Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Entwicklung eines prototypischen Systems zum Auslösen verschiedener Arten von Erklärungsbedarf

Development of a prototypical system for triggering different types of explanation needs

Bachelorarbeit

im Studiengang Informatik

von

Marvin Burchard

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Ann-Christin Klünder
Betreuer: Hannah Deters

Hannover, 15.08.2024

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 15.08.2024

Marvin Burchard

Zusammenfassung

Software-Systeme übernehmen eine zunehmend zentrale und umfassende Rolle in unserem täglichen Leben, was eine kontinuierliche Zunahme ihrer Komplexität zur Folge hat. Diese Entwicklung hebt die wachsende Bedeutung des Forschungsbereichs der Erklärbarkeit von Software-Systemen hervor. Zur Untersuchung des Erklärungsbedarfs von Nutzern sind Systeme erforderlich, die gezielt Erklärungsbedarf hervorrufen. Dabei ist es entscheidend, den Erklärungsbedarf durch realistische Szenarien zu erzeugen, anstatt durch Nutzerfrustration, um valide und unverzerrte Ergebnisse zu erzielen.

Im Rahmen dieser Arbeit wurde ein Software-System entwickelt, das gezielt Erklärungsbedarf in Bezug auf verschiedene Systemaspekte auslöst und dabei ein reales System mit Mehrwert für den Nutzer simuliert, um ein möglichst authentisches und unverzerrtes Szenario zu schaffen. In einer Nutzerstudie wurde das entwickelte System von fünf Teilnehmern getestet. Anschließend wurde durch Interviews ermittelt, ob das System Erklärungsbedarf auslösen konnte und ob diese Auslöseszenarien als negativ für die Benutzerfreundlichkeit wahrgenommen wurden.

Das System hat bei allen Teilnehmern mindestens einmal Erklärungsbedarf hervorgerufen. Besonders in den Bereichen Bedienung, Benutzeroberfläche und Systemverhalten erwies sich das System als effektiv bei der Auslösung von Erklärungsbedarf. Zudem wurden die Auslöseszenarien in den meisten Fällen nicht als negativ im Hinblick auf die Benutzerfreundlichkeit empfunden.

Abstract

Development of a prototypical system for triggering different types of explanation needs

Software systems are taking on an increasingly central and comprehensive role in our daily lives, resulting in a continuous increase in their complexity. This development emphasizes the growing importance of the research area of the explainability of software systems. In order to investigate the explanatory needs of users, systems are required that specifically generate a need for explanation. In order to achieve valid and unbiased results, it is crucial to generate the need for explanation through realistic scenarios rather than through user frustration.

As part of this work, a software system was developed that specifically triggers the need for explanation in relation to various system aspects and simulates a real system with added value for the user in order to create a scenario that is as authentic and unbiased as possible. The developed system was tested by five participants in a user study. Interviews were then conducted to determine whether the system could trigger the need for explanations and whether these trigger scenarios were perceived as having a negative impact on user-friendliness.

The system triggered the need for explanation at least once for all participants. The system proved to be particularly effective in triggering the need for explanation in the areas of operation, user interface and system behavior. In addition, the trigger scenarios were not perceived as negative in terms of user-friendliness in most cases.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Problemstellung . . . 2
- 1.2 Lösungsansatz . . . 2
- 1.3 Struktur der Arbeit . . . 2

2 Grundlagen **3**
- 2.1 Definition Erklärbarkeit . . . 3
  - 2.1.1 Systemverhalten . . . 4
  - 2.1.2 Sicherheit . . . 4
  - 2.1.3 Bedienung . . . 4
  - 2.1.4 Fachwissen . . . 5
  - 2.1.5 Benutzeroberfläche . . . 5
- 2.2 Compose Multiplatform . . . 5
- 2.3 Verwandte Arbeiten . . . 6

3 Anforderungen **7**
- 3.1 Übersicht . . . 7
- 3.2 Anforderungen . . . 7

4 Implementierung **11**
- 4.1 Use Cases . . . 11
- 4.2 Technische Details . . . 27
- 4.3 Wart- und Erweiterbarkeit . . . 28

5 Nutzerstudie **29**
- 5.1 Forschungsfragen . . . 29
- 5.2 Teilnehmer . . . 30
- 5.3 Ablauf . . . 30
- 5.4 Daten . . . 31

6 Ergebnisse und Diskussion **33**
- 6.1 Ergebnisse . . . 33
  - 6.1.1 Erklärungsbedarf . . . 33
  - 6.1.2 Benutzerfreundlichkeit . . . 37

6.2 Diskussion 38
6.2.1 RQ1 38
6.2.2 RQ2 38
6.2.3 RQ3 41
6.2.4 Gesamtziel 42
6.2.5 Limitationen 42

7 Zusammenfassung und Ausblick **45**
7.1 Zusammenfassung 45
7.2 Ausblick 46

A Fragebogen **47**

Kapitel 1

Einleitung

Software-Systeme nehmen immer wichtigere Rollen in unserem Alltag ein, mittlerweile sind sie sogar maßgeblich an kritischen Entscheidungen in der Medizin beteiligt [2]. Mit wachsender Bedeutung steigen auch die Anforderungen an die Software-Systeme. Eine wichtige nicht-funktionale Anforderung ist Erklärbarkeit [5]. Erklärbarkeit versucht unter anderem das Systemverhalten für den Nutzer verständlich zu machen [4]. Vorallem in kritischen Lebensbereichen ist die Erklärbarkeit der Systeme essenziell, da dort potenziell die Gesundheit von Menschen auf dem Spiel steht [1]. Aber auch in allen anderen, nicht kritischen, Bereichen ist Erklärbarkeit ebenfalls von Bedeutung, und es besteht eine Nachfrage seitens der Nutzer nach Erklärungen [5]. Wenn Nutzer eine Softwareentscheidung nicht verstehen können, verlieren sie oft das Vertrauen in die Software. Dadurch wird auch der Nutzen der Software beeinträchtigt, da das Vertrauen in ihre Zuverlässigkeit und Effektivität essentiell für ihre Nutzung ist. Es wäre zum Beispiel empfehlenswert, in einer Fitness-App dem Nutzer ausführliche Erklärungen zu den Fitnesstipps bereitzustellen. Das heißt das Nennen relevanter Daten zur Entscheidungsfindung, sowie eine Erklärung welches Ziel dieser Tipp verfolgt. Fehlen diese Erklärungen, besteht die Gefahr, dass der Nutzer nicht versteht, wie diese Tipps anhand seiner Daten generiert wurden, was möglicherweise zu einem Vertrauensverlust in die Tipps führt. Ein solcher Vertrauensverlust kann dazu führen, dass der Nutzer die Effektivität der App in Frage stellt und möglicherweise sogar die Nutzung der App einstellt. Neben Erklärungsbedarf zum Systemverhalten existieren vielfätige weitere Aspekte zu denen Nutzende Erklärungen brauchen, dazu zählen unter anderem: Bedienung und Sicherheit [8]. Wann und wodurch Nutzende Erklärungsbedarf haben ist Teil des Forschungsgebiets der Erklärbarkeit und Thema dieser Arbeit.

1.1 Problemstellung

Wie bereits in der Einleitung erwähnt, können unzureichende Erklärungen die Benutzer während der Nutzung von Software beeinträchtigen oder sie sogar vollständig abschrecken. In früheren Studien wurden verschiedene Arten von Erklärungsbedarf identifiziert [8]. Ziel dieser Arbeit ist die Konzeption und Implementierung eines Softwaresystems, das gezielt unterschiedliche Arten von Erklärungsbedarf bei den Nutzern hervorruft. Dabei wird ein realer Anwendungsfall simuliert, um authentische Nutzungsszenarien abzubilden. Ein besonderes Augenmerk liegt darauf, trotz der intendierten Generierung von Erklärungsbedarf eine hohe Benutzerfreundlichkeit des Systems sicherzustellen. Dies dient der Gewährleistung einer nahtlosen und effektiven Nutzung, die Frustration bei den Nutzern vermeidet und somit den Erklärungsbedarf nicht ungewollt beeinflusst. Die Simulation eines realen Anwendungsfalls ist entscheidend, um die Erzeugung von Erklärungsbedarf auf realitätsnahen Interaktionen basieren zu lassen und potenzielle Verzerrungen durch hypothetische Szenarien zu vermeiden.

1.2 Lösungsansatz

Um ein System, wie in 1.1 beschrieben, zu entwickeln, muss zunächst ein geeigneter Anwendungsfall identifiziert werden. In diesem Zusammenhang sind spezifische Szenarien zu entwerfen, die den Erklärungsbedarf gezielt auslösen. Da in diesem Entwicklungsprozess keine klaren Anforderungen im Voraus festgelegt werden müssen, wird auf eine vorgelagerte Studie verzichtet. Eine solche Studie würde zudem nur begrenzt verwertbare Ergebnisse hinsichtlich potenzieller Auslöseszenarien für den Erklärungsbedarf liefern. Stattdessen werden die Anforderungen sowie die Auslöseszenarien eigenständig definiert und konstruiert, um anschließend im Rahmen einer Nutzerstudie zu validieren, ob das entwickelte System den Erklärungsbedarf effektiv auslöst und die formulierten Anforderungen erfüllt.

1.3 Struktur der Arbeit

Diese Arbeit ist wie folgt strukturiert. In Kapitel 2 werden die für diese Arbeit notwendigen theoretischen Grundlagen erläutert. Danach werden in Kapitel 3 alle Anforderungen an das zu entwickelte System definiert. In Kapitel 4 werden alle zur Implementierung relevanten Details genannt und alle implementierten Use Cases dargestellt. Drauf folgt Kapitel 5 in welchem die Ziele und der Ablauf der durchgeführten Nutzerstudie dargelegt werden. In Kapitel 6 werden im ersten Schritt die Ergebnisse der Nutzerstudie aufgezeigt und im zweiten Schritt dann Diskutiert. Am Schluss folgt das Kapitel 7 in dem die Arbeit nochmal zusammengefasst wird.

Kapitel 2

Grundlagen

2.1 Definition Erklärbarkeit

Um den Begriff der Erklärbarkeit zu definieren, ist es zunächst erforderlich, ein Verständnis für den Begriff der Erklärung zu entwickeln. In der Literatur finden sich zahlreiche Definitionen von Erklärungen. Grob lassen sie sich in zwei Gruppen einteilen: Einerseits gibt es den pragmatischen Ansatz, der eine Erklärung als „Antwort auf eine Frage“ versteht [3],[15]. Andererseits begegnen wir technischeren Ansätzen, die an Ursachen gebunden sind [15],[12]. Das heißt, Halpern et al. [12] definiert eine Erklärung als eine Tatsache, die als Ursache für das erklärende Phänomen betrachtet wird. Wenn die Erklärung sich als wahr erweist, wird sie als die tatsächliche Ursache für das erklärte Phänomen betrachtet, unabhängig von der anfänglichen Unsicherheit des Erklärenden. Kurz gesagt, die Erklärung wird als die wahre Ursache für das Phänomen betrachtet, das sie erklärt. In dieser Arbeit reicht aber die Definition einer Erklärung als Antwort auf eine Frage aus.
Als grundlegende Definition von Erklärbarkeit wird im Rahmen dieser Arbeit die Definition von Chazette et al. [4] verwendet, welche den Begriff aus Sicht des Software-Engineering so definiert haben:

> A system S is explainable with respect to an aspect X of S relative to an addressee A in context C if and only if there is an entity E (the explainer) who, by giving a corpus of information I (the explanation of X), enables A to understand X of S in C.

Dies ist eine sehr generalisierte Definition und lässt sich daher auf verschiedene Aspekte X anwenden. Chazette et al. [4] konnten mithilfe einer systematischen Literaturrecherche drei Hauptkategorien solcher Aspekte identifizieren. Diese waren: Systemverhalten, Sicherheit und Bedienung. Droste et al. [8] führten eine Umfrage durch und identifizierten zwei weitere Kategorien: Benutzeroberfläche und Fachwissen. Daher wird in dieser Arbeit

angenommen, dass diese fünf Kategorien die Grundstrukturen sind, in die alle Aspekte der Erklärbarkeit eingeordnet werden können.
In den nächsten Unterkapiteln werden alle fünf Kategorien näher erläutert.

2.1.1 Systemverhalten

Im Fokus des Aspekts des Systemverhaltens stehen drei zentrale Fragen: Was ist passiert? Wie ist es passiert? Und warum ist es passiert [11]? Dieser Aspekt gewinnt insbesondere bei komplexen Algorithmen oder KI-Anwendungen an Bedeutung, aber auch bei Fehlern und Systemabstürzen [8]. Ein anschauliches Beispiel ist die Fitness-App aus der Einleitung: Der Nutzer kann nicht nachvollziehen, wie der Algorithmus aus den Eingaben genau diesen Fitnesstipp generiert. Zusätzlich fallen unter den Aspekt des Systemverhaltens auch Systemabstürze oder unerwartete Fehler. Solche Abstürze können Frustration und Ratlosigkeit hervorrufen, wobei häufig die Nutzer selbst die Verursacher sind. Die Bereitstellung von Erklärungen zu den Ursachen solcher Abstürze könnte dazu beitragen, die Gründe besser zu verstehen und an Lösungen zu arbeiten [8]. Gewährleistung eines transparenten Systemverhaltens könnte nicht nur zur Benutzerfreundlichkeit beitragen, sondern könnte auch das Vertrauen der Anwender in die Zuverlässigkeit und Funktionsweise der Anwendung oder des Algorithmus stärken.

