

Entwicklungsprojekt  
Interaktive Systeme  
Dokumentation

vorgelegt an der

Technische Hochschule Köln – Fakultät für  
Informatik und Ingenieurwissenschaften

ausgearbeitet von

Leonid Vilents

betreut durch

Prof. Dr. Kristian Fischer

Prof. Dr. Gerhard Hartmann

B. Sc. Sheree Saßmannshausen

B. Sc. Daniela Reschke

## Inhaltsverzeichnis

Tabellenverzeichnis .....	3
Abbildungsverzeichnis .....	4
Einleitung .....	5
Konzept .....	5
Domänenrecherche .....	5
Stakeholder-Identifizierung .....	6
Stakeholder-Analyse .....	6
Marktrecherche .....	7
Kommunikationsmodelle .....	10
Zielhierarchie .....	12
Unique Selling Points .....	21
Risiken .....	21
Proof of Concept .....	22
Methodischer Rahmen .....	23
Dokumentation: Usability Engineering Lifecycle als Vorgehensmodell .....	24
Änderungen am Modell zu Gunsten des Projektes .....	24
Dokumentation: Interaktion mit dem System .....	24
Benutzermodellierung .....	24
Benutzungsmodellierung .....	26
Anforderungsermittlung .....	28
Dokumentation: Architektur .....	29
Diagramm .....	29
Datenstrukturen .....	30
Anwendungslogik .....	32
Dokumentation: User Interface .....	33
Prototyp .....	33
Evaluation .....	41
Quellverzeichnis .....	43
Anhang .....	44
Use Cases .....	44

## Tabellenverzeichnis

Tabelle 1: Stakeholder "Casemodder" .....	6
Tabelle 2: Stakeholder "Sponsor" .....	6
Tabelle 3: Stakeholder-Analyse .....	7
Tabelle 4: User Profile "Sponsor" .....	25
Tabelle 5: User Profile "Casemodder" .....	26
Tabelle 6: USE CASE #3: Dashboard aufrufen .....	26
Tabelle 7: USE CASE #1: Suchstatus setzen .....	27
Tabelle 8: Heuristische Evaluation Mockups .....	42
Tabelle 9: USE CASE #1: Registrierung .....	44
Tabelle 10: USE CASE #2: Anmeldung .....	44
Tabelle 11: USE CASE #4: Benutzerprofil einsehen .....	45
Tabelle 12: USE CASE #5: Benutzerprofil bearbeiten .....	45
Tabelle 13: USE CASE #6: Nachrichten abrufen .....	46
Tabelle 14: USE CASE #7: Bestimmte Nachricht einsehen .....	46
Tabelle 15: USE CASE #8: Neue Nachricht erstellen .....	47
Tabelle 16: USE CASE #9: Nachricht löschen .....	47
Tabelle 17: USE CASE #11: Benutzerkonto schließen .....	48
Tabelle 18: USE CASE #12: Fremdes Profil einsehen .....	48
Tabelle 19: USE CASE #13: Fremdes Element "upvoten" .....	49
Tabelle 20: USE CASE #14: Projekt erstellen .....	49
Tabelle 21: USE CASE #15: Projektupdate anlegen .....	50
Tabelle 22: USE CASE #16: Nach Projekt suchen .....	50
Tabelle 23: USE CASE #17: Fremdes Projekt einsehen .....	51
Tabelle 24: USE CASE #18: Sponsoring-Team erstellen .....	51
Tabelle 25: USE CASE #19: Sponsoring-Team bearbeiten .....	52
Tabelle 26: USE CASE #20: Sponsoring-Team löschen .....	52
Tabelle 27: USE CASE #21: Projekt löschen .....	53

## Abbildungsverzeichnis

Abbildung 1: Sponsoring-Formular von Caseking .....	8
Abbildung 2: Unterforum "Sponsored Builds & Case Mods" von overclock.net .....	9
Abbildung 3: Deskriptives Kommunikationsmodell .....	10
Abbildung 4: Präskriptives Kommunikationsmodell .....	11
Abbildung 5: Architekturdiagramm.....	29
Abbildung 6: Entity Relationship-Diagramm .....	31
Abbildung 7: UI-Prototyp Login .....	33
Abbildung 8: UI-Prototyp Registrierung Casemodder .....	34
Abbildung 9: UI-Prototyp Registrierung Sponsor .....	34
Abbildung 10: UI-Prototyp Dashboard, Casemodder.....	35
Abbildung 11: UI-Prototyp Casemodder-Profil .....	35
Abbildung 12: UI-Prototyp Sponsor-Profil.....	36
Abbildung 13: UI-Prototyp Profil Eingabemaske .....	37
Abbildung 14: UI-Prototyp Rubrik "Projektwelt" .....	37
Abbildung 15: UI-Prototyp Projektansicht .....	38
Abbildung 16: UI-Prototyp Nachrichten .....	38
Abbildung 17: UI-Prototyp einzelne Nachricht .....	39
Abbildung 18: UI-Prototyp Nachricht löschen.....	39
Abbildung 19: UI-Prototyp Sponsor Team-Builder.....	40
Abbildung 20: UI-Prototyp Sponsor Team-Ansicht .....	40
Abbildung 21: UI-Prototyp Sponsor Team Entfernungsdialog .....	41

## Einleitung

Die nachfolgende Arbeit befasst sich mit der Dokumentation des fortgeführten Projektes im Rahmen des Moduls „Entwicklungsprojekt Interaktive Systeme“ an der TH Köln.

Ziel der Dokumentation ist die Aufzeichnung des Entwicklungsprozesses, insbesondere die Ausarbeitung der einzelnen geforderten Artefakte sowie die Ausarbeitung des im Konzept gewählten Vorgehensmodells.

Der Projektplan wird zur Bearbeitung der Dokumentation laufend aktualisiert.

## Konzept

### Domänenrecherche

Im Rahmen des zu erarbeitenden Systems, bewegt man sich im klar abgrenzbaren Bereich von der äußerlichen Modifizierung von PC-Systemen und deren Peripherie, auch im Gemeinschaftsjargon unter dem Anglizismus „Case Modding“ (von engl. *case* = *Gehäuse*, engl. *modification* = *Veränderung*) bekannt.

Hauptaktivität der Domäne ist die äußerliche Modifikation von PC-Systemen und deren Peripherie. Die wohl bekannteste Praktik des Casemodding ist die farbliche Anpassung von Rechnergehäusen, aber auch von Hardwarekomponenten und deren Einzelteile, wie zum Beispiel das Färben des Kühlergehäuses einer Grafikkarte. Andere Modifikationen beinhalten das Verbauen von maßgeschneiderter Beleuchtung (bspw. LED-Lampen), dessen Programmierung, zusätzliche Belüftungs- und Kühlungsmaßnahmen wie ein Wasserkühlsystem, aber auch im großen Stil die komplette Erstellung eines speziellen Gehäuses aus gängigen Baustoffen. Die großen Künstler der Community, die in eben dieser als „Casemodder“ bezeichnet werden, gehen weit über die herkömmliche Benutzung eines PCs hinaus, und ergänzen ihre Systeme entweder mit Kunst, oder mit Funktionalität.

Ein wichtiger Zweig der deutschen Case Modding-Gemeinde ist der regelmäßige Wettbewerb im Rahmen der jährlich stattfindenden Deutschen Casemod Meisterschaft. Diese wird seit 2002 ausgetragen und findet vor allem seit 2009 auf der in Köln stattfindenden Videospielemesse „gamescom“ statt (vgl. Die deutsche Casemod Meisterschaft). In verschiedenen Kategorien werden in diesem Wettbewerb die kreativsten Köpfe gekürt.

Ein guter Casemod kostet Geld, und ein wettbewerbsfähiger Casemod kostet Geld, das die meisten Casemodder nicht haben - eine Möglichkeit sind hierbei Sponsoren, die einen einzelnen Casemodder, oder ein kleines Case Modding-Team, bei der Teilnahme finanziell und materiell unterstützen, um von der eigenen Werbung zu profitieren. Während die Puristen unter den Casemoddern auf die rohe, komplett eigenhändige Variante setzen, bei der man seinen Casemod „from scrap“ aufbaut, sind in heutiger Zeit vor allem sehr viele Case Modding-Fachhandel auf dem Vormarsch, die viele Einzelteile und Komponenten anbieten, und Sponsoren sind.

Die Frage verbleibt aber, wie man schneller von Sponsoren entdeckt werden kann, und wie man die Kommunikation zwischen dem Casemodder und dem Sponsor unterstützt.

Um den Einstieg in die Entwicklung des Systems zu erleichtern und ein Maß an Vertrautheit zu schaffen (vgl. Preim & Dachzelt, 2010), werden in diesem Abschnitt domänenbasierte Metaphern und Paradigmen genannt, welche bei der Entwicklung berücksichtigt werden sollten. Diese Elemente eignen sich auch zur Gewährleistung der Gebrauchstauglichkeit während der Gestaltung und Evaluation der Benutzeroberfläche.

- **Bewerbung** – Repräsentiert den schriftlichen Ausdruck des Interesses an einem Sponsoring-Programm
- **Casting** – Repräsentiert das Suchen von Sponsoren nach geeigneten, talentierten Casemoddern
- **Werbung** – Repräsentiert den Vertragszweck, für den ein Sponsoring-Programm zwischen einem Sponsor und einem Casemodder entsteht.

### Stakeholder-Identifizierung

Um die genauen Wünsche der potentiellen Nutzer einzugrenzen, sollen zunächst alle Stakeholder identifiziert werden. Ihre Rollen werden zunächst in groben Tabellen dargestellt.

*Tabelle 1: Stakeholder "Casemodder"*

Rolle des Stakeholders	Anwender: Casemodder
Beschreibung	Der Casemodder ist die prozentuale Mehrheit der Benutzer. Er hat Interesse daran, seine Kreationen der Öffentlichkeit und einer Auswahl von potentiellen Sponsoren zu präsentieren, die diese bewerten können. Er möchte außerdem die Möglichkeit haben, sein Case Modding-Equipment jederzeit bei einem Fachhändler aufzustocken.
Wissensgebiet	Variierende Kenntnisse im Bereich des Case Moddings; sucht nach Modding-Komponenten, Feedback und Sponsoren
Begründung	Künftiger Anwender des Systems, möchte darüber die Möglichkeit haben, eine beschleunigte Kommunikation mit dem Ziel, einen Sponsoring-Vertrag zu bekommen, zu betreiben.

*Tabelle 2: Stakeholder "Sponsor"*

Rolle des Stakeholders	Anwender: Sponsor
Beschreibung	Als Sponsor wird hier der Vertreter einer Organisation oder Firma bezeichnet, welcher im Auftrag ihrer Organisation wettbewerbsfähige Casemodder finden und anwerben möchte. In den meisten Fällen ist der Sponsor entweder ein Hardwarehersteller oder -vertreiber. Er möchte ein „Portfolio“ potentieller Schützlinge einsehen und diese bei Interesse kontaktieren können.
Wissensgebiet	Professionelle Kenntnisse im Public Relations Marketing, „Talentscout“; versteht von Case Modding als Materie weniger als von der Möglichkeit des Werbeprofits für seine Organisation
Begründung	Künftiger Anwender des Systems, möchte darüber die Möglichkeit haben, eine beschleunigte Kommunikation mit dem Ziel, talentierte Casemodder zu finden, zu betreiben.

### Stakeholder-Analyse

Zu Erkennung von Interessen und Risiken, werden hier die identifizierten Stakeholder in ihrer relevanten Beziehung zum System beschrieben.

Die Norm DIN EN ISO 9241-210 sieht die folgenden Beziehungen als unterscheidungspflichtig vor:

- Anrecht (Rechtsverbindlicher Charakter)
- Anspruch (Teil einer Menge)
- Anteil (Forderung nach Berücksichtigung)
- Interesse (Bedürfnis der Teilhabe)

in Bezug auf einen Objektbereich des Systems – entweder das Gesamtsystem, oder ein Merkmal davon. Ferner müssen die Erfordernis (Ressource und Zweck) und die entsprechende Erwartung dazu beschrieben werden.

Tabelle 3: Stakeholder-Analyse

Stakeholder-Bezeichnung	Beziehung zum System	Objektbereich des Systems	Erfordernis	Erwartung
Casemodder	Anrecht	Gesamtsystem	Gewährleistung der Datensicherheit	Das System darf persönliche Daten nicht für Dritte freigeben
	Anspruch	Gesamtsystem	Nutzung des Gesamtsystems	Das System muss nutzbar und verfügbar sein
				Informationen müssen qualitativ und korrekt sein
				Das System muss seiner Architektur entsprechend lauffähig sein
	Anteil	Merkmal	Anwendungsdaten zur Angabe von persönlichen Daten	Das System soll die Daten einlesen und persistent speichern
			Ermittlung der Gesamtreputation	Das System soll aus den spezifischen Reputationen und deren Gewichtung eine Gesamtreputation errechnen (vgl. Unique Selling Points)
Sponsor	Interesse	Gesamtsystem	Verbesserung der Kommunikation	Das System soll die Kommunikation mit Sponsoren vereinfachen
	Anrecht	Gesamtsystem	Gewährleistung der Datensicherheit	Das System darf persönliche Daten nicht für Dritte freigeben
	Anspruch	Gesamtsystem	Nutzung des Gesamtsystems	Das System muss nutzbar und verfügbar sein
				Informationen müssen qualitativ und korrekt sein
				Das System muss seiner Architektur entsprechend lauffähig sein
	Anteil	Merkmal	Anwendungsdaten zur Angabe von persönlichen Daten	Das System soll die Daten einlesen und persistent speichern
			Benachrichtigung über geänderte Suchstatus	Das System soll den Sponsor über aktivierte Suchen von Casemoddern unterrichten (vgl. Unique Selling Points)
			Vorschläge für Teambildung anhand von Reputationsdurchschnitten	Das System soll dem Sponsor Vorschläge für potentiell talentierte Casemodder-Teamkonstellationen geben (vgl. Unique Selling Points)
	Interesse	Gesamtsystem	Verbesserung der Kommunikation	Das System soll die Kommunikation mit Sponsoren vereinfachen

## Marktrecherche

Wichtig bei der Konzeption eines Interaktiven Systems ist die Suche nach Parallelen und Referenzen. Diese Recherche geschieht auf dem Domänenmarkt, und kennzeichnet vollständige Konkurrenzprodukte oder Teilfunktionalitäten von Systemen, die als Konkurrenz zum zu konzipierenden System zu verstehen sind.

Während bei der Recherche kein eindeutiges Produkt gefunden wurde, welches in etwa den geplanten Merkmalen des zu entwickelnden Systems entspricht, wurden bei einigen, gängigen System Teilfunktionalitäten festgestellt, die an das geplante System erinnern.

### Sponsoring-Formular von Caseking

Abbildung 1: Sponsoring-Formular von Caseking

Der Online-Fachhandel für Case Modding Caseking bietet ein Sponsoring-Programm für aufstrebende Casemodder an. Als Teil der Prozedur, kann man in einem klassischen HTML-Formular, Informationen über sich selbst angeben, und diese zur Einsicht an eine entsprechende Abteilung von Caseking senden. Diese wiederum melden sich je nach Resonanz.

Klarer Vorteil ist hier das Angebot der direkten Kommunikation zwischen Casemodder und Sponsor, die das zu entwickelnde System ebenfalls anstrebt. Eine Suche nach Kontaktdaten und -personen erübrigt sich.



## overclock.net Unterforum "Sponsored Builds & Casemods"

Thread	Last Post	Replies	Views
<b>[Sponsored] Verktøy Cu   An Elegant Steampunk 2P Workstation</b> started by derickwm (2/6/16 at 3:11pm)	Wow just amazing the time, dedication to this rig is beyond... Today at 12:58 am by Charles1	329	29,717
<b>[SPONSORED] Custom Parvum R1.0 &lt;&lt;&lt;&lt;&lt; Kinetic Energy &gt;&gt;&gt;&gt;&gt;</b> started by Mads1 (12/7/15 at 8:57am)	Wandering in late to the party, but subbing for sure! Yesterday at 7:21 pm by ^DeMoN^	222	33,324
<b>[SPONSORED] IKONiK • Case Mod • (1st Official Mod Of The Be Quiet Dark Base 900 Pro)</b> started by ReaperN1 (8/12/16 at 5:38pm)	Painting Acrylic Passthrough Plate: This was a quick paint... Yesterday at 6:55 pm by ReaperN1	264	12,904

Abbildung 2: Unterforum "Sponsored Builds & Case Mods" von overclock.net

Das Unterforum "Sponsored Builds & Casemods" des Case Modding-Forums overclock.net ist eines von vielen Paradebeispielen von bewertungserwartender Zurschaustellung der eigenen Arbeit als Casemodder. In diesem Unterforum bewegen sich meist nur sehr erfahrene Casemodder, die ihre Arbeit visuell und textuell sehr professionell beschreiben, sowie die Schaulustigen, die wie auf einer virtuellen Ausstellung durch die Bilder und Beschreibungen in den Forumsthreads schauen. Auch besteht die extrem hohe Wahrscheinlichkeit, dass auch Talentscouts für mögliches Sponsoring in solche Foren gerne einen Blick werfen, auch, um potenzielle Casemodder abzuwerben.

Was hier vor allem ins Auge fällt, ist die Möglichkeit der Bewertung und Kritik, aber auch des allgemeinen Kommentars durch Zweite. Der Casemodder kann seine Kreationen mit so viel Detail präsentieren, wie er möchte, und kann seine Interessenten auch in einer fortschreitenden Entwicklung durch Updates seines Threads auf dem Laufenden halten.

