April 2016, v1.9

IPA-Bericht

Samuel Keller

Änderungen:

1.9

Implementationsvorgang weitergeführt, Textänderungen

1.7&1.8

Texte verfasst & Qualitätssicherung eingefügt

1.6

Implementierungs-Text hinzugefügt.

Tagesjournal 8

Änderungen im Zeitplan

1.5

Tagesjournal 7 hinzugefügt

Schreibfehleränderungen in Journale

1.4.

Journal Tag 6 erweitert

Formatierungen der Journale verändert

Funktionen, die in einer Version 2.0 hinzugefügt werden könnten, dokumentiert.

User deaktivieren Feature eingebaut

**Inhaltsverzeichnis**

[1 Obligatorischer Teil 6](#_Toc448404200)

[1.1 Ausgangslage 6](#_Toc448404201)

[1.2 Detaillierte Aufgabenstellung 6](#_Toc448404202)

[1.2.1 Login 6](#_Toc448404203)

[1.2.2 Benutzerverwaltung 6](#_Toc448404204)

[1.2.3 Die Verwaltung von Tickets 6](#_Toc448404205)

[1.2.4 Ticket Workflow 7](#_Toc448404206)

[1.2.5 Dokumentation 7](#_Toc448404207)

[1.2.6 Design 7](#_Toc448404208)

[1.2.7 Testing 7](#_Toc448404209)

[1.2.8 Versionsverwaltung 8](#_Toc448404210)

[1.2.9 Abgrenzungen 8](#_Toc448404211)

[1.2.10 Zeitplan 8](#_Toc448404212)

[1.2.10.1 Initialisierung (Montag, 04.04.2016) 8](#_Toc448404213)

[1.2.10.2 Analyse (Dienstag, 05.04.2016) 8](#_Toc448404214)

[1.2.10.3 Entwurf (Mittwoch, 06.04.2016) 8](#_Toc448404215)

[1.2.10.4 Implementierung (Mittwoch, 06.04.2016 - Dienstag, 12.04.2016) 8](#_Toc448404216)

[1.2.10.5 Testing (Montag, 11.04.2016 - Dienstag, 13.04.2016) 8](#_Toc448404217)

[1.2.10.6 Abschlussphase (Montag, 13.04.2016 - Dienstag, 14.04.2016) 8](#_Toc448404218)

[1.2.10.7 Buffer (Freitag, 15.04.2016) 8](#_Toc448404219)

[1.2.11 Mittel und Methoden 9](#_Toc448404220)

[1.2.12 Vorkenntnisse 9](#_Toc448404221)

[1.2.13 Vorarbeiten 9](#_Toc448404222)

[1.2.14 Neue Lerninhalte 9](#_Toc448404223)

[1.3 Projektorganisation 10](#_Toc448404224)

[1.3.1 Projektmanagementmethode 10](#_Toc448404225)

[1.3.1.1 Initialisierungsphase 10](#_Toc448404226)

[1.3.1.2 Analysephase 10](#_Toc448404227)

[1.3.1.3 Entwurfsphase 10](#_Toc448404228)

[1.3.1.4 Implementierungsphase 10](#_Toc448404229)

[1.3.1.5 Testingphase 10](#_Toc448404230)

[1.3.1.6 Abschlussphase 10](#_Toc448404231)

[1.3.2 Zeitplan 11](#_Toc448404232)

[1.3.2.1 Vorgang 12](#_Toc448404233)

[1.4 Arbeitsjournal 13](#_Toc448404234)

[1.4.1 Montag, 4. April 2016 13](#_Toc448404235)

[1.4.2 Dienstag, 5. April 2016 14](#_Toc448404236)

[1.4.3 Mittwoch, 6. April 2016 15](#_Toc448404237)

[1.4.4 Donnerstag, 7. April 2016 16](#_Toc448404238)

[1.4.5 Freitag, 8. April 2016 17](#_Toc448404239)

[1.4.6 Montag, 11. April 201 18](#_Toc448404240)

[1.4.7 Dienstag, 12. April 2016 19](#_Toc448404241)

[1.4.8 Mittwoch, 13. April 2016 20](#_Toc448404242)

[1.4.9 Donnerstag, 14. April 2016 21](#_Toc448404243)

[1.4.10 Freitag, 15. April 2016 22](#_Toc448404244)

[2 Projektdokumentation 23](#_Toc448404245)

[2.1 Soll / Ist vergleich 23](#_Toc448404246)

[2.1.1 Ausgangssituation 23](#_Toc448404247)

[2.1.2 Endsituation 24](#_Toc448404248)

[2.2 Risikoanalyse 25](#_Toc448404249)

[2.2.1 Datenverlust 25](#_Toc448404250)

[2.2.1.1 Wahrscheinlichkeit 25](#_Toc448404251)

[2.2.1.2 Folgen 25](#_Toc448404252)

[2.2.1.3 Gegenmassnahmen 25](#_Toc448404253)

[2.2.1.4 Schadensminimierung 25](#_Toc448404254)

[2.2.2 Zeitprobleme 25](#_Toc448404255)

[2.2.2.1 Wahrscheinlichkeit 26](#_Toc448404256)

[2.2.2.2 Folgen 26](#_Toc448404257)

[2.2.2.3 Gegenmassnahmen 26](#_Toc448404258)

[2.2.2.4 Schadensminimierung 26](#_Toc448404259)

[2.2.3 Internetbrowser-Probleme 26](#_Toc448404260)

[2.2.3.1 Wahrscheinlichkeit 26](#_Toc448404261)

[2.2.3.2 Folgen 26](#_Toc448404262)

[2.2.3.3 Gegenmassnahmen 26](#_Toc448404263)

[2.2.3.4 Schadensminimierung 27](#_Toc448404264)

[2.3 Versionierung 27](#_Toc448404265)

[2.3.1 Backup 27](#_Toc448404266)

[2.3.2 Commits 27](#_Toc448404267)

[2.4 Wieso Bootstrap 28](#_Toc448404268)

[2.4.1 Was ist Bootstrag 28](#_Toc448404269)

[2.4.2 Vorteile 28](#_Toc448404270)

[2.4.3 Nachteile 28](#_Toc448404271)

[2.5 Wieso Bootstrap benutz wird 28](#_Toc448404272)

[3 Design 29](#_Toc448404273)

[3.1 Datenbank 29](#_Toc448404274)

[3.1.1 Vor der Implementierung 29](#_Toc448404275)

[3.1.2 Nach der Erstellung der Applikation 30](#_Toc448404276)

[4 Implementierung 31](#_Toc448404277)

[4.1 Standard GUI 31](#_Toc448404278)

[4.1.1 header.inc.php 31](#_Toc448404279)

[4.1.2 navigation.inc.php 31](#_Toc448404280)

[4.1.3 footer.inc.php 31](#_Toc448404281)

[4.2 Autorisierungs-Seiten 32](#_Toc448404282)

[4.2.1 checkauth.inc.php 32](#_Toc448404283)

[4.2.2 checkrole.inc.php 32](#_Toc448404284)

[4.3 Loginseiten 32](#_Toc448404285)

[4.3.1 login.php 32](#_Toc448404286)

[4.3.2 logout.php 32](#_Toc448404287)

[4.4 Datenbank Verbindungsseiten 32](#_Toc448404288)

[4.4.1 dbsettings.inc.php 33](#_Toc448404289)

[4.4.2 dbconnection.inc.php 33](#_Toc448404290)

[4.4.3 pdosettings.inc.php 33](#_Toc448404291)

[4.4.4 pdoclass.inc.php 33](#_Toc448404292)

[4.5 Funktionsseiten 33](#_Toc448404293)

[4.5.1 functioncontroller.inc.php 33](#_Toc448404294)

[4.5.2 sendmail.inc.php 33](#_Toc448404295)

[4.6 Ticketverwaltungsseiten 34](#_Toc448404296)

[4.6.1 index.php 34](#_Toc448404297)

[4.6.2 ticketfilter.inc.php 34](#_Toc448404298)

[4.6.3 ticketliste.inc.php 35](#_Toc448404299)

[4.6.4 neuesticket.php 35](#_Toc448404300)

[4.6.5 ticket\_details.php 35](#_Toc448404301)

[4.7 Userverwaltung 35](#_Toc448404302)

[4.7.1 profil.php 35](#_Toc448404303)

[4.7.2 usermanagement.php 35](#_Toc448404304)

[4.7.3 neuernutzer.php 35](#_Toc448404305)

[4.7.4 benutzer\_details.php 35](#_Toc448404306)

[5 Testing 37](#_Toc448404307)

[5.1 Testfälle 37](#_Toc448404308)

[5.2 Testprotokolle 46](#_Toc448404309)

[5.2.1 Testprotokoll #1 46](#_Toc448404310)

[5.2.2 Testprotokoll #2 52](#_Toc448404311)

[6 Abschlussbericht 65](#_Toc448404312)

[6.1 Rückblick 65](#_Toc448404313)

[6.2 Lessons learned & Was ist gut gelaufen 65](#_Toc448404314)

[6.3 Was schief gelaufen ist 65](#_Toc448404315)

[6.4 Mit welchen Funktionen kann das Projekt sinnvoll erweitert werden? 65](#_Toc448404316)

[6.4.1 Tickets und Benutzer in der Detailübersicht 65](#_Toc448404317)

[6.4.2 Benutzerfilter und/oder Ansicht für deaktivierte Nutzer 65](#_Toc448404318)

[6.4.3 Passwort ändern Funktion auf eine eigene Seite auslagern 65](#_Toc448404319)

[6.4.4 Dem Ticket als Administrator einen Kommentar hinzuzufügen 65](#_Toc448404320)

[6.5 Resultat 66](#_Toc448404321)

[6.6 Schlusswort 66](#_Toc448404322)

[7 Quellenverzeichnis 66](#_Toc448404323)

[7.1 Versionierung 66](#_Toc448404324)

[7.2 Risikoanalyse 66](#_Toc448404325)

[7.3 Bootstrap 66](#_Toc448404326)

[8 Glossar 66](#_Toc448404327)

[9 Anhang 66](#_Toc448404328)

# Obligatorischer Teil

## Ausgangslage

Die Abteilung Web Service ist für verschiedenen Webapplikationen der Schweizerischen Bankiervereinigung verantwortlich. Sie ist zuständig für die Weiterentwicklung, die Aufrechterhaltung und teilweise Bewirtschaftung der Webapplikationen. Die Aufträge für die Bewirtschaftung der Webseiten werden via E-Mail oder in wenigen Fällen via Telefon an die Webabteilung mitgeteilt. Dies hat zur Folge, dass vielmals der Überblick über die Aufträge nicht mehr gewährleistet ist, und somit die Fehleranfälligkeit erhöht. Die Koordination und Konsolidierung der Aufträge durch ein zentrales webbasierter Ticketingsystem würde für die Benutzer sowie für die Webabteilung eine grosse Entlastung bringen.

## Detaillierte Aufgabenstellung

Im Rahmen der Prüfungsarbeit IPA soll ein webbasiertes Ticketingsystem für die Webabteilung entwickelt werden. Die Arbeit ist auf die Grundfunktionen der Applikation und die Usability ausgerichtet.

Das Ticketingsystem besteht aus drei Komponenten:

### Login

Die Implementierung eines einfachen und benutzerfreundlichen Log-ins für den Benutzer.

### Benutzerverwaltung

Es existieren zwei Benutzergruppen: Benutzer und Administratoren.

Benutzer können neue Tickets erstellen und ihre persönlichen Informationen, bestehend aus E-Mail, Telefonnummer und Passwort, bearbeiten. Administratoren können zusätzlich zu den Benutzern folgende Aktivitäten vornehmen: Jedem Ticket einen Administrator zuweisen, alle Tickets bearbeiten, den Status des Tickets verändern, neue Nutzer erstellen und deren persönlichen Informationen bearbeiten.

Die persönlichen Informationen sind über ein Formular erreichbar.

### Die Verwaltung von Tickets

Nach dem Einloggen werden die Benutzer auf eine Übersichtsseite weitergeleitet. Auf dieser Seite sind alle aktiven Tickets publiziert. Folgende Kriterien eines Tickets sind ersichtlich: das Erfassungsdatum, den Status des Tickets, eine allfällige Deadline, den Tickettitel, die Person, welche das Ticket erstellt hat sowie die Person, die das Ticket bearbeitet. Die Ticketübersicht kann nach dem Status des Tickets, der Kategorie, dem Ersteller und dem Administrator gefiltert werden.

Tickets können in einer Detailansicht geöffnet werden. In dieser Ansicht ist die genaue Beschreibung des Problems ersichtlich. Die Administratoren können in dieser Ansicht die Ticketinformationen verändern.

### Ticket Workflow

Jeder Benutzer hat die Berechtigung, ein neues Ticket zu erfassen. Das Ticket kann mittels eines Formulars erfasst werden. Im Formular können die Ticketinformationen (bestehend aus dem Tickettitel, der Kategorie, der Ticketbeschreibung und einer allfälligen Deadline) angegeben werden. Zudem kann er eine Datei z.B. ein Printscreen hochladen und anschliessend das Ticket zur Bearbeitung abschicken. Nach dem Abschicken des Tickets erhalten die Administratoren eine E-Mail Benachrichtigung. Ein Administrator kann dem Ticket einen Administrator zuweisen, welcher sich um das Ticket kümmert. Nach der Zuweisung erhält der Erfasser des Tickets eine E-Mail mit der Meldung, dem Ticket sei ein Administrator zugewiesen worden. Der Erfasser des Tickets erhält ebenfalls eine E-Mail, wenn der Status des Tickets verändert wird. Mit dem Status „Archiv“ wird das Ticket aus der Standardansicht, die alle offenen Tickets anzeigt, ausgeblendet.

### Dokumentation

Die IPA-Dokumentation enthält die Ausgangslage des Projektes, den Ist-Zustand sowie den Sollzustand, die Analyse des Projektes, das Vorgehen in der Entwicklung sowie das Testing der Applikation. Zudem geht die Dokumentation darauf ein, mit welchen zusätzlichen Optionen die Applikation nach der Arbeit erweitert werden könnte.

Es wird ein Benutzerhandbuch erstellt, das dem Benutzer als Unterstützung dient. Dieses kann der Benutzer auf der Ticketing Website herunterladen.

### Design

Die Webapplikation ist übersichtlich und benutzerfreundlich gestaltet. Sie ist programmatisch aufgebaut und hat ein schlichtes Design. Jedes Formular wird validiert.

### Testing

Getestet werden die Grundfunktionen der Applikation mittels mehreren Testfällen für Tickets, Benutzer und Administratoren. Ebenfalls werden alle Formulare mit Grenzwerten überprüft. Ein Testfall besteht aus einer Nummer mit einem Titel, einer Aktion und aus dem erwarteten Resultat. Die Testfälle werden während der Analysephase festgelegt.

Die Testfälle werden übersichtlich in einer Tabelle dargestellt. In dieser wird der Testfall mit Nummer und Titel angegeben. Zusätzlich sind das Datum des Tests, die Testperson sowie das Resultat festgehalten. Das Resultat hat die Optionen: „Wie erwartet“, „Nicht implementiert“ sowie „Funktioniert nicht“. „Wie erwartet“ wird nur vergeben, wenn der Testfall in den drei Browser Mozilla Firefox, Google Chrome und Microsoft Internet Explorer erfolgreich war. Die Testfälle werden ebenfalls separat in Testprotokollen festgehalten.

### Versionsverwaltung

Die Versionsverwaltung wird mit einem git GUI-Client erstellt.

### Abgrenzungen

* Installation und Wartung von Hard- und Software ist nicht Teil der Prüfung.
* Verwendung des Frameworks Bootstrap.
* Die Applikation ist so entwickelt, dass es auf den gängigen Browsern auf einem Desktop funktioniert, jedoch nicht auf Tablets und Smartphones.

### Zeitplan

Der Zeitplan und die beschriebenen Arbeitsschritte können sich noch ändern. An der Dokumentation wird jeden Tag gearbeitet.

#### Initialisierung (Montag, 04.04.2016)

Grundgerüst der Dokumentation, erste Texte erstellen, GIT einrichten

#### Analyse (Dienstag, 05.04.2016)

Analyse der Webseite, Erstellung der Testfälle

#### Entwurf (Mittwoch, 06.04.2016)

Erstellung des Webseitenlayouts.

#### Implementierung (Mittwoch, 06.04.2016 - Dienstag, 12.04.2016)

Implementierung der drei Komponenten Log-in, Benutzerverwaltung und Ticketverwaltung.

#### Testing (Montag, 11.04.2016 - Dienstag, 13.04.2016)

Testen der Applikation und Erstellung des Testberichtes

#### Abschlussphase (Montag, 13.04.2016 - Dienstag, 14.04.2016)

Abschluss der IPA-Dokumentation und des Benutzerhandbuches.