2.1.2 Sicherheit

Mit wachsender Bedeutung von Software wird die nicht-funktionale Anforderung der Pritvatsphäre und der Sicherheit immer wichtiger. Nutzende sehen ihre persöhnlichen Daten in Gefahr und wollen diese schützen [17]. Da die verbindlichen Datenschutzerklärungen in den meisten Fällen viel zu lang sind, und sie daher kaum gelesen werden, verbreiten sich viele Narrative unter den Nutzenden [17]. Oftmals basieren diese jedoch auf Vermutungen und nicht auf Fakten. Auch die Arbeit von Chazette et al. [8] bestätigt dies. Sie konnten eine Nachfrage an Erklärungen zu Privatsphäre und Sicherheitsaspekten feststellen. Deshalb könnte es hilfreich sein, zusätzlich zur Datenschutzerklärung, dem Nutzer die Möglichkeit zu geben in einzelnen Situationen Erklärungen zur Datenverarbeitung oder Ähnlichem anzufordern, um Narrative und Zweifel entgegezuwirken.

2.1.3 Bedienung

Der Aspekt der Bedienung kann als Gegenstück zu dem Aspekt des Systemverhaltens gesehen werden [11]. Im Gegensatz zum Aspekt des Systemverhaltens geht es hier nicht darum, was passiert ist, sondern darum, dass der Nutzer Fragen dazu hat, wie er das System dazu bringen kann, das zu tun, was er möchte. Unter diesen Aspekt fallen Fragen zur Navigation im System, also wie Benutzer sich innerhalb eines Systems bewegen

können, zum Beispiel durch Menüs, Schaltflächen oder Suchfunktionen, um bestimmte Funktionen oder Inhalte zu finden [8]. Auch Fragen zur Ausführung bestimmter Operationen sind inbegriffen, das heißt Anfragen oder Schwierigkeiten bei der Durchführung spezifischer Aktionen innerhalb des Systems, wie das Hinzufügen von Daten, das Löschen von Inhalten oder das Anpassen von Einstellungen [8]. Zusätzlich bezieht sich dieser Aspekt auf die Nachfrage nach einer Einführung oder Anleitung, besonders für Benutzer, die möglicherweise neu oder unerfahren sind und zusätzliche Hilfe benötigen, um das System effektiv nutzen zu können. [8]. Erklärungen dafür zu liefern könnte deutlich die Erlernbarkeit und Nutzbarkeit der Software steigern [8].

2.1.4 Fachwissen

Software-Systeme können mittlerweile in jedem Bereich ihren Einsatz finden. Daher kann es selbst für erfahrene Softwarenutzer zu Problemen führen, wenn ihnen Fachwissen aus der Domäne fehlt [8]. Zum Beispiel kann es zu Schwierigkeiten kommen, falls dem Nutzer bei dem Verwenden von Buchhaltungssoftware ein Fachbegriff nicht klar ist, weil er keine Kenntnisse von Buchhaltungskonzepten hat. Erklärungen zu domänenspezifischem Wissen liefern, sodass der Nutzer möglichst wenig Wissen selbst mitbringen muss, kann das Verstehen der Software fördern und somit auch die Benutzbarkeit verbessern [8].

2.1.5 Benutzeroberfläche

Der fünfte und letzte Aspekt der Erklärbarkeit bezieht sich auf Fragen zu Designentscheidungen innerhalb der Benutzeroberfläche. Nutzer können sich fragen, warum sich ein bestimmter Knopf an einer bestimmten Stelle befindet oder warum eine Schaltfläche plötzlich eine andere Farbe hat. Zusätzlich können Aktualisierungen der Benutzeroberfläche zu Verwirrung führen, da Nutzer möglicherweise die erlernte Bedienung nicht mehr wie gewohnt nutzen können [8]. Erklärungen zu Designentscheidungen können daher auch die Benutzerfreundlichkeit und das Verständnis fördern.

2.2 Compose Multiplatform

Für die Implementierung des Systems wird das Framework "Compose Multiplatform" von JetBrains eingesetzt. Dieses Framework basiert auf der Programmiersprache Kotlin und ermöglicht die effiziente Entwicklung von Benutzeroberflächen für mehrere Plattformen gleichzeitig [14]. Dabei verfolgt "Compose Multiplatform" einen deklarativen Ansatz [14]. Die Benutzeroberflächen werden durch sogenannte "Composables" implementiert. Ein Composable repräsentiert ein in sich geschlossenes UI-Element, das über einen eigenen, vom restlichen Interface unabhängigen Zustand verfügt [14].

Somit setzt sich die gesamte Benutzeroberfläche aus einer Vielzahl von Composables zusammen.

Dieser Ansatz bietet zahlreiche Vorteile. Zum einen können einzelne Composables systemweit wiederverwendet werden, was zur Vermeidung von Redundanzen beiträgt. Zum anderen fördert die Architektur die Modularität, da jedes Composable unabhängig vom restlichen System definiert wird und seinen eigenen Zustand besitzt.

2.3 Verwandte Arbeiten

Die Präsentation einer übermäßigen Anzahl von Erklärungen in Software hat nachteilige Auswirkungen auf die Nutzer, einschließlich einer erhöhten kognitiven Belastung [5],[18]. In diesem Zusammenhang führten Deters et al. [6] eine Studie zu bedarfsgerechten Erklärungen durch und stellten fest, dass diese von den Nutzern akzeptiert werden und gleichzeitig den Erklärungsbedarf präzise identifizieren können.

In ihrer Masterarbeit implementierte Ekaputeri [10] triggerbasierte Erklärungen. Zu diesem Zweck entwickelte sie ein System mit absichtlich schlechter Benutzerfreundlichkeit, um den Erklärungsbedarf zu aktivieren. Es wurde jedoch festgestellt, dass das Auslösen von Erklärungsbedarf durch schlechte Benutzerfreundlichkeit die Gültigkeit der Ergebnisse bedroht, da die Nutzer verwirrt und frustriert waren. Diese Methode eignet sich daher nicht, um realen Erklärungsbedarf darzustellen, da sie keinen realen Anwendungsfall wiederspiegelt.

Ein häufiges Problem bei Studien zum Erklärungsbedarf ist das Auftreten von Verzerrungen [7]. Ein bedeutender Aspekt ist der sogenannte "hypothetical bias", der entsteht, wenn Studienteilnehmer in hypothetische Situationen versetzt werden und sich dadurch unnatürlich verhalten. Beispielsweise entwickelte Fechner [11] einen Mikroblogging-Dienst, bei dem die Teilnehmer sich vorstellen mussten, einen Beitrag zu veröffentlichen. Da die Teilnehmer dazu aufgefordert wurden und keinen echten Mehrwert oder Eigenantrieb hatten, konnten solche Verhaltensverzerrungen auftreten. In anderen Studien, wie die von Deters et al. [7] bekamen die Teilnehmer eine Schritt für Schritt Anleitung für die Studie. Auch dies unterscheidet sich von einer realen Nutzung eines Systems und bringt einen "hypothetical bias" mit sich.

Um zukünftig die Erklärbarkeit und den Erklärungsbedarf der Nutzer optimal erforschen zu können, ist ein prototypisches System für Studien erforderlich, das alle bisher gewonnenen Erkenntnisse berücksichtigt. Meine Arbeit zielt darauf ab, ein solches System zu entwickeln, das einen realen Anwendungsfall ohne "hypothetical bias" simuliert und dabei den Erklärungsbedarf auslöst, ohne die Benutzerfreundlichkeit zu beeinträchtigen.

Kapitel 3

Anforderungen

3.1 Übersicht

Um zielgerichtet Software zu entwickeln, ist es essenziell vorher Anforderungen an diese zu erheben [9]. In den nächsten Unterkapiteln werden alle Anforderungen an das in dieser Arbeit zu entwickelnde System dargestellt und näher erläutert, sowie daraus resultierende Use Cases des Systems aufgezeigt.

Als Definition einer Anforderung an Software wird folgende verwendet: Anforderungen an Software definiert Ebert [9] mit drei Aspekten, von denen mindestens einer erfüllt sein muss: 1. Eigenschaft oder Bedingung, die von einem Benutzer zur Lösung eines Problems benötigt wird. 2. Eigenschaft oder Bedingung, die ein System erfüllen muss, um einen Vertrag oder eine Norm zu erfüllen. 3. Eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung, wie in den ersten beiden Punkten beschrieben.

3.2 Anforderungen

Wie eingangs erläutert ist das Ziel dieser Arbeit ein System zu entwickeln, welches im Rahmen eines echten Nutzungsszenarios Erklärungsbedarf bei Nutzenden hervorruft. Aufgrund dessen gab es vier nicht-funktionale Rahmenbedingungen die das System erfüllen sollte, um das Ziel zu erreichen:

NR1: Das System soll Erklärungsbedarf bei Nutzern auslösen.

NR2: Das System soll reale Use Cases simulieren.

NR3: Das System soll eine gute Nutzbarkeit aufweisen.

NR4: Das System soll einfach wartbar und erweiterbar sein.

Die primäre Zielvorgabe für das System war [NR1]. Es wurde angestrebt, dass das System in verschiedenen Anwendungsszenarien alle fünf Kategorien von Erklärungsbedarf bei den Nutzern auslöst. Dabei galt es jedoch zu berücksichtigen, dass [NR2] und [NR3] gleichermaßen umgesetzt werden sollten, um eine realitätsnahe Anwendung zu simulieren. Es war besonders darauf zu achten, dass die Benutzerfreundlichkeit des Systems nicht übermäßig eingeschränkt wird, nur um Erklärungsbedarf zu erzeugen. Die Auslöseszenarien sollten zwar neu und nicht bereits bekannt sein, jedoch dürfen sie nicht zu Frustration und Verwirrung führen. Mit [NR2] ist also konkret gemeint, dass die Nutzer sich auf natürliche Weise durch die Anwendung bewegen können, ohne Schritt-für-Schritt-Anleitungen befolgen zu müssen. Diese natürliche Nutzung ermöglicht es den Nutzern, die Software intuitiv zu erforschen und mit ihr zu interagieren, wie sie es auch in einer realen Umgebung tun würden. [NR3] zielt darauf ab, dass das System gut bedienbar ist, der Nutzer vor Fehlbedienung geschützt ist und eine ästetische Benutzeroberfläche aufweist. Ergänzend zu den genannten Anforderungen ist [NR4] von großer Bedeutung. Die Wartbarkeit des Systems spielt eine zentrale Rolle, da es für zukünftige Studienzwecke schnell und einfach angepasst werden können soll, um den sich wandelnden Anforderungen gerecht zu werden und eine langfristige Nutzung zu ermöglichen. Die ISO 25010 [13] definiert wesentliche Qualitätsmerkmale von Software, einschließlich der Wartbarkeit. Zu den für dieses System relevanten Kriterien zählen:

- Modularität: Das System sollte so strukturiert sein, dass es in voneinander unabhängige Module unterteilt werden kann.
- Wiederverwendbarkeit: Einzelne Komponenten des Systems sollten in anderen Kontexten erneut verwendet werden können.
- Modifizierbarkeit: Änderungen am System sollten problemlos und ohne die Einführung von Fehlern möglich sein.

Diese Merkmale tragen maßgeblich zur langfristigen Wartbarkeit und Weiterentwicklung von der Software bei.

Des Weiteren gab es zwei funktionale Anforderungen an das System:

R1: Das System soll es Nutzern ermöglichen jederzeit Erklärungen zum aktuellen Use Case abrufen zu können.

R2: Ein Entwickler soll [R1] deaktivieren können.

Um Anforderung [R1] zu erfüllen, sollte das System den Benutzern die Möglichkeit bieten, jederzeit Erklärungen zu verschiedenen Arten von Erklärungsbedarf und zum aktuellen Anwendungsfall abzurufen. Dadurch kann festgestellt werden, wann und warum Benutzer in verschiedenen Anwendungsfällen Erklärungen benötigen. Mit dieser Funktionalität ist es

einfacher zu bestimmen, wann Benutzer tatsächlich Erklärungen benötigen und somit das Hauptziel der Anwendung zu überprüfen. Darüber hinaus gemäß Anforderung [R2], sollte ein Entwickler die Möglichkeit bekommen, schnell und einfach die Funktion zur Abfrage von Erklärungen [R1] nach Bedarf zu deaktivieren. Dies trägt zur Anpassungsfähigkeit der Software an potenzielle zukünftige Studien bei.

