

Diplomarbeit

Der Titel der Arbeit

Untertitel der Arbeit

Imst, 10. 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 Ticket-system. 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

1. Projektmanagement	13
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
2.	Vorstellung des Produktes	24
2.1.	Realisierbarkeit OSTicket	24
2.1.1.	Codeausschnitte osTicket	25
2.1.2.	Evaluation Katak	28
2.1.3.	Evaluation OSTicky	28
2.2.	Systemdokumentation	28
2.3.	Datenbankentwurf	28
2.3.1.	Beschreibung der Tabellen OSTicket	28
3.	Problemanalyse	41
3.1.	USE-Case-Analyse	41
3.1.1.	Ablaufbeschreibung	44
3.2.	Wireframes	48
3.3.	Prototyp	49
3.4.	Domain-Class-Modelling	50
3.5.	User-Interface-Design	51
4.	Systementwurf	54
4.1.	Technologie	55
4.2.	Architektur	55
4.3.	Benutzerschnittstellen	56
4.4.	Klassenentwurf	56
4.5.	Sicherheit des Systems	59

5. Implementierung	60
6. Deployment	61
7. Tests	62
7.1. Systemtests	62
7.2. Akzeptanztests	62
8. Projektevaluation	63
9. Benutzerhandbuch	64
10. Zusammenfassung	65
Abbildungsverzeichnis	66
Tabellenverzeichnis	67
Quelltexte	69
A. Anhang-Kapitel	70
A.1. Anhang-Section	70

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

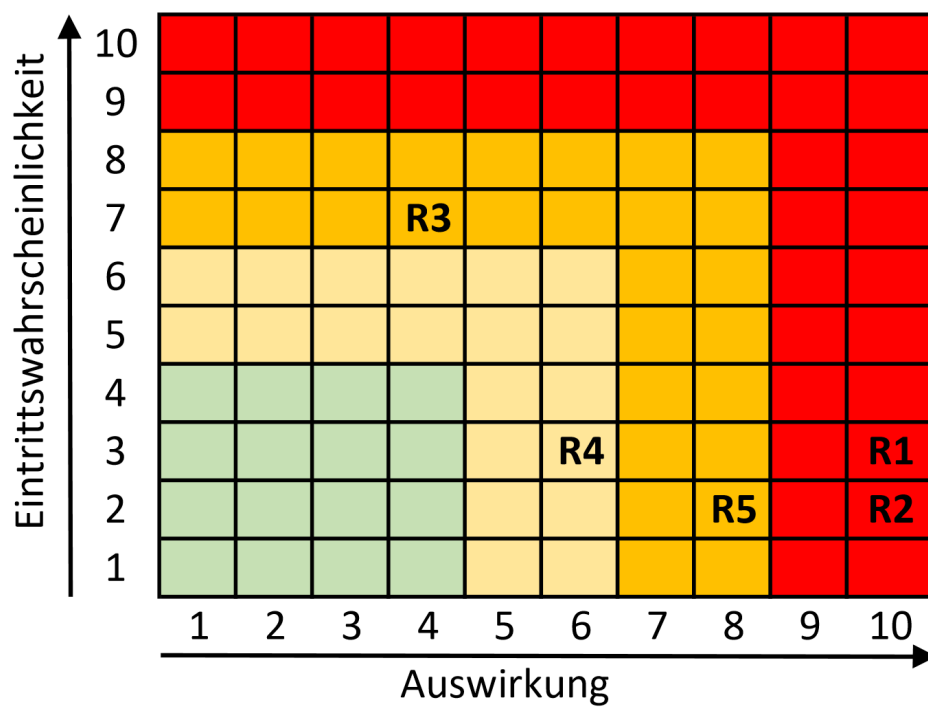


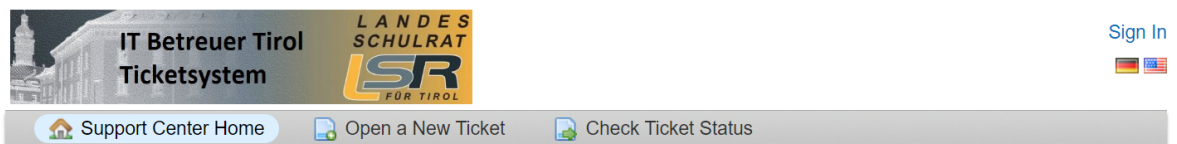
Abbildung 1.1.: Risikomatrix

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.



Willkommen im Supportbereich

Um unsere Supportanfragen besser organisieren zu können, verwenden wir dieses Support-Ticket-System. Jeder Anfrage wird eine eindeutige Ticket-Nummer zugewiesen, an Hand der Sie den Status Ihrer Anfrage jederzeit online nachverfolgen können. Zu Ihrem Überblick werden alle Anfragen mit den dazugehörigen Antworten in unserem Helpdesk gespeichert. So können Sie jederzeit die Antworten auf Ihre Anfragen online einsehen. Um ein Ticket zu eröffnen, ist eine gültige E-Mail-Adresse erforderlich.



Open a New Ticket

Please provide as much detail as possible so we can best assist you. To update a previously submitted ticket, please login.

Open a New Ticket



Check Ticket Status

We provide archives and history of all your current and past support requests complete with responses.

