

## Diplomarbeit

# Der Titel der Arbeit

**Untertitel der Arbeit**

Imst, 1. Juni 2017

Eingereicht von

Max Mustermann

Verantwortlich für IT: HTML, CSS, BWL: Kaufvertrag

Susanne Sorglos

Verantwortlich für IT: HTML, CSS, BWL: Kaufvertrag

Otto Normalverbraucher

Verantwortlich für IT: HTML, CSS, BWL: Kaufvertrag

Eingereicht bei

Alexander Scharmer, Claudio Landerer und Stefan Stolz

# **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbst verfasst und keine anderen als die angeführten Behelfe verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht. Ich bin damit einverstanden, dass meine Arbeit öffentlich zugänglich gemacht wird.

---

Ort, Datum

---

Jakob Tomasi

---

Peter Pollheimer

---

Elias Gabl

# **Abnahmeerklärung**

Hiermit bestätigt der Auftraggeber, dass das übergebene Produkt dieser Diplomarbeit den dokumentierten Vorgaben entspricht. Des Weiteren verzichtet der Auftraggeber auf unentgeltliche Wartung und Weiterentwicklung des Produktes durch die Projektmitglieder bzw. die Schule.

---

Ort, Datum

---

Auftraggeber

# **Vorwort**

Beauftragt wurde das Projektteam von Herr Hammerl im Namen des Landesschulrates Tirol. Der Kontakt mit Herr Hammerl wurde von Alexander Scharmer hergestellt. In einer ersten Besprechung erläuterte Herr Hammerl die Problemstellung, ein Ticketsystem verwenden zu müssen, welches auf Mobilgeräten kaum effektiv eingesetzt werden kann. Das Projektteam und Herr Hammerl einigten sich darauf, im Rahmen des Diplom- und Abschlussprojektes des Teams dessen Dienste in Anspruch zu nehmen und im Gegenzug die Projektbetreuung zu übernehmen.

# **Kurzfassung**

Ein bereits bestehendes Ticketsystem des Landesschulrats Tirol soll durch Änderungen funktional vereinfacht und optisch aufbereitet werden. Mit diesem System können IT-ManagerInnen Probleme melden, die die IT Infrastruktur einer Schule betreffen. Nach den ersten abgeschlossenen Phasen des Projektes, wurde jedoch vom Projektteam festgestellt, dass der Umfang des vorgegebenen Ticketsystems jedoch zu umfangreich ist.

Um die Projektziele zu erreichen und Änderungen zu bewerkstelligen, liegt der Fokus auf einer vom Projektteam verfassten Dokumentation um mit dem vorgegebenen Quellcode bestmöglichst zu arbeiten.

# **Abstract**

As of today, the education authority of Tyrol uses the open source ticketing system OSTicket. It enables IT-Managers to report problems with school's IT infrastructure. The scope of this project is to simplify the usage of OSTicket by creating a new web-based interface and removing unused functions. However, while getting used to the internal works of OSTicket, the team realized it was way too bulky.

In order to meet the project's requirements, it became necessary to find an alternative system, which enables the team to deliver a ticketing system capable of adaption in an efficient manner.

# Zusammenfassung

Communicational erweitert ein bestehendes System des Landesschulrats Tirol, basierend auf OSTicket. OSTicket ist ein Webbasiertes Open-Source Ticketsystem. Es wird verwendet, um IT-ManagerInnen an Tirols Schulen Probleme ihrer IT-Infrastruktur an die Zuständigen SystemadministratorInnen bekannt zu geben. Dies erfolgt unter der Domain itsys-tirol.at. Dieses Portal unterstützt nur Geräte mit großen Bildschirmen (ab 1024 Pixel Bildschirmbreite). Dies erschwert die Verwendung mit mobilen Geräten erheblich. Durch die im Rahmen dieses Projektes vorgenommenen Anpassungen an der Benutzeroberfläche wird die Benutzung mit verschiedensten Endgeräten wie z.B. Smartphones und Tablets ermöglicht.

Des Weiteren befindet der Projektpartner - OStR. Prof. Mag. Hammerl Helmut - die Grundkonfiguration von OSTicket für zu voluminös. Deshalb sollen möglichst viele nicht verwendete Funktionen im angepassten User-Interface weggelassen werden, um die Bedienbarkeit und Userfreundlichkeit zu erhöhen. Nach vielen Stunden der Einarbeitung in OSTicket wurde dem Projektteam klar, dass dieses durch seine allgemeine Beschaffenheit wie die fehlende Dokumentation des Codes und die Vermischung von Programmlogik und HTML-Elementen schlecht für die vom Projektteam geplante Anpassungen geeignet ist. Aufgrund dessen wurde eine Alternative gesucht, mit der die im Rahmen des Projektes durchzuführenden Änderungen effizienter zu bewerkstelligen sind. Gefunden wurde ein Fork von OSTicket mit dem Namen Katak (eine Abspaltung, die von einem anderen Entwicklerteam durchgeführt und gewartet wird), welcher für die Projektbedürfnisse eine hervorragende Alternative darstellt.

# Inhaltsverzeichnis

<b>1. Projektmanagement</b>	<b>13</b>
1.1. Metainformationen . . . . .	13
1.1.1. Team . . . . .	13
1.1.2. Betreuer . . . . .	13
1.1.3. Partner . . . . .	13
1.1.4. Ansprechpartner . . . . .	13
1.2. Vorerhebungen . . . . .	13
1.2.1. Projektzieleplan . . . . .	13
1.2.2. Projektumfeld . . . . .	14
1.2.3. Risikoanalyse . . . . .	14
1.3. Pflichtenheft . . . . .	16
1.3.1. Zielbestimmung . . . . .	16
1.4. IST Zustand . . . . .	16
1.5. SOLL Zustand . . . . .	17
1.5.1. SMART . . . . .	17
1.5.2. Produkteinsatz und Umgebung . . . . .	18
1.5.3. Projektumfeldanalyse . . . . .	18
1.5.4. Stakeholder . . . . .	19
1.5.5. Funktionalitäten . . . . .	19
1.5.6. Muss Anforderungen . . . . .	19
1.5.7. Soll Anforderungen . . . . .	20
1.5.8. Testszenarien und Testfälle . . . . .	20
1.6. Testfälle . . . . .	20
1.6.1. Testfall A . . . . .	20

1.6.2. Testfall B . . . . .	21
1.6.3. Testfall C . . . . .	22
1.6.4. Liefervereinbarung . . . . .	22
1.7. Planung . . . . .	23
1.7.1. Projektstrukturplan . . . . .	23
1.7.2. Meilensteine . . . . .	23
1.7.3. Gant-Chart . . . . .	23
1.7.4. Abnahmekriterien . . . . .	23
1.7.5. Pläne zur Evaluierung . . . . .	23
1.7.6. Ergänzungen und zu klärende Punkte . . . . .	23
<b>2. Vorstellung des Produktes</b>	<b>24</b>
2.1. Realisierbarkeit OSTicket . . . . .	24
2.1.1. Codebeispiele . . . . .	25
2.2. Systemdokumentation . . . . .	27
<b>3. Problemanalyse</b>	<b>28</b>
3.1. USE-Case-Analyse . . . . .	28
3.1.1. Ablaufbeschreibung . . . . .	31
3.2. Wireframes . . . . .	35
3.3. Prototyp . . . . .	36
3.4. Domain-Class-Modelling . . . . .	37
3.5. User-Interface-Design . . . . .	38
<b>4. Systementwurf</b>	<b>41</b>
4.1. Architektur . . . . .	41
4.2. Benutzerschnittstellen . . . . .	41
4.3. Datenbankentwurf . . . . .	42
4.4. Klassenentwurf . . . . .	42
4.5. Sicherheit des Systems . . . . .	43
<b>5. Implementierung</b>	<b>44</b>
<b>6. Deployment</b>	<b>45</b>

