

Eine gamifizierte Howto-App für Bachelorarbeiten

Tim-Pascal Lau

28.05.2018

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	4
1.2	Lösungsansatz	4
1.3	Zielsetzung	4
1.4	Aufgabenbeschreibung	5
1.5	Ausblick auf die Bachelorarbeit	6
1.5.1	Beschreibung der Kernkomponenten	6
1.5.2	Anforderungsanalyse	6
1.5.3	Architektur der Software	6
1.5.4	Implementierung	7
1.5.5	Validierung und Verifikation	7
1.5.6	Präsentation der Ergebnisse	7
2	Beschreibung der Kernkomponenten	8
2.1	Vorstellung des Frameworks Flutter	8
2.1.1	Flutter Systemarchitektur	8
2.1.2	Entwicklung von Betriebssystem übergreifenden Applikationen	8
2.1.3	Laufzeit-Performance	8
2.2	Gamification	8
2.2.1	Abgrenzung zu anderen Ansätzen	9
3	Untersuchung des Problembereiches	10
3.1	Hypothese	10
3.2	Identifikation der Interessengruppen	10
3.2.1	Studierende	10
3.2.2	AStA	10
3.2.3	Betreuer der eigenen Bachelorarbeit	12
3.2.4	Professoren	13
3.2.5	Präsidium	13
3.3	Wahl der Analysestrategie	14
3.4	Ergebnisse der Professoren	15
3.4.1	Ergebnisse der Einzelinterviews mit den Professoren	15
3.4.2	Ergebnisse der E-Mail Befragung der Professoren	19
3.5	Ergebnisse der Studierenden	19
3.5.1	Ergebnisse der Interviews der mit den Studierenden	19
3.5.2	Ergebnisse der Online-Befragung der Studierenden	19

3.6	Identifikation des Problembereiches	19
3.6.1	Ziele der Beteiligten (Warum muss es gelöst werden?)	19
3.6.2	Szenariensammlung (Wie könnte ein Softwaresystem helfen)	19
3.6.3	Analyse der Schwachstellen	19
3.7	Anforderungskatalog	19
3.8	Auswahl der Gamificationstrategie	19
4	Konzeptvorstellung der Applikation	20
4.1	Beschreibung der Software	20
4.1.1	Fortschrittsmanagement	20
4.1.2	Guide	21
4.1.3	Dashboard	21
4.1.4	Sonstige Softwareinhalte	22
4.2	Beschreibung der Gamification-Elemente	22
4.2.1	Anfangsphase Achievements	23
4.2.2	Bearbeitungsphase Achievements	23
4.2.3	Abschlussphase Achievements	23
4.2.4	App-Achievements Achievements	23
4.2.5	Fortschrittsverfolgung	23
5	Architektur der Software	24
5.1	Architekturbeschreibung	24
5.2	Entwurfsentscheidungen	24
5.3	Erweiterbarkeit der Software für andere Studiengänge	24
6	Implementierung	25
6.1	Umsetzung der Anforderungen	25
6.2	Entwurf der Benutzeroberfläche	25
7	Validierung und Verifikation	26
7.1	Softwaretests	26
7.2	Abdeckung der Softwareanforderungen	26
7.3	Ausführung der Usabilitytests	26
8	Präsentation der Ergebnisse	27
8.1	Lessons Learned	27
8.2	Abschlussbetrachtung	27
9	Literaturverzeichnis	28

1 Einleitung

Für Studierende im letzten Semester eines Bachelorstudiengangs, umfasst die wesentliche Prüfungsleistung das Verfassen einer Bachelorarbeit. Die Auseinandersetzung mit komplexen Problemstellungen stellt jedoch erfahrungsgemäß für viele Studierende eine große Herausforderung dar, welche sich aus dem erstmaligem Zusammenspiel von selbständigem und eigenverantwortlichen Arbeiten, sowie Problemlösen mittels erworbener Fach- und Methodenkenntnisse über einen längeren (in etwa dreimonatigen) Zeitraum ergibt.

1.1 Motivation

Im Verlauf des Studiums sollten Studierende folgendes Wissen und folgende Fähigkeiten erworben haben und zielgerichtet Anwenden können:

- Studiengang-spezifisches Grundlagenwissen
- Wissensansammlung über fachspezifische Methoden und deren Eigenschaften
- Fähigkeit, komplexe Probleme zu erkennen, zu strukturieren und systematisch mittels geeigneter Methoden zu bearbeiten

Es kommt im Kontext von Bachelorarbeiten dennoch oftmals zu Schwierigkeiten, das erworbene Wissen und Fähigkeiten zielgerichtet auf reale Probleme anzuwenden und deren Ergebnisse zusammenhängend zu dokumentieren.

1.2 Lösungsansatz

Ein möglicher im Rahmen dieser Bachelorarbeit zu verfolgender Lösungsansatz wäre es, eine Software zu entwickeln, welche unterstützend und wegweisend bei dem systematischen Vorgehen bei komplexen Problemstellungen fungieren könnte und diese den Studierenden zugänglich zu machen. Hierbei soll es nicht darum gehen, den Studierenden die eigentliche Arbeit abzunehmen, sondern vielmehr darum, Studierende hinsichtlich Vorgehen und Methodenauswahl zielgerichtet zu unterstützen.