Diese Funktionalität ist allerdings nur ein Teilprodukt dessen, was das Forum leisten soll, nämlich beitragsbasierte Kommunikation zwischen mehreren Parteien. Einen Thread als Projekttagetagebuch zu führen, erscheint hier als Zweckentfremdung des eigentlichen Systems, und erreicht nicht das optimale Maß an Gebrauchstauglichkeit hinsichtlich des Nutzungskontextes.

Ein weiteres Problem ist hier auch die Obsoleszenz der Suche nach Sponsoren, weil die Mehrheit der ausgestellten Objekte bereits durch Sponsoren gefördert sind – dadurch bleibt nur das Abwerben von bereits unterstützten Casemoddern, statt die Suche nach neuen Talenten.

## Fazit

Der Nachteil von Caseking ist, dass ein solches Angebot spezifisch nur für Caseking selbst gilt. Da Casemodder im Allgemeinen keine wirklichen Präferenzen bezüglich ihres Sponsors haben, ist eine solche Anfrage aufgrund ihrer Einzigartigkeit, mit hoher Wahrscheinlichkeit mit vielen anderen Anfragen verbunden, was zu einer deutlich höheren Wahrscheinlichkeit führt, ausgeschlossen zu werden.

Das Unterform von overclock.net erfährt mit der Nutzung als Sammlung von Projektstagebüchern, eine Zweckentfremdung, bei der viel Gebrauchstauglichkeit für den Nutzungskontext verloren geht. Es müsste eine Möglichkeit geben, sein Projektstagebuch auf einer dedizierten Plattform zu führen, bei der Bewertungen und wertfreie Kommentare klar abgegrenzt sind.

## Kommunikationsmodelle

Damit erste (technische) Vorteile des zu entwickelnden Systems sichtbar werden, werden die Kommunikationen zwischen Casemoddern und Sponsoren im Ist- und im Soll-Zustand betrachtet.

### Deskriptives Modell

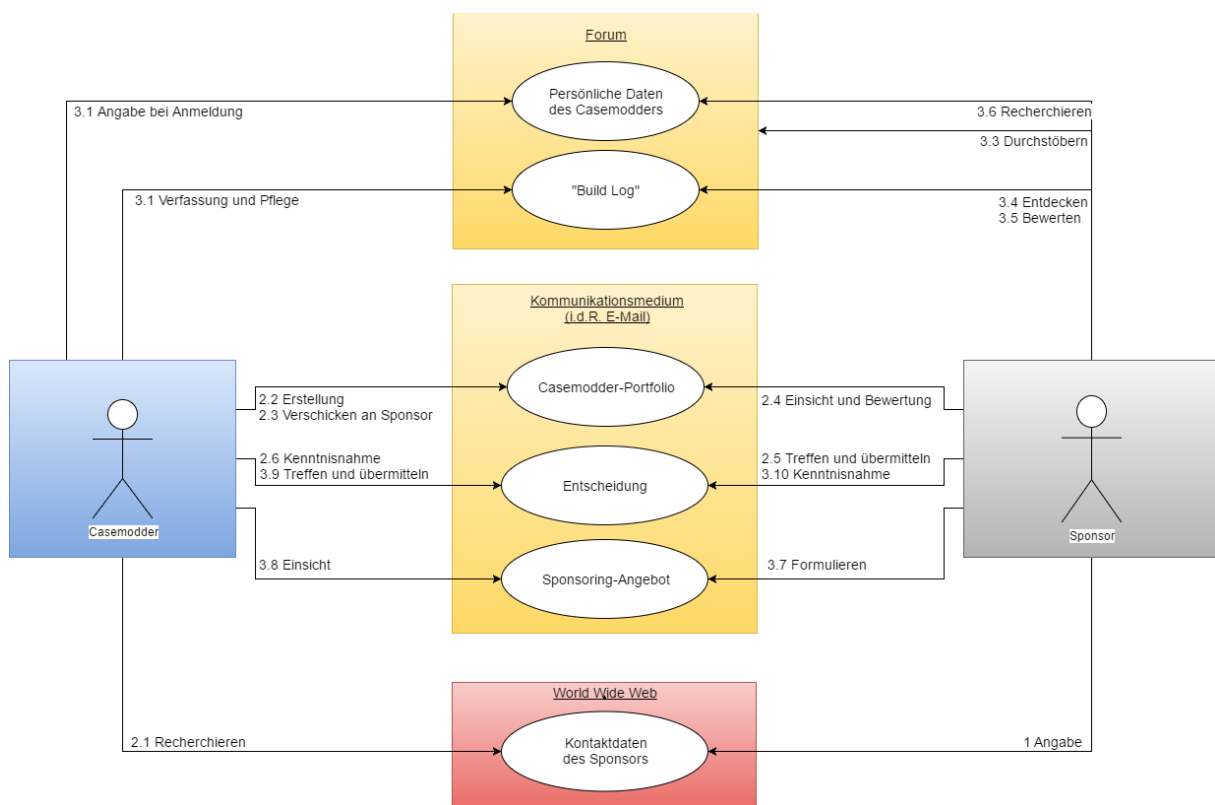


Abbildung 3: Deskriptives Kommunikationsmodell

Im Ist-Zustand ist die Kommunikation über mehrere mögliche Kanäle gestreut, dabei ist vor allem bei beim aktiven Kommunikationsbeginn von Seiten des Casemodders zu sehen, dass der Sponsor mögliche Kontaktdaten in öffentlichen Quellen, wie in diesem Beispiel dem World Wide Web, angeben muss (1), die vom Casemodder in derselben Quelle recherchiert werden können (2.1). Der Casemodder kann dann ein Portfolio von sich und seinen Arbeiten erstellen (2.2) und über ein designiertes Kommunikationsmedium, in der Regel E-Mail, an den Sponsor versenden (2.3). Der Sponsor kann das vom Casemodder erhaltene Portfolio einsehen und bewerten (2.4), und basierend

darauf eine Entscheidung bezüglich eines Sponsorings treffen und an den Casemodder übermitteln (2.5), die bei Versand an den Casemodder, von diesem zur Kenntnis genommen wird (2.6).

Beim aktiven Kommunikationsbeginn des Sponsors, wendet sich dieser oft themenbasierten Foren zu, welche er durchstöbert (3.3). Hier muss allerdings die Kommunikationsbedingung gegeben sein, dass ein interessanter Casemodder, seine persönlichen Daten im besagten Forum als Teil seiner Registrierung angibt (3.1) und bereits durch die Verfassung und Pflege von Inhalten, im besten Fall „Build Logs“ von Casemods (3.2), aufgefallen ist. Der Sponsor kann dann beim Stöbern ein solches Build Log entdecken (3.4) und bewerten (3.5). Zeichnet sich die Arbeit als interessant oder talenthaltig aus, werden die persönliche Daten des Casemodders recherchiert (3.6) und über ein designiertes Kommunikationsmedium ein Sponsoring-Angebot versendet (3.7). Der Casemodder kann dieses einsehen (3.8), und dem Sponsor seine Entscheidung übermitteln (3.9).

Die Kommunikation ist hierbei sichtbar eingeschränkt, da für die Initiierung bestimmte Voraussetzungen erfüllt sein müssen, die nicht zentralisiert einsehbar sind, und auch hier das tatsächliche Interesse an einer Zusammenarbeit, nicht ersichtlich wird, sondern von den Reaktionen der kontaktierten Parteien abhängig ist.

### Präskriptives Modell

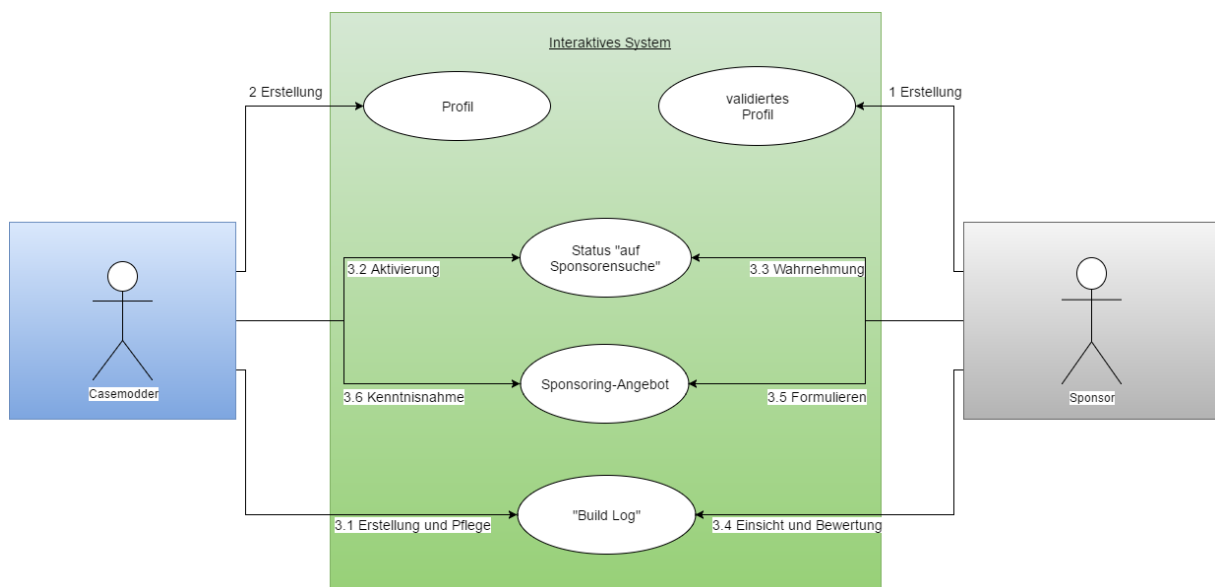


Abbildung 4: Präskriptives Kommunikationsmodell

Im Soll-Zustand gelingt die erforderliche Zentralisierung der Einsicht der aus dem Ist-Zustand benötigten Kommunikationsvoraussetzungen mithilfe des zu entwickelnden Systems. Vor allem werden hier besonders diejenigen Casemodder und Sponsoren zusammengeführt, die ein aktives Interesse an einem Sponsoring-Programm besitzen. Bevor die Kommunikation hier aber beginnen kann, müssen beide Parteien auf dem System Benutzerprofile anlegen (1, 2). Der Sponsor muss hierbei sein Profil auch validieren.

Ein Casemodder kann in seinem Profil Projekte in Form von Build Logs erstellen und pflegen (3.1), und seinen Status „auf Sponsorensuche“ aktivieren (3.2); diese Aktivierung wird für alle Sponsoren im System wahrnehmbar (3.3), und diese können im Profil des Casemodders seine Projekte einsehen und bewerten (3.4). Aus der Bewertung ergibt sich ein potentiellles Sponsoring-Angebot, welches

vom Sponsor formuliert und an den Casemodder gestellt werden kann (3.5), worauf dieser vom Angebot Kenntnis nehmen und sich entscheiden kann (3.6).

Durch die zentralisierte Information und der Zuordnung von gemeinsamen Interessen, entsteht eine verbesserte gerichtete Kommunikation zwischen potentiellen Sponsoren und an Sponsoring interessierten Casemoddern.

## Zielhierarchie

### **Strategisches Ziel 1**

Es muss ein Konzept erstellt werden, welches für den ersten Projektmeilenstein wesentliche Artefakte enthält.

#### **Taktisches Ziel 1.1**

Das Konzept muss die hier aufgelistete Zielhierarchie vollständig enthalten.

##### **Operatives Ziel 1.1.1**

Die Zielhierarchie muss, basierend auf den Aufgaben innerhalb des Projektes, ausgearbeitet werden.

##### **Operatives Ziel 1.1.2**

Die Zielhierarchie muss projektspezifische Ziele enthalten.

##### **Operatives Ziel 1.1.2**

Die Zielhierarchie kann iterativ feingranularer ausgearbeitet werden.

#### **Taktisches Ziel 1.2**

Das Konzept muss eine Domänenrecherche enthalten, die den Nutzungskontext des Case Moddings und die Benutzer des zu entwickelnden Interaktiven Systems beschreibt.

##### **Operatives Ziel 1.2.1**

Es müssen Quellen gefunden werden, welche die Domäne des Case Moddings als werbeorientierter Wettbewerb in Teilen oder als Ganzes beschreiben.

##### **Operatives Ziel 1.2.2**

Die Stakeholder der analysierten Domäne müssen identifiziert werden.

##### **Operatives Ziel 1.2.3**

Der identifizierte Stakeholder „Casemodder“ muss kurz beschrieben werden.

##### **Operatives Ziel 1.2.4**

Der identifizierte Stakeholder „Sponsor“ muss kurz beschrieben werden.

##### **Operatives Ziel 1.2.5**

Es sollen Metaphern gefunden und beschrieben werden, welche Aktivitäten innerhalb des Nutzungskontextes repräsentieren.

#### **Taktisches Ziel 1.3**

Die Projektdokumentation muss die deskriptive und präskriptive Kommunikation schildern.

##### **Operatives Ziel 1.3.1**

Die Kommunikation zwischen Casemoddern und Sponsoren im Ist-Zustand muss analysiert werden.

**Operatives Ziel 1.3.2**

Es muss ein deskriptives Kommunikationsmodell erstellt werden, welches die analysierte Kommunikation zwischen den Stakeholdern im Ist-Zustand darstellt.

**Operatives Ziel 1.3.3**

Basierend auf dem zu entwickelnden System, muss die Kommunikation zwischen den Stakeholdern im Soll-Zustand analysiert werden.

**Operatives Ziel 1.3.4**

Es muss ein präskriptives Kommunikationsmodell erstellt auf Basis der analysierten Kommunikation im Soll-Zustand erstellt werden.

**Operatives Ziel 1.3.5**

Der positive Schritt zwischen dem deskriptiven und präskriptiven Modell muss erkennbar sein.

**Taktisches Ziel 1.4**

Das Konzept soll eine Marktrecherche (Related Works) enthalten, in der mögliche Konkurrenzprodukte oder Systeme mit Teilfunktionalitäten aufgeführt sind, die für Case Modding spezifisch sind.

**Operatives Ziel 1.4.1**

Es müssen mögliche Konkurrenzprodukte ausfindig gemacht und analysiert werden.

**Operatives Ziel 1.4.2**

Jedes gefundene und analysierte Konkurrenzprodukt muss kurz beschrieben werden.

**Operatives Ziel 1.4.3**

Es muss ein Fazit verfasst werden, welches die Nachteile der einzelnen Produkte gegenüber dem Soll-Zustand schildert.

**Taktisches Ziel 1.5**

Das Konzept soll eine Beschreibung von Alleinstellungsmerkmalen (Unique Selling Points) enthalten.

**Operatives Ziel 1.5.1**

Die Alleinstellungsmerkmale des Systems müssen ausgearbeitet werden.

**Operatives Ziel 1.5.2**

Die Alleinstellungsmerkmale des Systems müssen beschrieben werden.

**Taktisches Ziel 1.6**

Das Konzept soll eine Erläuterung möglicher projektspezifischer Risiken enthalten, welche auftreten können. Insbesondere sollte darauf geachtet werden, dass „allgemeine“ Risiken möglichst weniger beachtet werden.

**Operatives Ziel 1.6.1**

Es müssen projektspezifische Risiken gefunden werden.

**Operatives Ziel 1.6.2**

Die gefundenen Risiken müssen beschrieben werden.

**Taktisches Ziel 1.7**

Das Konzept muss eine Beschreibung der Durchführungsmöglichkeit von Anwendungslogik in den Systemkomponenten der Alleinstellungsmerkmale (Proof of Concept) besitzen.

**Operatives Ziel 1.7.1**

Es müssen Machbarkeitsstudien zu den Alleinstellungsmerkmalen gemacht geführt werden.

**Operatives Ziel 1.7.2**

Es müssen Exit- und Fail-Kriterien zu allen Alleinstellungsmerkmalen beschrieben werden.

**Operatives Ziel 1.7.3**

Es müssen konkrete Fallback-Kriterien beschrieben werden.

**Operatives Ziel 1.7.4**

Die Ergebnisse der Machbarkeitsstudien müssen messbar sein.

**Operatives Ziel 1.7.5**

Die Machbarkeitsstudien können von den Risiken abgeleitet werden.

**Operatives Ziel 1.7.6**

Die Machbarkeitsstudien können auch nicht-technischer Natur sein.

**Taktisches Ziel 1.8**

Das Konzept muss den Methodischen Rahmen, in dem das Interaktive System entwickelt wird, beschreiben und begründen.

**Operatives Ziel 1.8.1**

Es muss begründet werden, für welche allgemeine Haltung gegenüber der Entwicklung sich entschieden wird.

**Operatives Ziel 1.8.2**

Es muss begründet werden, für welches aus ehemaligen Module bekannte Vorgehensmodell sich entschieden wird.

**Taktisches Ziel 1.9**

Das Konzept muss ein Quellenverzeichnis enthalten.

**Operatives Ziel 1.9.1**

Alle besuchten und genutzten Quellen müssen dokumentiert werden.

**Operatives Ziel 1.9.2**

Alle dokumentierten Quellen müssen im Quellenverzeichnis angegeben sein.

**Taktisches Ziel 1.10**

Das Konzept muss die Anforderungen für wissenschaftliches Arbeiten erfüllen.

**Operatives Ziel 1.10.1**

Das Konzept muss auf Rechtschreib- und Grammatikfehler überprüft werden.

**Operatives Ziel 1.10.2**

Das Konzept muss ein Inhalts-, Bild- und Tabellenverzeichnis enthalten.

**Operatives Ziel 1.10.3**

Das Konzept muss eine angemessene Zitierweise und gültige Quellenangaben besitzen.

**Operatives Ziel 1.10.4**

Tabellen und Bilder müssen im Konzept entsprechend der Anforderungen beschriftet werden.