Kapitel 4

Implementierung

4.1 Use Cases

Als Grundlage für die Tabellen sowie für das Diagramm wurde das UML-Poster aus 2018, welches auf der Internetseite des Instituts zu finden ist [19], verwendet. Der Übersichtlichkeit halber habe ich die Zeile Erklärungsbedarf hinzugefügt, um direkt sehen zu können welcher Use Case welche Art von Erklärungsbedarf hervorrufen soll. Da es, wie beschrieben, nur wenige funktionale Anforderungen an das zu entwickelnde System gab, boten sich zahlreiche Möglichkeiten für Use Cases. Letzendlich kam am Anfang des Entwicklungsprozesses die Idee eines Semesterplaners auf. Ein Semesterplaner bietet ein reales Anwendungsszenario und bietet die Möglichkeit auf verschiedene Art und Weisen Erklärungsbedarf auszulösen. Zum Beispiel kann man für die Generierung von Vorschlägen eine KI simulieren oder mit verschiedenen Eingabeinteraktionstechniken arbeiten. Auf dieser Idee basierend wurden dann von mir Use Cases entwickelt, die die oben beschriebenen Anforderungen erfüllen sollen. Im Folgenden werden die einzelnen Use Cases genauer erläutert.

Use Case Diagramm

Um die Komplexität der Use Cases zu verringern und deren Beziehungen übersichtlich darzustellen, folgt nun ein umfassendes Use-Case-Diagramm des gesamten Systems in Abbildung 4.1. Dieses Diagramm illustriert nicht nur, wie die Studierenden mit dem System interagieren, sondern modularisiert das System auch, sodass man einen guten Gesamtüberblick über die Funktionalitäten bekommt. Dies erleichtert auch die zukünftige Weiterentwicklung erheblich, da die Struktur und die Abhängigkeiten der Systemkomponenten klar erkennbar sind.

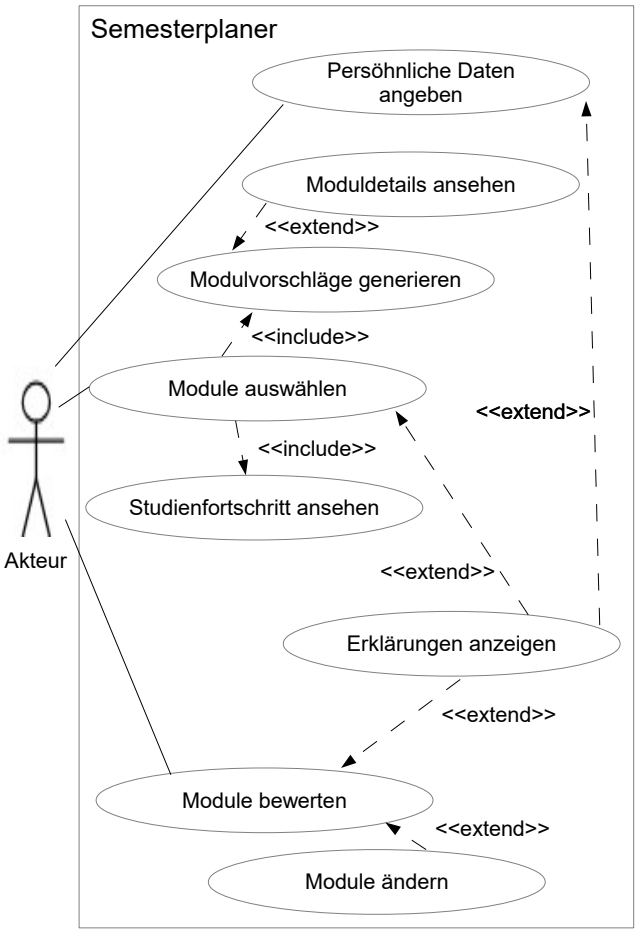

Abbildung 4.1: Use Case Diagramm des Systems

Use Case 1

Use Case 1	**Persöhnliche Daten angeben**
Erläuterung	Nachdem der Nutzer das Programm gestartet hat, wird ihm zuerst eine Oberfläche angezeigt, in der er seinen vollen Namen und sein Geburtsdatum eingeben muss.
Ebene	Hauptfunktion
Vorbedingung	Das Programm wurde erfolgreich gestartet.
Erfolgsgarantie	Es wird eine Benutzeroberfläche angezeigt in der der Nutzer seine Daten eintragen kann.
Hauptakteur	Student
Auslöser	Das Programm wird erfolgreich gestartet.
Hauptszenario	1. Der Student startet Programm. 2. Das System zeigt Oberfläche mit Eingabefeldern an. 3. Der Student gibt Vornamen, Nachnamen, Geburtsdatum und Matrikelnummer ein.
Erweiterungen	3a: WENN Daten unvollständig oder nicht korrekt formatiert, DANN deaktiviere 'Weiter' Knopf.
Erklärungsbedarf	Sicherheit, Bedienung
Verwendungshäufigkeit	Immer

Use Case 1 (UC1) wird unmittelbar nach dem erfolgreichen Programmstart ausgelöst. Das System fordert den Benutzer auf, persönliche Daten wie den vollständigen Namen, das Geburtsdatum und die Matrikelnummer einzugeben. Hier zeigt sich bereits eine Vorbedingung für die Systemnutzer beziehungsweise für die späteren Studienteilnehmer: Die Nutzer müssen Informatikstudenten an der Leibniz Universität Hannover (LUH) sein. Diese Einschränkung wird einerseits durch die Anforderung einer Matrikelnummer und andererseits durch die spätere Spezialisierung des Systems auf Informatikstudenten der LUH gesetzt. Diese Eingaben sind für den weiteren Programmverlauf zwar irrelevant und werden nicht gespeichert, sie sollen jedoch gleich zwei Arten von Erklärungsbedarf auslösen. Erstens ist nicht angegeben, was mit den eingegebenen Daten passiert oder wofür sie benötigt werden, was Bedenken hinsichtlich der *Datensicherheit* aufwerfen könnte. Zweitens wird das Geburtsdatum im Format 'TT.MM.JJJJ' gefordert. Andernfalls kann der Benutzer nicht weiter navigieren. Diese Anforderung soll zu Erklärungsbedarf hinsichtlich

der *Bedienung* führen. Das Eingabeformat wird aber als Platzhalter angezeigt, um die Benutzerfreundlichkeit nicht negativ zu beeinflussen, da der Erklärungsbedarf nicht durch Frustration ausgelöst werden soll. Ohne Hinweis auf das Format der Eingabe, könnte der Nutzer nur raten welches Format gefordert wird, oder eine Erklärung dafür einfordern. Dies wäre schlecht für die Benutzerfreundlichkeit. Trotzdem besteht die Möglichkeit, dass er diesen Hinweis übersieht und trotzdem Erklärungsbedarf hat. In Abbildung 4.2 ist die finale Implementierung zu sehen. Neben einer

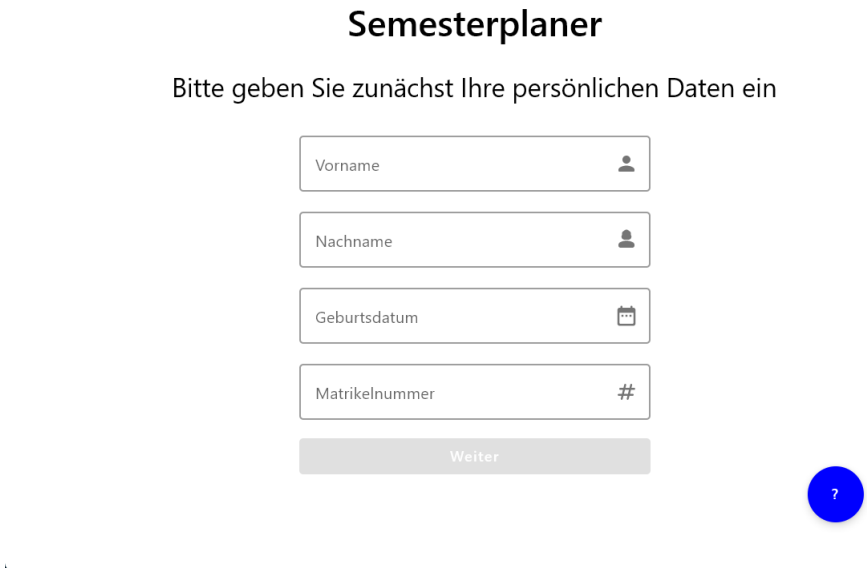

Abbildung 4.2: Implementierung Use Case 1

Überschrift und einer kurzen Beschreibung gibt es vier Eingabefelder für die persöhnlichen Daten. Unter den Eingabefeldern gibt es einen Knopf, um die Eingaben zu bestätigen und weiter zu navigieren. Dieser Knopf ist erst dann aktiviert, wenn alle Eingabefelder (im richtigen Format) ausgefüllt sind.

Use Case 2

Use Case 2	**Module auswählen**
Erläuterung	Das System zeigt eine Liste von Modulen an aus welcher der Nutzer von ihm bereits absolvierte Module auswählen soll.
Ebene	Hauptfunktion
Vorbedingung	Das System hat erfolgreich die Module aus einer Csv-Datei ausgelesen.
Erfolgsgarantie	Es wird eine Liste aller Module angezeigt.
Hauptakteur	Student
Auslöser	Der Student hat gültige persöhnliche Daten eingegeben und auf 'Weiter' gedrückt.
Hauptszenario	1. Der Der Student gibt gültige persöhnliche Daten ein und drückt auf 'Weiter'. 2. Das System zeigt Module. 3. Der Student wählt von ihm absolvierte Module aus.
Erweiterungen	2a: WENN Student weniger als drei Module ausgewählt, DANN deaktiviere 'Weiter' Knopf.
Erklärungsbedarf	Bedienung, Benutzeroberfläche
Verwendungshäufigkeit	Immer

Nach erfolgreicher Eingabe der persönlichen Daten wird der Benutzer zu UC2 weitergeleitet. Dort bekommt der Student eine Liste aller Module angezeigt und soll diejenigen auswählen, die er bereits absolviert hat. Die Auswahl der Module erfolgt durch ein Wischen nach rechts. Diese unkonventionelle Interaktionstechnik wurde bewusst gewählt, um gezielt Erklärungsbedarf hinsichtlich der *Bedienung* zu schaffen. Außerdem sind die Module je nach ihren Leistungspunkten mit verschiedenfarbigen Rahmen versehen, was Erklärungsbedarf bezüglich der *Benutzeroberfläche* hervorrufen soll. Nachdem der Student alle Module, die er bereits absolviert hat, ausgewählt hat, kann er zum nächsten Use Case weitergehen. Dabei ist zu beachten, dass mindestens drei Module ausgewählt werden müssen, um fortzufahren, da im nächsten Use Case drei Module bewertet werden müssen. Wenn der Student sich noch am Anfang seines Studiums befindet und noch keine drei Module abgeschlossen hat, scheidet er somit als Benutzer und Studienteilnehmer aus.

In Abbildung 4.3 ist die Umsetzung dieses Use Cases zu sehen. Es gibt eine Liste für alle verfügbaren Module und eine Liste für die vom

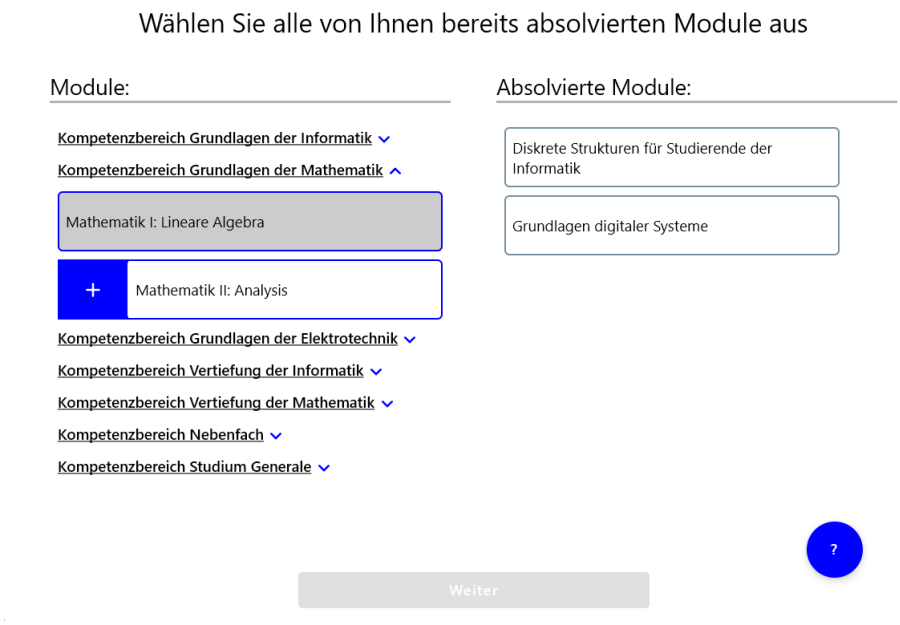

Abbildung 4.3: Implementierung Use Case 2

Studenten bereits absolvierten Module, welche zunächst leer ist. Die Liste aller Module ist nach Kompetenzbereich sortiert. Jeder Kopetenzbereich kann ein und ausgeklappt werden, um eine möglichst gute Übersicht zu gewährleisten. Um ein Modul in die Liste der absolvierten Module zu verschieben muss das jeweilige Modul nach rechts gewischt werden und dann auf das Pluszeichen gedrückt werden. Durch anklicken eines Modules wird dieses farblich markiert. Bei Verschieben eines Modules werden auch alle farblich markierten Module mit verschoben. So ist es möglich mehrere Module gleichzeitig zu verschieben.

