 00:19:7E:44:C7:C3 0 0-0 0 3
00:01:36:0E:AF:54 00:0D:88:45:23:89 -1 0-0 0 2555
(not associated) 00:16:6F:9E:24:46 0 0-0 0 2 faktar
(not associated) 00:16:6F:46:29:06 0 0-0 0 3 Rigas Lidosta
(not associated) 00:12:79:3E:06:45 0 0-0 0 2 HOME_ALMA
(not associated) 00:18:DE:BB:C1:01 0 0-0 0 3 SMC
(not associated) 00:15:AF:0E:59:CB 0 0-0 0 1 Levente
```

Abbildung 4.1: Beispielanzeige eines MAC-Ansatzes unter Windows. - <http://www.wifisoftwares.com/wp-content/gallery/software-screenshots/airodump-ng2.jpg>

⁷ Zentrum für Informationsdienste und Hochleistungsrechnen
⁸ Das Erfassen, Aufzeichnen und Auswerten von Datenverkehr in einem Netzwerk
<http://wirtschaftslexikon.gabler.de/Definition/sniffer.html>

4.2 IP-ANSATZ

Nachdem der erste Entwurf endgültig verworfen wurde, gingen die Überlegungen in der Netzwerkkommunikation eine Stufe nach oben. Schon bei der Suche einer ähnlichen App aus Ansatz 1 fiel auf, dass ein Netzwerk-Ansatz wohl nur über die IP-Ebene zu realisieren ist (solange man Android benutzt und auch ältere Geräte unterstützen will). Im Google Play Store fanden sich einige Apps, die diesen Ansatz verfolgten, beispielsweise „Fing - Network Tools“ (siehe Abbildung 4.2). Mit dieser App ist man in der Lage, ein Netzwerk, in dem man verbunden ist, nach anderen Geräten zu scannen. Als Rückgabe bekommt man die derzeit mit dem Netzwerk verbundenen IP-Adressen mit der MAC-Adresse des verbundenen Geräts. Da es zu Forschungszwecken möglich wäre, personenbezogene Daten, wie beispielsweise die MAC-Adresse ordnungsgemäß und gesichert zu verarbeiten, wurde dieser Ansatz weiter verfolgt.

Anwendungsfall 2

Der Benutzer der App sitzt in der Alten Mensa der TU Dresden. Mittels einer App, ähnlich wie „Fing - Network Tools“, scannt der Benutzer einen bestimmten Bereich der Mensa, in dem ein Access Point IP-Adressen an die nahe liegenden Geräte verteilt. Somit erhält der Nutzer alle Geräte in der Alten Mensa, die mit einem aussagekräftigen Access-Point innerhalb eines Essensraums verbunden sind. Aus der Anzahl der verbundenen Geräte kann, ebenfalls wie in Ansatz 1, auf den Füllstand der Mensa geschlossen werden. Die App gibt den Füllstand der Mensa zurück.

Vorteile

Vorteil dieser Methode ist die sehr hohe Genauigkeit. Wählt man einen repräsentativen Access-Point und die IP-Adressen, die dieser vergibt, so würde man, ähnlich wie im ersten Ansatz, eine sehr genaue Zahl bekommen, jedoch ohne den oben erwähnten Fehler der Sitzposition in der Mensa. Solange man mit dem Netzwerk verbunden ist, kann man einen gewissen IP-Bereich, der der Mensa entspricht, scannen, ohne direkt in der Mensa zu sein.

Probleme und Schlussfolgerung

Probleme bei diesem Ansatz traten jedoch sofern auf, dass die Infrastruktur an der TU Dresden diesen Ansatz nicht erlaubt. In einem kleinen Coffee Shop mit einem Access Point wäre die Variante durchaus realisierbar, in der TU Dresden und somit in der Alten Mensa nicht. Grund dafür ist, dass IP-Adressen unabhängig vom Ort vergeben werden. Die IP-Adresse, die gerade frei ist, wird vergeben, egal wo sich der eigene Standort auf dem Campus gerade befindet. Mit den IP-Adressen an sich ist somit keine Aussage darüber zu treffen, welchen Standort der Benutzer der App gerade hat. Um dies zu realisieren ist eine weitere Komponente nötig, die im folgenden Ansatz diskutiert werden soll. Der Ansatz der IP-Ebene wurde somit ebenfalls verworfen. In der TU Dresden könnte mit dem IP-Ansatz, mit enormer Messdauer, lediglich die Anzahl der im Campus verbundenen Geräte ermittelt werden, was jedoch nicht, beziehungsweise nur geringfügig, umrechenbar in den Füllstand der Mensa ist.



Abbildung 4.2: Fing - Network Tools zeigt eine Liste von verbundenen Geräten an. - <http://www.myandroiddevices.com/wp-content/uploads/2012/07/androidping.jpg>

4.3 WLAN-ANSATZ

Wie bereits im letzten Ansatz erkenntlich, besitzt die TU Dresden ein sehr weit entwickeltes System zum WLAN-Management. Eine Nachfrage bei der zuständigen Stelle hat dabei eine weitere Möglichkeit aufgebracht, wie der Anwendungsfall realisiert werden könnte. An der TU Dresden ist eine Cisco Prime-Infrastruktur umgesetzt. Dabei handelt es sich um eine sehr weit entwickelte Infrastruktur, die es ermöglicht, eine Vielzahl an Daten zu generieren, die Aufschluss über das Netzwerk geben. In gut eingepflegten Gebäuden, worunter auch die Alte Mensa zählt, ist es so z. B. möglich, die Anzahl der in der Alten Mensa verbundenen Geräte auszulesen. Die Daten sind dabei so genau, dass sie für den Anwendungsfall optimal erscheinen (siehe Abbildung 4.3).

Anwendungsfall 3

Der Benutzer der App sitzt in der Alten Mensa der TU Dresden. Mittels der App sendet er eine Anfrage an einen Server, der regelmäßig Daten vom WLAN-Management der TU Dresden erhält. Dieser sendet dem Benutzer als Antwort die zuletzt erhaltene Zahl zurück, welche die App verarbeitet und den Füllstand der Mensa anzeigt.

Vorteile

Vorteil dieser Methode ist die exakte Genauigkeit der gemessenen Daten. Mit Hilfe der Infrastruktur kann die genaue Anzahl verbundener, sowie verfügbarer Geräte in der Mensa bestimmt werden. Die Zahl könnte mittels eines Umrechnungsfaktors einfach auf den Füllstand der Mensa umgerechnet werden. Die Bestimmung der Daten erfolgt alle 5 Minuten und wäre so ebenfalls stets aktuell.

Probleme und Schlussfolgerung

Probleme ergeben sich bei diesem Ansatz lediglich bei der Umsetzung der vorgeschlagenen Variante. Da das Thema dieser Bachelorarbeit die Entwicklung eines Algorithmus zur Benutzerbewertung ist, würde die Umsetzung des Ansatzes, der durch die Cisco Infrastruktur sehr komplex ist, den Rahmen der Arbeit erheblich sprengen.

Für weitere Anwendungsfälle in naher Zukunft ist der Ansatz jedoch bestens geeignet, um Daten, die mit dem Netzwerk in Verbindung stehen, erheben zu können.

Client Count

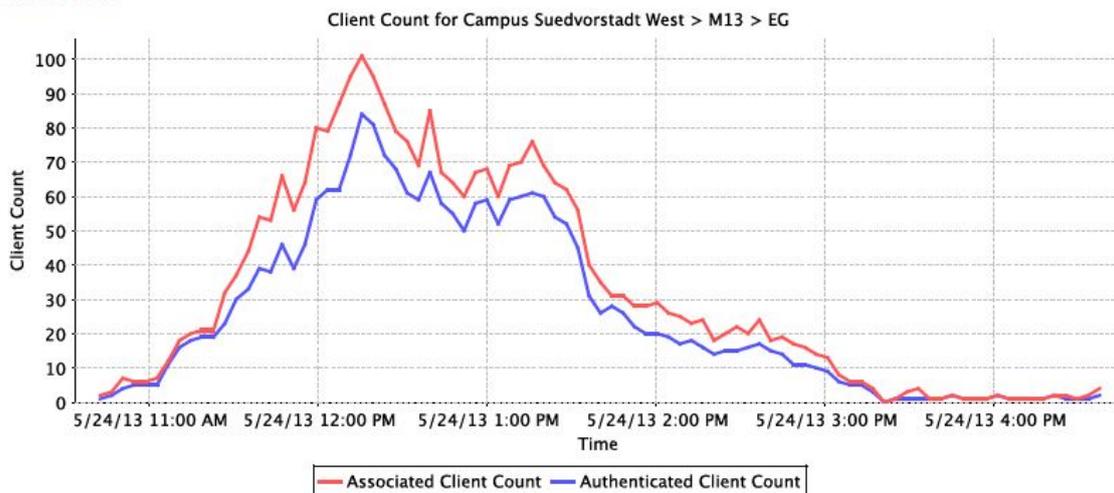


Abbildung 4.3: Der Graph wurde vom WLAN-Management der TU Dresden am 24. Mai 2013 erstellt. Dabei ist der Verlauf der Besucheranzahl der Alten Mensa gut erkenntlich. Neben der Darstellungsform als Graph können die Werte auch genauer zurückgegeben werden, beispielsweise als csv-Datei mit der Zeit der Messung, sowie der Anzahl an authentifizierten/bzw. verbundenen Benutzern. - ©TU Dresden

4.4 AKUSTISCHER ANSATZ

Beim letzten Ansatz, um ein indirektes Crowdsourcing zu realisieren, handelt es sich um einen Ansatz, der sich die zunehmende Lautstärke in der Mensa mit der wachsenden Besucheranzahl zu Nutze macht. Ziel des Ansatzes ist es, das Mikrofon des Smartphones anzusprechen und einen Lautstärke-Pegel zu bestimmen, der in den Füllstand der Mensa umgerechnet werden kann.

Anwendungsfall 4

Der Benutzer der App sitzt in der Alten Mensa der TU Dresden. Mittels der App wird der Nutzer angewiesen, sein Smartphone auf den Essenstisch zu legen. Anschließend wird die ungefähre Lautstärke berechnet durch Bestimmung der maximalen Amplitude, die in den Schalldruck umgerechnet wird. Aus mehreren gemessenen Werten wird das arithmetische Mittel genommen, um abweichende Werte abzuschwächen. Der gemittelte Wert wird in den Füllstand der Mensa umgewandelt, der durch die App zurückgegeben wird.

Vorteile

Als Vorteile des Ansatzes ergeben sich einerseits die einfache Implementierung, da lediglich beim Mikrofon mitgehört werden muss, andererseits, dass die Anwendung auf sehr alten Geräten lauffähig ist, da der benötigte MediaRecorder Teil der API Level 1 ist. Damit wird eine breite Anzahl an Benutzern angesprochen. Abschließend ist noch als Vorteil die Intuitivität der Methode zu nennen, da Benutzer die Methode einfach nachvollziehen können.

Probleme und Schlussfolgerung

Bei diesem Ansatz stellt es sich als Herausforderung dar, die gemessene maximale Amplitude in den entsprechenden Dezibelwert (Schalldruck) umzurechnen. Dabei können Probleme auftreten, weil der Schalldruck nur teilweise mit der Lautstärke zusammenhängt. Der Schalldruck ist kein direktes Maß für die Lautstärke. Es muss zudem untersucht werden, ob es eine Art Grundlautstärke gibt und mit welchen Faktoren diese zusammenhängt. Ebenso muss die Art des Wachstums festgestellt werden, der mit zunehmender Besucheranzahl stattfindet. Unterschiedliche Möglichkeiten sind dabei in Abbildung 4.4 zu sehen. Verschiedene Messungen haben zudem ergeben, dass es eine Höchstgrenze der menschlichen Stimme gibt. Als Problem bei diesem Ansatz stellt sich deshalb heraus, dass die maximale Lautstärke schon bei einer nicht vollen Mensa gegeben ist, wodurch der Ansatz für leerere Füllstände zwar sehr gut geeignet ist, mit zunehmend voller Mensa jedoch eine Sättigung erreicht und damit immer weniger geeignet ist.

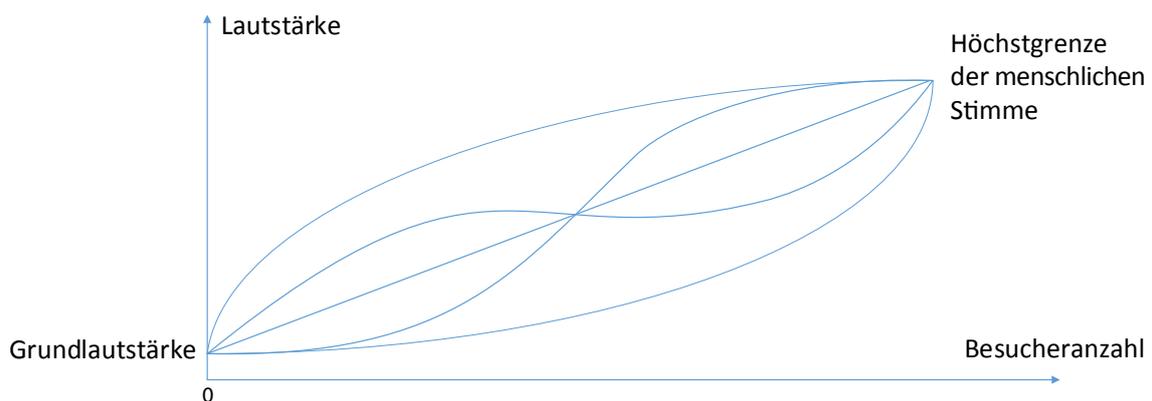


Abbildung 4.4: Der Graph zeigt mögliche Zusammenhänge zwischen der Lautstärke und der Besucheranzahl. Es ist anzunehmen, dass eine Grundlautstärke vorhanden ist, die den Anteil der Lautstärke ausmacht, der nicht durch Gespräche entsteht. Ebenso wurde herausgefunden, dass die Lautstärke bei gewisser Besucheranzahl konvergiert. Mögliche Ausschwankungen durch fallengelassene Tablets oder schreiende Kinder sind vorhanden, jedoch nicht grafisch dargestellt.

4.5 EXPLIZITER ANSATZ

Nachdem vier implizite Ansätze besprochen wurden, wird nun ein expliziter Ansatz vorgestellt. Dabei wird der Benutzer direkt in die Erfüllung einer Aufgabe einbezogen. Bei dem zu betrachtenden expliziten Ansatz handelt es sich um den Standardfall, dass der Benutzer direkt nach der geforderten Information gefragt wird.

Anwendungsfall 5

Der Benutzer der App sitzt in der Alten Mensa der TU Dresden. Mittels der App wird der Benutzer nach dem aktuellen Füllstand in der Mensa gefragt. Dabei kann er als Submission leer, halbvoll oder voll abgeben und damit die Fülle in der Mensa bewerten. Zudem ist die Einfachheit der Umsetzung der Methode zu erwähnen, da keine komplizierten Rechnungen oder Implementierung notwendig ist, sowie die breite Masse an Leuten, bei denen die Methode lauffähig ist.

Vorteile

Als Vorteil des expliziten Ansatzes ergibt sich, dass keine mit Fehlern verbundene Umrechnung stattfinden muss, da Benutzer den Füllstand direkt einschätzen können. Somit wird eine Vielzahl an Fehlerquellen bereits im Vorfeld allein durch den Ansatz ausgeschlossen. Die Methode ist zudem einfach umzusetzen und spricht eine Vielzahl an Geräten an. Wenn keine Vorgabe gegeben ist, läuft die Methode auf jedem internetfähigen Gerät. Neben der Einfachheit der Methode spricht die Genauigkeit ebenfalls für sie, die sehr hoch sein kann, wenn die Nutzer wahrheitsgemäße Angaben machen.

Probleme und Schlussfolgerung

Als Nachteil des Ansatzes ist zuallererst die objektive Einschätzung von Benutzern zu erwähnen. Da es keine festen Anhaltspunkte für den Füllstand der Mensa gibt, empfindet jeder Benutzer den Füllstand anders. Einige Leute werden es als sehr voll empfinden, wenn die Mensa nur halb gefüllt ist. Da es sich jedoch nur um drei zu bewertende Füllstände handelt, kann von diesem Nachteil abgesehen werden. Ein weiteres Problem des Ansatzes ist die hohe Wahrscheinlichkeit des Betrugs, da Benutzer sehr einfach dem System durch falsche Submissions Schaden zufügen können, was beim impliziten Verfahren je nach Ansatz weniger möglich ist. Als letzten Punkt möchte ich auf die Motivation von Benutzern eingehen. Da es sich beim expliziten Crowdsourcing um eine Methode handelt, bei der der Benutzer direkte Eingaben tätigen muss, ist es nötig, eine zusätzliche Motivationsquelle zu schaffen. Bei impliziten Methoden sinkt die Motivation weniger schnell, da lediglich ein Knopf betätigt werden muss. Trotz einiger Nachteile ist der explizite Ansatz geeignet, um mit ihm den zu entwerfenden Algorithmus zu testen.

"Mensa ist leer."

Loggen Sie sich ein, um eine eigene Eingabe zu treffen.

Benutzername:

Passwort:

(a) Benutzerlogin zur Abgabe von Submissions.

"Mensa ist leer."

Wie schätzen Sie den Füllstand der Mensa ein?

Die Mensa ist leer.

Die Mensa ist halbvoll.

Die Mensa ist voll.

(b) Layout zur Abgabe einer Submission.

Abbildung 4.5: Das Bild zeigt die mögliche Umsetzung eines expliziten Ansatzes. Nach einem Login bekommt der Benutzer die Möglichkeit selbst Submissions abzugeben. Im oberen Teil wird sowohl für eingeloggte, als auch nicht eingeloggte Benutzer der aktuelle Füllstand der Mensa, der mittels der expliziten Methode bestimmt wurde, angezeigt.

4.6 ZUSAMMENFASSUNG

Im vorangegangenen Abschnitt wurden verschiedene Möglichkeiten vorgestellt, mit denen Benutzer im Anwendungsfall implizites Crowdsourcing durchführen können. Auf explizite Methoden wurde nur kurz eingegangen, da diese sich meist sehr ähnlich sehen. Im Folgenden sollen die Vor- und Nachteile zusammengefasst und eine Variante ausgewählt werden.

	Vorteile	Nachteile
MAC-Ansatz	Einfache Implementierung	Positionsabhängig im Raum
	Feste Messweite möglich	Umsetzung mit API nicht möglich
	Umrechnung in Füllstand möglich	Sniffing-Ansatz verboten in TUD
IP-Ansatz	Gute Genauigkeit	Infrastruktur der TUD
	Messung von außerhalb möglich	
	Umrechnung in Füllstand möglich	
WLAN-Ansatz	Sehr hohe Genauigkeit	Rahmensprengung der BA
	Umrechnung in Füllstand möglich	
Akustischer Ansatz	Einfache Umsetzung	Schalldruck != Lautstärke
	Breite Zielgruppe	Grundlautstärke
	Intuitivität der Methode	Höchstgrenze menschl. Stimme
Expliziter Ansatz	Keine Umrechnung nötig	Betrug leicht möglich
	Kann sehr genau sein	Abhängig von Benutzern
	Einfache Umsetzung	Motivationsverlust
	Breite Zielgruppe	

Tabelle 4.1: Zusammenfassung der Vor- und Nachteile der besprochenen Ansätze.

Nachdem eine Zusammenfassung der Vor- und Nachteile der einzelnen Aspekte vollzogen wurde, wird nun die Frage geklärt, wie die Entscheidung der Auswahl einer dieser Verfahren getroffen wurde. Dazu werden alle Aspekte kurz auf Anwendbarkeit überprüft, sowie Kombinationen verschiedener Verfahren betrachtet.

Der MAC-Ansatz wies neben den aufgetretenen Schwierigkeiten im Datenschutz in der Implementierung erhebliche Schwierigkeiten auf, die mit der Android-API nicht gelöst werden konnten. Aufgrund der Tatsache, dass die Android-API zu verwenden war, wurde der Ansatz verworfen. Mit anderen Betriebssystemen wurde der Ansatz nicht betrachtet, da somit die Benutzermenge sehr eingeschränkt wäre und neben der Implementierung ebenfalls der Datenschutz ein nicht un schweres Problem darstellte.

Der IP-Ansatz wurde beispielhaft implementiert, ist jedoch in der Infrastruktur der TU Dresden nicht anwendbar. Somit kam es zu keinerlei brauchbaren Ergebnis, wodurch als Schlussfolgerung dessen entschieden wurde, ihn zu verwerfen. Zur Verifikation des Benutzers in expliziten Ansätzen ist der Ansatz ebenfalls nicht geeignet, da mittels IP-Adresse in der TU Dresden nicht auf die Position des Benutzers geschlossen werden kann.