#### Buffer (Freitag, 15.04.2016)

Ein Buffertag für nötige Ergänzungen in der Dokumentation und der Programmierung.

### Mittel und Methoden

* PHP
* CSS
* HTML
* MySQL
* Git
* Eclipse

### Vorkenntnisse

* HTML & CSS: 2 Jahre
* PHP: 2 Monate
* MySQL: Basiswissen

### Vorarbeiten

* Es werden im Voraus die Dokumentvorlagen für die Dokumentation der Prüfung vorbereitet, damit ab dem ersten Tag bereits dokumentiert werden kann.
* Erstellung einer statischen Website mit Referenzlayout.
* Die Datenbank wird im Voraus geplant und erstellt.

### Neue Lerninhalte

* PHP Log-in mit Usermanagement
* PHP in Verbindung mit MySQL
* PHP E-Mail Versand

## Projektorganisation

### Projektmanagementmethode

Der Auftrag ist mit knapp zwei Wochen als kleines Projekt anzusehen. Das Wasserfall-Modell ist ideal für diese Art von Projekt. Die klare Einteilung in Phasen und komplexe Strukturen macht das Modell übersichtlich. Das IPERKA-Modell hätte ich auch nutzen können. Dieses wurde in der Schule auch als Alternative beworben. Mangels Kenntnisse und Sicherheit in dieser Methode wurde schliesslich aber das Wasserfall-Modell gewählt.

Das Wasserfall-Modell wurde in die folgenden Phasen unterteilt:

#### Initialisierungsphase

In dieser Phase werden die Vorbereitungen für das anschliessende Projekt getroffen. Beispielsweise wird die Grundstruktur der Dokumentation und ein Zeitplan erstellt, sowie eine Back-up-Lösung angelegt

#### Analysephase

Die Ausgangssituation wird genauer Analysiert und mit Text und Grafiken beschrieben. Mit einem Soll / Ist-Vergleich und einer Risikoanalyse werden mögliche Gefahren identifiziert, um diese in den weiteren Phasen umgehen zu können.

#### Entwurfsphase

In der Entwurfsphase werden vorgehende Diagramme und Konzepte erweitert. Ebenso wird ein GUI-Entwurf erstellt. Aufgrund dieses wird dann die Implementation geplant.

#### Implementierungsphase

In der Implementierungsphase wird das Projekt, welches in den vorigen Phasen geplant wurde, umgesetzt.

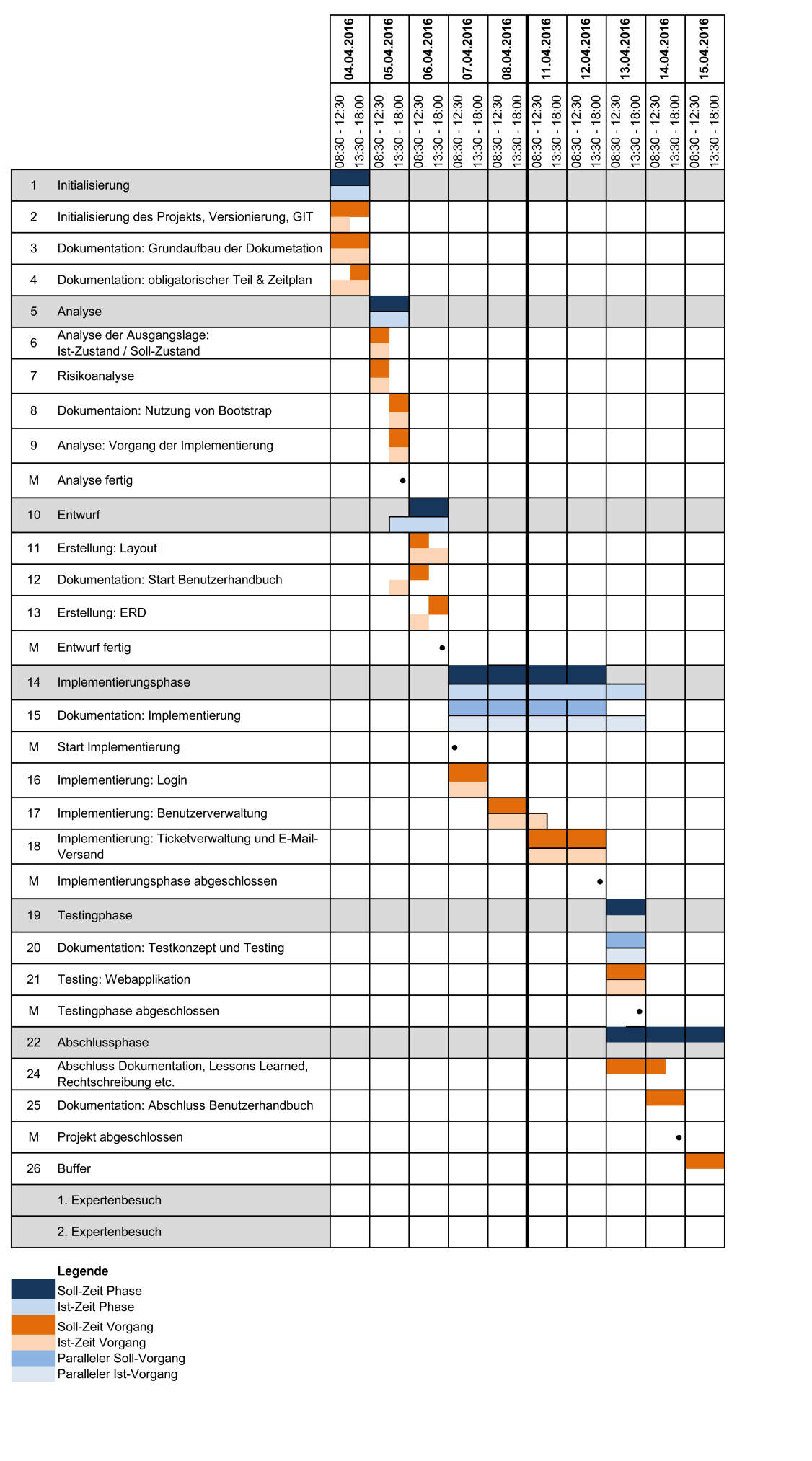
#### Testingphase

In dieser Phase wird die Qualitäts der Applikation sichergestellt. Aufgrund eines Testkonzepts wird ein Test-Szenario erstellt. Auf diesem basieren Testfälle, mit denen man die Software auf ihre Funktionen testet und auswertet. Diese werden in Test-Protokollen festgehalten.

#### Abschlussphase

In der Abschlussphase werden die letzten Arbeiten an dem Projekt beendet. Die Dokumentation wird auf Orthographie und Grammatik geprüft, und die letzten Dokumente werden geschrieben und der Kriterienkatalog überprüft. Am Schluss wird ein ausführlicher Rückblick auf das Projekt verfasst und das Benutzerhandbuch geschrieben

### Zeitplan



## Arbeitsjournal

### Montag, 4. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsplatz einrichten * Lokales Repository erstellen * Repository auf GitHub erstellen * Dokumentstruktur erstellen * Dokumentation des obligatorischen Teils anfangen * Versionierung und Back-up-Lösung des Repositorys definieren und dokumentieren * Zeitplan erstellen |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Für das USB-Stick-Back-up habe ich in der Mittagspause einen neuen USB-Stick gekauft. |
| Geplante Arbeiten nach Arbeitsschluss | Texte zu Hause korrigieren lassen. |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | Ich habe Detlef Nünninghoff nach einer Vorlage für die Dokumentation mit dem SwissBanking Design gefragt und gezeigt bekommen, wo ich dieses finde. |
| Soll / Ist Zeitvergleich | Alles läuft nach Zeitplan. Jedoch habe ich gewisse Schritte in einer anderen Reihenfolge bearbeitet als im Zeitplan definiert. Das liegt nicht daran, dass es so unbedingt mehr Sinn ergibt, sondern daran, dass ich gerne zwischen den einzelnen Dokumenten und Texten wechsle. So versteife ich mich auch nicht auf ein Dokument und/oder Textteil und kann laufend Dokumente ergänzen. |
| Sonstiges | Dokumentationstexte werden separat geschrieben und später zusammengetragen. |
| Arbeitszeit | 08:30 Uhr – 10:15 Uhr  Kurze Pause  10:20 Uhr – 12:30 Uhr  Mittagspause  13:30 Uhr – 17:00 Uhr  Kurze Pause  17:10 Uhr – 18:00 Uhr |
| Visum |  |

### Dienstag, 5. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Versionierungstext in die Dokumentation übernehmen * Arbeitsjournal (Tag 1) in die Dokumentation übernehmen * Risikoanalyse dokumentieren * „Wieso Bootstrap“ dokumentieren * Soll / Ist-Use-Case erstellen * Implementierungsvorgang (Planung) dokumentieren |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | Kurzes Brainstorming zur Nutzerhandbuch-Struktur |
| Ungeplante Arbeiten | keine |
| Geplante Arbeiten nach Arbeitsschluss | Texte zu Hause korrigieren lassen. |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | Nach der Erstellung der Soll / Ist-Grafiken habe ich diese meinem Vater und Alexandra Arni (FV) gezeigt. Die Modellstruktur blieb so wie geplant, die Darstellung änderte sich. |
| Soll / Ist Zeitvergleich | Ich bin im Moment im Zeitplan. Das Benutzerhandbuch habe ich bereits mit einer Inhalts-Struktur angefangen und über das Testkonzept bin ich mir bereits am Gedanken machen. |
| Sonstiges | Ich habe die Ausgangslage mit dem Soll / Ist-Vergleich verbunden, da ein Text zur Ausgangslage schon vorhanden ist. Das Test-Konzept wurde zum Testing verschoben. |
| Arbeitszeit | 08:30 Uhr – 10:00 Uhr  Kurze Pause  10:10 Uhr – 12:30 Uhr  Mittagspause  13:30 Uhr – 15:00 Uhr  Kurze Pause  15:10 Uhr – 18:00 Uhr |
| Visum |  |

### Mittwoch, 6. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 2) in die Dokumentation übernehmen * Use-Case dokumentieren * Bootstraptext in die Dokumentation übernehmen * Risikoanalyse in die Dokumentation übernehmen * Zwei ERD’s erstellen * Webseitenentwurf erstellen, auf dem bei der Implementierung aufgebaut werden kann * Benutzerhandbuch mit Text erweitern |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Ich habe zwei zusätzliche Use-Case Grafiken erstellt, die den Kommunikationsweg aufzeigen. |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | keine |
| Soll / Ist Zeitvergleich | Laut Zeitplan ist alles in Ordnung. |
| Sonstiges | Da ich die ERD’s mit dem Tool „MySQL Workbench“ erstellt habe, habe ich etwas länger für die Grafiken benötigt. Der Webseitenentwurf gefällt mir sehr gut und ich freue mich auf die Implementierung. Das Benutzerhandbuch hat eine genauere Gliederung sowie neue Texte erhalten. |
| Arbeitszeit | 08:30 Uhr – 10:15 Uhr  Kurze Pause  10:20 Uhr – 12:40 Uhr  Mittagspause  13:40 Uhr – 18:00 Uhr |
| Visum |  |

### Donnerstag, 7. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 3) in die Dokumentation übernehmen * Datenbank erstellen und teilweise anpassen * Datenbankanbindung erstellen * Login und Session implementieren * Navigation fertigstellen |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Ich habe gewisse Feldtypen in der Datenbank angepasst. Am Anfang hatte jedes Feld den Typ „Text“. Dieser Typ wird nach und nach bei mehreren Tabellen ersetzt, da „Text“ nicht für den Benutzernamen oder eine Telefonnummer geeignet ist. |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | Die Implementierung des Logins hat mich mehr Zeit gekostet als erwartet, da ich noch nicht viel Erfahrung mit MySQLi habe. Ich habe zuvor immer mit PDO-Verbindungen gearbeitet. Da die IPA aber ausschliesslich Prozedural ist, konnte ich PDO leider nicht verwenden. Der Implementierungaufwand mit PDO wäre um einiges geringer gewesen. |
| Unterstützung Dritter | Ich habe die Datenbankanpassung mit Detlef Nünninghoff besprochen. Er hat ebenfalls empfohlen die Feldtypen zu ändern. |
| Soll / Ist Zeitvergleich | Für heute war die Implementation des Logins angedacht. Dieses wurde vollumfänglich implementiert. |
| Sonstiges | Ich bin froh, dass ich endlich bei der Implementierung angekommen bin. Dieser Teil des Projekts macht mir am meisten Spass. |
| Arbeitszeit | 08:30 Uhr – 13:10 Uhr  Mittagspause  13:40 Uhr – 16:00 Uhr  Kurze Pause  16:10 Uhr – 17:30 Uhr |
| Visum |  |

### Freitag, 8. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 4) in die Dokumentation übernehmen * Implementierung des Benutzerprofilos mit Validierung * Implementierung der Funktion „Neuer Nutzer erstellen“ * Implementierung der Benutzerliste |
| Nicht erreichte Ziele | Ich hatte nicht genug Zeit, um die Detailansicht der Benutzer zu erstellen. Diese muss am Montag und Dienstag zusammen mit den Ticketfunktionen umgesetzt werden, um wieder im Zeitplan zu liegen. |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Dokumentierung der eingefügten Funktionen per Hand. |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | keine |
| Soll / Ist Zeitvergleich | Wie bereits erwähnt, bin ich nicht mehr im Zeitplan, da ich nicht genug Zeit hatte, um alles an einem Tag umzusetzen. |
| Sonstiges | Da ich bemerkt habe, dass es mir zeitlich nicht für die Benutzerübersicht reichen wird, habe ich angefangen, Diagramme der Funktionen / Teilprogramme zu zeichnen. |
| Arbeitszeit | 08:30 Uhr – 12:40 Uhr  Mittagspause  13:40 Uhr – 15:20 Uhr  Kurze Pause  15:30 Uhr – 17:30 Uhr |
| Visum |  |

### Montag, 11. April 201

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 5) übernehmen * Projektvorgangtext in die Dokumentation übernehmen * Soll / Ist Vergleichstext dokumentieren * Use-Case-Grafiken dokumentieren * Leere Tagesjournale in die Dokumentation einfügen * Detailansicht für die Benutzer implementieren * Benutzer-Löschfunktion implementieren * Datenbank angepasst * Nutzer Löschfunktion umgebaut * PDO Datenbankverbindung einbauen * Ticketliste implementieren * Ticketfilter implementieren |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | Während dem Dokumentieren ist mir eine zusätzliche Funktion eingefallen, mit dem ich das Ticketsystem in einer zweiten Version erweitern könnte. Diese habe ich im Abschlussbericht eingefügt. Da ich jetzt PDO nutze, und mit dem mehr vertraut bin, geht die Implementation der Funktionen einiges schneller. |
| Ungeplante Arbeiten | Ich habe der Nutzertabelle ein Feld mit dem Typ Boolean hinzugefügt, welches anzeigt ob der Nutzer aktiv ist oder nicht. Daraufhin wurden ebenfalls die SQLi Scripts vom Login und der Benutzerübersicht angepasst. |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | keine |
| Soll / Ist Zeitvergleich | Ich bin etwas in Zeitdruck, da die Benutzerübersicht ebenfalls heute implementiert werden muss. Kurz vor dem Expertenbesuch war ich damit fertig und bin jetzt wieder im Zeitplan. |
| Sonstiges | Ich habe ein paar der von Hand erstellten Texte endlich in die Dokumentation übernommen und alle zukünftigen Journale in die Dokumentation eingefügt, um Formatier Probleme entgegen zu wirken. Die neue Wordversion verträgt sich scheinbar nicht gut mit der älteren.  Die Experten waren heute für etwa eine halbe Stunde da. Diese Zeit werde ich nachholen. |
| Arbeitszeit | 08:30 Uhr – 11:15 Uhr  Expertenbesuch  11:45 Uhr – 12:30 Uhr  Mittagspause  13:30 Uhr – 18:35 Uhr |
| Visum |  |

### Dienstag, 12. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 6) übernehmen * Detailansicht für Tickets implementieren * Datenbank Feldtypen ändern * Datenbankeinträge zu den Kategorien ändern * „Ticket erstellen“-Funktion implementieren. * Benutzer- und Profilvalidierung vereinen * Fileupload implementieren * E-Mail versenden implementieren * Ticket und File-Validierung implementiere |
| Nicht erreichte Ziele | Es hat mir nicht gereicht, das Projekt ausführlich zu dokumentieren. Dies muss in den nächsten Tagen erledigt werden. |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Diverse Änderungen am Navigationsdesign |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | keine |
| Soll / Ist Zeitvergleich | Mit der Programmierung bin ich soweit fertig. Die Dokumentation ist aber hinter dem Zeitplan, weshalb ich in den nächsten Tagen neben dem Testen wohl noch viel Dokumentieren muss. |
| Sonstiges | Ich bin froh, dass die Applikation endlich fertig ist. |
| Arbeitszeit | 08:30 Uhr 12:30 Uhr  Mittagspause  13:30 Uhr – 18:00 Uhr |
| Visum |  |