Check Ticket Status

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 Projektergebnis. 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	Unterschiedenes Pflichten
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

Gabi SorglosPeter Pollheimer

2. Vorstellung des Produktes

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. Codeausschnitte osTicket

Quelltext 2.1: main.inc.php

```
1 #Disable direct access.
2 if(isset($_SERVER['SCRIPT_NAME']))
3 && !strcasecmp(basename($_SERVER['SCRIPT_NAME']),
4 ,basename(__FILE__)))
5 die('kwaheri rafiki!');
6
7 require('bootstrap.php');
8 Bootstrap::loadConfig();
9 Bootstrap::defineTables(TABLE_PREFIX);
10 Bootstrap::i18n_prep();
11 Bootstrap::loadCode();
12 Bootstrap::connect();
13
14 #Global override
15 $_SERVER['REMOTE_ADDR'] = osTicket::get_client_ip();
```

In diesem Codeauszug wird die Lesbarkeit des Codes sehr gut veranschaulicht, der Aufruf von elf statischen Funktionen ohne Dokumentation erschwert die Erweiterbarkeit um ein vielfaches. Der einzige Kommentar des Programmierers *Don't Monkey around with it* ist weder aussagekräftig noch hilfreich.

Quelltext 2.2: pwreset.php

```
1 $inc = 'register.confirmed.inc.php';
2 $acct->confirm();
3 // FIXME:
4     The account has to be uncached in order for the lookup
5 // in the ::processSignOn to detect the confirmation
6 ModelInstanceManager::uncache($acct);
7 // Log the user in
8 if ($client =
9     UserAuthenticationBackend::processSignOn($errors))
10 {
11     if ($acct->hasPassword() &&
12         !$acct->get('backend'))
13     {
14         $acct->cancelResetTokens();
15     }
16     // No password setup yet -- force one to be created
17 else
18 {
19     $_SESSION['_client']['reset-token'] =
20     $_GET['token'];
21     $acct->forcePasswdReset();
22 }
```

Auch in der Datei *pwreset.php* ist nur eine sehr dürftige Dokumentation vorzufinden. Es sind lediglich vier Inhaltslose Kommentare wie *makes includes happy* und ein *FIXME* vorzufinden. Diese geben keinerlei Aufschluss über die Funktionalität und Zuständigkeit des Codes.

Quelltext 2.3: offline.php

```
1 require_once('client.inc.php');
2 if(is_object($ost) && $ost->isSystemOnline())
3 {
4     @header('Location: index.php');
5     //Redirect if the system is online.
6     include('index.php');
7     exit;
8 }
9 $nav=null;
10 require(CLIENTINC_DIR.'header.inc.php');
```

Der einzige Kommentar des Programmierers *modify to fit your needs* erschwert die Lesbarkeit dieser Datei enorm.

Quelltext 2.4: client.inc.php

```
1 /* include what is needed on client stuff */
2 require_once(INCLUDE_DIR.'class.client.php');
3 require_once(INCLUDE_DIR.'class.ticket.php');
4 require_once(INCLUDE_DIR.'class.dept.php');
5
6 //clear some vars
7 $errors=array();
8 $msg='';
9 $nav=null;
10 //Make sure the user is valid..before doing anything else.
11 $thisclient = UserAuthenticationBackend::getUser();
```

In der Datei client.inc.php wird lediglich dokumentiert, dass diese Datei in jeder client page inkludiert wird. Der Code in dieser Datei besteht hauptsächlich aus statischen Funktionsaufrufen.

2.1.2. Evaluation Katak

In progress

2.1.3. Evaluation OSTicky

in progress

2.2. Systemdokumentation

2.3. Datenbankentwurf

2.3.1. Beschreibung der Tabellen OSTicket

In dieser Dokumentation finden Sie eine grobe Beschreibung der Datenbank des Systems OS Ticket. OS Ticket ist ein Ticketsingsystem das für einfache Support Anwendung Entwickelt worden ist. Die Datenbank besteht aus 59 Tabellen von diesen sind manche nicht mehr aktuell bzw. werden nicht mehr gebraucht aber sind doch noch vorhanden. Daher werden in dieser Dokumentation nur die wichtigsten Tabellen der Datenbank beschrieben.

Bei gewissen Spalten ist es leider nicht möglich den Inhalt anzugeben da es keine gute Dokumentation der Datenbank gibt und man den Inhalt der Spalten aus dem Kontext schließen muss.

An dieser Stelle sollte eigentlich ein Datenmodell sein aber da dieses von OSTicket recht groß und komplex ist hat es hier nicht Platz aber Sie finden es auf dem mitgelieferten Datenträger.(Der Dateiname ist: OSTicketDB.mwb).

ost_attachment

Sämtliche Informationen über den Anhang eines Tickets finden sich in dieser Tabelle. Dies beinhaltet den Datentyp, Name, und id des Anhanges.

Column Name:	Datatype:	Content:
id	INT(10)	Eindeutige ID des Anhanges. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere
object_id	int(11)	Referenz auf ein Object das dem Anhang übergeben wird
type	CHAR(1)	Datentyp des Anhanges hat
file_id	int(11)	Referenz auf das File das sich im Anhang befindet
name	VARCHAR(255)	Name des Anhanges
inline	TINYINT(1)	Beschreibt den Zitierstil des Anhangs
lang	VARCHAR(16)	Die Sprache in der Abteilung

Tabelle 2.1.: tab:ost-attachment

ost_canned_response

In dieser Tabelle werden vorgefertigte Antworten für Tickets gespeichert. Über die Antwort wird der Titel, die Sprache, der Inhalt, Erstelldatum und das Änderungsdatum gespeichert.