**Operatives Ziel 1.10.5**

Das Konzept soll eine angemessene Gliederung und eine gut lesbare Struktur aufweisen.

**Strategisches Ziel 2**

Es muss ein Projektplan erstellt werden, welcher die Aktivitäten im Projekt beschreibt, und den geplanten sowie tatsächlichen Arbeitsaufwand in Stunden festhält.

**Taktisches Ziel 2.1**

Der Projektplan muss bis zum ersten Meilenstein eine abgabefähige Version besitzen.

**Operatives Ziel 2.1.1**

Es müssen alle projektrelevanten Aktivitäten ausgearbeitet werden, welche im Zeitraum des Projektes bewältigt werden sollen.

**Operatives Ziel 2.1.2**

Alle ausgearbeiteten Aktivitäten müssen eine Aufwandschätzung erfahren.

**Operatives Ziel 2.1.3**

Der Projektplan soll zeigen, dass die geplanten Aktivitäten mit dem gewählten Methodischen Rahmen aus dem Konzept konform sind.

**Operatives Ziel 2.1.4**

Der geschätzte Aufwand soll in geplanten Stunden im Projektplan festgehalten werden.

**Operatives Ziel 2.1.5**

Alle bisherigen Aktivitäten sowie deren tatsächlicher Aufwand müssen im Projektplan festgehalten werden.

**Taktisches Ziel 2.2**

Der Projektplan soll im weiteren Verlauf des Projektes iterativ aktualisiert werden.

**Operatives Ziel 2.2.1**

Der tatsächliche Aufwand erledigter Aktivitäten muss im Projektplan festgehalten werden.

**Operatives Ziel 2.2.2**

Neue Aktivitäten oder Änderungen bestehender Aktivitäten sollen im Projektplan festgehalten werden.

**Strategisches Ziel 3**

Es muss zum ersten Projektmeilenstein ein Rapid Prototype erstellt werden, welcher eines oder mehrere Alleinstellungsmerkmale des zu entwickelnden Systems präsentiert.

**Taktisches Ziel 3.1**

Der Rapid Prototype muss zum Abgabetermin eine funktionsfähige Version besitzen.

**Operatives Ziel 3.1.1**

Es muss die wesentliche Struktur des Prototyps programmiert werden.

**Operatives Ziel 3.1.2**

Es muss ein Alleinstellungsmerkmal des Systems programmiert werden.

**Operatives Ziel 3.1.3**

Es sollen Dummy-Daten für das System zu Demonstrationszwecken erstellt werden.

**Operatives Ziel 3.1.4**

Es können alle Alleinstellungsmerkmale des Systems programmiert werden.

**Taktisches Ziel 3.2**

Die Installation des Rapid Prototype auf fremden Systemen muss zu Testzwecken einwandfrei laufen.

**Operatives Ziel 3.2.1**

Es muss auf mindestens einem Fremdsystem die Funktionsfähigkeit des Rapid Prototype nachgewiesen werden.

**Operatives Ziel 3.2.2**

Es sollen auf weiteren Fremdsystemen die Funktionsfähigkeit des Rapid Prototype nachgewiesen werden.

**Taktisches Ziel 3.3**

Der Rapid Prototype soll die Anforderungen an professionelle Programmierung erfüllen können.

**Operatives Ziel 3.3.1**

Alle Funktionen und Merkmale des Systems müssen ausreichend dokumentiert werden.

**Operatives Ziel 3.3.2**

Es muss eine Installationsdokumentation vorliegen, die die Installation durch Zweitpersonen möglich macht.

**Operatives Ziel 3.3.3**

Es können Unit-Tests für die Funktionen und Merkmale des Systems geschrieben werden, um deren Funktionalität prüfen zu können.

**Strategisches Ziel 4**

Es muss zum zweiten Projektmeilenstein eine Projektdokumentation erstellt werden, welche die wesentlichen angeforderten Artefakte enthält.

**Taktisches Ziel 4.1**

Die Projektdokumentation muss Anforderungen an das zu entwickelnde System beschreiben.

**Operatives Ziel 4.1.1**

Die Anforderungen an das System müssen analysiert werden.

**Operatives Ziel 4.2.1**

Die analysierten Anforderungen müssen unterschieden werden zwischen funktionalen, qualitativen und organisationalen Anforderungen.

**Operatives Ziel 4.2.3**

Die gruppierten Anforderungen müssen beschrieben werden.



#### **Operatives Ziel 4.1.4**

Die Anforderungen können iterativ geändert oder ergänzt werden.

#### **Taktisches Ziel 4.2**

Die Projektdokumentation muss eine Beschreibung der möglichen Arten von Benutzern und deren Aufgaben beinhalten.

##### **Operatives Ziel 4.2.1**

Die Benutzermodelle müssen abhängig vom gewählten Vorgehensmodell erstellt werden.

##### **Operatives Ziel 4.2.2**

Die erstellten Benutzermodelle müssen in der Projektdokumentation erläutert werden.

##### **Operatives Ziel 4.2.3**

Die erstellten Benutzermodelle können teilweise in den Anhang versetzt werden, um die Anzahl der Wörter nicht unnötig zu belasten.

#### **Taktisches Ziel 4.3**

Die Projektdokumentation muss die möglichen Aufgaben der verschiedenen Benutzer beschreiben und erläutern.

##### **Operatives Ziel 4.3.1**

Es müssen alle Aufgaben für alle Benutzergruppen ausgearbeitet werden.

##### **Operatives Ziel 4.3.2**

Die deskriptiven Modelle müssen als Grundlage für die präskriptiven Modelle der Aufgaben dienen.

##### **Operatives Ziel 4.3.3**

Sowohl die deskriptiven als auch die hergeleiteten präskriptiven Aufgabenmodelle müssen beschrieben werden.

#### **Taktisches Ziel 4.4**

Die Projektdokumentation muss die Architektur des zu entwickelnden Systems schildern und begründen.

##### **Operatives Ziel 4.4.1**

Die Architektur des zu entwickelnden Systems muss ausgearbeitet werden.

##### **Operatives Ziel 4.4.2**

Es müssen Entscheidungen über die Kommunikationsprinzipien und -protokolle getroffen werden.

##### **Operatives Ziel 4.4.3**

Die Verteiltheit der Anwendungslogik muss nachvollziehbar sein.

##### **Operatives Ziel 4.4.4**

Basierend auf der ausgearbeiteten Architektur, muss ein Architekturdiagramm erstellt werden.

##### **Operatives Ziel 4.4.5**

Alle Entscheidungen hinsichtlich der Architektur müssen begründet oder abgewägt sein.

#### **Taktisches Ziel 4.5**

Die Projektdokumentation muss die im System verwendeten Datenstrukturen beschreiben.

##### **Operatives Ziel 4.5.1**

Es müssen zu verwendende Datenstrukturen ermittelt werden.

##### **Operatives Ziel 4.5.2**

Die ermittelten Datenstrukturen müssen angemessen in der Dokumentation beschrieben und repräsentiert werden.

##### **Operatives Ziel 4.5.3**

Die Datenstruktur kann anschließen durch Objektspezifikationen oder ER-Diagramme aufgezeigt werden.

#### **Taktisches Ziel 4.6**

Die Projektdokumentation muss die Merkmale der Systemarchitektur beschreiben.

##### **Operatives Ziel 4.6.1**

Es muss eine angemessene Darstellung für die Anwendungslogik von Systemkomponenten gewählt werden (bspw. Prosa, Pseudocode, Diagramme...).

##### **Operatives Ziel 4.6.2**

Die gewählte Darstellung muss in der Projektdokumentation für alle Systemkomponenten aufgeführt werden.

##### **Operatives Ziel 4.6.3**

Wesentliche Architekturmerkmale müssen zusätzlich im Rahmen des Vorgehensmodells modelliert werden (bspw. REST-Tabellen).

#### **Taktisches Ziel 4.7**

Die Projektdokumentation muss einen Prototyp der Benutzerschnittstelle enthalten.

##### **Operatives Ziel 4.7.1**

Es soll ein Mockup konzipiert werden, welches die Benutzerschnittstelle des Systems darstellt.

##### **Operatives Ziel 4.7.2**

Die Gestaltungsentscheidungen müssen eine Begründung haben.

##### **Operatives Ziel 4.7.3**

Das Mockup soll in die Dokumentation integriert und erläutert werden.

#### **Taktisches Ziel 4.8**

Die Projektdokumentation muss eine Evaluation des Benutzerschnittstellen-Prototyps beinhalten.

##### **Operatives Ziel 4.8.1**

Als Grundlage muss der Benutzerschnittstellen-Prototyp bereits vorliegen.

##### **Operatives Ziel 4.8.2**

Die Evaluation muss alle Erkenntnisse zielgerichteter Evaluationen eines Aspektes oder Aspektgruppen eines Systems beinhalten.

#### **Taktisches Ziel 4.9**

Die Projektdokumentation muss die Anforderungen für das Projekt erfüllen.

**Operatives Ziel 4.9.1**

Die Projektdokumentation muss auf Rechtschreib- und Grammatikfehler überprüft werden.

**Operatives Ziel 4.9.2**

Die Projektdokumentation muss ein Inhalts-, Bild- und Tabellenverzeichnis enthalten.

**Operatives Ziel 4.9.3**

Die Projektdokumentation muss eine angemessene Zitierweise und gültige Quellenangaben besitzen.

**Operatives Ziel 4.9.4**

Tabellen und Bilder müssen in der Projektdokumentation entsprechend der Anforderungen beschriftet werden.

**Operatives Ziel 4.9.5**

Die Projektdokumentation soll eine angemessene Gliederung und eine gut lesbare Struktur aufweisen.

**Operatives Ziel 4.9.6**

Die Projektdokumentation darf ohne Anhänge nicht länger als 8.000 Worte lang sein.

**Strategisches Ziel 5**

Es muss zum dritten Projektmeilenstein eine Implementierung des konzipierten und dokumentierten Systems erstellt werden.

**Taktisches Ziel 5.1**

Die Implementierung muss zum Abgabetermin eine funktionsfähige Version besitzen.

**Operatives Ziel 5.1.1**

Alle Merkmale der Implementierung müssen zum Abgabetermin voll funktionsfähig sein.

**Operatives Ziel 5.1.2**

Es muss die wesentliche Struktur des Systems programmiert werden.

**Operatives Ziel 5.1.3**

Es müssen alle Alleinstellungsmerkmale des Systems programmiert werden.

**Operatives Ziel 5.1.4**

Es müssen alle dokumentierten Datenstrukturen implementiert werden.

**Operatives Ziel 5.1.5**

Es müssen alle dokumentierten Systemkomponenten implementiert werden.

**Operatives Ziel 5.1.6**

Es muss die konzipierte Benutzerschnittstelle implementiert werden.

**Operatives Ziel 5.1.7**

Es sollen Dummy-Daten für das System zu Demonstrationszwecken erstellt werden.

**Taktisches Ziel 5.2**

Die Implementierung muss ein verteiltes System aus einem Client und einem Server bestehen.

**Operatives Ziel 5.2.1**

Es muss eine Client-Anwendung programmiert werden.

**Operatives Ziel 5.2.2**

Es muss eine Server-Anwendung programmiert werden.

**Operatives Ziel 5.2.3**

Client- und Server-Anwendung müssen miteinander kommunizieren können.

**Taktisches Ziel 5.3**

Die Installation des Systems auf fremden Systemen muss zu Testzwecken einwandfrei laufen.

**Operatives Ziel 5.3.1**

Es muss auf mindestens einem Fremdsystem die Funktionsfähigkeit des Systems nachgewiesen werden.

**Operatives Ziel 5.3.2**

Es sollen auf weiteren Fremdsystemen die Funktionsfähigkeit des Systems nachgewiesen werden.

**Taktisches Ziel 5.4**

Die Implementierung soll die Anforderungen an professionelle Programmierung erfüllen können.

**Operatives Ziel 5.4.1**

Alle Funktionen und Merkmale des Systems müssen ausreichend dokumentiert werden.

**Operatives Ziel 5.4.2**

Es muss eine Installationsdokumentation vorliegen, die die Installation durch Zweitpersonen möglich macht.

**Operatives Ziel 5.4.3**

Es sollen Unit-Tests für die Funktionen und Merkmale des Systems geschrieben werden, um deren Funktionalität prüfen zu können.

**Strategisches Ziel 6**

Es müssen zum dritten Projektmeilenstein eine Diskussion des Zielerreichungsgrades (Fazit) sowie eine kritische Reflexion des Projektes bezüglich der Einhaltung des Projektplans / Methodischen Rahmens (Prozessassessment) verfasst werden.

**Taktisches Ziel 6.1**

Es muss ein Fazit verfasst werden.

**Operatives Ziel 6.1.1**

Der Zielerreichungsgrad des Projektes soll schriftlich diskutiert werden.

**Operatives Ziel 6.1.2**

Das Fazit soll ergebnisorientiert formuliert sein.

**Operatives Ziel 6.1.3**

Es kann ein Ausblick auf weitere Entwicklung des Systems beschrieben werden.

**Taktisches Ziel 6.2**

Es muss ein Prozessassessment verfasst werden.

**Operatives Ziel 6.2.1**

Das Projekt bezüglich der Einhaltung des Projektplans muss kritisch reflektiert werden.

**Operatives Ziel 6.2.2**

Der Umgang mit Schwierigkeiten und / oder Herausforderungen muss beschrieben werden.

**Operatives Ziel 6.2.3**

Das Prozessassessment muss prozessorientiert formuliert werden.

**Unique Selling Points**

Anhand der Marktrecherche entstehen mehrere Grundlagen für die Betonung der Einzigartigkeit und Innovation des zu entwickelnden Systems durch Alleinstellungsmerkmale innerhalb der Domäne (engl. Unique Selling Points) – im folgenden USPs genannt – von denen ein USP auch im Rahmen des Rapid Prototypings realisiert wird.

**Status „auf Sponsorensuche“**

Casemodder, welche das System benutzen, sollen die Möglichkeit erhalten, ihrer Anfrage nach einem Sponsor Gehör zu verschaffen. Dies kann erfolgen, indem sie systemweit zeigen können, dass sie auf der Suche nach einem Sponsor sind – die Sponsoren, die das System benutzen, sollen über solche „globalen“ Anfragen informiert werden, um so ein effizientes Talentscouting vorzunehmen, bei dem beide Parteien den selben Enthusiasmus teilen.

**Aktives Reputationssystem für Casemodder und ihre Kreationen**

Ein Casemodder erhält zu Beginn seiner Profilerstellung einen Reputationszähler, welcher durch Beiträge, Kommentare und Projekte erhöht werden kann. Die Gesamtreputation wird zusammengerechnet aus den Reputationen der einzelnen Unterasspekte. Casemodder mit einer hohen Reputation im Bereich der Projekte werden bei Sponsoren, die auf Talentsuche sind, schneller gezeigt.

Die einzelnen Objekte, die mit Reputationspunkten versehen werden können, haben eine unterschiedliche Gewichtung, welche bei der Berechnung der Gesamtreputation relevant wird. Im Klartext bedeutet dies, dass ein Reputationspunkt für einen konstruktiven Kommentar weniger zur Gesamtreputation des Casemodders beiträgt, als ein Reputationspunkt für ein bewertetes Build Log.

**Erstellungstool für Case Modding-Teams von Sponsoren**

Ein Sponsor erhält die Möglichkeit, neben einzelnen Casemoddern, auch mögliche Propositionen für Teams zu erhalten. Dabei kann er bei der Erstellung eines Teams Vorschläge erhalten, welches theoretische Team im Durchschnitt die höchste Reputation in allen Gesamtpunkten besitzen.

**Risiken****Missbrauch der Sponsorensuche**

Die Suche nach Sponsoren sollte denjenigen Casemoddern obliegen, die auch die Beweislast tragen können, dass ihre Fähigkeiten des Sponsorings würdig sind. Ein Risiko besteht jedoch darin, dass grundsätzlich jeder Casemodder nach den bisherigen Entwürfen die Möglichkeit hat, seinen Status zu setzen – dieses Risiko ist kritisch, da es für einen verfälschten Informationsfluss sorgt. Es muss eine Möglichkeit gefunden werden, den Status nur unter bestimmten Voraussetzungen setzen zu dürfen.

**Akkreditierungsbetrug von Sponsoren**

Die absichtliche Angabe falscher Daten ist im Allgemeinen ein bekanntes Risiko, allerdings ist die Spezialisierung hier eher auf die Art der Stakeholder an sich bezogen. Vor allem falsche Fachhändler, können mit Scheinidentitäten Casemodder und Sponsoren gleichermaßen um Geld für Verkäufe von nichtexistenter Hardware betrügen.

### **Mobverhalten bei der Reputationssteigerung und „gekaufte Punkte“**

Vor allem bekannt aus den aktuellen Marketingparadigmen von YouTube, kann es passieren, dass Casemodder sich mit Geldmitteln Reputation „erkaufen“ – wo die Bewertung sonst aus der persönlichen Motivation heraus passiert, werden stattdessen finanzielle Geschenke im Austausch für Reputationssteigerung akzeptiert. Ein solches Verhalten lässt sich nur bedingt verhindern, da solche Transaktionen meist außerhalb der dafür vorgesehenen Plattformen passieren.

### **Proof of Concept**

#### **Status „auf Sponsorensuche“**

Casemodder sollen in der Lage sein, einen Status zu aktivieren, der sie als „auf Sponsorensuche“ ausweist. Diese Statusaktivierung soll nahezu in Echtzeit systemweit für alle Sponsoren sichtbar sein.

Exit: Der zeitliche Abstand zwischen Casemodder-Statusaktivierung und Sichtbarkeit bei Sponsoren beträgt nicht mehr als 1 Sekunde.

Fail 1: Der Casemodder kann seinen Status nicht aktivieren.