### Mittwoch, 13. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 7) in die Dokumentation übernehmen * Testfällen erstellen * Testen * Kommentare im Code hinzufügen * Dokumentation mit Implementierungsteil erweitern. |
| Nicht erreichte Ziele | Implementierungs-Dokumentation fertigstellen. |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | Kleinere Bug fixes am Programm vor dem Testen. Diese sind im GIT-Backup erwähnt. |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme | keine |
| Unterstützung Dritter | Jonas Arab und Rolf Keller testen die Applikation für mich. |
| Soll / Ist Zeitvergleich | Die Applikation ist fertig. Die Dokumentation ist zwar laut Zeitplan hinterher, aber sicher auf dem richtigen Weg. Ich werde nicht alles ganz ausführlich dokumentieren können. |
| Sonstiges | Die Testprotokolle sind noch nicht bei der Dokumentation dabei, da diese erst nach 18:00 fertiggestellt wurden. |
| Arbeitszeit | 08:30 Uhr 12:30 Uhr  Mittagspause  13:30 Uhr – 18:00 Uhr |
| Visum |  |

### Donnerstag, 14. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 8) in die Dokumentation übernehmen * Dokumentieren * Qualitätssicherung in die Dokumentation übernehmen |
| Nicht erreichte Ziele | Code komplett auskommentieren |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | kleine |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme |  |
| Unterstützung Dritter |  |
| Soll / Ist Zeitvergleich |  |
| Sonstiges |  |
| Arbeitszeit | 08:30 Uhr 12:30 Uhr  Mittagspause  13:30 Uhr – 18:00 Uhr |
| Visum |  |

### Freitag, 15. April 2016

|  |  |
| --- | --- |
| Erreichte Ziele | * Arbeitsjournal (Tag 9) in die Dokumentation übernehmen * Arbeitsjournal (Tag 10) in die Dokumentation übernehmen |
| Nicht erreichte Ziele | keine |
| Frühzeitig erreichte Ziele | keine |
| Ungeplante Arbeiten | kleine |
| Geplante Arbeiten nach Arbeitsschluss | keine |
| Aufgetretene Probleme |  |
| Unterstützung Dritter |  |
| Soll / Ist Zeitvergleich |  |
| Sonstiges |  |
| Arbeitszeit |  |
| Visum |  |

# Projektdokumentation

## Ausgangslage

Die Abteilung Web Service ist für verschiedenen Webapplikationen der Schweizerischen Bankiervereinigung verantwortlich. Sie ist zuständig für die Weiterentwicklung, die Aufrechterhaltung und teilweise Bewirtschaftung der Webapplikationen. Die Aufträge für die Bewirtschaftung der Webseiten werden via E-Mail oder in wenigen Fällen via Telefon an die Webabteilung mitgeteilt. Dies hat zur Folge, dass vielmals der Überblick über die Aufträge nicht mehr gewährleistet ist.

#### Ist-Situtation

Die Grafik zeigt wie die Web-Abteilung in Kommunikation mit den anderen Nutzerinnen und Nutzer steht.

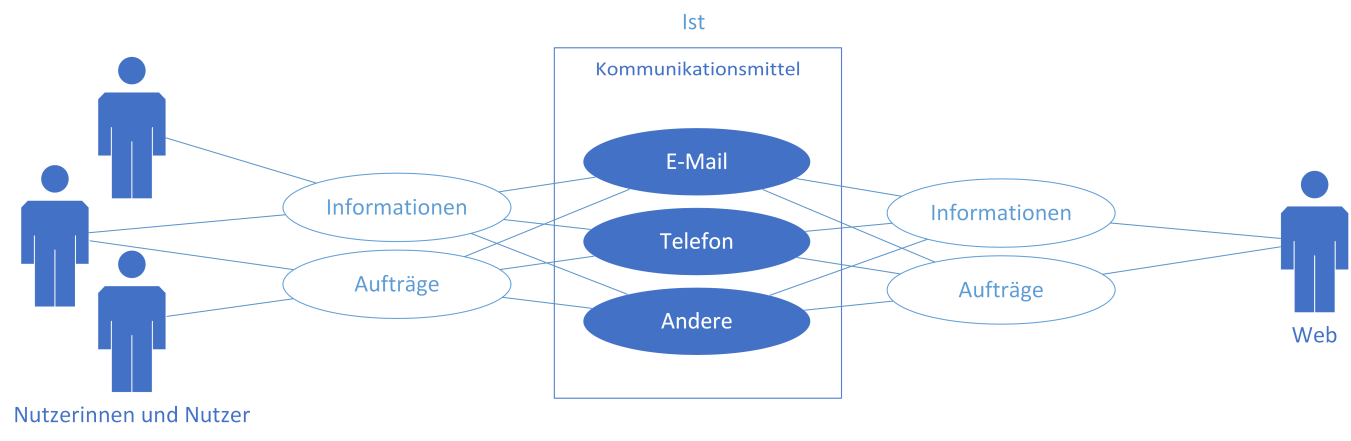


Abbildung: Aktueller Kommunikationsfluss der Nutzerinnen und Nutzer zum Web

Die Kommunikation zwischen den Nutzerinnen und Nutzer wird nicht ausschliesslich über ein Medium getätigt. Die Nutzerinnen und Nutzer kommunizieren hauptsächlich über E-Mail und Telefon mit dem Web. Das System hat den Nachteil, dass an arbeitsintensiven Tagen Informationen mit den Aufträgen vermischt werden. Da Informationen und Aufträge gleichzeitig auf allen Kommunikationsmitteln erhalten werden können, müssen diese beiden Komponenten vom Web wieder getrennt werden. Dabei kann sehr schnell die Übersicht verloren gehen.

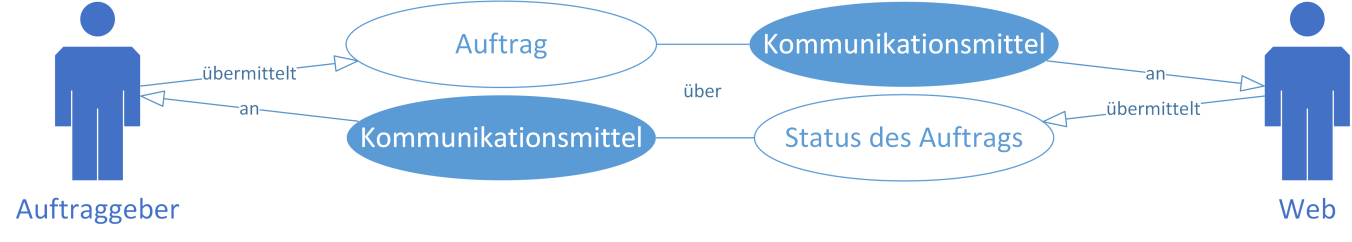


Abbildung: Kommunikation eines Auftrages

Durch den Austausch via Telefon oder E-Mail wissen immer nur die beteiligten über den Stand des Auftrages.

#### Soll-Situation

Die folgende Grafik stellt die Situation dar, die mit der Einführung des Ticketsystems eintreten soll.

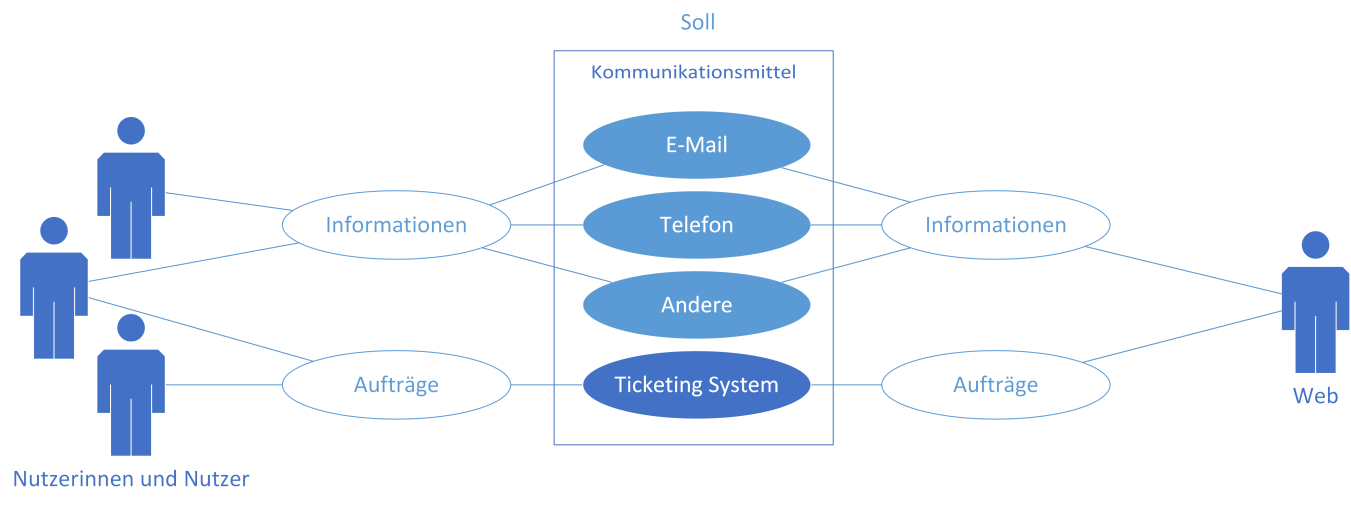


Abbildung: Aktueller Kommunikationsfluss der Nutzerinnen und Nutzer zum Web.

Wie vor der Einführung wird die Kommunikation nicht ausschliesslich über ein Medium getätigt. Die Nutzerinnen und Nutzer kommunizieren hauptsächlich über E-Mail und Telefon mit dem Web. Die Aufträge für das Web werden jedoch über das Ticketsystem kommuniziert. So werden an arbeitsintensiven Tagen die Informationen nicht mehr Aufträgen vermischt. Die Nutzerinnen und Nutzer ändern das Kommunikationsmittel für die Aufträge und erleichtern damit der Web-Abteilung die Übersicht.

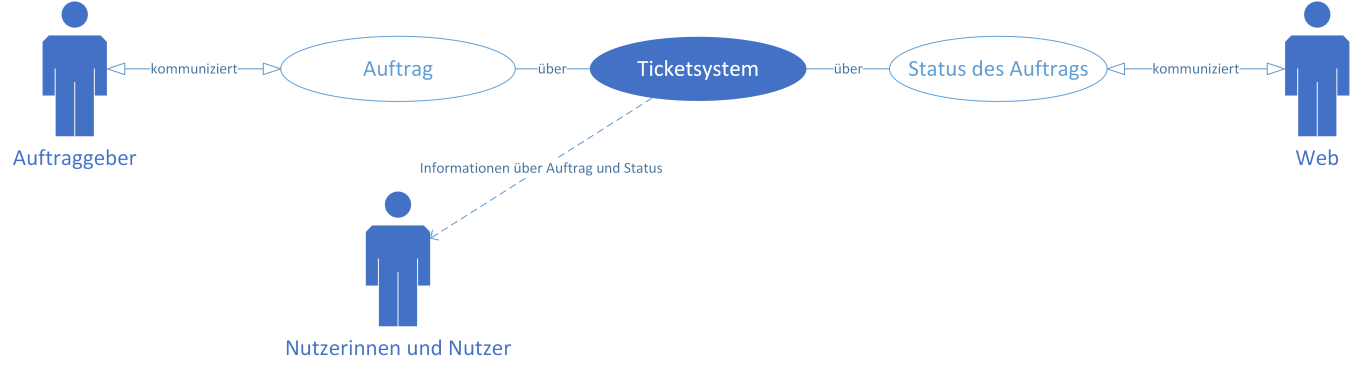


Abbildung: Kommunikation eines Auftrages nach der Einführung des Ticketsystems

Ein weiterer Vorteil des Ticketsystems ist, dass durch die Nutzung nicht nur der Auftraggeber der Stand des Auftrags weiss, sondern auch alle anderen Nutzer des Systems auf diese Informationen zugreifen können.

## Risikoanalyse

### Datenverlust

Datenverlust ist ein grosses Problem während der Arbeit. Der Verlust von Daten kann hauptsächlich durch Softwareprobleme, wie beispielsweise Absturz des Texteditors, oder durch Unachtsamkeit entstehen. Ohne entsprechende Schutzmassnahmen kann der Verlust von Daten fatale Folgen haben.

#### Wahrscheinlichkeit

Die Wahrscheinlichkeit eines Datenverlustes ist dank mehreren Back-ups sehr gering. Jedoch kann sich die Umschreibung eines Codeteiles auf die anderen Codeteile auswirken, ohne dass dies bemerkt wird. Wenn so grössere Fehler auftreten, muss auch auf eine jüngere Version zurückgesprungen werden, um das Rückverfolgen des Fehlers zu vereinfachen.

#### Folgen

Die Folgen eines Datenverlustes sind bedingt durch den Stand des Verlustes und des Standes des Backups. Bei einem grösseren Verlust kann dies zum Scheitern des Projektes führen. Falls die Auswirkungen nicht drastisch sind, wirkt es sich vor allem auf die Zeitplanung aus, da man zuerst den Stand von vor dem Datenverlust wieder erreichen muss.

#### Gegenmassnahmen

Es existiert ein USB-Stick-Back-up, ein lokales GIT-Back-up sowie ein öffentliches zugängliches Repository auf GitHub. Alle Back-ups sind immer auf demselben Stand und erhalten auf in GIT eine Versionierung. Durch das mehrfache Absichern der Dokumentation und der Applikation kann jederzeit auf ein altes Backup zurückgegriffen werden. Falls ein Harddiskfeher auftritt, kann ohne grössere Probleme an einem anderen Arbeitsplatz weiter gearbeitet werden. Um die Entwicklung zu vereinfachen, wird XAMPP benutzt.

#### Schadensminimierung

Um den Schaden zu minimieren wird mindestens einmal am Tag ein komplettes Back-up gemacht. Bei grösseren Änderungen an Dokumentation oder Programmcode kann ebenfalls ein neues Backup erstellt werden, damit der Datenverlust möglichst gering gehalten werden kann.

### Zeitprobleme

Bei einer Arbeit von knapp 10 Arbeitstagen und einer knapp bemessenen Zeit sind Zeitprobleme das grössten Problem, die während der Arbeit auftreten können. Falls unvorhergesehene Tätigkeiten oder Fehler aufkommen, führt das zu einer Veränderung des Zeitplanes und somit zur Zeitverzögerung des Projekts.

#### Wahrscheinlichkeit

Die Wahrscheinlichkeit für Zeitprobleme sind sehr hoch. Es gibt viele Fehler die bei der Zeitplanung gemacht werden können. So können für verschiedene Punkte zu viel Zeit gebraucht werden oder die bemessene Zeit für das Projekt falsch eingeteilt sein.

#### Folgen

Zeitprobleme können vom nicht korrigierten Text bis zur nicht Fertigstellung des Projektes führen. Das verpassen des Abgabetermins ist dabei auch ein Problem das auftreten kann.

#### Gegenmassnahmen

Um genügend Zeit für alles zu haben, wurde ein Zeitbuffer eingeplant. An diesem können letzte anstehende Arbeiten erledigt werden, falls diese existieren. Jedoch würde ich lieber ein nicht fertiges Produkt abgeben als den Abgabetermin zu verpassen.

#### Schadensminimierung

Falls schon vor dem Buffertag bemerkt wird, das bestimmte arbeiten mehr Zeit benötigen, kann noch etwas dagegen getan werden. Wie zum Beispiel den Fokus auf etwas anderes setzen oder im Notfallmehr arbeiten. Falls erst im Zeitbuffer Probleme auftreten, kann das fatale Folgen haben, da der Abgabetermin am selben Tag ist.

### Internetbrowser-Probleme

Die geplante Applikation wird für die meistgebrauchten Browser entwickelt. Jeder dieser Browser hat seine Eigenheiten und kann zum Beispiel in der Darstellung des geplanten GUI variieren. Diese manuell anzupassen kann viel Zeit kosten.

#### Wahrscheinlichkeit

Browserprobleme werden vor allem bei der Darstellung des GUI auftreten können. Die Wahrscheinlichkeit, dass etwas nicht wie gewünscht funktioniert oder aussieht, ist bei der Unterstützung von mehreren Browsern hoch.

#### Folgen

Bei Browserproblemen wird es zu Zeitverzögerungen kommen, da Browserspezifische Anpassungen implementiert werden müssen.

#### Gegenmassnahmen

Um den Problem entgegenzukommen wird das Framework Bootstrap benutzt. Dieses hat bereits viele Probleme der unterschiedlichen Browser gelöst. Somit kann der Fokus auf etwas anderes als das Styling der Webseite gelegt werden.