Column Name:	Datatype:	Content:
canned_id	INT(10)	Eindeutige ID der Nachricht. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
dept_id	int(10)	Referenz auf die Abteilung die diese Nachricht verwenden können
isenabled	TINYINT(1)	Ob die Nachricht freigegeben ist oder nicht.
titel	VARCHAR(255)	Titel der Nachricht
response	TEXT	Text den die Nachricht beinhaltet
lang	VARCHAR(16)	Sprache der Nachricht
notes	TEXT	Anmerkung zur Nachricht
created	DATETIME	Erstelldatum der Nachricht
updated	DATETIME	Änderungsdatum der Nachricht

Tabelle 2.2.: tab:ost-canned-response

ost_config

In dieser Tabelle werden wichtige Informationen über das System OSTicket gespeichert. Die Werte werden mit einem Schlüssel in die Datenbank gespeichert (Key und Value). Diese Tabelle hat keine Referenz zu anderen Tabellen.

Column Name:	Datatype:	Content:
id	INT(11)	Eindeutige ID der Information. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
namespace	VARCHAR(64)	
key	VARCHAR(64)	Key der Information um die Suche zu erleichtern
value	TEXT	Informations Inhalt
updated	DATETIME	Erstelldatum der Information

Tabelle 2.3.: tab:ost-config

ost_content

In dieser Tabelle werden Inhalte der Seiten gehalten. Es wird angegeben ob der Inhalt aktiv ist oder nicht. Jeder Inhalt hat auch eine Titel und eine Body. Es werden auch noch der Typ und das Erstell-/Änderungsdatum angegeben.

Column Name:	Datatype:	Content:
id	INT(10)	Eindeutige ID des Contents. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
isactive	TINYINT(1)	Gib an ob der Inhalt aktiv ist oder nicht
type	VARCHAR(32)	Gib den Typ vom Inhalt an
name	VARCHAR(255)	Name vom Inhalt
body	TEXT	Der Bodyinhalt des Seiten Content
notes	TEXT	Anmerkungen zum Content
created	DATETIME	Erstelldatum des Content
updated	DATETIME	Änderungsdatum des Content

Tabelle 2.4.: tab:ost-content

ost_department

Diese Tabelle speichert Informationen über eine Abteilung. Jede Abteilung hat einen Namen und eine Signatur. Ein Feld das angibt ob die Abteilung

öffentlich ist oder nicht. Es wird auch festgehalten zu welcher Gruppe die Abteilung gehört und wann die Abteilung erstellt worden ist.

Column Name:	Datatype:	Content:
id	INT(10)	Eindeutige ID der Abteilung. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
pid	INT	Referenz zu der Tabelle ost_plugin
tpl_id	INT(10)	Referenz zum verwendeten Template für die Abteilung
sla_id	INT(10)	Referenz zur verwendeten Sla Vorlage
email_id	INT(10)	Referenz zu der Verwendeten E-Mail der Abteilung
autores_email_id	INT(10)	
manager_id	INT(10)	Referenz zum User der zum Manager der Abteilung ernannt wurde
flags	INT(10)	
name	VARCHAR(128)	Name der Abteilung
signature	TEXT	Signatur der Abteilung
ispublic	TINYINT(1)	Gib an ob die Abteilung öffentlich sichtbar ist
group_membership	TINYINT(1)	Gib an zu welcher Gruppe die Abteilung gehört
ticket_auto_response	TINYINT(1)	Das vordefinierte Standartticket der Abteilung
message_auto_response	TINYINT(1)	Die vordefinierte Standartantwort der Abteilung auf Tickets
path	VARCHAR	Pfad der Abteilung
updated	INT(10)	Änderungsdatum der Abteilung
created	INT(10)	Erstelldatum der Abteilung

Tabelle 2.5.: tab:ost-department

ost_faq

In dieser Tabelle werden die meist gestellten Fragen und die Antworten dazu gespeichert. Weiteres werden noch Schlüsselwörter und Anmerkungen zur

Frage gespeichert.

Column Name:	Datatype:	Content:
faq_id	INT(10)	Eindeutige ID der Frage. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
category_id	INT(10)	Referenz zu der Kategorie der die Frage zugeordnet ist
ispublished	TINYINT(1)	Gib an ob die Frage öffentlich abrufbar ist oder nicht
question	VARCHAR(255)	Die Frage
answer	TEXT	Die Antwort zur Frage
keywords	TINYTEXT	Die Schlüsselwörter der Frage
notes	TEXT	Anmerkung zur Frage
created	DATETIME	Erstelldatum der Frage
updated	DATETIME	Änderungsdatum der Frage

Tabelle 2.6.: tab:ost-faq

ost_staff

Diese Tabelle hält sämtliche Informationen über die Staff Mitglieder des Systems OSTicket. Es wird der Vorname, Nachname, Username, das Passwort, die Email-Adresse, die Telefonnummer, die Sprache, ob das Mitglied aktiv ist, ob das Mitglied ein Administrator ist usw. gespeichert. Die Tabelle enthält auch Informationen über die Abteilung und die Rolle des Mitglieds.