<b>7. Tests</b>	<b>46</b>
7.1. Systemtests . . . . .	46
7.2. Akzeptanztests . . . . .	46
<b>8. Projektevaluation</b>	<b>47</b>
<b>9. Benutzerhandbuch</b>	<b>48</b>
<b>10. Zusammenfassung</b>	<b>49</b>
<b>11. Beispielkapitel</b>	<b>50</b>
11.1. Beispiele zitieren . . . . .	50
11.1.1. Beispiele Abbildungen . . . . .	51
11.2. Beispiele Listen . . . . .	52
11.3. Beispiel Codesequenz . . . . .	54
11.3.1. Quicksort in JAVA . . . . .	54
11.4. Beispieltext . . . . .	56
<b>Abbildungsverzeichnis</b>	<b>59</b>
<b>Tabellenverzeichnis</b>	<b>60</b>
<b>Quelltexte</b>	<b>61</b>
<b>Literaturverzeichnis</b>	<b>62</b>
<b>A. Anhang-Kapitel</b>	<b>64</b>
A.1. Anhang-Section . . . . .	64

# **Einleitende Bemerkungen**

# **Notationen**

Beschreibung wie Code, Hinweise, Zitate etc. formatiert werden

# **1. Projektmanagement**

## **1.1. Metainformationen**

### **1.1.1. Team**

### **1.1.2. Betreuer**

### **1.1.3. Partner**

### **1.1.4. Ansprechpartner**

## **1.2. Vorerhebungen**

### **1.2.1. Projektzieleplan**

### **1.2.2. Projektumfeld**

- Identifikation der Stakeholder
- Charakterisierung der Stakeholder
- Maßnahmen
- Grafische Darstellung des Umfeldes

### **1.2.3. Risikoanalyse**

#### **Risikoidentifikation**

Folgende Risiken können während der Projektdurchführung erwartet werden:

- R1** Projektmitglied steigt aus dem Projekt aus.
- R2** Projektpartner stellt seine Kooperation ein.
- R3** Projektpartner ändert seine Anforderungen.
- R4** Termine können nicht eingehalten werden.
- R5** Anforderungen werden nicht erreicht.

#### **Bewertung und Behandlung**

Die aufgelisteten Risiken werden nach Auswirkung und Eintrittswahrscheinlichkeit bewertet. Je höher die Zahl in der Tabelle, desto höher ist die Auswirkung bzw. Einwirkung.

Risiko	Wahrscheinlichkeit	Auswirkung
R1 Ausstieg Projektmitglied	3	10
R2 Projektpartner stellt Kooperation ein	2	10
R3 Projektpartner ändert seine Anforderungen	4	7
R4 Termine können nicht eingehalten werden	3	6
R5 Anforderungen werden nicht erreicht	2	8

Tabelle 1.1.: Analyse Einwirkung & Auswirkung

### Risikomatrix

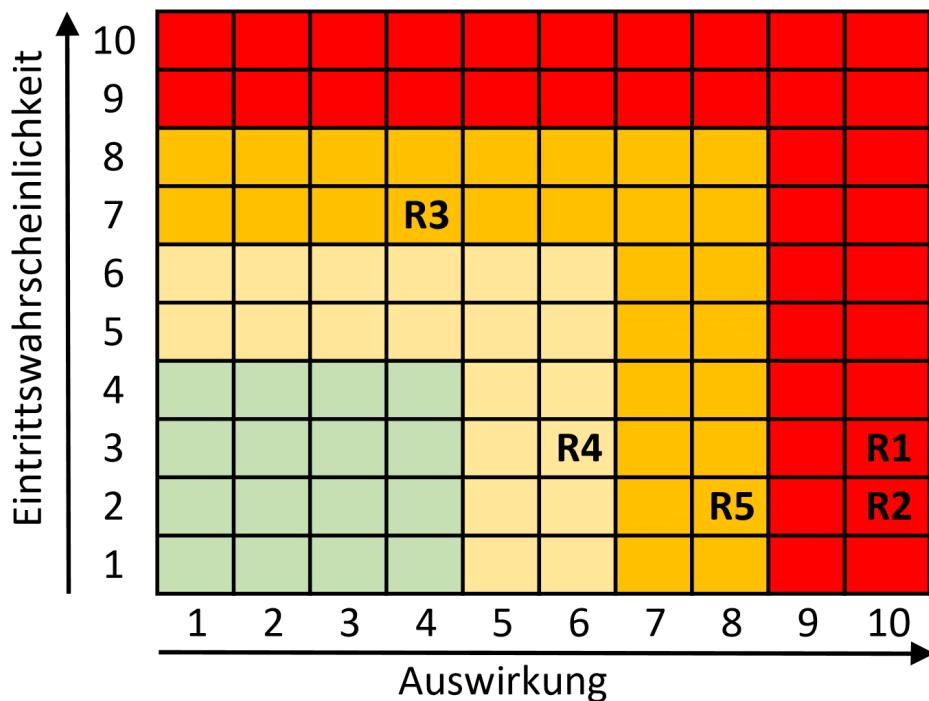


Abbildung 1.1.: Risikomatrix

## 1.3. Pflichtenheft

### 1.3.1. Zielbestimmung

## 1.4. IST Zustand

IT-ManagerInnen an Tirols Schulen können Probleme mit der Infrastruktur melden und Anfragen zur Beschaffung von Ressourcen/Komponenten stellen. Sie können den Bearbeitungsverlauf ihrer Tickets beobachten. SystembetreuerInnen empfangen die Tickets der IT-ManagerInnen, welche sich in ihrem Cluster befinden. SystembetreuerInnen bearbeiten die Tickets und antworten auf die Anfragen.

The screenshot shows the homepage of the 'IT Betreuer Tirol Ticketsystem'. At the top, there's a banner with the logo of the Landes Schulrat für Tirol (LSR) and a building illustration. Below the banner, there are three navigation links: 'Support Center Home' (with a house icon), 'Open a New Ticket' (with a document icon), and 'Check Ticket Status' (with a ticket icon). On the right side, there are language selection buttons for German and English, and a 'Sign In' link. The main content area has a heading 'Willkommen im Supportbereich' and a paragraph explaining the ticket system's purpose. Below this, there are two service cards: one for 'Open a New Ticket' (with a white shirt and green plus sign icon) and one for 'Check Ticket Status' (with a white ticket and blue information icon).

Abbildung 1.2.: IST-Zustand OS-Ticket

Die Benutzerschnittstelle ist derzeit nur auf die Benutzung mit großen Bild-

schirmen (ab 1024 Pixel Bildschirmbreite) ausgelegt. Sie kann sich nicht an kleinere Formate (Smartphone, etc.) anpassen.

