

Cloudbasiertes Praxisrufsystem

IP 5

28. Juli 2021



Abbildung 0.1: Titlebild

Studenten	Joshua Villing, Kevin Zellweger
Fachbetreuer	Daniel Jossen
Auftraggeberin	Daniel Jossen
Studiengang	Informatik
Hochschule	Hochschule für Technik

Zusammenfassung

Das Abstract ist eine Art Zusammenfassung des ganzen Dokuments. Es gibt einen Einblick in die Aufgabenstellung, wie diese umgesetzt wurde und welches Ergebnis erreicht wurde. Aus diesem Grund wird das Abstract immer ganz am Schluss der Arbeit verfasst. Es besteht aus einem zusammengehörenden Absatz und umfasst ungefähr 10 bis 20 Zeilen. Formeln, Referenzen oder andere Unterbrechungen haben im Text nichts zu suchen. Direkt unter dem Abstract folgt eine Liste von drei bis vier Stichworten/Keywords. Diese werden in alphabetischer Reihenfolge aufgelistet und beschreiben das Themengebiet der Arbeit.

Keywords: Anleitung, LaTeX, Thesis, Vorlage

Management Summary siehe PF-IK.

Vorwort

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Fakultativ, siehe PF-IK (URL) Balalbala some edits. [1]

Inhaltsverzeichnis

1	Einleitung	1
2	Vorgehensweise	2
2.1	Stakeholder	2
2.2	Projektplan	3
2.3	Organisation	4
3	Anforderungen	5
3.1	User Stories	5
3.2	Rahmenbedingungen	7
3.3	Features	8
4	Konzept	9
4.1	Systemarchitektur	9
4.2	Mobile Client	11
4.2.1	Architektur	11
4.2.2	User Interface	11
4.3	Cloud Service	13
4.3.1	Architektur	13
4.3.2	Domänenmodell	13
4.3.3	Laufzeitmodell	13
4.4	Admin UI	14
4.5	Proof Of Concept	15
5	Evaluation Technologien	16
5.1	Mobile Client	16
5.2	Cloud Service	16
5.3	Betrieb und Plattform	16
6	Umsetzung	17
7	Schluss	18
8	Literaturverzeichnis	19
	Abbildungsverzeichnis	21

9	Anhang	21
9.1	Benutzerhandbuch	21
9.2	Betriebshandbuch	21
9.3	Entwicklerdokumentation	21
9.4	Ehrlichkeitserklärung	21

1 Einleitung

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Einleitungsbeispiele siehe PF-IK (URL)

2 Vorgehensweise

2.1 Stakeholder

Am Projekt IP5 Cloudbasiertes Praxisrufsystem sind folgende drei Stakeholder beteiligt.

Prof. Daniel Jossen

- Rolle: Auftraggeber und Betreuer
- Kontakt: daniel.jossen@fhnw.ch

Joshua Villing

- Rolle: Student
- Kontakt: joshua.villing@students.fhnw.ch

Kevin Zellweger

- Rolle: Student
- Kontakt: kevin.zellweger@students.fhnw.ch

2.2 Projektplan

Übersicht



Abbildung 2.1: Projektplan

Milestones

Milestones

POC: Mobile Client ->Cloud Nachricht schicken und etwas persistieren

Wahrscheinlich über HTTP / Rest

POC: Cloud ->Mobile Client Nachricht schicken und etwas anzeigen

Wahrscheinlich über Message Broke

Versenden mit hinterlegter Konfiguration

Konfigurierte Notification Types

1:N Versenden, empfangen und konfigurieren

Setup Wizard (Neu oder z.B. wie Praxiszimmer XY)

Voice to Speech

Voice Chat 1:n (Out Of Scope?)

2.3 Organisation

Kommunikation

Das Projekt IP5 Cloudbasiertes Praxisrufsystem wurde im FS21 gestartet. Die Organisation und Kommunikation des Projektes mussten dementsprechend für die Einschränkungen wegen Corona angepasst werden. Um sicherzustellen, dass die Kommunikation über die gesamte Projektdauer funktionieren kann, haben wir uns deshalb von Anfang an entschieden die Kommunikation über Remote- und Online Tools zu organisieren. Für Besprechungen und Planungen wurde Microsoft Teams gewählt. Die entsprechende Infrastruktur wurde von der FHNW zur Verfügung gestellt.

Dokumentation

Der Bericht wurde mit LaTeX und zusammen mit dem Quellcode verwaltet. Kurze Besprechungen, Notizen und interne Dokumentation erfolgten über ein geteiltes One Note Notizbuch.