Use Case 3 und 4

Use Case 3	**Module bewerten**
Erläuterung	Das System sucht drei zufällige Module aus, welche der Student absolviert hat, und fordert ihn auf diese zu bewerten.
Ebene	Hauptfunktion
Vorbedingung	Der Student hat mindestens drei Module ausgewählt, die er absolviert hat.
Erfolgsgarantie	Es werden nacheinander drei Oberflächen zum Berwerten von jeweils einem Modul angezeigt.
Hauptakteur	Student
Auslöser	Der Student hat mindestens drei Module ausgewählt und auf 'Weiter' gedrückt.
Hauptszenario	1. Der Student wählt mindestens drei Module aus und drückt auf 'Weiter'. 2. Das System sucht drei zufällige Module aus, die der Student absolviert hat. 3. Der Student bewertet nacheinander jedes Modul.
Erweiterungen	2a: WENN Student zu kurzen Kommentar geschrieben hat, DANN deaktiviere 'Weiter' Knopf.
Erklärungsbedarf	Bedienung, Fachwissen, Sicherheit
Verwendungshäufigkeit	Immer

Use Case 4	**Module ändern**
Erläuterung	Der Student hat ein falsches Modul eingegeben oder ein Modul vergessen und kann seine Eingabe korrigieren.
Ebene	Nebenfunktion
Vorbedingung	Der Student hat noch kein Modul bewertet.
Erfolgsgarantie	Der Student gelangt zur Liste aller Module zurück.
Hauptakteur	Student
Auslöser	Der Student hat auf 'Zurück' gedrückt.
Hauptszenario	1. Der Student drückt auf 'Zurück'. 2. Das System navigiert zur Modulauswahl zurück.
Erweiterungen	1a: WENN Student bereits ein Modul kommentiert hat, DANN zeige den 'Zurück' Knopf nicht mehr an,
Erklärungsbedarf	-
Verwendungshäufigkeit	Selten

Nachdem der Student erfolgreich mindestens drei Module ausgewählt hat, wird er zu UC3 weitergeleitet. In diesem UC wird er aufgefordert drei zufällig aus seinen absolvierten Modulen ausgewählte Module zu bewerten. Es gibt drei Bewertungskategorien: Organisation, Dozent und Kohärenz der Konzepte. Jede Kategorie kann mit bis zu fünf Sternen bewertet werden. Die Kategorien "Dozent" und "Kohärenz der Konzepte" sind dafür vorgesehen Erklärungsbedarf hinsichtlich des *Fachwissens* auslösen, da die Begriffe "Dozent" und "Kohärenz" dem Studenten nicht bekannt sein könnten. Zusätzlich zu den Sternebewertungen gibt es ein Freitextfeld, in dem der Student aufgefordert wird, aufzuschreiben, was ihm besonders positiv oder negativ an dem Modul in Erinnerung geblieben ist. Es müssen mindestens 90 Zeichen in das Freitextfeld eingegeben werden, um fortfahren zu können. Andernfalls kann der Student nicht weiter navigieren. Diese Anforderung soll Erklärungsbedarf bezüglich der *Bedienung* hervorrufen. Es wird aber durch eine Anzeige unten rechts auf die Mindestlänge hingewiesen, um die Benutzerfreundlichkeit nicht zu gefährden. Trotzdem könnte es sein, dass diese Anzeige übersehen wird und Erklärungsbedarf ausgelöst wird. Außerdem soll dieser UC Erklärungsbedarf zur *Sicherheit*, genauer zur Privatsphäre, hervorrufen, da nicht darüber informiert wird, was mit den Eingaben passiert. Zusätzlich besteht vor Beginn der Bewertung mit UC4 die Möglichkeit, zum vorherigen UC zurückzukehren, um frühere Eingaben zu korrigieren. Nachdem der Student jedoch ein Modul vollständig

bewertet hat, ist ein Zurückkehren zum vorherigen UC nicht mehr möglich. Andernfalls könnte das bewertete Modul aus den absolvierten Modulen entfernt werden, was die vorherige Bewertung inhaltlich bedeutungslos machen würde. Hier soll kein Erklärungsbedarf ausgelöst werden.

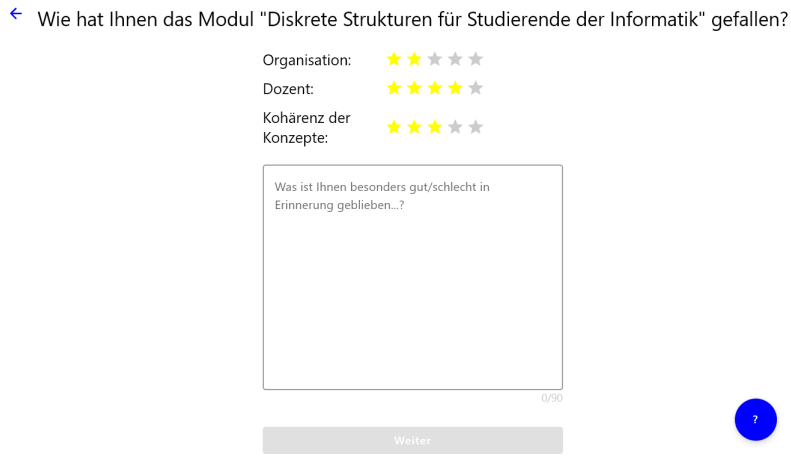

Abbildung 4.4: Implementierung Use Case 3 und 4

In Abbildung 4.4 wird die Implementierung beider Use Cases dargestellt. UC4 ist als Schaltfläche in der oberen linken Ecke des Bildschirms realisiert. Durch Betätigen dieser Schaltfläche gelangt der Benutzer zum vorherigen Bildschirm zurück, um die Auswahl seiner Module zu modifizieren. Zentral im Bild ist die Umsetzung von UC3 zu sehen. Im oberen Bereich wird der Name des zu bewertenden Moduls angezeigt, begleitet von der Frage: "Wie hat Ihnen das Modul 'XY' gefallen?" Darunter befinden sich drei Bewertungsskalen, jeweils mit einer Fünf-Sterne-Einstufung. Anschließend folgt ein großes Freitextfeld, in dem der Benutzer seine Meinung zum entsprechenden Modul festhalten soll. Unterhalb des Freitextfelds, in der unteren rechten Ecke, wird eine Anzeige integriert, die sowohl die aktuelle Zeichenanzahl als auch die Mindestanzahl der erforderlichen Zeichen darstellt. Direkt darunter befindet sich der 'Weiter'-Button, der erst aktiviert wird, wenn mindestens 90 Zeichen in das Freitextfeld eingegeben wurden. Durch das Betätigen dieses Buttons wird ein weiteres Modul zur Bewertung angezeigt. Nach der erfolgreichen Bewertung von drei Modulen führt das erneute Drücken des 'Weiter'-Buttons den Benutzer zum nächsten Bildschirm.

Use Case 5

Use Case 5	**Modulvorschläge generieren**
Erläuterung	Das System zeigt Vorschläge, welche Module der Student in Zukunft noch belegen könnte.
Ebene	Hauptfunktion
Vorbedingung	Der Student hat nicht alle Module als absolviert angegeben.
Erfolgsgarantie	Das System zeigt fünf verschiedene Modulvorschläge an.
Hauptakteur	Student
Auslöser	Der Student hat drei Module bewertet und dann auf 'Weiter' gedrückt, sowie im Bottommenü 'Vorschläge' ausgewählt.
Hauptszenario	1. Der Student bewertet drei Module drückt dann auf 'Weiter' und wählt dann 'Vorschläge' im Bottommenü aus. 2. Das System sucht vier zufällige Module aus den Modulen, die der Student noch nicht absolviert hat, und ein völlig fachfremdes Modul aus. 3. Das System zeigt eine Liste der ausgesuchten Module an.
Erweiterungen	-
Erklärungsbedarf	Systemverhalten
Verwendungshäufigkeit	sehr häufig

Nachdem der Student auch diese Eingaben erfolgreich gemacht und weiter navigiert hat, generiert das System fünf Modulvorschläge. Diese Vorschläge basieren jedoch nicht auf den bisherigen Eingaben des Studenten, sondern werden zufällig aus den von ihm noch nicht absolvierten Modulen ausgewählt. Zusätzlich wird das Modul 'Tier- und Humanphysiologie' vorgeschlagen, obwohl es fachlich nicht zu den Informatikmodulen passt. Dies soll absichtlich Erklärungsbedarf hinsichtlich des *Systemverhaltens* beim Studenten auslösen. Auch die zufällig ausgewählten Modulvorschläge könnten je nach Fall diesen Erklärungsbedarf schon hervorrufen, da sie wahrscheinlich nicht auf den Präferenzen oder bisherigen Angaben des Studenten basieren.

In Abbildung 4.5 sieht man die Implementierung von UC5. Es werden fünf Karten mit vorgeschlagenen Modulen angezeigt. Jede Karte kann angeklickt werden, um mehr Informationen zu erhalten. Unten sieht man

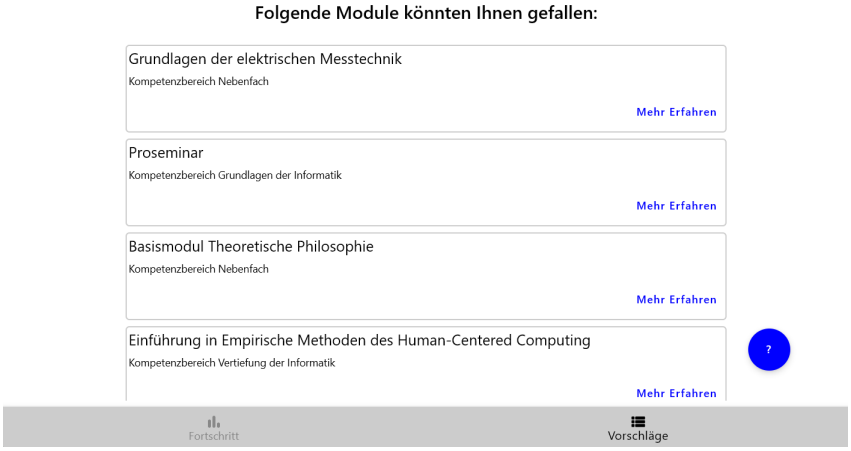

Abbildung 4.5: Implementierung Use Case 5

ein Navigationsmenü, sodass der Nutzer frei zwischen diesem und dem UC7 navigieren kann.

Use Case 6

Use Case 6	**Moduldetails und Kommentare ansehen**
Erläuterung	Der Student kann sich Detailinfos sowie Kommentare zu jeweils einem Modul ansehen.
Ebene	Nebenfunktion
Vorbedingung	Das System hat Modulvorschläge erfolgreich generiert.
Erfolgsgarantie	Der Student bekommt auf Abruf mehr Informationen zu dem jeweiligen Modul sowie Kommentare zum Modul gezeigt.
Hauptakteur	Student
Auslöser	Student drückt auf 'Mehr erfahren'.
Hauptszenario	1. Der Student drückt auf 'Mehr erfahren'. 2. Das System zeigt Vorschläge. 3. Der Student liest Titel und kann bei Bedarf auf 'Mehr erfahren' drücken, um weitere Informationen zu bekommen.
Erweiterungen	-
Erklärungsbedarf	-
Verwendungshäufigkeit	regelmäßig

Der Student hat die Möglichkeit, auf einen Modulvorschläg zu klicken, um detailliertere Informationen zu diesem Modul zu erhalten. Zudem kann er Kommentare zu diesem Modul lesen. Diese Kommentare wurden von mir verfasst oder basieren auf Meinungen, die ich eingeholt habe. Um das gesamte System und die Kommentare authentischer wirken zu lassen, sind die Namen der Verfasser dieser Kommentare zufallsgeneriert. Würde ausschließlich mein Name als Verfasser erscheinen, könnte das System künstlich wirken und damit den Eindruck eines realistischen Use Cases für den Studenten mindern, auch wenn diese Kommentare eine reale Meinung zeigen. Dieser Use Case ist nicht als Auslöser für Erklärungsbedarf vorgesehen. Vielmehr soll er dem Studenten einen Mehrwert bieten und die Anforderung an einen realen Anwendungsfall erfüllen.