Der WLAN-Ansatz konnte nicht direkt umgesetzt werden, da im Rahmen der Bachelorarbeit die nötige Zeit der Umsetzung einer Schnittstelle zur WLAN-Infrastruktur der TU Dresden fehlte. Die durch das WLAN-Management der TU Dresden vorhandenen Messdaten wurden jedoch in der Evaluation verwendet, da es sich dabei um sehr genaue Daten handelt, mit denen Vergleiche vorgenommen werden können. In einer späteren Arbeit könnte der WLAN-Ansatz komplett zur Verifikation von expliziten Submissions betrachtet werden und damit eine gute Kombination von

impliziten und expliziten Verfahren stattfinden.

Der akustische Ansatz wurde ebenfalls implementiert. In der Testphase fielen einige Probleme auf, wie sie vorher bereits genannt wurden. Mittels Debugging könnten einige der Fehler behoben werden. Damit der Ansatz verwendbar wäre, müsste eine Eichung jedes Gerätes vorgenommen werden, dass Submissions abgibt. Das könnte entweder automatisch oder durch den Benutzer direkt durchgeführt werden. Wenn diese Punkte berücksichtigt würden, könnte der Ansatz dazu verwendet werden, zu Zeiten mit geringen Füllständen eine implizite Methode bereitzustellen, die sehr gut funktional ist. In Kombination mit einem expliziten Ansatz, der Zeiten von vollen Mensafüllständen abdeckt, ist der akustische Ansatz sehr gut anwendbar. Aufgrund von Zeitmangel im Rahmen der Bachelorarbeit konnte der oben vorgeschlagene kombinierte Ansatz für den Anwendungsfall der Mensa nicht betrachtet werden.

Es wurde entschieden den Anwendungsfall komplett mit einem expliziten Ansatz umzusetzen. Grund dafür war einerseits die gute Funktionalität und große Menge an Benutzern, die angesprochen werden und teilnehmen können, sowie andererseits, dass der Ansatz im Rahmen der Bachelorarbeit bleiben musste, wodurch zwei mögliche Ansätze wie zuvor besprochen nicht realisiert werden konnten.

5 KONZEPTION DES ALGORITHMUS

Thema dieses Kapitels wird die Konzeption des Algorithmus zur Bewertung von Nutzern anhand der Qualität ihrer Submissions sein. Dabei soll zuerst eine Anforderungsanalyse durchgeführt werden, um benötigte Aspekte, die Bestandteil der Arbeit sein sollen, aufzulisten. Dazu werden die von Kapitel 3 gesammelten Merkmale um weitere ergänzt, auf die Bewertung von Crowdsourcing angepasst, sowie in einer Liste zusammengefasst. Anschließend werden wichtige Aspekte im Detail diskutiert. Schlussendlich wird der entworfene Algorithmus algorithmisch durchdacht und in Grafiken veranschaulicht. In der Zusammenfassung folgen wichtige Erkenntnisse, die in der Implementierung genauer betrachtet werden müssen.

5.1 ANFORDERUNGSANALYSE

Bei der Betrachtung der verschiedenen Bewertungsverfahren in Kapitel 3 fiel auf, dass Bewertungen nur durch registrierte Benutzer durchgeführt werden dürften. Im Algorithmus muss dieser Punkt berücksichtigt werden, auch wenn die Registrierung andersweitig durchgeführt wird. Ziel der Abgabe von Submissions nur durch registrierte Benutzer ist ein erstes Aussortieren von versehentlich bzw. vorsätzlich falsch erhobenen Inhalt. Somit wird eine Art Grundsicherheit geschaffen, um das System nicht unnötig zu belasten.

Bei der Eingabe in den Algorithmus soll es sich lediglich um eine UserID und eine Submission handeln. Informationen über Gerät, Benutzer oder Art der Submission sind nicht bekannt. Der Algorithmus muss Aussagen treffen können, ohne die Art der Submission mittels Eingabe zu bekommen. Die Gleichheit oder Ähnlichkeit einer Submission zu untersuchen wird nicht Teil des Algorithmus sein. Das Ergebnis muss dem Algorithmus zu Beginn durch den Crowdfunder geliefert werden. Ausgabe des Algorithmus ist eine Bewertung, bei der es mehr als eine positive und eine negative Bewertung geben sollte. Anschließend soll die neue Benutzerbewertung auf Grundlage der alten Benutzerbewertung und der bewerteten Submission berechnet werden.

Im Algorithmus muss es anfangs eine Standardbewertung geben, die für alle normalen Benutzer gleich ist. Zudem sollte es ein Minimum und ein Maximum der Benutzerbewertung geben, sodass es Grenzen für sehr positive und sehr negative Benutzer gibt, wodurch der Fakt realisiert wird, dass Bewertungen nicht ins unendlich Positive, bzw. Negative verlaufen.

Zudem soll der Algorithmus die Sperrung eines Benutzers durchführen, welche automatisch bei einer zu negativen Bewertung gesetzt werden soll.

Im Algorithmus soll eine Alterung der Bewertungen erfolgen, sodass positive Bewertungen nach gewisser inaktiver Zeit wieder nach unten fallen und negative Bewertungen wieder angehoben werden. Benutzer, die sehr lange inaktiv waren, werden so wieder als Normalbenutzer bewertet. Dabei soll auch eine Möglichkeit der automatischen Entsperrung realisiert werden.

Zur Entscheidung innerhalb des Algorithmus sollen Submissions von positiv bewerteten Benutzern höher bewertet werden als Submissions von negativ bewerteten Benutzern, wodurch die Submissions von positiv bewerteten Benutzern eher angenommen werden sollen.

Im Algorithmus soll eine Warteschlange vorgesehen werden, falls keine Aussage über eine Submission getroffen werden kann. Zudem soll der Algorithmus in der Lage sein, mit mehreren Submissions gleichzeitig zurechtzukommen, ohne zu verstopfen. Gelangt in den Datenbestand durch gezielte Angriffe mehrerer Nutzer, die dem System Schaden zufügen wollen, eine falsche Submission, so soll er in der Lage sein, dies nachträglich zu korrigieren. Bei der Umsetzung des Algorithmus soll auf minimale Serverauslastung geachtet werden, um möglichst ohne Verzögerung eine Antwort auf eine Submission zu bekommen.

Bei der Einrichtung des Algorithmus sollen verschiedene Eingaben durch den Crowdfunder getroffen werden können. Dabei soll er den Beginn der Alterung, sowie das Alterungsende, bestimmen können. Zudem soll er die Möglichkeit haben, eine Zeit festzulegen, in denen Submissions abgegeben werden dürfen. Da es oft keinen Sinn macht, zu bestimmten Zeitpunkten Submissions abzugeben, sollte in der nicht festgelegten Zeit (z. B. in der Nacht) eine negative Bewertung erfolgen. Der Crowdfunder soll zudem festlegen können, ab welcher Bewertung eine Sperrung in Kraft tritt und ab welcher diese wieder aufgehoben wird. Zudem soll er die Höhe der Bewertungen festlegen können, sowie eine Lebenszeit von Submissions, wie lange diese in Warteschlange und Datenbestand vergleichbar sein sollen. Zu guter Letzt sollte er Parameter beeinflussen können, die zur Annahme bzw. zum Ablehnen von Submissions genutzt werden.

Neben dem Festlegen von Werten sollte der Crowdfunder zudem eine Reihe anderer Funktionen besitzen. Er sollte in der Lage sein, Benutzer permanent oder normal zu sperren, sowie die Möglichkeit besitzen, alle gesperrten Benutzer abzurufen, wobei die Alterung regelmäßig durchgeführt werden sollte. Weiterhin soll der Crowdfunder die Möglichkeit haben, für Nutzer eine Gruppe zu setzen, was sinnvoll ist, wenn Benutzern mehr Vertrauen zugestanden werden soll (z. B. für Mitarbeiter). Durch das Setzen einer Gruppe ändert sich der Standardwert des Benutzers. Zudem soll der Crowdfunder in der Lage sein, die zuvor festgelegten Werte nachträglich zu ändern, falls er feststellt, dass diese nicht der Richtigkeit entsprechen.

Die in der Anforderungsanalyse besprochenen Aspekte sollen nun zusammengefasst werden:

1. **Der Algorithmus an sich**

- (a) Submissions können nur durch registrierte Benutzer durchgeführt werden.
- (b) Eingabe in den Algorithmus besteht nur aus UserID und Submission.
- (c) Der Algorithmus liefert eine Entscheidung und berechnet die Nutzer-Bewertung neu.
- (d) Es muss eine Standardbewertung geben, sowie eine Maximal- und Minimalbewertung.
- (e) Es muss eine Möglichkeit der automatischen Sperrung und Entsperrung geben.
- (f) Bewertungen sollen gegen einen Wert altern.
- (g) Submissions von positiv bewerteten Benutzern sind wichtiger als von negativ Bewerteten.
- (h) Der Algorithmus soll mit mehreren Submissions gleichzeitig zurechtkommen.
 - (i) Der Algorithmus soll sich bei Falschentscheidungen nachträglich korrigieren können.
 - (j) Durch minimale Laufzeit soll im Algorithmus keine Verzögerung stattfinden.

2. **Festlegung von Werten durch Crowdfunder**

- (a) Festlegung von Alterungsbeginn und Alterungsende.
- (b) Festlegung einer Zeit in der Submissions getätigt werden können.
- (c) Festlegung von Sperrungs- und Entsperrungswert.
- (d) Festlegung der Höhe von Bewertungen.
- (e) Festlegung der Lebenszeit von Submissions.
- (f) Festlegung von weiteren Parametern.

3. **Möglichkeiten des Crowdfunders**

- (a) Manuelle Sperrung von Benutzern (permanent/normal).
- (b) Abrufen von gesperrten Benutzern.
- (c) Realisierung von verschiedenen Benutzergruppen.
- (d) Nachträgliche Änderung der unter 2. eingetragenen Werte.

5.2 ENTWICKLUNG EINES KONZEPTS

Im Folgenden sollen Einzelheiten des Algorithmus im Detail diskutiert werden. Dabei wird auf einzelne Aspekte der Anforderungsanalyse eingegangen und diese auf Umsetzbarkeit überprüft. Letztendlich soll es zur algorithmischen Umsetzung der einzelnen entwickelten Teile des Konzeptes kommen, in der sich die einzelnen Aspekte der Anforderungsanalyse wiederfinden.

Grundsätzliches

Im Algorithmus ist es wichtig, dass es, ähnlich wie bei eBay, verschiedene Bewertungen gibt und diese miteinander in Verbindung gebracht werden. Bei eBay geschieht das über den prozentualen Anteil positiver Bewertungen. Eine solche Bewertung ist in unserem Fall jedoch nicht sehr hilfreich, da das Maximum (100-Prozent) sehr schnell erreicht werden kann. Ziel dieser Arbeit ist es vielmehr, einen Algorithmus zu entwickeln, durch den Bewertungen erst Stück für Stück anwachsen und dann sehr spät das Maximum erreichen. Daher ist es nötig, eine eigene Bewertungsstrategie zu entwickeln.

Die Bewertung basiert darauf, dass jeder Benutzer eine Zahl von 0 bis 1000 besitzt, die dessen Qualität bei der Abgabe von Submissions darstellt, wobei 0 die niedrigste und 1000 die höchste Bewertung darstellt. Dabei handelt es sich um 1001 verschiedene Werte, die eine Benutzerbewertung annehmen kann. Die Höhe dieser Werteanzahl wurde gewählt, um Fehler, die beim Runden entstehen können, zu verringern. Eine ungerade Werteanzahl wurde gewählt, um einen eindeutigen Mittelpunkt zu haben, bei dem die Bewertung weder negativ, noch positiv orientiert ist, der sogenannte Standardwert einer Bewertung. Hierbei kann man auch vom Startpunkt einer Benutzerbewertung reden, der genau dann vom System zugewiesen wird, wenn keine Aussagen über den Benutzer bekannt sind. Bei bestimmten Benutzergruppen hingegen könnte der Standardwert auf einen anderen, höheren Wert, festgelegt werden, da ihnen mehr Vertrauen zugewiesen werden kann. Im Normalfall liegt der Standardwert jedoch neutral bei 500.

Im Algorithmus gibt es drei verschiedene Bewertungen einer Submission, die aus dem Algorithmus herauskommen können. Die höchste dieser Bewertungen ist eine positive Bewertung für eine akzeptierte Submission, gefolgt von einer geringer positiven Bewertung, die der Benutzer für seinen Einsatz bekommen kann. Im negativen Bereich finden wir lediglich eine negative Bewertung vor. Über die Höhe der jeweiligen Werte kann der Crowdfunder selbst entscheiden. Dabei handelt es sich jedoch nur um Standardwerte, die aus dem Bewertungsalgorithmus kommen. Diese werden nur in Spezialfällen angenommen. Je nach vorhandener Benutzerbewertung, werden die Werte auf einen anderen Wert umgerechnet.

Beispiel

Ein Benutzer, dessen Bewertung auf 0 ist, gibt eine vom System als positiv bewertete Submission ab. Das Gleiche tut ein Benutzer mit einer Bewertung von 900. Der Benutzer mit der Bewertung von 0 muss höher bewertet werden, als der Benutzer mit der Bewertung von 900, da Bewertungen gegen 1000 konvergieren.

Bewertung einer Submission

Wie zuvor angesprochen kann eine Bewertung einer Submission folgende Werte annehmen: positiv, gering positiv und negativ. Im Folgenden wird die Umrechnung dieser Werte auf die eigentliche Bewertung einer Submission eines Benutzers diskutiert. Je nach Höhe der Benutzerbewertung werden dabei unterschiedliche Bewertungen errechnet, die anschließend auf die Benutzerbewertung addiert werden bei positiv bewerteten Submissions, bzw. subtrahiert werden bei negativ bewerteten Submissions. Im Algorithmus wird dabei ein linearer Ansatz verwendet.

Positive Bewertungen

Im Fall einer positiven oder gering positiven Bewertung nimmt diese mit zunehmender Benutzerbewertung linear ab. In Abbildung 5.1 ist das veranschaulicht.

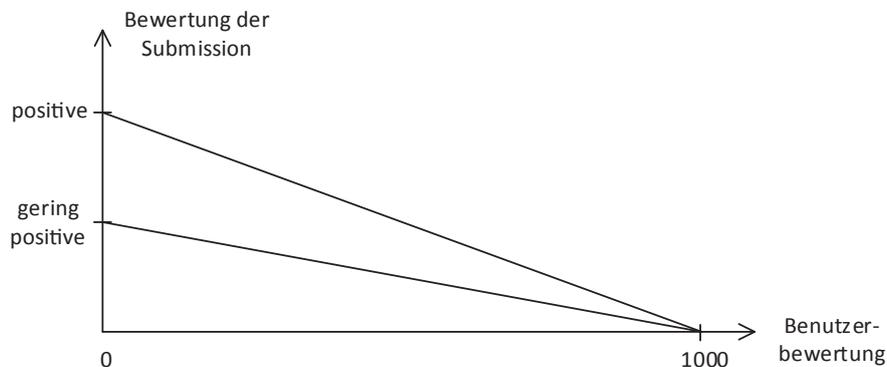


Abbildung 5.1: Der Graph zeigt den Zusammenhang zwischen einer positiven Submissionbewertung und einer Benutzerbewertung.

Je näher man an die Maximalbewertung kommt, desto niedriger ist der Zuwachs der Bewertung. Die resultierende Bewertung berechnet sich wie folgt:

$$\text{Neue_Benutzerbewertung} = \text{Alte_Bewertung} + \frac{\text{Differenz_zu_Max}}{\text{Maximalbewertung}} \cdot \text{Positive_Bewertung}$$

Ein ähnliches Verfahren findet sich bei Haftpflichtversicherungen von Autos wieder. Dazu wird jedem Versicherungsnehmer zu Beginn eine Schadensfreiheitsklasse zugeordnet, die ähnlich wie im Algorithmus eine Standardbewertung darstellt. In Folge von unfallfrei gefahrenen Jahren ist es möglich, eine höhere Schadensfreiheitsklasse zu erlangen. Je höher man aufsteigt, desto weniger Beiträge bezahlt man und desto geringer sinkt der prozentuale Abzug.

Negative Bewertung

Im Fall einer negativen Bewertung nimmt die Abnahme mit zunehmender Benutzerbewertung linear zu. In Abbildung 5.2 ist das veranschaulicht.

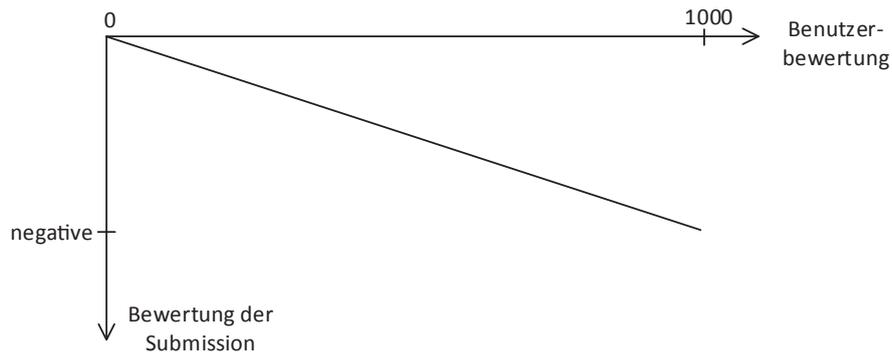


Abbildung 5.2: Der Graph zeigt den Zusammenhang zwischen einer negativen Submissionbewertung und einer Benutzerbewertung.

Ähnlich wie im positiven Bewertungsfall gibt es im negativen Fall ebenfalls eine Sättigungsgrenze, gegen den die Bewertung immer kleiner wird. Die Berechnung erfolgt laut folgender Formel:

$$\text{Neue_Benutzerbewertung} = \text{Alte_Bewertung} + \frac{\text{Differenz_zu_Min}}{\text{Maximalbewertung}} \cdot \text{Negative_Bewertung}$$

Der Alterungsprozess

Jeder Benutzer, dessen Bewertung ungleich der Standardbewertung ist, vollführt regelmäßig einen Alterungsprozess. Ziel dieses Alterungsprozesses ist es, dass Bewertungen von Benutzern nicht ewig bestehen, sondern nach der Zeit gegen den Standardwert konvergieren sollen. Somit wird die Aktivität im Algorithmus berücksichtigt und inaktive Benutzer auf den Standardwert zurückgeführt. Da Aktivität beim Crowdsourcing bei Benutzern als sehr wichtig einzuschätzen ist, ist dieser Schritt unverzichtbar beim Algorithmus.

Dabei sollen alle Benutzer in der Datenbank durchgegangen werden und dessen Bewertungen gealtert werden, wenn nötig. Hierbei ist die Eingabe des Crowdfunders notwendig, der einen Wert für den Beginn der Alterung festlegt, sowie einen Wert, wann die Alterung wieder den Standardwert erreicht haben soll. Für die Alterung wird ebenfalls ein linearer Zusammenhang im Algorithmus verwendet, wie Abbildung 5.3 veranschaulicht.

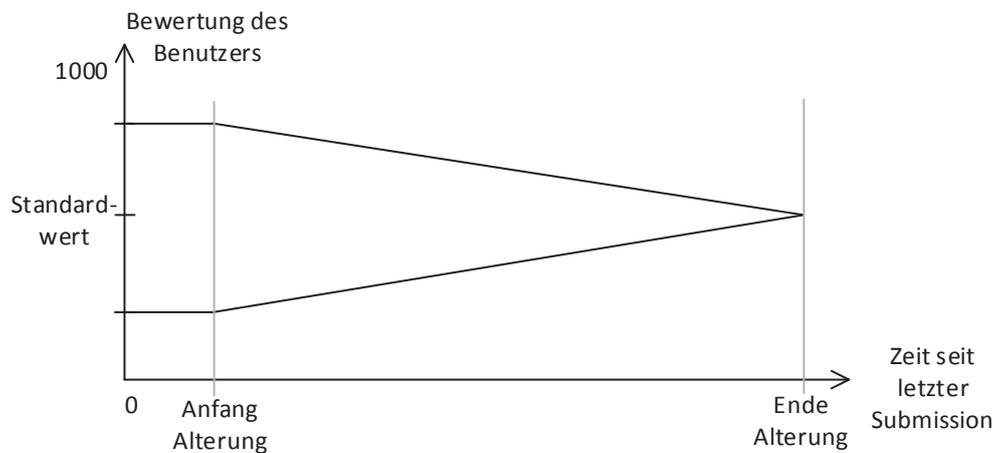


Abbildung 5.3: Der Graph zeigt den Zusammenhang zwischen einer Benutzerbewertung und der vergangenen Zeit seit der letzten Submission.