Sämtliche Diagramme, Mockups und Skizzen wurden direkt in den Tools verwaltet, die zur Erstellung gebraucht wurden. Zum Schluss wurden alle für den Bericht relevanten Darstellungen exportiert und in den Bericht integriert.

Quellcodeverwaltung

Sämtlicher Quellcode der im Rahmen des Projektes entsteht, wurde mit Git verwaltet. Der Quellcode ist für Berechtigte unter dem Projekt IP5-Cloudbasiertes-Praxisrufsystem auf github.com einsehbar. (Referenz <https://github.com/IP5-Cloudbasiertes-Praxisrufsystem>). Berechtigungen können bei Joshua Villing oder Kevin Zellweger angefordert werden.

- IP5-praxis-mobile-client
- IP5-praxis-cloud-service
- IP5-praxis-admin-ui
- IP5-praxis-documentation

Tools und Werkzeuge

- draw.io
- moqus.com
- Visual Studio Code
- IntelliJ
- Git
- github.com

3 Anforderungen

Die im Rahmen des Projektes umzusetzenden Anforderungen wurden während des Projektes iterativ zusammen mit dem Kunden erarbeitet. Alle Anforderungen werden zuerst aus Fachlicher Sicht mit User Stories festgehalten, die ein konkretes Bedürfnis der Benutzer beschreiben. Weiter werden User Stories aus Sicht des Kunden festgehalten, welche Rahmenbedingungen und Bedürfnisse des Auftraggebers festhalten. Aufgrund der User Stories werden anschliessend Features definiert, welche umsetzbare Anforderungen an die Systemkomponenten definieren.

3.1 User Stories

Praxismitarbeiter

Id	Anforderung	Features
U01	Als Praxismitarbeiter möchte ich Benachrichtigungen versenden können, damit ich andere Mitarbeiter über Probleme und Anfragen informieren kann.	F0x
U02	Als Praxismitarbeiter möchte ich Benachrichtigungen empfangen können, damit ich auf Probleme und Anfragen anderer Mitarbeiter reagieren kann.	F0x
U03	Als Praxismitarbeiter möchte ich nur Benachrichtigungen sehen, die für mich relevant sind, damit ich meine Arbeit effizient gestalten kann.	F0x
U04	Als Praxismitarbeiter möchte ich über empfangene Benachrichtigungen aufmerksam gemacht werden, damit ich keine Benachrichtigungen verpasse.	F0x
U05	Als Praxismitarbeiter möchte ich sehen welche Benachrichtigungen ich verpasst habe, damit ich auf verpasste Benachrichtigungen reagieren kann.	F0x
U06	Als Praxismitarbeiter möchte ich eine Rückmeldung erhalten, wenn eine Benachrichtigung nicht versendet werden kann, damit Benachrichtigungen nicht verloren gehen.	F0x
U7	Als Praxismitarbeiter möchte ich einen physischen Knopf am Behandlungsstuhl haben damit ich notifikationen darüber versenden kann.	F0x
U8	Als Praxismitarbeiter möchte ich, dass mir Benachrichtigungen vorgelesen werden, damit ich informiert werde, ohne meine Arbeit unterbrechen zu müssen.	F0x
U9	Als Praxismitarbeiter möchte ich einen anderen Client anrufen können damit Fragen direkt geklärt werden können.	F0x
U10	Als Praxismitarbeiter möchte ich Unterhaltungen mit mehreren anderen Clients gleichzeitig führen können damit komplexe Fragen direkt geklärt werden können.	F0x

Praxisverantwortlicher

”

”””

3.2 Rahmenbedingungen

Id	Anforderung	Features
T01	Als Auftraggeber möchte ich, dass das Praxisrufsystem über iPads bedient werden kann, damit ich von bestehender Infrastruktur profitieren kann.	F0x
T02	Als Auftraggeber möchte ich, dass das Praxisrufsystem über Android Tablets bedient werden kann, damit es in Zukunft für eine weitere Zielgruppe verwendet werden kann.	F0x
T03	Als Auftraggeber möchte ich, dass die Codebasis für das Praxisrufsystem für Android und IOS verwendet werden kann, damit ich die Weiterentwicklung optimieren kann.	F0x
T04	Als Auftraggeber möchte ich, dass wo möglich der Betrieb von Serverseitigen Dienstleistungen über AWS betrieben wird, damit ich von bestehender Infrastruktur und Erfahrung profitieren kann.	F0x

3.3 Features

Nummeriert mit F01-Fxx - Lower Level, näher am technischen - Spezifiziert die Anforderungen die Umgesetzt werden sollen