Column Name:	Datatype:	Content:
staff_id	INT(11)	Eindeutige ID des Mitgliedes. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
dept_id	INT(10)	Referenz zur Abteilung des Mitgliedes
role_id	INT(10)	Referenz zur Rolle des Mitgliedes
username	VARCHAR(32)	Der Username des Mitgliedes
firstname	VARCHAR(32)	Der Vorname des Mitgliedes
lastname	VARCHAR(32)	Der Nachname des Mitgliedes
passwd	VARCHAR(128)	Das Passwort des Mitgliedes in Hash Form
backend	VARCHAR(32)	
email	VARCHAR(128)	Die Email-Adresse des Mitgliedes
phone	VARCHAR(24)	Die Telefonnummer des Mitgliedes
phone_ext	VARCHAR(6)	Referenz zur Email an die, die Info zum Ticketeingang geschickt wird
mobile	VARCHAR(24)	Die Handynummer des Mitgliedes
signature	TEXT	Die Signatur des Mitgliedes
lang	VARCHAR(16)	Die Sprache des Mitgliedes
timezone	VARCHAR(64)	Die Zeitzone in der sich das Mitglied befindet
locale	VARCHAR(16)	Wo sich das Mitglied genau befindet(Ort, Stadt, Land)
notes	TEXT	Anmerkungen zum Mitglied
isactive	TINYINT(1)	Gib an ob das Mitglied aktiv ist
isadmin	TINYINT(1)	Gib an ob das Mitglied ein Administrator ist
isvisible	TINYINT(1)	Gib an ob das Mitglied für andere sichtbar ist
onvacation	TINYINT(1)	Gib an ob der Mitarbeiter im Urlaub ist
assigned_only	TINYINT(1)	
show_assigned_tickets	TINYINT(1)	Hat das Mitglied zugeteilte Tickets
changed_passwd	TINYINT(1)	Gib an ob das Mitglied das Passwort schon mal geändert hat
max_page_size	INT(11)	Gibt die maximale Anzahl an Tickets an die auf der Übersichtsseite des Mitgliedes angezeigt werde
auto_refresh_rate	INT(10)	Gib die Taktrate an wie oft die Übersichtsseite des Mitgliedes automatisch aktualisiert werden soll
default_signature_type	ENUM('none', 'mine', 'dept')	Gib die default Signatur bei der Beantwortung eines Tickets an
default_paper_size	ENUM('Letter', 'Legal', 'Ledger', 'A4', 'A3')	Gib das default Format der Antwort auf ein Ticket an
extra	TEXT	Hier können zusätzliche Informationen über das Mitglied gespeichert werden
permissions	TEXT	Die Zugriffsrechte des Mitgliedes
created	DATETIME	Erstelldatum des Mitgliedes
lastlogin	DATETIME	Datum an dem sich das Mitglied zuletzt angemeldet hat

ost_ticket

Sämtlicher Informationen über ein Ticket. Dazu gehören der User der das Ticket abgesetzt hat, die Erkennungsnummer, die Email-Adresse des Absenders, das Thema des Tickets und wer es bearbeiten muss. Es gibt auch ein Feld das speichert ob man auf das Ticket schon geantwortet hat, wann es erstellt und gegebenenfalls verändert worden ist und ob es schon geschlossen worden ist.

ost_user

Hier werden die User gespeichert die keine Mitglieder sind. Diese User können im OS Ticketsystem nur Tickets abschicken. Es wird der Name, der Status und wann er erstellt wurde gespeichert. Weiteres kann er auch einer Organisation zugeordnet werden.

ost_user_account

Hier wird auf Basis der ost_user Tabelle ein Account abgespeichert. Der Account enthält eine Referenz auf einen User. Der Account besteht aus folgenden Informationen: Status des Users, die Sprache, in welcher Zeitzone er sich aufhält, den Username, das Passwort und wann er registriert wurde.

ost_ticket_priority

Diese Tabelle enthält alle Informationen über die Priorität die ein Ticket haben kann. Im gesamten kann ein Ticket eine von vier Prioritäten haben.

ost_help_topic

Diese Tabelle speichert die Hilfethemen. Es kann jedem Thema eine bestimmte Priorität zugeordnet werden. Weiter könne Themen auch bestimmten Abteilungen, Teams, Administratoren oder Mitarbeiter zugeteilt werden.

Column Name:	Datatype:	Content:
ticket_id	INT(11)	Eindeutige ID des Tickets. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
number	VARCHAR(20)	Ticket Erkennungsnummer
user_id	INT(11)	Referenz zum User der das Ticket abgesetzt hat.
user_email_id	INT(11)	Referenz zur Email-Adresse des Users der das Ticket abgesetzt hat
status_id	INT(10)	Referenz zum Status des Tickets
dept_id	INT(10)	Referenz zur Abteilung der das Ticket zugewiesen wurde
sla_id	INT(10)	Referenz zum Sla des Tickets
topic_id	INT(10)	Referenz zum Thema dem das Ticket zugeordnet worden ist
staff_id	INT(10)	Referenz zur Tabelle ost_staff
team_id	INT(10)	Referenz zum Team dem das Ticket zugewiesen worden ist
email_id	INT(10)	Referenz zur Email an die, die Info zum Ticketeingang geschickt wird
lock_id	INT(10)	Referenz zur Tabelle ost_lock
flags	INT(10)	
ip_address	VARCHAR(64)	Die IP-Adresse von der das Ticket abgesetzt worden ist
source	ENUM(Web, Email, Phone, API, Other)	
source_extra	VARCHAR(40)	
isoverdue	TINYINT(1)	Gib an ob der Bearbeiter überfällig mit der Bearbeitung ist
isanswered	TINYINT(1)	Gib an ob dem Absender des Tickets schon geantwortet wurde
duedate	DATETIME	Gib an bis wann das Ticket bearbeitet sein sollte
est_duedate	DATETIME	Bis wann es geplant ist das Ticket bearbeitet zu haben
reopened	DATETIME	Gib an wann ein geschlossenes Ticket zuletzt geöffnet worden ist
closed	DATETIME	Speichert das Datum an dem das Ticket geschlossen worden ist
lastupdate	DATETIME	Gib das letzte Änderungsdatum an
created	DATETIME	Erstelldatum des Tickets
updated	DATETIME	Änderungsdatum des Tickets