Die Sperrung

Grundsätzlich ist die Sperrung dazu da, Benutzer, die vorsätzlich und wiederholt falsche Submissions einreichen, vom System auszuschließen. Dazu gibt es zwei verschiedene Möglichkeiten eine Sperrung durchzuführen. Einerseits kann der Crowdfunder eine Sperrung eines Benutzers direkt durchführen. Solange diese Sperrung aktiv ist, kann der Benutzer keine Submissions mehr abgeben. Dabei kann es sich um eine permanente Sperrung handeln oder eine Sperrung, die durch den Altersprozess aufgehoben wird, wie es in der folgenden Methode auch der Fall ist. Im Gegensatz zur Sperrung durch den Crowdfunder direkt gibt es eine automatische Sperrung, wenn ein Benutzer unter einen bestimmten Bewertungswert fällt. Dieser Wert wird vom Crowdfunder festgelegt. Ist der Benutzer auf diese Weise vom System gesperrt worden, kann er keine Submissions mehr abgeben. Durch die stattfindende Alterung wird er nach einer gewissen Zeit jedoch wieder entsperrt, wodurch eine Möglichkeit einer natürlichen Entsperrung gegeben wird. Dabei wird eine Hysterese verwendet. Die Sperrung und Entsperrung erfolgen nach verschiedenen Werten, wodurch verhindert werden soll, dass Sperrung und Entsperrung sich ständig abwechseln. Veranschaulicht wird das Problem in Abbildung 5.4.

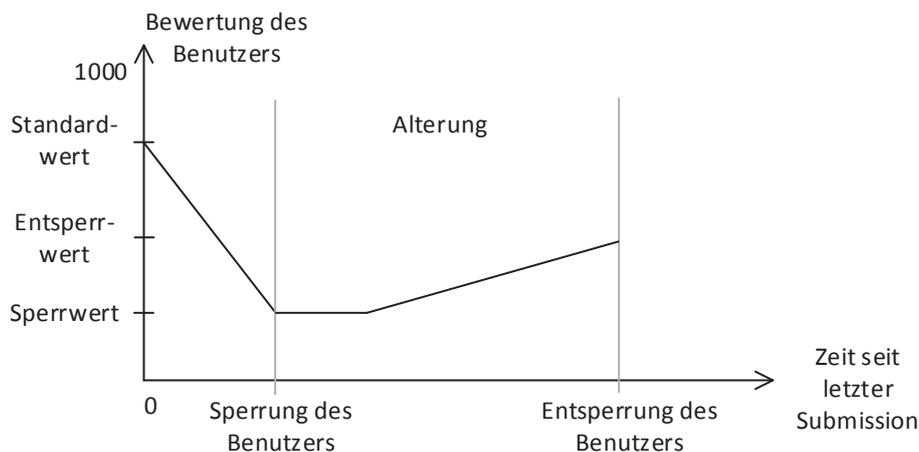


Abbildung 5.4: Der Graph zeigt den Verlauf einer automatische Sperrung und Entsperrung. Die hysteresebedingten Schranken wurden dabei eingetragen.

Erhebung einer Submissions

Submissions können nur durch angemeldete Benutzer erhoben werden, wodurch sichergestellt wird, dass eine Art Grundsicherheit besteht, wie in der Anforderungsanalyse besprochen. Dabei gestatten Benutzeraccounts ein erstes Aussortieren von versehentlich bzw. vorsätzlich falsch erhobenen Submissions.

Andererseits können Submissions nur in einem vom Crowdsourcer festgelegten Zeitraum abgegeben werden. Denken wir beispielsweise an eine Mensa, so machen Submissions mitten in der Nacht keinen Sinn, da die Mensa dann geschlossen hat. Benutzer sollen negativ bewertet werden, wenn sie in diesem Zeitraum Submissions abgeben.

Warteliste

Eine Warteliste ist genau dann notwendig, wenn Benutzer Submissions abgeben und das System im Datenbestand keine Daten der letzten, vom Crowdfunder bestimmten Minuten, zur Verfügung stehen hat. Im Algorithmus soll die Warteliste so realisiert werden, dass eine abgegebene Submission direkt in ihr gespeichert wird und erst danach geschaut wird, ob der Datensatz bereits Informationen enthält. Bei der weiteren Arbeit mit dem Algorithmus kann sich dies als Vorteil herausstellen.

Falls ein Datensatz vorhanden ist und eine Entscheidung getroffen werden kann, wird der Wert wieder aus der Warteliste entfernt. Falls kein Datensatz vorhanden ist, dient die Warteliste dazu, um Werte zu hinterlegen und auf weitere Submissions zu warten, um zu entscheiden, ob der Wert, der gerade in der Warteliste gespeichert wurde, eine richtige oder falsche Submission darstellt. Die Warteliste wird immer dann aktualisiert, wenn ein neuer Wert in das System kommt.

Historie

Die Historie stellt ebenfalls eine Art Warteliste bereit, indem sie eine Liste erstellt, in der alle Submissions gespeichert werden, ohne dass je ein Wert entfernt wird. Sie stellt also eine Gesamtheit aller Werte dar, die regelmäßig betrachtet werden muss, um falsche Submissions nachträglich zu entfernen.

Bewertungsalgorithmus

Nachdem zuvor alle für den Algorithmus wichtigen Aspekte genannt wurden, wird nun der Algorithmus an sich diskutiert. Dabei werden die zuvor vorgestellten Aspekte in einzelne Teile zusammengefasst und anschließend in Pseudo-Code formalisiert. Eingabe des Algorithmus ist eine Submission sowie die UserID des im System angemeldeten Benutzers. Einen Überblick über den Algorithmus findet man in Abbildung 5.5.

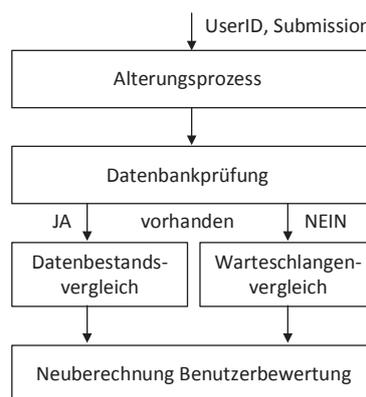


Abbildung 5.5: Bestandteile des Bewertungsalgorithmus.

Der erste Schritt ist es, die **Alterung** anzuwenden, um den aktuellen Wert des Benutzers zu bekommen. Dabei muss zuvor geschaut werden, ob der Benutzer permanent gesperrt ist, da sonst keine Alterung stattfinden soll.

Algorithmus 1: Alterungsprozess

```
Data: Alterungsbeginn; Alterungsende; Sperrvariable; LetzteSubZeit; AlteBewertung
Result: NeueBewertung
1 while Sperrung != permanent do
2   führe Alterung durch;
3   if LetzteSubZeit >= Alterungsbeginn then
4     if LetzteSubZeit <= Alterungsende then
5       Alterungsgesamtzeit = Alterungsende - Alterungsbeginn;
6       Alterung = LetzteSubZeit - Alterungsbeginn;
7       Alterungsfaktor = (Alterung/Alterungsgesamtzeit)*Diff(AlteBewertung,
8         Standardbewertung);
9       if AlteBewertung >= Standardwert then
10        | NeueBewertung = AlteBewertung - Alterungsfaktor;
11      else
12        | NeueBewertung = AlteBewertung + Alterungsfaktor;
13      end
14    else
15      komplette Alterung stattgefunden;
16      NeueBewertung = Standardbewertung;
17    end
18  else
19    keine Alterung stattgefunden;
20    NeueBewertung = AlteBewertung;
21  end
22  entsperre wenn nötig;
end
```

Nachdem der Alterungsprozess algorithmisch entworfen wurde, wird er noch in Abbildung 5.6 veranschaulicht.

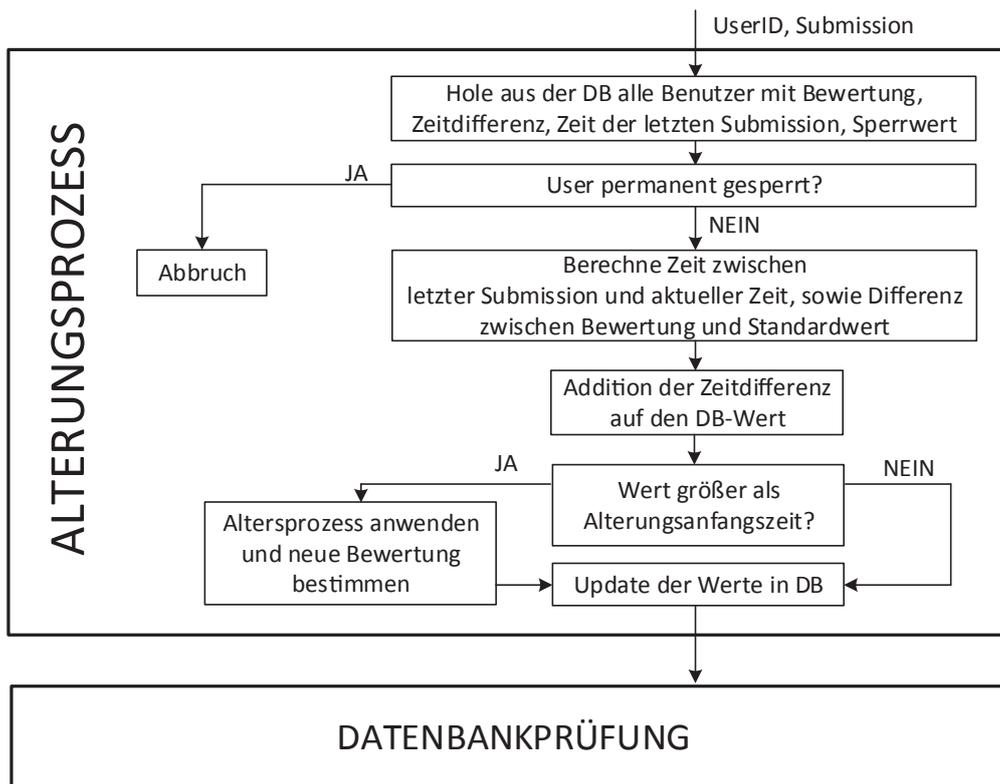


Abbildung 5.6: Der Alterungsprozess des Algorithmus.

Anschließend folgt die **Datenbankprüfung**. Es wird geprüft, ob der Benutzer normal gesperrt ist. Falls dies nicht der Fall ist, wird der Wert in die Warteliste übernommen und im System geschaut, ob bereits Werte der letzten, vom Crowdfunder bestimmten Minuten, vorhanden sind oder nicht.

<p>Algorithmus 2: Datenbankvergleich</p> <p>Data: NeueBewertung; Datenbestand; Sperrvariable; Warteschlange; Lebenszeit</p> <p>Result: Funktionsaufruf</p> <pre> 1 if Sperrvariable != Sperrung then 2 Warteschlange.add(EintragVonVorher); 3 if Diff(Datenbestand.lastEntry().time(), AktuelleZeit) < Lebenszeit then 4 Datenbestandsvergleich(); 5 else 6 Warteschlangenvergleich(); 7 end 8 else 9 Sperrung aktiv; 10 return; 11 end </pre>

Eine Übersicht über die Datenbankprüfung findet sich in Abbildung 5.7

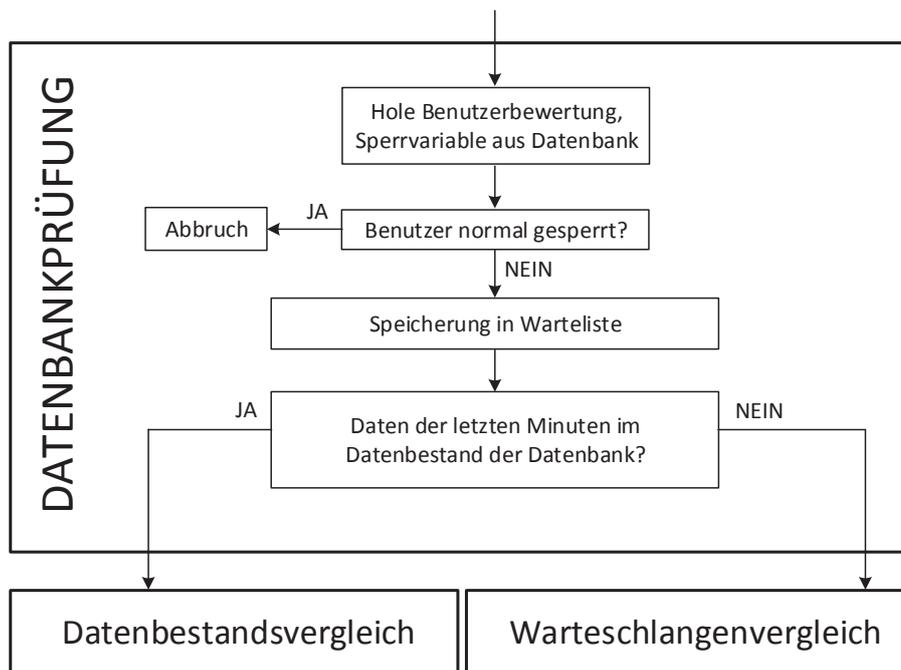


Abbildung 5.7: Die Datenbankprüfung des Algorithmus, die den weiteren Pfad auswählt.

Wenn bereits Werte vorhanden sind, wird die Submission mit dem Datensatz verglichen, im sogenannten **Datenbestandsvergleich**. Dabei wird geschaut, ob die Submission mit der letzten Submission im Datenbestand ähnlich ist. Falls ja, wird sie eingetragen und positiv gewertet, falls nicht, wird sie verworfen und negativ gewertet. Anschließend wird die Submission aus der Warteschlange gelöscht.

Algorithmus 3: Datenbestandsvergleich

Data: Datenbestand; Submission; Warteschlange

Result: Bewertungsfaktor

```

1 if Submission = Datenbestand.lastEntry().submission() then
2   | Datenbestand.add(Submission);
3   | Bewertungsfaktor = positiv;
4   | Warteschlange.delete(Submission);
5 else
6   | verwerfeSubmission;
7   | Bewertungsfaktor = negativ;
8   | Warteschlange.delete(Submission);
9 end
  
```

Der Datenbestandsvergleich ist grafisch in Abbildung 5.8 dargestellt.

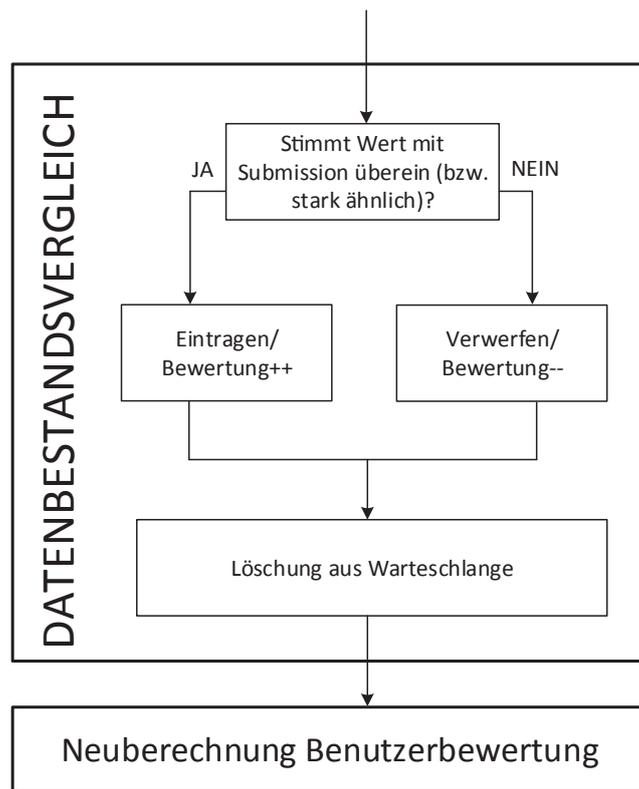


Abbildung 5.8: Der Datenbestandsvergleich des Algorithmus, bei dem geprüft wird, ob die Submission mit der letzten Submission des Datenbestands übereinstimmt/ähnlich ist.

Falls noch keine Werte im Datensatz vorhanden sind, wird die Warteliste durchgegangen (**Warteschlangenvergleich**), und alle Werte, deren vom Crowdfunder bestimmte Lebenszeit abgelaufen ist, darauf überprüft, ob dessen Erheber eine sehr hohe, vom Crowdfunder bestimmte Bewertung, besitzt. Falls dies der Fall ist, wird der Wert in den Datensatz übernommen, obwohl er keine Vergleichswerte besitzt, andernfalls verworfen. In beiden Fällen erfolgt eine gering positive Bewertung des Benutzers für seinen Einsatz. Die Werte werden aus der Warteliste gelöscht. Falls lediglich ein Wert in der Warteliste übrig bleibt, so wird auf weitere gewartet. Befinden sich mehrere Werte in der Warteliste, werden diese mit derselben Submission zusammengefasst und es wird geschaut, ob dieser Wert einen, vom Crowdfunder bestimmten Prozentwert, überschreitet. Wenn ja, werden die Submissions mit dem Wert in den Datensatz eingetragen und positiv bewertet und alle anderen negativ bewertet und verworfen. Anschließend werden alle aus der Warteschlange gelöscht. Falls kein Wert den Prozentwert überschreitet, wird auf weitere Werte gewartet.

Algorithmus 4: Warteschlangenvergleich

```
Data: Warteschlange; Submission; Lebenszeit; Datenbestand
Result: Bewertungsfaktor
1 foreach Eintrag von Warteliste do
2   if Diff(Eintrag.time(),AktuelleZeit)>Lebenszeit then
3     if Eintrag.bewertung()>Vertrauenswert then
4       Datenbestand.add(Eintrag.submission());
5     end
6     Warteschlange.delete(Eintrag.submission());
7     Bewertungsfaktor = gering positiv;
8   else
9     Hilfsliste.add(Submission);
10  end
11 end
12 if Hilfsliste.size()>1 then
13   foreach Eintrag von Hilfsliste do
14     if Eintrag.submission() = 0 then
15       Wert1+=Eintrag.bewertung();
16     end
17     if Eintrag.submission() = 1 then
18       Wert2+=Eintrag.bewertung();
19     end
20     if  $(\text{Wert1}/(\text{Wert1}+\text{Wert2}))>\text{Prozentwert}$  then
21       Entscheidung = 0;
22     end
23     if  $(\text{Wert2}/(\text{Wert1}+\text{Wert2}))>\text{Prozentwert}$  then
24       Entscheidung = 1;
25     end
26   end
27   if Entscheidung != NULL then
28     foreach Eintrag von Hilfsliste do
29       if Eintrag.submission == Entscheidung then
30         Bewertungsfaktor = positiv;
31         Datenbestand.add(Eintrag.submission());
32       else
33         Bewertungsfaktor = negativ;
34       end
35       Warteschlange.delete(Eintrag.submission());
36     end
37   end
38 else
39   wait();
40 end
```

Eine Übersicht des Warteschlangenvergleichs findet sich in Abbildung 5.9.

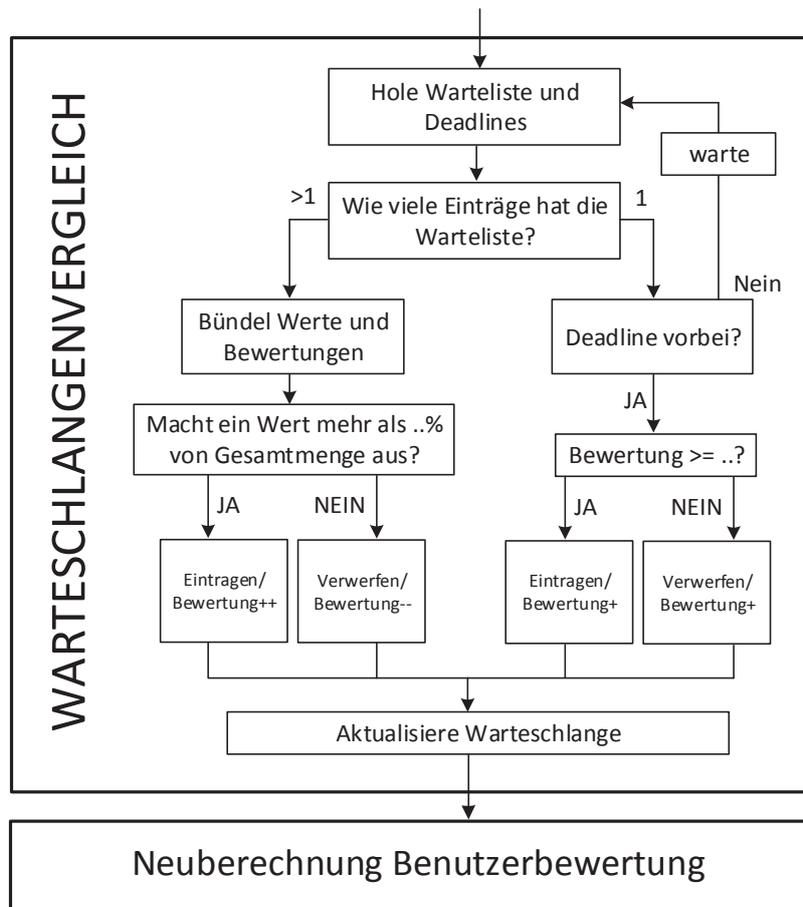


Abbildung 5.9: Der Warteschlangenvergleich des Algorithmus, der dann genutzt wird, wenn keine aktuellen Daten im Datenbestand vorhanden sind.

Je nach Bewertung wird im letzten Schritt die **Benutzerneuberechnung** durchgeführt.