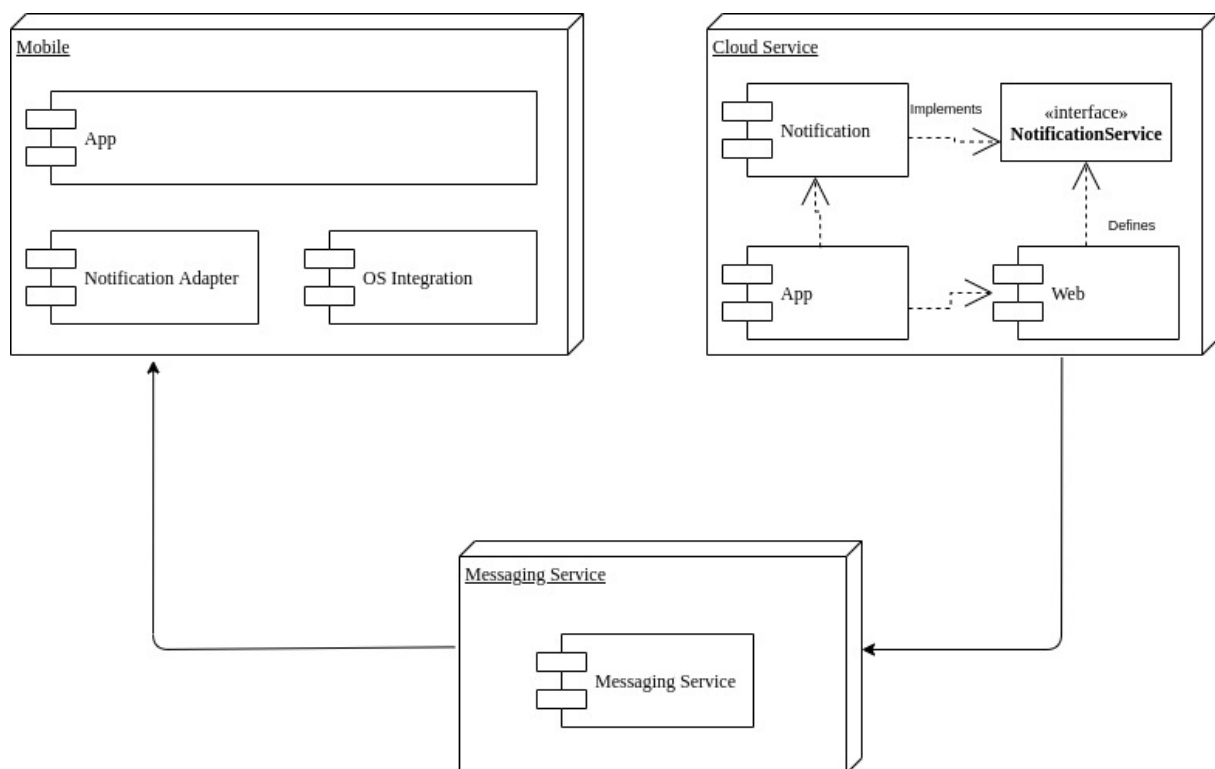
Projekt Scope

Id	Feature	User Stories
F01	Der Betrieb aller Cloud Services und Web UIs muss über AWS erfolgen.	Uxx
F02	Das Praxisrufsystem muss über einen Mobile Client auf iPads und Android Tables verwendbar sein.	Uxx
F03	Benachrichtigungen müssen über einen zentralen Cloud Service versendet werden.	Uxx
F04	Das Praxisrufsystem muss über einen zentralen Cloud Service verwaltet werden können.	Uxx
F05	Der Mobile Client muss Ton abspielen, wenn Notification eingeht	Uxx
F06	Mobile Client muss Ton nach X Sekunden abspielen wenn Notifikation nicht quittiert wurde	Uxx
F07	Mobile Client muss Möglichkeit bieten Notifikationen zu quittieren	Uxx
F08	Praxismitarbeiter muss sich anmelden können	Uxx
F09	Praxismitarbeiter muss relevanten client auswählen können	Uxx
F10	Mobile Client muss Buttons anzeigen, die vorkonfigurierte Notifikationen versenden	Uxx
F11	Verfügbare Buttons und Notifications müssen gemäss gewähltem client angezeigt werden	Uxx
F12	Der Mobile Client muss den Praxismitarbeiter darauf hinweisen, wenn der Versand einer Notifikation fehlgeschlagen ist	Uxx
F13	Der Mobile Client muss es ermöglichen fehlgeschlagene sends direkt zu wiederholen	Uxx
F14	CRUD für alle Konfigurationen müssen über ein Web UI möglich sein	Uxx
F15	CRUD müssen syntaktisch korrekt sein.	Uxx
F15	CRUD sollen semantisch korrekt sein.	Uxx
F16	Notification Types müssen Konfigurierbar sein	Uxx
F17	Clients müssen Konfigurierbar sein	Uxx
F18	Clients müssen Konfigurationen haben können	Uxx
F19	Pro Client muss definiert werden können, welche Notifikationen ihn interessieren	Uxx
F20	Pro Client muss definiert werden können, welche Notifikationen versendet werden können	Uxx
F21	Pro Client muss definiert werden können, welche Notifikationen versendet werden können	Uxx
F22	Administrator muss sich anmelden können	Uxx
F22	Mobile Client muss Background Push Benachrichtigungen unterstützen	Uxx

