
April 2016, Final-Version

IPA-Bericht

Samuel Keller

Inhaltsverzeichnis

1 Obligatorischer Teil	6
1.1 Ausgangslage	6
1.2 Detaillierte Aufgabenstellung	6
1.2.1 Login	6
1.2.2 Benutzerverwaltung	6
1.2.3 Die Verwaltung von Tickets	6
1.2.4 Ticket Workflow	7
1.2.5 Dokumentation	7
1.2.6 Design	7
1.2.7 Testing	7
1.2.8 Versionsverwaltung	8
1.2.9 Abgrenzungen	8
1.2.10 Zeitplan	8
1.2.10.1 Initialisierung (Montag, 04.04.2016)	8
1.2.10.2 Analyse (Dienstag, 05.04.2016)	8
1.2.10.3 Entwurf (Mittwoch, 06.04.2016)	8
1.2.10.4 Implementierung (Mittwoch, 06.04.2016 - Dienstag, 12.04.2016)	8
1.2.10.5 Testing (Montag, 11.04.2016 - Dienstag, 13.04.2016)	8
1.2.10.6 Abschlussphase (Montag, 13.04.2016 - Dienstag, 14.04.2016)	8
1.2.10.7 Buffer (Freitag, 15.04.2016)	8
1.2.11 Mittel und Methoden	9
1.2.12 Vorkenntnisse	9
1.2.13 Vorarbeiten	9
1.2.14 Neue Lerninhalte	9
1.3 Projektorganisation	10
1.3.1 Projektmanagementmethode	10
1.3.1.1 Initialisierungsphase	10
1.3.1.2 Analysephase	10
1.3.1.3 Entwurfsphase	10
1.3.1.4 Implementierungsphase	10
1.3.1.5 Testingphase	10
1.3.1.6 Abschlussphase	10
1.3.2 Zeitplan	11
1.4 Arbeitsjournal	13
1.4.1 Montag, 4. April 2016	13
1.4.2 Dienstag, 5. April 2016	14
1.4.3 Mittwoch, 6. April 2016	15
1.4.4 Donnerstag, 7. April 2016	16
1.4.5 Freitag, 8. April 2016	17
1.4.6 Montag, 11. April 2016	18
1.4.7 Dienstag, 12. April 2016	19
1.4.8 Mittwoch, 13. April 2016	20
1.4.9 Donnerstag, 14. April 2016	21

1.4.10	Freitag, 15. April 2016	22
2	Projektdokumentation	23
2.1	Ausgangslage	23
2.1.1.1	Ist-Situation	23
2.1.1.2	Soll-Situation	24
2.2	Risikoanalyse	25
2.2.1	Datenverlust	25
2.2.1.1	Wahrscheinlichkeit	25
2.2.1.2	Folgen	25
2.2.1.3	Gegenmassnahmen	25
2.2.1.4	Schadensminimierung	25
2.2.2	Zeitprobleme	26
2.2.2.1	Wahrscheinlichkeit	26
2.2.2.2	Folgen	26
2.2.2.3	Gegenmassnahmen	26
2.2.2.4	Schadensminimierung	26
2.2.3	Internetbrowser-Probleme	27
2.2.3.1	Wahrscheinlichkeit	27
2.2.3.2	Folgen	27
2.2.3.3	Gegenmassnahmen	27
2.2.3.4	Schadensminimierung	27
2.3	Versionierung	28
2.3.1	Backup	28
2.3.2	Commits	28
2.4	Wieso Bootstrap	29
2.4.1	Was ist Bootstrap	29
2.4.2	Vorteile	29
2.4.3	Nachteile	29
2.4.4	Wieso Bootstrap benutzt wird	29
2.5	Planung der Implementierung	30
2.5.1	Login	30
2.5.2	Benutzerverwaltung	30
2.5.3	Die Verwaltung von Tickets	30
2.5.4	Ticket Workflow	30
3	Design	31
3.1	Datenbank	31
3.1.1	Vor der Implementierung	31
3.1.2	Nach der Entwicklung der Applikation	32
4	Implementierung	33
4.1	Standard GUI	33
4.1.1	header.inc.php	33
4.1.1.1	getHeader()	33
4.1.2	navigation.inc.php	33
4.1.3	footer.inc.php	34