#### Schadensminimierung

Falls dennoch Probleme auftreten muss der Fehler ausgewertet werden. Falls der Fehler nur optisch ist und nicht die Funktionalität der Applikation einschränkt, kann dieser Fehler in der Bufferzeit oder einer nächsten Version der Applikation bearbeitet werden. Falls das Problem die Funktionalität einschränkt, muss schnell gehandelt werden und das Problem manuell mit Hilfe von Dritter und dem Internet gelöst werden.

## Versionierung

Für die Versionierung des GIT-Repositorys wurde ein klassisches System mit Hauptversionsnummer, Nebenversionsnummer, Revisionsnummer und Buildnummer mit ergänzender Informationen zum Entwicklungsstadium der Software gewählt. Die Versionsnummern beziehen sich auf die Software und nicht auf die Dokumentation. Diese enthält eine eigene Versionsnummer unabhängig von dem des Repositorys. Das Repository startet nur mit Revisions- und Buildnummer, da am Anfang nur die Dokumentation existiert. Mit der Implementation und Umsetzung der Software wird diese hinzugefügt. Während dem Entwickeln erhält die Versionsnummer den Zusatz „b“ am Schluss, um auf den Beta-Status hinzuweisen. Die finale Version wird mit einem „rc“ für den Veröffentlichungskandidat gekennzeichnet. Für das Arbeiten mit dem Repository wird die Software „SourceTree“ verwendet.

Die Versionsnummer des Repositorys setzt sich folgendermassen zusammen:

0.0.0-0000x

│ │ │ │ └─── Stadium

│ │ │ └────── Buildnummer

│ │ └───────── Revisionsnummer

│ └─────────── Nebenversionsnummer

└───────────── Hauptversionsnummer

Abbildung: Quelle Wikipedia, erweitert mit dem Softwarestadium

### Backup

Durch die Versionierung mit GIT wird das Zurückspringen in der Entwicklung oder das Wiederherstellen einer vorherigen Version vereinfacht. Um die Sicherheit noch zu verstärken ist das Repository auf GitHub erreichbar. Das öffentliche Repository kann jederzeit wieder lokal hergestellt werden. Zusätzlich sind die Releases in GitHub übersichtlich dargestellt. Ebenso werden alle Dateien zur Sicherheit auf einem USB-Stick gespeichert. Das Backup wird mindestens einmal am Ende des Tages erstellt.

### Commits

Die Commits haben jeweils einen Titel. Diesem folgen die Änderungen im Repository auf separaten Linien. Für jede Meldung wird eine separate Linie beansprucht. Diese wird nicht mit einem Satzzeichen beendet. Die Meldungen sind im Imperativ geschrieben und zeigen die grössten Änderungen auf. Falls Fehler entfernt wurden, können diese unter den Änderungen aufgeführt werden.

## Wieso Bootstrap

### Was ist Bootstrag

Bootstrap ist das meist genutzte Front-End Framework für Webentwickler, welches optimal für responsive, Mobile-first Webseiten genutzt werden kann. Viele Webseiten heutzutage benutzen dieses Framework. Die Meinungen zum Framework sind von Person zu Person sehr verschieden. Die Nutzung des Frameworks bringt viele Vor- und Nachteile mit sich, weshalb Entwickler komplett andere Meinungen dazu haben können

Um die Entscheidung Bootstrap zu benutzen besser aufzuzeigen, sind hier die grössten Vor- und Nachteile von Bootstrap:

### Vorteile

Bootstrap

* erleichtert und beschleunigt die Entwicklung
* erlaubt schnelles Prototyping.
* funktioniert in allen modernen Browsern
* unterstützt JavaScript und jQuery
* arbeitet mit eigener Gitter-Struktur
* ist Mobile-First
* löst viele CSS Probleme von Anfang an
* benötigt nicht viel custom CSS
* ist Open Source

### Nachteile

* Durch das vordefinierte CSS des Frameworks, sehen viele Webseiten durch kaum eigenes Styling gleich aus.
* Bootstrap legt den Schwerpunkt nicht auf das Design, sondern auf die Entwicklung.
* Es ist nicht einfach, der Webseite mit CSS ein anderes Design zu geben.
* Die Bootstrap Struktur bzw. Logik ist überladen.
* Bootstrap zu aktualisieren ist sehr mühsam, da sich immer sehr viel der Logik verändert. Oft wird die Version von Bootstrap, mit der die Applikation erstellt wurde die finale Bootstrap-Version.

### Wieso Bootstrap benutz wird

Bootstrap 3 bietet mit dem bereits vorhandenen CSS und dem Gitter-System eine vorhandene Struktur, mit welcher mehr einfach arbeiten kann. Da Bootstrap sehr gut Dokumentiert ist, ist es ein leichtes, die Webseite innerhalb des Frameworks so auszusehen zu lassen wie man möchte, ohne viel eigenes CSS zu schreiben. Das spart bei der Entwicklung sehr viel Arbeit und vor allem Zeit. Das Bootstrap Webseitendesign wird mittlerweile von vielen Webseiten benutz. Deshalb sind viele Nutzerinnen und Nutzer bereits an die Darstellungsweise einer Bootstrap-Seite gewöhnt.

## Planung der Implementierung

Das Ticketingsystem besteht aus drei Komponenten.Da diese Applikationsteile aufeinander aufbauen, müssen diese in einer bestimmten Reihenfolge Implementiert werden. Angefangen wird mit der rohen Struktur der Webseite in der Entwurfsphase. Auf dieser wird dann das Loginsystem aufgebaut. Nach der Erstellung des Logins wird die Nutzerverwaltung implementiert. Erst am Schluss wird das Erstellen und Bearbeiten von Tickets ermöglicht, da diese die Benutzerverwaltung und das Login voraussetzen

### Login

Ein Login mit Usernamen und Passwort, welches als Hash (Sha512) hinterlegt ist.

### Benutzerverwaltung

Es werden zwei Benutzergruppen erstellt: Benutzer und Administratoren.

Benutzer können neue Tickets erstellen und ihre persönlichen Informationen bearbeiten. Diese bestehen aus E-Mail, Telefonnummer und Passwort, sowie Vor- und Nachnamen. Administratoren können zusätzlich zu den Benutzern folgende Aktivitäten vornehmen: Jedem Ticket einen Administrator zuweisen, alle Tickets bearbeiten, den Status des Tickets verändern, neue Nutzer erstellen und deren persönlichen Informationen bearbeiten.

### Die Verwaltung von Tickets

Auf der Ticketübersicht sind alle aktiven Tickets publiziert. Die Ticket lassen sich nach den folgenden Kriterien filtern: Kategorie, Ersteller, Supporter und Status.

Die Tickets werden in einer Tabelle mit Angaben zum Titel, Kategorie, Erfasser, Supporter, Status und Deadline angezeigt. Das Erstellungsdatum wird nicht in der Ticketliste angezeigt, um die Übersicht zu erleichtern.Tickets können in einer Detailansicht geöffnet werden. In dieser Ansicht ist die genaue Beschreibung des Problems ersichtlich. Die Administratoren können in dieser Ansicht die Ticketinformationen verändern.

### Ticket Workflow

Ein Ticket wird mittels eines Formulars erfasst. Im Formular können die Ticketinformationen (bestehend aus dem Tickettitel, der Kategorie, der Ticketbeschreibung und einer allfälligen Deadline) angegeben werden. Zudem kann eine Datei z.B. ein Printscreen hochgeladen werden. Nach dem Abschicken des Tickets erhaltet der die Administratoren eine E-Mail-Benachrichtigung. Da Ticket wird dann manuell einem Administrator zugewiesen. Nach der Zuweisung erhält der Erfasser des Tickets eine E-Mail mit der Meldung, dem Ticket sei ein Administrator zugewiesen worden. Der Erfasser des Tickets erhält ebenfalls eine E-Mail, wenn der Status des Tickets verändert wird. Mit dem Status „Archiv“ wird das Ticket aus der Standardansicht, die alle offenen Tickets anzeigt, ausgeblendet.

# Design

## Datenbank

### Vor der Implementierung

Vor der Implementierung wurde eine Datenbank erstellt, welche die Applikation für das Speichern von Tickets und Benutzer benötigt.

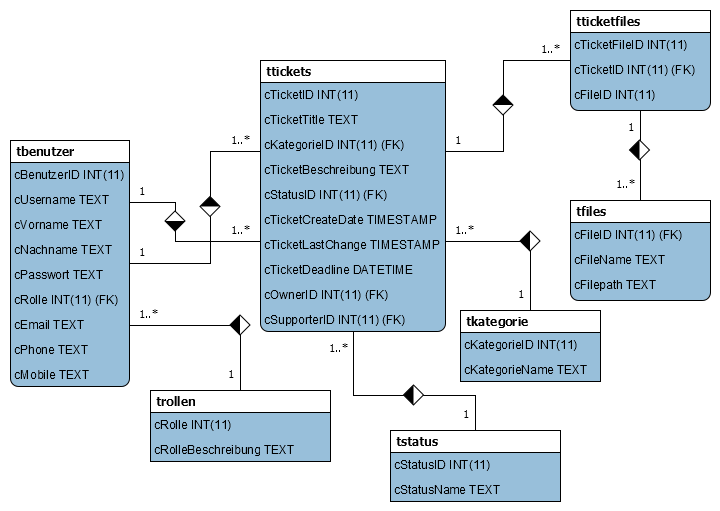


Abbildung: Datenbank vor der Erstellung der Applikation

Die Datenbank besteht aus sieben Tabellen. Die Tabelle „tbenutzer“ inklusive „trollen“ wird für die Identifikation der Nutzerinnen und Nutzer verwendet. Die Tabelle „ttickets“ enthält alle gespeicherten Tickets mit ihrem Status, welcher aus der Tabelle „tstatus“ genommen wird, der Kategorie, welche aus der Tabelle „tkategorien“ genommen wird, und die hochgeladen Dateien, welche über den Pfad in der Tabelle „tfiles“ referenziert werden. Diese gespeicherten Files sind durch die Verbindungstabelle „tticketfiles“ mit der „tticket“ Tabelle verbunden.

Um die Datenbank übersichtlich zu machen, wurde vorerst für jeden Eintrag in der Datenbank mit Text den Feldtyp „Text“ verwendet. Alle Zahlen haben den Feldtyp „int(11)“ erhalten. Das Datum des Tickets wir mittels dem Feldtyp „TIMESTAMP“ und „DATETIME“ gespeichert.

### Nach der Erstellung der Applikation

Die Datenbank hat sich während der Entwicklung der Applikation verändert. Die meisten „Text“-Feldtypen wurden zu „vachar“ umgeändert, da der„Text“-Feldtyp eigentlich für Text und nicht für einen Vorname oder Nachnamen angedacht ist. Zudem ist der Typ „varchar“ perfomanter.

Neben den Änderungen in den Feldtypen hat die Benutzertabelle ein weiteres Feld erhalten. Das Feld „cAktiv“ wurde hinzugefügt, um Benutzer ausblenden zu können. Dieses Feature wurde nach dem ersten Gespräch mit den Experten implementiert. Es wird benötigt, damit man erstellte Benutzer nicht löschen muss. Ein deaktivierter user kann sich nicht mehr einloggen.

Die veränderte Datenbank hat die folgende Struktur:

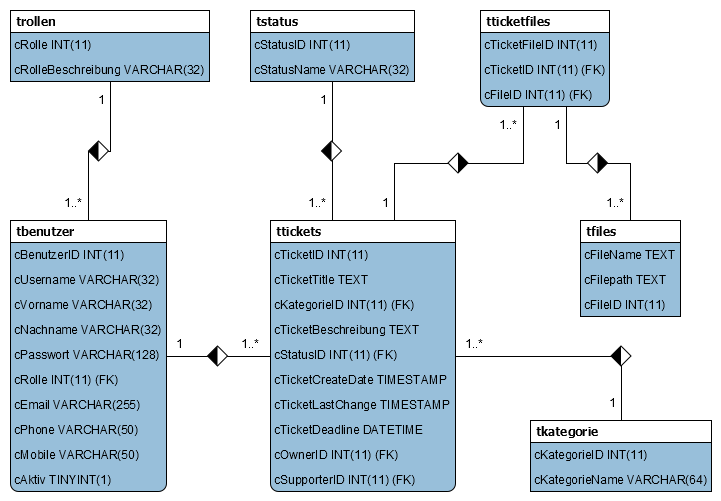


Abbildung: Neue Datenbankstruktur

Um die Darstellung zu vereinfachen, wurde die zweite Verbindung von der „ttickets“ Tabelle zu der „tbenutzer“ Tabelle ausgeblendet.

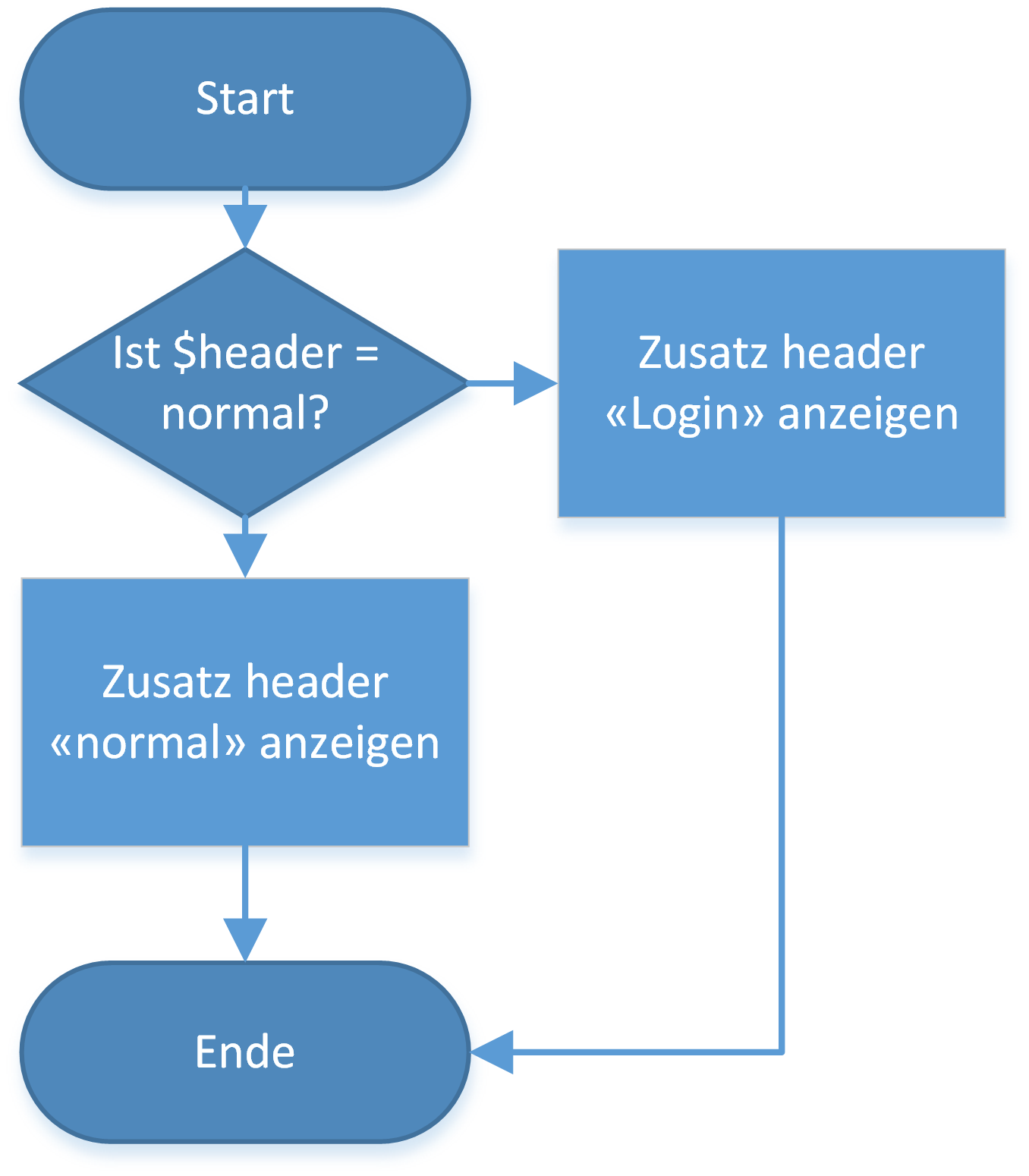
# Implementierung

## Standard GUI

Zuerst wurde das Grundgerüst der Webseite erstellt. Dieses besteht aus dem Header der Webseite, der Webseiten-Navigation und dem Footer.