In Abbildung 4.6 ist die Implementierung des UC. Im oberen Bereich werden der Name des Moduls sowie dessen zugehöriger Kompetenzbereich aufgeführt. Darunter folgen Angaben zu einem Ansprechpartner sowie den im Modul zu erwerbenden Leistungspunkten. Im Anschluss daran werden die in diesem Modul behandelten Themen aufgeführt. Sämtliche dieser Informationen entstammen dem Modulkatalog des Bachelorstudiengangs Informatik an der Leibniz Universität Hannover. Im unteren Abschnitt

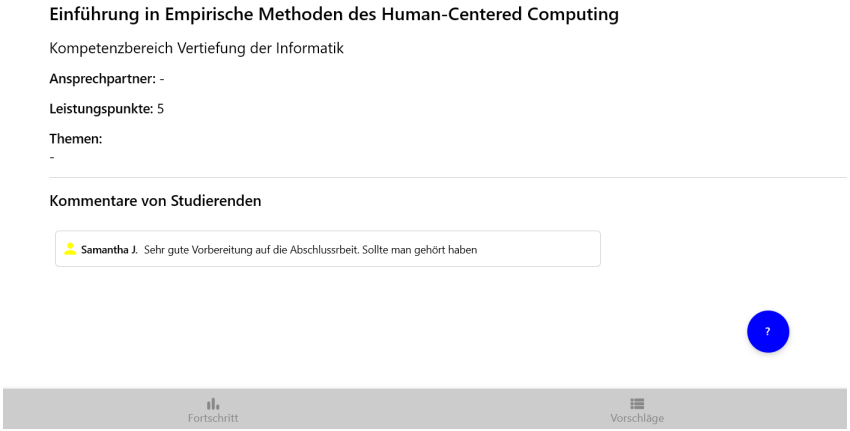

Abbildung 4.6: Implementierung Use Case 6

ist eine Liste aller zu diesem Modul von mir formulierten Kommentare dargestellt.

Use Case 7

Use Case 7	**Studienfortschritt anzeigen**
Erläuterung	Der Student bekommt seinen Studienfortschritt durch Fortschrittsbalken angezeigt.
Ebene	Hauptfunktion
Vorbedingung	Der Student hat mindestens drei Module als absolviert angegeben.
Erfolgsgarantie	Das System zeigt Fortschrittsbalken für verschiedene Kompetenzbereiche an.
Hauptakteur	Student
Auslöser	Der Student hat drei Module bewertet und dann auf 'Weiter' gedrückt, sowie im Bottommenü 'Fortschritt' ausgewählt.
Hauptszenario	1. Der Student bewertet drei Module und drückt dann auf 'Weiter' und wählt im Bottommenü 'Fortschritt' aus. 2. Das System generiert Fortschritsbalken und zeigt diese an.
Erweiterungen	-
Erklärungsbedarf	-
Verwendungshäufigkeit	Immer

Neben der Ansicht von Modulvorschlägen hat der Student die Möglichkeit, über ein Bottom-Navigationsmenü zu einer Seite zu wechseln, auf der er seinen bisherigen Studienfortschritt einsehen kann. Das System stellt diesen Fortschritt in Form mehrerer Balkendiagramme dar, die zeigen, wie viele Leistungspunkte der Student insgesamt sowie in den verschiedenen Pflichtkompetenzbereichen bereits erreicht hat. Ein Klick auf einen dieser Balken zeigt zudem eine Liste der noch zu belegenden Module im jeweiligen Kompetenzbereich an. Auch dieser Use Case soll dem Studenten einen klaren Mehrwert bieten, ohne Erklärungsbedarf hervorzurufen.

In Abbildung 4.7 wird die Umsetzung des UC7 dargestellt. Im oberen Bereich wird der Gesamtfortschritt visualisiert, indem angezeigt wird, wie viele der insgesamt 180 erforderlichen Leistungspunkte bereits erreicht wurden. Darunter folgt eine detaillierte Aufschlüsselung der bisher erworbenen Punkte in den fünf Pflichtkompetenzbereichen: Grundlagen der Informatik, Mathematik, Elektrotechnik, Vertiefung Informatik und Studium Generale. Kompetenzbereiche, in denen keine Leistungspunkte erworben werden müssen, wurden in dieser Ansicht bewusst ausgelassen, um eine möglichst übersichtliche Darstellung zu gewährleisten. Diese Maßnahme dient auch

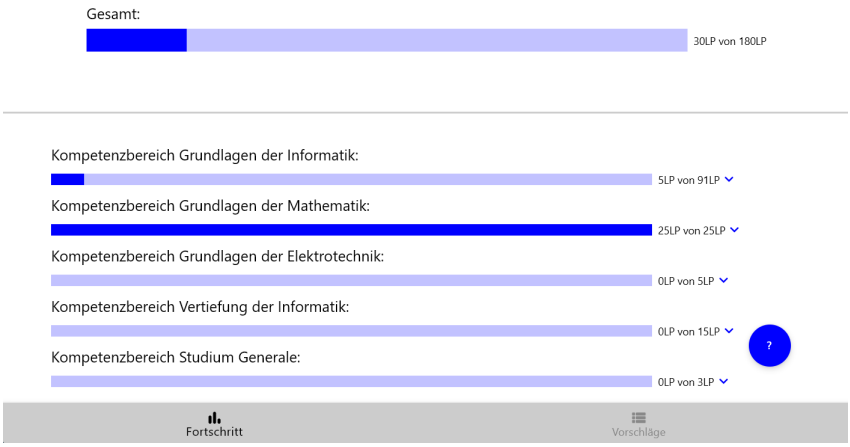

Abbildung 4.7: Implementierung Use Case 7

dazu, Missverständnisse zu vermeiden, die entstehen könnten, wenn der Eindruck erweckt würde, dass in diesen optionalen Kompetenzbereichen ebenfalls Punkte erforderlich seien.

Use Case 8

Use Case 8	**Erklärungen anzeigen**
Erläuterung	Der Student hat Erklärungsbedarf und bekommt auf Abruf Erklärungen vom System angezeigt.
Ebene	Nebenfunktion
Vorbedingung	-
Erfolgsgarantie	Das System zeigt eine ausklappbare Liste mit allen Kategorien von Erklärungsbedarf.
Hauptakteur	Student
Auslöser	Student drückt auf die Schaltfläche mit '?'.
Hauptszenario	1. Der Student hat Erklärungsbedarf und drückt auf die Schaltfläche mit '?'. 2. Das System zeigt Erklärungen an. 3. Der Student liest die Erklärung(-en) und schließt Liste wieder.
Erweiterungen	-
Erklärungsbedarf	-
Verwendungshäufigkeit	regelmäßig

Dieser UC erweitert alle anderen Use Cases. Er bietet den Studierenden die Möglichkeit, bei Bedarf eine Erklärung zu erhalten. Abhängig davon, in welchem Use Case sich der Studierende aktuell befindet, können Erklärungen zu den fünf Kategorien abgerufen werden, die im Kapitel 2 erläutert wurden. Um eine Erklärung anzufordern, drückt der Studierende den dauerhaft unten rechts angezeigten Knopf. Anschließend kann er aus den fünf verfügbaren Kategorien wählen und sich, falls vorhanden, die bereitgestellte Erklärung durchlesen.

Abbildung 4.8: Implementierung Use Case 8

In Abbildung 4.8 wird die Implementierung der Funktion zur Anzeige von Erklärungen veranschaulicht. Betätigt der Nutzer den Knopf in der unteren rechten Ecke eines Bildschirms, öffnet sich, wie in der Abbildung dargestellt, ein Menü, das alle Kategorien von Erklärungsbedarf auflistet. Der Nutzer hat die Möglichkeit, eine oder mehrere Kategorien auszuwählen, deren Erklärung er einsehen möchte, und diese anschließend auszuklappen und zu lesen. Durch erneutes Betätigen des Knopfes schließt sich das Menü wieder. Dieses Verhalten wird durch die rote Färbung des Knopfes bei geöffnetem Menü visuell hervorgehoben.

4.2 Technische Details

Das System mit allen oben beschriebenen Anwendungsfällen wurden als Desktop-Applikationen entwickelt. Dafür kam das UI-Framework Compose Multiplatform zum Einsatz, das sich hervorragend für die Entwicklung von Desktop-Anwendungen auf Basis von Kotlin eignet [14]. Um die Komplexität des Systems zu minimieren, wurde bewusst auf die Implementierung eines Backends, also auf eine Serverseitige Datenbank, verzichtet. Stattdessen werden die Module des Studiengangs aus einer integrierten CSV-Datei ausgelesen. Diese Csv-Datei wird lokal gespeichert und durch eine Csv-Parser Bibliothek ausgelesen. Diese Umsetzung hat den Vorteil, dass sie Netzwerkunabhängig ist und daher eine niedrige Reaktionszeit aufweist. Zudem fallen Serverwartungsarbeiten komplett weg.

4.3 Wart- und Erweiterbarkeit

Eine wichtige Anforderung an das System war die einfache Wart- und Erweiterbarkeit. Es folgt eine kurze Erläuterung wie diese Anforderungen umgesetzt worden sind.

Modularität

Das verwendete Framework "Compose Multiplatform" erfüllt diese Anforderung, wie in 2 erwähnt, durch seine Architektur automatisch. Jedes UI-Element ist einzeln und unabhängig definiert. Zum Beispiel repräsentiert die Benutzeroberfläche von UC1 einen Composable, der wiederum aus den vier Textfeld-Composables, dem Composable für den Knopf sowie weiteren Composables besteht. Jeder dieser Composables ist dabei unabhängig definiert.

Wiederverwendbarkeit

Dadurch, dass wie in 2 beschrieben jedes einzelne UI-Element unabhängig definiert ist, kann jedes Element überall wiederverwendet werden. Zum Beispiel wird der Knopf, um Erklärungen abzurufen in jedem einzelnen Use Case benötigt. Deshalb wird das Composable einmal definiert und dann in jedem Use Case wiederverwendet.

Modifizierbarkeit

Um sicherzustellen, dass zukünftige Anpassungen durch andere Entwickler reibungslos vorgenommen werden können, wurde eine umfassende und detaillierte Code-Dokumentation erstellt. Diese Dokumentation erklärt nicht nur den bestehenden Code, sondern bietet auch Anleitungen für mögliche zukünftige Änderungen. Dies stellt sicher, dass die Wartung und Erweiterung des Systems in der Zukunft effizient und nachvollziehbar bleibt. Zudem wurden Architekturentscheidungen so getroffen, dass es einfach möglich ist weitere Benutzeroberflächen anzubinden ohne große Codeänderungen vornehmen zu müssen. Zum Beispiel muss eine neue Benutzeroberfläche nur in die bestehende Navigationslogik hinzugefügt werden, um sie zu verwenden.

Kapitel 5

Nutzerstudie

5.1 Forschungsfragen

Das Ziel dieser Arbeit bestand in der Entwicklung eines Systems, das bei den Nutzern Erklärungsbedarf hervorruft. Hierzu wurden zunächst verschiedene Szenarien entworfen, die diesen Bedarf potenziell auslösen können. Anschließend wurden diese Szenarien gemäß der beschriebenen Kriterien implementiert. Im abschließenden Schritt wurde eine Nutzerstudie durchgeführt, um zu evaluieren, ob die zuvor definierten Anforderungen erfolgreich umgesetzt wurden. Zudem sollte die Studie Aufschluss darüber geben, welche der konzipierten Szenarien besonders effektiv im Auslösen von Erklärungsbedarf sind und welche vom Nutzer als weniger benutzerfreundlich wahrgenommen werden. Da der Erklärungsbedarf nicht durch schlechte Benutzerfreundlichkeit und damit einhergehende Frustration ausgeöst werden soll. Konkret lauteten die Forschungsfragen, die diese Nutzerstudie beantworten sollte wie folgt:

RQ1: Löst das System Erklärungsbedarf bei Nutzern aus?

RQ2: Welche Auslöseszenarien lösen besonders oft Erklärungsbedarf aus und welche nicht?

RQ3: Wirken sich die Auslöseszenarien aus Sicht des Nutzers negativ auf die Benutzerfreundlichkeit aus?

Die erste Forschungsfrage (RQ1) zielt darauf ab, zu untersuchen, ob das System grundsätzlich geeignet ist, den beabsichtigten Erklärungsbedarf bei den Nutzenden zu erzeugen. Die zweite Forschungsfrage (RQ2) befasst sich detaillierter mit der Identifizierung von Szenarien, die besonders effektiv darin waren, Erklärungsbedarf auszulösen, sowie mit solchen, die keinen Erklärungsbedarf hervorgerufen haben. Die dritte Forschungsfrage (RQ3) untersucht, ob das Ziel erreicht wurde, den Erklärungsbedarf nicht durch

mangelnde Benutzerfreundlichkeit und daraus resultierende Frustration zu erzeugen.

5.2 Teilnehmer

Wie bereits aus den Use Cases hervorgeht, wurde das System für Bachelorstudenten der Informatik an der Leibniz Universität Hannover entwickelt. Demzufolge waren nur Teilnehmer zugelassen, die diese Bedingung erfüllten. Zudem mussten die Teilnehmer mindestens drei Module in ihrem Studium erfolgreich absolviert haben, wie aus der Beschreibung des dritten Use Cases hervorgeht. Weitere Einschränkungen bei der Auswahl der Teilnehmer wurden nicht vorgenommen.

5.3 Ablauf

Der Ablauf der Nutzerstudie hat sich in drei Teile aufgeteilt: Einführung, Nutzung des Systems und Ausfüllen des Fragebogens.

Einführung

Zu Beginn der Studie wurde den Teilnehmern eine kurze Einführung in den Ablauf gegeben. Es wurde ihnen erläutert, dass sie ein von mir entwickeltes System verwenden werden. Dieses System fungiert als Semesterplaner, der Modulvorschläge für das zukünftige Studium generiert und eine Übersicht über den aktuellen Studienfortschritt bietet. Zudem wurde der Teilnehmer auf den in der rechten unteren Ecke verfügbaren Knopf hingewiesen, der bei Fragen während der Nutzung des Systems verwendet werden kann.