Fail 2: Die Sponsoren nehmen die Statusaktivierung des Casemodders nicht wahr.

Fallback: Der zeitliche Abstand zwischen Casemodder-Statusaktivierung und Sichtbarkeit bei Sponsoren beträgt mehr als 1 Sekunde, jedoch nicht mehr als 3 Minuten.

#### **Reputationssystem für Casemodder und ihre Kreationen**

Die Reputationszähler für die Objekte eines Casemodders müssen in einer Datenbank gespeichert und bei Änderung aktualisiert werden. Die serverseitige Anwendungslogik kann die Gesamtreputation aus den Reputationswerten der einzelnen Objekte, die der Casemodder referenziert, unter Zunahme der Gewichtung der einzelnen Objekte, errechnen.

Exit: Die Gesamtreputation des Casemodders errechnet sich aus den Einzelreputationen seiner Objekte.

Fail 1: Die Gesamtreputation lässt sich nicht berechnen.

Fail 2: Die Gesamtreputation liefert nicht das zu erwartende Rechnungsergebnis.

Fallback: Die Gesamtreputation kann nach jeder Aktualisierung von referenzierten Objekten neu berechnet und als feste Variable abgespeichert werden.

#### **Team-Erstellungstool für Case Modding-Teams für Sponsoren**

Sponsoren können bei der Erstellung von Teams Vorschläge erhalten, welche Teamkonstellation sich anhand der sponsorensuchenden Casemodder und deren spezifischer Reputation am besten eignet.

Exit: Aus X sponsorensuchenden Casemoddern, soll ein Sponsor  $Y < X$  Casemodder gezeigt bekommen, deren durchschnittliche spezifische Reputation Reputation am Höchsten ist.

Fail: Der Sponsor bekommt keine Teamvorschläge.

**Fallback:** Der Sponsor bekommt einen Vorschlag für ein Team, das nicht die höchste durchschnittliche spezifische Reputation besitzt. Eine Alternative würde hier die durchschnittliche Gesamtreputation der einzelnen Casemodder bieten.

## Methodischer Rahmen

Für die effiziente Entwicklung des Systems wird ein methodischer Rahmen spezifiziert, in dem die Designprinzipien, das Vorgehensmodell sowie deren Auswahlbegründungen geschildert werden.

### Entscheidung Design-Prinzipien

Das zu entwickelnde Interaktive System hat ein sehr weit gestreutes Spektrum von Stakeholdern mit unterschiedlichen Fähigkeiten und Eigenschaften. Den Schwerpunkt auf den Verwendungszweck des Systems zu setzen, hätte zur Folge, dass das maximale Maß an Gebrauchstauglichkeit für alle Stakeholder nicht gewährleistet werden kann. Daher ist es sinnvoll, nach Design-Prinzipien zu entwickeln, welche die Merkmale der Benutzer als Ausgangspunkt für Konzeption und Entwicklung betrachten. Das „User-centered Design“ stellt den Benutzer des Systems in den Mittelpunkt der Entwicklung, und erwirbt das Verständnis für dessen Aufgaben, Ziele, Nutzungskontext und Wissen, um es entsprechend anzuwenden (vgl. Norman & Draper, 1986). Diese Eigenschaften machen das User-centered Design zur besten Wahl für das Projekt.

### Entscheidung Vorgehensmodell

Der erste Kandidat ist das „Discount Usability Engineering“-Modell nach Jakob Nielsen, und gilt als kostengünstigstes und zeitschonendstes Vorgehensmodell zur Erreichung eines zufriedenstellenden Grads an Usability. Zu den Aktivitäten des Modells gehören papierbasierte Prototypen als Szenarien, die „think-aloud“-Methode und die Anwendung der zehn Heuristiken nach Nielsen (vgl. Nielsen, 1993).

Eine Anforderungsanalyse ist im Rahmen des Modells nicht erforderlich, und die Ergebnisse der heuristischen Analyse erweisen sich als zu subjektiv in Bezug auf den Analysierenden. Letzterer Umstand, erschwert die Erstellung einer guten Basis zur Berücksichtigung der Eigenschaften der Stakeholder. Auch die zehn Heuristiken erfordern eine kritische Betrachtung, da bei deren Anwendung die gute Gebrauchstauglichkeit der Heuristiken selbst, auf Kosten eines guten Ergebnisses in den Vordergrund treten. Dies macht das Vorgehensmodell unattraktiv für das Projekt.

Kandidat Nummer 2 ist der Prozess des „Scenario-based Design“ nach Mary Beth Rosson und John Carrol, welcher als zentrale Elemente des Entwicklungsprozesses, narrativ beschriebene Szenarien über die Nutzung des Systems aus Benutzersicht, sowie die Identifizierung derer wesentlichen Aspekte, die Auswirkungen auf die User Experience im Nutzungskontext haben, durch die Claims Analysis. Die Szenarien unterstützen die kontinuierliche Beurteilung und Elaboration der Nützlichkeit, Bedienungssimplizität und Benutzerzufriedenheit eines Systems (vgl. Rosson & Carrol, 2003). Nachteil des Vorgehensmodells ist, dass aus allen Szenarien, die verfasst werden, fortlaufend Prototypen entwickelt werden:

*“In SBD, prototypes are built in parallel with writing scenarios. A scenario describes how features of a system contribute to the user’s experience; a prototype is a preliminary implementation of a system or its features.”*

*(Rosson & Carrol, 2002)*

Die kontinuierliche Ausarbeitung von Prototypen zu Gunsten des Modells auf Kosten der projektrelevanten Artefakte zu priorisieren, wird als kritisch bewertet, weswegen die Anwendung des Modells nicht in Betracht gezogen wird.

Zuletzt steht zur Wahl das „Usability Engineering Lifecycle“ nach Deborah Mayhew. Dabei wird die Gebrauchstauglichkeit anhand drei iterativer Merkmale (Anforderungsanalyse, Design/Testen/Entwickeln von Modellen und Prototypen, Installation und Rückmeldung durch tatsächliche Benutzer) hergestellt (vgl. Mayhew, 1999).

Das Vorgehensmodell ist auf die generelle Gliederung der Projektvorgaben, also die allgemeinen Projektmeilensteine, gut abgestimmt, und erlaubt die nahezu einheitliche Zuordnung der einzelnen Phasen des Modells auf die Projektmeilensteine selbst – Ausnahme bildet hierbei die Benutzermodellierung, die im Rahmen der Projektdokumentation, anstatt des Konzeptes, erfolgt.

Änderungen am Vorgehen, die abseits des Vorgehensmodells liegen, werden in der Projektdokumentation beschrieben und begründet. Da der zeitliche Rahmen vergleichsweise sehr gering ist, muss darauf geachtet werden, dass das Vorgehensmodell die Erreichung von möglichst hoher Gebrauchstauglichkeit zwar unterstützt, aber dabei möglichst zeiteffizient sein muss.

## Dokumentation: Usability Engineering Lifecycle als Vorgehensmodell

### Änderungen am Modell zu Gunsten des Projektes

Eines der Hauptziele im Konzept war die Umsetzung der Entwicklung des Interaktiven Systems nach dem von Deborah Mayhew entworfenem „Usability Engineering Lifecycle“. Der Umfang des Projektes sowie die Anforderung von relevanten Artefakten zu bestimmten Meilensteinen kollidiert allerdings inhaltstechnisch mit dem Umfang des Vorgehensmodells von Mayhew, weswegen die generelle Idee der Befolgung des Modells verworfen wird.

Wichtig ist jetzt zu entscheiden, ob bestimmte Vorgänge im Modell weitergeführt, ersetzt oder ganz entfernt werden:

- Die User Profiles für die Benutzermodellierung werden nach dem klassischen Mayhew-Modell erstellt, als vorhergehende Referenzdaten werden die im Anhang befindliche Stakeholder-Identifikation sowie -analyse genutzt.
- Im Rahmen der Benutzungsmodellierung, werden die nach Cockburn modellierten Use Cases verwendet.
- Die jeweiligen Styleguides für die Requirements Analysis sowie die 3 Ebenen der Testing, Design, Development-Phase werden nicht erstellt.
- Im Rahmen der Conceptual Models, werden das Architekturdiagramm, die Architekturmerkmale und die Datenstrukturen angelegt.
- Die Ebene „Screen Design Standards“ wird nicht durchgeführt.
- Im Rahmen der „Detailed User Interface Design“-Ebene, wird der UI-Prototyp der Anwendung sowie seine Evaluation durchgeführt.
- Die Verifizierung der Erfüllung der Usability Goals wird nicht durchgeführt.

## Dokumentation: Interaktion mit dem System

### Benutzermodellierung

Für die konkrete Analyse der Stakeholder wird im Mayhew-Modell die Erstellung von User Profiles benötigt; diese werden durch eine Umfrage von potenziellen Anwendern und die Auswertung der Umfragedaten realisiert.

Im Rahmen des Konzeptes wurden bereits eine Identifikation und eine Analyse der Stakeholder „Casemodder“ und „Sponsor“ durchgeführt (siehe Anhang).



Der erste Schritt zur Erstellung von User Profiles ist nun die Identifikation und Analyse von Fragen, die man den potenziellen Anwendern stellen kann, sowie das Spektrum an Antwortmöglichkeiten, welche zur Auswertung angegeben werden können.

Im Rahmen dieser Identifikation wurden über die Plattform Google zwei Umfrageformulare erstellt, welche für User Profiles relevante Informationen der beiden Stakeholder im Detail erfassen sollen.

Kontaktierte Sponsoren, welche mit dem Fragebogen betraut wurden, haben leider nach längerer Zeit keine Antwort geschickt, weswegen zum Zeitpunkt der endgültigen Verfassung der User Profiles keine erfassbaren Daten zur Auswertung vorgelegen haben. Das User Profile für den Sponsor muss daher mit den rudimentären Daten erstellt werden (vgl. Tabelle 4).

*Tabelle 4: User Profile "Sponsor"*

<b>User Profile: Stakeholder „Sponsor“</b>	
Demografische Charakteristika	Variierend im Alter, Geschlecht, Wohnort, und sozial-ökonomischem Status
Berufserfahrung	Hauptsächlich Berufserfahrung in der IT-Branche, teilweise IT-Marketing
Computerkenntnisse und -Erfahrung	Fortgeschrittene Kenntnisse der Computernutzung
Fachwissen	Umgang mit Computerhardware, Kenntnisse im IT-Marketing
Informationen zum Unternehmen	Firma oder Organisation aus der IT-Branche, Hardware- oder Peripheriehersteller, Vertrieb von Hardware oder Peripherie; Marketingabteilungen besagter Firmen oder Organisationen
Aufgabe	Primär: Anreicherung von Publicity und Interesse für den eigenen Konzern mithilfe von Marketing Sekundär: Rekrutierung und Management von Casemodern im Rahmen von Wettbewerben verschiedener Größen und Arten
Spezielle Produkterfahrung	Erfahrung mit Talent-Scouting und Evaluation von Casemodder-Material
Motivation	Effizientere Rekrutierung von Casemodern, die sich positiv auf das Marketing der Firma auswirken.
Auswirkungen von Fehlern / Versehen	Auswirkungen nur innerhalb der Domäne, nicht im System

Die Umfrage innerhalb der Casemodder-Gemeinde erfolgte innerhalb von spezialisierten Interessentengruppen auf der Social Media-Plattform Facebook. Insgesamt wurde das Formular für die Casemodder, 53-mal beantwortet, woraus sich detailliertere Daten für die Erstellung des User Profiles ableiten lassen.

*Tabelle 5: User Profile "Casemodder"*

<b>User Profile „Casemodder“</b>	
Demografische Charakteristika	Variierend im Alter, Geschlecht, Wohnort, und sozial-ökonomischem Status
Berufserfahrung	Variierende Berufserfahrung, teilweise Erfahrung im Case Modding
Computerkenntnisse und -Erfahrung	Fortgeschrittene Kenntnisse im Umgang mit Hardware, kreative Konstruktion
Fachwissen	Wissen über Hardware, Konstruktion und Zurschaustellung erfolgreicher Projekte
Aufgabe	Primär: Kritische Reflexion der eigenen Projekte durch Zweite Sekundär: Aussicht auf die vereinfachte Kommunikation zum Zweck des Sponsorings
Spezielle Produkterfahrung	Erfahrung mit Zurschaustellung von Projekten innerhalb gegebener Plattformen und Foren
Motivation	Zusammenarbeit mit Sponsoren im Rahmen eines Wettbewerbs im Soll-Zustand, sowie finanzielle und materielle Unterstützung durch diese

## Benutzungsmodellierung

Neben den im Konzept erarbeiteten Alleinstellungsmerkmalen, sollen im Rahmen der Benutzungsmodellierung auch die anderen Aufgaben und Interaktionen zwischen dem System und dem Benutzer modelliert werden. Eine kompakte Möglichkeit der Identifikation dieser Aufgaben außerhalb der Alleinstellungsmerkmale, bietet die Erstellung der Use Cases nach Cockburn (vgl. Dix, Filay, Abowd und Beale, 2004).

Nachfolgend werden vor allem die Use Cases der Alleinstellungsmerkmale vorgestellt, alle übrigen Use Cases befinden sich im Anhang.

*Tabelle 6: USE CASE #3: Dashboard aufrufen*

USE CASE #3	Dashboard aufrufen
Goal in Context	Aufrufen der Rubrik „Dashboard“, welche dem Benutzer eine Übersicht über seine Projekte und Daten gibt.

		Berechnet dabei die Gesamtreputation des Benutzers, wenn dieser ein Casemodder ist
Scope & Level		Client, primary task
Preconditions		Der Benutzer ist registriert und eingeloggt
Success End Condition		Das Dashboard wird dem Benutzer zur Einsicht gezeigt und die Gesamtreputation wird berechnet
Failed End Condition		Das Dashboard wird nicht aufgerufen
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte seine Nachrichten einsehen; Der Casemodder möchte seine Gesamtreputation einsehen; Der Casemodder möchte ein Projekt anlegen oder ändern; Der Casemodder möchte ein Projektupdate anlegen oder ändern
DESCRIPTION	Step	Action
	1	Casemodder wählt Funktion „Dashboard“ aus
	2	System berechnet die Gesamtreputation und zeigt dem Benutzer das Dashboard

Tabelle 7: USE CASE #10: Suchstatus setzen

USE CASE #10		Setzen des Sponsorsuchstatus
Goal in Context		Aktivieren des Sponsorsuchstatus, mit dem Sponsoren erkennen, dass ein Casemodder an einem Sponsoring-Programm interessiert ist
Scope & Level		Casemodder client, secondary task
Preconditions		Benutzerkonto ist aktiviert, und die Reputation des Benutzers ist ausreichend hoch
Success End Condition		Der Suchstatus ist gesetzt, und für Sponsoren in der Rubrik sichtbar
Failed End Condition		Der Suchstatus wird nicht gesetzt, und wird von den Sponsoren nicht wahrgenommen.
Primary, Secondary Actors		Casemodder, Sponsor
Trigger		Der Casemodder möchte aktiv die Suche nach einem Sponsor beginnen
DESCRIPTION	Step	Action
	1	Casemodder wählt Funktion „Dashboard“ aus
	2	Casemodder prüft seine Reputation auf dem Dashboard. Ist die Reputation ausreichend hoch, steht die Option „Suchstatus setzen“ zur Verfügung

	3	Casemodder clickt den Button „Suchstatus setzen“
	4	Sponsor bekommt in seiner Rubrik den Suchstatus des Casemodders mitgeteilt

## Anforderungsermittlung

Basierend auf der Benutzungsmodellierung und der Analyse der Stakeholder, erfolgt nun die Auflistung der erarbeiteten Anforderungen.

### Selbstständige Systemaktivitäten

/A0001/ - Das System muss die Gesamtreputation eines Casemodders bei Aufruf seiner Informationen berechnen können.

/A0002/ - Das System muss einem Sponsor aus einer Auswahl von Casemoddern, Vorschläge für die Bildung eines Teams machen können.

/A0003/ - Das System muss Sponsoren in nahezu Echtzeit über neu aktivierte Suchstatus von Casemoddern informieren können.

/A0004/ - Das System muss Daten aus einer Datenbank abrufen und darin abspeichern können.

/A0005/ - Das System sollte dem Benutzer unbekannte (sprich, neue) Aktivitäten an seinen Objekten kenntlich machen können.

/A0006/ - Das System sollte die Reputationspunkte von Objekten anhand der Datenbankergebnisse berechnen können.

/A0007/ - Das System sollte die Login-Sessions aller Benutzer prüfen können.

/A0008/ - Das System sollte bei Schließung eines Benutzerkontos, alle damit zusammenhängenden Objekte löschen können.

### Benutzerinteraktionen

/A0009/ - Das System muss Casemoddern die Möglichkeit geben, einen Sponsorsuchstatus zu setzen, der von Sponsoren im System gesehen wird.

/A0010/ - Das System muss Benutzern die Möglichkeit bieten, rollenspezifische Benutzerkonten zu erstellen.

/A0011/ - Das System muss Benutzern die Möglichkeit bieten, sich mit bestehenden Benutzerdaten einzuloggen.

/A0012/ - Das System muss Benutzern die Möglichkeit bieten, ihr eigenes öffentliches Profil einzusehen.

/A0013/ - Das System muss Benutzern die Möglichkeit bieten, ihre öffentlichen Benutzerdaten zu ändern.

/A0014/ - Das System muss Casemoddern die Möglichkeit bieten, Projekte zu erstellen.

/A0015/ - Das System muss Casemoddern die Möglichkeit bieten, Projekte um Updates zu ergänzen.

/A0016/ - Das System muss Benutzern die Möglichkeit geben, fremde Projekte zu suchen.