### header.inc.php

Die Headerdatei startet die Html-Struktur mit dem Webseiten Icon sowie den links zu den CSS und JS Dateien, die für alle Seiten verfügbar sein müssen. Der Restliche Header wird durch die Funktion getHeader() angefügt.



#### getHeader()

Diese Funktion determiniert aufgrund einer Variable, welche mit jeder Seite mitgegeben wird, welcher zusätzliche Header geladen werden müssen. Da nicht jede Seite jede Header benötigt, wurde dies so gelöst.

Im Header wird das Bootstrap 3 Framework geladen, sowohl als auch das Bootstrap 3 Plugin „Datepicker“ von Eonasdan welches ich für die eingabe der Deadline benutze.

Abbildung: getHeader

### navigation.inc.php

Die Webseitennavigation gibt es in drei Varianten.

1. Ein eingeloggter Benutzer erhält die Standardnavigation mit der Ticketübersicht, der Funktion „Ticket erfassen“, dem Benutzerhandbuch, dem Nutzerprofil und dem Log-Out.
2. Ein Administrator erhält eine Navigation mit zusätzlichem Funktionen. Diese bestehen aus dem Usermanager und der Funktion „User Erfassen“
3. Alle Seiten die etwas mit Login oder Berechtigungen zu tun haben, erhalten die dritte Navigation. Diese beinhaltet nur den Link zu dem Benutzerhandbuch. Die Navigation wird durch eine Variabel bestimmt, die in der PHP-Datei checkauth.inc.php festgelegt wird.

### footer.inc.php

Der Footer beinhaltet den Schluss der HTML Struktur mit einer Copyright und Datumsangabe.

## Autorisierungs-Seiten

Die Ticketingapplikation beinhaltet zwei Authoritäts. Einen für den Zugang zum Ticketsystem selbst, sowohl einen für die Administratorenberechtigungen.

### checkauth.inc.php

Diese Prüfung wird auf jeder Ticketinginternen Seite durchgeführt. Es wird geprüft, ob der User, der auf die Ticketingapplikation zugreiffen möchte, eingeloggt ist. Falls dies nicht der Fall ist, wird diesem ein Hinweis auf das Einloggen angezeigt, die eingegebene URL kopiert und auf die Loginseite weitergeleitet. Die Kopierte URL wird zudem In die Login-URL kopiert, um nach dem erfolgreichen einloggen des Benutzers automatisch die vorher aufgerufene Seite anzuzeigen.

### C:\Users\KellerS\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checkrole.pngcheckrole.inc.php

Die checkrole.inc.php prüft die Session des eingeloggten Benutzers ab. Ist der User kein Administrator bzw. Supporter, wird Ihm der Zugriff auf die Administratorsfunktionen verweigert. Der Benutzer wird auf die zu niedrigen Berechtigungen aufmerksam gemacht, und zu der Ticketübersicht weitergeleitet.

**if**($\_SESSION['user\_role']!= 1)

{

$navbar = 3;

**include** ('include/navigation.inc.php');

**echo** '<div class="container">';

header('Refresh: 2; URL = index.php');

**die**('<h2>Keine Berechtigung für diese Seite

</h2> Bitte <a href="index.php">melden</a>

Sie sich an.');

**echo** '</div>';

}

Abbildung und Code-Ausschnitt der Seite checkrole.inc.php

## Loginseiten

Die Ticketapplikation verfügt über eine Log-In und eine Log-Out Seite.

### login.php

Die Loginseite prüft nach der Eingabe von Benutzernamen und Passwort die Richtigkeit der Daten. Fall Benutzernamen oder Passwort falsch ist, wird eine Fehlermeldung ausgegeben. Ebenfalls wird diese ausgegeben, wenn der der Benutzer, der sich einloggen Will, deaktiviert ist.

### logout.php

Die Logout-Seite ist für das Abmelden des eingeloggten Benutzers. Beim Aufruf der Seite wird die Funktion logOut() aufgerufen. Diese befindet sich im Functioncontroller.

#### logOut()

Die Logout Funktion beendet die aktive Session des Nutzers und löscht das vom Browser erstellte Cookie. Danach wird der Benutzer auf die Loginseite weitergeleitet.

**function** logOut()

{

session\_start();

session\_unset();

session\_destroy();

session\_write\_close();

setcookie(session\_name(),'',0,'/');

**echo** '<div class="container">';

**echo** '<h2>Logout</h2>';

**echo** '<p>Sie werden ausgeloggt</p>';

**echo** '</div>';

header('Refresh: 1; URL = login.php');

}

Abbildung: Codes der Funktion logOut();

## Datenbank Verbindungsseiten

Die Applikation enthält zwei verschiedene Datenanbindungen. Eine über MySQLi, und eine über PDO. Da das Projekt ursprünglich komplett Prozedural implementiert werden sollte, wurde der erste Teil in MySQLi umgesetzt. Nach dem ersten Gespräch mit den Experten wurde entschieden, für den Teil der Applikation welche noch nicht implementiert ist, PDO zu verwenden, da dies die Implementation sehr stark erleichtert und beschleunigt.

### dbsettings.inc.php

Enthält die MySQLi Datenbankinformationen, die für die MySQLi Verbindung benötigt werden

### dbconnection.inc.php

Hier wird die MySQLi Datenbankverbindung hergestellt. Falls diese nicht zustanden kommen kann, wird eine gut sichtbare Fehlermeldung und die Login-Navigation angezeigt.

### pdosettings.inc.php

Enthält die PDO Datenbankinformationen, die für die PDO Verbindung benötigt werden

### pdoclass.inc.php

Die pdoclass.inc.php beinhaltet eine von John Richardson erstellte PDO-Klasse, die die Handhabung der PDO-Verbindung einiges erleichtert. Es wurde das Error-Handling, Umgeschrieben, die Formatierung geändert und die Kommentare an die weiteren Codeteilen angepasst.

**class** dbConn

{

…

**catch** (PDOException $e) {

/\*\* Datebase connection error \*/

$navbar = 3;

**echo** '<div class="alert alert-danger"><strong>Fehler:</strong> Es kann keine Verbindung zur Datenbank hergestellt werden.</div>';

**die**;

}

…

}

Abbildung: Angepasster Code.

## Funktionsseiten

Um Funktionen einfacher im Code zu finden, wurden diese in separate PHP-Files verlagert. Es existieren zwei Funktionsfiles.

### functioncontroller.inc.php

Hier befinden sich fast alle Funktionen, die mehrmals in der Applikation verwendet werden. Es handeln sich dabei um Funktionen, die benötigt werden, um Daten aus der Datenbanken zu laden, oder Benutzer und Tickets zu Validieren. Eine Ausnahme bildet die E-Mail-Versandsfunktion.

### sendmail.inc.php

In dieser PHP Datei wird der E-Mail-versand abgewickelt. Dies wir über die Funktion sendMail() gemacht.

#### sendMail()

Die E-Mail-Funktion ist für den Versand der E-Mails in der gesamten Applikation verantwortlich. Der Funktion werden als Parameter die Ticket-Informationen sowie der Typ des E-Mails und den Empfänger mitgegeben. Jedes E-Mail erhält Standartinformationen, wie den Titel des E-Mails. Aufgrund des E-Mailtyp-Parameters wird dann die Art des E-Mails definiert. Diese bestimmt dann den weiteren Inhalt des Emails und schreibt diese in variablen.

Das ganze E-Mail wird am Schluss in einer Variabel zusammengeführt, und versendet.

/\*\*

\* build the whole email from variables

\*/

$nachricht = '

<html>

'.$mailHead.'

<body>

'.$mailtTitel.'

'.$mailText.'

'.$mailLink.'

<table>

'.$mailTableTitle.'

'.$mailTableDefault.'

'.$mailTable.'

'.$mailTableAction.'

</table>

'.$mailFooter.'

</body>

</html>';

Abbildung: Code-Ausschnitt aus der sendMail() funktion. Zeigt die zusammengestellte Variable.

Über den Link „Zum Ticket“ kann der Administrator oder Benutzer mit wenigen Klicks zum Ticket.

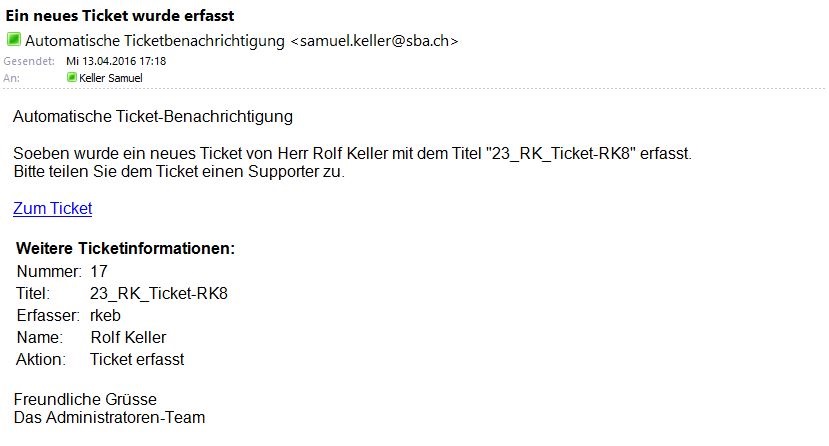


Abbildung: Ein E-Mail, das während der Qualitätssicherung versendet wurde.

## Ticketverwaltungsseiten

### index.php

Die Indexseite ist das Herzstück des Ticketingsystems. Hier werden Tickets über den Ticketfilter gefiltert und über die Ticketliste angezeigt.

#### ticketfilter.inc.php

Mit dem Ticketfilter kann die Ticketübersicht nach vier Kriterien sortiert werden.

Um die Tickets filtern zu können, wird jede Filteroptionen in eine Dropdownbox geladen. Die einzelnen Optionen werden dazu aus der Datenbank geladen. Die Ticketliste greift bei einem Post-Back auf die Variablen des Filters zu, in welche die Optionen für die Sortierung stehen.

Die Optionen „Alle“, und „Alle Aktiven“ stellen dabei eine grosse Ausnahme dar. Um wirklich alle der Tickets zu bekommen, wird anstatt einer Zahl „%“ in das SQL Script geschrieben. Nach dem Zeichen „%“ darf jedes andere Zeichen folgen. Dies führt dazu, dass alle Tickets zurückgegeben werden. Die Sonderreglung bei dem Status der Tickets war schwerer zu lösen. Die Angabe „Alle Aktiven“ schliesst nur die Gelöschten, Geschlossenen und Archivierten Tickets aus der Sortierung aus. Mit der Hilfe des Users Sammaye, konnte ich ein PDO IN()-Statment erstellen, welches ich mit einem Array befüllen konnte. So kann ich genau sagen, welche der Ticketstati ich bei der Option „Alle Aktiven“ (Tickets) aufrufen möchte.

Mit diesem Code werden die Werte für die Kriterien „Alle“ und „Alle Aktiven“ angepasst.

/\*\*

\* define values for filtering

\* % used for sql

\*/

**if** ($fTicketKategorie == 0)

{

$fTicketKategorie = "%";

}

**if** ($fTicketStatus == 0)

{

$fTicketStatus = **array**(1,2,3,4,6);

}

**else**

{

$fTicketStatus = **array**($fTicketStatus);

}

**if** ($fTicketOwner == 0)

{

$fTicketOwner = "%";

}

**if** ($fTicketSupporter == -1)

{

$fTicketSupporter = "%";

}

Abbildung: Codeausschnitt aus der Ticketfilter

Dieser Codeteil zeigt das IN()-Statement, das mit PDO nicht einfach zu lösen ist.

...

WHERE ttickets.cKategorieID LIKE :kategorie &&

ttickets.cStatusID IN ( '.implode(",",$fTicketStatus).') &&

ttickets.cOwnerID LIKE :owner &&

ttickets.cSupporterID LIKE :support

ORDER BY cTicketID Desc');

$stmt->bindParam(':kategorie', $fTicketKategorie, PDO::*PARAM\_STR*);

$stmt->bindParam(':owner', $fTicketOwner, PDO::*PARAM\_STR*);

$stmt->bindParam(':support', $fTicketSupporter, PDO::*PARAM\_STR*);

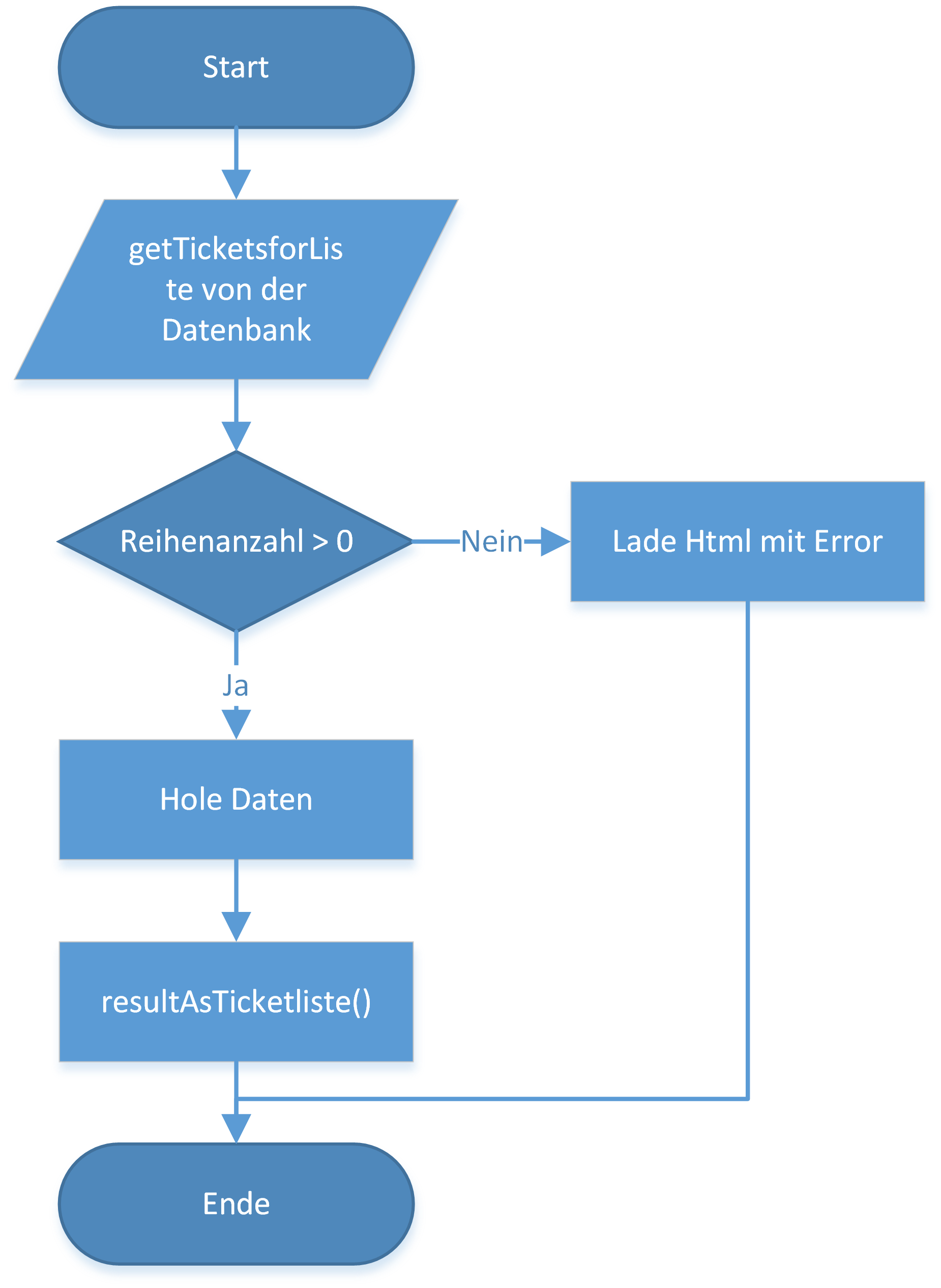
$stmt->execute();

…

Abbildung: Codeausschnitt aus der getTicketforListe() funktion

### ticketliste.inc.php

Die Ticketliste ladet alle Ticketdatensätze mit den im Ticketfilter ausgewählten Kriterien mit der Funktion getTicketsforListe() aus der Datenbank. Wenn ein oder mehrere Ticket unter den Voraussetzungen existieren, werden die aus der Datenbank erhaltenen Tickets mit der Funktion resultAsTicketliste() aufgelistet.



Falls kein Ticket unter den ausgewählten Kriterien existiert, wird ein Hinweis in die Liste geladen, der auf das Fehlen eines Tickets unter diesen Kriterien hinweist.

Abbildung: Darstellungsprozess der Ticketliste

### neuesticket.php

### ticket\_details.php

## Userverwaltung

### profil.php

Im Profil kann ein eingeloggte Benutzer seine persönlichen Daten, bestehend aus Vorname, Nachnamen, E-Mail-Adresse, Telefon- und Handynummer sowie das Passwort verändern.