Column Name:	Datatype:	Content:
id	INT(10)	Eindeutige ID des Users. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
org_id	INT(10)	Referenz auf die Organisation die der User zugeordnet ist
default_email_id	INT(10)	
status	INT(10)	Gibt den Status des Users an
name	TEXT	Name des Users
created	DATETIME	Erstelldatum des Users
updated	DATETIME	Änderungsdatum des Users

Tabelle 2.9.: tab:ost-user

Column Name:	Datatype:	Content:
id	INT(11)	Eindeutige ID des Accounts. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
user_id	INT(10)	Referenz auf den User dem der Account gehört
status	INT(11)	Den Status des Users
timezone	VARCHAR(64)	Gibt die Zeitzone an in dem sich der User befindet
lang	VARCHAR(16)	Die Sprache des Users
username	VARCHAR(64)	Username des Users
passwd	VARCHAR(128)	Das Passwort des Users in Hash form
backend	VARCHAR(32)	
extra	TEXT	Zusatz Informationen zum User
registered	TIMESTAMP	Hält fest wann sich der User registriert hat

Tabelle 2.10.: tab:ost-user-account

Column Name:	Datatype:	Content:
priority_id	TINYINT(4)	Eindeutige ID der Priorität. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
priority	VARCHAR(60)	Beschreibt die Priorität
priority_desc	VARCHAR(30)	Leg fest wie die Prioritäten geordnet werden
priority_color	VARCHAR(7)	Leg die Farbe der Prioritäten fest
priority_urgency	TINYINT(1)	Leg fest welche Dringlichkeit die Priorität hat
ispublic	TINYINT(1)	Gib an ob die Priorität öffentlich ist oder nicht

Tabelle 2.11.: tab:ost-ticket-priotity

Column Name:	Datatype:	Content:
topic_id	INT(11)	Eindeutige ID des Hilfsthemas. Dieser Wert ist rein technisch und hat neben der Eindeutigkeit keine weitere Aussagekraft.
topic_pid	INT(10)	Referenz zu einem Plugin für das Thema
isactive	TINYINT(1)	Leg fest ob das Thema aktiv ist oder nicht
ispublic	TINYINT(1)	leg fest ob das Thema für User zur Verfügung steht
noautoresp	TINYINT(3)	
flags	INT(10)	
status_id	INT(10)	Referenz zu dem Status
priotity_id	TINYINT(4)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema
topic_id	INT(10)	Referenz zu einem Plugin für das Thema

Tabelle 2.12.: tab:ost-help-topic

3. Problemanalyse

3.1. USE-Case-Analyse

Akteure:

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

Name	Anmelden am Portal
ID	C00
Beschreibung	Ein Anwender/Systembetreuer nutzt seine Login-Daten um sich in das System einzuwählen.
Akteure	Anwender, Systembetreuer
Häufigkeit	5/5
Auslöser	Der Anwender möchte das Portal benutzen.
Bedingungen	Anmeldedaten vorhanden und Rechte vergeben.
Endzustand	Eine Session wurde geöffnet und der Anwender kann das System nutzen.
Hauptablauf	1. Benutzerdaten eingeben 2. Login Vorgang initiieren
Ausnahmen	Error → Anmeldedaten prüfen. Error → Anmeldedaten neu beziehen.

Tabelle 3.1.: Use-Case C00

Name	Erstellung eines Standardtickets
ID	C01
Beschreibung	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.
Akteure	Anwender
Häufigkeit	2/5
Auslöser	Es wird eine Komponente/Ressource für die Schule benötigt.
Bedingungen	C00
Endzustand	Ticket wurde in der Datenbank gespeichert.
Hauptablauf	<ol style="list-style-type: none">1. Formular öffnen2. Formular ausfüllen3. Ticket überprüfen4. Ticket einreichen
Ausnahmen	Error - Ticket kann nicht abgegeben werden → C01 erneut ausführen.

Tabelle 3.2.: Use-Case C01

Name	Erstellung eines Incident
ID	C02
Beschreibung	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.
Akteure	Anwender
Häufigkeit	4/5
Auslöser	Ein für den Anwender nicht lösbares technisches Problem.
Bedingungen	C00, Auslöser
Endzustand	Incident wurde in der Datenbank gespeichert und der Systembetreuer hat eine Benachrichtigung erhalten.
Hauptablauf	<ol style="list-style-type: none">1. Formular öffnen2. Formular ausfüllen3. Incident überprüfen4. Incident einreichen
Ausnahmen	Error - Ticket kann nicht abgegeben werden → C02 erneut ausführen.

Tabelle 3.3.: Use-Case C02

Name	Ticketstatus prüfen
ID	C03
Beschreibung	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.
Akteure	Anwender
Häufigkeit	3/5
Auslöser	Anwender hat Anfrage erhalten
Bedingungen	C00, zu überprüfendes Ticket muss vorhanden sein
Endzustand	Keine Veränderungen am System
Hauptablauf	1. Ticket öffnen 2. Status einsehen
Ausnahmen	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.