## **1.5. SOLL Zustand**

Bessere Usability soll mit Hilfe von Mobile-First Orientierung auf Basis von Bootstrap erreicht und wenn möglich die Ticketerstellung vereinfacht werden. Das Backend soll die Aufteilung in mehrere hierarchische Organisationseinheiten ermöglichen und eine Erweiterung von Landesebene auf Bundesebene zulassen. Des Weiteren gilt es, den AnwenderInnen den Ticketingprozess intuitiver zu gestalten.

### **1.5.1. SMART**

#### **S Spezifisch**

Systembetreuer und IT-Manager können Support- und Beschaffungsanfragen mit Hilfe des Ticketsystems abwickeln.

#### **M Messbar**

Systembetreuer empfangen die Tickets und kümmern sich um die Probleme. Die Schulen werden in Cluster eingeteilt und von Systembetreuern verwaltet.

#### **A Attraktiv**

Die Plattform muss auf jedem Endgerät verfügbar sein (Responsive Design). Das Absetzen und Ansehen von Tickets soll vereinfacht werden, die Plattform bietet einige Funktionen die für das System relevant sind.

#### **R Realisierbar**

Zum Realisieren wird eine Testumgebung von Seiten des Betreuers zur

Verfügung gestellt. Das Responsive Design wird mithilfe eines Framework (Bootstrap) realisiert.

**T Terminisierbar**

Im Juni 2017 wird das Projekt abgeschlossen und eine technische Dokumentation des Projekts liegt vor.

### **1.5.2. Produkteinsatz und Umgebung**

### **1.5.3. Projektumfeldanalyse**

#### **Einflussfaktoren**

Das Projekt wurde durch den Landesschulrat Tirol in Auftrag gegeben. Die Ansprechperson, Herr Helmut Hammerl, informiert uns über den IST und SOLL-Zustand der Plattform und unterstützt das Projekt mit Ideen und Hilfestellungen bei Problemstellungen.

Des Weiteren beeinflussen die Anwender (IT-Manager) und die Systembetreuer der Plattform das Projektresultat. Da auf die Anwenderfreundlichkeit viel Wert gelegt wird, spielen diese Faktoren eine wirkliche Rolle.

Die Projektbetreuer Stefan Stolz und Alexander Scharmer sind für auftretende Fragen, bezüglich Problemstellungen die während des Projekts auftreten können, enorm einflussreich.

### 1.5.4. Stakeholder

#### Stakeholder Identifizieren

Stakeholder	Einfluss	Konfliktpotential
Mag. Helmut Hammerle	3	0
Dr. Stefan Walch	2	0
Stefan Stolz	1	+
Alexander Scharmer	1	+
Michael Gamper	0	0
LSI DI Anton Lendl	3	0
Team Mitglieder	2	0

Tabelle 1.2.: Stakeholder Identifikation

#### Stakeholder Klassifizieren

Stakeholder	Risiken durch Stakeholder	Strategien
Mag. Helmut Hammerle	Ändern der Ansprüche	Unterschriebenes Pflichtenheft
Dr. Stefan Walch	Nicht bestätigen des Projektantrages	Durchdachter Projektantrag
Stefan Stolz	falsche Informationen, kein Interesse	regelmäßiges Treffen
Alexander Scharmer	falsche Informationen, kein Interesse	regelmäßiges Treffen
Michael Gamper	Nicht bestätigen des Projektantrages	Durchdachter Projektantrag
LSI DI Anton Lendl	Nicht bestätigen des Projektantrages	Durchdachter Projektantrag
Team Mitglieder	Mangelnde Motivation	Faire Arbeitsverteilung

Tabelle 1.3.: Stakeholder Klassifikation

### 1.5.5. Funktionalitäten

### 1.5.6. Muss Anforderungen

IT-ManagerInnen und Systembetreuer müssen sich unter itsys-tirol.at, einem Portal des Landesschulrates, basierend auf OSticket anmelden können. Des

Weiteren sollen die Schulen selbst bestimmen, wer einen Zugang zum Portal erhalten soll, um Tickets erstellen zu können.

Die angemeldeten IT-ManagerInnen müssen Probleme mit der Infrastruktur melden können und Anfragen zur Beschaffung von Ressourcen bzw. Komponenten einreichen können.

SystembetreuerInnen müssen die Tickets der IT-ManagerInnen empfangen, welche sich in ihrem Cluster befinden. Die eingereichten Tickets sollen von den Systembetreuern bearbeitet werden können.

### **1.5.7. Soll Anforderungen**

Es soll eine neue Weboberfläche entwickelt werden, die auf dem HTML & CSS Framework Bootstrap basiert. Dieses soll sich auf Einfachheit in der Anwendung und Benutzerfreundlichkeit fokussieren. Die Latenzzeit sollte so niedrig wie möglich gehalten werden um ein Reibungsloses Arbeiten zu ermöglichen.

### **1.5.8. Testszenarien und Testfälle**

Die Testfälle in unserem Projekt beziehen sich auf das Ticketingsystem OSTicket.

## **1.6. Testfälle**

### **1.6.1. Testfall A**

- **Beschreibung:** Ein IT-Manager möchte ein Ticket erstellen.

- **Vorbedingung:** Der User benötigt ein internetfähiges Gerät und muss im Portal eingeloggt sein.
- **Aktion:** Der Benutzer wählt den Tab "Neues Ticket" und füllt die notwendigen Felder aus.
- **Soll-Reaktion:** Das System setzt das Ticket für den zuständigen Systembetreuer sichtbar.

Tester:

M Datum:

### **1.6.2. Testfall B**

- **Beschreibung:** Ein Systembetreuer möchte ein Ticket bearbeiten.
- **Vorbedingung:** Der Systembetreuer benötigt ein internetfähiges Gerät und muss im Portal eingeloggt sein.
- **Aktion:** Der Systembetreuer wählt den Tab "Meine Tickets" und wählt ein Ticket aus das Bearbeitet werden muss. Durch die Beschreibung des Tickets, weiß der Systembetreuer über die Problemstellung Bescheid und kann dementsprechend handeln.
- **Soll-Reaktion:** Der Systembetreuer kann sich um die Problemstellung kümmern und das Ticket nach erfolgreicher Bearbeitung wieder schließen.

Tester:

Datum:

### **1.6.3. Testfall C**

- **Beschreibung:** Ein Anwender möchte ein bestimmtes Ticket suchen und dieses begutachten.
- **Vorbedingung:** Der Anwender benötigt ein internetfähiges Gerät und muss im Portal eingeloggt sein.
- **Aktion:** Der Anwender gibt im Suchfeld ein Stichwort ein nach dem er suchen möchte.
- **Soll-Reaktion:** Das gesuchte Ticket soll angezeigt werden.

Tester:

Datum:

### **1.6.4. Liefervereinbarung**

- Lieferumfang
- Modus
- Verteilung(Deployment)

## **1.7. Planung**

### **1.7.1. Projektstrukturplan**

### **1.7.2. Meilensteine**

### **1.7.3. Gant-Chart**

### **1.7.4. Abnahmekriterien**

### **1.7.5. Pläne zur Evaluierung**

### **1.7.6. Ergänzungen und zu klärende Punkte**

## 2. Vorstellung des Produktes

In diesem Kapitel werden die einzelnen Ansätze aufgelistet und evaluiert, die während des Projekts durchprobiert wurden.