Out Of Scope

Weitere Features wurden noch nicht konkreter definiert. Da jeweils je SSprint"die als nächstes umzusetzenden Features definiert wurden.

- Soll Client für Raspberry Pi geben
- Client Configuration für einzelnen Button soll möglich sein
- Mobile Client soll 1:1 Unterhaltungen unterstützen
- Mobile Client soll 1:n Unterhaltungen unterstützen.



4 Konzept

4.1 Systemarchitektur

Überblick

Für das Cloudbasierte Praxisruf System sehen wir fünf Komponenten vor:

- Messaging Service
- Cloud Service
- Mobile Client
- Admin UI
- VOIP Mediator

Mobile Client

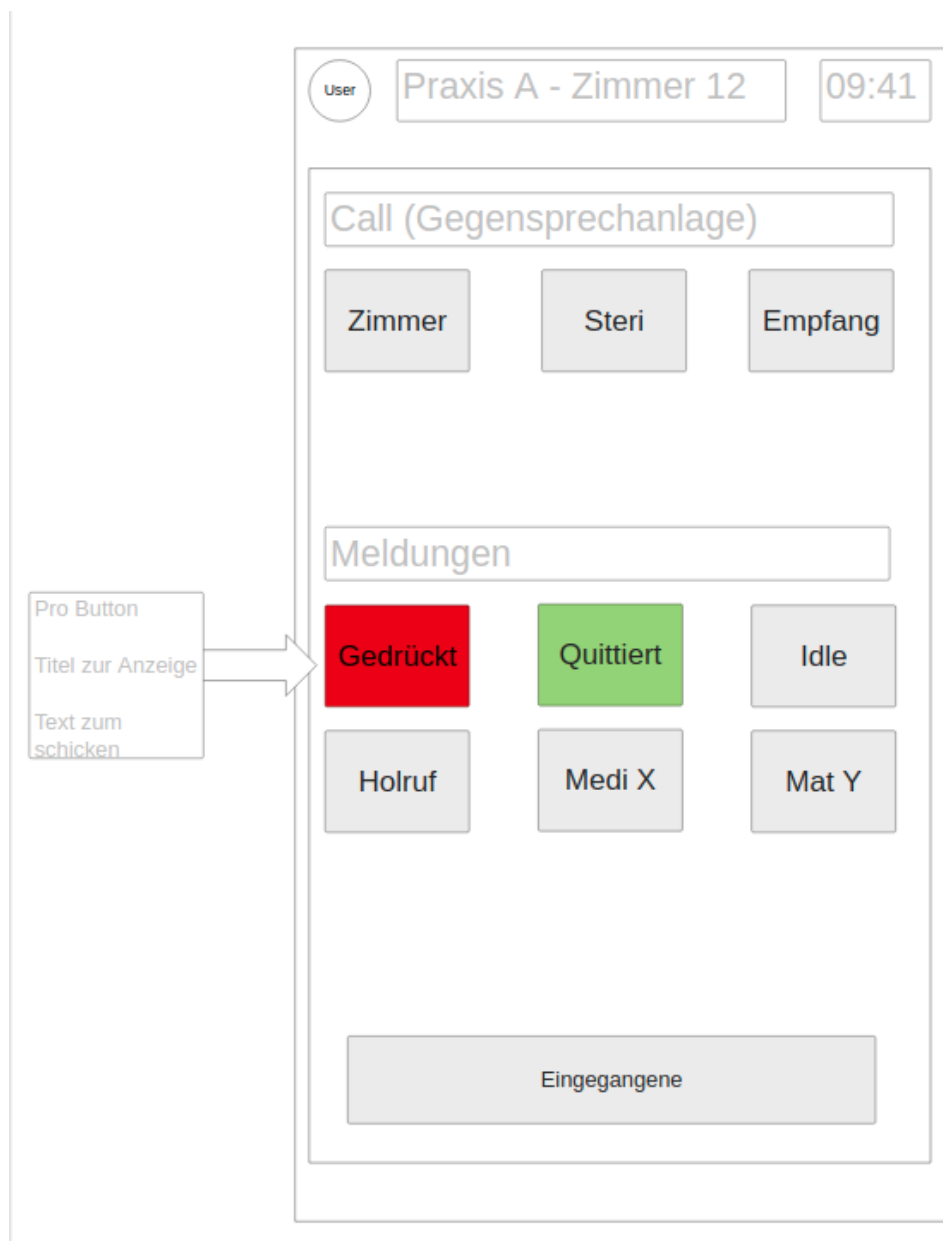
- Der Mobile Client implementiert die Anbindung an den Messaging Service.
- Als Reaktion auf eine Notification wird eine Rückmeldung im UI angezeigt.
- Als Reaktion auf eine Notification wird eine OS Push Notifikation gesendet. Das UI bietet einen Button der eine Anfrage an die REST Schnittstelle im Cloud Service sendet.