/A0017/ - Das System muss Benutzern die Möglichkeit geben, fremde Projekte und ihre Updates einzusehen.

/A0018/ - Das System muss Benutzern die Möglichkeit geben, fremde öffentliche Profile einzusehen.

/A0019/ - Das System muss Benutzern die Möglichkeit bieten, fremde Projekte, Projektupdates und Kommentare mit „Upvotes“ zu versehen.

/A0020/ - Das System muss Sponsoren die Möglichkeit geben, Teams zu erstellen.

/A0021/ - Das System muss Sponsoren die Möglichkeit geben, Teams zu bearbeiten.

/A0022/ - Das System muss Sponsoren die Möglichkeit geben, Teams zu löschen.

/A0023/ - Das System sollte Benutzern die Möglichkeit bieten, die aktuellen Aktivitäten ihrer Objekte in einer zentralen Ansicht einzusehen.

/A0024/ - Das System sollte Benutzern die Möglichkeit bieten, die Reputationspunkte der eigenen Projekte, Projektupdates und Kommentare einzusehen.

/A0025/ - Das System sollte Casemoddern die Möglichkeit bieten, Projekte zu löschen.

/A0026/ - Das System sollte Casemoddern die Möglichkeit bieten, Projektupdates zu löschen.

/A0027/ - Das System sollte Benutzern die Möglichkeit geben, ihr Benutzerkonto zu schließen.

/A0027/ - Das System sollte Benutzern die Möglichkeit geben, systeminterne Nachrichten zu erhalten.

/A0027/ - Das System sollte Benutzern die Möglichkeit geben, systeminterne Nachrichten zu erstellen und zu versenden.

/A0028/ - Das System sollte Benutzern die Möglichkeit geben, systeminterne Nachrichten zu löschen.

## Dokumentation: Architektur

In dieser Rubrik wird im Detail die Architektur des zu entwickelnden Systems erklärt.

### Diagramm

Zunächst wird anhand eines Diagramms die Architektur des zu entwickelnden Systems visuell veranschaulicht.

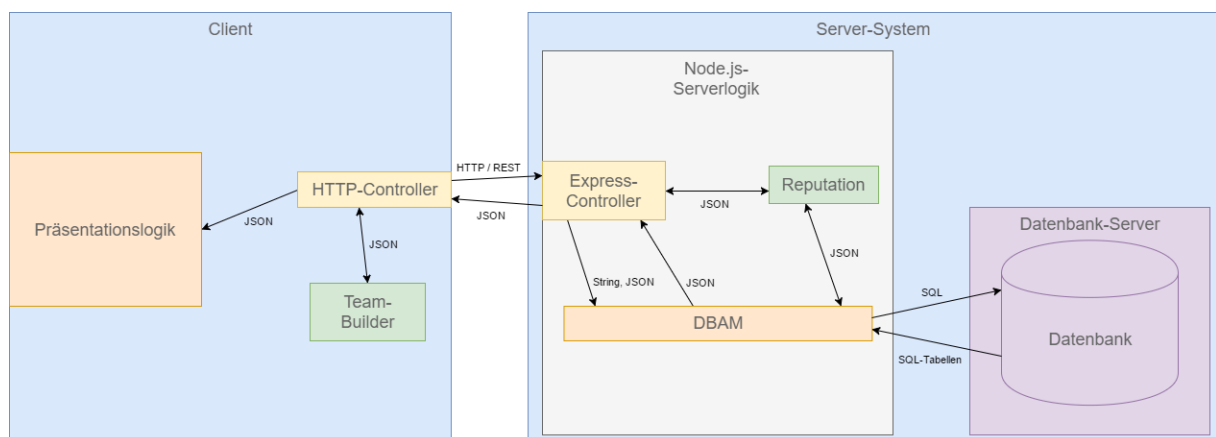


Abbildung 5: Architekturdiagramm

Die Abbildung 5 zeigt die geplante Architektur des Systems im Soll-Zustand. Eine farbliche Unterteilung soll gewährleisten, dass man Controlling-Objekte (gelb), Logik (grün) und I/O-Module (orange) auseinanderhalten kann. Die Präsentationslogik wird hierbei trotz der Umwandlung von Daten als I/O-Objekt gezeigt, da die Daten nur in ihrer Darstellung umgewandelt werden, aber inhaltlich identisch bleiben.

Die Client-Anwendung ist Desktop-basiert, und wird in Java implementiert. Die zusätzliche Bibliothek „JavaFX“ ermöglicht als Präsentationslogik das generieren von „Scenes“, also Benutzeroberflächen, im Client.

Die Logik-Komponente „Team-Builder“, welche das reputationsmäßig beste Team von suchenden Casemoddern vorschlagen soll, befindet sich auf dem Client, da ein Sponsorbenutzer vom Server bereits eine Liste suchender Casemodder findet – der Client kann daraufhin mit den erhaltenen Daten arbeiten.

Zuletzt befindet sich auf dem Client ein Controller-Objekt, welches vor allem die HTTP-Kommunikation und den Erhalt von Daten von der Server-Anwendung regelt.

Die Server-Anwendung ist eine Kombination aus einer Serverlogik und einem Datenbank-Server, welche auf dem gleichen Hardwaresystem laufen sollen. Durch die variable Anbindungsmöglichkeit der Serverlogik und des Datenbankservers, ist das Betriebssystem des Hardwaresystems nicht relevant – positiv auf Implementierung und Installation wirkt sich hier aber ein UNIXoides System (UNIX, Linux, Mac) aus, da sowohl der gewünschte Server für die Datenbank, als auch Node.js als Framework-Fundament für die Serverlogik, mühelos als Module über die Unix-Kommandozeile installiert werden können.

Die Serverlogik beruht auf einer JavaScript-Anwendung, mit einem Node.js-Framework als Anwendungsfundament. Erweitert wird dies durch die Middleware-Bibliothek „Express“, welche open-source ist, und eine große Menge an Funktionen für HTTP-Routing bietet. Die Implementierung einer eigenen Routing-Middleware wird dadurch redundant.

## Datenstrukturen

Der Datenbankserver der Server-Anwendung beinhaltet eine relationale, SQL-basierte Datenbank. Geplant für die Implementierung ist hierbei explizit MySQL, da die bisherigen Erfahrungen mit dieser SQL-Variante am größten sind, und diese auch nahezu vollständig positiv sind. Durch viele Zusammenhänge und Zugehörigkeiten, entfällt auch die Nutzung einer nicht-relationalen Datenbank als Option. Ersichtlich wird dies anhand der Abbildung 2, welche die Zugehörigkeiten der einzelnen Entitäten zeigt.

Zusammen mit dem DBAM-Modul der Serverlogik, bildet der Datenbankserver ein Datenbankmanagementsystem.

## XML und JSON

Die Verwendung von JSON-Objekten erweist sich im Gegensatz zu XML-Dokumenten beim zu entwickelnden System als sinniger, da für die Verwendung von XML-Dokumenten Schemata aufgebaut werden müssen – weiterhin werden für die serverseitige Logik zusätzliche Bibliotheken benötigt, um XML zu verarbeiten. Durch die ECMAScript 5-Kompatibilität von Node.js können JSON-Objekte nativ verarbeitet werden.



## Anwendungslogik

### reputation.js

Dieses Modul übernimmt die Berechnung der Gesamtreputation der Benutzer.

Das Modul enthält insgesamt 6 Konstanten, welche als Modifikatoren die Gewichtung der Reputationspunkte aus den verschiedenen Rubriken bestimmen. Geplant ist, dass Reputationspunkte aus den Bereichen „Projekte“ und „Projektupdates“ stärker gewichtet werden, als Reputationspunkte aus den Rubriken „Kommentare“.

Im Modul befindet sich zunächst eine private Funktion, die als Parameter einen Benutzernamen, einen Elementbegriff (wie bpsw. Projekte, Kommentare etc.) sowie einen Modifikator entgegennimmt. Die Menge der zum Benutzer zugehörigen Elemente wird anhand des Benutzernamens aus der Datenbank entnommen, und das Produkt aus der Elementmenge und dem Modifikator wird zurückgegeben:

```
Funktion berechneWert(benutzername, element, modifikator)
{
    var elementsumme = 0;
    switch(element){
        // ermittle Elementart und führe entsprechende DB-Anfrage durch:
        elemente = dbam.holeElementeEinesBenutzers(benutzername);
        elementsumme += elemente.length;
    }
    return elementsumme * modifikator;
}
```

Zwei weitere Methoden sind öffentlich zugänglich:

Die erste Methode gibt die Gesamtreputation des Benutzers zurück, indem sie mit dem entgegengenommenen Benutzernamen die private Funktion für alle benötigten Rubriken ausführt, und die Summe als abgerundeter Integer zurückgibt.

Die zweite Methode gibt die rohen Reputationswerte eines Benutzers in den Rubriken „Projekte“ und „Projektupdates“ zurück, indem sie die private Funktion nur für diese beiden Rubriken und jeweils mit einem neutralen Modifikator ausführt. Diese Funktion wird genutzt, um zu ermitteln, ob ein Benutzer anhand seiner Reputation innerhalb dieser Rubriken, bereits auf Sponsorensuche gehen darf.

### TeamBuilder.java

Die Java-Klasse „TeamBuilder“ übernimmt bei Einsicht der sponsorensuchenden Casemodder, für Sponsoren die vorsorgliche Erstellung eines Case Modding-Teams. Dabei wird vor allem auf die durchschnittliche Reputation bestimmter Merkmale des Teams geachtet.

Das Modul nimmt die übergebene Liste von sponsorensuchenden Casemoddern entgegen, und durchsucht diese nach passenden Werten. Es werden immer bis zu 3 Casemodder zu einem Teamvorschlag zusammengefasst. Findet sich ein Casemodder mit besseren Werten, werden die Teamvorschläge aktualisiert.



## Dokumentation: User Interface

Die nachfolgende Rubrik befasst sich mit dem Prototyp des User Interface.

### Prototyp

Die nachfolgenden Abbildungen, auf die in den Erklärungen eingegangen wird, sind keine finalen Versionen, sondern sollen, wie bei einem Prototyp üblich, lediglich die Funktionalitäten der Oberfläche darstellen – Änderungen am Design oder am Layout der Elemente, sind nicht ausgeschlossen.

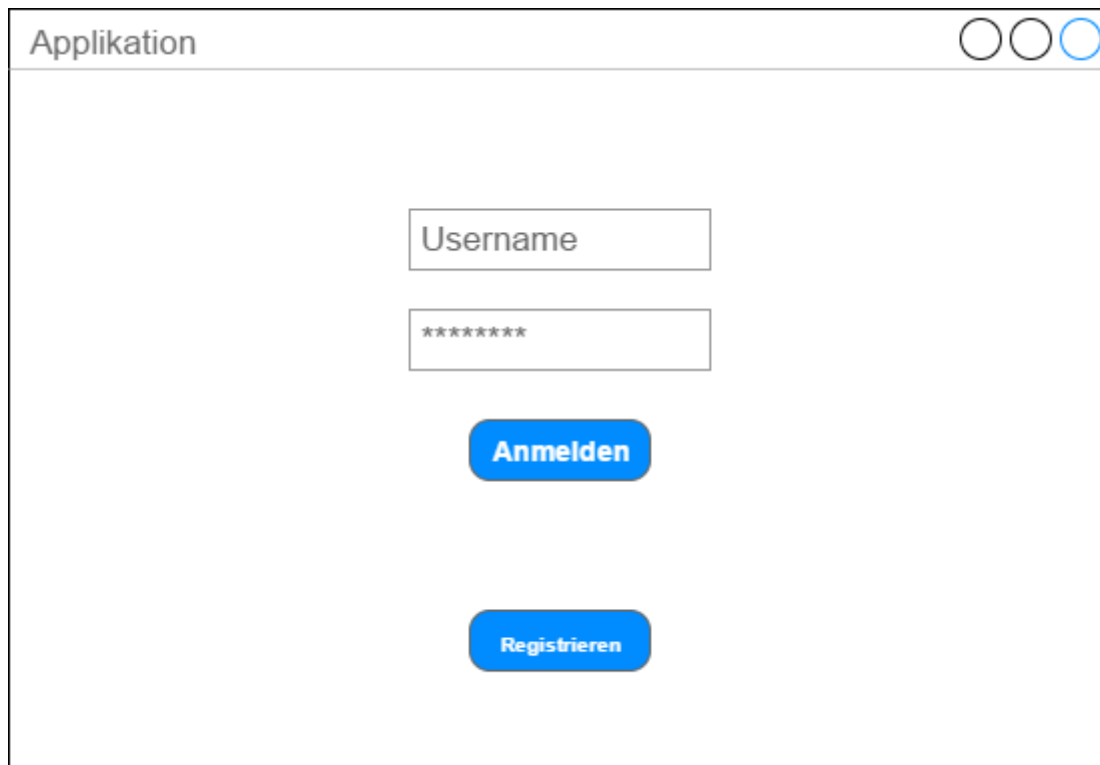
The image shows a wireframe of a login application window. The window has a title bar with the text 'Applikation' on the left and three standard window control buttons (minimize, maximize, close) on the right. The main content area is centered and contains a login form. The form consists of two text input fields: the top one is labeled 'Username' and the bottom one contains a series of asterisks '\*\*\*\*\*' to represent a password. Below the password field are two blue, rounded rectangular buttons. The top button is labeled 'Anmelden' (Login) and the bottom button is labeled 'Registrieren' (Register). The entire interface is enclosed in a thin black border.

Abbildung 7: UI-Prototyp Login

Die Client-Anwendung startet mit einem Login-Bildschirm, der, wie die Abbildung 3 zeigt, eine Eingabemaske für den Usernamen und das Passwort anbietet. Hat man noch kein Benutzerkonto, kann man sich alternativ auch registrieren lassen. Die beiden Registrierungsmasken für Casemodder und Sponsoren sind unter Abbildung 8 und 9 zu sehen.

Nach einem erfolgreichen Login wird man auf das benutzerkontenspezifische Dashboard weitergeleitet. Die Abbildung 10 zeigt die Casemodder-Version des Dashboards, auf dem der Benutzer seine Gesamtreputation einsehen kann, sowie eine Übersicht über Aktivitäten im System, die auf mit dem Benutzer zusammenhängende Daten referenzieren. Auf der Abbildung ist auch eine mögliche Darstellung gegeben, wie der Casemodder seinen Sponsorsuchstatus aktivieren kann.

Applikation - Registrierung

Registrierung

☒ Casemodder
☐ Sponsor

Email-Adresse

Passwort

Geburtsdatum

Registrieren!

[Zurück zum Login](#)

Abbildung 8: UI-Prototyp Registrierung Casemodder

Applikation - Registrierung

Registrierung

☐ Casemodder
☒ Sponsor

Email-Adresse

Passwort

Geburtsdatum

Verifikationsdatei

Durchsuchen

Registrieren!

[Zurück zum Login](#)

Abbildung 9: UI-Prototyp Registrierung Sponsor



Abbildung 10: UI-Prototyp Dashboard, Casemodder

Unter der Rubrik "Mein Profil", kann man sein eigenes, öffentliches Profil betrachten. Bei Casemoddern gibt es unterhalb der Daten die Liste der Projekte, die zu den Casemoddern gehören (vgl. Abbildung 11), bei Sponsoren eine Auflistung der letzten Aktivitäten des Benutzers (vgl. Abbildung 12).