### 2.1. Realisierbarkeit OSTicket

OSTicket hat sich als umfangreicher herausgestellt wie zu Beginn des Projekts angenommen wurde. Nach Ausarbeitung des konzeptuellen Ziels des Projektes erfolgte eine lange Phase der Evaluation, Einarbeitung und Dokumentation von OSTicket. Nach ca. 30 Stunden dieser Phase, in der die Komplexität und Schwerfälligkeit des Systems OSTicket langsam zu Tage gefördert wurde. Die Anzahl der Dateien setzt sich folgendermaßen zusammen:

- 414 .php Dateien
- 15 .css Dateien
- 9 .less Dateien
- 61 .sql Dateien
- 1 .html Datei
- **Ges. 500 Dateien**

Ein weiteres Hindernis ergibt sich durch die Absenz einer auch nur annähernd aktuellen Dokumentation des laufend erweiterten und angepassten OSTicket.

Um die Realisierbarkeit zu veranschaulichen nun einige Codeauszüge aus oSTicket.

### 2.1.1. Codebeispiele

Quelltext 2.1: Auszug aus oSTicket

```
1 <?php
2 ****
3 main.inc.php
4
5 Master include file which must be included at the
6 start of every file.
7 The brain of the whole system. Don't monkey with it.
8
9 Peter Rotich <peter@osticket.com>
10 Copyright (c) 2006-2013 oSTicket
11 http://www.osticket.com
12
13 Released under the GNU General Public License WITHOUT
14 ANY WARRANTY.
15 See LICENSE.TXT for details.
16
17 vim: expandtab sw=4 ts=4 sts=4:
18 ****
19
20 #Disable direct access.
21 if(isset($_SERVER['SCRIPT_NAME']))
22 && !strcasecmp(basename($_SERVER['SCRIPT_NAME'])
23 , basename(__FILE__));
24 die('kwaheri rafiki!');
25
26 require('bootstrap.php');
27 Bootstrap::loadConfig();
```



```
58 if ($_SESSION['::sysmsgs'])
59 {
60     extract($_SESSION['::sysmsgs']);
61     unset($_SESSION['::sysmsgs']);
62 }
63 ?>
```

In diesem Codeauszug wird die Komplexität von osTicket sehr gut veranschaulicht, der Aufruf von elf statischen Funktionen und der dürftigen Dokumentation der Programmierer "Don't monkey around with it", machen den Code unlesbar.

## 2.2. Systemdokumentation

# 3. Problemanalyse

## 3.1. USE-Case-Analyse

### Akteure:

- Systembetreuer
- Anwender (IT-Manager und eventuell Lehrer, in Folgendem Anwender genannt)

<b>Name</b>	Anmelden am Portal
<b>ID</b>	C00
<b>Beschreibung</b>	Ein Anwender/Systembetreuer nutzt seine Login-Daten um sich in das System einzuwählen.
<b>Akteure</b>	Anwender, Systembetreuer
<b>Häufigkeit</b>	5/5
<b>Auslöser</b>	Der Anwender möchte das Portal benutzen.
<b>Bedingungen</b>	Anmelde Daten vorhanden und Rechte vergeben.
<b>Endzustand</b>	Eine Session wurde geöffnet und der Anwender kann das System nutzen.
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Benutzerdaten eingeben</li><li>2. Login Vorgang initiieren</li></ol>
<b>Ausnahmen</b>	Error → Anmelde Daten prüfen. Error → Anmelde Daten neu beziehen.

Tabelle 3.1.: Use-Case C00

# Der Titel der Arbeit

<b>Name</b>	Erstellung eines Standardtickets
<b>ID</b>	C01
<b>Beschreibung</b>	Ein Anwender meldet sich am Portal an und möchte ein neues Netzwerkgerät bestellen. Dieser eröffnet ein neues Standardticket und beschreibt den Grund für die Anschaffung. Der Anwender reicht das Ticket ein.
<b>Akteure</b>	Anwender
<b>Häufigkeit</b>	2/5
<b>Auslöser</b>	Es wird eine Komponente/Ressource für die Schule benötigt.
<b>Bedingungen</b>	C00
<b>Endzustand</b>	Ticket wurde in der Datenbank gespeichert.
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Formular öffnen</li><li>2. Formular ausfüllen</li><li>3. Ticket überprüfen</li><li>4. Ticket einreichen</li></ol>
<b>Ausnahmen</b>	Error - Ticket kann nicht abgegeben werden → C01 erneut ausführen.

Tabelle 3.2.: Use-Case C01

## Der Titel der Arbeit

<b>Name</b>	Erstellung eines Incident
<b>ID</b>	C02
<b>Beschreibung</b>	Ein Anwender meldet sich am Portal an und öffnet ein Ticket, er meldet ein Problem, das er selbst nicht lösen kann. Der Anwender markiert das Ticket als Incident und speichert es.
<b>Akteure</b>	Anwender
<b>Häufigkeit</b>	4/5
<b>Auslöser</b>	Ein für den Anwender nicht lösbares technisches Problem.
<b>Bedingungen</b>	C00, Auslöser
<b>Endzustand</b>	Incident wurde in der Datenbank gespeichert und der Systembetreuer hat eine Benachrichtigung erhalten.
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Formular öffnen</li><li>2. Formular ausfüllen</li><li>3. Incident überprüfen</li><li>4. Incident einreichen</li></ol>
<b>Ausnahmen</b>	Error - Ticket kann nicht abgegeben werden → C02 erneut ausführen.

Tabelle 3.3.: Use-Case C02

<b>Name</b>	Ticketstatus prüfen
<b>ID</b>	C03
<b>Beschreibung</b>	Ein Anwender meldet sich am Portal an und sieht sich seine eigenen Tickets an. Er kann einsehen ob der Systembetreuer das Ticket erhalten hat und wie weit seine Bestellung vorgeschritten ist.
<b>Akteure</b>	Anwender
<b>Häufigkeit</b>	3/5
<b>Auslöser</b>	Anwender hat Anfrage erhalten
<b>Bedingungen</b>	C00, zu überprüfendes Ticket muss vorhanden sein
<b>Endzustand</b>	Keine Veränderungen am System
<b>Hauptablauf</b>	<ol style="list-style-type: none"> <li>1. Ticket öffnen</li> <li>2. Status einsehen</li> </ol>
<b>Ausnahmen</b>	Error – Ticket kann nicht überprüft werden → C03 erneut ausführen.

Tabelle 3.4.: Use-Case C03

### 3.1.1. Ablaufbeschreibung

#### C00:

Der Benutzer wird aufgefordert seine Benutzerdaten einzugeben. Hat er die Daten richtig eingegeben wird er angemeldet. Sind die Daten falsch kommt er zurück zur Dateneingabe.

#### C01:

Der Benutzer muss zuerst ein Formular öffnen, um ein Ticket erstellen zu können. Anschließend muss das Formular ausgefüllt werden.

Nach dem Überprüfen des Tickets kann der Benutzer sich entscheiden ob er das Ticket abschickt oder ob er Änderungen vornehmen möchte.

Wenn das Abschicken fehlgeschlagen ist kommt er zum Anfang zurück. Wenn das abschicken erfolgreich war, wird das Ticket in der Datenbank gespeichert.