Name	Ticketlöschung beantragen
ID	C04
Beschreibung	Ein Anwender meldet sich am Portal an und beantragt die Löschung seines Tickets.
Akteure	Anwender
Häufigkeit	1/5
Auslöser	Problem hat sich erübrigt; Ticket ist falsch
Bedingungen	C00, zu löschendes Ticket muss vorhanden sein
Endzustand	Systembetreuer hat die Anfrage auf Löschung erhalten
Hauptablauf	1. Ticket öffnen 2. Löschung beantragen
Ausnahmen	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. Schlug 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

Name	Ticket löschen
ID	C05
Beschreibung	Der Systembetreuer meldet sich am Portal an und kümmert sich um Löschanfragen
Akteure	Systembetreuer
Häufigkeit	1/5
Auslöser	C04
Bedingungen	C00, zu löschendes Ticket muss vorhanden sein
Endzustand	Das Ticket wurde entfernt
Hauptablauf	<ol style="list-style-type: none">1. Ticket öffnen2. Ticket löschen
Ausnahmen	Error – Löschvorgang 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.

Name	Ticketstatus ändern
ID	C06
Beschreibung	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.
Akteure	Systembetreuer
Häufigkeit	4/5
Auslöser	C01
Bedingungen	C00, Ticket vorhanden
Endzustand	Der Ticketstatus wurde geändert
Hauptablauf	1. Ticket öffnen 2. Ticketstatus ändern
Ausnahmen	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.

Name	Incident bearbeiten
ID	C07
Beschreibung	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.
Akteure	Systembetreuer
Häufigkeit	3/5
Auslöser	C02
Bedingungen	C00, Incident vorhanden
Endzustand	Incident ist abgearbeitet
Hauptablauf	<ol style="list-style-type: none">1. Benachrichtigung erhalten2. Incident öffnen3. Incident bearbeiten4. C08
Ausnahmen	Error – Incident öffnen nicht erfolgreich → C07 erneut ausführen.

Tabelle 3.8.: Use-Case C07

3.2. Wireframes

Name	Ticket schließen
ID	C08
Beschreibung	Der Systembetreuer meldet sich am Portal an und schließt ein Ticket bzw. einen Incident nach dessen Erledigung.
Akteure	Systembetreuer
Häufigkeit	5/5
Auslöser	C01, C02
Bedingungen	C00, Ticket vorhanden und abgearbeitet
Endzustand	Ticket ist geschlossen
Hauptablauf	1. Ticket öffnen 2. Ticket schließen
Ausnahmen	Error – Ticket konnte nicht geschlossen werden → C08 erneut ausführen.

Tabelle 3.9.: Use-Case C08

3.3. Prototyp

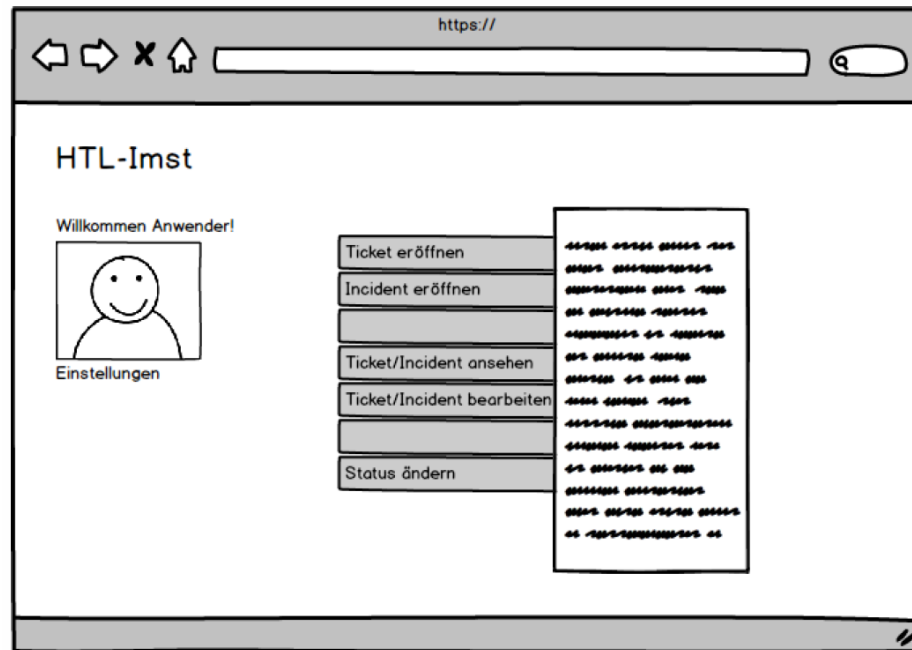


Abbildung 3.1.: Mockup Anwendersicht

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)