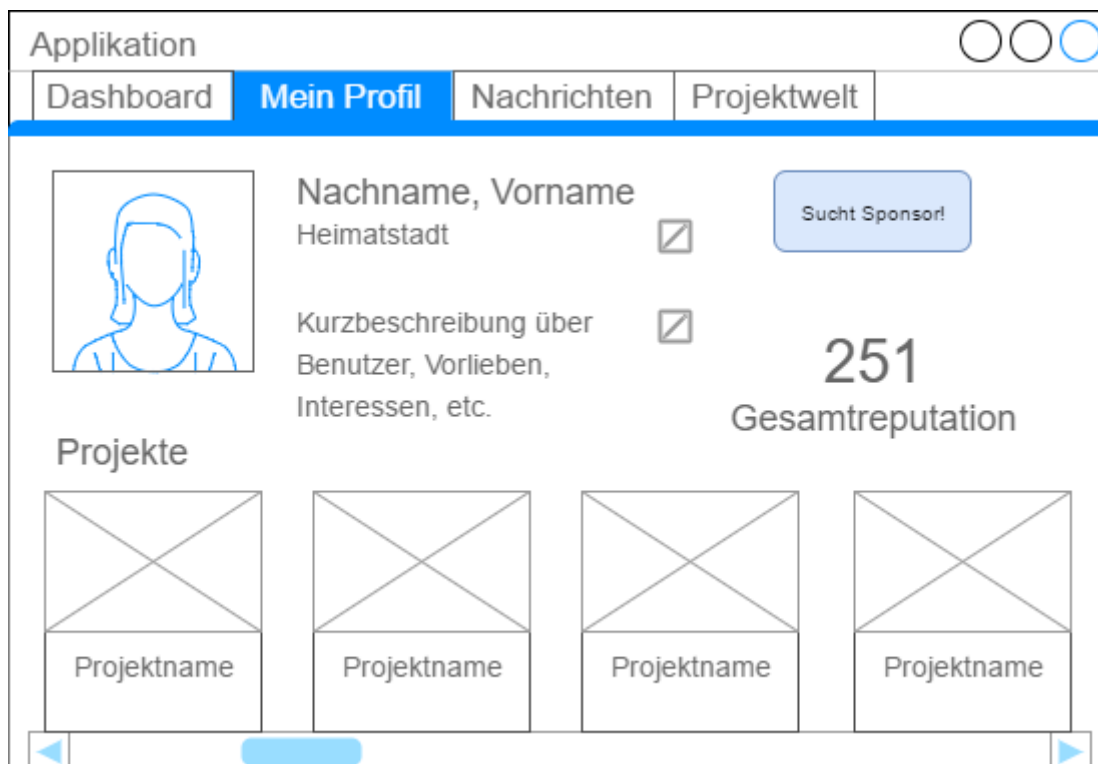
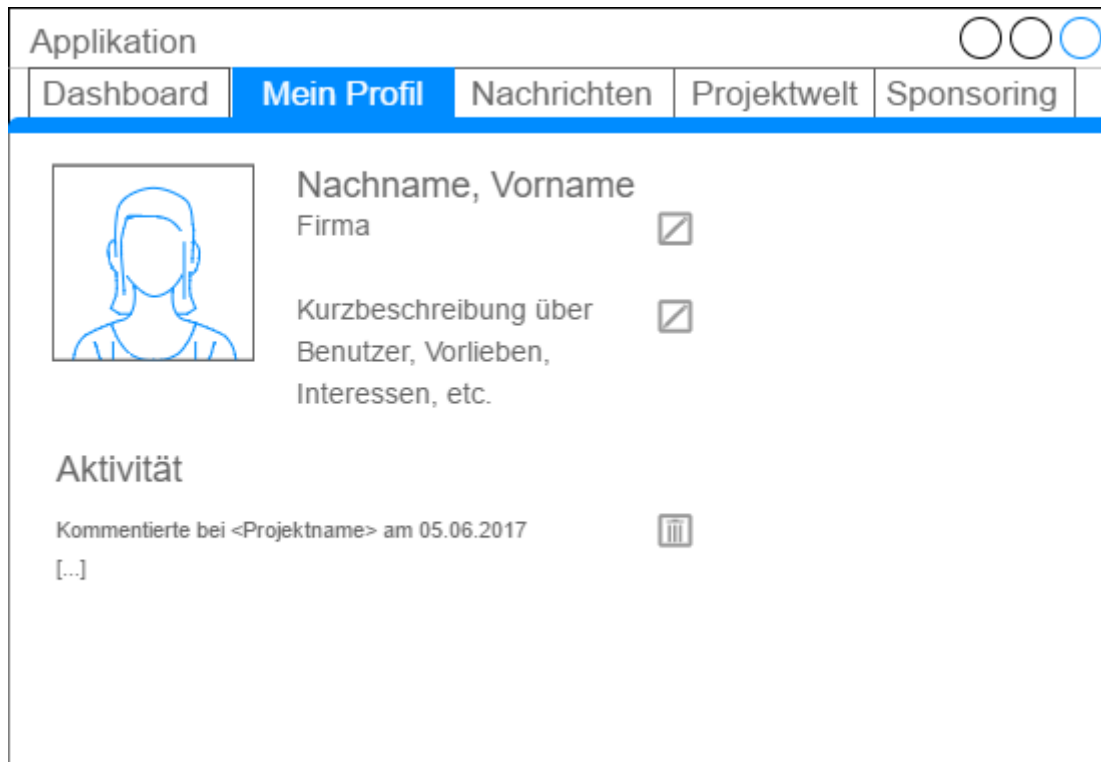


Abbildung 11: UI-Prototyp Casemodder-Profil

Die Bearbeitungs-Icons wandeln die Textstellen in Eingabefelder um, die man dann verändern kann (vgl. Abbildung 13).

Die Rubrik „Projektwelt“ zeigt eine Liste neuester Projekte, sortiert nach der höchsten Reputation (vgl. Abbildung 14). Casemodder haben daneben eine Einsicht ihrer eigenen Projekte, Sponsoren dagegen haben eine Abkürzung zu ihrem Team. In der Suchmaske kann man Suchbegriffe eingeben, nach denen das System sucht.



The image shows a UI prototype for a 'Sponsor-Profil' (Sponsor Profile) within an application. At the top, there is a header bar with the title 'Applikation' and three circular icons on the right. Below the header is a navigation menu with five tabs: 'Dashboard', 'Mein Profil' (which is highlighted in blue), 'Nachrichten', 'Projektwelt', and 'Sponsoring'. The main content area is divided into two sections. The first section, titled 'Nachname, Vorname', contains two input fields: 'Firma' and 'Kurzbeschreibung über Benutzer, Vorlieben, Interessen, etc.'. Each input field has a small square icon with a diagonal line, indicating it is an editable text field. The second section, titled 'Aktivität', shows a single activity entry: 'Kommentierte bei <Projektname> am 05.06.2017' followed by an ellipsis '[...]'. To the right of this entry is a small icon of a document with a list, representing a comment or activity log.

Abbildung 12: UI-Prototyp Sponsor-Profil

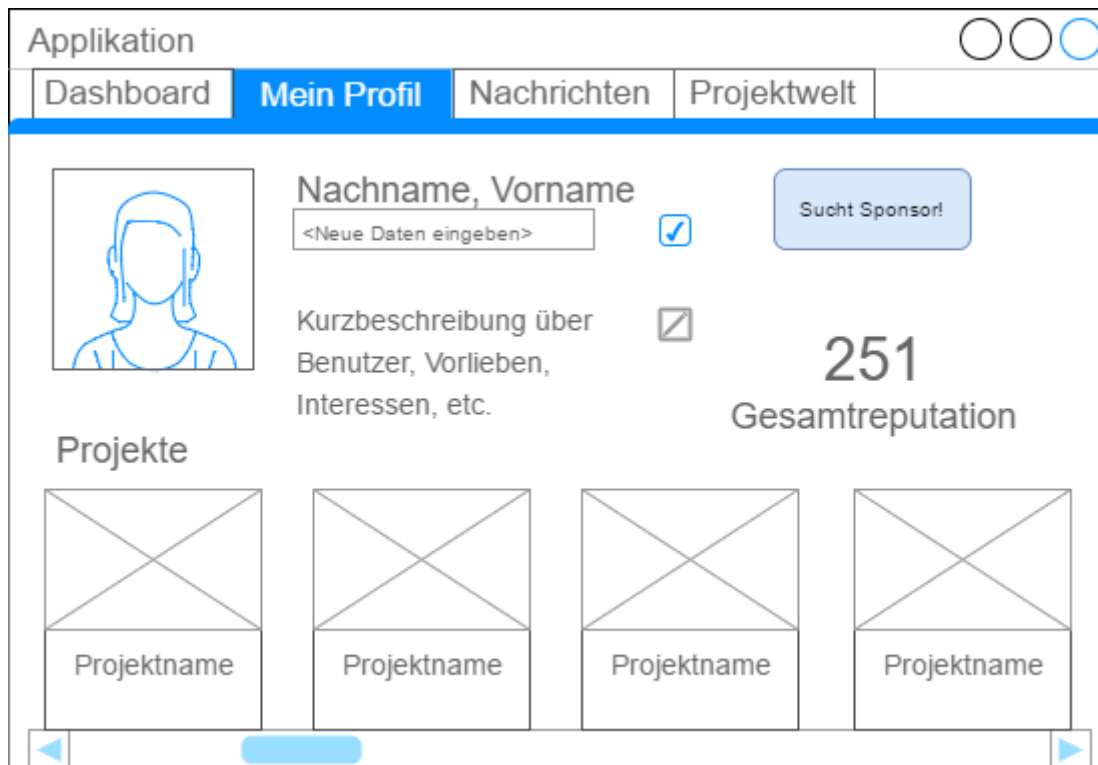


Abbildung 13: UI-Prototyp Profil Eingabemaske

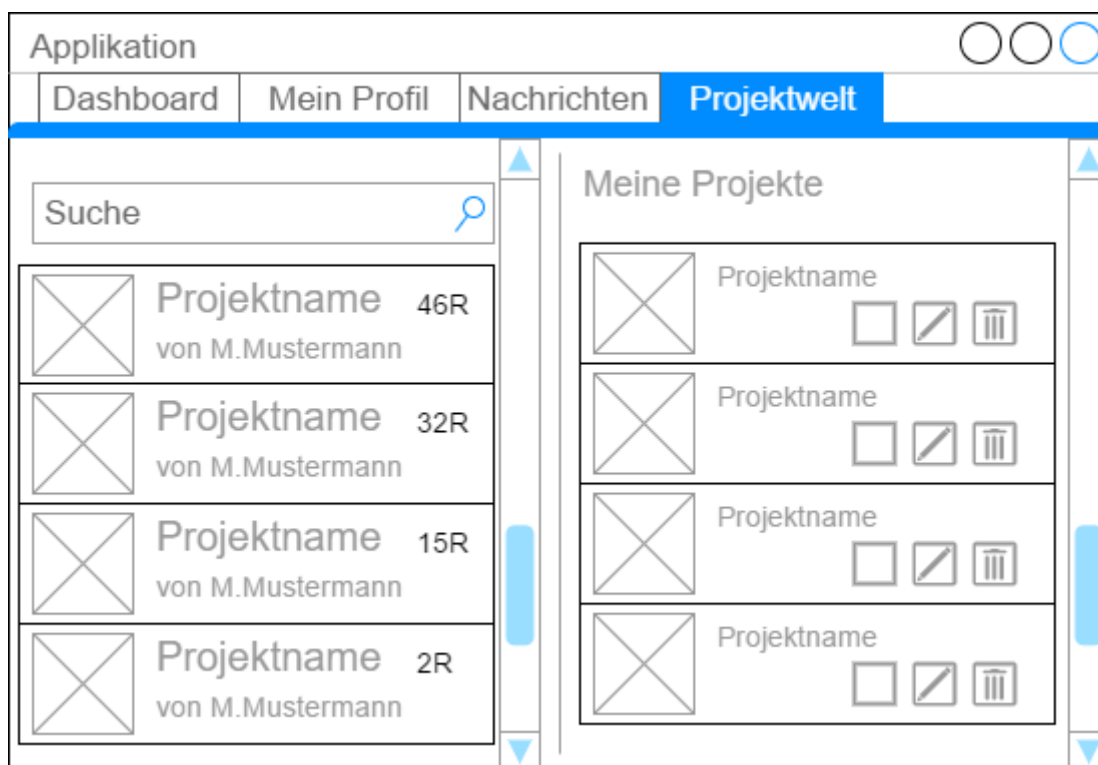


Abbildung 14: UI-Prototyp Rubrik "Projektwelt"

In der Projektansicht Kann ein Benutzer ein Fremdprojekt und dessen Kommentare mit Upvotes versehen (vgl. Abbildung 15). Eigene Kommentare können zudem gelöscht werden. Ähnlich wie bei der Profilansicht, hat die Ansicht der eigenen Projekte Bearbeitungsoptionen.

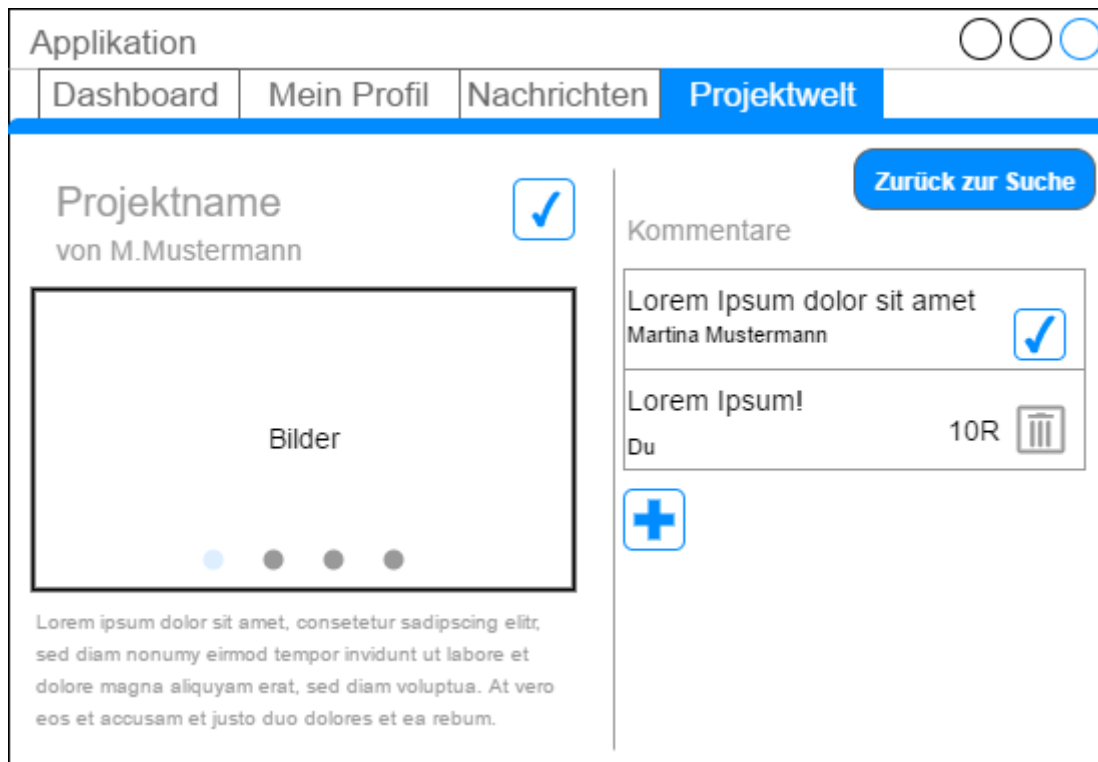


Abbildung 15: UI-Prototyp Projektansicht

In der Rubrik „Nachrichten“ erscheinen neue Nachrichten hervorgehoben auf (vgl. Abbildung 16).

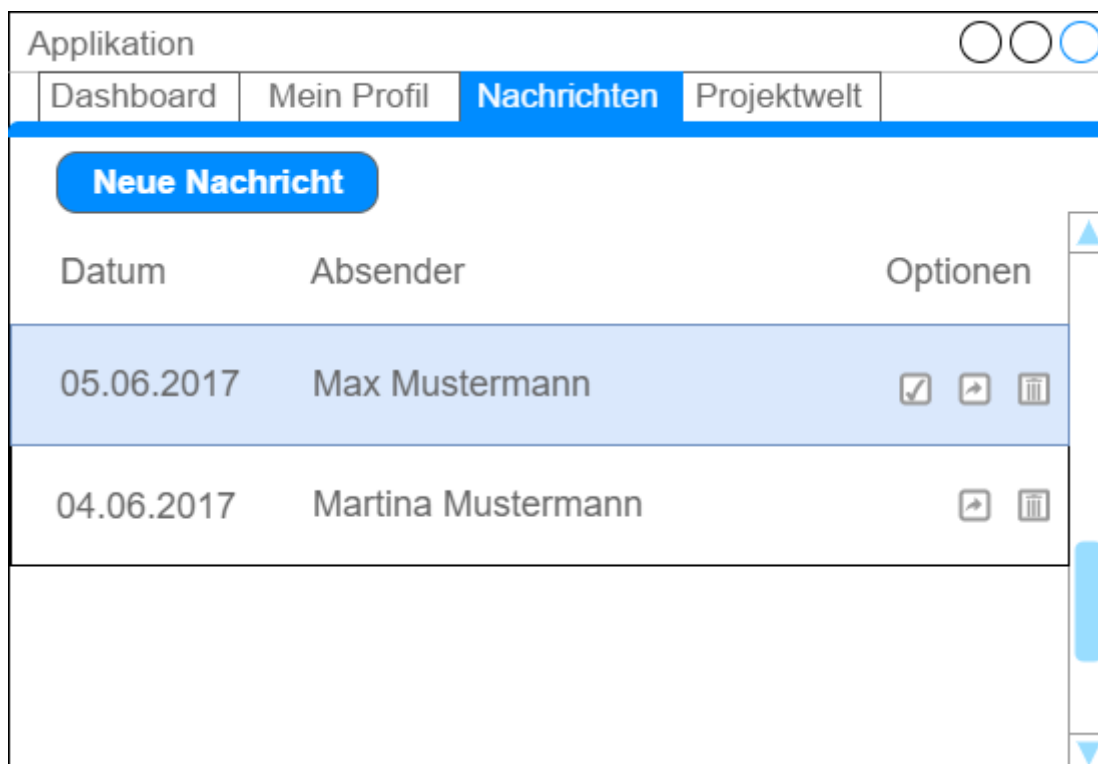


Abbildung 16: UI-Prototyp Nachrichten

Das Auswählen einer Nachricht klappt diese auf und markiert sie automatisch als gelesen (vgl. Abbildung 17).