Alle Eingabefelder werden über die Funktion validateUser() validiert. Die Validierung ist so aufgebaut, das kein Username und keine E-Mail-Adresse zweimal verwendet werden darf. Der Vorname und Nachname darf nicht länger als 32 Zeichen sein und die E-Mail muss korrekt sein. Die Telefonnummern werden mit 13 Zahlen angegeben. Mit dem Speichern des Profils wird jede Eingabe nacheinander geprüft. Falls Die eingegebenen Daten nicht den Kriterien entsprechen, entstehen Fehlermeldungen. Alle Fehlermeldungen werden in eine Variable gesammelt und zusammen ausgegeben. Nur wenn die Validierung korrekt durchgeführt worden ist, wird das Profil aktualisiert.

/\*\*

\* checks is email is already used

\*/

$stmt = mysqli\_stmt\_init($conn);

mysqli\_stmt\_prepare($stmt, "SELECT \* FROM tbenutzer WHERE cEmail=? && cBenutzerID!=?");

mysqli\_stmt\_bind\_param($stmt, "si", $userEmail, $userid);

mysqli\_stmt\_execute($stmt);

mysqli\_stmt\_store\_result($stmt);

**if** (mysqli\_stmt\_num\_rows($stmt) > 0) {

$validationError .= "· Die angegebene E-mail wird bereits verwendet. Bitte geben Sie eine andere an.</br>";

$error = 1;

### }

Abbildung: Codeausschnitt aus der Validierung der E-Mail-Adresse

/\*\* mobile \*/

**if** (!**empty**($userMobile))

{

**if**(!preg\_match("/^[0-9]{13}$/", $userMobile))

{

$validationError .= "· Geben Sie eine gültige Mobiltelefon-Nummer ein. Bsp. 0041000000000</br>";

$error = 1;

}

}

Abbildung: Codeausschnitt aus der Validierung der Handynummer

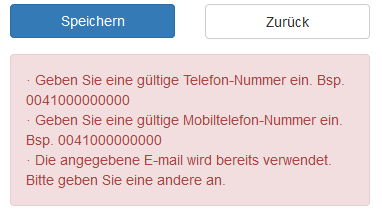


Abbildung: Eine Fehlermeldung, ausgelöst durch die Eingabe einer bereits verwendeten E-Mail-Adresse und flascher Telefon- und Handynummer.

### usermanagement.php

Auf die Userverwaltung haben nur Administratoren Zugriff. Hier lassen sich neue User erstellen, verändern und deaktivieren.

Um eine Übersicht über die editierbaren Benutzer zu haben, werden alle aktiven Benutzer in einer Liste angezeigt. Die Benutzerliste ist ähnlich wie die Ticketliste, kann aber nicht gefiltert werden.

Mit einem Klick auf „User erfassen“ wird man zu der Eingabeform für einen neuen Nutzer geführt.

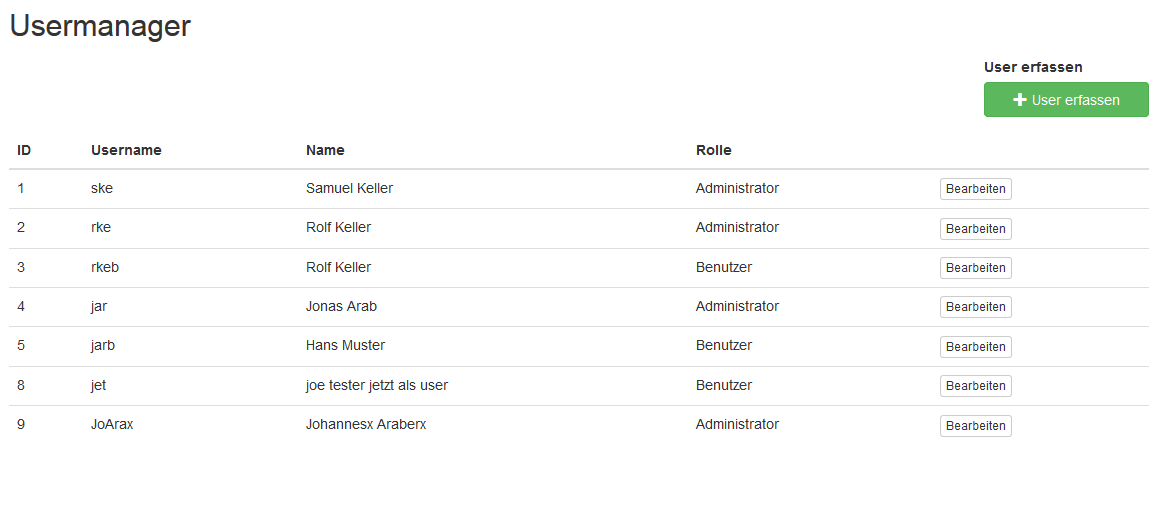


Abbildung: Benutzerliste im usermanager

neuernutzer.php

#### benutzer\_details.php

# Testing

Das Testen der Applikation wurde durch Rolf Keller und Jonas Arab durchgeführt. Die Funktionen der Applikation sind auf modernen Windowsumgebungen mit den neusten Versionen der Browser Google Chrome und Mozilla FireFox mit 28 Testfällen ausführlich getestet worden.

## Endresultat

Wie auf den Testprotokollen zu sehen ist , gab es keine grösseren Probleme mit der Applikation. Einzig der Testfall mit der Nummer 24 konnte teilweise nicht wie erwartet abgeschlossen werden. Beim erstellen eines Tickets mit zDieser Fehler kann in einer Version 2.0 oder einem Patch behoben werden

## Testfälle

|  |  |
| --- | --- |
| #01 | Login mit falschen Daten |
| Beschreibung | Bei der Eingabe eines falschen Passwortes und/oder Usernamens wird der Benutzer nicht eingeloggt |
| Test | Login mit falschen Logindaten |
| Erwartetes Resultat | Fehlermeldung: Falscher Benutzername oder falsches Passwort |

|  |  |
| --- | --- |
| #02 | Zugriff auf die Applikation ohne Login |
| Beschreibung | Beim Zugriff auf eine Seite, auf die nur mit Login zugegriffen werden kann, wird der User auf das Nicht-eingeloggt-sein hingewiesen und zum Login weitergeleitet. |
| Test | Zugriff auf ipa.sbvg.eu/index.php |
| Erwartetes Resultat | Weiterleitung auf das Login |

|  |  |
| --- | --- |
| #03 | Login mit richtigen Daten |
| Beschreibung | Bei der Eingabe eines richtigen Usernamens mit Passwort wird der Benutzer auf die Ticketübersicht weitergeleitet. |
| Test | Login mit richtigen Logindaten |
| Erwartetes Resultat | Weiterleitung auf die Ticketübersicht |

|  |  |
| --- | --- |
| #04 | Login eines deaktivierten Users |
| Beschreibung | Bei der Eingabe von Logindaten eines deaktivierten Benutzers wird der Benutzer nicht eingeloggt |
| Test | Login mit deaktiviertem User |
| Erwartetes Resultat | Fehlermeldung: Falscher Benutzername oder falsches Passwort |

|  |  |
| --- | --- |
| #05 | Login mit übergebener URL |
| Beschreibung | Beim Zugriff auf eine Seite, auf die nur mit Login zugegriffen werden kann, wird der User auf das Nicht-eingeloggt sein hingewiesen und zum Login weitergeleitet. Die URL der vorherigen Seite wird dem Login übergeben. |
| Test | Zugriff auf http://ipa.sbvg.eu/neuesticket.php  Login mit richtigen Daten |
| Erwartetes Resultat | Weiterleitung zur Ticket-Eingabeform Eingabeformular? |

|  |  |
| --- | --- |
| #06 | Anzeigen der Admin-Navigation |
| Beschreibung | Dem Administrator wird eine Webseiten-Navigation mit dem Usermanager angezeigt. |
| Test | Login mit einem Administrator |
| Erwartetes Resultat | Nach erfolgreichem Login wird dem Administrator eine Navigation mit den Optionen Ticketübersicht, Ticket erfassen, Hilfe, Profil, Usermanager mit der Funktionen „Neuer Nutzer“ und „Übersicht“ sowie Log-out angezeigt. |

|  |  |
| --- | --- |
| #07 | Anzeigen der Benutzer-Navigation |
| Beschreibung | Dem Benutzer wird eine Webseiten-Navigation ohne Usermanager angezeigt. |
| Test | Login mit einem Benutzer |
| Erwartetes Resultat | Nach erfolgreichem Login wird dem Benutzer eine Navigation mit den Optionen Ticketübersicht, Ticket erfassen, Hilfe, Profil sowie Log-out angezeigt. |

|  |  |
| --- | --- |
| #08 | Korrekte Anzeige der Ticketübersicht |
| Beschreibung | Die Ticketübersicht wird mit Ticketfilter und Ticketliste angezeigt. |
| Test | Aufruf der Ticketübersicht |
| Erwartetes Resultat | Die Ticketübersicht zeigt alle aktuellen offenen Tickets an. |

|  |  |
| --- | --- |
| #09 | Ticketfilterung |
| Beschreibung | Die Tickets lassen Sich erfolgreich nach einzelnen oder mehreren Kriterien filtern. |
| Test | Auswahl der Filterkriterien in den Dropdownfeldern.  Bestätigung der Filterung durch betätigen des Buttons „Filtern“. |
| Erwartetes Resultat | Die Ticketliste wird nach den ausgewählten Kriterien gefiltert. Falls keine Tickets mit den Kriterien existieren, wird ein entsprechender Hinweis angezeigt. |

|  |  |
| --- | --- |
| #10 | Zugriff auf die Administrator-Funktionen als Administrator |
| Beschreibung | Kann der Administrator die Administrator-spezifischen Funktionen „Neuer Nutzer“ und “Übersicht“ zugreifen? |
| Test | Login als Administrator  Zugriff auf die Funktionen des Usermanagers durch die Navigation |
| Erwartetes Resultat | Anzeige der jeweiligen Seiten und Eingabemasken. |

|  |  |
| --- | --- |
| #11 | Zugriff auf die Administrator-Funktionen als Benutzer |
| Beschreibung | Kann der Administrator die Administrator-spezifischen Funktionen „Neuer Nutzer“ und „Nutzerübersicht“ zugreifen? |
| Test | Login als Benutzer  Zugriff auf die Funktionen des Usermanagers durch die manuelle Eingabe der URL:  http://ipa.sbvg.eu/usermanager.php  http://ipa.sbvg.eu/neueenutzer.php |
| Erwartetes Resultat | Hinweis auf die minderen Berechtigungen und Weiterleitung auf die Ticketübersicht |

|  |  |
| --- | --- |
| #12 | Individuelles Benutzerprofil |
| Beschreibung | Jeder Benutzer kann auf sein Individuelles Profil zugreifen. Dieses ist über die Navigation erreichbar. |
| Test | Aufruf des Benutzerprofils durch einen Klick auf den Nutzernamen in der Navigation |
| Erwartetes Resultat | Anzeige des individuellen Benutzerprofils mit Eingabefeldern und Möglichkeit, das Benutzerprofil zu ändern. |

|  |  |
| --- | --- |
| #13 | Benutzerprofil Validierung |
| Beschreibung | Für das Speichern der Profil Daten muss jede Eingabe korrekt validiert werden. Die Eingabemaske gibt einen Fehler aus, wenn:   * der Vorname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Nachname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * keine korrekte oder eine von einem anderen Benutzer verwendete E-Mail-Adresse angegeben wurde * die Telefonnummern nicht dem angezeigten Format entsprechen. |
| Test | Eingabe von falschen Daten in Felder Vorname, Nachname, E-Mail, Telefonnummern.  Profil Speichern. |
| Erwartetes Resultat | Das Profil wird nicht aktualisiert und die Maske gibt Fehlermeldungen für die jeweils falsch angegebenen Daten an. |

|  |  |
| --- | --- |
| #14 | Benutzerprofil aktualisieren |
| Beschreibung | Für das Speichern der Profil Daten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:   * der Vorname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Nachname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * keine korrekte oder eine von einem anderen Benutzer verwendete E-Mail-Adresse angegeben wurde * die Telefonnummern nicht dem angezeigten Format entsprechen. |
| Test | Eingabe von korrekten Daten in Vorname, Nachname, E-Mail, Telefonnummern.  Profil Speichern. |
| Erwartetes Resultat | Das Profil wird aktualisiert. |

|  |  |
| --- | --- |
| #15 | Detailansicht des Tickets als Benutzer |
| Beschreibung | Mit dem Klick auf „Ansehen“ in der Ticketliste wird das jeweilige Ticket in der Detailansicht geöffnet. |
| Test | „Ansehen“ bei einem beliebigen Ticket klicken |
| Erwartetes Resultat | Dier Detailansicht der Tickets zeigt die ID, den Titel, das Thema, die Beschreibung, den Erfasser, den Supporter, das Erstellungsdatum, den Endtermin, das Last-Change-Datum, den Status sowie die Datei an, wenn vorhanden. |

|  |  |
| --- | --- |
| #16 | Detailansicht des Tickets als Administrator |
| Beschreibung | Mit dem Klick auf „Ansehen“ in der Ticketlist wird das jeweilige Ticket in der Detailansicht geöffnet. |
| Test | „Ansehen“ bei einem beliebigen Ticket klicken |
| Erwartetes Resultat | Die Detailansicht der Tickets zeigt die ID, den Titel, das Thema, die Beschreibung, den Erfasser, den Supporter, das Erstellungsdatum, den Endtermin, das Last-Change-Datum, den Status sowie die Datei an, wenn vorhanden. Die in Dropdown und Eingabefeldern angezeigten Daten können verändert werden. |

|  |  |
| --- | --- |
| #17 | Verändern des Tickets in der Detailansicht durch den Administrator |
| Beschreibung | Die in Dropdown und Eingabefeldern angezeigten Daten können verändert und gespeichert werden. |
| Test | Manuelle Veränderung der Daten des Tickets  Betätigung des Knopfes „Speichern und Zurück“ |
| Erwartetes Resultat | Die neuen Ticketeingaben werden gespeichert und der Administrator wird auf die Ticketübersicht weitergeleitet |

|  |  |
| --- | --- |
| #18 | Verändern des Ticketstatus in der Detailansicht durch den Administrator |
| Beschreibung | Die in Dropdown und Eingabefeldern angezeigten Daten können verändert und gespeichert werden. |
| Test | Manuelle Veränderung des Ticketstatus  Betätigung des Knopfes „Speichern“ |
| Erwartetes Resultat | Nach dem Speichern des Tickets mit verändertem Ticketstatus wird der Ticketerfasser über diesen neuen Status per E-Mail benachrichtigt. |

|  |  |
| --- | --- |
| #19 | Verändern des Supporters in der Detailansicht durch den Administrator |
| Beschreibung | Die in Dropdown und Eingabefeldern angezeigten Daten können verändert und gespeichert werden. |
| Test | Manuelle Veränderung des Supporters  Betätigung des Knopfes „Speichern“ |
| Erwartetes Resultat | Nach dem Speichern des Tickets mit verändertem Supporter werden der Ticketerfasser und der neue Supporter per E-Mail benachrichtigt. |

|  |  |
| --- | --- |
| #20 | Ticket erfassen |
| Beschreibung | Über den Knopf „Ticket erfassen“ in der Navigation wird der Benutzer oder Supporter zu der Eingabeform für ein neues Ticket weitergeleitet. |
| Test | Betätigen des Knopfes „Ticket erfassen“ in der Navigation. |
| Erwartetes Resultat | Erfolgreiche Anzeige der Eingabemaske. Dieses beinhaltet Eingabefelder für Titel, Thema, Beschreibung, Endtermin und Datei. |

|  |  |
| --- | --- |
| #21 | Ticket Validierung |
| Beschreibung | Nach dem eingeben des neuen Tickets wird dieses vor dem Erstellen validiert. |
| Test | Eingabe eines Testtickets ohne Titel, Beschreibung und Thema  Betätigen des „Speichern“ Knopfes. |
| Erwartetes Resultat | Das Ticket kann nicht ohne diese drei Daten erstellt werden. Das Formular fordert den Ticketerfasser auf, den Titel, die Beschreibung und das Thema einzugeben. |

|  |  |
| --- | --- |
| #22 | Ticket-Deadline Validierung |
| Beschreibung | Dem Ticket kann eine Deadline hinzugefügt werden. |
| Test | Eingabe eines Testtickets mit Titel, Beschreibung und Thema  Zusätzliche Eingabe der Deadline durch die Datum-Auswahlfunktion im Deadline-Eingabefeld.  Betätigen des „Speichern“ Knopfes. |
| Erwartetes Resultat | Das Ticket wird mit der Deadline gespeichert. |