4.2 Autorisierungs-Seiten.....	34
4.2.1 checkauth.inc.php	34
4.2.2 checkrole.inc.php	35
4.3 Loginseiten	35
4.3.1 login.php.....	35
4.3.2 logout.php.....	35
4.3.2.1 logOut().....	36
4.4 Datenbank Verbindungsseiten	36
4.4.1 dbsettings.inc.php	36
4.4.2 dbconnection.inc.php	36
4.4.3 pdosettings.inc.php	36
4.4.4 pdoclass.inc.php	37
4.5 Funktionsseiten.....	37
4.5.1 functioncontroller.inc.php.....	37
4.5.2 sendmail.inc.php	37
4.5.2.1 sendMail()	38
4.6 Ticketverwaltungsseiten	39
4.6.1 index.php.....	39
4.6.1.1 ticketfilter.inc.php.....	39
4.6.2 ticketliste.inc.php	40
4.6.3 neuesticket.php.....	41
4.6.4 ticket_details.php	46
4.7 Userverwaltung.....	47
4.7.1 profil.php	47
}	47
4.7.2 usermanagement.php.....	48
4.7.3 neuernutzer.php	49
4.7.4 benutzer_details.php	49
5 Testing	51
5.1 Endresultat	51
5.2 Testfälle.....	51
5.3 Testprotokolle	60
5.3.1 Testprotokoll #1.....	60
5.3.2 Testprotokoll #2.....	66
6 Abschlussbericht	79
6.1 Rückblick.....	79
6.2 Lessons learned & Was ist gut gelaufen	79
6.3 Was schief gelaufen ist	79
6.4 Resultat.....	79
6.5 Mit welchen Funktionen kann das Projekt sinnvoll erweitert werden?	79
6.5.1 Tickets und Benutzer in der Detailübersicht.....	80
6.5.2 Benutzerfilter und / oder Ansicht für deaktivierte Nutzer.....	80
6.5.3 Passwort ändern Funktion auf eine eigene Seite auslagern	80
6.5.4 Dem Ticket als Administrator einen Kommentar hinzuzufügen.....	80

6.6	Schlusswort	80
7	Quellenverzeichnis.....	81
7.1	Versionierung.....	81
7.2	Risikoanalyse	81
7.3	Bootstrap.....	81
7.4	Weitere Quellen	81
8	Glossar	81

1 Obligatorischer Teil

1.1 Ausgangslage

Die Abteilung Web Service ist für verschiedene 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, und somit die Fehleranfälligkeit erhöht ist. Die Koordination und Konsolidierung der Aufträge durch ein zentrales webbasiertes Ticketingsystem würde für die Benutzer sowie für die Webabteilung eine grosse Entlastung bringen.

1.2 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:

1.2.1 Login

Die Implementierung eines einfachen und benutzerfreundlichen Logins für den Benutzer.

1.2.2 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.

1.2.3 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, der Status des Tickets, eine allfällige Deadline, der 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.

1.2.4 Ticket Workflow

Jeder Benutzer hat die Berechtigung, ein neues Ticket zu erfassen. Das Ticket kann mittels einem Formular 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.

1.2.5 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.

1.2.6 Design

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

1.2.7 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 ist 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 Ex-

plerer erfolgreich war. Die Testfälle werden ebenfalls separat in Testprotokollen festgehalten.

1.2.8 Versionsverwaltung

Die Versionsverwaltung wird mit einem git GUI-Client erstellt.

1.2.9 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.

1.2.10 Zeitplan

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

1.2.10.1 Initialisierung (Montag, 04.04.2016)

Grundgerüst der Dokumentation, erste Texte erstellen, GIT einrichten

1.2.10.2 Analyse (Dienstag, 05.04.2016)

Analyse der Webseite, Erstellung der Testfälle

1.2.10.3 Entwurf (Mittwoch, 06.04.2016)

Erstellung des Webseitenlayouts.

1.2.10.4 Implementierung (Mittwoch, 06.04.2016 - Dienstag, 12.04.2016)

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

1.2.10.5 Testing (Montag, 11.04.2016 - Dienstag, 13.04.2016)

Testen der Applikation und Erstellung des Testberichtes

1.2.10.6 Abschlussphase (Montag, 13.04.2016 - Dienstag, 14.04.2016)

Abschluss der IPA-Dokumentation und des Benutzerhandbuchs.