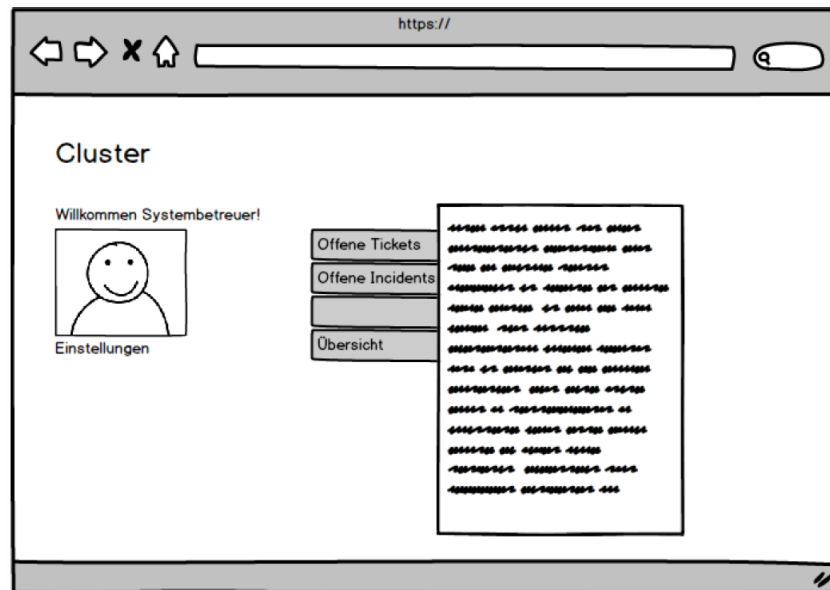


Abbildung 3.2.: Mockup Systembetreuer

3.5. User-Interface-Design

- Mockups
- Wireframes

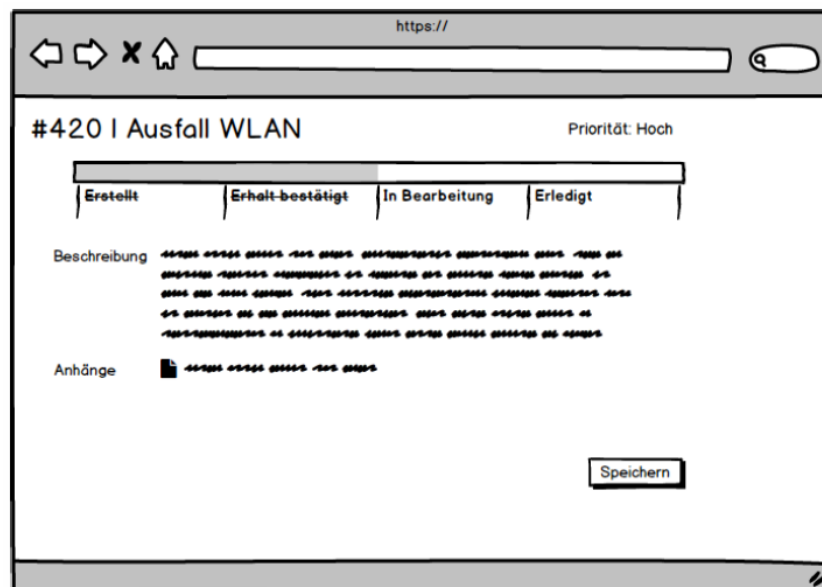


Abbildung 3.3.: Mockup Ticketstatus

Ausfall WLAN #420

Priorität: **hoch**

Erstellt

Erhalt bestätigt

In Bearbeitung

Erledigt

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.

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. Entwurf/Machbarkeitsfeststellung eines kleinen Java EE Ticketsystems

Aufgrund der bereits geschilderten Nicht Durchführbarkeit einer Adaption von wurde ein Prototyp für ein Ticketsystem entworfen, welcher die vom Landes-
schulrat und den Tiroler Schulen benötigten Funktionen mitbringt. Die Anforderungen für dieses System sind im Grunde die gleichen, welche zu Beginn an die -Adaption gestellt wurden.

Ein Auszug dieser Anforderungen sind:

- Usability: Das System soll ohne lange Schulungsphasen oder Einlernzeit verwendet werden können. Unter Verwendung ist hauptsächlich das Absetzen von Supporttickets definiert.
- Mobility: Das Web-Frontend soll auch auf Smartphones und Tablets verwendet werden können und die gleichen Funktionen wie auf dem PC liefern.
- Adaptability: Sollten sich Anforderungen, Best Practices oder Sicherheitsanforderungen ändern, sollen diese mit so geringem Aufwand als möglich implementiert werden können.

4.1. Technologie

Für den Prototyp des Ticketsystems wurde Java Enterprise Edition ausgewählt. Diese Entscheidung basiert auf der Absicht, die bei gezogenen Schlüssen zu beachten und die Probleme die bei auftraten, zu vermeiden. JavaEE erscheint hierfür besonders geeignet, da es (in dem Verwendungsmodus, der an der Schule gelehrt wurde) von sich aus das MVC-Entwurfsmuster anwendet (mehr im nächsten Abschnitt). Des Weiteren eignet sich JavaEE für die Beseitigung der Schwächen durch die relativ strengen Sprachkonventionen und die (beinahe) unausweichliche Objektorientierung.

4.2. Architektur

Als Basis für die Systemarchitektur wird das Model View Controller Muster verwendet. Das bedeutet die Trennung zwischen JavaBeans (Model), die direkt mit der Persistenzebene (Datenbank) arbeitet, der Benutzerschnittstelle (View; Webschicht) und der Logik (Controller; Anwendungsschicht).

Das Lehrbuch fasst das Entwurfsmuster wie folgt zusammen und bringt dessen Sinn sowie Existenzberechtigung im Evaluationsprogramm „Ticketsystem“ auf den Punkt:

Das MVC ist ein Muster, das vorgibt, wie Darstellung, Logik und Daten in einer Applikation getrennt werden sollen. Ziel dieser Trennung ist die Verbesserung der Programmstruktur und damit die Wartbarkeit, Erweiterbarkeit, und Wiederverwendbarkeit des Codes. Das Modell kapselt die Daten und enthält je nach MVC-Ausprägung ggf. auch die fachliche Logik. Die View visualisiert das Modell und der Controller realisiert die Anwendungssteuerung. Der Controller reagiert auf Benutzerinteraktionen innerhalb der View.

und aktualisiert ggf. die Daten am Modell. Die View wiederum passt sich je nach Ausprägung des MVC entweder automatisch an das veränderte Modell an oder wird durch den Controller über die Ausprägung informiert. ?