|  |  |
| --- | --- |
| #23 | Ticket-Deadline Validierung |
| Beschreibung | Das Ticket kann nicht mit vergangenem Datum als Deadline erstellt werden |
| Test | Eingabe eines Testtickets mit Titel, Beschreibung und Thema  Zusätzliche Eingabe eines vergangenen Tages durch die Datum-Auswahlfunktion im Deadline-Eingabefeld.  Betätigen des „Speichern“ Knopfes. |
| Erwartetes Resultat | Error mit dem Hinweis, dass das Enddatum nicht vor dem Erstellungsdatum liegen kann. Die eingegeben Daten werden wieder in die Form geladen, so dass nur das Datum neu eingegeben werden muss |

|  |  |
| --- | --- |
| #24 | Datei Validierung |
| Beschreibung | Dem Ticket kann eine Datei mit den Formaten JPG, JPEG, PNG, GIF, PDF und ZIP beigefügt werden. Die Datei darf nicht grösser als 5 MB sein. |
| Test | Eingabe eines Testtickets mit Titel, Beschreibung und Thema  Zusätzliche Angabe einer Datei, welche nicht den erlaubten Formaten angehört und grösser als 5 MB ist. Betätigen des „Speichern“ Knopfes. |
| Erwartetes Resultat | Error mit dem Hinweis, dass die angefügte Datei nicht den erlaubten Formaten entspricht und / oder zu gross ist. |

|  |  |
| --- | --- |
| #25 | Neuer Nutzer erfassen als Administrator |
| Beschreibung | Für das Speichern des neuen Nutzers muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:   * der Vorname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Nachname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Username kürzer als 3 Zeichen oder länger als 32 Zeichen oder bereits vergeben ist * keine korrekte oder eine von einem anderen Benutzer verwendete E-Mail-Adresse angegeben wurde, * die Telefonnummern nicht dem angezeigten Format entsprechen. |
| Test | Eingabe von falschen Daten in Vorname, Nachname, Usernamen, E-Mail, Telefonnummern.  Rolle entweder „Benutzer“ oder „Administrator“.  Eingabe eines Passworts  Profil erfassen. |
| Erwartetes Resultat | Das Profil wird nicht erstellt und die Maske gibt Fehlermeldungen für die jeweils falsch angegebenen Daten an. |

|  |  |
| --- | --- |
| #26 | Veränderung der Nutzerdaten durch den Administrator in der Userdetailübersicht. |
| Beschreibung | Für das aktualisieren der Benutzerdaten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:   * der Vorname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Nachname kürzer als 3 Zeichen oder länger als 32 Zeichen ist, * der Username kürzer als 3 Zeichen oder länger als 32 Zeichen oder bereits vergeben ist, * keine korrekte oder eine von einem anderen Nutzer benutze E-Mail-Adresse angegeben wurde, * die Telefonnummern nicht dem angezeigten Format entsprechen. |
| Test | Mit einem Klick auf den Button „Bearbeiten“ im Usermanager beim zu editierenden User wird dieser in der Detailübersicht geöffnet.  Geben Sie falsche Daten in Vorname, Nachname, Usernamen, E-Mail, Telefonnummern ein.  Rolle entweder „Benutzer“ oder „Administrator“.  Profil speichern. |
| Erwartetes Resultat | Das Profil wird nicht aktualisiert und die Maske gibt Fehlermeldungen für die jeweils falsch angegebenen Daten an. |

|  |  |
| --- | --- |
| #27 | Deaktivieren eines Benutzers durch den Administrator in der Userdetailübersicht. |
| Beschreibung | Die Administratoren können einen Benutzer über die Detailübersicht deaktivieren. |
| Test | Klick auf den Knopf „Deaktivieren“ |
| Erwartetes Resultat | Das Profil wird deaktiviert, der Administrator wird zur Benutzerübersicht weitergeleitet. Der deaktivierte User wird nicht mehr in der Benutzerübersicht angezeigt. |

|  |  |
| --- | --- |
| #28 | Veränderung der Benutzerdaten durch den Administrator in der Benutzerdetailübersicht. |
| Beschreibung | Für das aktualisieren der Benutzerdaten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:   * der Vorname kürzer als 3 Zeichen oder grösser als 32 Zeichen ist, * der Nachname kürzer als 3 Zeichen oder grösser als 32 Zeichen ist, * der Username kürzer als 3 Zeichen oder grösser als 32 Zeichen oder bereits vergeben ist, * keine korrekte oder eine von einem anderen Benutzer verwendete E-Mail-Adresse angegeben wurde, * die Telefonnummern nicht dem angezeigten Format entsprechen. |
| Test | Mit einem Klick auf den Button „Bearbeiten“ im Usermanager beim zu editierenden User wird dieser in der Detailübersicht geöffnet.  Geben Sie korrekte Daten in Vorname, Nachname, Usernamen, E-Mail, Telefonnummern ein.  Rolle entweder „Benutzer“ oder „Administrator“.  Profil speichern. |
| Erwartetes Resultat | Das Profil wird aktualisiert. |

## Testprotokolle

### Testprotokoll #1

Tester: Rolf Keller

**Testmittel:**

Windows 7 Pro

FireFox 45.0.2

|  |  |
| --- | --- |
| Testfall-Nummer (#01) | Login mit falschen Daten |
| Testdaten | rkeller - benutzer |
| Aktionen | Eingabe in Login-Felder |
| Resultat | Fehlermeldung: Falscher Benutzername oder Passwort; Log-In-Felder werden erneut angeboten |

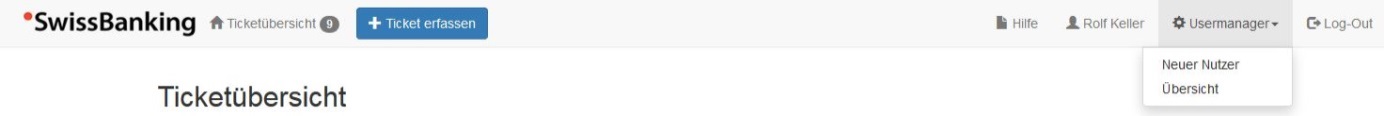
|  |  |
| --- | --- |
| Testfall-Nummer (#02) | Zugriff auf die Applikation ohne Login |
| Testdaten |  |
| Aktionen | Zugang versucht mit http://ipa.sbvg.eu/index.php |
| Resultat | Textmeldung: Sie sind nicht eingeloggt. Bitte melden Sie sich an.  Automatische Anzeige der Login-Felder |

|  |  |
| --- | --- |
| Testfall-Nummer (#03) | Login mit richtigen Daten |
| Testdaten | a ) rke - admin |
| Aktionen | a) Zugangsdaten Admin: Eingabe in Login-Felder  b) Logout  c) Zugangsdaten Benutzer: Eingabe in Login-Felder |
| Resultat | a) Login korrekt als Admin, Anzeige Ticketübersicht  b) Textmeldung: Logout Sie werden ausgeloggt. Automatische Anzeige der Login-Felder  c) Login korrekt als Benutzer, Anzeige Ticketübersicht |

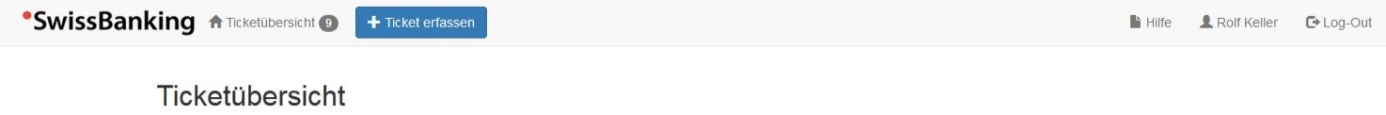
|  |  |
| --- | --- |
| Testfall-Nummer (#04) | Login eines deaktivierten Users |
| Testdaten |  |
| Aktionen | „duser“ deaktiviert |
| Resultat | kein Einloggen mehr möglich |

|  |  |
| --- | --- |
| Testfall-Nummer (#05) | Login mit übergebener URL |
| Testdaten |  |
| Aktionen | Zugang versucht mit http://ipa.sbvg.eu/neuesticket.php  Nach Anzeige der Login-Felder angemeldet als rke - admin |
| Resultat | Textmeldung: Sie sind nicht eingeloggt.  Bitte melden Sie sich an.  Automatische Anzeige der Login-Felder  Eingabeseite für neue Ticket wird sofort angezeigt: (http://ipa.sbvg.eu/neuesticket.php) |

|  |  |
| --- | --- |
| Testfall-Nummer (#06) | Anzeigen der Admin-Navigation |
| Testdaten |  |
| Aktionen | Login als Administrator |
| Resultat | vollständige Admin-Menüleiste wird angezeigt |



|  |  |
| --- | --- |
| Testfall-Nummer (#07) | Anzeigen der Benutzer-Navigation |
| Testdaten |  |
| Aktionen | Login als Benutzer |
| Resultat | vollständige Benutzer-Menüleiste wird angezeigt |



|  |  |
| --- | --- |
| Testfall-Nummer (#08) | Korrekte Anzeige der Ticketübersicht |
| Testdaten |  |
| Aktionen | Klick auf „Ticketübersicht“ |
| Resultat | Ticketübersicht wird angezeigt mit allen offenen Tickets und den Auswahllisten für die Filterung |



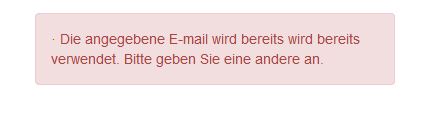
|  |  |
| --- | --- |
| Testfall-Nummer (#09) | Ticketfilterung |
| Testdaten |  |
| Aktionen | Neues Ticket angelegt, um mehrstufige Filterung testen zu können. |
| Resultat | Mehrstufige Filterung funktioniert korrekt. Falls kein passendes Ticket vorhanden, erscheint Textmeldung „Kein Ticket erfüllt die ausgewählten Suchkriterien“. Wunsch: Button zur Rücksetzung aller Filter |

|  |  |
| --- | --- |
| Testfall-Nummer (#10) | Zugriff auf die Administrator-Funktionen als Administrator |
| Testdaten |  |
| Aktionen | a) Usermanager 🡪 Neuer Nutzer  b) Usermanager 🡪 Übersicht |
| Resultat | a) Eingabemaske für neuen User erscheint  b) Übersicht der vorhandenen User erscheint. Über den Button „User erfassen“ kann hier ebenfalls die Eingabemaske „Neuer Nutzer“ aufgerufen werden. |

|  |  |
| --- | --- |
| Testfall-Nummer (#11) | Zugriff auf die Administrator-Funktionen als Benutzer |
| Testdaten |  |
| Aktionen | Da der Usermanager nicht angezeigt wird, ist kein Zugriff auf die Userverwaltung möglich. |
| Resultat |  |

|  |  |
| --- | --- |
| Testfall-Nummer (#12) | Individuelles Benutzerprofil |
| Testdaten |  |
| Aktionen | Klick auf Benutzernamen |
| Resultat | Profil wird geöffnet |

|  |  |
| --- | --- |
| Testfall-Nummer (#13) | Benutzerprofil Validierung |
| Testdaten |  |
| Aktionen | a) Telefonnummern ergänzt  b) E-Mail-Adresse geändert nach Fehlermeldung, Telefonnummern ergänzt |
| Resultat | a) Fehlermeldung (s. Screenshot), dass die E-Mail-Adresse bereits verwendet werde, eingetippte Telefonnummern sind verloren.  b) andere E-Mail-Adresse eingetragen, Telefonnummern wieder eingegeben, nun alles korrekt gespeichert |



|  |  |
| --- | --- |
| Testfall-Nummer (#14) | Benutzerprofil aktualisieren |
| Testdaten |  |
| Aktionen | a) „falsche“ Eingaben  b) korrekte Eingaben |
| Resultat | a) relevante Fehlermeldung erscheint (E-Mail siehe #13)  b) Speicherung der geänderten Daten erfolgt. |

|  |  |
| --- | --- |
| Testfall-Nummer (#15) | Detailansicht des Tickets als Benutzer |
| Testdaten |  |
| Aktionen | diverse Tickets aufgerufen |
| Resultat | werden korrekt angezeigt, ebenfalls angehängte jpg und png |

|  |  |
| --- | --- |
| Testfall-Nummer (#16) | Detailansicht des Tickets als Administrator |
| Testdaten |  |
| Aktionen | diverse Tickets aufgerufen |
| Resultat | Darstellung ist vollständig |

|  |  |
| --- | --- |
| Testfall-Nummer (#17) | Verändern des Tickets in der Detailansicht durch den Administrator |
| Testdaten |  |
| Aktionen | diverse Änderungen am zuvor eingegebenen Ticket vorgenommen |
| Resultat | Änderung / Speicherung funktioniert korrekt |

|  |  |
| --- | --- |
| Testfall-Nummer (#18) | Verändern des Ticketstatus in der Detailansicht durch den Administrator |
| Testdaten |  |
| Aktionen | Status am zuvor eingegebenen Ticket geändert |
| Resultat | ok, gespeichert |

|  |  |
| --- | --- |
| Testfall-Nummer (#19) | Verändern des Supporters in der Detailansicht durch den Administrator |
| Testdaten |  |
| Aktionen | Supporter auf „duser“ geändert |
| Resultat | ok, gespeichert |

|  |  |
| --- | --- |
| Testfall-Nummer (#20) | Ticket erfassen |
| Testdaten |  |
| Aktionen | Ticket erfassen |
| Resultat | Ticket 10 wurde korrekt angelegt (und seither mehrmals geändert) |

|  |  |
| --- | --- |
| Testfall-Nummer (#21) | Ticket Validierung |
| Testdaten |  |
| Aktionen | verschiedene Tickets angelegt / validiert als Admin und Benutzer |
| Resultat | ok, gespeichert |

|  |  |
| --- | --- |
| Testfall-Nummer (#22) | Ticket-Deadline Validierung |
| Testdaten |  |
| Aktionen | Deadline in der Zukunft eingetragen |
| Resultat | wird korrekt gespeichert |

|  |  |
| --- | --- |
| Testfall-Nummer (#23) | Ticket-Deadline Validierung |
| Testdaten |  |
| Aktionen | vergangenes Datum als Deadline eingefügt |
| Resultat | zurückliegendes Datum wird zurückgewiesen.  unangenehm: die Meldung muss neu getippt werden nach der zurückgewiesenen Validierung |

|  |  |
| --- | --- |
| Testfall-Nummer (#24) | Datei Validierung |
| Testdaten |  |
| Aktionen | Anlegen von mehreren Tickets mit angehängter Datei |
| Resultat | Bei korrekter Dateigrösse / Dateityp: ok Bei zu grosser Datei (NEF 18 MB) Rückweisung (Fehlermeldung), wenn unter „Speichern“ verschickt.  Wenn unter „Speichern und Zurück“ verschickt, keine Fehlermeldung, aber auch kein neuer Datensatz angelegt. |

|  |  |
| --- | --- |
| Testfall-Nummer (#25) | Neuen Nutzer erfassen als Administrator |
| Testdaten |  |
| Aktionen | Validierung getestet |
| Resultat | neuer Admin korrekt angelegt |

|  |  |
| --- | --- |
| Testfall-Nummer (#26) | Veränderung der Nutzerdaten durch den Administrator in der Userdetailübersicht. |
| Testdaten |  |
| Aktionen | Änderungen aller Felder / Validierung getestet |
| Resultat | Fehlermeldungen werden adäquat gezeigt und fehlerhafte Änderungen verhindert. Korrekte Änderungen sind möglich. |

|  |  |
| --- | --- |
| Testfall-Nummer (#27) | Deaktivieren eines Benutzers durch den Administrator in der Userdetailübersicht. |
| Testdaten |  |
| Aktionen | Usermanager 🡪 „duser“ bearbeiten 🡪 deaktivieren |
| Resultat | „duser“ ist korrekt aus der Liste entfernt worden. Wunsch: Rückfrage à la „Wollen Sie User xxy wirklich löschen?“ |

|  |  |
| --- | --- |
| Testfall-Nummer (#28) | Veränderung der Benutzerdaten durch den Administrator in der Benutzerdetailübersicht. |
| Testdaten |  |
| Aktionen | Usermanager 🡪 „duser“ bearbeiten |
| Resultat | diverse Änderungen eingegeben / validiert. Funktioniert korrekt.  Auch Rolle lässt sich ändern. |