Cloud Service

- Responsibilities (Notification and Configuration)
- Microservice Granularity

Messaging Service

- Dies wird ein externer Service den wir in die Applikationen einbinden. Standard hierfür ist Firebase Notifications.
- Der Messaging Service nimmt Notifikationen vom Cloud Service entgegen und gibt diese an den Mobile Client wieder.
- Dafür müssen auf beiden Seiten Komponenten eingebaut werden, die mit dem Messaging Service kommunizieren.




4.2 Mobile Client

4.2.1 Architektur

4.2.2 User Interface

Home Screen

Empfangene Meldungen



Praxis A - Zimmer 12

09:41

Pending

Titel Zimmer Nachricht lorem ipsum bla	X
Titel Zimmer Nachricht lorem ipsum bla	X
Titel Zimmer Nachricht lorem ipsum bla	X
Titel Zimmer Nachricht lorem ipsum bla	X

4.3 Cloud Service

4.3.1 Architektur

4.3.2 Domänenmodell

4.3.3 Laufzeitmodell

4.4 Admin UI

4.5 Proof Of Concept

Anforderungen

- Als <Sender Rolle>möchte ich Notifikationen versenden können.
- Als <Empfänger Rolle>möchte ich Notifikationen in der Applikation sehen, wenn die Applikation geöffnet ist.
- Als <Empfänger Rolle>möchte ich Notifikationen über das OS erhalten, wenn die Applikation minimiert ist.

Restriktionen

- Nur 1 Client.
- Nur 1 fixe Notifikation. Keine Types.
- Notifikation wird vom Client gesendet und vom selben Client empfangen.
- Keine Authentication oder Authorization.

5 Evaluation Technologien

5.1 Mobile Client

<https://kotlinlang.org/lp/mobile/>

+Jet Brains Infrastructure +We like Kotlin

-iOS Env. Needed to develop for Apple -Still has to develop separate API und UI Modules for Platforms

<https://web.dev/progressive-web-apps/>

+No need of Native Codebase +Perfect for Android -Eventually drawbacks because no entire API Access

-PWAs on IOS suck

<https://cordova.apache.org/>

+ Popular Framework + Tons of plugins to access apis

-Still need to have a Mac for iOS development -Not a truly native app -i API Issues

<https://nativescript.org/>

+Provides a Workaround for nasty X-tools +Claims to be truly Native -Do we really trust it? (sorta new and passion project of a few people)

<https://flutter.dev>

-Why do you hate me?

SSimply Write Everything twice”

+Would definitely work

-Do most things twice -We don't have time for that -Kunde wünscht ausdrücklich nur eine Codebasis für beide Clients.

<https://stackshare.io/stackups/apache-cordova-vs-nativescript>

<https://nativescript.org/blog/build-nativescript-apps-remotely-from-windows-or-l>

5.2 Cloud Service

<https://aws.amazon.com/>

<https://spring.io/projects/spring-boot>

Konfig der Clients könnte sich als No-SQL anbieten.

Config muss nur gelesen und an den Client geschickt oder abgespeichert werden

<https://www.mongodb.com/>

5.3 Betrieb und Platform

AWS ist MUSS

6 Umsetzung

7 Schluss

8 Literaturverzeichnis

[1] D. E. Knuth, *The T_EXbook*. Addison – Wesley, 1990.

Abbildungsverzeichnis

0.1	Titlebild	1
2.1	Projektplan	3

9 Anhang

9.1 Benutzerhandbuch

9.2 Betriebshandbuch

9.3 Entwicklerdokumentation

9.4 Ehrlichkeitserklärung