#### C02:

Der Benutzer muss zuerst ein Formular öffnen um einen Incident erstellen zu können. Anschließend muss das Formular ausgefüllt werden. Nach dem Überprüfen des Incident kann der Benutzer sich entscheiden, ob er den Incident abschickt oder ob er Änderungen vornehmen möchte.

<b>Name</b>	Ticketlöschung beantragen
<b>ID</b>	C04
<b>Beschreibung</b>	Ein Anwender meldet sich am Portal an und beantragt die Löschung seines Tickets.
<b>Akteure</b>	Anwender
<b>Häufigkeit</b>	1/5
<b>Auslöser</b>	Problem hat sich erübrig; Ticket ist falsch
<b>Bedingungen</b>	C00, zu lösches Ticket muss vorhanden sein
<b>Endzustand</b>	Systembetreuer hat die Anfrage auf Löschung erhalten
<b>Hauptablauf</b>	1. Ticket öffnen 2. Löschung beantragen
<b>Ausnahmen</b>	Error – Löschvorgang nicht erfolgreich → C04 erneut ausführen.

Tabelle 3.5.: Use-Case C04

Wenn das Abschicken fehlgeschlagen ist kommt er zum Anfang zurück. Wenn das abschicken erfolgreich war wird der Incident in der Datenbank gespeichert und der Systembetreuer erhält eine Benachrichtigung.

**C03:**

Der Benutzer muss das Formular öffnen um den Status zu sehen. Ist das Öffnen fehlgeschlagen kommt er wieder zum Ausgangspunkt und kann es nochmal versuchen.

**C04:**

Um die Löschung beantragen zu können muss der Benutzer zuerst das Ticket öffnen. Danach kann er die Löschung beantragen. Schlägt dies fehl kommt er wieder zurück an den Anfang und kann es nochmal versuchen. War der Antrag auf Löschung erfolgreich erhält der Systembetreuer eine Anfrage zur Löschung.

**C05:**

Um ein Ticket zu löschen muss der Systembetreuer das Ticket öffnen und löschen. Schlägt dies fehl kommt er wieder zurück an den Anfang und kann es nochmal versuchen. War das Löschen erfolgreich wurde das Ticket aus der Datenbank entfernt.

**C06:**

Um den Ticketstatus ändern zu können muss der Systembetreuer das Ticket

# Der Titel der Arbeit

<b>Name</b>	Ticket löschen
<b>ID</b>	C05
<b>Beschreibung</b>	Der Systembetreuer meldet sich am Portal an und kümmert sich um Löschanfragen
<b>Akteure</b>	Systembetreuer
<b>Häufigkeit</b>	1/5
<b>Auslöser</b>	C04
<b>Bedingungen</b>	C00, zu lösches Ticket muss vorhanden sein
<b>Endzustand</b>	Das Ticket wurde entfernt
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Ticket öffnen</li><li>2. Ticket löschen</li></ol>
<b>Ausnahmen</b>	Error – Löschtorgang nicht erfolgreich → C05 erneut ausführen.

Tabelle 3.6.: Use-Case C05

öffnen und ändern. Schlug dies fehl kommt er wieder zurück an den Anfang und kann es nochmal versuchen. War das ändern erfolgreich wurde der Ticketstatus geändert.

<b>Name</b>	Ticketstatus ändern
<b>ID</b>	C06
<b>Beschreibung</b>	Der Systembetreuer meldet sich am Portal an und sieht die Tickets der Anwender in seinem Cluster. Er ändert den Status eines Tickets in Bearbeitung und kümmert sich um die Anfrage.
<b>Akteure</b>	Systembetreuer
<b>Häufigkeit</b>	4/5
<b>Auslöser</b>	C01
<b>Bedingungen</b>	C00, Ticket vorhanden
<b>Endzustand</b>	Der Ticketstatus wurde geändert
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Ticket öffnen</li><li>2. Ticketstatus ändern</li></ol>
<b>Ausnahmen</b>	Error – Statusänderung nicht erfolgreich → C06 erneut ausführen.

Tabelle 3.7.: Use-Case C06

### C07:

Um ein Ticket schließen zu können muss der Systembetreuer das Ticket öffnen um es dann zu schließen. Schlug dies fehl kommt er wieder zurück an den Anfang und kann es nochmal versuchen. War das schließen erfolgreich ist das Ticket geschlossen.

<b>Name</b>	Incident bearbeiten
<b>ID</b>	C07
<b>Beschreibung</b>	Der Systembetreuer erhält eine E-Mail-Benachrichtigung über einen Incident. Er klickt auf den mitgelieferten Link und meldet sich am Portal an. Er bearbeitet den Incident mit erhöhter Priorität.
<b>Akteure</b>	Systembetreuer
<b>Häufigkeit</b>	3/5
<b>Auslöser</b>	C02
<b>Bedingungen</b>	C00, Incident vorhanden
<b>Endzustand</b>	Incident ist abgearbeitet
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Benachrichtigung erhalten</li><li>2. Incident öffnen</li><li>3. Incident bearbeiten</li><li>4. C08</li></ol>
<b>Ausnahmen</b>	Error – Incident öffnen nicht erfolgreich → C07 erneut ausführen.

Tabelle 3.8.: Use-Case C07

### 3.2. Wireframes

<b>Name</b>	Ticket schließen
<b>ID</b>	C08
<b>Beschreibung</b>	Der Systembetreuer meldet sich am Portal an und schließt ein Ticket bzw. einen Incident nach dessen Erledigung.
<b>Akteure</b>	Systembetreuer
<b>Häufigkeit</b>	5/5
<b>Auslöser</b>	C01, C02
<b>Bedingungen</b>	C00, Ticket vorhanden und abgearbeitet
<b>Endzustand</b>	Ticket ist geschlossen
<b>Hauptablauf</b>	<ol style="list-style-type: none"><li>1. Ticket öffnen</li><li>2. Ticket schließen</li></ol>
<b>Ausnahmen</b>	Error – Ticket konnte nicht geschlossen werden → C08 erneut ausführen.