1.2.10.7 Buffer (Freitag, 15.04.2016)

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

•SwissBanking

1.2.11 Mittel und Methoden

- PHP
- CSS
- HTML
- MySQL
- Git
- Eclipse

1.2.12 Vorkenntnisse

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

1.2.13 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.

1.2.14 Neue Lerninhalte

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

1.3 Projektorganisation

1.3.1 Projektmanagementmethode

Der Auftrag ist mit 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 in dieser Methode wurde schliesslich das Wasserfall-Modell gewählt.

Das Wasserfall-Modell wurde in die folgenden Phasen unterteilt:

1.3.1.1 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.

1.3.1.2 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.

1.3.1.3 Entwurfsphase

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

1.3.1.4 Implementierungsphase

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

1.3.1.5 Testingphase

In dieser Phase wird die Qualität der Applikation sichergestellt. Aufgrund eines Testkonzepts wird ein Test-Szenario erstellt. Auf diesem basieren Testfälle, mit denen die Software auf ihre Funktionen getestet und ausgewertet werden. Die Resultate werden in Test-Protokollen festgehalten.

1.3.1.6 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.

1.3.2 Zeitplan

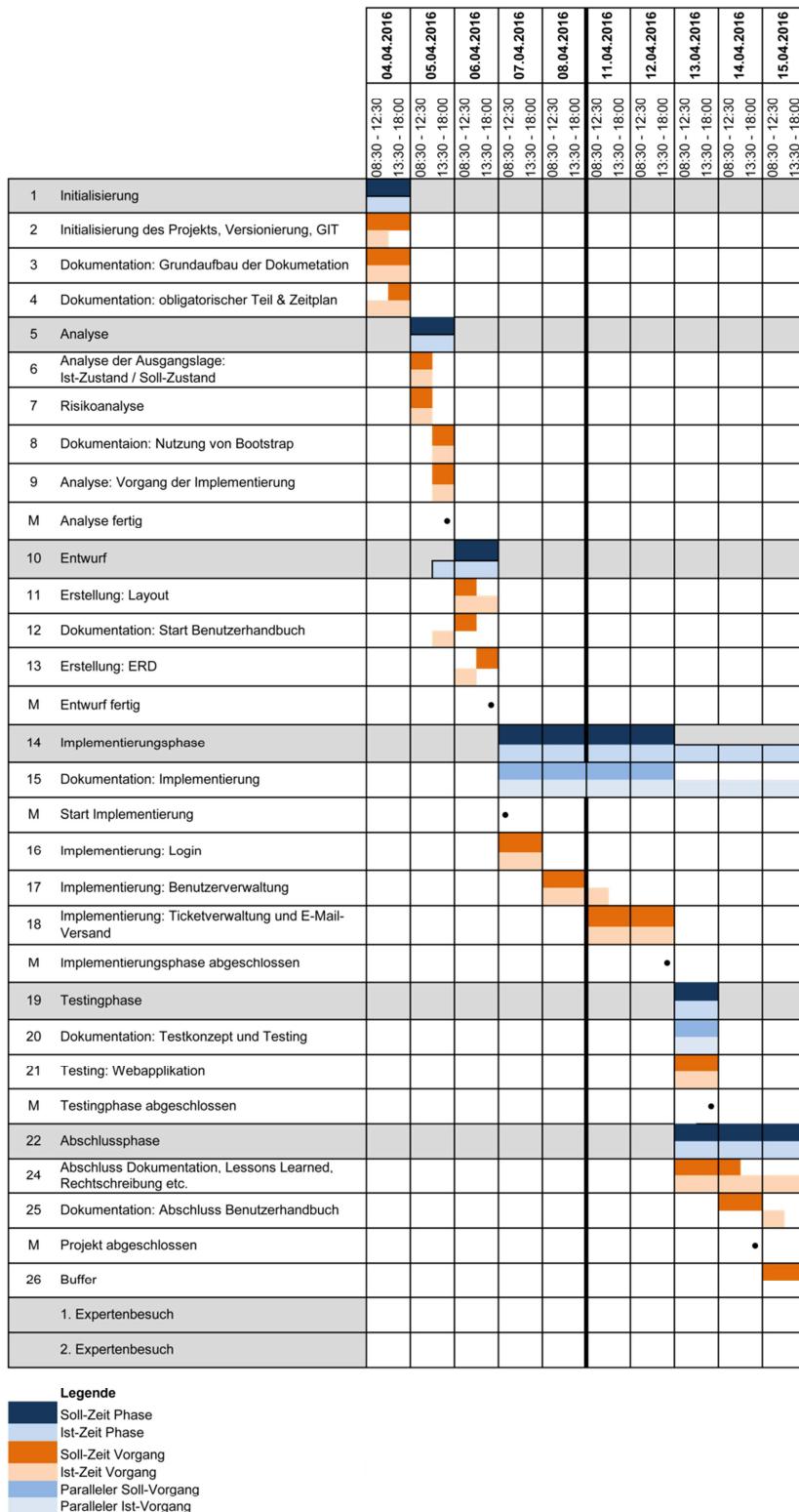


Abbildung: Der Zeitplan am Schluss des Projektes

•SwissBanking

1.4 Arbeitsjournal

1.4.1 Montag, 4. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - 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 korrekturlesen 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 mehr Sinn ergibt, sondern daran, dass ich gerne zwischen den einzelnen Dokumenten und Texten wechsle. So fixiere 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	

1.4.2 Dienstag, 5. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - 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 korrekturlesen 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 mache ich mir bereits Gedanken.
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	

1.4.3 Mittwoch, 6. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - Arbeitsjournal (Tag 2) in die Dokumentation übernehmen - Use-Case dokumentieren - Bootstraptex in die Dokumentation übernehmen - Risikoanalyse in die Dokumentation übernehmen - Zwei ERDs 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 ERDs 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	