Nutzung des Systems

Im Anschluss wurde der Teilnehmer angewiesen, das System selbstständig zu verwenden, mit dem Hinweis, sich dabei so viel Zeit zu nehmen, wie erforderlich. Während dieser Phase wurde der Teilnehmer allein gelassen, um eine ungestörte Nutzung des Systems zu gewährleisten und das Gefühl einer Beobachtung zu vermeiden, damit er sich natürlich verhält und die Ergebnisse nicht verfälscht werden.

Ausfüllen des Fragebogens

Nachdem der Teilnehmer signalisiert hatte, dass er die Nutzung abgeschlossen hat, wurde der im Anhang befindliche Fragebogen ausgefüllt. Die Bearbeitung des Fragebogens erfolgte in Form eines Interviews. Diese Methode wurde gewählt, um die Gedankengänge und Beweggründe hinter den Antworten besser zu verstehen und qualitativ hochwertige Ergebnisse

zu erzielen [16]. Das Ziel war, alle Aspekte zur Benutzerfreundlichkeit und zum Erklärungsbedarf vollständig zu erfassen. Bei einer eigenständigen Bearbeitung des Fragebogens durch den Teilnehmer bestand die Gefahr, dass Antworten kürzer und weniger ausführlich ausfallen, da Aspekte vergessen oder aus Bequemlichkeit weggelassen werden könnten. Dies sollte unbedingt vermieden werden.

5.4 Daten

Im Rahmen dieser Nutzerstudie wird zum einen durch das System automatisch erfasst welche Erklärungen durch den Teilnehmer abgerufen wurden. Aus diesen Daten kann man dann auf einen Erklärungsbedarf schließen. Andererseits werden durch die Beantwortung des Fragebogens verschiedene Freitextantworten erhoben. Da die Bearbeitung des Fragebogens in Form eines geleiteten Interviews erfolgt, können relevante Antworten direkt in Form von Stichpunkten dokumentiert werden, wodurch eine nachträgliche Kodierung der Antworten nicht mehr gebraucht wird.

Demographische Daten

An der Nutzerstudie nahmen fünf Bachelor-Studierende der Informatik der Leibniz Universität Hannover (LUH) teil. Zum Zeitpunkt der Teilnahme befanden sich die Teilnehmer durchschnittlich im 8,2. Fachsemester, also am Ende ihres Studiums. Sie waren daher die ideale Zielgruppe für die Studie, da eine Voraussetzung war, dass die Teilnehmer bereits Module in ihrem Studium abgeschlossen hatten.

Kapitel 6

Ergebnisse und Diskussion

6.1 Ergebnisse

Die Ergebnisse der im vorherigen Kapitel beschriebenen Nutzerstudie werden in diesem Kapitel dargestellt. Wie im zugehörigen Use Case erläutert, hat das System das Abrufen von Erklärungen in einer Datei protokolliert, um nachvollziehen zu können, in welchen Situationen bei den Studierenden ein Bedarf an Erklärungen entstand. Bei der Durchführung der Nutzerstudie stellte sich heraus, dass 80% der Teilnehmer Erklärungen aus Neugierde und nicht aufgrund eines tatsächlichen Erklärungsbedarfs abgerufen haben. Aus diesem Grund waren die in diesen Fällen entstandenen Dateien nicht verwertbar. Stattdessen wurde am Ende des Fragebogens explizit durch Nachfragen erfasst, an welchen Stellen Erklärungsbedarf bestand und welche Erklärungen tatsächlich abgerufen wurden. Auf diese Weise konnten dennoch verwertbare Ergebnisse zum Erklärungsbedarf gewonnen werden.

Im Folgenden werden nun die beiden Aspekte Erklärungsbedarf und Benutzerfreundlichkeit ausführlich dargestellt.

6.1.1 Erklärungsbedarf

Module auswählen (Use Case 2)

Tabelle 6.1 und Abbildung 6.1 zeigen die Anzahl der Teilnehmer, bei denen die jeweiligen Szenarien während der Studie Erklärungsbedarf ausgelöst haben. Es wird deutlich, dass der Use Case "Module auswählen" bei allen Teilnehmern Erklärungsbedarf ausgelöst hat. Häufig wurde berichtet, dass die Teilnehmer zunächst versuchten, durch Klicken ein Modul auszuwählen. Aufbauend darauf illustriert Tabelle 6.2, welche der Teilnehmer, die bei diesem Szenario Erklärungsbedarf hatten, tatsächlich eine Erklärung abgerufen haben und wie viele dies nicht taten. Diese Information dient dazu, um zu Unterscheiden zwischen Erklärungsbedarf bei dem der Nutzer auch eine Erklärung möchte und Erklärungsbedarf bei dem es dem Nutzer

Szenario	Kategorie	Häufigkeit
Persöhnliche Daten eingeben	Sicherheit	20,00%
Geburtsdatum eingeben	Bedienung	0,00%
Module auswählen	Bedienung	100,00%
Modulumrandungen	Benutzeroberfläche	60,00%
Modul bewerten	Fachwissen	40,00%
Kommentar schreiben	Bedienung	20,00%
Kommentar schreiben	Sicherheit	20,00%
Modulvorschläge	Systemverhalten	60,00%

Tabelle 6.1: Häufigkeit des Auslösens von Erklärungsbedarf

Szenario	Erklärung abgerufen	Keine Erklärung abgerufen
Persöhnliche Daten eingeben	0,00%	100,00%
Geburtsdatum eingeben	0,00%	0,00%
Module auswählen	80,00%	20,00%
Modulumrandungen	33,33%	66,66%
Modul bewerten	0,0%	100,00%
Kommentar schreiben (Bedienung)	100,00%	0,00%
Kommentar schreiben (Sicherheit)	0,00%	100,00%
Modulvorschläge	50,00%	50,00%

Tabelle 6.2: Häufigkeit des Abrufens von einer Erklärung, falls Erklärungsbedarf vorhanden war (Erfassung durch extra Fragen im Fragebogen)

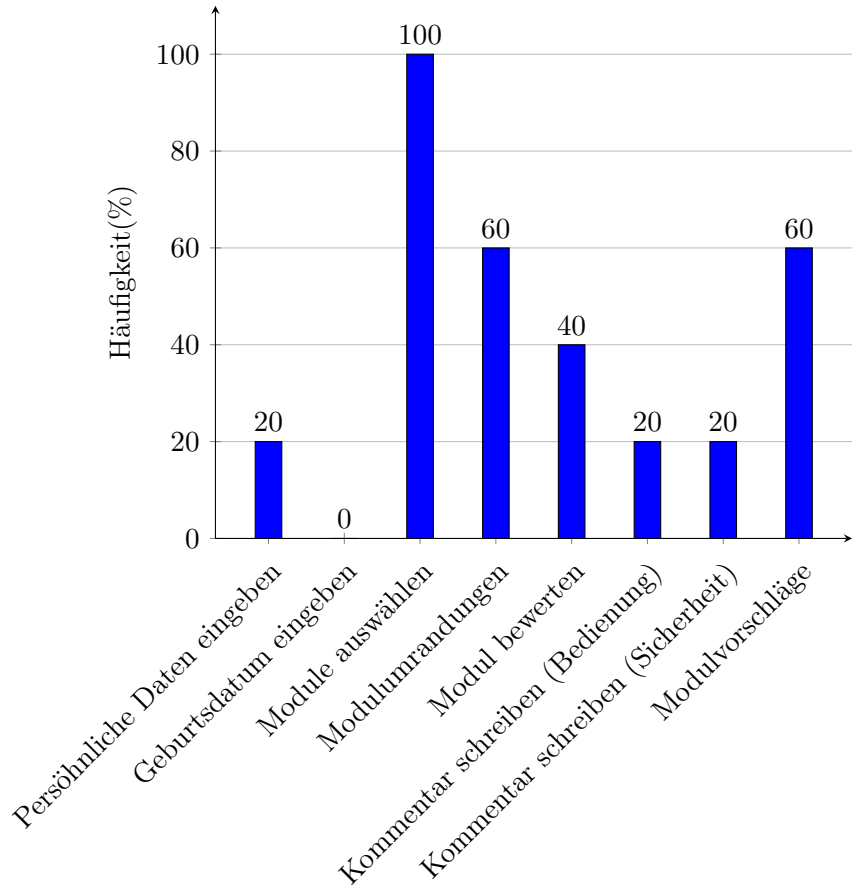

Abbildung 6.1: Diagramm zur Auslösequote der verschiedenen Szenarien

nicht so wichtig ist eine Erklärung zu erhalten. Aus dieser Tabelle geht hervor, dass 20% der Teilnehmer, die beim Szenario "Module auswählen" Erklärungsbedarf hatten, keine Erklärung abgerufen haben. Als Grund gaben diese Teilnehmer im Interview an, dass sie durch Ausprobieren die Funktionsweise der Interaktionstechnik verstanden haben und deshalb keine Erklärung benötigten.

Modulumrandungen (Use Case 2)

Bei drei (60%) der Teilnehmer haben die unterschiedlichen Modulumrandungen während des Auswählens Erklärungsbedarf ausgelöst. Viele Teilnehmer äußerten, dass sie sich über die Bedeutung der verschiedenen Farben wunderten. Häufig wurden eigene Annahmen getroffen, wie zum Beispiel, dass die Farben die Verfügbarkeit im nächsten Semester anzeigen. Tabelle 6.2 zeigt, dass nur ein Drittel der Teilnehmer, die in diesem Szenario Erklärungsbedarf hatten, eine Erklärung in Anspruch genommen haben.

Zwei Drittel gaben an, eine Annahme getroffen zu haben und daher keine Erklärung benötigt zu haben, da ihnen die Bedeutung nicht so wichtig erschien. Hier lässt sich beobachten, dass ein Erklärungsbedarf nicht immer bedeutet, dass auch eine Erklärung gewünscht ist.

Modulvorschläge (Use Case 5)

Auch bei drei (60%) der Teilnehmer haben die vom System generierten Modulvorschläge Erklärungsbedarf ausgelöst. Es wurde angegeben, dass manche Vorschläge für die jeweiligen Teilnehmer tatsächlich interessant waren. Jedoch sorgte spätestens der eingefügte fachfremde Modulvorschlag auch bei diesen Teilnehmern für Verwirrung. Tabelle 6.2 zeigt, dass nur 50% dieser Teilnehmer eine Erklärung abgerufen haben. Gründe, keine Erklärung abgerufen zu haben oder erst gar kein Erklärungsbedarf gehabt zu haben, waren, dass es auch gute Vorschläge gab und dass es den Teilnehmern "egal" war.

Module bewerten (Use Case 3)

Auch der Begriff "Kohärenz" hat während der Bewertung der Module wie geplant Erklärungsbedarf ausgelöst, da zwei (40%) der Teilnehmer die Bedeutung dieses Begriffs nicht verstanden. Der Begriff "Dozent" hingegen hat keinen Erklärungsbedarf ausgelöst, da alle Teilnehmer diesen Begriff kannten. Tabelle 6.2 zeigt, dass niemand zu diesem Aspekt eine Erklärung in Anspruch genommen hat. Ein konkreter Grund dafür konnte im Interview nicht genannt werden.

Persöhnliche Daten eingeben (Use Case 1) / Kommentar schreiben (Sicherheit;Use Case 3)

Die beiden Szenarien, welche Erklärungsbedarf zur Sicherheit auslösen sollten haben jeweils nur bei einem (20%) der Teilnehmer für Erklärungsbedarf gesorgt. Bei dem Bewerten der Module wurde sich von einem (20%) der Teilnehmern gefragt, was mit dem geschriebenen Kommentar passiert. Und bei demselben Teilnehmer hat auch die Anforderung der Eingabe der Matrikelnummer für Bedenken gesorgt, weil nicht klar war wofür diese Angabe wichtig für das System ist. Angegeben wurde, dass es trotz der Verwunderung über die Nötigkeit dieser Angaben keine Bedenken zur Sicherheit gab und deshalb auch keine Erklärung abgerufen wurde.

Kommentar schreiben (Bedienung;Use Case 3)

Die Mindeslänge von 90 Zeichen für den Kommentar bei der Bewertung hat auch nur bei einem (20%) der Teilnehmern für Erklärungsbedarf

gesorgt. Dieser Teilnehmer hat dann auch eine Erklärung benutzt, um herauszufinden, dass es eine Mindestlänge gibt.

Geburtsdatum eingeben (Use Case 1)

Und bei keinem der Teilnehmer gab es Erklärungsbedarf zum Eingabeformat des Geburtsdatums.

Bewertung der bereitgestellten Erklärungen

Die Teilnehmer wurden auch zur Qualität der bereitgestellten Erklärungen befragt. Nur ein Teilnehmer äußerte sich negativ zur Erklärung bezüglich des Auswählens der Module. Bei genauerer Nachfrage stellte sich heraus, dass die Erklärung nicht vollständig gelesen worden war, weshalb die Kritik anschließend zurückgezogen wurde. Alle anderen Teilnehmer waren vollkommen zufrieden mit den bereitgestellten Erklärungen zu den Auslöseszenarien.