Tabelle 3.9.: Use-Case C08

### 3.3. Prototyp

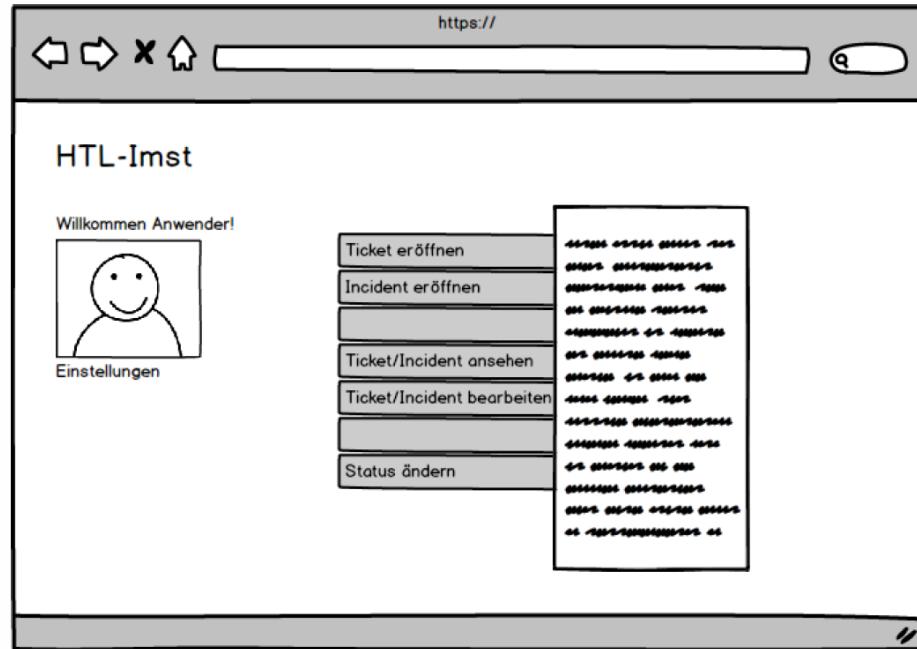


Abbildung 3.1.: Mockup Anwendersicht

### 3.4. Domain-Class-Modelling

- "Dinge" (Rollen, Einheiten, Geräte, Events etc.) identifizieren, um die es im Projekt geht
- ER-Modellierung oder Klassendiagramme
- Zustandsdiagramme (zur Darstellung des Lebenszyklus von Domain-Klassen darstellen)

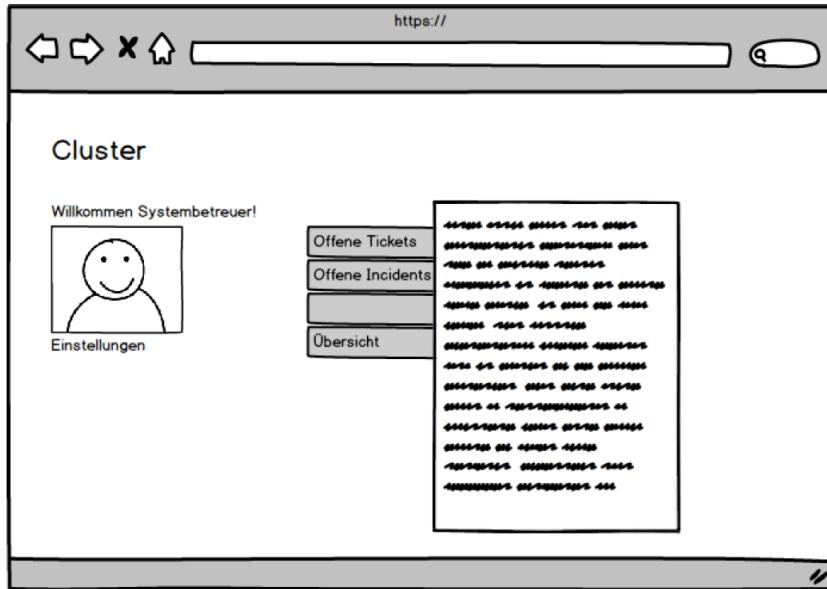


Abbildung 3.2.: Mockup Systembetreuer

### 3.5. User-Interface-Design

- Mockups
- Wireframes

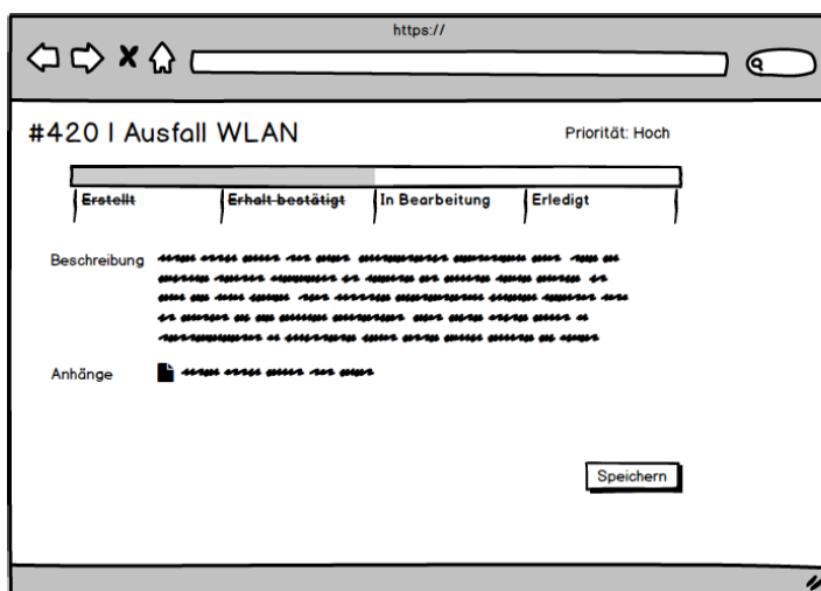


Abbildung 3.3.: Mockup Ticketstatus

Ausfall WLAN #420

Priorität: hoch

The screenshot shows a ticket status prototype. At the top, there is a dark blue horizontal bar. Below it, a grey header row contains four status boxes: 'Erstellt' (green), 'Erhalt-bestätigt' (green), 'In Bearbeitung' (light blue), and 'Erledigt' (light blue). The 'In Bearbeitung' box is currently highlighted. Below the header, there is a large light grey area containing the word 'Beschreibung' (Description) in large bold letters. Underneath 'Beschreibung', there is a block of placeholder text (Lorem ipsum).

Beschreibung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

The screenshot shows a ticket attachments prototype. It features a large light grey area with the word 'Anhänge' (Attachments) in large bold letters. Below it is a smaller block of placeholder text. At the bottom left of this area, there is a dark blue button labeled 'Speichern' (Save).

Anhänge

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore.

Speichern

Abbildung 3.4.: Prototyp Ticketstatus

# **4. Systementwurf**

## **4.1. Architektur**

Darstellung und Beschreibung der Systemarchitektur (z.B. Komponentendiagramme). Beispiele für Architekturen:

- MVC
- Schichten
- Pipes
- Request Broker
- Service-Oriented

## **4.2. Benutzerschnittstellen**

Kompletter Entwurf aller Benutzerschnittstellen

## **4.3. Datenbankentwurf**

Komplettes ER-Diagramm incl. Beschreibungen zu jeder einzelnen Tabelle und jeder Beziehung.

## **4.4. Klassenentwurf**

Design jedes einzelnen USE-Cases

- Design-Klassendiagramme vom Domain-Klassendiagramm ableiten (incl. detaillierter Darstellung und Verwendung von Vererbungshierarchien, abstrakten Klassen, Interfaces)
- Sequenzdiagramme vom System-Sequenz-Diagramm ableiten
- Detaillierte Zustandsdiagramme für wichtige Klassen

Verwendung von CRC-Cards (Class, Responsibilities, Collaboration) für die Klassen

- um Verantwortlichkeiten und Zusammenarbeit zwischen Klassen zu definieren und
- um auf den Entwurf der Geschäftslogik zu fokussieren

Design-Klassen für jeden einzelnen USE-Case können sein:

- UI-Klassen
- Data-Access-Klassen
- Entity-Klassen (Domain-Klassen)
- Controller-Klassen
- Business-Logik-Klassen
- View-Klassen

Optimierung des Entwurfs (Modularisierung, Erweiterbarkeit, Lesbarkeit):

- Kopplung optimieren
- Kohäsion optimieren
- SOLID
- Entwurfsmuster einsetzen

## **4.5. Sicherheit des Systems**

Beschreibung aller sicherheitsrelevanten Designentscheidungen;

# **5. Implementierung**

Detaillierte Beschreibung der Implementierung aller Teilkomponenten der Software entlang der zentralsten Use-Cases:

- GUI-Implementierung
- Controllerlogik
- Geschäftslogik
- Datenbankzugriffe

Detaillierte Beschreibung der Teststrategie (Testdriven Development):

- UNIT-Tests (Funktional)
- Integrationstests

Zu Codesequenzen:

- kurze Codesequenzen direkt im Text (mit Zeilnummern auf die man in der Beschreibung verweisen kann)
- lange Codesequenzen in den Anhang (mit Zeilennummer) und darauf verweisen (wie z.B. hier ??)