1.4.4 Donnerstag, 7. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - 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 Implementierungsaufwand 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	

1.4.5 Freitag, 8. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - Arbeitsjournal (Tag 4) in die Dokumentation übernehmen - Implementierung des Benutzerprofils mit Validierung - Implementierung der Funktion „Neuer Nutzer erstellen“ - <u>Implementierung der Benutzerliste</u>
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	

1.4.6 Montag, 11. April 2011

Erreichte Ziele	<ul style="list-style-type: none"> - Arbeitsjournal (Tag 5) übernehmen - Projektvorgangstext 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 Ticketingsystem in einer zweiten Version erweitern könnte. Diese habe ich im Abschlussbericht eingefügt. Da ich jetzt PDO nutze, mit dem ich mehr vertraut bin, kommt die Implementation der Funktionen um einiges schneller voran.
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 von Login und 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 11 Uhr war ich damit fertig und bin jetzt wieder im Zeitplan.
Sonstiges	Ich habe ein paar der von Hand erstellten Texte in die Dokumentation übernommen und alle zukünftigen Journale in die Dokumentation eingefügt, um Formatierungsprobleme entgegen zu wirken. Die neue Word-Version verträgt sich scheinbar nicht gut mit der älteren. Die Experten waren heute für etwa eine halbe Stunde da. Sie gaben Tipps für die Erweiterung der Dokumentation und Applikation. 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	

1.4.7 Dienstag, 12. April 2016

Erreichte Ziele	<ul style="list-style-type: none">- Arbeitsjournal (Tag 6) übernehmen- Detailansicht für Tickets implementieren- Datenbank Feldtypen ändern- Datenbankeinträge zu den Kategorien ändern- „Ticket erstellen“-Funktion implementieren.- Benutzer- und Profilvalidierung vereinen- File-Upload implementieren- E-Mail versenden implementieren- Ticket und File-Validierung implementieren
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 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 fertiggestellt ist.
Arbeitszeit	08:30 Uhr 12:30 Uhr Mittagspause 13:30 Uhr – 18:00 Uhr
Visum	

1.4.8 Mittwoch, 13. April 2016

Erreichte Ziele	<ul style="list-style-type: none">- Arbeitsjournal (Tag 7) in die Dokumentation übernehmen- Testfälle 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	