1.3 Zielsetzung

Mit einem solchen Ansatz soll es Studierenden ermöglicht werden, ihr gelerntes Wissen durch Fokussierung bestimmter Aufgaben und Zusammenhänge im Rahmen ihres

eigenen Bachelorprojekts auf die Realität zu übertragen und somit einen motivierenden, sowie gleichermaßen fordernden Rahmen zu schaffen, um ihr Bachelorprojekt erfolgreich abzuschließen.

1.4 Aufgabenbeschreibung

Im Rahmen dieser Bachelorarbeit soll hierfür eine mobile Applikation entwickelt werden, die den Studierenden während der Dauer der Bachelorarbeit kontinuierlich "begleitet". Dabei sollen Gamificationansätze realisiert werden, welche motivierend im bei der Bearbeitung und dem Vorgehen der eigenen Bachelorarbeit wirken sollen. Dies soll beispielhaft für den Studiengang Informatik/Softwareentwicklung erfolgen. Eine Erweiterbarkeit für andere Studiengänge ist hierbei jedoch konzeptionell vorzusehen. Primäre Funktionen der Software ist Studierenden bei den folgenden Aufgaben begleitend zu unterstützen und fortwährend zu motivieren am Ball zu bleiben":

- Brainstorming (zur Unterstützung der Ideenfindung für Bachelorarbeiten)
- Recherche und Literaturverwaltung
- Gliederung (unterschiedlicher Kategorien von Bachelorarbeiten, zum Beispiel mittels bewehrter Templates)
- Zeitplanung/Fortschrittsverfolgung, sowie Erinnerungs- und Benachrichtigungsfunktion
- Problem-orientierte Anforderungsanalyse und deren Dokumentation
- Problem-orientierte Methoden- und Tool-/Frameworkselektion und deren Dokumentation
- Methoden-spezifische Aufbereitung von Ergebnissen
- Problem-orientierte Nachweisführung und deren Dokumentation

Die Applikation soll mittels Flutter für Android und iOS entwickelt werden. Dabei soll erhoben werden, inwiefern sich Flutter für die Entwicklung solcher Apps eignet (Lessons Learned). Die im Rahmen der Aufgabenbeschreibung entstandenen Anforderungen werden durch die folgenden Teilaufgaben spezifiziert:

- Detaillierte Anforderungsanalyse oben angegebener Funktionen. Hierbei sind Studenten und Professoren des Studiengangs Informatik/Softwaretechnik geeignet einzubeziehen und relevante Literatur (insbesondere zu Gamification und Methoden der Informatik und des Softwareengineering) zu berücksichtigen.
- Architekturentwurf der Anwendung (Erweiterbarkeit für andere Studiengänge ist konzeptionell vorzusehen)
- Implementierung der Anwendung

- Die Funktionsfähigkeit der App soll mittels Softwaretests geeignet nachgewiesen werden.
- Die Nutzbarkeit der App soll systematisch evaluiert werden. Hierbei sind Studenten und Professoren des Studiengangs Informatik/Softwaretechnik geeignet einzubeziehen.
- Dokumentation der oben angegebenen Schritte inklusive Bewertung der Nutzbarkeit des Frameworks Flutter für solche Arten von Apps.

1.5 Ausblick auf die Bachelorarbeit

In der folgenden Dokumentation der Bachelorarbeit werden verschiedene aufeinander aufbauende Prozesse, Teilschritte und Ergebnisse der Softwareentwicklung dokumentiert sein. Hierbei liegt die Priorität vor allem bei dem Pflegen der Nachvollziehbarkeit der dargestellten Informationen durch aufeinander aufbauende Kapitel und dem reflektieren der eigenen Gedankengänge.

1.5.1 Beschreibung der Kernkomponenten

Der Nachvollziehbarkeit halber empfiehlt es sich, Grundkenntnisse über die Basiskomponenten, wie dem Flutter Framework zu besitzen. Sollte dies nicht der Fall sein, so lassen sich im Kapitel **Beschreibung der Basiskomponenten** die nötigen Informationen nachlesen.

1.5.2 Anforderungsanalyse

Das Kapitel **Anforderungsanalyse** stellt in detaillierter Ausführung und Beschreibung die Prozesse der Anforderungsermittlung und deren Auswertung, sowie Definition dar und legt somit wichtige Grundlagen und Anforderungen an die Software fest. Weiterhin werden im Laufe des Kapitels Einblicke in Strategien und Gedankengänge ermöglicht, welche zusätzliche Anhaltspunkte für die Nachvollziehbarkeit der weiteren Kapitel beitragen können.

1.5.3 Architektur der Software

Die Entscheidung eine mobile Applikation für Android und iOS mittels Flutter zu entwickeln, definiert bereits frühzeitig verschiedene Möglichkeiten und Pflichten, welche im Kapitel **Anforderungsmanagement** unter Zunahme anderer erhobener Anforderungen detailliert dokumentiert und beschrieben werden. Alle nötigen Informationen zur Softwarearchitektur, zu den Entwurfsentscheidungen, sowie der Berücksichtigung der Erweiterbarkeit der Software, werden im Kapitel **Architektur der Software** behandelt.