# **6. Deployment**

- Design der Ausführungsumgebung (Produktivenvironment)
- Umsetzung der Ausführungsumgebung
- Deployment
- DevOps-Thema

# **7. Tests**

## **7.1. Systemtests**

Systemtests aller implementierten Funktionalitäten lt. Pflichtenheft

- Beschreibung der Teststrategie
- Testfall 1
- Testfall 2
- Tesfall 3
- ...

## **7.2. Akzeptanztests**

## **8. Projektevaluation**

siehe Projektmanagement-Unterricht

## **9. Benutzerhandbuch**

falls im Projekt gefordert

# **10. Zusammenfassung**

- Etwas längere Form des Abstracts
- Detaillierte Beschreibung des Outputs der Arbeit

# **11. Beispielkapitel**

## **11.1. Beispiele zitieren**

Das ist ein Zitat mit Klammern, (Resnick, 1996), das ein Zitat ohne Klammern: Harel und Papert (1991). Hier das selbe Zitat mit einer Seitenangabe und Klammern (Resnick, 1996, S. 23).

Wird ein Absatz aus einer Quelle sinngemäß übernommen (nicht wörtlich), dann kann nach dem Absatz das entsprechende Zitat in Klammern angeführt werden. (Anastopoulou u. a., 2012, S. 33)

Wenn ein Zitat im Text angegeben wird, wie z.B. so Beer, Rudolf und Benischek, Isabella (2011), können die Klammern weggelassen werden.

Der folgende Absatz zeigt ein Blockzitat (wörtlich übernommene Textpassage aus einer Quelle):

Dr. Heinrich Faust ist ein angesehener Wissenschaftler und Akademiker, der trotz seiner wissenschaftlichen Studien und einer guten Bildung seinen Wissensdurst nicht stillen kann. Eines Nachts sitzt er in seinem Studierzimmer und grübelt über den Sinn des Lebens nach, findet jedoch keine Antworten. Daraufhin wendet er sich der Geisterwelt zu. Er beschwört einen Erdgeist, versucht sich den Geistern gleich zu stellen, was ihm jedoch nicht gelingt. Von Ohnmacht

getrieben will er sich das Leben nehmen. Sein Selbstmordversuch wird jedoch von Glockenläuten zum Ostertag und seinen Kindheitserinnerungen gestört. (Ackermann, 2001, S. 21)

Hier wird ein wörtliches Zitat inline angegeben: „Das ist ein kleines direktes Zitat.“ Göhlich und Zirfas (2007), und danach geht es gleich wieder direkt weiter. Ob ein wörtliches Zitat inline oder als eigener Block angezeigt wird, entscheidet Latex auf Basis der Länge.

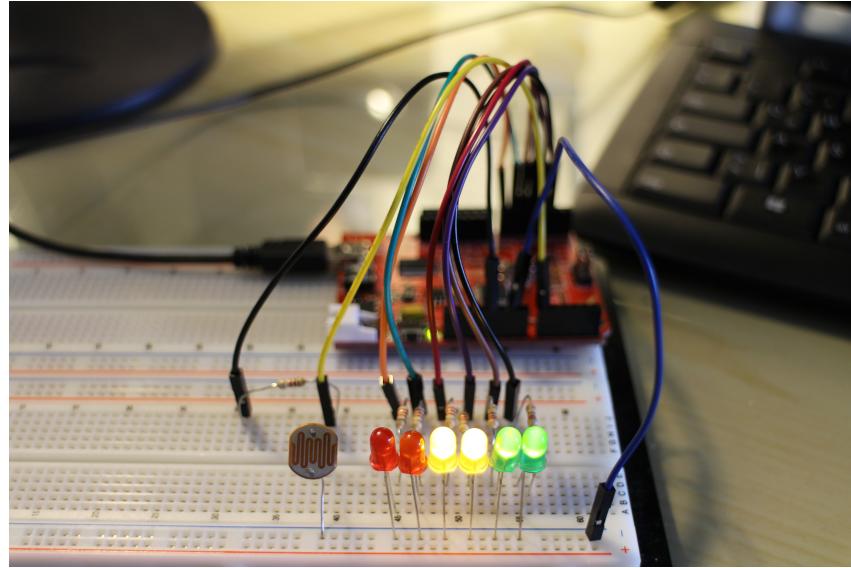
### **11.1.1. Beispiele Abbildungen**

Auf diese Weise kann man zum Beispiel in Latex auf die Abbildung 11.1 verweisen. Die Kennung für den Verweis vergibt man selbst mit dem „label“ Kommando bei der Abbildung.

Jede Abbildung muss nicht nur mindestens einen Verweis im Text haben. Es wird außerdem eine Bildunterschrift verlangt. Für diese ist festgesetzt, dass die Abbildungsunterschrift alleine ausreichend sein muss, um zu verstehen, was am Bild zu erkennen ist.

Der nächste wichtige Punkt sind die Quellenangaben bei Abbildungen. Der Author muss zu jeder Abbildung die notwendigen Rechte haben und idealer Weise gibt man diese bei der Abbildung mit an. In Abbildung 11.1 auf Seite 52 sieht man das.

Es ist wichtig zu verstehen, dass Latex die Positionierung von Abbildungen übernimmt. Man definiert die Abbildung über begin-figure dort, wo man die Abbildung in etwa haben möchte, den Rest übernimmt Latex



© Stefan Stolz (CC BY-SA 3.0)

Abbildung 11.1.: Hintergrund: Arduino Board; Vordergrund: eine Lichterreihe und ein Lichtsensor (Fotowiderstand); In diesem Beispiel wird die Lichterreihe je nach Helligkeit des Umgebungslichtes gesteuert. Durch leichte Modifikationen kann man damit eine Lichtschranke oder auch eine Helligkeitssteuerung für das Smartphone simulieren.

### **Beispiele Tabellen**

Tabelle 11.1 ist ein Beispiel für eine aufwändiger Tabelle mit einer Abbildung und Überschrift.

Tabellen sind in Latex sehr kompliziert zu erzeugen. Alternativ kann man die Tabellen auch in einem anderen Programm gestalten und als Bild wieder einfügen. Dieses Bild kann dann innerhalb von begin-Table verwendet werden.