### Testprotokoll #2

Tester: Jonas Arab

**Testmittel:**

Windows 7 Pro

Google Chrome (Version 49.0.2623.112m)

|  |  |
| --- | --- |
| #01 | Login mit falschen Daten |
| Testdaten | Benutzer: invalid; Passwort: abc123 |
| Aktionen | Die oben genannten Daten wurden in die jeweiligen Textboxen eingegeben.  C:\Users\Jonas\Pictures\screens for dok\1.PNG |
| Resultat | Es wird eine Fehlermeldung ausgegeben.  C:\Users\Jonas\Pictures\screens for dok\2.PNG |

|  |  |
| --- | --- |
| #02 | Zugriff auf die Applikation ohne Login |
| Testdaten | URL: ipa.sbvg.eu/index.php |
| Aktionen | Es wurde versucht, per Link auf die Detailübersicht zuzugreifen ohne Login |
| Resultat | Die Applikation verweist auf den Login-Screen. |

|  |  |
| --- | --- |
| #03 | Login mit richtigen Daten |
| Testdaten | Benutzer: jar; Passwort: admin |
| Aktionen | Die oben genannten Daten wurden in die jeweiligen Textboxen eingegeben.  C:\Users\Jonas\Pictures\screens for dok\3.PNG |
| Resultat | Die Applikation verweist auf die Ticketübersicht. |

|  |  |
| --- | --- |
| #04 | Deaktivieren eines Benutzers durch den Administrator in der Userdetailübersicht. |
| Testdaten | Benutzer „duser2“ in der Benutzerübersicht |
| Aktionen | Als Administrator wurde im Usermanager das Profil „duser2“ bearbeitet.  C:\Users\Jonas\Pictures\screens for dok\5.PNG  Anschliessend wird auf „User deaktivieren geklickt  C:\Users\Jonas\Pictures\screens for dok\4.PNG |
| Resultat | Der User ist nicht mehr im Usermanager sichtbar. |

|  |  |
| --- | --- |
| #05 | Login mit übergebenen URL |
| Testdaten | URL: http://ipa.sbvg.eu/neuesticket. php |
| Aktionen | Es wurde versucht, per URL auf die Ticketerstellung zuzugreifen, ohne sich zuvor einzuloggen. |
| Resultat | Die Applikation verweist auf den Login-Screen. Nach Eingabe von korrekten Logindaten (jar/admin), verweist die Applikation auf „Neues Ticket“. |

|  |  |
| --- | --- |
| #06 | Anzeigen der Admin-Navigation |
| Testdaten | Benutzer: jar; Passwort: admin |
| Aktionen | Es wurde versucht, sich mit den obengenannten Daten sich einzuloggen. |
| Resultat | Die Applikation verweist auf die Ticketübersicht mit Usermanager.  C:\Users\Jonas\Pictures\screens for dok\6.PNG |

|  |  |
| --- | --- |
| #07 | Anzeigen der Benutzer-Navigation |
| Testdaten | Benutzer: jarb; Passwort: benutzer |
| Aktionen | Es wurde versucht, sich mit den obengenannten Daten sich einzuloggen. |
| Resultat | Die Applikation verweist auf die Ticketübersicht ohne Usermanager.  C:\Users\Jonas\Pictures\screens for dok\7.PNG |

|  |  |
| --- | --- |
| #08 | Korrekte Anzeige der Ticketübersicht |
| Testdaten | Benutzer: jar; Passwort: admin |
| Aktionen | Auf dem Navigationsbalken wurde auf „Ticketübersicht“ geklickt. |
| Resultat | Die Applikation zeigt die Ticketübersicht an.  C:\Users\Jonas\Pictures\screens for dok\8.PNG |

|  |  |
| --- | --- |
| #09 | Ticketfilterung |
| Testdaten | Kategorie: SBC; Status: Offen; Ersteller: ske; Supporter: Nicht Zugewiesen |
| Aktionen | Die oben genannten Kriterien wurden selektiert. |
| Resultat | Die Ticketübersicht zeigt nun das gefilterte Resultat an.  C:\Users\Jonas\Pictures\screens for dok\9.PNG |

|  |  |
| --- | --- |
| #10 | Zugriff auf die Administratorfunktionen als Administrator |
| Testdaten | Benutzer: jar; Passwort: admin |
| Aktionen | Login mit den oben genannten Daten.  Anschliessend Aufruf von „Neuer Benutzer“ und „Übersicht“ |
| Resultat | Der Administrator kann auf die jeweiligen Seiten zugreifen. |

|  |  |
| --- | --- |
| #11 | Zugriff auf die Administratorfunktionen als Benutzer |
| Testdaten | Benutzer: jarb; Passwort: benutzer |
| Aktionen | Login mit den obengenannten Daten.  Anschliessend Aufruf von „Neuer Benutzer“ und „Übersicht“ mittels URL. |
| Resultat | Die Applikation zeigt eine Fehlermeldung und verweist auf die Ticketübersicht.  C:\Users\Jonas\Pictures\screens for dok\10.PNG |

|  |  |
| --- | --- |
| #12 | Individuelles Benutzerprofil |
| Testdaten | Benutzer: jarb; Passwort: benutzer |
| Aktionen | Auf der Navigation wurde auf den Benutzernamen geklickt. |
| Resultat | Das Benutzerprofil wurde aufgerufen.  C:\Users\Jonas\Pictures\screens for dok\11.PNG |

|  |  |
| --- | --- |
| #13 | Individuelles Benutzerprofil |
| Testdaten | Vorname: Ai  Nachname: 33x „A“  Telefonnummer: 123  Mobilnummer: 456 |
| Aktionen | Es wurden ungültige Eingaben in der Benutzerbearbeitung eingegeben. |
| Resultat | Für die jeweiligen Felder gibt die Applikation eine Fehlermeldung aus. |

|  |  |
| --- | --- |
| #14 | Benutzerprofil aktualisieren |
| Testdaten | Vorname: Hans  Nachname: Muster  E-Mail: [jemandhalt@yahoo.de](mailto:jemandhalt@yahoo.de)  Telefonnummer: 0041614823216  Handynummer: 0041764083216 |
| Aktionen | Die oben genannten Daten wurden ein die jeweiligen Textfelder eingegeben.  C:\Users\Jonas\Pictures\screens for dok\14.PNG |
| Resultat | Die Daten werden ohne Fehler übernommen. (Überprüfung mittels Admin)  C:\Users\Jonas\Pictures\screens for dok\15.PNG |

|  |  |
| --- | --- |
| #15 | Detailansicht des Tickets als Benutzer |
| Testdaten | Benutzer: jarb; Passwort: benutzer |
| Aktionen | Auf der Ticketübersicht wurde bei einem Test-Ticket auf „Ansehen“ geklickt.  C:\Users\Jonas\Pictures\screens for dok\17.PNG |
| Resultat | Der Benutzer wird auf die Detailübersicht weitergeleitet.  C:\Users\Jonas\Pictures\screens for dok\16.PNG |

|  |  |
| --- | --- |
| #16 | Detailansicht des Tickets als Administrator |
| Testdaten | Benutzer: jar, Passwort: admin |
| Aktionen | Auf der Ticketübersicht wurde bei einem Test-Ticket auf „Ansehen“ geklickt.  C:\Users\Jonas\Pictures\screens for dok\17.PNG |
| Resultat | Der Admin wird auf die Detailübersicht weitergeleitet und kann zusätzlich die Daten bearbeiten.  C:\Users\Jonas\Pictures\screens for dok\18.PNG |

|  |  |
| --- | --- |
| #17 | Verändern des Tickets in der Detailansicht durch den Administrator |
| Testdaten | Titel: 123  Thema: One Voice  Beschreibung: Neue Beschreibung  Supporter: jar  Statur: In Beobachtung |
| Aktionen | Die oben genannten Daten werden in die jeweiligen Felder eingegeben.  C:\Users\Jonas\Pictures\screens for dok\19.PNG |
| Resultat | Die Daten werden korrekt übernommen.  C:\Users\Jonas\Pictures\screens for dok\20.PNG |

|  |  |
| --- | --- |
| Testfall-Nummer (#18) | Verändern des Ticketstatus in der Detailansicht durch den Administrator |
| Testdaten | Status: In Bearbeitung |
| Aktionen | Der Ticketstatus „In Bearbeitung“ wurde aus dem Dropdown-Menü selektiert.  C:\Users\Jonas\Pictures\screens for dok\21.PNG |
| Resultat | Der Status wird korrekt übernommen.  C:\Users\Jonas\Pictures\screens for dok\22.PNG |

|  |  |
| --- | --- |
| #19 | Verändern des Supporter in der Detailansicht durch den Administrator |
| Testdaten | Supporter: rke |
| Aktionen | Der Supporter „rke“ wurde aus dem Dropdown-Menü selektiert.  C:\Users\Jonas\Pictures\screens for dok\23.PNG |
| Resultat | Der Supporter wird korrekt übernommen.  C:\Users\Jonas\Pictures\screens for dok\24.PNG |

|  |  |
| --- | --- |
| #20 | Ticket erfassen |
| Testdaten | - |
| Aktionen | Auf der Navigation wurde auf „Ticket erfassen“ geklickt |
| Resultat | Der „Neues Ticket“-Dialog wird aufgerufen.  C:\Users\Jonas\Pictures\screens for dok\27.PNG |

|  |  |
| --- | --- |
| #21 | Ticket Validierung |
| Testdaten | - |
| Aktionen | Es wurde versucht, ein neues Ticket zu erstellen ohne irgendwelche Eingaben zu betätigen. |
| Resultat | Die Maske weist darauf hin, dass die Felder ausgefüllt werden sollen. |

|  |  |
| --- | --- |
| #22 | Ticket-Deadline Validierung |
| Testdaten | Titel: New Ticket  Thema:insight  Beschreibung: Neues Ticket  Endtermin: 2016-04-23 (02:03:55) Datei: rösti.jpg |
| Aktionen | Die obengenannten Daten wurden im „Neues Ticket“-Dialog eingegeben.  C:\Users\Jonas\Pictures\screens for dok\25.PNG |
| Resultat | Das neue Ticket wurde erfolgreich erstellt.  C:\Users\Jonas\Pictures\screens for dok\26.PNG |

|  |  |
| --- | --- |
| #23 | Ticket-Deadline Validierung |
| Testdaten | Endtermin: 2016-03-24 (00:00:00) |
| Aktionen | Es wurde versucht, ein Ticket mit bereits vergangenem Endtermin zu erstellen.  C:\Users\Jonas\Pictures\screens for dok\28.PNG |
| Resultat | Die Applikation gibt eine Fehlermeldung aus.  C:\Users\Jonas\Pictures\screens for dok\29.PNG |

|  |  |
| --- | --- |
| #24 | Datei Validierung |
| Testdaten | Datei: Waters of Megalovania.mp3 (Grösse: 8.04 MB) |
| Aktionen | Es wurde versucht, ein Ticket zu erstellen, welches eine Datei mit ungültigem Format und unzulässiger Grösse beinhaltet. |
| Resultat | Bei „Speichern“ gibt die Applikation eine Fehlermeldung aus.  C:\Users\Jonas\Pictures\screens for dok\31.PNG  Bei „Speichern und zurück“ gibt die Applikation keine Fehlermeldung aus. Sie verweist lediglich auf die Ticketübersicht zurück. Das Ticket wird jedoch nicht erstellt. |

|  |  |
| --- | --- |
| #25 | Neuer Nutzer erfassen als Administrator |
| Testdaten | Vorname:ai  Nachname: 33x „A“  Username: Oy  E-Mail: jonas[at]mail.ch  Telefon: 123  Handy: 456  Passwort: 123  Rolle: Benutzer |
| Aktionen | Es wurde versucht, als Administrator einen neuen Benutzer mit ungültigen eingaben zu erstellen.  C:\Users\Jonas\Pictures\screens for dok\32.PNG |
| Resultat | Die Applikation gibt für jede ungültige Eingabe eine Fehlermeldung aus.  C:\Users\Jonas\Pictures\screens for dok\33.PNG |

|  |  |
| --- | --- |
| #26 | Veränderung der Nutzerdaten durch den Administrator in der Userdetailübersicht. |
| Testdaten | Zu bearbeitender Benutzer: JoAra - Rolle: Benutzer  Vorname:ai  Nachname: 33x „A“  Username: Oy  E-Mail: jonas[at]mail.ch  Telefon: 123  Handy: 456  Passwort: 123 |
| Aktionen | Es wurde versucht, einen bereits vorhandenen Benutzer mit ungültigen Eingaben zu ergänzen.  C:\Users\Jonas\Pictures\screens for dok\34.PNG |
| Resultat | Es werden keine Änderungen übernommen. Die Applikation gibt eine Fehlermeldung aus.  C:\Users\Jonas\Pictures\screens for dok\33.PNG |

|  |  |
| --- | --- |
| #27 | Login eines deaktivierten Users |
| Testdaten | Der Benutzer „duser2“ |
| Aktionen | Der Benutzer „duser2“ wurde deaktiviert. Es wurde versucht, sich nun mit den Logindaten des deaktivierten Users einzuloggen. |
| Resultat | Das Login schlägt fehl. Der Benutzer ist nicht mehr vorhanden. |

|  |  |
| --- | --- |
| #28 | Veränderung der Benutzerdaten durch den Administrator in der Userdetailübersicht. |
| Testdaten | Der zuvor erstellte Benutzer „JoAra“ |
| Aktionen | Der Vorname, Nachname, Username und die E-Mail wurden mit einem „x“ ergänzt.“  Die letzte Zahl der Telefonnummern wurde mit der Nummer 2 ausgetauscht.  Das Passwort wurde auf 456123 umgeändert.  Die Rolle wurde von „Benutzer“ auf „Administrator“ umgeändert.  Es wurde versucht, die Änderungen zu speichern.  C:\Users\Jonas\Pictures\screens for dok\36.PNG |
| Resultat | Die Änderungen wurden erfolgreich übernommen.  C:\Users\Jonas\Pictures\screens for dok\37.PNG |

# Abschlussbericht

## Rückblick

## Lessons learned & Was ist gut gelaufen

## Was schief gelaufen ist

## Mit welchen Funktionen kann das Projekt sinnvoll erweitert werden?

Hier zeige ich nennenswerte Funktionen der Applikation auf, die ich gerne bei einer Version 2.0 hinzufügen würde.

### Tickets und Benutzer in der Detailübersicht

Momentan werden die Benutzer per ID in der URL geladen. Diese möchte ich per Hash ansprechen. Dieser Hash wird beim Erstellen des Nutzers oder des Tickets generiert und ist zufallsbasiert. Diese zusätzliche Funktion hat Änderungen in der Datenbank und am Code zufolge. Die Datenbank erhält für jeweils den Benutzer und die Tickets ein weiteres Feld mit dem zufälligen Hash. Der Hash wird beim Erstellen vor dem speichern des Tickets / Nutzers in die Datenbank generiert. Nach dem generieren des Hashes wird gecheckt, ob dieser schon benutzt wird, falls nicht, wird das Ticket / der Benutzer mit diesem Hash abgespeichert. Der Hash wird für das aufrufen in der Detailansicht verwendet.

### Benutzerfilter und/oder Ansicht für deaktivierte Nutzer

Beim Deaktivieren eines Nutzers wird dieser nicht mehr in der Nutzerübersicht angezeigt und kann nur noch per Datenbank verändert werden. Mit einer zusätzlichen Ansicht und / oder Filterfunktion für die Benutzer wie diese bei den Tickets kann dies umgehen werden.

### Passwort ändern Funktion auf eine eigene Seite auslagern

Ich möchte die Passwort ändern Funktion auf eine eigene Seite auslagern.

### Dem Ticket als Administrator einen Kommentar hinzuzufügen

Zum aktuellen Zeitpunkt kann beim Schliessen des Tickets kein Schliessungsgrund angegeben werden. Das Möchte ich mit der Version 2.0 Ändern. Der Supporter, welcher das Ticket bearbeitet, soll dem Ticket ein Text hinzufügen können, mit welchem er dem Ticket antworten kann. Dieser soll auch per E-Mail an den Ticketersteller gesendet werden.

## Resultat

## Schlusswort

# Quellenverzeichnis

## Versionierung

<https://de.wikipedia.org/wiki/Versionsnummer>

* Darstellung der Versionierung übernommen und erweitert.
* Als Referenz für den Text benutzt

## Risikoanalyse

https://de.wikipedia.org/wiki/Risikoanalyse

* Als Referenz für den Text benutzt

## Bootstrap

<http://stackoverflow.com/questions/14227000/pros-cons-of-switching-to-twitter-bootstrap>

<http://www.hyperarts.com/blog/twitter-bootstrap-sucks-and-its-awesome/>

<http://www.htmlcenter.com/blog/the-bootstrap-framework-controversy-should-you-use-it-or-not/>

* Vor- und Nachteile übersetzt und übernommen

# Glossar

Administrator 32

Cookie 34

CSS 32

Footer 32

Framework 28

Front-End 28

GIT 24

GitHub 24

GUI 10

Header 32

Html 32

IPERKA 10

JS 32

Log-Out 32

Mobile-first 28

PHP-Datei 32

Prototyping 28

Repository 24

responsive 28

Session 33

Updaten 28

Wasserfall-Modell 10

XAMPP 24

# Anhang