Algorithmus 5: Benutzerneuberechnung

Data: MaxBewertung; MinBewertung; AlteBewertung; Submission; Bewertungsfaktor

Result: Neue Bewertung

- 1 **if** *Bewertungsfaktor* == *positiv* // *Bewertungsfaktor* == *gering positiv* **then**
- 2 | Zuwachs = (Diff(AlteBewertung, MaxBewertung)/Diff(MaxBewertung, MinBewertung))*Bewertungsfaktor;
- 3 | Neue Bewertung = Alte Bewertung + Zuwachs;
- 4 **else**
- 5 | Zuwachs = (Diff(AlteBewertung, MinBewertung)/Diff(MaxBewertung, MinBewertung))*Bewertungsfaktor;
- 6 | Neue Bewertung = Alte Bewertung - Zuwachs;
- 7 **end**

Eine Übersicht der Neuberechnung gibt es in Abbildung 5.10.

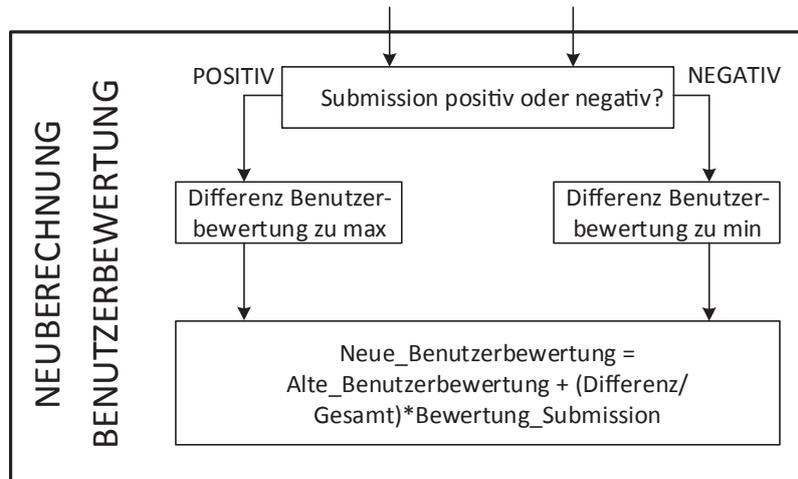


Abbildung 5.10: Die Berechnung der neuen Benutzerbewertung nach Entscheidung des Algorithmus über die Submission.

5.3 ZUSAMMENFASSUNG

Im aktuellen Kapitel wurde eine Anforderungsanalyse durchgeführt, in der die aus Kapitel 3 gesammelten Aspekte auf den zu entwickelnden Algorithmus zur Bewertung von Crowdsourcing-Nutzern und ihren Submissions angepasst, sowie um weitere ergänzt. Dabei wurde eine Liste entwickelt, die alle Aspekte unter drei Hauptpunkten: „Algorithmus an sich“, „Festlegung von Werten durch Crowdfunder“ und „Möglichkeiten des Crowdfunders“ zusammenfasst. Diese einzelnen Punkte der Liste sollen in Kapitel 7 auf ihre Umsetzung in der Implementierung geprüft werden. Nach der Durchführung der Anforderungsanalyse wurden verschiedene der Aspekte hinsichtlich ihrer konzeptuellen Umsetzung genauer diskutiert. Ziel dieser Diskussion war die Zusammenfassung der besprochenen Aspekte in fünf große Teilgebiete, die zusammen den Algorithmus bilden. Dabei wurde eine Grafik entwickelt, welche die einzelnen Teile des Algorithmus zeigt. Der erste Schritt des Algorithmus ist der Alterungsprozess, indem die Bewertung des Benutzers, der die Submission abgegeben hat, anhand der letzten in der Datenbank vorhandenen Submission gealtert wird. Danach folgt die Datenbankprüfung, in welcher der Datenbestand auf kürzlich eingetragene Submissions überprüft wird. Sind Submissions vorhanden, so erfolgt ein Datenbestandsvergleich, andernfalls ein Warteschlangenvergleich. Im Datenbestandsvergleich wird die Submission direkt mit dem letzten im Datenbestand vorhandenen Eintrag verglichen. Im Warteschlangenvergleich hingegen wird der Wert in einer Warteschlange gespeichert, wo er auf weitere Werte zum Vergleichen wartet, sodass eine Entscheidung getroffen werden kann. Folgen keine weiteren Werte, so wird der Wert verworfen, falls die Benutzerbewertung nicht außerordentlich hoch ist. Gleichgültig welche der beiden Varianten durchgeführt wurde, folgt die Neuberechnung der Benutzerbewertung, die den letzten Punkt des Algorithmus darstellt. Jedes dieser Teilgebiete wurde in diesem Kapitel kurz besprochen und in Pseudo-Code verdeutlicht sowie grafisch aufbereitet.

Im nächsten Kapitel soll die Implementierung des Konzepts erfolgen, wobei neben der Umsetzung auf aufgetretene Probleme, sowie verwendete Technologien und Programme eingegangen wird. Im Umsetzungsteil werden dabei Änderungen des Konzepts vorgestellt, sowie die Realisierung der einzelnen Teile des Algorithmus diskutiert.

6 IMPLEMENTIERUNG

Nachdem im vorangegangenen Kapitel ein Konzept für den Algorithmus entwickelt wurde, wird sich dieses Kapitel mit der Implementierung der Überlegungen befassen. Der zuvor entworfene Pseudo-Code wird in einer geeigneten Programmiersprache ausgeweitet und umgesetzt. Dabei wird auf Kernaspekte der Implementierung, sowie auf aufgetretene Änderungen zwischen Konzeption und Implementierung eingegangen. Neben der Umsetzung werden aufgetretene Probleme und Schwierigkeiten genannt. Im letzten Teil des Kapitels werden verwendete Technologien und Hilfsmittel vorgestellt, die während der Implementierung verwendet wurden

6.1 PROBLEME UND SCHWIERIGKEITEN

Auf der Suche nach einem geeigneten Algorithmus, der gegebenenfalls übernommen werden könnte, kam schnell die Erkenntnis, dass es keine komplett passende Lösung geben wird. Es musste somit zur Verschmelzung verschiedener Aspekte anderer Algorithmen kommen, die zusammen den Algorithmus bilden sollten, nach dem gesucht wurde. Nachdem der Algorithmus im Konzept ausgearbeitet war und es an die Implementierung ging, mussten einige kleinere Veränderungen vorgenommen werden. Grund dafür war die Automatisierung der Alterung, sowie der Grundgesamtheit, die mittels Cron Job umgesetzt wurde.

Cron Jobs

Laut [Sch10] bezeichnen Cron Jobs die Abnahme von immer wiederkehrenden Aufgaben, wie beispielsweise Sicherungen des Datenbestandes oder das Archivieren von Logfiles. Es handelt sich dabei um Routineaufgaben, die regelmäßig stattfinden sollen. Im Algorithmus wurde dies zweimal angewendet. Einerseits ist es notwendig, dass der Alterungsprozess regelmäßig stattfindet, sodass der Crowdfunder jederzeit einen Überblick über die aktuellen Bewertungen samt Alterung seiner Benutzer hat und Sperrungen beispielsweise abrufen kann. Andererseits wurde die Grundgesamtheit über einen Cron Job realisiert, der regelmäßig die Ergebnisse des Algorithmus mit der Grundgesamtheit überprüft und gegebenenfalls Korrekturen vornimmt. Damit können eventuell falsch eingetragene Submissions entfernt werden, sodass künftige Submissions davon nicht negativ profitieren. Algorithmisch vorgesehen anstatt eines Cron Jobs war hingegen die direkte Überprüfung, was aufgrund der enormen benötigten Rechenzeit nicht realisierbar war. Dabei stellen Cron Jobs eine serverlastfreundliche Alternative dar, wodurch die Grundfunktionalität erhalten

bleibt. In Abbildung 6.1 erhält man einen Überblick über die stattgefundenen Veränderungen mittels Cron Jobs.

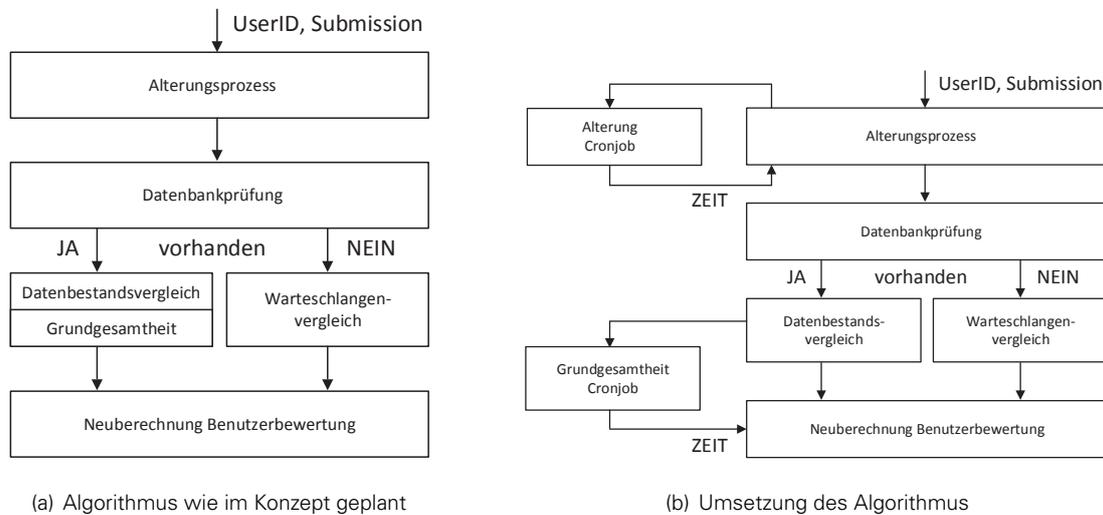


Abbildung 6.1: Die Veränderungen an der Struktur des Algorithmus.

Prüfung der Ähnlichkeit

Da im Algorithmus keine Vorgabe gemacht wurde, wie Submissions aussehen, kam es bei der Prüfung der Ähnlichkeit der Submissions zu weiteren Problemen. Aufgrund der Tatsache, dass die Bachelorarbeit auf der Entwicklung des eigentlichen Algorithmus basiert, handelt es sich dabei um eine Baustelle, die weniger Bestandteil dieser Arbeit sein wird. Da es in der Implementierung jedoch notwendig war zu wissen, wann es sich um ähnliche Beiträge handelt, wurde der Fall der Gleichheit implementiert, der im Anwendungsfall, mit dem der Algorithmus getestet wird, funktional ist. Eine Submission ist ähnlich, wenn sie gleich ist. Weitere Betrachtungen der Ähnlichkeit müssen in einer future work behandelt werden, um andere Anwendungsfälle, wie sie beispielsweise im MapBiquitous-Projekt der TU Dresden auftreten, zu realisieren.

Anwendungsfall

Die wohl meiste Arbeit steckte in der Umsetzung des Anwendungsfalles. Zunächst wurden verschiedene implizite Methoden (wie in Kapitel 4 vorgestellt) implementiert, die jedoch erhebliche Probleme aufwiesen. Die Implementierung des MAC-Ansatzes wurde auf verschiedenste Weisen versucht, die jedoch nie eine Lösung ergaben. Danach ging es an die Implementierung des IP-Ansatzes, wobei nach Fertigstellung der Implementierung auffiel, dass der Ansatz in der TU Dresden nicht realisierbar ist. Grund dafür ist die WLAN-Infrastruktur des Campus, die vorsieht, dass IP-Adressen nach Bedarf und nicht nach Standort vergeben werden. Anschließend wurde der akustische Ansatz umgesetzt, um eventuell doch einen impliziten Ansatz zu verfolgen. Dabei wurde neben der bereits vorher gedachten Höchstgrenze der menschlichen Stimme (siehe Abbildung 4.4) ebenfalls eine enorme Abweichung der Messergebnisse festgestellt, wie man in Tabelle 6.1 sieht, wodurch auch dieser Ansatz letztendlich verworfen wurde. Daher wurde der explizite Ansatz umgesetzt und wird zur Evaluation herangezogen.

	HTC Desire	HTC Desire X	ALCATEL ONE TOUCH 991D
API Level	17	15	10
Tag	2013-07-18	2013-07-18	2013-07-18
Zeit	14:09:09.028	14:09:27.543	14:09:23.802
Wert alle 10ms	80dB	73dB	64dB
Wert alle 30ms	79dB	72dB	64dB
Wert alle 50ms	80dB	71dB	63dB
Wert alle 1s	80dB	71dB	60dB

Tabelle 6.1: Aufzeigen der Abweichungen bei einer Messung mittels Sound-Ansatz. Dabei befanden sich alle drei verwendeten Geräte im selben Raum, sowie am selben Tisch nebeneinander.

6.2 UMSETZUNG

Grundsätzlich wurde der Algorithmus, wie in Kapitel 5 beschrieben, in fünf großen Teilen implementiert. Jeder der Bestandteile bildet ein extra Teil des Algorithmus, der sich von den anderen Teilen abgrenzt. In der Implementierung fließen diese Teile ineinander über, was aber nichts an der Funktionalität des Konzepts ändert. Im Folgenden werden die fünf einzelnen Teile, die in der Implementierung mit Hilfe des vorher entwickelten Pseudocodes umgesetzt wurden, näher beschrieben. Dabei wird auf mögliche Änderungen, sowie besondere Stellen innerhalb der Implementierung, eingegangen.

Alterungsprozess

Der Alterungsprozess findet vor jeder Submission statt. Außerdem wird er alle fünf Minuten über einen Cron Job ausgeführt, wodurch der Crowdfunder jederzeit die Benutzerbewertungen samt aktueller Alterung abrufen kann, ohne abhängig von Submissions von einzelnen Benutzern zu sein. Im Alterungsprozess wird zuerst die Sperrvariable überprüft. Wurde diese auf permanent gesperrt gesetzt (im Algorithmus ist sie dann auf 2 gesetzt), so findet keine Alterung statt. Bei einer normalen Sperrung (wenn sie auf 1 gesetzt ist) wird normal die Alterung durchgeführt und gegebenenfalls der Benutzer entsperrt. Zur Sperrung wird eine Hysterese verwendet. Diese dient dazu, ein ständiges Abwechseln von Sperrung und Entsperrung zu verhindern, indem ab einer bestimmten Bewertung gesperrt wird und es danach eine Art Sperrphase gibt und der Benutzer erst wieder entsperrt wird, wenn er eine andere höhere Bewertung hat (die Sperrvariable wird dann auf 0 gesetzt). In Abbildung 5.4 ist der Zusammenhang noch einmal grafisch aufbereitet. Nach Überprüfung der Sperrvariable wird der Alterungsprozess, wie in der Konzeption beschrieben, durchgeführt. Implementiert wurde die Alterung als einzelnes Skript, da es regelmäßig vom Cron Job aufgerufen werden muss. Wenn der Benutzer vorher normal gesperrt war und durch die Alterung eine festgelegte Grenze wieder überschreitet, wird er automatisch entsperrt, was ebenfalls auch parallel mittels Cron Job stattfinden kann. Zum Aufrufen des Cron Jobs wurde der Dienst von cronjob.de verwendet, wodurch alle fünf Minuten das Alterungsskript aufgerufen werden kann. Ein häufigeres Aufrufen wäre mit der Premium-Version des Dienstes, oder mit einem anderen Dienst, ebenfalls möglich. Für den ausgedachten Anwendungsfall waren fünf Minuten jedoch ausreichend oft.

Datenbankprüfung

In der Phase der Datenbankprüfung erfolgte zuerst die Überprüfung der Sperrvariable des Benutzers. Nach Durchführung der Alterung kann sich diese verändert haben. Eine aktuelle Sperrvariable kann aus der Datenbank geholt werden. Falls der Benutzer gesperrt ist kommt es zu einem Abbruch des Algorithmus. Im Falle, dass keine Sperrung vorhanden ist, erfolgt einerseits der Eintrag der neuen Bewertung des Benutzers und seiner Submission in die Warteliste, andererseits wird geschaut, ob im Datenbestand bereits Werte der letzten Minuten eingetragen sind. Falls ja so findet ein Datenbestandsvergleich statt, andernfalls ein Warteschlangenvergleich. In der Implementierung ist das Ganze so realisiert, dass die $\text{MAX}(\text{time})$ mittels SQL-Abfrage aus dem Datenbestand geholt wird. Falls die Differenz zwischen dieser und der aktuellen Zeit nicht größer als die angesprochenen letzten Minuten ist, so ist ein Datensatz dieser Zeit vorhanden.

Datenbestandsvergleich

Im Datenbestandsvergleich wird die letzte Submission aus dem Datenbestand geholt und mit der zu prüfenden Submission auf Ähnlichkeit verglichen. Wie bereits vorher angesprochen, wurde in der Implementierung eine Ähnlichkeit in Form einer Gleichheit angenommen, wodurch die Ähnlichkeit lediglich mit $==$ realisiert wird. Stimmt die Submission mit der letzten Submission des Datenbestandes überein, so findet eine positive Bewertung statt, die in der Neuberechnung der Benutzerbewertung zur Berechnung herangezogen wird. Stimmt die Submission nicht mit der letzten Submission des Datenbestandes überein, so findet eine negative Bewertung statt. Ein weiterer Bestandteil des Datenbestandsvergleichs ist die Betrachtung der Grundgesamtheit, die in der Implementierung jedoch nur angedeutet werden konnte. Dabei soll mit Hilfe eines Cron Jobs regelmäßig überprüft werden, ob die eingetragenen Submissions wirklich stimmen oder ob Betrüger falsche Daten in das System gebracht haben. Nachträglich können so Fehler korrigiert werden. Die Realisierung der Betrachtung der Grundgesamtheit muss in einer fortsetzenden Arbeit erfolgen, da noch keine funktionale Lösung gefunden wurde.

Warteschlangenvergleich

Ein Warteschlangenvergleich findet genau dann statt, wenn es keine Daten der letzten Minuten im Datenbestand gibt. Die Warteschlange ist eine Art Hinterlegung von Werten, die noch nicht auf Richtigkeit geprüft werden konnten. Alle Werte bleiben solange in der Warteschlange bis ihre Lebenszeit abgelaufen ist. Ist bis dahin keine Entscheidung gefallen, werden sie nur dann eingetragen, wenn die Benutzerbewertung sehr hoch ist. Grundsätzlich ist das Ziel der Warteschlange das Warten auf andere Werte und die Bündelung von gleichen Werten, um danach zu entscheiden, ob ein Großteil der Daten dasselbe angibt. Besonders dabei ist zu erwähnen, dass jede Submission nicht als ein Punkt gezählt wird, sondern die Bewertung des Benutzers, der die Submission abgegeben hat, als Wert genommen wird. Dabei betrachten wir das in Tabelle 6.2 dargestellte Beispiel.

In Beispiel 1 haben wir 2 Submissions die „ja“ sagen, jeweils mit Benutzerbewertung 50 und eine Submission mit „nein“ und Benutzerbewertung 50. Nun wird „ja“ überprüft, indem die Bewertungen beider Submission addiert ($50+50=100$) und durch die Summe aller Bewertungen geteilt werden ($50+50+50=150$), wodurch $100/150 = 0,666..$ als Ergebnis folgt. Die Submission wird angenommen, wodurch sowohl Sub1 als auch Sub3 positiv und Sub2 negativ bewertet werden.

In Beispiel 2 wird wieder die Submission „ja“ überprüft, die 100 beträgt, da nur Sub1 mit „ja“ geantwortet hat. Betrachten wir die Summe aller Bewertungen ($100+25+25=150$), so folgt das selbe Ergebnis wie in Beispiel 1, wodurch gezeigt ist, wie stark die Bewertung in die Annahme einer Submission einfließt.

Sub1 (Bewertung)	Sub2 (Bewertung)	Sub3 (Bewertung)	Entscheidung
ja(50)	nein(50)	ja(50)	ja wird akzeptiert
ja(100)	nein(25)	nein(25)	ja wird akzeptiert

Tabelle 6.2: Zwei verschiedene Beispiele mit der getroffenen Entscheidung des Algorithmus. Die Höhe des Annahmewerts liegt bei 0,6, die eine Submission von der Gesamtheit haben muss.

Neuberechnung Benutzerbewertung

Nachdem mittels Datenbestandsvergleich oder Warteschlangenvergleich eine Entscheidung gefällt wurde, wie die Submission zu bewerten ist, wird die Benutzerbewertung des Benutzers, der die Submission abgegeben hat, neu berechnet. Dabei ist die Höhe der Bewertung davon abhängig, wie hoch die Bewertung des Benutzers vorher bereits war. Die Implementierung erfolgte anhand des Pseudo-Codes, der in Algorithmus 5 dargestellt ist.