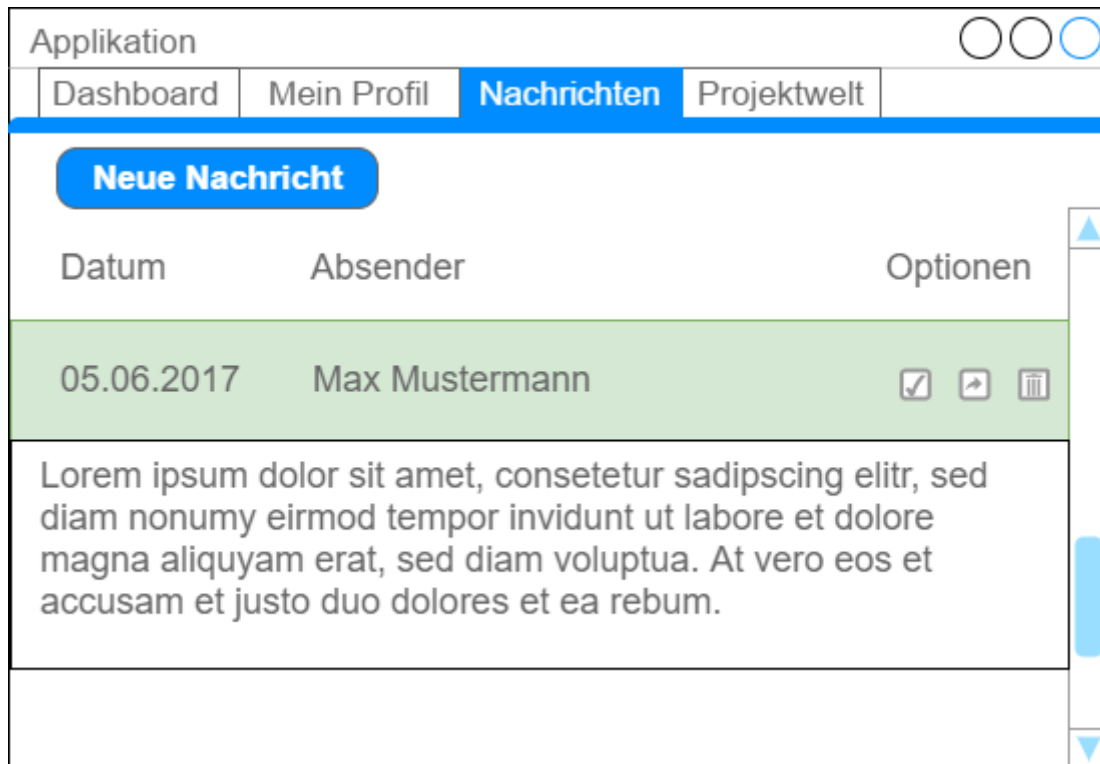


Abbildung 17: UI-Prototyp einzelne Nachricht

Nachrichten können unter Bestätigung eines Lösungsdialogs gelöscht werden (vgl. Abbildung 18).

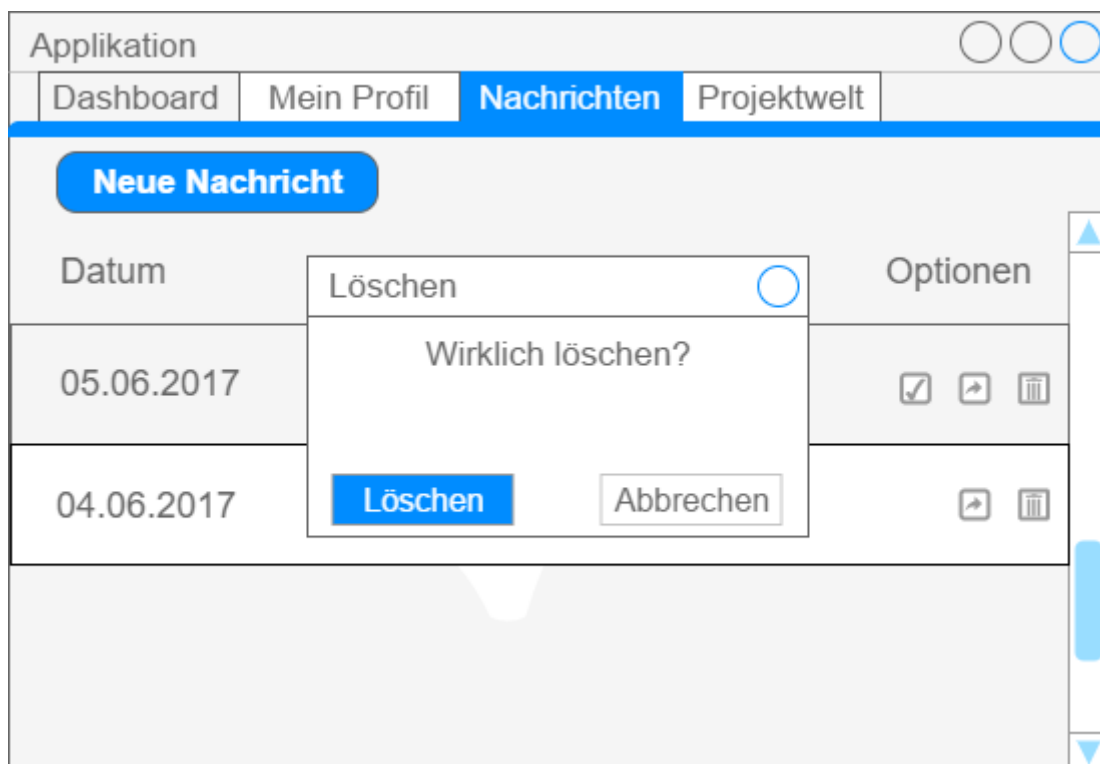


Abbildung 18: UI-Prototyp Nachricht löschen

Sponsoren haben eine eigene Rubrik „Sponsoring“, die nur für diese Benutzerrolle einsehbar ist. In dieser sehen Sponsoren eine Liste von Casemoddern, die auf der Suche nach Sponsoren sind, sowie einen Vorschlag für ein mögliches Casemodder-Team (vgl. Abbildung 19). Auch können sie hier ihr eigenes Team verwalten. Die Abbildung 20 zeigt den Verwaltungsbildschirm für das eigene Team.

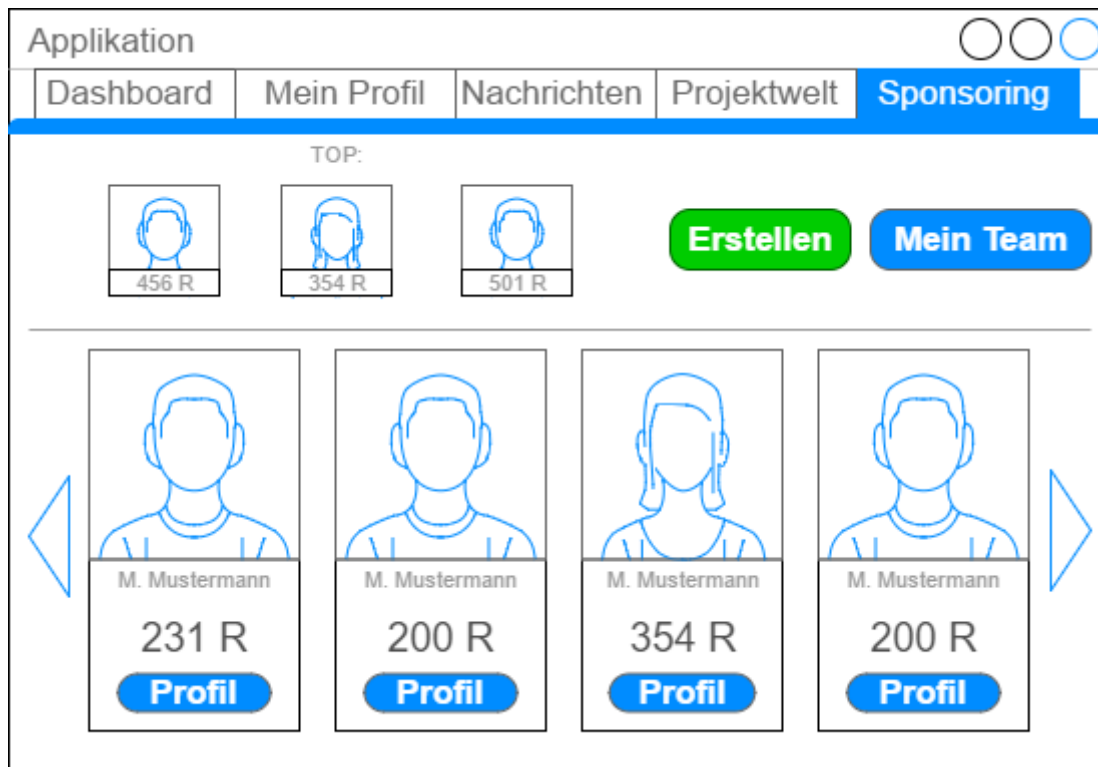


Abbildung 19: UI-Prototyp Sponsor Team-Builder

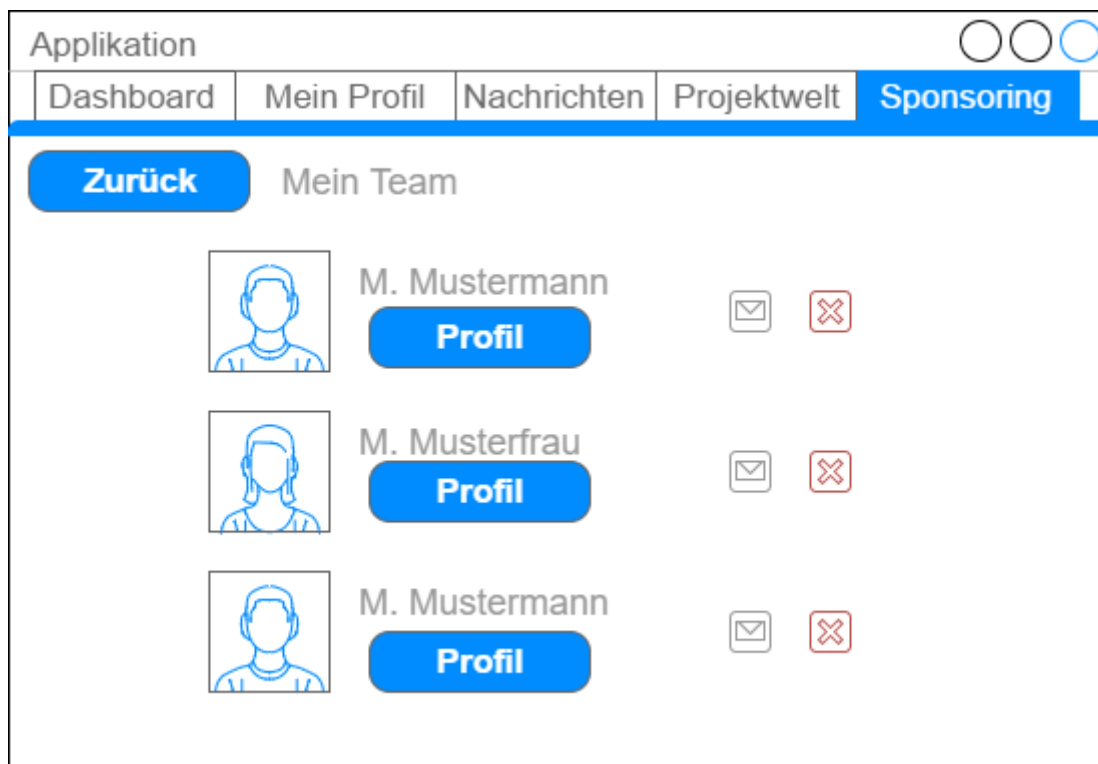


Abbildung 20: UI-Prototyp Sponsor Team-Ansicht



Casemodder können aus einem Team entfernt werden. Für die Entwicklung ist vorgesehen, dass ein Team aus nicht mehr als 3 Mitglieder bestehen soll.

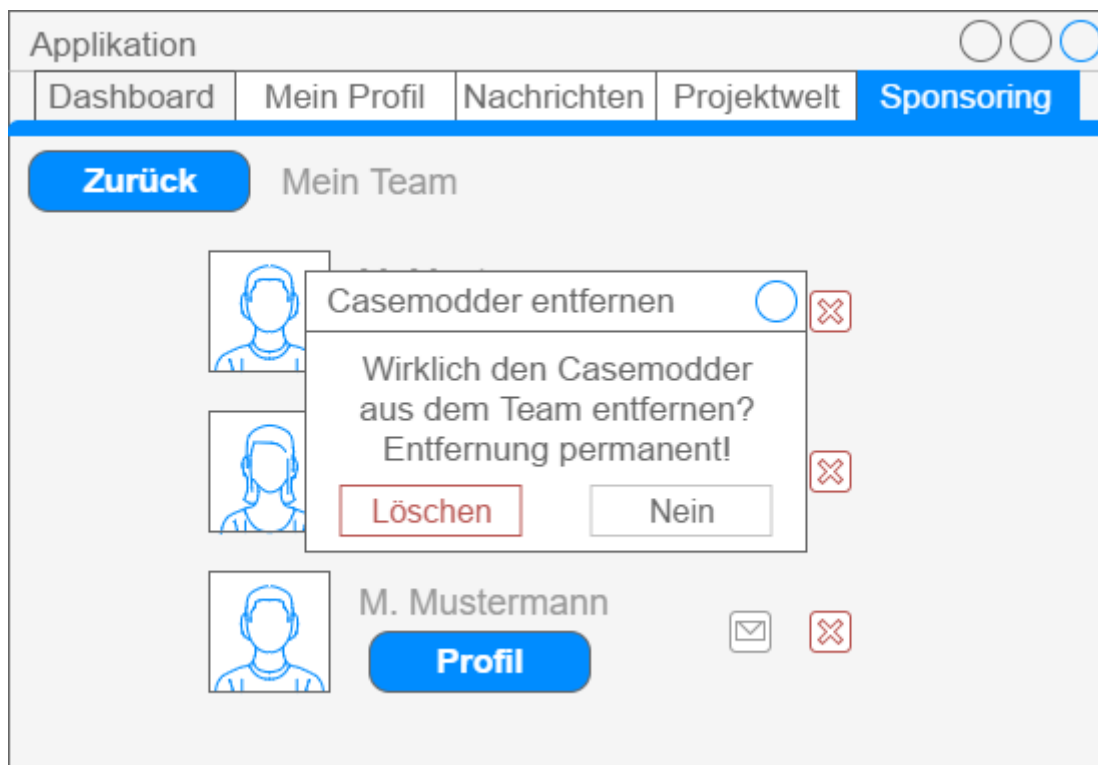


Abbildung 21: UI-Prototyp Sponsor Team Entfernungsdialog

## Evaluation

Den Abschluss der Dokumentation bildet die Evaluation der Benutzeroberfläche.

### Evaluationsmethode

Als Evaluationsmethode wird die von Nielsen und Molich angeführte Heuristische Evaluation genutzt.

Wichtigster (negativer) Faktor bei der Wahl der Evaluationsmethode ist die Abwesenheit der Stakeholder. Zum Zeitpunkt der Evaluation konnten weder Casemodder, noch Sponsoren für eine Evaluation zur Rate gezogen werden, was die Schwierigkeit erzeugt, dass kaum eine Evaluationsmethode ansonsten infrage kommt.

Die Heuristische Evaluation erlaubt eine zeit- und kostengünstige Evaluation mit wenig Aufwand, und wurde auch genau aus diesen Gründen entwickelt (vgl. Nielsen & Molich, 1990). Wichtig ist auch, dass die Stakeholder als Evaluationsteilnehmer nicht zwingend zu instrumentalisieren sind, obgleich Kenntnisse auf dem Gebiet der Domäne oder Usability-Expertise, entsprechend bessere Ergebnisse bringen.

Als Evaluator stellt sich der Projektleiter selbst zur Verfügung. Er selbst ist ein Entwickler, der ein ernsthaftes Interesse an der Aufdeckung von Usability-Problemen und deren Beseitigung hat.

Die Bewertung der einzelnen UI-Mockups wird in der nachfolgenden Tabelle aufgeführt, wobei die Punkte A – I jeweils für die nach Nielsen bekannten Heuristiken stehen:

- Visibility of system status
- Aesthetic and minimalist design
- User control and freedom
- Consistency and standards
- Recognition rather than recall
- Flexibility and efficiency of use
- Recognition, diagnoses and recovery from errors
- Help and documentation
- Match between system and real world

Die Heuristiken werden nach einer Punktzahl von 1 bis 10 bewertet – auf diese Weise lässt sich der Grad an Usability anhand der Gesamtpunktzahl ablesen.

*Tabelle 8: Heuristische Evaluation Mockups*

<b>Mockup</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
Login	8	10	7	9	10	10	9	6	7
Registrierung Casemodder	8	10	7	9	10	9	9	8	7
Registrierung Sponsor	8	10	7	9	10	5	9	8	8
Dashboard	10	8	5	8	10	6	6	9	10
Eigenes Profil Casemodder	10	7	8	8	10	8	6	7	7
Eigenes Profil Sponsor	10	7	8	8	10	8	6	7	7
Projektwelt	8	6	6	8	7	7	6	5	6
Eigenes Projekt	8	8	8	9	8	8	6	7	7
Fremdes Profil Casemodder	8	7	6	8	9	6	8	8	9
Fremdes Profil Sponsor	8	7	6	8	9	6	8	8	9
Fremdes Projekt	8	8	6	9	9	8	8	6	9
Nachrichten	10	9	8	8	10	7	7	7	8
Nachricht erstellen	8	9	7	8	10	7	7	7	9
Sponsoring-Welt	8	7	6	7	7	8	6	7	6
Team-Verwaltung	10	9	8	8	10	8	8	8	7

## Quellverzeichnis

Leonid Vilents: *Umfrage zum Sponsoring im Rahmen von Case Modding – Sponsorenfragebogen*.

[https://docs.google.com/forms/d/e/1FAIpQLSeEuQzn5TZdNf\\_EszUwIGoyZ4hS1cfgwAV-DlWdbipOlsaoAg](https://docs.google.com/forms/d/e/1FAIpQLSeEuQzn5TZdNf_EszUwIGoyZ4hS1cfgwAV-DlWdbipOlsaoAg) (Zugriff am 17.05.2017)

Leonid Vilents: *Umfrage zum Sponsoring im Rahmen von Case Modding – Casemodderfragebogen*.