1.5.4 Implementierung

Informationen zur detaillierten Implementierung der in der Aufgabenbeschreibung definierten Funktionen, sowie die Umsetzung der Benutzeroberfläche, werden im Kapitel **Implementierung der Softwarefunktionen** behandelt. Unter Einbezug beispielhafter Codeauszüge werden hier die Funktionsweisen der Software aufgeführt und beschrieben.

1.5.5 Validierung und Verifikation

Die Nachweisführung der Softwareanforderungen, der Usability-Anforderungen, sowie die Auswertung der Nützlichkeit der Verwendung des Frameworks Flutter bei der Entwicklung dieser App lassen sich im Kapitel **Validierung und Verifikation** nachlesen.

1.5.6 Präsentation der Ergebnisse

Abschließend folgt im Kapitel **Präsentation der Ergebnisse** eine Zusammenfassung der erreichten Ergebnisse und eine Reflexion der Teilschritte, sowie die Abschlussbetrachtung des gesamten Projekts.

2 Beschreibung der Kernkomponenten

2.1 Vorstellung des Frameworks Flutter

Flutter ist ein von Google entwickeltes opensource Framework, welches auf die Entwicklung mobiler 2D-Applikationen für Android- und iOS-Betriebssysteme ausgelegt ist. Beworben wird Flutter durch das Hervorheben der Einfachheit der Benutzung, die schnell zu erreichenden Fortschritte bei der Implementierung von Softwarefunktionen, sowie den Gestaltungsmöglichkeiten der Benutzeroberfläche und den hochqualitativen Ergebnissen.

2.1.1 Flutter Systemarchitektur

Das Flutter Framework besteht aus drei verschiedenen Basiskomponenten, welche im folgenden Abschnitt kurz erläutert werden.

- **Flutter Engine**

Die C/C++ basierte Flutter Engine, setzt sich aus verschiedenen Kerntechnologien zusammen. Zum einen die open source 2D Graphics Library Skia[9], welche seit 2005 zu Google gehört und zum anderen die Dart Virtual Machine.

- **Foundation Library**

Die Foundation Library, welche in Dart geschrieben wurde, stellt Basisklassen und -funktionen zur Verfügung und dient der Konstruktion von Applikationen mittels Flutter

- **Design-specific Widgets**

Das Flutter Framework stellt zwei verschiedene Arten von Widgets zur Verfügung, welche zugehörig zu den jeweiligen Design Sprachen von Google Material Design[7], welche 2014 entwickelt wurde und die iOS Design kopierende Design Sprache Cupertino[3].

...folgt

2.1.2 Entwicklung von Betriebssystem übergreifenden Applikationen

2.1.3 Laufzeit-Performance

2.2 Gamification

(Vergleiche [10])

2.2.1 Abgrenzung zu anderen Ansätzen

3 Untersuchung des Problembereiches

3.1 Hypothese

Die Auseinandersetzung mit komplexen Problemstellungen, wie die als abschließende Prüfungsleistung des Studiums zu erarbeitende Bachelorarbeit, stellt erfahrungsgemäß für viele Studierende eine große Herausforderung dar. Diese Herausforderung ergibt sich aus dem erstmaligen Zusammenspiel von selbstständigem und eigenverantwortlichem Arbeiten, sowie Problemlösen mittels erworbener Fach- und Methodenkenntnisse über einen längeren (in etwa dreimonatigen) Zeitraum. Trotz des Verlaufs des Studiums, des angeeigneten Wissens und der somit zahlreich erworbenen Fähigkeiten, kommt es im Kontext von Bachelorarbeiten dennoch oftmals zu Schwierigkeiten, diesen Zusammenhang auf reale Probleme abzubilden und zu dokumentieren.

3.2 Identifikation der Interessengruppen

In diesem Kapitel werden die identifizierten Interessengruppen dokumentiert und in Form von Steckbriefen in den verschiedenen Kategorien **Einfluss**, **Einstellung**, **Erwartungen**, sowie **Bemerkungen** beschrieben.

3.2.1 Studierende

Einfluss: Hoch

Einstellung: Positiv

Erwartungen:

Optimales Ergebnis und weniger „Fallstricke“ während der Bearbeitung der Bachelorarbeit, sowie ein allgemein „nicht zu überfordernder“ Durchlauf der Bachelorarbeit durch weniger Unwissenheit vor/während der Bachelorarbeit

Bemerkungen:

Hauptzielgruppe des Projekts/Spätere potenzielle Anwender der Applikation

3.2.2 AStA

Einfluss: Hoch

Einstellung: Positiv

Erwartungen:

... folgt

Bemerkungen:

Stellt das Sprachrohr der Studierenden dar und vertritt somit deren Meinung, Interessen und Ziele

3.2.3 Betreuer der eigenen Bachelorarbeit

Einfluss: Hoch

Einstellung: Positiv

Erwartungen:

Eine Steigende Bereitschaft/Motivation der Studierenden im Rahmen des Bachelor-Seminars Beiträge zu erbringen, sowie eine steigende Qualität der Kommunikation mit dem Betreuer und der damit zusammenhängenden Qualität der Bearbeitung der Bachelorarbeit und des Ergebnisses.