6.3 VERWENDETE TECHNOLOGIEN UND PROGRAMME

Zur Implementierung wurde die Programmiersprache PHP⁹ verwendet. Die Auswahl hatte verschiedenste Gründe, die sich von der Einfachheit, bis hin zur Verfügbarkeit auf verschiedensten Free-Hostern ziehen. Zur Umsetzung wurde dazu der kostenlose Service von bplaced.net genutzt. Dabei steht für den Benutzer in der kostenlosen Standardversion 1GB Webspace bereit, ebenso verschiedenste FTP-Accounts, MySQL-Datenbanken oder auch PostgreSQL-Datenbanken. Neben PHP war es wichtig, eine MySQL-Datenbank zu benutzen, um die Vielzahl an Daten, die zur Bearbeitung gebraucht werden, zu speichern und gegebenenfalls dauerhaft in der Datenbank zu belassen. Zum Zugriff auf diese Datenbank wurde MySQLi gewählt, was eine Erweiterung von PHP ist, um den Zugriff zu gewährleisten. Im Vergleich zu MySQL ist MySQLi schneller, einheitlicher, sicherer und bietet mehr Funktionen als MySQL. Neben dem Service von bplaced.net wurde XAMPP¹⁰ genutzt, um die Implementierung auch offline lokal auf dem Rechner zu testen. Zum Zugriff auf den Webservice von bplaced.net wurde Notepad++¹¹ mit seiner Erweiterung NPP-FTP genutzt, sowie FileZilla¹², um schnell und einfach Daten via ftp hinzuzufügen, zu ändern oder zu löschen. Um ein geräteunabhängiges Design zu gewährleisten, wurde die HTML-Boilerplate Skeleton¹³ benutzt, wobei es sich um ein schlankes CSS-Framework handelt mit integriertem Gridsystem, wodurch die Seite auf den verschiedensten Geräten anzeigbar und lesbar ist. Eine vollständige Liste aller verwendeten Technologien und Programme findet sich im Anhang.

6.4 ZUSAMMENFASSUNG

Zusammenfassend ist zu sagen, dass das Konzept weitestgehend übernommen wurde, auch wenn teilweise aufgrund von Rechenleistung einige Änderungen vorgenommen werden mussten. Der eigentliche Algorithmus wurde wenig geändert. Die Wahl von PHP als Programmiersprache erwies sich als sehr gut, da die Implementierung sehr schnell erfolgte und das Ergebnis wenig fehleranfällig ist. Auch die MySQL-Datenbank erwies sich bei der Wahl als richtig, da das Speichern, sowie Herausholen aus der Datenbank sehr einfach und schnell vonstattengeht.

⁹ <http://php.net/> (aufgerufen am 16.08.2013)

¹⁰ <http://www.apachefriends.org/de/xampp.html> (aufgerufen am 16.08.2013)

¹¹ <http://notepad-plus-plus.org/> (aufgerufen am 16.08.2013)

¹² <https://filezilla-project.org/> (aufgerufen am 16.08.2013)

¹³ <http://www.getskeleton.com/> (aufgerufen am 16.08.2013)

7 EVALUATION

Im Folgenden wird die Evaluation des in der Konzeption entwickelten Algorithmus erfolgen. Dabei wird zuerst die aufgestellte Anforderungsanalyse herangezogen und die einzelnen Punkte mittels der Implementierung auf Umsetzung überprüft. Im zweiten Teil der Evaluation soll der Algorithmus im zuvor ausgedachten Anwendungsfall überprüft werden, sodass gezeigt werden kann, dass er mit realen Daten lauffähig ist. Dazu wird eine Simulation des expliziten Ansatzes durchgeführt, wodurch sehr gute, realitätsnahe Werte erhoben werden können. Grund dieser Simulation ist der benötigte Datenbestand, der nicht zu klein ausfallen darf, um den Algorithmus optimal zu testen. Im nächsten Teil der Evaluation wird der Algorithmus unter Laborbedingungen getestet, wodurch Sonderfälle auf Umsetzung geprüft werden können, die zuvor noch nicht evaluierbar waren. Die Evaluation endet mit einer Zusammenfassung des Kapitels, wobei die gezeigten Schwächen und Stärken des Algorithmus noch einmal zusammengefasst werden.

7.1 ÜBERPRÜFUNG DER ANFORDERUNGSANALYSE

Im ersten Teil der Evaluation werden die in Abschnitt 5.1 entwickelten Anforderungen hinsichtlich ihrer Umsetzung in Konzeption und Implementierung überprüft. Dabei wird geschaut, welche Aspekte in der Implementierung vorgesehen wurden und in welcher Art und Weise diese umgesetzt sind. Im Folgenden werden die drei Hauptpunkte in jeweils einer Tabelle untersucht mit abschließender Schlussfolgerung. Ziel dieses Abschnitts ist die Überprüfung der aufgestellten Ziele, um zu zeigen, inwiefern diese umgesetzt wurden.

7.1.1 Der Algorithmus an sich

Nr.	Ziel	Erg.	Kommentar
1a	Submissions können nur durch registrierte Benutzer durchgeführt werden.	✓	Im umgesetzten expliziten Anwendungsfall können Submissions nur nach Login des Benutzers mit vorangegangener Registrierung abgegeben werden.
1b	Eingabe in den Algorithmus besteht nur aus UserID und Submission.	✓	Der Algorithmus nimmt als Eingabe lediglich UserID und Submission entgegen. Weitere Parameter werden durch den Crowdfunder getroffen.
1c	Der Algorithmus liefert eine Entscheidung und berechnet die Bewertung neu.	✓	Je nach vom Algorithmus ausgewähltem Pfad wird direkt eine Entscheidung getroffen oder nach einer abgelaufenen Wartezeit.
1d	Es muss eine Standardbewertung geben, sowie eine Maximal- und Minimalbewertung.	✓	Minimal- und Maximalbewertung wurden mit 0 und 1000 festgelegt. Standardbewertung wird durch den Crowdfunder bei der Einrichtung des Algorithmus bestimmt
1e	Es muss eine Möglichkeit der automatischen Sperrung und Entsperrung geben.	✓	Sperrung erfolgt bei der Neuberechnung der Benutzerbewertung, wenn der Wert einen Wert unterschreitet. Entsperrung findet sich im Alterungsprozess, der regelmäßig ausgeführt wird.
1f	Bewertungen sollen gegen einen Wert altern.	✓	Positive und negative Bewertungen altern gegen den Standardwert. Die Alterung erfolgt regelmäßig mittels Cron Job.
1g	Submissions von positiv bewerteten Benutzern sind wichtiger als von negativ Bewerteten .	✓	Die Umsetzung wurde in der Implementierung am Beispiel aus Tabelle 6.2 vorgestellt.
1h	Der Algorithmus soll mit mehreren Submissions gleichzeitig zurechtkommen.	✓	Die Überprüfung dieses Aspektes erfolgt in Abschnitt 7.3.
1i	Der Algorithmus soll sich bei Fehlentscheidungen nachträglich korrigieren können.	✗	Der Aspekt konnte im Rahmen der Arbeit nicht implementiert werden.
1j	Durch minimale Laufzeit soll im Algorithmus keine Verzögerung stattfinden.	✓	In der Implementierung wurde auf MySQLi gesetzt. Eine Verzögerung bei Ausführung des Algorithmus findet nicht statt.

Tabelle 7.1: Überprüfung des Punktes „Der Algorithmus an sich“ der Anforderungsanalyse.

In Tabelle 7.1 wurden die ersten Aspekte der Anforderungsanalyse auf Umsetzung in der Implementierung überprüft. Dabei handelte es sich um Aspekte, die den Algorithmus direkt betreffen, wobei alle erfüllt wurden. Die Prüfung von 1h erfordert die Evaluation unter Laborbedingungen und wird später genauer überprüft.

7.1.2 Festlegung von Werten durch Crowdfunder

Nr.	Ziel	Erg.	Kommentar
2a	Festlegung von Alterungsbeginn und Alterungsende.	✓	Sowohl Alterungsbeginn und Alterungsende werden bei der Einrichtung vom Crowdfunder abgefragt.
2b	Festlegung einer Zeit in der Submissions getätigt werden können.	✓	Der Crowdfunder muss bei der Einrichtung zwei Tageszeiten wählen, zwischen denen Submissions getätigt werden dürfen.
2c	Festlegung von Sperrungs- und Entsperrungswert.	✓	Parallel zu 2a erfolgt die Eingabe durch den Crowdfunder bei der Einrichtung.
2d	Festlegung der Höhe von Bewertungen.	✓	Alle 3 Arten möglicher Bewertungen können festgelegt werden.
2e	Festlegung der Lebenszeit von Submissions.	✓	Ebenfalls wie in 2a erfolgt die Eingabe bei der Einrichtung.
2f	Festlegung von weiteren Parametern.	✓	Variablen, die zum Schreiben einer Submission in die DB benötigt werden, können bestimmt werden.

Tabelle 7.2: Überprüfung des Punktes „Festlegung von Werten durch Crowdfunder“ der Anforderungsanalyse.

7.1.3 Möglichkeiten des Crowdfunders

Nr.	Ziel	Erg.	Kommentar
3a	Manuelle Sperrung von Benutzern (permanent/normal).	✓	Der Crowdfunder kann jedem der Benutzer eine Sperrvariable mittels Knopfdruck vergeben.
3b	Abrufen von gesperrten Benutzern.	✓	Der Crowdfunder kann sich eine Liste gesperrter Benutzer anzeigen lassen.
3c	Realisierung von verschiedenen Benutzergruppen.	✓	Der Crowdfunder kann ähnlich wie in 3a die Gruppe mittels Knopfdruck ändern.
3d	Nachträgliche Änderung der unter 2. eingetragenen Werte.	✓	Der Crowdfunder kann alle Werte nachträglich ändern.

Tabelle 7.3: Überprüfung des Punktes „Möglichkeiten des Crowdfunders“ der Anforderungsanalyse.

In Tabelle 7.2 und Tabelle 7.3 wurden weitere Aspekte der Anforderungsanalyse auf Umsetzung überprüft. Dabei ging es speziell um die Festlegung von verschiedenen Parametern, wobei neben der Festlegung auch auf die Verwendung dieser geprüft wurde. Dazu wurde geschaut, ob diese wirklich Einfluss auf das Ergebnis haben. Zudem ging es um weitere Möglichkeiten des Crowdfunders, die im Algorithmus vorgesehen werden sollen.

7.2 SIMULATION DES ANWENDUNGSFALLS

Im folgenden Teil der Evaluation soll der Algorithmus mit einer repräsentativen Datenmasse geprüft werden. Dazu wird eine Simulation des Anwendungsfalls durchgeführt, bei der die Daten mit denen bei einem expliziten Ansatz erhobenen Daten vergleichbar sein sollen. Dazu wird ein Fragebogen mit 20 Fragen in Form von Bildern, die einen Füllstand der Mensa zeigen, an 30 verschiedene Probanden verteilt. Ziel ist es, diese Füllstände, entsprechend wie im expliziten Ansatz, wenn man sich direkt in der Mensa befindet, mit „leer“, „halbvoll“ oder „voll“ einzuschätzen. Um die Vergleichbarkeit der Daten mit Realdaten zu garantieren, werden Messdaten als Grundlage betrachtet und eine Verteilung der Fragen anhand gemessener Füllstände vorgenommen. Ebenso wird eine feste Anzahl an Fragen an die Probanden verteilt, sowie ein Blindtest durchgeführt, was im Folgenden genauer auf Sinnhaftigkeit überprüft wird.

7.2.1 Vorüberlegungen und Vorbereitungen

Die Simulation wurde gewählt, um ein Mindestmaß an Daten garantieren zu können. Hierbei wurde entschieden, die vom WLAN-Management der TU Dresden erhobenen Daten als Grundlage für die Durchführung der Simulation zu wählen. Dabei handelt es sich um reale Messwerte der Tage vom 01.07.2013 bis 05.07.2013, sowie vom 08.07.2013 bis 12.07.2013, was zwei Mensawochen von je Montag bis Freitag darstellt. Diese zwei Zeiträume liegen in der Vorlesungszeit des Semesters, wodurch Daten erhoben werden können, die den normalen Mensa-Alltag darstellen, was in den Semesterferien nur teilweise realisiert werden könnte, da dort eine volle Mensa nur an einigen Uhrzeiten vorhanden ist. Die dabei verwendeten Daten finden sich im Anhang wieder. Neben den zwei Wochen, für die real gemessene Daten vorliegen, werden zwei weitere Wochen gewählt, in dem ein Blindtest der Probanden durchgeführt wird. Grund dafür ist, dass im Fragebogen die Zeit, sowie der Tag, angezeigt wird, welche im realen expliziten Ansatz ebenfalls bekannt wären. Damit sichergestellt werden kann, dass die Benutzer ihre Submissions vom Füllstand und nicht von anderen Faktoren abhängig machen, wird diese Art von Test durchgeführt. Zur Erstellung des Fragebogens allgemein ist es notwendig, anhand der Messergebnisse zu bestimmen, wie oft jeder einzelne Füllstand im Fragebogen vergeben werden muss.

Beispiel

Wenn die Alte Mensa öffnet, befinden sich weniger Besucher in der Mensa, wodurch davon auszugehen ist, dass in dieser Zeit weniger Submissions in den Algorithmus gelangen. Wenn die Alte Mensa sehr voll ist, müssen dementsprechend mehr Submissions erhoben werden.

Im Fragebogen wird das dadurch realisiert, dass es keinen festen Fragebogen gibt, sondern dieser automatisch erzeugt wird, sodass mehr Probanden an einem Tag eine Abbildung der Mensa zu einer stark besuchten Zeit erhalten (z.B. 13:00 Uhr), als eine Abbildung der Mensa zu einer wenig besuchten Zeit (z.B. 10:50Uhr).

Hierbei wird ausgeschlossen, dass Submissions eintreffen, wenn die Mensa geschlossen ist. Das wird dadurch realisiert, dass nur Abbildungen der Mensa zu geöffneten Zeiten ausgegeben werden. Grund dafür ist, dass der Algorithmus an sich auf Sinnhaftigkeit überprüft werden soll und die Grenzen nur eine optionale Funktion darstellen. Zudem wird die spätere Rechnung eine sehr geringe Wahrscheinlichkeit darstellen, dass Submissions abgegeben werden, falls die Mensa geschlossen ist, wodurch keine Daten in dieser Zeit verteilt werden müssen.

Als erster Schritt, der bei der Verteilung der Zeiten vorgenommen wird, müssen verschiedene Zeiträume gebildet werden, die repräsentativ für einen Füllstand sein werden. Dabei wurde die Öffnung der Alten Mensa mit 10:45Uhr und die Schließung der Mensa mit 15:00Uhr betrachtet, wobei 10:45Uhr als auch 15:00Uhr noch als Minute gezählt werden. Es handelt sich somit um 256 Minuten, in der die Mensa geöffnet ist. Bei der Auswahl eines Zeitintervalls wurde zuerst

ein 15-Minuten-Intervall betrachtet, was sich als zu engmaschig herausstellte und somit zu starken Schwankungen führt. Danach wurde ein 45-Minuten-Intervall betrachtet, was sich jedoch als zu groß herausstellte, da die Werte zu sehr abgeschwächt werden. Um einen guten Mittelwert zwischen den beiden betrachteten Intervallen zu realisieren, wurde ein 32-Minuten-Intervall gewählt, da dadurch äquidistante Intervalle realisiert werden können und das Intervall sich weder als zu klein, noch zu groß, herausstellte, wodurch der betrachtete Zeitraum ein gutes Mittelmaß für den Füllstand dieser Zeit darstellt. Die Realisierung von äquidistanten Intervallen ist wichtig, da so feste Zeiträume von der Öffnung bis zur Schließung der Mensa vorhanden sind. Erst damit ist es möglich, die durch die Simulation erhaltenen Werte miteinander zu vergleichen. Die vorgenommene Verteilung für den Fragebogen ist in Tabelle 7.4 dargestellt.

	1	2	3	4	5	6	7	8
Start	10:45:00	11:17:00	11:49:00	12:21:00	12:53:00	13:25:00	13:57:00	14:29:00
Ende	11:16:59	11:48:59	12:20:59	12:52:59	13:24:59	13:56:59	14:28:59	15:00:59

Tabelle 7.4: Die Verteilung der Intervalle der Uhrzeiten.

Anhand der Daten des WLAN-Managements der TU Dresden, die sich im Anhang befinden, wurde für jeden Tag und für sämtliche Zeiträume das arithmetische Mittel aller gemessenen verbundenen Geräte berechnet und in der Tabelle 7.5 zusammengefasst. Zur Darstellung wird auf 2 Stellen nach dem Komma gerundet, was in der richtigen Rechnung, die zur Weiterrechnung benutzt wurde, nicht geschah.

	1	2	3	4	5	6	7	8	Summe
01.07.	73,00	116,25	140,88	226,50	207,25	109,71	75,00	70,25	1018,84
02.07.	98,38	124,14	145,75	219,00	195,88	99,86	74,13	71,63	1028,77
03.07.	115,86	151,88	174,50	271,00	292,25	158,71	113,38	69,75	1347,33
04.07.	92,25	161,50	215,71	225,14	220,13	128,25	92,71	74,13	1209,82
05.07.	64,25	97,13	150,50	166,00	147,38	75,00	37,88	32,63	770,77
08.07.	105,13	140,14	166,25	234,25	175,75	90,71	70,71	71,75	1054,69
09.07.	94,14	110,88	130,88	191,86	219,71	110,13	74,75	63,88	996,23
10.07.	109,13	155,25	165,38	232,43	218,63	108,75	83,40	59,00	1131,97
11.07.	75,38	122,13	142,00	169,67	178,25	93,25	58,50	39,25	878,43
12.07.	46,88	63,63	95,50	141,63	116,14	62,75	38,38	37,88	602,79

Tabelle 7.5: Die Berechnung der arithmetischen Mittel für die Intervalle.

Mit Hilfe des arithmetischen Mittels der Intervalle und der Summe wurde ein Prozentwert bestimmt, den das Intervall vom Tageswert ausmacht.

Beispiel

Im Intervall 1 finden wir am 01.07. einen Durchschnitt von 73 verbundenen Benutzern.

$$\text{Prozentzahl} = \frac{\text{Durchschnitt_Intervall}}{\text{Gesamt}} \cdot 100 = \frac{73}{1018,84} \cdot 100 \approx 7,165\%$$

Das Intervall 1 macht am 01.07. rund 7,165% der Gesamtbesuchermenge pro Tag aus, d.h. 7,165% der Besucher waren in dieser Zeit in der Mensa.

Nachdem dieser Prozentsatz der Gesamtbesuchermenge pro Tag je Intervall errechnet wurde, wurde bestimmt, wie viele der 30 Benutzer je Zeitintervall in der Mensa waren.

$$\text{Benutzer_pro_Zeitintervall} = \frac{30}{\text{Prozentzahl}} \cdot 100$$

Diese Werte wurden anschließend gerundet und pro Tag summiert, wobei 30 als Ergebnis folgen sollte. In Fällen, in denen durch die Rundung keine 30 Besucher als Ergebnis folgten, wurde bei einzelnen Werten manuell geschaut, welche Differenz zur Komma Fünf vorhanden ist. Je nachdem, ob zur Anzahl von 30 ein Wert fehlte, oder zu viel war, wurde der mit der kleinsten Differenz entweder auf- oder abgerundet.

Beispiel

Am 01.07. wurden die Werte entsprechend bestimmt. Die Summe der Werte ergab 29, wodurch ein Wert noch aufgerundet werden musste. Der Wert am nächsten an der Komma Fünf war 3,423. Dieser wurde auf 4 aufgerundet, da es am wahrscheinlichsten ist, dass innerhalb dieses Intervalls der fehlende Benutzer in der Mensa war.

Das Ergebnis der Verteilung ist in Tabelle 7.6 zu sehen. Eingefärbt dargestellt sind die Werte, die manuell auf- bzw. abgerundet werden mussten.

	1	2	3	4	5	6	7	8	Summe
01.07.	2	4	4	7	6	3	2	2	30
02.07.	3	4	4	6	6	3	2	2	30
03.07.	3	3	4	6	6	4	2	2	30
04.07.	2	4	5	6	6	3	2	2	30
05.07.	3	4	6	6	6	3	1	1	30
08.07.	3	4	5	7	5	2	2	2	30
09.07.	3	3	4	6	7	3	2	2	30
10.07.	3	4	4	6	6	3	2	2	30
11.07.	3	4	5	6	6	3	2	1	30
12.07.	2	3	5	7	6	3	2	2	30

Tabelle 7.6: Das Ergebnis der Verteilung der Probanden auf die einzelnen Tage.

Nachdem eine Verteilung aufgestellt wurde, wie oft jedes Intervall in der Gesamtheit aller Fragebögen vorkommen muss, wird Tabelle 7.5 erneut aufgegriffen. Da es sich hierbei um Werte handelt, die den Durchschnitt der verbundenen Geräte darstellen, müssen diese noch in die Besucherzahl umgerechnet werden.

Dazu wurden die Werte gegen den Höchstwert der Messungen des WLAN-Managements normiert, der laut Tabelle im Anhang am 03.07.2013 um 12:51:33Uhr gemessen wurde und 347 beträgt. Dazu wurde angenommen, dass dieser Wert einem Füllstand von 100% entspricht, was in unserem Fall 1163 Besucher sind.