## **11.2. Beispiele Listen**

Im Folgenden wird eine Liste gezeigt:

DW OR N PACKAGE (TOP VIEW)	
NC	20
V <sub>CC</sub>	19 GND
SER IN	18 SER OUT
DRAIN0	17 DRAIN7
DRAIN1	16 DRAIN6
DRAIN2	15 DRAIN5
DRAIN3	14 DRAIN4
SRCLR	13 SRCK
$\overline{G}$	12 RCK
GND	11 GND
NC – No internal connection	
$V_{cc}$	Positive supply voltage
GND	Ground
SER IN	Daten Pin
SRCK	Clock Pin
RCK	Latch Pin
$\overline{SRCLR}$	Wenn <b>shift-register clear</b> LOW ist, werden die input Register gelöscht Wenn <b>output enable</b> HIGH ist, werden die Daten im Output Buffer LOW gehalten
$\overline{G}$	

Tabelle 11.1.: Aufwändige Tabelle mit Abbildung und Caption

- Ich weiß, dass viele Geräte des täglichen Lebens durch Computer gesteuert werden und kann für mich relevante nennen und nutzen.
  1. Und jetzt eine Numerierung
  2. Und jetzt eine Numerierung
- Ich kann wichtige Bestandteile eines Computersystems (Eingabe-, Ausgabegeräte und Zentraleinheit) benennen, kann ihre Funktionen beschreiben und diese bedienen.

Und jetzt eine Numerierung:

1. Aufzählungspunkt
  - a) Unteraufzählung
  - b) Unteraufzählung
    - Und jetzt noch eine Ebene ohne Aufzählung
    - Und jetzt noch eine Ebene ohne Aufzählung
2. Aufzählungspunkt

3. Aufzählungspunkt
4. Aufzählungspunkt
5. Aufzählungspunkt

## 11.3. Beispiel Codesequenz

In Listing 11.1 sieht man ein Quick-Sort-Listing in der Programmiersprache JAVA. Das Listings-Paket übernimmt die Formatierung von Codebausteinen und kann in der Präambel nach Belieben auf eine andere Sprache konfiguriert werden.

### 11.3.1. Quicksort in JAVA

Quelltext 11.1: QuickSort in Java

```
1 public class QuickSort
2 {
3     public static void main(String[] args)
4     {
5         int [] x =
6         {
7             9, 2, 4, 7, 3, 7, 10
8         }
9         ;
10        System.out.println(Arrays.toString(x));
11
12        int low = 0;
13        int high = x.length - 1;
14
15        quickSort(x, low, high);
```

```
16     System.out.println(Arrays.toString(x));
17 }
18
19 public static void quickSort(int[] arr, int low,
20 int high)
21 {
22     if (arr == null || arr.length == 0)
23         return;
24
25     if (low >= high)
26         return;
27
28     // pick the pivot
29     int middle = low + (high - low) / 2;
30     int pivot = arr[middle];
31
32     // make left < pivot and right > pivot
33     int i = low, j = high;
34     while (i <= j)
35     {
36         while (arr[i] < pivot)
37         {
38             i++;
39         }
40
41         while (arr[j] > pivot)
42         {
43             j--;
44         }
45
46         if (i <= j)
47         {
48             int temp = arr[i];
49             arr[i] = arr[j];
50             arr[j] = temp;
51             i++;
52             j--;
53         }
54     }
55 }
```

```
48         arr[i] = arr[j];
49         arr[j] = temp;
50         i++;
51         j--;
52     }
53 }
54
55 // recursively sort two sub parts
56 if (low < j)
57     quickSort(arr, low, j);
58
59 if (high > i)
60     quickSort(arr, i, high);
61 }
62 }
```

## 11.4. Beispieltext

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Text-

ausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen

## *Der Titel der Arbeit*

Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

# **Abbildungsverzeichnis**

1.1.	Risikomatrix . . . . .	15
1.2.	IST-Zustand OS-Ticket . . . . .	16
3.1.	Mockup Anwendersicht . . . . .	37
3.2.	Mockup Systembetreuer . . . . .	38
3.3.	Mockup Ticketstatus . . . . .	39
3.4.	Prototyp Ticketstatus . . . . .	40
11.1.	Arduino mit Lichtsensor und Lichterkette . . . . .	52

# **Tabellenverzeichnis**

1.1.	Analyse Einwirkung & Auswirkung . . . . .	15
1.2.	Stakeholder Identifikation . . . . .	19
1.3.	Stakeholder Klassifikation . . . . .	19
3.1.	Use-Case C00 . . . . .	28
3.2.	Use-Case C01 . . . . .	29
3.3.	Use-Case C02 . . . . .	30
3.4.	Use-Case C03 . . . . .	31
3.5.	Use-Case C04 . . . . .	32
3.6.	Use-Case C05 . . . . .	33
3.7.	Use-Case C06 . . . . .	34
3.8.	Use-Case C07 . . . . .	35
3.9.	Use-Case C08 . . . . .	36
11.1.	Aufwändige Tabelle mit Abbildung und Caption . . . . .	53

# **Quelltexte**

2.1. Auszug aus osTicket . . . . .	25
11.1. QuickSort in Java . . . . .	54

# Literaturverzeichnis

[Ackermann 2001] ACKERMANN, Edith: Piaget's constructivism, Papert's constructionism: What's the difference. In: *Future of learning group publication* 5 (2001), Nr. 3, S. 438. – URL [http://lovettresourcenetwork.wiki.lovett.org/file/view/EA.Piaget+\\_+Papert.pdf](http://lovettresourcenetwork.wiki.lovett.org/file/view/EA.Piaget+_+Papert.pdf). – Zugriffsdatum: 2014-07-09

[Anastopoulou u. a. 2012] ANASTOPOULOU, Stamatina ; BERLAND, Matthew ; FRANT, Janete B. ; BOYTCHEV, Pavel ; BRENNAN, Karen ; CHRONAKI, Anna ; CLAYSON, James ; CORREIA, Secundino ; DAGIENE, Valentina ; DEKOLI, Margarita: Constructionism 2012 Theory Practice and Impact. (2012), August. – URL [http://users.uoa.gr/~zsmyrnaiou/conferences\\_after2008/constructionism%201\\_2012.pdf](http://users.uoa.gr/~zsmyrnaiou/conferences_after2008/constructionism%201_2012.pdf). – Zugriffsdatum: 2014-03-26

[Beer, Rudolf und Benischek, Isabella 2011] BEER, RUDOLF ; BENISCHEK, ISABELLA: Aspekte kompetenzorientierten Lernens und Lehrens. In: BIFIE (Hrsg.): *Kompetenzorientierter Unterricht in Theorie und Praxis*. Graz : Leykam, 2011

[Göhlich und Zirfas 2007] GÖHLICH, Michael ; ZIRFAS, Jörg: *Lernen: Ein pädagogischer Grundbegriff*. Stuttgart : Kohlhammer, April 2007. – ISBN 9783170188693

[Harel und Papert 1991] HAREL, Idit ; PAPERT, Seymour: *Situating Con-*

*Der Titel der Arbeit*

*structionism.* Norwood, N.J : Ablex Publishing Corporation, U.S., 1991. – ISBN 9780893917869

[Resnick 1996] RESNICK, Mitchel: Distributed constructionism. In: *Proceedings of the 1996 international conference on Learning sciences*, International Society of the Learning Sciences, 1996, S. 280–284. – URL <http://dl.acm.org/citation.cfm?id=1161173>. – Zugriffsdatum: 2015-04-20

# **A. Anhang-Kapitel**

## **A.1. Anhang-Section**

Testtext