Detaillierte Angabe der Erwartungen:

- Zielgerichteter(er) Methoden-Einsatz von allen Methoden, die im Informatik/SWT-Studium gelehrt werden
- Zielgerichtete(re) Vorbereitung auf Besprechungen mit dem Betreuer
- Bessere Lesbarkeit von Abschlussarbeiten
- Bessere „rote Fäden“ in Abschlussarbeiten
- Bessere Zeitplanung
- Systematische(re) Problemanalyse und Anforderungserhebung und deren Dokumentation
- Bessere Architekturentwicklung und deren Dokumentation
- Systematische(re) Nachweisführung und deren Dokumentation

Bemerkungen:

Führt das Bachelorseminar und hat somit direkten Kontakt mit der Zielgruppe, trägt wichtige Erfahrungswerte mit sich, woran es üblicherweise bei der Fertigstellung von Bachelorarbeiten mangelt

3.2.4 Professoren

Einfluss: Hoch

Einstellung: Positiv

Erwartungen:

... folgt

Bemerkungen:

Tragen Erfahrungswerte durch das Betreuen und Bewerten von Bachelorarbeiten, kennt die Probleme der Studenten und hat detaillierten Einblick in die Schwierigkeiten der Zielgruppe

3.2.5 Präsidium

Einfluss: Gering

Einstellung: Positiv

Erwartungen:

- Wenn Applikation positiven Einfluss auf die Ergebnisse von Abschlussarbeiten, kann die Fachhochschule Lübeck ihren Ruf festigen werden
- Erhöhung der Wettbewerbsfähigkeit durch bessere Leistung/bessere Abschlüsse der Studierenden

Bemerkungen:

Machtpromotor

3.3 Wahl der Analysestrategie

Um den Ist-Zustand zu ermitteln und somit eine Analysegrundlage zu erschaffen, werden Personengruppen der Fachhochschule Lübeck, durch verschiedene Befragungs- und Analysemethoden in das Projekt miteinbezogen. Dies soll einen detaillierten Einblick in die Sichtweisen der unterschiedlich beteiligten Personen und Interessengruppen ermöglichen und somit eine Grundlage für das Verständnis der aktuellen Situation bilden.

Als primäre Einflussgeber wurden in diesem Rahmen die Gruppe der Professoren, sowie die Gruppe der Studierenden identifiziert. Diese Entscheidung wurde aufgrund der im Rahmen einer Bachelorarbeit existierenden unterschiedlichen Sichtweisen, sowie Erfahrungsständen von Betreuern und Bacheloranden getätigt.

Professoren

Für die Interessengruppe der Professoren aus dem Fachbereich Informatik, ist als Grundlage der Datenerhebung einerseits das Durchführen von Einzelinterviews mit einer ausgewählten Gruppe von Professoren vorgesehen, während eine weitere Gruppe von Professoren schriftlich per E-Mail befragt wird.

Dies bietet sowohl den Zugriff auf die unmittelbaren Erfahrungen der einzelnen Professoren als Spezialisten in den jeweiligen Fachgebieten, als auch auf die Erfahrungen der Professoren in der Position eines Betreuers und Ansprechpartners für Bacheloranden. Durch das Durchführen von Einzelinterviews wird ermöglicht, die Erwartungen seitens der Professoren an die Bacheloranden im Detail zu identifizieren und die, in dieser Hinsicht priorisierten inhaltlichen und methodischen Aspekte bei der Bearbeitung einer Bachelorarbeit herauszuarbeiten. Die Aufteilung auf Einzelinterviews und E-Mail Befragungen bietet den Vorteil, beide Strategien simultan zu verfolgen und nach Abschluss der Datenerhebung sowohl die detaillierten Einzelinterviews, als auch die oberflächlicher ausfallenden E-Mail Antworten in bereits dokumentierter Form vorliegen zu haben, um diese dann auszuwerten.

Studierende

Für die Studierenden aus dem Studiengang Informatik/Softwareentwicklung, ist es aufgrund der hohen Menge an Personen und den zu betreibenden Aufwand vorgesehen, priorisiert Gruppen- und nur in Ausnahmefällen auch Einzelinterviews zu durchzuführen. Durch das Durchführen von Gruppeninterviews, sollen die Studierenden zu Diskussionen angeregt sein, welche von dem Leiter des Interviews durchaus auch motiviert werden können.

Miteinbezogen werden vorzugsweise alle Studierende der oberen Semester, unabhängig davon, ob sie sich noch in der Phase vor Beginn der Bearbeitung, während der Bearbeitung oder nach Abschluss der Bearbeitung der Bachelorarbeit befinden. Hierbei ist wichtig zu beachten, dass die verschiedenen Erfahrungsstände in den Gruppeninterviews nicht aufeinandertreffen, da dieser Umstand somit zu einer gegenseitigen Beeinflussung führen könnte und wichtige Informationen nicht gewonnen werden könnten. Die verschiedenen zu extrahierenden Sichtweisen auf die Bearbeitung der Bachelorarbeit können auf diese Weise differenziert betrachtet und analysiert werden.