Die Zahl von 1163 Besuchern kommt daher, dass die einzelnen Plätze, die im Grundriss der Mensa eingezeichnet wurden, gezählt wurden.

$$\text{Besucherzahl_in_Intervall} = \frac{\text{Durchschnitt_Intervall}}{347} \cdot 1163$$

Schlussendlich wurden diese Werte gerundet. Das Ergebnis wird in Tabelle 7.7 dargestellt.

	1	2	3	4	5	6	7	8	Summe
01.07.	245	390	472	759	695	368	251	235	30
02.07.	330	416	488	734	656	335	248	240	30
03.07.	388	509	585	908	980	532	380	234	30
04.07.	309	541	723	755	738	430	311	248	30
05.07.	215	326	504	556	494	251	127	109	30
08.07.	352	470	557	785	589	304	237	240	30
09.07.	316	372	439	643	736	369	251	214	30
10.07.	366	520	554	779	733	364	280	198	30
11.07.	253	409	476	569	597	313	196	132	30
12.07.	157	213	320	475	389	210	129	127	30

Tabelle 7.7: Die Besucheranzahl der einzelnen Intervalle, wie sie im Fragebogen benutzt werden soll.

Ausgehend von Tabelle 7.7 war es nun die Aufgabe, die Füllstände geeignet darzustellen. Dazu wurde der vorher entworfene Sitzplan der Alten Mensa um rot eingefärbte Plätze ergänzt, die einen belegten Sitzplatz darstellen sollen. Die Verteilung wurde je Tag verschieden vorgenommen, um den Probanden des Fragebogens nicht das Gefühl zu geben, dass die Füllstände manuell verteilt wurden. Bei der Verteilung wurde darauf geachtet, dass das Verhalten von Besuchern der Alten Mensa berücksichtigt wird und beispielsweise die ersten Besucher an Fensterplätze, sowie einzelne Besucher an kleinere Tisch gesetzt werden. Dabei entstanden 80 verschiedene Bilder für die 10 Tage, die nun entsprechend der Verteilung nach Tabelle 7.6 auf die einzelnen Benutzer aufgeteilt werden mussten.

Dazu wurde ein Verteilungsmodell entwickelt, welches sich im Anhang wiederfindet. Das Modell basiert darauf, dass jeder Füllstand eine Anzahl besitzt, wie oft dieser vergeben werden soll. Um nun eine Verteilung durchzuführen, wird zuerst der Füllstand vergeben, der die höchste Anzahl besitzt. Je Tag ändert sich der Startpunkt dieser Verteilung. Motivation dahinter war, dass eine Verteilung stattfinden sollte, bei der jeder Benutzer sowohl leere, halbvolle, als auch volle Mensafüllstände angezeigt bekommt, wodurch kein Einfluss mehr dadurch gegeben wird, dass Benutzer während des Fragebogens durch immer wiederkehrende ähnliche Füllstände falsche Submissions treffen.

Im letzten Schritt der Vorbereitung wurden die Daten in einer Login-Tabelle gespeichert, in der alle 30 Benutzer vorhanden sind. Neben den 10 Fragen mit jeweils einem Wert für einen Mensafüllstand wurde ein Passwort generiert für den Benutzer, sowie eine Variable für die Lügner gesetzt, was jedoch nur der Orientierung dient, um später nachvollziehen zu können, wer Lügner war. Für den Blindtest wurde die doppelte Zeit der tatsächlichen Messzeit genommen, wodurch aus zwei Wochen vier Wochen wurden. Nachdem alle Werte der 10 Fragen eingetragen waren, wurde überlegt, wie die Verteilung des Blindtests gestaltet wird. Dazu mussten die Bilder alle noch einmal erscheinen. Damit das nicht auffällt, wurde entschieden, das Ganze rückwärts durchzuführen, weil Menschen sich eine Reihe vorwärts oft merken, rückwärts jedoch nicht. Da Bild 10 nicht nach dem Bild 10 kommen sollte, wurde bei Bild 5 rückwärts angefangen bis zur 1 und danach von Bild 10 rückwärts bis Bild 6. Die Verteilung der Fragen des Fragebogens einschließlich Blindtest wird in Tabelle 7.8 dargestellt.

	1(11)	2(12)	3(13)	4(14)	5(15)	6(16)	7(17)	8(18)	9(19)	10(20)
Frage 1-10	1	2	3	4	5	6	7	8	9	10
Blindtest 11-20	5	4	3	2	1	10	9	8	7	6

Tabelle 7.8: Das Ergebnis der Verteilung der Fragen im Blindtest.

7.2.2 Durchführung

Nach Abschluss aller Vorbereitungen wurde jedem der 30 Probanden einer der Accounts, bestehend aus Benutzernamen und Passwort, zugeschickt, sowie die URL zur Fragebogenseite. Da das System auch mit Benutzern zurecht kommen sollte, welche die Unwahrheit sagen, erhielten 4 der Probanden die Information darüber, dass sie den Test nicht ordnungsgemäß durchführen sollen. 2 dieser Probanden sollten dabei die Antworten immer so vertauschen, dass das Ergebnis gerade dem Ergebnis entspricht, was der Proband nicht empfindet sowie die 2 anderen Probanden, die komplett falsche Submissions liefern sollten. Dabei wurde auf die Verwendung von Sonderzeichen verwiesen, um den Algorithmus mit unsinnigen Submissions zu überprüfen. Somit wurden 4 von 30 Probanden als Lügner festgelegt, was eine Prozentzahl von rund 13% darstellt. Im Rahmen der Testung des Algorithmus schien dies ein realistischer Wert zu sein. Bestandteil der Arbeit war es jedoch nicht, diesen Wert weiter zu untersuchen.

Zur Durchführung des Fragebogens an sich, der sich im Anhang wiederfindet, muss der Proband die erhaltene URL in den Browser eingeben, und landet daraufhin im Login-Fenster, wo er aufgefordert wird, sich anzumelden. Die Anmeldung mit den vorgefertigten Accounts dient dazu, den Fragebogen in einem geregelterem Umfeld durchzuführen, um eventuelle Fehlerquellen auszuschließen.

Nach erfolgreicher Anmeldung gelangt der Proband auf die Übersichtsseite, wo er in der Überschrift direkt sehen kann, worum es gehen soll. Der Fragebogen an sich ist typisch wie jeder Standardfragebogen aufgebaut. Zuerst wird dem Probanden für seine Zeit, die er sich nimmt, gedankt. Anschließend erfolgen Informationen über den Fragebogen, wobei darauf hingewiesen wird, dass kein Zwang besteht einzelne Frage auszufüllen und der Proband sie auch überspringen kann. Am Ende des ersten Absatzes wird dem Proband jedoch Mut gemacht, dass seine Antworten richtig sein werden und es keine falschen Antworten geben wird.

Danach wird der Sinn und Zweck des Fragebogens kurz vorgestellt, sowie auf eine weitere Seite verwiesen, falls der Proband weitere Informationen über den Zweck des Fragebogens erhalten mag.

Nachdem der Proband mit dem Sinn und Zweck des Fragebogens vertraut gemacht wurde, folgt auf der Übersichtseite eine Beschreibung, in der erklärt wird, was auf den folgenden Bildern zu sehen ist und worauf diese Daten beruhen, sowie eine kurze Anleitung, in der dem Probanden mitgeteilt wird, was er zu tun hat. Nach der Beschreibung folgen weitere Hinweise, die zur Erfüllung der kommenden Aufgaben von Wichtigkeit geprägt sind. Falls der Proband keine weiteren Fragen hat, kann er den Fragebogen starten.

Anschließend gelangt er in den Fragenteil. Im oberen Teil der Seite findet sich die Frage: „leer, halbvoll oder voll?“, die vom Probanden zu erledigen ist. Dabei wurde er vorher darauf hingewiesen, dass er seinem eigenen Empfinden nach beurteilen soll, was er für richtig und falsch empfindet und es keine vorgeschriebene Lösung gibt. Oben rechts findet sich die aktuelle Seitenanzahl, wobei 21 gewählt wurde, da die letzte Seite die Danksagung an den Benutzer darstellt und er bei Seite 20 nicht plötzlich aufhören soll. Im Hauptteil der Seite ist das Bild vom Grundriss der Alten Mensa mit eingezeichnetem Füllstand zu sehen, welches zu bewerten ist. Rechts daneben findet sich die Zeit, auf der diese Daten, die eingezeichnet sind, beruhen. Im unteren Teil der Seite finden sich die Hinweise wieder, die zum Ausfüllen des Fragebogens essentiell sind.

Die Aufteilung der Seite wurde auf diese Weise gewählt, da das zu bewertende Bild im Vordergrund der Aufmerksamkeit stehen soll und der Benutzer sich nicht von Uhrzeit oder sonstigem ablenken lassen soll. Wenn er mit dem expliziten Ansatz in der Mensa Daten erhebt, ist die Uhr beispielsweise auch weniger im Vordergrund als die Mensa selbst.

Auf Seite 21 folgt die letzte Seite des Fragebogens, wo dem Probanden für seine Teilnahme und Mithilfe gedankt wird, sowie die Information gegeben wird, dass er den Fragebogen erfolgreich abgeschlossen hat. Anschließend hat er die Möglichkeit sich auszuloggen, wodurch die SESSION des Probanden zerstört wird. Anhand des Fragebogens wurde ein Datensatz erhoben, der im Folgenden zur Auswertung herangezogen wird, um den Algorithmus mit Daten zu versorgen.

7.2.3 Auswertung

Im Folgenden findet die Auswertung des Fragebogens, sowie die damit verbundene Prüfung des Algorithmus mit den Simulationsdaten, statt. Zuerst werden die durch den Fragebogen erhobenen Ergebnisse aufgelistet. Anschließend folgt ein Vergleich der durch den Blindtest erhobenen Submissions mit den normal erhobenen Submissions. Danach soll überprüft werden, ob ein gut besuchter Tag sich ebenfalls in den Ergebnissen widerspiegelt. Im letzten Schritt wird der Datensatz, der durch den Fragebogen erhoben wurde, zum Testen des Algorithmus herangezogen.

Ergebnis des Fragebogens

In Tabelle 7.9 wurden die Ergebnisse des Fragebogens tabellarisch aufbereitet. Im Folgenden wird die Tabelle zur Untersuchung des Blindtests, sowie zum Vergleich mit den realen Daten, genutzt.

	leer	halbvoll	voll	andere
01.07.	09 (09)	13 (11)	06 (08)	02 (02)
02.07.	09 (08)	08 (09)	11 (11)	02 (02)
03.07.	06 (08)	10 (10)	12 (10)	02 (02)
04.07.	08 (07)	06 (04)	14 (17)	02 (02)
05.07.	16 (13)	12 (14)	00 (01)	02 (02)
08.07.	09 (07)	11 (13)	08 (08)	02 (02)
09.07.	09 (10)	07 (05)	12 (13)	02 (02)
10.07.	08 (09)	13 (11)	07 (08)	02 (02)
11.07.	11 (12)	16 (13)	01 (03)	02 (02)
12.07.	20 (18)	06 (09)	02 (01)	02 (02)

Tabelle 7.9: Das Ergebnis des Fragebogens, bei dem je Tag die Anzahl der erhobenen Füllstände aufgelistet ist, sowie in Klammern zum Vergleich die Daten des Blindtests.

Folgend soll untersucht werden, wie stark die Benutzerbewertung vom angegebenen Tag abhängig ist. Dazu werden die in der Tabelle aufgelisteten Werte grafisch aufbereitet. Da im Fall von falsch eingegebenen Submissions, die in der Tabelle unter „andere“ aufgelistet sind, keine Schwankungen stattfanden, wird auf die Darstellung dieser verzichtet.

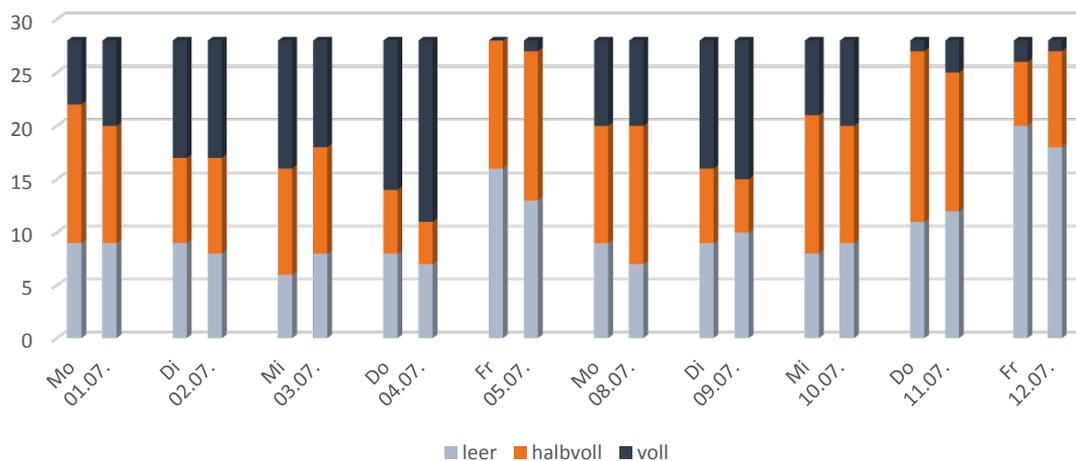


Abbildung 7.1: Das Diagramm zeigt die Tage, an denen Messergebnisse genutzt wurden, mit normal erhobenen Ergebnis (linker Balken), sowie dem Ergebnis des Blindtests (rechter Balken).

Der Blindtest im Fragebogen ergab, dass sich die Daten einerseits zwar sehr stark ähneln, andererseits jedoch ein Einfluss, der mit dem angezeigten Datum und der Uhrzeit zusammenhängt, vorhanden ist. Da die Abweichung, wie in Abbildung 7.1 ersichtlich, sehr gering auf das Gesamtergebnis hin ausfällt, wurde davon abgesehen und zum Testen des Algorithmus später die mit der richtigen Uhrzeit erhobenen Daten verwendet. Im tatsächlichen expliziten Ansatz kann ebenfalls Datum und Uhrzeit als Einfluss bei der Erhebung der Daten gesehen werden, wodurch sie in der Simulation ebenfalls angezeigt wurden.

Nachdem der Blindtest genauer betrachtet wurde, wird das Diagramm um die tatsächlich vorhandenen Daten ergänzt. Dazu wird das Diagramm aus Abbildung 7.1 um die in Tabelle 7.5 errechneten Summen erweitert, welche die Summe des Durchschnitts der einzelnen Intervalle darstellt.

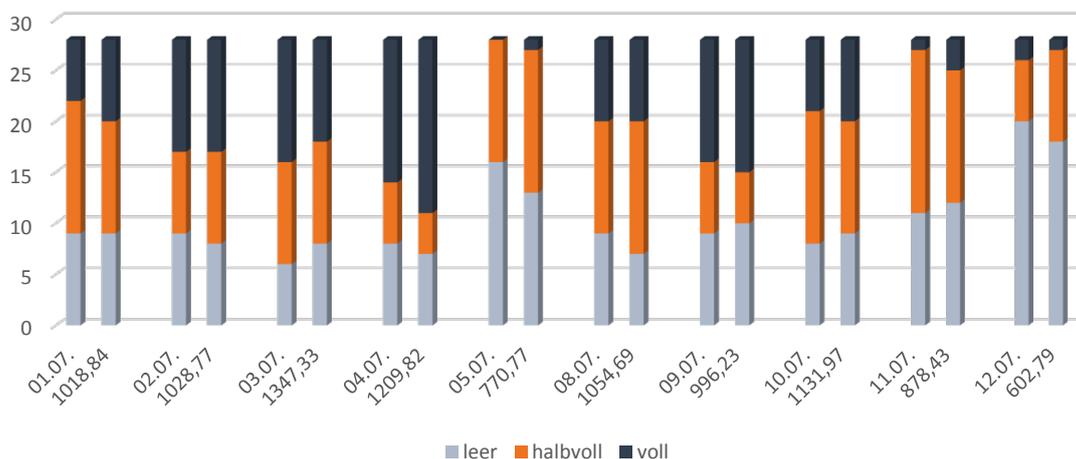


Abbildung 7.2: Das Diagramm zeigt, ähnlich wie Abbildung 7.1, die durch den Fragebogen erhobenen Ergebnisse. Anhand der unten genannten Werte soll überprüft werden, ob die Probanden diese in ihren Submissions widerspiegeln.

Abbildung 7.2 zeigt den Zusammenhang zwischen der Gesamtheit der Durchschnitte der Intervalle, sowie der Ergebnisse des Fragebogens. Ersichtlich ist, dass stark besuchte Tage wie der 02.07. oder 03.07. öfter als „voll“ eingeschätzt werden, als Tage wie der 05.07., an dem selbst

die Stoßzeiten als „halbvoll“ eingeschätzt wurden. Ebenfalls ersichtlich ist, dass von der Gesamtheit der Durchschnitte der Intervalle nur ungefähr auf die Anzahl von „leeren“, „halbvollen“ und „vollen“ Submissions geschlossen werden kann. Das kommt daher, dass wie in Tabelle 7.6 erkenntlich, Rundungen durchgeführt werden mussten und die Ergebnisse von reinen Einschätzungen von Benutzern abhängig sind, wodurch ein blankes Vertrauen auf verschiedenste Personen stattfinden muss.

Testung des Algorithmus

Nachdem die Ergebnisse untereinander, sowie mit den vom WLAN-Management der TU Dresden erhobenen Daten, verglichen wurden, folgt die Testung des Algorithmus mit dem erhobenen Datenbestand. Da herausgefunden wurde, dass mittels Anzeige des richtigen Datums und der Uhrzeit nur ein geringer Einfluss auf das Gesamtergebnis geschieht und dieser Einfluss durchaus berechtigt ist, wird im Folgenden der Datensatz ohne den Blindtest betrachtet.

Die folgenden Daten beruhen auf Ergebnissen des Algorithmus, die dadurch zu Stande gekommen sind, dass die im Fragebogen erhobenen Daten ausgelesen und durch den Algorithmus verarbeitet wurden. Zu Beginn soll darauf geschaut werden, wie stark der Anstieg der Benutzerbewertungen im Durchschnitt war, wie die Falschbewerter bewertet wurden und ebenso wo die maximale und minimale Bewertung liegt. Dazu wurden Parameter verwendet, die für den Autor der Arbeit als realistisch eingeschätzt wurden. Es wurde mit folgenden Werten getestet:

- Standardbewertung 500,
- Gering positive Bewertung 40,
- Positive Bewertung 80,
- Negative Bewertung -80,
- Annahme einer Submission ab 60% und
- Schreiben einer Submission ohne Vergleichswerte ab einer Bewertung von 800.

Eine Betrachtung des Einflusses der Parameter an sich folgt im späteren Teil dieses Kapitels. Zum reinen Test des Algorithmus wurde ohne Alterung und Sperrung getestet, sowie eine geringe Lebenszeit genommen, wodurch diese keinen Einfluss auf die folgenden Ergebnisse haben wird, da nur alle 32 Minuten ein Datensatz in das System kommt. Abbildung 7.3 zeigt die resultierenden Ergebnisse des Algorithmus.

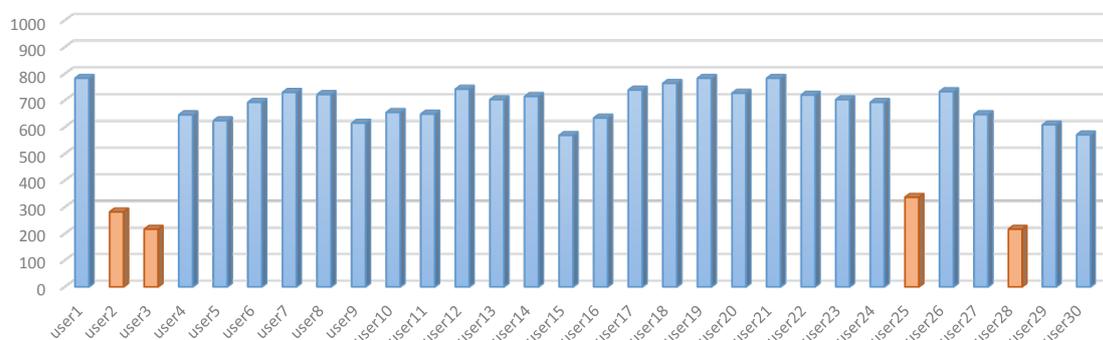


Abbildung 7.3: Das Balkendiagramm zeigt die Bewertungen der einzelnen Benutzer nach Ablauf der Simulation. Dabei befindet sich die Standardbewertung bei 500. User2 und User25 sind die beiden Benutzer, die immer das Gegenteil ihres Empfindens erhoben haben. User3 und User28 hingegen sind die beiden Benutzer, die immer komplett falsche Submissions abgegeben haben.

Von 300 eingegangenen Submissions wurden 195 in den Datenbestand übernommen, d.h. dass rund 2/3 aller Submissions positiv bewertet wurden. In Abbildung 7.4 wird die Verteilung der positiv bewerteten Submissions auf die Benutzer gezeigt.