1.4.9 Donnerstag, 14. April 201

Erreichte Ziele	<ul style="list-style-type: none"> - Arbeitsjournal (Tag 8) in die Dokumentation übernehmen - Dokumentieren - Qualitätssicherung in die Dokumentation übernehmen - Implementierung dokumentieren
Nicht erreichte Ziele	Ich hatte keine Zeit, das Benutzerhandbuch zu schreiben
Frühzeitig erreichte Ziele	keine
Ungeplante Arbeiten	keine
Geplante Arbeiten nach Arbeitsschluss	Texte zu Hause korrekturlesen lassen.
Aufgetretene Probleme	Zeitmangel
Unterstützung Dritter	IPA-Dokumentation durch Rolf Keller korrekturgelesen. Felix Krieger hat Screenshots für das Benutzerhandbuch erstellt.
Soll / Ist Zeitvergleich	Ich bin im starken Zeitdruck. Die Dokumentation benötigt mehr Zeit als erwartet.
Sonstiges	Die Experten haben mich um 11:15 Uhr besucht. Sie haben sich die Applikation und die Dokumentation angesehen und mir gesagt, dass ich noch viel zu dokumentieren habe und der letzte Tag ziemlich stressig werden wird.
Arbeitszeit	08:30 Uhr – 11:15 Uhr Expertenbesuch 11:45 Uhr – 12:30 Uhr Mittagspause 13:30 Uhr – 18:00 Uhr
Visum	

1.4.10 Freitag, 15. April 2016

Erreichte Ziele	<ul style="list-style-type: none"> - Arbeitsjournal (Tag 9) in die Dokumentation übernehmen - Arbeitsjournal (Tag 10) in die Dokumentation übernehmen - Implementierung Dokumentieren - Benutzerhandbuch schreiben - IPA-Kurzbericht schreiben - Web-Summary schreiben
Nicht erreichte Ziele	keine
Frühzeitig erreichte Ziele	keine
Ungeplante Arbeiten	kleine
Geplante Arbeiten nach Arbeitsschluss	keine
Aufgetretene Probleme	Zeitmangel für das fertigstellen der Dokumentation.
Unterstützung Dritter	IPA-Dokumentation durch FV korrigieren lassen. Benutzerhandbuch durch FV korrigieren lassen.
Soll / Ist Zeitvergleich	Alles bis auf die Dokumentation und ist fertig. Ich hätte den Fokus mehr auf die Dokumentation legen müssen und muss jetzt das Beste aus der verbleibenden Zeit machen.
Sonstiges	Mir geht es durch den Zeitstress und Zeitdruck gar nicht gut.
Arbeitszeit	08:30 Uhr – 12:00 Uhr Mittagspause 12:30 Uhr – 17:00 Uhr
Visum	

2 Projektdokumentation

2.1 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.

2.1.1.1 Ist-Situation

Die Grafik zeigt wie die Web Abteilung (im Folgenden als „Web“ bezeichnet) in Kommunikation mit den anderen Nutzerinnen und Nutzer steht.

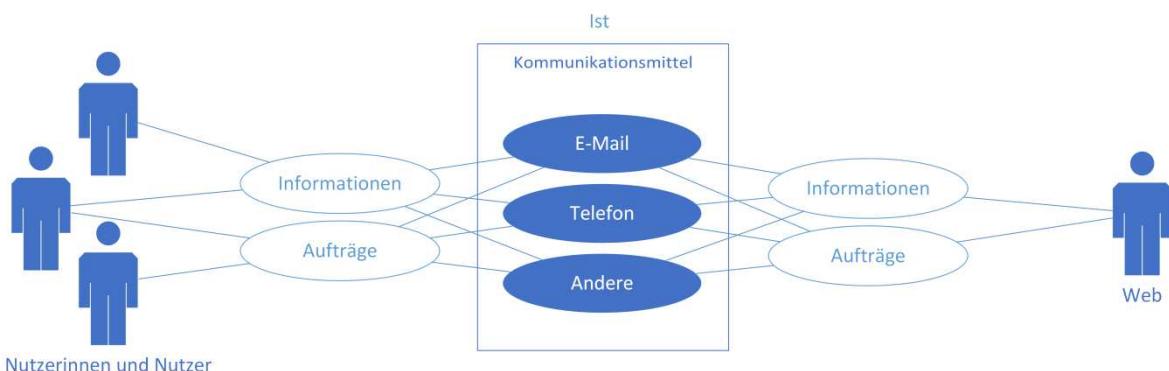


Abbildung: Aktueller Kommunikationsfluss der Nutzerinnen und Nutzer zum Web

Die Kommunikation zwischen den Nutzerinnen und Nutzern wird nicht ausschliesslich über ein Medium getätigt. Die Nutzerinnen und Nutzer kommunizieren hauptsächlich über E-Mail und Telefon mit dem Web. Dieses 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.