Des weiteren wird eine Online-Umfrage den Teil der Datenerhebung darstellen, der quantitative Ergebnisse erzielt soll und deren Aussage somit eine nicht durch die Interviews abgebildete Menge darstellt.

3.4 Ergebnisse der Professoren

3.4.1 Ergebnisse der Einzelinterviews mit den Professoren

Im Folgenden sind die gewonnenen Eindrücke und Kenntnisse der Einzelinterviews mit den Professoren des Fachbereichs Informatik durch Themenkategorien geordnet und in zusammengefasster Form dokumentiert.

Es haben insgesamt sechs Professoren an den Einzelinterviews teilgenommen, wobei es für fünf Interviews gestattet wurde, eine Tonaufzeichnung anzufertigen.

Allgemeine Informationen

Der zeitliche Rahmen der fünf aufgezeichneten Interviews erstreckt sich über einen Zeitraum von etwa 30 bis 45 Minuten. Für die Auswertung der aufgezeichneten Interviews wurden die Aussagen der Interviewpartner, auf Grundlage der vorliegenden Audioaufnahmen, unter Berücksichtigung des Kontextes aufbereitet und werden nachfolgend dargestellt.

Ein weiteres Einzelinterview, welches nicht aufgezeichnet wurde, erstreckte sich über einen Zeitraum von 75 Minuten. Für dieses Interview wurden lediglich begleitende Feldnotizen angefertigt. Diese Feldnotizen wurden im Anschluss des Interviews aufbereitet und werden in der folgenden Beschreibung einfließen.

Die gewählten Kategorien ergeben sich aus dem gewählten Auswertungsverfahren der qualitativen Inhaltsanalyse nach Mayring[8] und basieren in diesem Kontext auf die gemäß des Verfahrens herausgearbeiteten Codings.

Art der Arbeit

Im Laufe der Interviews wurden verschiedene Typen von Arbeiten versucht zu identifizieren. Dabei geht es vor allem darum, die Vielfalt der typischen Arbeiten des Studiengangs Informatik/Softwareentwicklung zu erfassen und somit einen Überblick über die Situation zu bekommen.

Als im allgemeinen auftretenden Arten der Arbeit wurden die Klassen **Entwickelnde Arbeit** und **Evaluierende Arbeit** identifiziert. Weiterhin gibt es auch **reine Literaturarbeiten**, welche in dem Studiengang Informatik/Softwareentwicklung jedoch nicht oder nur in einem sehr geringen Vorkommen auftreten.

Es folgt eine stichpunktartige Ausführung der gewonnenen Erkenntnisse:

- **Konstruktiv/Entwickelnd - Durchlauf des Softwareentwicklungszyklus**

- **Anforderungsanalyse**

Unterschiedlich komplex, je nach individueller Aufgabenstellung und Rahmenbedingungen. Ausschlaggebend hierfür ist vor allem, ob es sich um eine interne Arbeit an der Fachhochschule oder eine externe Arbeit in einem Unternehmensumfeld handelt, bei der gegebenenfalls die Anforderungen schon definiert sind. Sollten bereits Anforderungen existieren, so ist das Infragestellen dieser Anforderungen häufig Bestandteil der Aufgabe.

- **Entwurf einer Softwarearchitektur**

Je nach individueller Aufgabenstellung und Rahmenbedingungen können auch hier große Unterschiede in der Bearbeitung liegen. Ein wichtiger Orientierungspunkt hierbei ist das Vorhandensein von schon existierenden Softwareprodukten, welche entweder erweitert oder ersetzt werden könnten. Dies ist häufig bei externen Bachelorarbeiten zu erwarten. Eine weitere Möglichkeit ist, dass es keine bereits vorhandene Softwarelösung gibt, sondern diese von Grund auf entwickelt werden soll.

- **Implementierung und Evaluation eines Softwareprototyps**

Es wird betont, dass es zumindest bei internen Arbeiten nicht unbedingt darum geht, nach Abschluss der Bearbeitung der Bachelorarbeit, ein Softwareprodukt vorliegen zu haben, was für die Markteinführung geeignet ist. Es wird deshalb oftmals von einer prototypischen Implementierung gesprochen. Bei externen Arbeiten kann dies jedoch, im Sinne der Unternehmen oder Organisationen, durchaus den Zielanforderungen entsprechen. Im Rahmen der Evaluation gibt es eine Vielzahl von Möglichkeiten, die je nach Schwerpunkt der Arbeit den Fokus auf verschiedene Ziele legt.

Softwaretesting findet in den entwickelnden Arbeiten bei Implementierung grundsätzlich statt, jedoch gibt es eine Vielzahl an weiteren Aspekten, wie Usability-Tests und Evaluation der Nützlichkeit einer Software, welche je nach Themenschwerpunkt untersucht werden können.

- **Sonstige Anmerkungen**

Typische Aufgabenstellungen könnten sein:

- 1 Entwicklung einer mobilen Applikation zur Interpretation von Bildmaterial.
- 2 Entwicklung einer mobilen Applikation zur Steigerung der Bereitschaft bei Senioren und Seniorinnen, Fitnessaktivitäten auszuführen unter Einbezug von Gamificationelementen.
- 3 Entwicklung einer Software zur Optimierung der täglichen Arbeitsabläufe in Unternehmen A.

