

# **Eine gamifizierte Howto-App für Bachelorarbeiten**

Tim-Pascal Lau

28.05.2018

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
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
<b>2</b>	<b>Beschreibung der Kernkomponenten</b>	<b>8</b>
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	Dart . . . . .	8
<b>3</b>	<b>Anforderungsanalyse</b>	<b>9</b>
3.1	Identifikation des Ist-Zustands . . . . .	9
3.1.1	Problemabgrenzung (Welches Problem soll gelöst werden?) . . . .	9
3.1.2	Kontext des Problems (Wo liegt das Problem?) . . . . .	9
3.1.3	Ziele der Beteiligten (Warum muss es gelöst werden?) . . . . .	9
3.1.4	Szenariensammlung (Wie könnte ein Softwaresystem helfen) . . . .	9
3.2	Identifikation des Soll-Zustands . . . . .	9
3.3	Analyse der Schwachstellen . . . . .	9
3.4	Anforderungskatalog . . . . .	9
3.5	Lastenheft . . . . .	9
3.6	Pflichtenheft . . . . .	9
3.6.1	Funktionale Anforderungen . . . . .	9
3.6.2	Nicht funktionale Anforderungen . . . . .	9
<b>4</b>	<b>Architektur der Software</b>	<b>10</b>
4.1	Architekturbeschreibung . . . . .	10
4.2	Entwurfsentscheidungen . . . . .	10

4.3	Erweiterbarkeit der Software für andere Studiengänge . . . . .	10
<b>5</b>	<b>Implementierung</b>	<b>11</b>
5.1	Umsetzung der Anforderungen . . . . .	11
5.2	Entwurf der Benutzeroberfläche . . . . .	11
<b>6</b>	<b>Validierung und Verifikation</b>	<b>12</b>
6.1	Softwaretests . . . . .	12
6.2	Abdeckung der Softwareanforderungen . . . . .	12
6.3	Ausführung der Usabilitytests . . . . .	12
<b>7</b>	<b>Präsentation der Ergebnisse</b>	<b>13</b>
7.1	Lessons Learned . . . . .	13
7.2	Abschlussbetrachtung . . . . .	13
<b>8</b>	<b>Literaturverzeichnis</b>	<b>14</b>

# 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[7], 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[6], welche 2014 entwickelt wurde und die iOS Design kopierende Design Sprache Cupertino[2].

#### 2.1.2 Entwicklung von Betriebssystem übergreifenden Applikationen

#### 2.1.3 Laufzeit-Performance

### 2.2 Dart



# **3 Anforderungsanalyse**

## **3.1 Identifikation des Ist-Zustands**

**3.1.1 Problemabgrenzung (Welches Problem soll gelöst werden?)**

**3.1.2 Kontext des Problems (Wo liegt das Problem?)**

**3.1.3 Ziele der Beteiligten (Warum muss es gelöst werden?)**

**3.1.4 Szenariensammlung (Wie könnte ein Softwaresystem helfen)**

## **3.2 Identifikation des Soll-Zustands**

## **3.3 Analyse der Schwachstellen**

## **3.4 Anforderungskatalog**

## **3.5 Lastenheft**

## **3.6 Pflichtenheft**

**3.6.1 Funktionale Anforderungen**

**3.6.2 Nicht funktionale Anforderungen**

## **4 Architektur der Software**

### **4.1 Architekturbeschreibung**

### **4.2 Entwurfsentscheidungen**

### **4.3 Erweiterbarkeit der Software für andere Studiengänge**

## **5 Implementierung**

### **5.1 Umsetzung der Anforderungen**

### **5.2 Entwurf der Benutzeroberfläche**

## **6 Validierung und Verifikation**

### **6.1 Softwaretests**

### **6.2 Abdeckung der Softwareanforderungen**

### **6.3 Ausführung der Usabilitytests**

# **7 Präsentation der Ergebnisse**

## **7.1 Lessons Learned**

## **7.2 Abschlussbetrachtung**

## **8 Literaturverzeichnis**

# Literatur

- [1] *Build beautiful native apps in record time.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/>.
- [2] *Cupertino (iOS-style) Widgets.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/widgets/cupertino/>.
- [3] *Foundation Libaryn.* (Zugriff Juni 3, 2018). URL: <https://docs.flutter.io/flutter/foundation/foundation-library.html>.
- [4] *Introduction.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/faq/#introduction>.
- [5] *Material Components Widgets.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/widgets/material/>.
- [6] *Material Design.* (Zugriff Juni 3, 2018). URL: <https://material.io/design/>.
- [7] *Skia Graphics Library.* (Zugriff Juni 3, 2018). URL: <https://skia.org/>.
- [8] *Technical Overview.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/technical-overview/>.
- [9] *Technology.* (Zugriff Juni 3, 2018). URL: <https://flutter.io/faq/#what-technology-is-flutter-built-with>.
- [10] *Welcome to the Engine Wiki.* (Zugriff Juni 3, 2018). URL: <https://github.com/flutter/engine/wiki>.