6.1.2 Benutzerfreundlichkeit

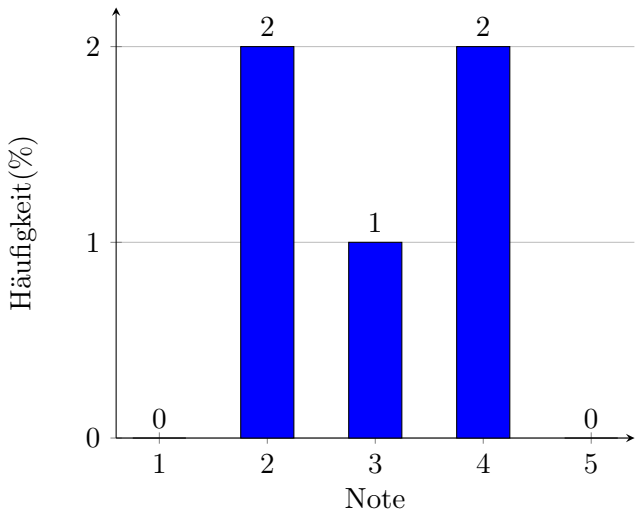

Abbildung 6.2: Bewertung der Benutzerfreundlichkeit des Systems

Wie dem Fragebogen zu entnehmen ist, wurde die Benutzerfreundlichkeit des Systems auf einer Notenskala von 1 (sehr gut) bis 5 (sehr schlecht) bewertet. Der Median der vergebenen Noten lag bei 3. Die beste vergebene Note war 2, die schlechteste 4. Die Notenverteilung ist in 6.2 zu sehen. Die folgenden Punkte wurden als negative Einflüsse auf die Benutzerfreundlichkeit genannt:

- Unklarheit darüber, dass der eingegebene Kommentar mindestens 90 Zeichen lang sein muss (2 mal)

- 90 Zeichen Mindestlänge zu hoch (1 mal)
- Fehlende Indikatoren, dass man scrollen kann (1 mal)
- Tab-Taste funktioniert nicht, um zum nächsten Feld zu gelangen (1 mal)
- Verschiedene Nebenfächer sind bei der Auswahl nicht kategorisiert (1 mal)
- Schwierigkeit, Module durch Wischen auszuwählen (1 mal)
- Unwissenheit, ob man das Ende erreicht hat (2 mal)

Am häufigsten wurden dabei die Punkte zu der Kommentar-Mindestlänge und über das Erreichen des Endes geäußert. Alle anderen Punkte wurden jeweils nur von einem einzelnen Teilnehmer genannt.

6.2 Diskussion

Die Ergebnisse aus dem vorangegangenen Unterkapitel werden in diesem Unterkapitel nun eingeordnet, diskutiert und zum Beantworten der aufgstellten Forschungsfragen genutzt.

6.2.1 RQ1

RQ1: Löst das System Erklärungsbedarf bei Nutzern aus?

Beantwortung

Wir gehen davon aus, dass alle Teilnehmenden den Fragebogen vollständig ehrlich beantwortet haben. Auf dieser Grundlage konnte festgestellt werden, dass das System bei jedem Teilnehmenden mindestens einmal, im Durchschnitt jedoch mehrmals, Erklärungsbedarf hervorrief. Da es sich um eine kleine Nutzerstudie mit nur fünf Teilnehmenden handelt, sollten die Ergebnisse lediglich als erste Anhaltspunkte betrachtet werden. Dennoch deuten die Resultate darauf hin, dass die entwickelte Software grundsätzlich in der Lage ist, Erklärungsbedarf auszulösen. Somit gibt die Studie einen Anhaltspunkt darauf, dass die erste Forschungsfrage positiv beantwortet werden kann.

6.2.2 RQ2

RQ2: Welche Auslöseszenarien lösen besonders oft Erklärungsbedarf aus und welche nicht?

Interpretation der Ergebnisse

Durch die erzielten Ergebnisse kann festgestellt werden, welche der entwickelten Szenarien effektiv waren, um Erklärungsbedarf auszulösen, und welche sich als ineffektiv erwiesen haben. Die Interaktionstechnik des Wischens zum Auswählen der Module hat sich als besonders effektiv erwiesen. Diese Interaktionstechnik war allen Teilnehmern unbekannt. Vier (80%) der Teilnehmer benötigten eine Erklärung, während einer sie durch Ausprobieren verstanden hat. Diese Ergebnisse legen nahe, dass neuartige Interaktionstechniken besonders geeignet sind, um Erklärungsbedarf auszulösen. Gleichzeitig scheint es so, dass bei dieser Art Erklärungsbedarf auch vermehrt Erklärungen gewünscht sind.

Das Szenario zur Kategorie Benutzeroberfläche erwies sich ebenfalls als sehr effektiv, mit drei (60%) mal ausgelöstem Erklärungsbedarf. Die Module hatten je nach ihren Leistungspunkten unterschiedliche Farben als Umrandung. Ungewöhnliche Details in der Benutzeroberfläche scheinen demnach ebenfalls gut geeignet zu sein, um Erklärungsbedarf zu generieren. Im Gegensatz zur Interaktionstechnik wird dieser Erklärungsbedarf jedoch häufig übergangen oder es werden Annahmen über die Bedeutung getroffen, da es für die Nutzung des Systems nicht notwendig ist, die verschiedenen Farben zu verstehen. Aufgrund dessen kann diese Art des Erklärungsbedarfs schwierig sein zu erfassen, deshalb muss besonders darauf geachtet werden, diesen Erklärungsbedarf korrekt zu erfassen.

Auch bei drei (60%) der Teilnehmern hat das Szenario zur Kategorie Systemverhalten Erklärungsbedarf erzeugt. Das System generierte zufällige, nicht auf den Eingaben basierende Modulvorschläge. Auch hier zeigt sich, dass nur die Hälfte der Teilnehmer mit Erklärungsbedarf in Bezug auf dieses Szenario eine Erklärung abgerufen hat. Da das Verständnis dieses Szenarios für die Nutzung des Systems ebenfalls nicht notwendig ist, zeigt sich, dass bei Szenarien, deren Verständnis für die weitere Nutzung essenziell ist, mehr Teilnehmer eine Erklärung abrufen als bei Szenarien, bei denen das Verständnis nicht zwingend erforderlich ist. Dennoch scheint auch dieses Szenario gut geeignet zu sein, Erklärungsbedarf auszulösen.

Das Szenario in der Kategorie Fachwissen führte bei zwei (40%) der Teilnehmern zu einem Bedarf an Erklärungen. Bei der Bewertung der Module wurde die Kategorie "Kohärenz der Konzepte" herangezogen, wobei der Begriff "Kohärenz" diesen zwei (40%) der Teilnehmern unbekannt war. Bemerkenswert ist, dass niemand eine Erklärung zu diesem Erklärungsbedarf abgerufen hat. Dies bestätigt die Beobachtung, dass weniger Erklärungen für als nicht notwendig erachtete Szenarien abgerufen werden. Eine Erklärung dafür, warum keine Erklärung abgerufen wurde, wurde nicht angegeben und kann daher nur vermutet werden. Der Erklärungsbedarf in Bezug auf Fachwissen kann in bestimmten Fällen mit Peinlichkeit verbunden sein, was dazu führen könnte, dass keine Erklärung abgerufen wird. Andererseits

waren die Teilnehmer unbeobachtet und wussten nicht, dass abgerufene Erklärungen gespeichert werden. Zudem gaben sie im Fragebogen freiwillig an, dass sie einen Erklärungsbedarf hatten. Diese Punkte sprechen gegen die Theorie, dass Schamgefühle eine Rolle spielt. Ein wahrscheinlicherer Grund, ähnlich wie bei anderen Szenarien, ist, dass es den Teilnehmern nicht wichtig genug war, bzw. sie es als "egal" empfanden. Es zeigt sich jedoch hier, dass mit im Alltag seltener gebräuclichen Begriffen Erklärungsbedarf erzeugt werden kann. Die Tatsache, dass kein Teilnehmer eine Erklärung abgerufen hat, könnte darauf hindeuten, dass eine an dieser Stelle angezeigte Erklärung als störend wahrgenommen und daher als unerwünscht betrachtet werden könnte.

Mit einer Auslösequote von jeweils nur einmal (20%) haben die beiden Szenarien zur Kategorie Sicherheit nicht wie geplant funktioniert. Die Studie zeigt, dass den Teilnehmenden Sicherheits- und Privatsphäreaspekte nicht besonders wichtig waren. Die übrigen 80% äußerten, dass es ihnen egal war, was mit den eingegebenen Daten geschieht beziehungsweise wofür diese Daten benötigt werden. Daraus lässt sich folgern, dass es schwierig ist, einen Erklärungsbedarf zum Aspekt Sicherheit auszulösen. Die implementierten Szenarien scheinen nicht ausreichend zu sein und müssen bei einer Weiterentwicklung möglicherweise überarbeitet werden. Ein möglicher Grund hierfür ist, dass es sich um ein Studienumfeld handelt, in dem die Teilnehmenden Vertrauen gegenüber der Studie haben und keine Gefahr vermuten.

Zuletzt gab es zwei weitere Szenarien in der Kategorie Bedienung, die sich ebenfalls als ineffizient erwiesen haben. Das Szenario, bei dem die Teilnehmer während der Bewertung einen Kommentar zu einem Modul verfassen sollten, der mindestens 90 Zeichen lang sein musste, löste nur bei einem (20%) der Teilnehmer einen Erklärungsbedarf aus. Ein wahrscheinlicher Grund dafür ist eine kleine Anzeige unten rechts, die auf die Mindestlänge hinwies. Diese Anzeige wurde implementiert, um Einbußen in der Benutzerfreundlichkeit zu vermeiden. Der Teilnehmer, bei dem Erklärungsbedarf auftrat, gab an, diese Anzeige nicht gesehen zu haben. Ähnlich verhielt es sich bei dem Szenario, in dem die Teilnehmer ihr Geburtsdatum im Format 'TT.MM.JJJJ' eingeben sollten. Auch hier wurde das Formatmuster aus denselben Gründen angezeigt. In diesem Fall übersah kein Teilnehmer die Anzeige, und folglich trat bei niemandem Erklärungsbedarf auf. Diese beiden Szenarien zeigen sich als ungeeignet und sollten überarbeitet werden. Es besteht ein Konflikt zwischen dem Auslösen von Erklärungsbedarf und der Benutzerfreundlichkeit. Zukünftig sollten Szenarien entwickelt werden, die besser geeignet sind, Erklärungsbedarf auszulösen, ohne die Benutzerfreundlichkeit zu beeinträchtigen.

Beantwortung

Die Studie legt nahe, dass die Wischbewegung zur Auswahl von Modulen eine effektive Methode darstellt, um Erklärungsbedarf auszulösen. Darüber hinaus haben die unterschiedlich farbigen Rahmen sowie der simulierte Algorithmus zur Generierung von Vorschlägen häufig zu Erklärungsbedarf geführt. Zusammenfassend weist die Studie darauf hin, dass neuartige Interaktionstechniken und ungewöhnliche Details in der Benutzeroberfläche als besonders effektiv bei dem Auslösen von Erklärungsbedarf zu sein scheinen, während Szenarien im Bereich der Privatsphäre weniger erfolgreich zu sein scheinen. Die Ergebnisse verdeutlichen, dass Erklärungen vor allem in jenen Szenarien abgerufen werden, deren Verständnis als essenziell für die Systemnutzung wahrgenommen wird.

6.2.3 RQ3

RQ3: Wirken sich die Auslöseszenarien aus Sicht des Nutzers negativ auf die Benutzerfreundlichkeit aus?

Beantwortung

Der Median der Bewertungen lag bei 3. Zur Beantwortung der Fragestellung müssen jedoch die angegebenen Gründe genauer analysiert werden. Hauptsächlich wurden Gründe genannt, die nicht mit den Auslöseszenarien in Zusammenhang standen. Dennoch gab es zwei Kritikpunkte bezüglich der Benutzerfreundlichkeit der Auslöseszenarien.