Ein weiterer wichtiger Aspekt sind die möglichen Interessenunterschiede zwischen dem externen Unternehmen und dem internen Betreuer der Fachhochschule, welche einen Einfluss auf die Inhalte der Bachelorarbeit haben können. Externe Unternehmen sind tendenziell eher an dem resultierenden Ergebnis interessiert, während die internen Betreuer darüber hinausgehend einen hohen Wert auf nachvollziehbare Methodik, Herangehensweise, sowie Planung und dem sauberen wissenschaftlichen Arbeiten legen und somit ein hohes Interesse an dem Gesamtprozess haben.

- **Analytisch/Evaluierend - Vergleich, Auswertung und/oder Nachweis eines Aufgabengegenstandes**

- **Erstellen eines Kriterienkatalogs**

Messbare Kriterien stellen die Grundlage des Experiments/der Auswertung dar und werden meistens bereits zu Beginn, in Form eines Kriterienkatalogs festgelegt. Ein wichtiger Aspekt ist hierbei vor allem die Frage, wie die Kriterien gemessen werden können. Die damit zusammenhängende Aussagekraft der Kriterien soll hierbei hinterfragt und diskutiert werden.

- **Aufbau des Experiments**

Der Aufbau des Experiments hängt stark von dem Anwendungsfall, der Zielstellung und des Untersuchungsgegenstandes ab. Hierbei werden die gewählten Strategien und Rahmenbedingungen zusammenhängend erläutert und beschrieben, um ein nachvollziehbares Fundament für die Durchführung des Experiments zu erschaffen, Abhängigkeiten darzustellen und Besonderheiten zu klären.

Im Rahmen der Untersuchung werden beispielsweise Datenerhebungsmethoden wie Online-Umfragen und Interviews geführt.

Sollten Vergleiche verschiedener Technologien Gegenstand der Arbeit sein, so werden zum Beispiel auch Fallstudien durchgeführt.

Bei Auswertung vorhandener Technologien sind oftmals auch Machbarkeitsstudien zentraler Bestandteil der Arbeit.

- **Durchführung des Experiments**

Je nach Ausrichtung der Aufgabenstellung und des Themengebietes können hier unterschiedliche Ansätze ausgeprägt und beschrieben sein, welche zuvor im Aufbau des Experiments dargelegt wurden.

- **Evaluation und Ergebnisauswertung**

Die Evaluation der Ergebnisse und die damit zusammenhängende Diskussion ist der zentrale Bestandteil der Arbeit. Alle vorherig getätigten Entscheidungen und Strategien werden nun zusammenhängend mit der Problemstellung ausgewertet und weiterhin diskutiert.

- **Sonstige Anmerkungen**

Typische Aufgabenstellungen könnten sein:

- 1 Evaluation der Gesichtserkennungsdienste von Unternehmen A, Unternehmen B und Unternehmen C.
- 2 Untersuchung des Verhaltens einer neuen Technologie A, im Vergleich mit einer alten Technologie B.
- 3 Datenbankanalyse unter Anwendung von Machine-Learning-Algorithmen

- **Reine Literaturarbeiten**

Reine Recherchierende Arbeiten finden in dem Studiengang Informatik/Softwareentwicklung aufgrund dem geringen Interesse seitens der Studierenden kaum statt und werden aus Gründen der Vollständigkeit lediglich erwähnt und nicht ausgeführt.

Erwartungen an den Bacheloranden

Im Laufe der Interviews wurden die Professoren hinsichtlich Ihrer Erwartungen an die Bacheloranden befragt und haben in diesem Rahmen häufig gleiche oder ähnliche Punkte ausgeführt. Aus diesem Grund werden im folgenden Verlauf die Meinungen der befragten Professoren aus der Sicht als Betreuer, unter den jeweiligen Aspekten als zusammengefasstes Meinungsbild wiedergegeben.

Es folgt die Ausführung der gewonnenen Erkenntnisse:

-

Häufig Auftretende Probleme bei der Bearbeitung der Bachelorarbeit

Es stellte sich im Verlauf des Interviews heraus, dass unterschiedliche Studierende immer wieder mit gleichen oder ähnlichen Problemen zu kämpfen haben. Im folgenden Verlauf werden diese genannten Probleme in aufbereiteter Form stichpunktartig beschrieben.

Es folgt die Ausführung der gewonnenen Erkenntnisse:

-

Die Applikation - Erwartungen, Chancen und Risiken

In jedem Interview bekamen die Professoren abschließend die Möglichkeit, ihre Erwartungen an eine solche Applikationen auszuführen und besonders auf die, aus ihrer Sicht mögliche Risiken und Chancen einzugehen und diesbezüglich auch Anmerkungen oder Empfehlungen zu tätigen. Diese Anmerkungen werden im folgenden Verlauf zusammengefasst dargestellt.