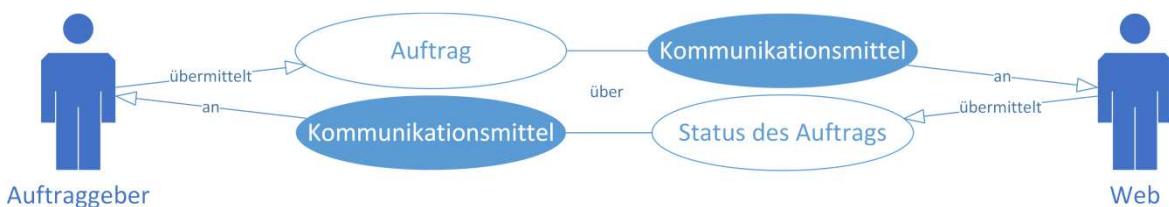


Abbildung: Kommunikation eines Auftrages

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

2.1.1.2 Soll-Situation

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

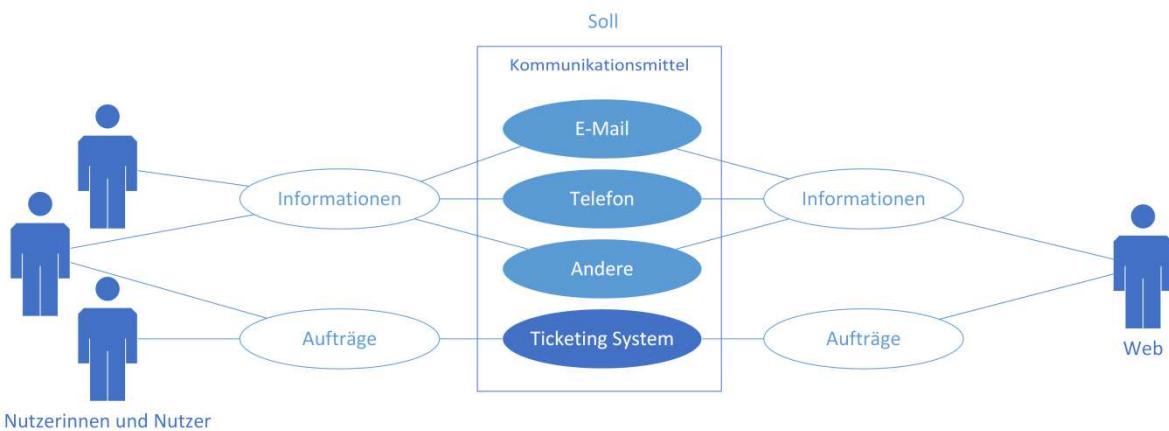


Abbildung: Aktueller Kommunikationsfluss der Nutzerinnen und Nutzer zum Web.

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 Ticketingsystem kommuniziert. So werden an arbeitsintensiven Tagen die Informationen nicht mit Aufträgen vermischt. Die Nutzerinnen und Nutzer ändern das Kommunikationsmittel für die Aufträge und erleichtern damit der Web Abteilung die Übersicht.

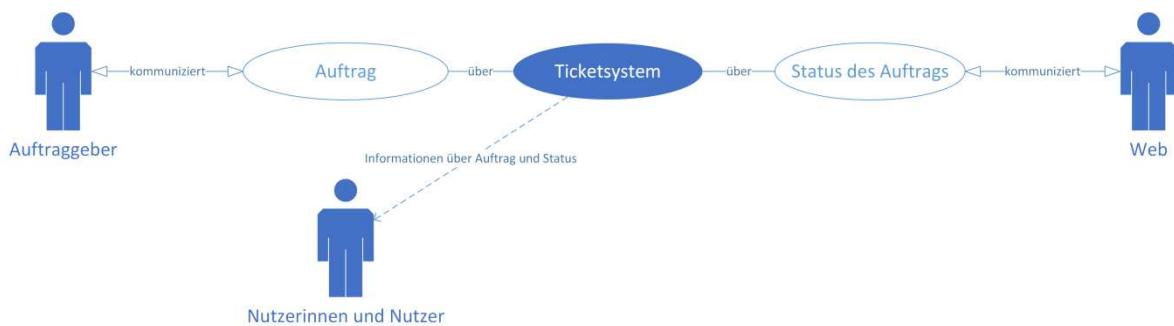


Abbildung: Kommunikation eines Auftrages nach der Einführung des Ticketingsystems

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

2.2 Risikoanalyse

2.2.1 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.

2.2.1.1 Wahrscheinlichkeit

Die Wahrscheinlichkeit eines Datenverlustes ist Dank mehreren Backups 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.