[https://docs.google.com/forms/d/e/1FAIpQLSeEuQzn5TZdNf\\_EszUwIGoyZ4hS1cfgwAV-DlWdbipOlsaoAg](https://docs.google.com/forms/d/e/1FAIpQLSeEuQzn5TZdNf_EszUwIGoyZ4hS1cfgwAV-DlWdbipOlsaoAg) (Zugriff am 23.05.2017)

Allan Dix; Janet Filay; Gregory D. Abowd; Russel Beale: *Human-Computer Interaction – 3rd Edition*. Pearson Prentice Hall, 2004. ISBN: 0130-461091

PCGamer: *A beginner's guide to case modding*. In: *PC Gamer*, 07.10.2015

<http://www.pcgamer.com/a-beginners-guide-to-case-modding> (Zugriff am 02.05.2017)

Selim Baykara: *Casemodding: Grundlagen und Anleitungen fürs PC-Tuning*. In: *GIGA Software*, 22.07.2015

<http://www.giga.de/extra/ratgeber/specials/casemodding-grundlagen-und-anleitungen-fuers-pc-tuning/> (Zugriff am 02.05.2017)

Tom Portsmouth: *PC case modding: a beginner's guide*. In: *techradar*, 03.10.2010

<http://www.techradar.com/news/computing/pc/pc-case-modding-a-beginner-s-guide-719878> (Zugriff am 02.05.2017)

Dcmm.de: Die deutsche Casemod Meisterschaft

<http://dcmm.de/de/was-ist-die-deutsche-casemod-meisterschaft.htm> (Zugriff am 04.05.2017)

Preim, B.; Dachzelt, R.: *Interaktive Systeme – Band 1: Grundlage, Graphical User Interfaces, Informationsvisualisierung*. Springer Verlag, Berlin, 2010. ISBN 978-3-642-05401-3

Caseking: *Sponsoring-Formular*

<http://www.caseking.de/sponsoring-formular> (Zugriff am 06.05.2017)

Overclock.net: *Sponsored Builds & Case Mods*

<http://www.overclock.net/f/362/sponsored-builds-case-mods> (Zugriff am 06.05.2017)

Norman, D.; Draper, S.: *User centered system design – New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates Publishers, London, 1986. ISBN 978-0-898-59872-8

Mayhew, D.: *The Usability engineering lifecycle - A Practitioner's Handbook for User Interface Design*. Morgan Kaufmann, 1999. ISBN 978-1-558-60561-9

DIN Deutsches Institut für Normung e.V.: *Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme (ISO 9241-210:2010); Deutsche Fassung EN ISO 9241-210:2010*. Beuth Verlag GmbH, Berlin, 2011.

Rosson, M.B.; Carrol, J.M.: *Usability Engineering – Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann, 2002. ISBN 978-1-558-60712-5

Rosson, M.B.; Carrol, J.M.: *Scenario-Based Design*. In: Jacko, J.A.; Sears, A.: *The human-computer interaction handbook – fundamentals, evolving technologies, and emerging applications* (S. 1041 – 1060). Routledge, London, 2003. ISBN 978-1-410-60672-3

Nielsen, Jakob; Molich, Rolf: *Heuristic evaluation of user interfaces: Proceedings of the ACM, CHI90 Conference*. Seattle, WA, April 1-5 1990, S. 249-256

## Anhang

### Use Cases

Tabelle 9: USE CASE #1: Registrierung

USE CASE #1		Registrierung
Goal in Context		Anlegen eines Benutzerkontos zur Interaktion mit dem System
Scope & Level		Client, primary task
Preconditions		Der Benutzer verfügt über eine gültige Email-Adresse, die noch nicht im System registriert ist.
Success End Condition		Das Benutzerkonto wurde angelegt.
Failed End Condition		Das Benutzerkonto wurde nicht angelegt.
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte mit dem System interagieren
DESCRIPTION	Step	Action
	1	Benutzer wählt auf dem Startbildschirm „Registrieren“ aus
	2	Benutzer füllt die Eingabemaske mindestens mit Pflichtdaten und schickt das Formular ab
	3	System gibt Feedback über den Erfolg der Registrierung und sendet an die angegebene Email ein Aktivierungstoken

Tabelle 10: USE CASE #2: Anmeldung

USE CASE #2		Anmeldung
Goal in Context		Einloggen in ein vorhandenes Benutzerkonto zur Interaktion mit dem System
Scope & Level		Client, primary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Das Login ins Benutzerkonto ist erfolgreich
Failed End Condition		Das Login schlägt fehl
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte mit dem System interagieren
DESCRIPTION	Step	Action
	1	Benutzer wählt auf dem Startbildschirm „Login“ aus
	2	Benutzer gibt die Zugangsdaten ein und schickt das Formular ab

	3	System prüft die Zugangsdaten und leitet bei Erfolg auf das Dashboard des Benutzers um
--	---	--

Tabelle 11: USE CASE #4: Benutzerprofil einsehen

USE CASE #4		Profil einsehen
Goal in Context		Das eigene, öffentliche Benutzerprofil einsehen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Das Benutzerprofil wurde erfolgreich eingesehen
Failed End Condition		Das Benutzerprofil kann nicht eingesehen werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte sein öffentliches Profil einsehen
DESCRIPTION	Step	Action
	1	Benutzer wählt in der Rubrik „Dashboard“ die Option „Öffentliches Profil einsehen“ aus.

Tabelle 12: USE CASE #5: Benutzerprofil bearbeiten

USE CASE #5		Profil aktualisieren
Goal in Context		Das eigene, öffentliche Benutzerprofil aktualisieren
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Das Benutzerprofil wurde erfolgreich aktualisiert
Failed End Condition		Das Benutzerprofil kann nicht aktualisiert werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte seine persönlichen Daten, die öffentlich sichtbar sind, aktualisieren
DESCRIPTION	Step	Action
	1	Benutzer wählt in der Rubrik „Dashboard“ die Option „Öffentliches Profil einsehen“ aus.
	2	Benutzer ändert gewünschte Datensätze.

Tabelle 13: USE CASE #6: Nachrichten abrufen

USE CASE #6		Nachrichten abrufen
Goal in Context		Die eigenen Nachrichten einsehen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Die eigenen Nachrichten wurden den Wünschen nach verwaltet
Failed End Condition		Die Nachrichten konnten nicht eingesehen werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte einen Überblick über seine systeminterne Kommunikation haben.
DESCRIPTION	Step	Action
	1	Benutzer wählt die Rubrik „Nachrichten“ aus.

Tabelle 14: USE CASE #7: Bestimmte Nachricht einsehen

USE CASE #7		Nachricht einsehen
Goal in Context		Eine bestimmte Nachricht in den Nachrichten einsehen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Die Nachricht wurde eingesehen
Failed End Condition		Die Nachricht konnte nicht eingesehen werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte eine bestimmte Nachricht in seinem Posteingang lesen
DESCRIPTION	Step	Action
	1	Benutzer wählt die Rubrik „Nachrichten“ aus.
	2	Aus der Liste an erhaltenen Nachrichten, wählt der Benutzer eine Nachricht zur Einsicht aus.

Tabelle 15: USE CASE #8: Neue Nachricht erstellen

USE CASE #8		Nachricht erstellen
Goal in Context		Eine neue Nachricht erstellen, die an den Empfänger gesendet wird
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Nachrichterstellung und Versand gelingt
Failed End Condition		Nachrichterstellung schlägt fehl
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte eine Kommunikation beginnen oder pflegen
DESCRIPTION	Step	Action
	1	Benutzer wählt die Rubrik „Nachrichten“ aus.
	2	Benutzer wählt „Neue Nachricht“ aus.
	3	Benutzer füllt Empfänger und Nachricht aus, und wählt Absenden. Die Nachricht ist erstellt und versandt.

Tabelle 16: USE CASE #9: Nachricht löschen

USE CASE #9		Nachricht löschen
Goal in Context		Eine Nachricht löschen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Die Nachricht wurde gelöscht
Failed End Condition		Die Nachricht konnte nicht gelöscht werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte eine Nachricht löschen
DESCRIPTION	Step	Action
	1	Benutzer wählt die Rubrik „Nachrichten“ aus.
	2	Aus der Liste an erhaltenen Nachrichten, wählt der Benutzer eine Nachricht zur Einsicht aus.
	3	Benutzer wählt in der Nachricht-Einsicht „Löschen“.
	4	Benutzer bestätigt die Löschung.
SUBVARIATIONS		Branching Action
	2a	In der Liste an erhaltenen Nachrichten, wählt der Benutzer die zu löschende Nachricht, und wählt die Schnelloption „Löschen“
	3a	Benutzer bestätigt die Löschung.

Tabelle 17: USE CASE #11: Benutzerkonto schließen

USE CASE #11		Benutzerkonto schließen
Goal in Context		Das eigene Benutzerkonto schließen und alle zusammenhängenden Daten löschen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Das Benutzerkonto wurde gelöscht
Failed End Condition		Das Benutzerkonto konnte nicht gelöscht werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte sein Benutzerkonto schließen
DESCRIPTION	Step	Action
	1	In der Rubrik „Dashboard“, wählt der Benutzer die Option „Benutzerkonto schließen“ aus.
	2	Der Benutzer gibt zur Bestätigung der Benutzerkontenlöschung sein Passwort ein.
	3	Das Konto wird gelöscht, und der Benutzer wird auf den Login-Bildschirm geleitet.

Tabelle 18: USE CASE #12: Fremdes Profil einsehen

USE CASE #12		Fremdes Profil einsehen
Goal in Context		Ein fremdes Benutzerprofil einsehen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten
Success End Condition		Die Einsicht des fremden Profils gelingt
Failed End Condition		Die Einsicht des fremden Profils gelingt nicht
Primary, Secondary Actors		Benutzer des Systems
Trigger		Ein Benutzer möchte die öffentlichen Informationen eines anderen Benutzers einsehen
DESCRIPTION	Step	Action
	1	In der Rubrik „Projektwelt“, wählt der Benutzer ein Projekt aus.
	2	In der Projektansicht, wählt er das Profilbild des Projektinhabers aus.
SUBVARIATIONS		Branching Action
	1a	In der Rubrik „Sponsoring“, wählt der Sponsor einen der Benutzer aus.



Tabelle 19: USE CASE #13: Fremdes Element "upvoten"

USE CASE #13		Fremdes Element mit „Upvote“ versehen
Goal in Context		Ein fremdes Element mittels eines „Upvotes“ positiv bewerten und dadurch die Reputation des Projektinhabers erhöhen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Das fremde Element erhält ein neues Upvote
Failed End Condition		Das fremde Element behält seine alte Zahl an Upvotes
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte das fremde Element positiv bewerten
DESCRIPTION	Step	Action
	1	In der Projektansicht, wählt der Benutzer den Upvote-Button für das respektive Element aus. Das Upvote wird registriert.

Tabelle 20: USE CASE #14: Projekt erstellen

USE CASE #14		Projekt erstellen
Goal in Context		Ein neues Projekt erstellen
Scope & Level		Client, primary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Das neue Projekt wird angelegt und ist öffentlich sichtbar
Failed End Condition		Das neue Projekt wird nicht angelegt oder ist nicht sichtbar
Primary, Secondary Actors		Casemodder
Trigger		Der Benutzer möchte seine Fähigkeiten demonstrieren und einer Bewertung durch Zweite aussetzen
DESCRIPTION	Step	Action
	1	Benutzer wählt „Projektwelt“ aus
	2	Benutzer wählt „Neues Projekt“ aus
	3	Benutzer gibt die benötigten Daten ein und speichert das neue Projekt ab.

Tabelle 21: USE CASE #15: Projektupdate anlegen

USE CASE #15		Projektupdate erstellen
Goal in Context		Ein Update zu einem Projekt erstellen, dass den Fortschritt kennzeichnen soll.
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert und er hat bereits ein Projekt
Success End Condition		Das Projekt erhält ein zugehöriges, neues Update
Failed End Condition		Das Anlegen des Updates schlägt fehl
Primary, Secondary Actors		Casemodder
Trigger		Der Benutzer möchte den Fortschritt seines Projektes dokumentieren.
DESCRIPTION	Step	Action
	1	In der Projektansicht, wählt der Benutzer „neues Update“ aus.
	2	Benutzer gibt alle benötigten Daten an, und, speichert das neue Update.

Tabelle 22: USE CASE #16: Nach Projekt suchen

USE CASE #16		Nach einem Projekt suchen
Goal in Context		Ein fremdes Projekt anhand von Suchbegriffen suchen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Es werden Suchergebnisse präsentiert, die mit dem Suchbegriff konform sind
Failed End Condition		Die Suche schlägt fehl oder liefert keine wirklichen Ergebnisse
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte anhand von thematischen Schwerpunkten Projekte anschauen.
DESCRIPTION	Step	Action
	1	Benutzer wählt „Projektwelt“ aus
	2	Benutzer spezifiziert Suchbegriffe
	3	Benutzer inspiziert Suchergebnisse und wählt eines davon zur Einsicht aus.

Tabelle 23: USE CASE #17: Fremdes Projekt einsehen

USE CASE #17		Ein fremdes Projekt einsehen
Goal in Context		Ein fremdes Projekt und seine Details betrachten
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Die Projektinformationen stehen zur Einsicht bereit
Failed End Condition		Das Projekt kann nicht eingesehen werden
Primary, Secondary Actors		Benutzer des Systems
Trigger		Der Benutzer möchte detaillierte Informationen über die Arbeit eines anderen Casemodders erhalten
DESCRIPTION	Step	Action
	1	Benutzer wählt „Projektwelt“ aus
	2	Benutzer wählt aus der Liste ein Projekt aus
SUBVARIATIONS		Branching Action
	2a	Benutzer spezifiziert Suchbegriffe
	3a	Benutzer wählt aus den Suchergebnissen ein Projekt aus
	1b	Sponsor wählt „Sponsoring“ aus
	2b	Sponsor wählt einen Casemodder aus der Liste aus
	3b	Sponsor wählt ein Projekt des Casemodders aus

Tabelle 24: USE CASE #18: Sponsoring-Team erstellen

USE CASE #18		Ein Sponsoring-Team erstellen
Goal in Context		Es soll ein Team aus Casemoddern erstellt werden, die zu einem Sponsor gehören.
Scope & Level		Client, primary task
Preconditions		Der Benutzer verfügt über ein angelegtes Sponsoren-Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Das Team wird erfolgreich erstellt
Failed End Condition		Die Teamerstellung schlägt fehl
Primary, Secondary Actors		Sponsor
Trigger		Der Sponsor möchte ein Team aus Casemoddern zusammenstellen
DESCRIPTION	Step	Action
	1	Sponsor wählt „Sponsoring“ aus
	2	Sponsor wählt Casemodder aus, die er seinem Team hinzufügen will

	3	Sponsor speichert seine Auswahl
SUBVARIATIONS		Branching Action
	2b	Sponsor erstellt das Team aus den vorgeschlagenen Casemoddern
	3b	Sponsor speichert seine Auswahl

Tabelle 25: USE CASE #19: Sponsoring-Team bearbeiten

USE CASE #19		Ein Sponsoring-Team bearbeiten
Goal in Context		Die Mitglieder eines Sponsoring-Teams verwalten
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Sponsoren-Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Der Sponsor kann das Team erfolgreich verwalten
Failed End Condition		Die Änderung des Teams schlägt fehl
Primary, Secondary Actors		Sponsor
Trigger		Der Sponsor möchte die Aufstellung des Teams ändern
DESCRIPTION	Step	Action
	1	Sponsor wählt „Sponsoring“ aus
	2	Sponsor wählt „Mein Team“ aus
	3	Sponsor entfernt oder fügt neue Casemodder hinzu

Tabelle 26: USE CASE #20: Sponsoring-Team löschen

USE CASE #20		Ein Sponsoring-Team löschen
Goal in Context		Ein Sponsoring-Team auflösen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Sponsoren-Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert
Success End Condition		Der Sponsor kann das Team erfolgreich löschen
Failed End Condition		Die Löschung des Teams schlägt fehl
Primary, Secondary Actors		Sponsor
Trigger		Der Sponsor möchte das Team auflösen
DESCRIPTION	Step	Action
	1	Sponsor wählt „Sponsoring“ aus
	2	Sponsor wählt „Mein Team“ aus
	3	Sponsor wählt „Team auflösen“ aus
	4	Sponsor bestätigt den Lösungsdialog

Tabelle 27: USE CASE #21: Projekt löschen

USE CASE #21		Eigenes Projekt löschen
Goal in Context		Ein eigenes Projekt aus dem System löschen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Casemodder-Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert, und es ist mindestens ein Projekt vorhanden
Success End Condition		Der Casemodder kann sein Projekt löschen
Failed End Condition		Die Löschung des Projektes schlägt fehl
Primary, Secondary Actors		Casemodder
Trigger		Der Casemodder möchte sein Projekt aus dem System entfernen
DESCRIPTION	Step	Action
	1	Casemodder wählt „Projektwelt“ aus
	2	Casemodder wählt das zu löschende Projekt aus
	3	Casemodder wählt „Löschen“ aus
	4	Casemodder bestätigt die Löschung

USE CASE #22		Eigenes Projektupdate löschen
Goal in Context		Ein eigenes Projektupdate von einem Projekt entfernen
Scope & Level		Client, secondary task
Preconditions		Der Benutzer verfügt über ein angelegtes Casemodder-Benutzerkonto sowie dessen Zugangsdaten, das Benutzerkonto ist aktiviert, und es ist ein Projekt mit mindestens einem Projektupdate vorhanden
Success End Condition		Der Casemodder kann sein Projektupdate löschen
Failed End Condition		Die Löschung des Projektupdates schlägt fehl
Primary, Secondary Actors		Casemodder
Trigger		Der Casemodder möchte sein Projektupdate aus dem System entfernen
DESCRIPTION	Step	Action
	1	Casemodder wählt „Projektwelt“ aus
	2	Casemodder wählt das Projekt aus, dessen Update gelöscht werden soll
	3	Casemodder beim gewünschten Projektupdate „Löschen“ aus
	4	Casemodder bestätigt die Löschung