Zum einen wurde angemerkt, dass die Interaktionstechnik des Wischens und die Mindestlänge des Kommentars schwer verständlich waren. Dies war jedoch beabsichtigt, um einen Erklärungsbedarf zu erzeugen. Wichtig ist, dass die Nutzung der Interaktionstechnik benutzerfreundlich ist. Da keine Kritikpunkte zur Handhabung des Wischens geäußert wurden, kann angenommen werden, dass diese Technik für die Teilnehmer benutzerfreundlich war. Der zweite mehrfach genannte Kritikpunkt betraf die Mindestlänge von 90 Zeichen für Kommentare, die als zu lang empfunden wurde. Teilnehmer äußerten, dass ihnen oft nicht genügend zum Kommentieren einfiel. Dieser Aspekt beeinträchtigte die Benutzerfreundlichkeit, jedoch ist er für dieses Auslöseszenario notwendig. Um dies zu vermeiden müsste also das gesamte Szenario geändert werden. Mindestanforderungen an Kommentare entsprechen jedoch der Realität und sind häufig in Software-Systemen implementiert, sodass dieses Auslöseszenario im aktuellen Zustand eine realistische Situation widerspiegelt. Aus den von den Teilnehmern genannten Gründen lässt sich schließen, dass das Auslösen eines Erklärungsbedarfs an den meisten Stellen die Benutzerfreundlichkeit nicht negativ beeinflusst. Es muss jedoch darauf geachtet werden, dass Szenarien und Bedingungen nicht zu schwer verständlich sind. Andernfalls, wie in den Ergebnissen zu sehen,

wirken sich diese Szenarien negativ aus und beeinträchtigen den Nutzer. Es ist zu beachten, dass der Median der Bewertungen wahrscheinlich höher ausgefallen wäre, wenn die anderen genannten Aspekte, wie fehlende Scroll-Indikatoren oder Abkürzungen, nicht vorhanden gewesen wären. Daher kann festgestellt werden, dass die meisten Auslöseszenarien die Benutzerfreundlichkeit nicht negativ beeinflusst haben. An einigen Stellen ist jedoch eine genauere Betrachtung und möglicherweise auch Anpassung erforderlich, um die Benutzerfreundlichkeit zu gewährleisten.

6.2.4 Gesamtziel

Das Ziel dieser Arbeit bestand in der Entwicklung eines Systems, das Erklärungsbedarf hervorruft. Dabei sollte der Erklärungsbedarf nicht durch mangelnde Benutzerfreundlichkeit, sondern durch reale Anwendungsfälle entstehen. Die Ergebnisse sowie die Beantwortung der Forschungsfragen deuten darauf hin, dass dieses Ziel erreicht wurde. Das System löste bei den Teilnehmern Erklärungsbedarf aus, und bis auf wenige Ausnahmen wurden die Anwendungsfälle nicht als benutzerunfreundlich wahrgenommen.

6.2.5 Limitationen

In diesem Unterkapitel wird auf die Limitationen der Studie eingegangen. Es wird dafür eine Betrachtung nach Wohlin et al. [20] vorgenommen, wobei die Limitationen in vier Kategorien aufgeteilt werden: Konstruktvalidität, interne Validität, Schlussfolgerungsvalidität und externe Validität.

Konstruktvalidität

Zur Konstruktvalidität gehören Fehler, die während der Planung aufgetreten sind oder sich nicht vermeiden ließen [20]. Wie in den Ergebnissen beschrieben, war geplant, den Erklärungsbedarf in einer Datei zu dokumentieren, um nachvollziehen zu können, an welchen Stellen Erklärungen abgerufen wurden und somit ein Erklärungsbedarf bestand. Diese Planung stellte sich jedoch als Fehler heraus, da nicht berücksichtigt wurde, dass Teilnehmer aus Neugierde und nicht aufgrund eines tatsächlichen Erklärungsbedarfs auf Erklärungen klicken könnten. Um dennoch aussagekräftige Ergebnisse zum Erklärungsbedarf zu erhalten, wurden am Ende des Fragebogens zusätzliche Fragen gestellt. Diese Methode birgt jedoch das Risiko, dass Teilnehmer nicht ehrlich antworten oder den Erklärungsbedarf vergessen, was zu Verzerrungen der Ergebnisse führen kann.

Interne Validität

Die interne Validität bezieht sich darauf, inwieweit die erzielten Ergebnisse tatsächlich das messen was sie messen sollen [20]. Es ist wichtig anzumerken,

dass verschiedene Störeinflüsse die Ergebnisse möglicherweise verfälscht haben. Erstens könnten sich die Teilnehmer trotz der Information, dass sie nicht beobachtet werden, dennoch beobachtet gefühlt haben. Zweitens könnten die Teilnehmer im Fragebogen nicht wahrheitsgemäße Antworten gegeben haben, um das entwickelte System aus Rücksicht auf den Studienleiter nicht negativ darzustellen. Beide Faktoren können potenziell erhebliche Auswirkungen auf die Ergebnisse haben.

Schlussfolgerungsvalidität

Die Schlussfolgerungsvalidität bezieht sich darauf, inwieweit aus den erzielten Ergebnissen valide Schlussfolgerungen gezogen werden können [20]. Aufgrund der sehr kleinen Stichprobe von nur fünf Teilnehmern sind die Ergebnisse nicht statistisch signifikant. Sie sollten daher als vorläufige Hinweise betrachtet werden, die weitergehender Untersuchung bedürfen.

Externe Validität

Die externe Validität bezieht sich darauf, ob die Ergebnisse auch in anderen Kontexten gelten [20]. Da die Teilnehmerzahl so gering war, sind die Ergebnisse möglicherweise nicht für eine größere Gruppe representativ. Außerdem könnten Merkmale wie Vorerfahrung mit Software die Generalisierbarkeit einschränken. Um die externe Validität zu erhöhen ist eine größere Anzahl an Teilnehmer nötig.

Kapitel 7

Zusammenfassung und Ausblick

Dieses Kapitel dient dazu, einen abschließenden Überblick über die Inhalte und die wichtigsten Ergebnisse der Arbeit zu geben.

7.1 Zusammenfassung

In dieser Arbeit wurde die Entwicklung eines Software-Systems angestrebt, das in der Lage ist, einige in der Literatur beschriebenen Kategorien von Erklärungsbedarf – wie Systemverhalten, Bedienung, Sicherheit, Benutzeroberfläche und Fachwissen – auszulösen. Dabei wurde besonderes Augenmerk darauf gelegt, einen realen Anwendungsfall zu schaffen, der durch eine hohe Benutzerfreundlichkeit den Erklärungsbedarf authentisch und nicht durch Frustration oder hypothetische Szenarien hervorruft. Dieses System soll zukünftig in Studien zum Thema Erklärungsbedarf eingesetzt werden.

Zur Umsetzung dieses Ziels wurde ein Semesterplaner implementiert. Dieser Anwendungsfall bietet sich besonders an, da er zahlreiche Gelegenheiten zur Auslösung unterschiedlicher Erklärungsbedarfe, etwa durch die Simulation einer Künstlichen Intelligenz, bietet. Gleichzeitig liefert das System einen echten Mehrwert für Studierende, wodurch die Notwendigkeit entfällt, dass sich die Teilnehmenden in hypothetische Szenarien hineinversetzen müssen.

Nach der Implementierung wurde eine Nutzerstudie durchgeführt, um zu überprüfen, ob die gesetzten Ziele erreicht wurden. Die Ergebnisse zeigten, dass die entwickelte Software erfolgreich in der Lage war, Erklärungsbedarf auszulösen. Erste Erkenntnisse deuten darauf hin, dass sich neue Interaktionstechniken gut eignen, um Erklärungsbedarf zu generieren, während Erklärungsbedarf im Bereich der Sicherheit schwieriger zu initiieren war. Zudem wurden einige Rückmeldungen zur Benutzerfreundlichkeit des Systems gewonnen. Diese Mängel scheinen jedoch keinen Einfluss auf den ausgelösten Erklärungsbedarf gehabt zu haben, was darauf hinweist, dass

der Erklärungsbedarf durch die entwickelten Szenarien und nicht durch Benutzerfrustration hervorgerufen wurde.

Abschließend lässt sich sagen, dass das entwickelte System eine solide Grundlage für zukünftige Forschung im Bereich des Erklärungsbedarfs bietet und erfolgreich die gewünschten Anforderungen erfüllt hat.

7.2 Ausblick

Die Ergebnisse der Studie haben gezeigt, dass es noch Optimierungspotenzial hinsichtlich der Benutzerfreundlichkeit des Systems gibt. Die von den Teilnehmenden identifizierten Schwachstellen sollten in zukünftigen Iterationen behoben werden, um eine mögliche Frustration der Nutzenden auszuschließen.

Zudem wurde festgestellt, dass das System derzeit kein hinreichend effizientes Szenario zur Auslösung von Erklärungsbedarf in der Kategorie *Sicherheit* bietet. Es wird empfohlen, in zukünftigen Untersuchungen gezielt das Auslösen von Erklärungsbedarf in diesem Bereich zu erforschen und daraufhin die bestehenden Szenarien zu überarbeiten oder neue zu entwickeln, um sicherzustellen, dass das System auch diese Art von Erklärungsbedarf effektiv adressieren kann.

Darüber hinaus ist es notwendig, das System weiter zu testen, da die durchgeführte Studie aufgrund ihrer begrenzten Stichprobengröße keine statistisch signifikanten Ergebnisse liefert. Um belastbare Aussagen über die Effizienz des Systems bei der Auslösung von Erklärungsbedarf zu treffen, sollten umfangreichere Studien durchgeführt werden.

Anhang A

Fragebogen

Bachelorarbeit - Fragebogen

Ausfüllen nach Nutzung des Systems

1. Alter

2. Fachsemester

3. Wie bewerten Sie die Benutzerfreundlichkeit des Systems?

Markieren Sie nur ein Oval.

	1	2	3	4	5	
gut	◯	◯	◯	◯	◯	schlecht

4. Was ist Ihnen positiv oder negativ im Bezug auf die Benutzerfreundlichkeit aufgefallen?

Abbildung A.1: Fragebogen der Nutzerstudie - Seite 1

5. Falls Sie Erklärungsbedarf bei der Nutzung hatten, als wie hilfreich empfanden Sie die bereitgestellten Erklärungen?

Markieren Sie nur ein Oval.

	1	2	3	4	5	
sehr	◯	◯	◯	◯	◯	gar nicht

6. Wo hätten Sie sich eine genauere oder andere Erklärung gewünscht?

7. Hatten Sie Erklärungsbedarf bei dem Sie nicht die bereitgestellten Erklärungen genutzt haben ? Falls ja, aus wechem Grund?

8. Haben Sie zusätzliche Kommentare oder Anmerkungen zum System?

Abbildung A.2: Fragebogen der Nutzerstudie - Seite 2

Literaturverzeichnis

[1] J. Amann, A. Blasimme, E. Vayena, D. Frey, V. I. Madai, and P. Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[2] S. Bharati, M. R. H. Mondal, P. Podder, and U. Kose. Explainable artificial intelligence (xai) with ioht for smart healthcare: A review. *Interpretable Cognitive Internet of Things for Healthcare*, pages 1–24, 2023.

[3] S. Bromberger. *On what we know we don't know: Explanation, theory, linguistics, and how questions shape them*. University of Chicago Press, 1992.

[4] L. Chazette, W. Brunotte, and T. Speith. Exploring explainability: a definition, a model, and a knowledge catalogue. In *2021 IEEE 29th international requirements engineering conference (RE)*, pages 197–208. IEEE, 2021.

[5] L. Chazette and K. Schneider. Explainability as a non-functional requirement: challenges and recommendations. *Requirements Engineering*, 25(4):493–514, 2020.

[6] H. Deters, J. Droste, M. Fechner, and J. Klünder. Explanations on demand-a technique for eliciting the actual need for explanations. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, pages 345–351. IEEE, 2023.

[7] H. Deters, J. Droste, and K. Schneider. On the pulse of requirements elicitation: Physiological triggers and explainability needs. In *REFSQ Workshops*, 2024.

[8] J. Droste, H. Deters, M. Obaidi, and K. Schneider. Explanations in everyday software systems: Towards a taxonomy for explainability needs. *arXiv preprint arXiv:2404.16644*, 2024.

[9] C. Ebert. *Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten.* dpunkt. verlag, 2022.

[10] G. K. A. Ekaputeri. Concept and implementation of trigger-based explanations, 2024. Master's thesis, Leibniz University Hannover.

[11] M. Fechner. Konzept und implementierung einer komponente zur untersuchung des erklärungsbedarfs von software, 2023. Bachelor's thesis, Leibniz Universität Hannover.

[12] J. Y. Halpern and J. Pearl. Causes and explanations: A structural-model approach. part ii: Explanations. *The British journal for the philosophy of science*, 2005.

[13] International Organization for Standardization. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model.* International Organization for Standardization, Vernier, Geneva, Switzerland, ISO/IEC 25010:2023 edition, 2023.

[14] JetBrains. Compose multiplatform. `https://www.jetbrains.com/de-de/lp/compose-multiplatform/`, 2024. Accessed: 8. Juli 2024.

[15] M. A. Köhl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender. Explainability as a non-functional requirement. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 363–368. IEEE, 2019.

[16] U. Kuter and C. Yilmaz. Survey methods: Questionnaires and interviews. *Choosing Human-Computer Interaction (HCI) Appropriate Research Methods*, 2001.

[17] Z. S. Li, M. Sihag, N. N. Arony, J. B. Junior, T. Phan, N. Ernst, and D. Damian. Narratives: the unforeseen influencer of privacy concerns. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 127–139. IEEE, 2022.

[18] I. Nunes and D. Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27:393–444, 2017.

[19] Software Engineering (Leibniz Universität Hannover). Auszug aus uml 2.4.1. `https://www.pi.uni-hannover.de/fileadmin/pi/se/templates/UML-Poster-v2.1.pdf`, 2018. Accessed: 12. Juni 2024.

[20] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, et al. *Experimentation in software engineering*, volume 236. Springer, 2012.