Es folgt die Ausführung der gewonnenen Erkenntnisse:

-

3.4.2 Ergebnisse der E-Mail Befragung der Professoren

3.5 Ergebnisse der Studierenden

3.5.1 Ergebnisse der Interviews der mit den Studierenden

3.5.2 Ergebnisse der Online-Befragung der Studierenden

3.6 Identifikation des Problembereiches

Als äußerst interessanter Aspekt, wurden die stark unterschiedlich ausfallenden Sichtweisen, sowie Erwartungen an den Prozess der Bearbeitung der Bachelorarbeit der Studierenden und der Professoren identifiziert.

3.6.1 Ziele der Beteiligten (Warum muss es gelöst werden?)

3.6.2 Szenariensammlung (Wie könnte ein Softwaresystem helfen)

3.6.3 Analyse der Schwachstellen

3.7 Anforderungskatalog

3.8 Auswahl der Gamificationstrategie

- Zielgruppe Studenten oder INFORMATIKStudenten? Erweiterbarkeit beachten! - Informatikstudenten wenig Interesse an sozialem Anschluss - Motivationstheorie (Maslow/McCelland)
- Motivation durch ETWAS ERREICHEN - Emotionen ERNSTHAFTER SPAß - Veränderungen erreichen, etwas schaffen - Soziale Spieler: Eher nicht Informatiker. - Erfolgsorientierter Spieler (Orientiert sich an zielen, die im Spiel zu erreichen sind) - Sammeln von Punkten, Aufsteigen im Level, Bestmarken -> Badges für Meilensteine, Unteraufgaben,

4 Konzeptvorstellung der Applikation

4.1 Beschreibung der Software

Die Applikation ist in verschiedene Kern-Softwareabschnitten unterteilt. Diese Abschnitte lassen sich in folgende namensgebende Aufgabenbereiche unterteilen, welche sich an den erhobenen Anforderungen und der Aufgabenstellung orientieren (siehe Kapitel 3.7).

Es folgt eine stichpunktartige Auflistung der Kernfeatures des Tools, welche im Laufe dieses Kapitels ausführlich beschrieben werden:

- **Fortschrittsmanagement**
- **Guide**
- **Dashboard**
- **Sonstige Softwareinhalte**

4.1.1 Fortschrittsmanagement

Das Fortschrittsmanagement ist ein Tool, welches das Anlegen, Planen und Verwalten von Meilensteinen, sowie Unteraufgaben der Meilensteine ermöglichen soll.

- Auf Basis agiler Softwareentwicklung. Arbeitspaketbildung. - Nur einen Meilenstein pro Woche - Timeboxing + Smart Methode - Vordefinierte Aufgaben? - Beginn und Ende eingeben - Meilensteine anpassen - Vordefinierte Schablone? Wochenweise Meilensteine mit Unteraufgaben.

Es folgt eine stichpunktartige Auflistung und Beschreibung der Kernfeatures des Tools:

- **Ein Zeitstrahl als Grundlage**
Der Zeitstrahl bietet die Arbeitsgrundlage des Fortschrittsmanagement-Tools. Hier können wochenweise Meilensteine mit Unteraufgaben angelegt werden.
- **Erstellung und Verwaltung von Meilensteinen**
Es können Meilensteine erstellt, bearbeitet, verschoben und entfernt werden.
- **Erstellung und Verwaltung von Unteraufgaben**
Jeder Meilenstein enthält Unteraufgaben, die von dem Benutzer einem Meilenstein zugeordnet, bearbeitet, verschoben und gelöscht werden können.

4.1.2 Guide

Der Guide stellt den Teil der Applikation dar, der die Bereitstellung von Hinweisen, Tipps und weiteren hilfreichen Informationen zur Bearbeitung der Bachelorarbeit abdecken soll. Dieser Inhalt ergibt sich aus der Verwendung des, von der Fachhochschule Lübeck ausgehändigten Dokuments, welches als Ratgeber bei der Erstellung von Bachelorarbeiten funktionieren und somit die speziellen Anforderungen der FH-Lübeck berücksichtigen soll (vgl. [1, Kapitel 1]).

In dieser Hinsicht soll der Guide als genereller Anlaufpunkt funktionieren, der beispielsweise eine Hilfestellung für Bacheloranden darstellen.

Die Bandbreite der Hinweise, Tipps und Informationen sollen somit den gesamten Verlauf der Bachelorarbeit abdecken und lassen sich in diesem Umfang in folgende Bereiche und Inhalte zerlegen, welche sich an dem Leitfaden der FH-Lübeck[1] zur Erstellung einer Bachelorarbeit orientieren.

Es folgt eine eigene Interpretation und Beschreibung der Unterteilung der Themengebiete:

- **Allgemeine Informationen**

Dieser Abschnitt beinhaltet die allgemeinen Informationen, welche vor allem zur Klärung der Formalien bei der Bearbeitung der Abschlussarbeit wichtig sind.

- **Strukturierung der Abschlussarbeit**

Der Abschnitt befasst sich näher mit dem Aufbau der Bachelorarbeit und stellt vor allem Informationen und Empfehlungen bereit, welche sich auf die Strukturierung und die Bedeutung der einzelnen Kapitel der Bachelorarbeit beziehen.

- **Hinweise zum Schreiben der Arbeit**

In diesem Abschnitt werden tiefgehende Informationen und Empfehlungen behandelt, welche sich auf die praktische Umsetzung des Schreibens im Detail beziehen. Beispielhaft hierfür sind der Schreibstil, die Verwendung von Zeilen oder die Einbindung von Bildern, Tabellen und Programmcode.