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

4.3. Benutzerschnittstellen

Kompletter Entwurf aller Benutzerschnittstellen

4.4. Klassenentwurf

In Abbildung 4.1 ist ein simpler Entwurf der Model-Klassen des Ticketsystems als Klassendiagramm dargestellt. Der Benutzer (User) hält Informationen wie deren Namen, Kontaktdetails und eine Liste der vom User erstellten Tickets für die folgenden Rollen:

IT Manager bzw. Managerinnen (Managers): Eine Person an Tirols Schulen, verantwortlich für die lokale Infrastrukturbetreuung und Fehlerreporting

Systembetreuer bzw. Systembetreuerinnen (Supervisors): Bearbeitet Meldungen (Tickets) für einen Schulcluster

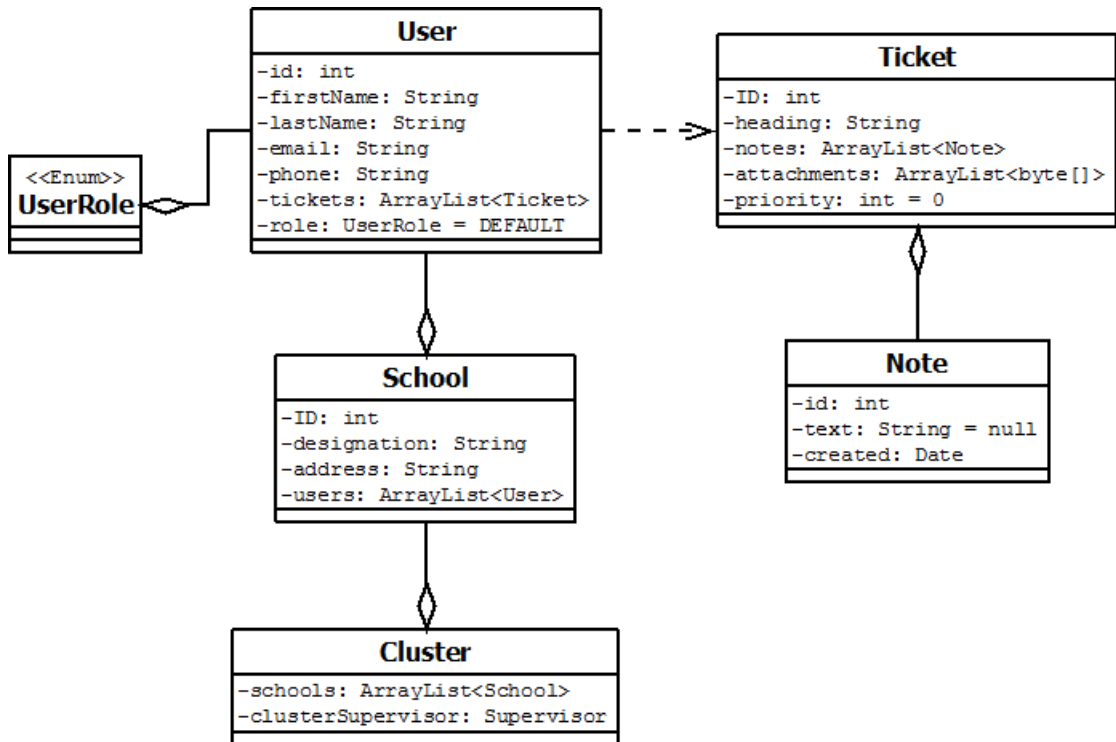


Abbildung 4.1.: Klassenentwurf Java EE Ticketsystem

Administratoren bzw. Administratorinnen (Admins): Eine Person des Landesschulrates, verantwortlich für das Gesamtsystem ().

Damit steht die Klasse **User** im Zentrum des Systems. Die **Ticket**-Klasse (oder eine Entität des Typs **Ticket**) hat eine Überschrift, eine Priorität, welche vom **User** (meist IT Manager(in)) definiert wird, eine Liste an **Notes** (Fehlerbeschreibungen, Nachrichten in Blogpost-Form) und eine Menge an Dateianhängen.

- 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 Zeilennummern 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
- Testfall 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

Abbildungsverzeichnis

1.1. Risikomatrix	15
1.2. IST-Zustand OS-Ticket	16
3.1. Mockup Anwendersicht	50
3.2. Mockup Systembetreuer	51
3.3. Mockup Ticketstatus	52
3.4. Prototyp Ticketstatus	53
4.1. Klassenentwurf Java EE Ticketsystem	57

Tabellenverzeichnis

1.1. Analyse Einwirkung & Auswirkung	15
1.2. Stakeholder Identifikation	19
1.3. Stakehodler Klassifikation	19
2.1. tab:ost-attachment	29
2.2. tab:ost-canned-response	30
2.3. tab:ost-config	31
2.4. tab:ost-content	31
2.5. tab:ost-department	32
2.6. tab:ost-faq	33
2.7. tab:ost-staff	34
2.8. tab:ost-ticket	38
2.9. tab:ost-user	39
2.10. tab:ost-user-account	39
2.11. tab:ost-ticket-priotity	40
2.12. tab:ost-help-topic	40
3.1. Use-Case C00	41
3.2. Use-Case C01	42
3.3. Use-Case C02	43
3.4. Use-Case C03	44
3.5. Use-Case C04	45
3.6. Use-Case C05	46
3.7. Use-Case C06	47
3.8. Use-Case C07	48

3.9. Use-Case C08	49
-----------------------------	----

Quelltexte

2.1.	main.inc.php	25
2.2.	pwreset.php	26
2.3.	offline.php	27
2.4.	client.inc.php	27

A. Anhang-Kapitel

A.1. Anhang-Section

Testtext