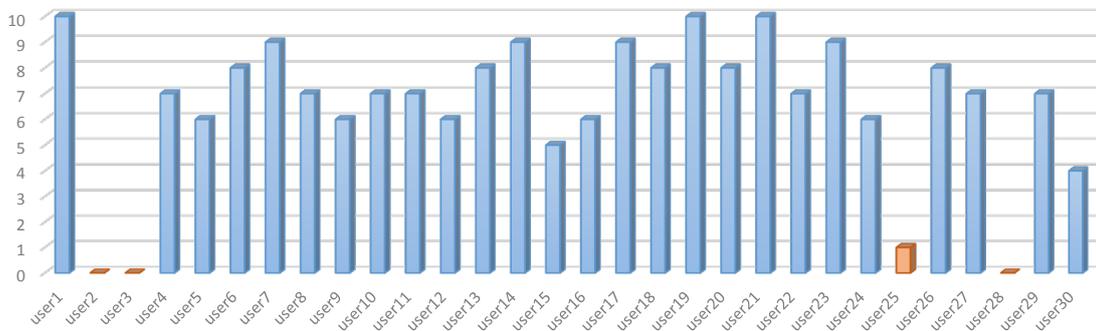


Abbildung 7.4: Das Balkendiagramm zeigt die Verteilung der positiv bewerteten Submissions auf die Benutzer. Genau wie in Abbildung 7.3 stellen User2, User3, User25 und User28 die Benutzer dar, die falsche Submissions erheben.

Das Diagramm aus Abbildung 7.3 zeigt das Ergebnis der Bewertungen der einzelnen Benutzer nach Ablauf der zwei Wochen, die in der Simulation erhoben wurden. Dabei ist die Funktionalität des Algorithmus zu sehen, der alle vier Probanden, die falsche Submissions erhoben haben, negativ bewertet. Alle anderen Probanden, die sinnvolle Submissions liefern sollten, wurden positiv bewertet.

Da es sich bei der bisherigen Testung lediglich um vom Autor ausgedachte Werte handelt, die sinnvoll für den Anwendungsfall erschienen, muss im Folgenden der Einfluss dieser Parameter genauer untersucht werden. Dazu soll jeder der einflussnehmenden Faktoren genauer beleuchtet, sowie das Ganze tabellarisch aufgearbeitet werden.

Im ersten Schritt wird die **Einflussnahme des Motivationsfaktors** untersucht.

Kriterien	20	40	60	80	100	120	140	160	180	200
Maximale Bewertung	783	783	783	783	800	817	834	848	863	876
Minimale Bewertung	217	217	217	217	217	217	217	217	217	217
Durchschnittsbewertung	627	634	640	646	651	668	675	682	688	694
Anz. akzept. Submissions	195	195	195	195	195	193	193	193	194	194

Tabelle 7.10: Der Einfluss des Motivationswertes im Algorithmus.

Tabelle 7.10 zeigt den Zusammenhang zwischen wachsendem Motivationsfaktor, der auch als geringe positive Bewertung bezeichnet wurde, und der Änderung der einzelnen Aspekte. Dabei fällt auf, dass der Motivationsfaktor Einfluss auf die maximale Bewertung im System hat, sowie daraus resultierend auch auf die Durchschnittsbewertung und die Anzahl akzeptierter Submissions.

Im zweiten Schritt wird die **Einflussnahme des positiven Bewertungsfaktors** untersucht.

Kriterien	20	40	60	80	100	120	140	160	180	200
Maximale Bewertung	624	667	730	783	826	861	889	913	931	946
Minimale Bewertung	217	217	217	217	217	217	217	217	217	217
Durchschnittsbewertung	504	548	593	634	669	699	725	748	767	785
Anz. akzept. Submissions	193	195	195	195	195	195	195	197	202	203

Tabelle 7.11: Der Einfluss des positiven Bewertungsfaktors im Algorithmus.

In Tabelle 7.11 wird der Einfluss der Änderung des positiven Bewertungsfaktors untersucht. Schnell wird festgestellt, dass die minimale Bewertung unverändert bleibt, die maximale Bewertung jedoch sehr stark steigt mit Zunahme des Bewertungsfaktors. Ebenso steigt natürlich die Durchschnittsbewertung stark an. Letztendlich werden mit höherem positiven Bewertungsfaktor mehr Submissions durch das System angenommen, als mit niedrigerem Faktor.

Nachdem der positive Bewertungsfaktor hinsichtlich seines Einflusses auf verschiedene Aspekte betrachtet wird, erfolgt in Tabelle 7.12 dasselbe für den **negativen Bewertungsfaktor**.

Kriterien	-20	-40	-60	-80	-100	-120	-140	-160	-180	-200
Maximale Bewertung	783	783	783	783	783	783	783	783	783	783
Minimale Bewertung	408	333	270	217	174	139	111	87	69	54
Durchschnittsbew.	693	674	656	634	617	602	584	571	559	551
Anz. akzept. Subm.	193	193	193	195	195	195	196	196	196	194

Tabelle 7.12: Der Einfluss des negativen Bewertungsfaktors im Algorithmus.

In Tabelle 7.12 ersichtlich ist die starke Veränderung des Minimums der Bewertung mit zunehmend größer negativer Bewertung. Je größer der Wert, desto schneller erreichen Probanden, die lügen, das Minimum. Die negative Bewertung nimmt keinen Einfluss auf die maximale Bewertung. Die Durchschnittsbewertung sinkt mit zunehmend größer negativer Bewertung. Die Anzahl akzeptierter Submissions schwankt je angenommenen Wert.

Nach der Betrachtung der Abhängigkeit von den einzelnen Bewertungsfaktoren folgt nun die **Betrachtung des Prozentsatzes**, der vorhanden sein muss, damit eine Submission angenommen wird.

Kriterien	55	60	65	70	75	80	85	90	95	100
Maximale Bewertung	783	783	783	773	773	773	773	773	773	773
Minimale Bewertung	217	217	217	217	217	217	217	217	217	217
Durchschnittsbewertung	623	634	646	651	652	655	658	658	658	658
Anz. akzept. Submissions	198	195	185	174	166	151	139	139	139	139

Tabelle 7.13: Der Einfluss des Prozentsatzes, der zur Annahme einer Submission benötigt wird.

Wie in Tabelle 7.13 erkenntlich, ist der Prozentsatz dazu da, zu bestimmen, wie viele Submissions vom Algorithmus als positiv bewertet werden sollen. Je kleiner der Wert ist, desto mehr Submissions werden als positiv bewertet, aber desto größer ist auch die Gefahr, dass es sich dabei um falsch erhobene Submissions handelt. Zudem ist in der Tabelle die Konvergierung der Werte ab einem bestimmten Prozentsatz zu sehen, ab dem sich nichts mehr ändert.

Nach der Prüfung der verschiedenen Parameter, die den eigentlichen Algorithmus ausmachen und mit der Simulation sinnvoll getestet werden können, soll als letztes der **Einfluss von Standardbewertungen** diskutiert werden.

Kriterien	100	200	300	400	500	600	700	800	900
Maximale Bewertung	609	653	696	740	783	825	869	914	956
Minimale Bewertung	44	86	131	175	217	260	304	347	391
Durchschnittsbewertung	447	495	542	588	634	679	732	778	823
Anz. akzept. Submissions	198	198	197	195	195	195	198	212	223

Tabelle 7.14: Der Einfluss des Prozentsatzes, der zur Annahme einer Submission benötigt wird.

Wie in Tabelle 7.14 erkenntlich, kann durch das Setzen von verschiedenen Standardbewertungen direkt Einfluss auf sämtliche genannten Aspekte genommen werden. Durch die Realisierung von Gruppen, bei denen verschiedene Standardbewertungen genutzt werden, können so Benutzergruppen geschaffen werden, die leichter oder schwerer Submissions erheben können.

Mit der Simulation konnte der Algorithmus mit realitätsnahen Daten getestet werden, wodurch die Korrektheit der entwickelten Implementierung gezeigt werden konnte. Zudem konnten verschiedene, für den Algorithmus sehr wichtige, Parameter hinsichtlich ihres Einflusses betrachtet werden.

7.3 TEST UNTER LABORBEDINGUNGEN

Im folgenden Unterabschnitt erfolgt der Test der Implementierung unter Laborbedingungen. Dazu wurden fünf verschiedene Geräte gewählt, die mit fünf verschiedenen Benutzer-Accounts eingeloggt sind. Anschließend geben diese möglichst gleichzeitig eine Submission ab. Der Algorithmus soll in Hinsicht darauf getestet werden, was dann passiert, wenn Submissions gleichzeitig in das System kommen. Die folgenden 3 Tests, wie in den Tabellen dargestellt, wurden durchgeführt.

User	Bewertung	Submission	Zeitstempel
user5	500	leer	1378063542
user1	500	leer	1378063542
user4	500	leer	1378063542
user3	500	leer	1378063543
user2	500	leer	1378063543

Tabelle 7.15: Im Test sollten alle Probanden die Bewertung „leer“ abgeben.

In Tabelle 7.15 haben 3 der Probanden die Submission in der selben Sekunde erhoben, sowie die anderen 2 in der Sekunde darauf. Als Ergebnis wurde die Submission „leer“ angenommen und jeder der 5 Benutzer erhielt eine positive Bewertung.

User	Bewertung	Submission	Zeitstempel
user5	540	voll	1378063682
user1	540	voll	1378063682
user4	540	voll	1378063682
user3	540	voll	1378063683
user2	540	voll	1378063683

Tabelle 7.16: Im Test sollten alle Probanden die Bewertung „voll“ abgeben.

In Tabelle 7.16 haben 3 der Probanden die Submission in der selben Sekunde erhoben, sowie die anderen 2 in der Sekunde darauf. Auffällig ist die gleiche Verteilung wie beim ersten Durchlauf. Ähnlich wie in Durchlauf 1 wurden alle 5 Probanden positiv bewertet.

User	Bewertung	Submission	Zeitstempel
user5	577	leer	1378063840
user1	577	voll	1378063840
user4	577	leer	1378063840
user3	577	voll	1378063841
user2	577	voll	1378063841

Tabelle 7.17: Im Test sollten 2 der Probanden die Bewertung „leer“, sowie 3 der Probanden die Bewertung „voll“ abgeben.

In Tabelle 7.17 fällt ebenfalls wieder die gleiche Reihenfolge der Benutzer auf, die bei der Ausführung herrscht. Der Algorithmus handelt im implementierten Sinn richtig, da er nach der Submission von user4 die Submission leer annimmt. Durch den Test mittels gleichzeitiger Submissions konnten keine Probleme festgestellt werden, die durch ein gleichzeitiges Submittieren geschehen können. Dies ist auf die Verwendung von bplaced zurückzuführen, die ihre Benutzer vor unvorhersehbaren Problemen wie Verklemmungen beschützen.

Da es aber möglich ist, dass der Algorithmus in anderen verwendeten Diensten verklemmen könnte, muss eine Lösung für dieses Problem geschaffen werden, die auch funktioniert, wenn ohne den Service von bplaced gearbeitet wird. Der folgende Ansatz wird dabei einen möglichen Lösungsansatz nennen.

Um Probleme mit gleichzeitigen Submissions zu verhindern, werden alle eingehenden Submissions in einer Warteschlange gespeichert. Dort werden sie mittels Cronjob regelmäßig sequentiell herausgenommen und verarbeitet. Durch diese Art von Puffern könnte verhindert werden, dass jegliche Probleme mit gleichzeitigen Submissions auftreten können. Im Rahmen der Arbeit wurde der Ansatz nur genannt, jedoch nicht implementiert, da im Anwendungsfall keine Probleme mit gleichzeitigen Submissions auftraten. In einer future-work sollte der Ansatz, ebenso wie die Grundgesamtheit, eingearbeitet werden.

8 ZUSAMMENFASSUNG UND AUSBLICK

ZUSAMMENFASSUNG

In der Arbeit wurde ein Bewertungsalgorithmus für Crowdsourcing-Benutzer und deren Submissions erarbeitet und in einer Simulation eines Anwendungsfalls auf Funktionalität getestet.

Nachdem der Benutzer zu Beginn für die Arbeit motiviert wurde, folgten die Grundlagen zum Thema, in denen der Leser mit den in der Arbeit benutzten Begrifflichkeiten vertraut gemacht wurde. Dabei wurde ihm der Begriff des Crowdsourcing näher gebracht, indem bekannte Beispiele wie der Super Bowl aufgegriffen wurden, um den Einstieg in die Materie zu erleichtern. Danach wurde Ausschau nach verschiedenen Ansätzen für den umzusetzenden Crowdsourcing-Anwendungsfall gehalten, indem verschiedene mögliche Methoden kurz beschrieben wurden. Nachdem alle Grundlagen geklärt wurden, begann die Phase der Ideenfindung. Dazu wurde das Thema von Bewertungssystemen genauer betrachtet, indem der Sinn und Zweck von Bewertungen, sowie die Art des zu entwickelnden Bewertungsverfahrens, diskutiert wurde. Hierbei kam die Erkenntnis, dass Crowdsourcing zwar sehr effektiv in der Datenerhebung ist, jedoch ohne Bewertungsverfahren keinen Aufschluss über die Art der erhobenen Submission geben kann. Zudem wurde eine Grenze zwischen der Bewertung der Submission und der Benutzerbewertung gezogen, die dazu dient, dass die Bewertung der Submission nicht direkt in die Benutzerbewertung einbezogen wird, sondern nach Höhe der Benutzerbewertung erst berechnet wird. Im Folgenden wurden die existierende Bewertungsverfahren YouTube, eBay und Stack Overflow betrachtet, da es dabei um drei Webdienste handelt, die sehr weit entwickelte und funktionale Bewertungsverfahren verwenden. Ziel war es, eine Liste von Aspekten zu erarbeiten, die in das Konzept einfließen sollte. Erst dadurch wurde es möglich zu verstehen, worauf der Erfolg eines solchen Algorithmus basiert. Danach wurden die in den Grundlagen beschriebenen Lösungsansätze für den Crowdsourcing-Anwendungsfall betrachtet, woraufhin der explizite Ansatz in der Zusammenfassung des Kapitels ausgewählt wurde. Dabei handelt es sich um eine sehr einfache, jedoch sehr gut funktionale Lösung, die eine breite Benutzergruppe anspricht. In der Konzeptionsphase wurden die zuvor erarbeiteten Aspekte in einer Anforderungsanalyse, sowie danach in einem Entwurf, zusammengefasst. Dabei wurden die Kernaspekte, die Bewertung einer Submission, der Alterungsprozess, die Sperrung, die Erhebung einer Submission, die Warteliste, sowie die Historie, genauer betrachtet. Ziel des Kapitels war es, den Entwurf geeignet darzustellen, was dadurch realisiert wurde, dass zuerst fünf Hauptbestandteile des Bewertungsalgorithmus gebildet wurden und diese anschließend in Pseudo-Code und grafisch verdeutlicht wurden. Diese fünf Bestandteile des Algorithmus wurden danach in der Implementierungsphase aufgegriffen

und prototypisch umgesetzt. Die dabei aufgetretenen Probleme zogen sich von enormer Rechenlast bis hin zur Nichtumsetzbarkeit in der vorhandenen Zeit. Im Folgenden wurde der Algorithmus mittels Anforderungsanalyse geprüft, sowie eine Simulation des Anwendungsfalls realisiert, wodurch die Effizienz anhand eines Beispiels gezeigt werden konnte. Dabei war es möglich die Funktionalität des zu entwickelnden Algorithmus zu zeigen.

AUSBLICK

Ziel dieser Arbeit war die Schaffung eines automatisierbaren Algorithmus zur Bewertung von Crowdsourcing-Nutzern und deren Submissions. Mit der zuvor durchgeführten Evaluation wurde gezeigt, dass eine funktionale und effektive Methode geschaffen wurde, welche die gestellte Aufgabe erfüllt. Weitere Aspekte, wie Funktionalitäten des Crowdfunders, wurden ebenfalls implementiert. Im Folgenden sollen Erweiterungen kurz angesprochen werden, die beim Schreiben der Arbeit entstanden.

Einerseits sollte der Algorithmus noch um eine Historie, sowie eine Warteschlange erweitert werden, um die Effizienz noch weiter zu steigern. Die Historie prüft den erhobenen Datenbestand regelmäßig daraufhin, inwieweit die vom Algorithmus erhobenen Daten mit der Grundgesamtheit übereinstimmen. Dies könnte dazu dienen, den Algorithmus nachträglich zu korrigieren, falls Betrüger versuchen, falsche Submissions in den Datenbestand zu bekommen, um damit anderen Benutzern Schaden zuzufügen. Die Umsetzung der Warteschlange dient dazu, jegliche Art von Fehlerquellen, die durch das gleichzeitige Erheben von Submissions geschehen könnten, auszuschließen. Dabei wird jede Submission zwischengespeichert und lediglich zu bestimmten Zeiten sequentiell in den Algorithmus gegeben, wodurch Probleme ausgeschlossen und die Effizienz bei der Ausführung noch einmal gesteigert wird.

Andererseits fiel in der Implementierung ein weiterer Aspekt auf. Im Rahmen der Bachelorarbeit wurde die Gleichheit mit Ähnlichkeit einer Submission gleichgesetzt, was je nach Anwendungsfall nicht immer ausreichend sein wird. Ein weiteres Projekt könnte sich mit der Untersuchung der Ähnlichkeit von Crowdsourcing-Submissions beschäftigen und diese in den Algorithmus integrieren. Damit wäre es möglich, weitere Anwendungsfälle, in denen die Ähnlichkeit nicht der Gleichheit entspricht, mittels Crowdsourcing umzusetzen, wodurch das Anwendungsfeld sehr stark vergrößert werden könnte.

Im Rahmen des MapBiquitous-Projekts der TU Dresden, indem Crowdsourcing-Benutzer vorgesehen sind, könnte der Bewertungsalgorithmus aufgegriffen und sinnvoll integriert werden. Voraussetzung dafür ist die vorherige Betrachtung der Ähnlichkeit von Crowdsourcing-Submissions, da diese sehr unterschiedlich aussehen können.

A ANHANG

VERWENDETE PROGRAMME, TOOLS UND WEBDIENSTE

Folgende Programme, Tools und Webdienste wurden zum Schreiben der Bachelorarbeit verwendet:

- Adobe Reader XI,
- Adobe Illustrator CS6,
- Adobe Photoshop CS6,
- Android SDK v21.1.0,
- bplaced.net,
- cronjob.de,
- Eclipse Juno,
- Filezilla 3.6.0.2,
- golang.de,
- Google Translate,
- Microsoft Office Excel 2013,
- Microsoft Office Visio 2013,
- Microsoft Office Word 2013,
- MiKTeX 2.9,
- Notepad++ v6.2.2,
- Skeleton 1.2,
- Snipping Tool,
- stackoverflow.com,
- tex.stackexchange.com,
- Tortoise SVN 1.7.10,
- wikibooks.org,
- wikipedia.org und
- XAMPP v3.1.0 3.1.0.