- **FAQ**

Dieses Kapitel behandelt oft gestellte Fragen und verdeutlicht die Informationen in der Form eines Frage-Antwort Schemas.

4.1.3 Dashboard

Das Dashboard stellt den Ausgangspunkt und Hauptbildschirm der Applikation dar und trägt somit die Aufgabe, aktuelle Informationen für den Nutzer aufbereitet anzuzeigen, sowie die Funktionen der Applikation auf einfache Weise zugänglich zu machen. Von diesem Punkt aus soll der Benutzer die verschiedenen Softwareabschnitte öffnen können,

weshalb es eine essenzielle Eigenschaft des Dashboards ist, die Inhalte übersichtlich, strukturiert und visuell ansprechend darzustellen.

Es folgt eine stichpunktartige Auflistung und Beschreibung der Inhalte:

- **Dynamische Anzeige des Fortschritts**

Der bisher geleistete Gesamtfortschritt wird verdeutlicht, indem zu sehen ist, wie viele Meilensteine bisher abgeschlossen wurden, in Relation zu den Gesamtmeilensteinen.

- **Dynamische Anzeige der Wöchentlichen Aufgaben**

Der bisher geleistete Wochenfortschritt wird verdeutlicht, indem zu sehen ist, wie viele Aufgaben in der jetzigen Woche erfüllt wurden, in Relation zu den vorher definierten gesamten Aufgaben für diese Woche.

- **Dynamische Anzeige der letzten erreichten Achievements**

Die Anzeige zeigt die letzten n Achievements an. Zugehörig sind in diesem Fall die Darstellung des Typs des Achievements in Form einer Medaille, sowie den Titel der Herausforderung.

4.1.4 Sonstige Softwareinhalte

Über die eigentlichen individuellen Funktionalitäten hinaus, sind weitere Grundfunktionalitäten vorhanden, welche unter den Punkt **Sonstige Softwareinhalte** fallen.

Es folgt eine stichpunktartige Auflistung und Beschreibung der Inhalte:

- **Menü**

Das Menü zeigt die existierenden Funktionalitäten und Softwareabschnitte in einer klassischen gelisteten Menüstruktur.

- **Einstellungen**

Die Einstellungen beinhalten Optionen, die sich auf die Eigenschaften der Applikation auswirken. Beispielsweise das Verändern des Designs.

4.2 Beschreibung der Gamification-Elemente

Auf Basis der Problemanalyse konnten bereits ...

- Einführung in die Software? Durch Video? Etc.

4.2.1 Anfangsphase Achivements

4.2.2 Bearbeitungsphase Achivements

4.2.3 Abschlussphase Achivements

4.2.4 App-Achivements Achivements

4.2.5 Fortschrittsverfolgung

- Weekly Tasks - Abschlussbericht pro Woche?

5 Architektur der Software

5.1 Architekturbeschreibung

5.2 Entwurfsentscheidungen

5.3 Erweiterbarkeit der Software für andere Studiengänge

6 Implementierung

6.1 Umsetzung der Anforderungen

6.2 Entwurf der Benutzeroberfläche

7 Validierung und Verifikation

7.1 Softwaretests

7.2 Abdeckung der Softwareanforderungen

7.3 Ausführung der Usabilitytests

8 Präsentation der Ergebnisse

8.1 Lessons Learned

8.2 Abschlussbetrachtung

9 Literaturverzeichnis

Literatur

- [1] (Prof. Dr. rer. nat. Andreas Hanemann. “Anleitung zum Erstellen einer Abschlussarbeit”. Herausgegeben an Studenten der Fachhochschule Lübeck, die das Bachelorseminar besuchen.
- [2] *Build beautiful native apps in record time.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/>.
- [3] *Cupertino (iOS-style) Widgets.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/widgets/cupertino/>.
- [4] *Foundation Library.* (Zugriff Juni 3, 2018). URL: <https://docs.flutter.io/flutter/foundation/foundation-library.html>.
- [5] *Introduction.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/faq/#introduction>.
- [6] *Material Components Widgets.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/widgets/material/>.
- [7] *Material Design.* (Zugriff Juni 3, 2018). URL: <https://material.io/design/>.
- [8] Philipp Mayring. *Qualitative Inhaltsanalyse - Grundlagen und Techniken.* Langensalza: Beltz, 2015. ISBN: 978-3-407-25730-7.
- [9] *Skia Graphics Library.* (Zugriff Juni 3, 2018). URL: <https://skia.org/>.
- [10] Susanne Strahringer und Christian Leyh. *Gamification und Serious Games - Grundlagen, Vorgehen und Anwendungen.* Berlin Heidelberg New York: Springer-Verlag, 2017. ISBN: 978-3-658-16742-4.
- [11] *Technical Overview.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/technical-overview/>.
- [12] *Technology.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/faq/#what-technology-is-flutter-built-with>.
- [13] *Welcome to the Engine Wiki.* (Zugriff Juni 3, 2018). URL: <https://github.com/flutter/engine/wiki>.