WLAN-MANAGEMENT TESTDATEN I

01.07.2013		02.07.2013		03.07.2013		04.07.2013		05.07.2013	
Time	ACC ¹⁴	Time	ACC	Time	ACC	Time	ACC	Time	ACC
10:27:21	24	10:26:39	52	10:27:47	36	10:25:48	34	10:26:08	21
10:31:20	22	10:30:40	51	10:31:50	37	10:29:51	37	10:30:13	21
10:35:23	23	10:34:45	56	10:35:55	43	10:33:54	40	10:34:13	26
10:39:27	25	10:38:51	55	10:39:59	44	10:38:06	37	10:38:16	32
10:43:34	34	10:42:53	59	10:44:02	47	10:42:06	39	10:42:19	37
10:47:37	38	10:46:57	77	10:48:10	63	10:46:10	53	10:46:27	39
10:51:42	43	10:51:06	79	10:52:18	73	10:50:19	56	10:50:27	43
10:55:52	47	10:56:02	86	10:56:28	100	10:55:12	58	10:54:35	56
11:01:07	71	10:59:36	99	11:00:49	124	10:58:54	95	10:58:42	62
11:05:05	101	11:03:44	121	11:04:50	142	11:03:03	125	11:02:51	79
11:09:09	110	11:07:51	106	11:08:59	159	11:07:08	115	11:06:53	80
11:13:13	101	11:12:30	110	11:13:03	150	11:11:54	109	11:11:04	80
11:17:42	105	11:16:28	109	11:17:07	150	11:15:52	127	11:15:04	75
11:21:31	107	11:20:32	121	11:21:12	157	11:19:54	137	11:19:09	79
11:25:35	113	11:24:36	124	11:25:17	163	11:24:01	147	11:23:11	89
11:29:39	115	11:29:07	122	11:29:20	166	11:28:21	165	11:27:15	91
11:33:54	123	11:32:58	118	11:33:25	163	11:32:13	163	11:31:20	99
11:37:50	125	11:37:02	120	11:37:29	141	11:36:18	156	11:35:25	95
11:41:54	127	11:41:07	131	11:41:33	145	11:40:21	171	11:39:25	106
11:45:58	115	11:45:30	133	11:45:38	130	11:44:49	173	11:43:28	102
11:50:11	116	11:49:23	128	11:49:42	133	11:48:39	180	11:47:32	116
11:54:08	125	11:53:27	121	11:53:46	146	11:52:43	189	11:51:46	123
11:58:13	107	11:57:32	131	11:57:53	156	11:56:50	207	11:55:45	121
12:02:19	126	12:02:17	142	12:01:58	174	12:01:38	222	11:59:50	134
12:06:34	141	12:05:57	155	12:06:02	170	12:05:24	215	12:03:54	158
12:10:37	163	12:10:01	156	12:10:06	188	12:09:30	228	12:08:03	163
12:14:41	176	12:14:06	164	12:14:11	213	12:13:34	232	12:12:03	163
12:18:46	173	12:18:32	169	12:18:18	216	12:17:55	217	12:16:08	170
12:22:54	179	12:22:28	191	12:22:23	233	12:21:50	225	12:20:12	172
12:26:59	188	12:26:34	204	12:26:32	227	12:25:55	230	12:24:27	170
12:31:06	208	12:30:42	226	12:30:37	233	12:30:02	238	12:28:30	173
12:35:11	223	12:35:13	219	12:34:42	244	12:34:22	236	12:32:35	176
12:39:17	235	12:39:03	225	12:38:52	269	12:38:21	223	12:36:41	156

siehe nächste Seite ↪

¹⁴ Associated Client Count

01.07.2013		02.07.2013		03.07.2013		04.07.2013		05.07.2013	
Time	ACC								
12:43:31	258	12:43:14	218	12:43:06	284	12:42:30	213	12:40:48	167
12:47:45	254	12:47:24	250	12:47:21	331	12:46:44	211	12:44:54	158
12:51:53	267	12:55:55	265	12:51:33	347	12:55:19	253	12:49:02	162
12:56:08	266	13:00:03	244	12:55:46	341	12:59:25	237	12:53:07	157
13:00:15	236	13:04:07	214	12:59:49	335	13:03:30	236	12:57:28	160
13:04:14	225	13:08:12	188	13:03:53	315	13:07:37	228	13:01:22	165
13:08:20	223	13:12:18	187	13:07:58	311	13:11:41	229	13:05:26	157
13:12:23	213	13:16:22	175	13:12:17	301	13:15:46	212	13:09:31	149
13:16:30	190	13:20:37	152	13:16:20	285	13:19:52	192	13:13:37	136
13:20:33	161	13:24:35	142	13:20:25	238	13:23:56	174	13:17:39	135
13:24:37	144	13:28:40	139	13:24:31	212	13:28:03	153	13:21:43	120
13:28:42	138	13:32:47	124	13:28:41	187	13:32:07	152	13:25:48	103
13:32:51	139	13:36:50	107	13:32:43	172	13:36:12	137	13:30:12	82
13:36:52	131	13:40:53	84	13:36:47	171	13:40:15	127	13:34:09	86
13:40:57	111	13:44:58	82	13:40:54	157	13:44:27	125	13:38:09	75
13:45:01	94	13:49:02	86	13:45:01	144	13:48:29	117	13:42:18	67
13:49:10	77	13:53:06	77	13:49:04	146	13:52:34	113	13:46:18	65
13:53:11	78	13:57:11	79	13:53:09	134	13:56:38	102	13:50:21	63
13:57:17	73	14:01:23	80	13:57:13	126	14:00:47	97	13:54:28	59
14:01:24	73	14:05:24	80	14:01:34	127	14:04:49	99	13:58:30	56
14:05:28	70	14:09:29	73	14:05:31	114	14:08:53	90	14:02:32	44
14:09:34	71	14:13:33	74	14:09:37	122	14:12:57	89	14:06:35	32
14:13:38	76	14:17:40	65	14:13:40	106	14:17:08	94	14:10:38	34
14:17:43	77	14:21:44	72	14:17:52	111	14:21:11	94	14:14:41	33
14:21:50	80	14:25:49	70	14:21:53	102	14:25:14	86	14:18:44	34
14:25:55	80	14:29:57	70	14:25:58	99	14:29:22	90	14:22:47	35
14:34:31	58	14:38:25	73	14:34:34	87	14:37:42	80	14:30:54	42
14:38:29	79	14:42:28	88	14:38:27	94	14:41:52	82	14:34:59	38
14:42:33	76	14:46:32	82	14:42:33	87	14:46:00	67	14:39:04	39
14:46:58	73	14:50:42	75	14:46:40	68	14:50:06	60	14:43:07	37
14:50:58	73	14:54:46	62	14:51:02	56	14:54:11	64	14:47:11	32
14:55:03	66	14:58:50	56	14:54:55	42	14:58:16	64	14:51:14	26
14:59:07	60	15:02:54	46	14:59:00	37	15:02:20	56	14:55:27	24
15:03:12	53	15:06:59	50	15:03:04	34	15:06:25	50	14:59:26	23
15:07:16	46	15:11:06	44	15:07:14	26	15:10:29	42	15:04:42	27

Tabelle A.1: Woche 1 der vom WLAN-Management der TU Dresden erhobenen Daten, die zum Erstellung des Fragebogens herangezogen wurden.

WLAN-MANAGEMENT TESTDATEN II

08.07.2013		09.07.2013		10.07.2013		11.07.2013		12.07.2013	
Time	ACC ¹⁵	Time	ACC	Time	ACC	Time	ACC	Time	ACC
10:27:16	15	10:27:46	38	10:25:56	27	10:28:56	18	10:25:42	16
10:31:15	17	10:31:57	38	10:30:00	30	10:33:03	25	10:29:47	18
10:35:18	17	10:35:57	39	10:34:03	31	10:37:04	23	10:33:47	17
10:39:25	21	10:40:02	46	10:38:07	37	10:41:07	29	10:37:56	17
10:43:28	22	10:44:08	58	10:42:11	36	10:45:14	40	10:41:58	17
10:47:31	29	10:48:18	66	10:46:16	42	10:49:28	39	10:46:02	27
10:51:38	43	10:52:28	69	10:50:24	53	10:53:28	45	10:50:07	34
10:55:52	46	10:56:39	100	10:54:35	62	10:57:44	53	10:54:19	41
11:00:48	95	11:00:58	110	10:58:52	98	11:01:55	83	10:58:19	46
11:04:15	138	11:06:00	119	11:03:00	137	11:06:46	105	11:02:25	50
11:08:21	162	11:09:53	100	11:07:08	155	11:10:45	119	11:06:29	58
11:12:31	163	11:13:54	95	11:11:13	160	11:14:53	119	11:10:40	62
11:16:54	165	11:18:01	99	11:15:18	166	11:19:14	117	11:14:39	57
11:20:39	160	11:22:25	102	11:19:23	166	11:23:27	118	11:18:43	60
11:24:43	153	11:26:22	102	11:23:27	161	11:27:17	122	11:22:47	51
11:28:46	144	11:30:25	101	11:27:32	167	11:31:21	125	11:26:52	57
11:32:55	129	11:34:32	104	11:31:36	160	11:35:27	114	11:30:54	63
11:36:58	134	11:38:37	128	11:35:42	151	11:39:38	121	11:34:58	72
11:41:03	131	11:42:40	128	11:39:47	150	11:43:38	125	11:39:02	76
11:45:09	130	11:46:46	123	11:43:51	150	11:47:41	135	11:43:22	66
11:49:15	129	11:50:50	124	11:47:55	137	11:51:47	129	11:47:15	64
11:53:18	139	11:54:55	118	11:52:02	146	11:55:59	123	11:51:19	81
11:57:22	149	11:59:01	121	11:56:08	135	12:00:02	140	11:55:22	81
12:01:29	165	12:03:06	115	12:00:23	153	12:04:05	151	11:59:31	91
12:05:38	170	12:07:12	126	12:04:23	155	12:08:10	145	12:03:33	91
12:09:41	181	12:11:18	141	12:08:31	172	12:12:16	139	12:07:37	98
12:13:45	190	12:15:24	145	12:12:36	180	12:16:22	157	12:11:41	101
12:17:52	207	12:19:31	157	12:16:42	188	12:20:27	152	12:15:49	116
12:22:00	218	12:23:36	166	12:20:48	194	12:24:31	160	12:19:51	105
12:26:04	218	12:27:43	183	12:24:55	201	12:28:37	168	12:23:55	115
12:30:10	241	12:31:50	192	12:28:59	204	12:32:42	156	12:28:00	115
12:34:16	235	12:35:57	182	12:33:08	234	12:36:48	163	12:32:20	126
12:38:47	243	12:40:06	168	12:37:14	241	12:40:55	174	12:36:16	146

siehe nächste Seite ↪

¹⁵ Associated Client Count

08.07.2013		09.07.2013		10.07.2013		11.07.2013		12.07.2013	
Time	ACC								
12:42:40	232	12:48:40	215	12:41:23	234	12:49:33	197	12:40:21	153
12:46:53	239	12:52:49	237	12:45:41	247	12:53:40	201	12:44:28	158
12:51:02	248	12:56:55	257	12:49:51	266	12:57:49	209	12:48:41	167
12:55:23	246	13:01:06	246	12:54:00	277	13:02:18	198	12:52:42	153
12:59:25	208	13:05:14	229	12:58:07	242	13:06:02	192	12:56:48	148
13:03:30	188	13:09:19	230	13:02:17	239	13:10:06	175	13:00:57	129
13:07:37	180	13:13:25	217	13:07:04	229	13:14:12	173	13:05:04	116
13:11:48	164	13:17:37	190	13:11:02	214	13:18:17	144	13:09:06	103
13:15:50	139	13:21:40	169	13:15:06	203	13:22:21	134	13:13:10	114
13:19:56	144	13:25:47	149	13:19:11	186	13:26:25	129	13:17:15	100
13:24:01	137	13:29:50	133	13:23:40	159	13:30:31	112	13:21:28	103
13:28:21	112	13:33:55	121	13:27:29	139	13:34:35	107	13:25:29	94
13:36:35	107	13:37:58	98	13:31:34	131	13:38:38	98	13:29:33	82
13:40:36	103	13:42:02	94	13:35:40	119	13:42:44	90	13:33:37	74
13:44:43	87	13:46:07	96	13:39:48	113	13:46:47	75	13:37:43	57
13:48:44	86	13:50:13	98	13:43:50	104	13:50:50	73	13:41:43	58
13:52:48	71	13:54:17	92	13:47:54	95	13:54:56	62	13:45:47	49
13:56:51	69	13:58:23	81	13:52:04	90	13:59:00	60	13:49:50	46
14:01:06	64	14:02:27	79	13:56:09	79	14:03:06	68	13:54:05	42
14:05:03	65	14:06:32	76	14:00:29	82	14:07:09	62	13:58:01	43
14:09:08	67	14:10:40	70	14:04:23	85	14:11:14	66	14:02:05	39
14:13:14	68	14:14:43	76	14:08:27	84	14:15:18	53	14:06:08	43
14:17:19	74	14:18:46	73	14:17:25	84	14:19:23	57	14:10:18	31
14:21:23	75	14:22:51	73	14:22:05	82	14:23:30	52	14:14:19	28
14:25:29	82	14:27:01	70	14:58:05	59	14:27:36	50	14:18:23	36
14:29:35	86	14:31:08	71	15:01:22	48	14:31:44	46	14:22:26	43
14:33:52	75	14:35:21	65	15:05:22	43	14:35:54	47	14:26:32	44
14:37:57	73	14:39:31	71	15:09:26	41	14:40:01	48	14:30:35	38
14:42:08	85	14:43:41	67	15:13:30	26	14:44:07	41	14:34:39	36
14:46:18	67	14:47:45	66	15:17:35	21	14:48:12	38	14:38:43	41
14:50:26	64	14:51:52	57	15:21:39	17	14:52:17	38	14:42:51	39
14:54:29	66	14:55:56	61	15:25:43	17	14:56:21	30	14:46:51	38
14:58:38	58	15:00:00	53	15:29:47	18	15:00:33	26	14:50:56	42
15:02:39	52	15:04:05	41	15:33:52	19	15:04:33	29	14:55:00	35
15:06:46	41	15:08:10	35	15:37:55	17	15:08:37	26	14:59:05	34
15:10:48	28	15:12:15	32	15:42:00	21	15:12:40	25	15:03:06	26
15:14:52	25	15:16:19	24	15:46:03	17	15:16:46	19	15:07:09	23

Tabelle A.2: Woche 2 der vom WLAN-Management der TU Dresden erhobenen Daten, die zum Erstellung des Fragebogens herangezogen wurden.

VERTEILUNGSTABELLE FÜR DEN FRAGEBOGEN

Benutzer\Tag	1	2	3	4	5	6	7	8	9	10	∅
user1	759	335	585	248	504	470	736	520	476	320	495,3
user2	759	248	908	309	556	557	369	554	569	475	530,4
user3	695	240	980	541	494	785	316	779	597	389	581,6
user4	759	734	532	723	251	589	372	733	253	320	526,6
user5	695	656	380	755	215	304	439	364	409	475	469,2
user6	390	734	234	738	326	237	643	366	476	389	453,3
user7	472	656	908	430	504	240	736	520	569	213	524,8
user8	759	416	980	311	556	352	369	554	597	320	521,4
user9	695	488	908	248	494	470	251	779	313	475	512,1
user10	390	734	980	755	251	557	214	733	253	389	525,6
user11	472	656	585	738	127	785	316	364	409	210	466,2
user12	759	330	908	723	109	589	372	280	476	157	470,3
user13	695	416	980	755	504	304	439	198	569	213	507,3
user14	368	488	532	738	556	237	643	366	597	320	484,5
user15	245	734	388	541	494	240	736	520	313	475	468,6
user16	390	656	509	723	504	785	369	554	196	389	507,5
user17	472	335	585	755	556	785	251	779	253	210	498,1
user18	759	330	908	738	494	557	214	733	409	129	527,1
user19	695	416	980	541	326	785	736	364	476	127	544,6
user20	368	488	532	723	504	589	643	280	569	157	485,3
user21	251	734	388	755	556	470	736	198	597	213	489,8
user22	235	656	509	738	494	557	643	779	313	320	524,4
user23	245	335	585	430	215	785	736	733	196	475	473,5
user24	390	248	908	309	326	589	439	779	132	389	450,9
user25	472	240	980	541	504	352	643	733	569	210	524,4
user26	759	330	532	723	556	470	736	520	597	129	535,2
user27	695	416	380	755	494	557	316	554	476	127	477
user28	368	488	234	738	251	785	372	779	569	475	505,9
user29	251	734	388	430	215	589	439	733	597	475	485,1
user30	235	656	509	311	326	352	643	366	409	389	419,6

Tabelle A.3: Die Verteilungstabelle für die Benutzer und die einzelnen 10 Tage, die durch den Fragebogen simuliert werden sollen.

FRAGEBOGEN ZUR SIMULATION DES ANWENDUNGSFALLS

Nachfolgend ist der verwendete Fragebogen dargestellt. Die Frage-Seite zieht sich mit gleichem Aufbau durch den ganzen Fragenbogen, wodurch sie nur einmal dargestellt wurde.

Fragebogen zum Füllstand der Mensa

Anmeldung

Geben Sie ihren vorher erhaltenen Benutzernamen, sowie das Passwort ein.

Benutzername:

Passwort:

Fragebogen zum Füllstand der Mensa

Fragebogen zum Mensafüllstand

Vielen Dank dafür, dass Sie sich die Zeit nehmen, diesen Fragebogen auszufüllen. Er umfasst 20 Fragen. Falls Sie eine Frage nicht verstehen oder nicht beantworten wollen, drücken Sie auf weiter, um die Frage so zu überspringen. Falls Sie sich bei einer Frage nicht sicher sind, tragen Sie das Ergebnis ein, das Sie für am glaubwürdigsten halten. Es gibt keine richtigen oder falschen Antworten. Sie können also nichts falsch machen.

Die Ergebnisse der Umfrage sollen als Testdaten für einen Algorithmus zur Bewertung von Benutzern und ihren Submission dienen. Die Umfrage findet völlig anonym statt. Es werden keine auf Sie bezogenen Daten gespeichert.

Falls Sie mehr Informationen über den Hintergrund des Fragebogens erfahren wollen, klicken sie [>>hier<<](#).

Beschreibung

Nachfolgend erhalten Sie eine Reihe von Bildern, die den Grundriss der Alten Mensa der TU Dresden zeigen.

Dabei wurden verschiedene Füllstände, die auf den Daten des WLAN-Managements der TU Dresden basieren, grafisch dargestellt.

Ihre Aufgabe wird es sein, die dargestellten Füllstände zu bewerten, indem Sie sich in die Lage eines Besuchers der Alten Mensa versetzen.

Hinweise

Bewerten Sie die folgenden Bilder jeweils mit einem Füllstand: "leer", "halbvoll" oder "voll".

Geben Sie dabei keine anderen Antworten an.

Bewerten Sie nur auf Grundlage der Bilder, welchen Füllstand Sie für richtig empfinden.

Stühle sind als Kreise dargestellt und Tische als Rechtecke.

Rot gefüllte Kreise sind belegte Plätze.

Erklärung

Im Rahmen meiner Bachelorarbeit wurde ein Bewertungsalgorithmus für Benutzer und ihre Submissions entwickelt. Dabei handelt es sich speziell um Submissions, die beim Crowdsourcing erhoben werden. Crowdsourcing bezeichnet die Auslagerung von Teilaufgaben an eine Gruppe freiwilliger Benutzer, z.B. über das Internet.

Wozu Crowdsourcing?

Bei einer Anwendung ist es immer sehr wichtig, dass die Daten, die diese verwendet, stets aktuell sind. Je nach Art der Anwendung und des Datenbestands kostet die Pflege des Datenbestands sehr viel Zeit und ist meist unverhältnismäßig teuer. Crowdsourcing bietet dabei eine günstige, sehr gut funktionale Lösung an.

Anwendungsfall Alte Mensa

Mit Hilfe von Crowdsourcing sollen Daten über den Füllstand der Alten Mensa der TU Dresden erhoben werden. Dabei wird ein expliziter Crowdsourcing-Ansatz verwendet, was nichts anderes bedeutet, als dass der Benutzer direkt nach dem Füllstand gefragt wird.

Aber wozu der Fragebogen?

Der Fragebogen dient dazu, den Anwendungsfall zu simulieren. Auf Grundlage von gemessenen Daten durch das WLAN-Management der TU Dresden wurden verschiedene Füllstände grafisch dargestellt. Ziel ist die Erhebung von Submissions von Benutzern, um diese anschließend zu bewerten.

Zurück zur Bachelorarbeit

Der in der Bachelorarbeit entwickelte Algorithmus zur Bewertung von Benutzern und ihren Submission wird anhand des durch diese Simulation erhaltenen Datenbestands getestet.

Vielen Dank für Ihre Mithilfe,
gez. Tommy Kubica

Zurück

LITERATURVERZEICHNIS

- [Bom12] BOMBACH, Gerd: *Untersuchung von Methoden des expliziten Crowd-Sourcing für Location-Based Services in MapBiquitous*, Technische Universität Dresden, Belegarbeit, 2012
- [EG12] ESTELLÉS-AROLAS, E. ; GONZÁLEZ-LADRÓN-DE-GUEVARA, F.: *Towards an integrated crowdsourcing definition*, In: Journal of Information Science (2012), 02. – in press; DOI: 10.1177/0165551500000000
- [Har12] HARA, Tenshi C.: *Towards a reliable architecture for Crowdsourcing in context of the MapBiquitous project*, Technische Universität Dresden, Diplomarbeit, 2012
- [Mi12] MIßFELDT, Martin: *Video-SEO: YouTube-Ranking-Faktoren (Infografik)*, <http://www.tagseoblog.de/video-seo-youtube-ranking-faktoren-infografik>, 21. Juni 2012 (aufgerufen am 04.08.2013)
- [Ros10] ROSKOS, Matias: *Crowdsourcing und Qualität - Doritos Superbowl-Spot auf dem 2. Platz*, <http://www.socialnetworkstrategien.de/2010/02/crowdsourcing-und-qualitat-doritos-superbowl-spot-auf-dem-2-platz/>, 08. Februar 2010 (aufgerufen am 22.08.2013)
- [Sch08] SCHMIDT, Christoph: *Kritik an neuen eBay-Regeln*, http://www.t-online.de/computer/id_15140348/ebay-starke-kritik-an-neuen-bewertungsregeln.html, 26. Mai 2008 (aufgerufen am 09.08.2013)
- [Sch10] SCHLEGEL, Markus: *Cron Jobs für Einsteiger bildhaft erklärt via Sheldon Cooper*, <http://www.drweb.de/magazin/cron-jobs-fuer-einsteiger-bildhaft-erklart-via-sheldon-cooper/>, 25. Oktober 2010 (aufgerufen am 16.08.2013)
- [Zis11] ZISLER, Andrea: *Das Wissen der Vielen- über Crowdsourcing*, <http://blog.lingoking.com/background/das-wissen-der-vielen>, 07. Oktober 2011 (aufgerufen am 22.08.2013)