2.2.1.2 Folgen

Die Folgen eines Datenverlustes hängen ab vom Ausmass des Datenverlustes und von der Aktualität des Backups. Ein grösserer Verlust kann 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 vor dem Datenverlust wieder erreichen muss.

2.2.1.3 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 in GIT eine Versionierung. Durch das mehrfache Absichern der Dokumentation und der Applikation kann jederzeit auf ein Backup zurückgegriffen werden. Falls ein Harddiskfehler auftreten sollte, kann ohne grössere Probleme an einem anderen Arbeitsplatz weiter gearbeitet werden. Um die Entwicklung zu vereinfachen, wird XAMPP benutzt.

2.2.1.4 Schadensminimierung

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

2.2.2 Zeitprobleme

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

2.2.2.1 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 aufgewendet werden oder die bemessene Zeit für das Projekt falsch eingeteilt sein.

2.2.2.2 Folgen

Zeitprobleme können vom nicht korrigierten Text bis zur verpassten Fertigstellung des Projektes führen. Das Verpassen des Abgabetermins ist dabei auch ein Problem, das auftreten kann. Die Folge wäre die Abgabe einer nicht vollständigen Dokumentation.

2.2.2.3 Gegenmassnahmen

Um genügend Zeit für alles zu haben, wurde ein Zeitbuffer eingeplant. In dieser Zeitreise können letzte anstehende Arbeiten erledigt werden. Jedoch würde ich lieber ein nicht fertiges Produkt abgeben, als den Abgabetermin zu verpassen.

2.2.2.4 Schadensminimierung

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

2.2.3 Internetbrowser-Probleme

Die geplante Applikation wird für die meistgenutzten 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.

2.2.3.1 Wahrscheinlichkeit

Browserprobleme können vor allem bei der Darstellung des GUI auftreten. Die Wahrscheinlichkeit, dass etwas nicht wie gewünscht funktioniert oder aussieht, ist bei der Unterstützung von mehreren Browsern über 60%.

2.2.3.2 Folgen

Bei Browser-Problemen wird es zu Zeitverzögerungen kommen, da browserspezifische Anpassungen implementiert werden müssen. Dies kann wiederum zu Zeitproblemen führen.

2.2.3.3 Gegenmassnahmen

Um den Darstellungsproblemen in den verschiedenen Browsern entgegenzuwirken, 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.

2.2.3.4 Schadensminimierung

Falls dennoch Probleme auftreten, muss der Fehler ausgewertet werden. Wenn der Fehler nur optisch ist und die Funktionalität der Applikation nicht einschränkt, kann er 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 Dritter und dem Internet gelöst werden.

2.3 Versionierung

Für die Versionierung des GIT-Repositorys wurde ein klassisches System mit Hauptversionsnummer, Nebenversionsnummer, Revisionsnummer und Buildnummer mit ergänzenden 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 derjenigen des Repositorys. Das Repository startet nur mit Revisions- und Buildnummer, da am Anfang nur die Dokumentation existiert. Mit der Implementierung 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:

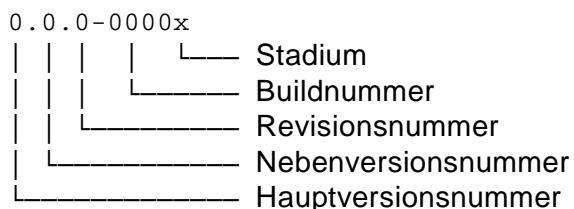


Abbildung: Quelle Wikipedia, erweitert mit dem Softwarestadium

2.3.1 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.

2.3.2 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.

2.4 Wieso Bootstrap

2.4.1 Was ist Bootstrap

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

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

2.4.2 Vorteile

Bootstrap

- erleichtert und beschleunigt die Entwicklung
- erlaubt schnelles Prototyping
- funktioniert in allen modernen Browsern
- unterstützt JavaScript und jQuery
- arbeitet mit einer eigenen Gitter-Struktur (Grid)
- unterstützt prioritätär die Darstellung auf mobilen Geräten („Mobile-First“)
- löst viele CSS Probleme von Anfang an
- benötigt wenig zusätzliches (custom) CSS
- ist Open Source

2.4.3 Nachteile

- Durch das vordefinierte CSS des Frameworks sehen viele Webseiten ähnlich 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 bleibt die Version von Bootstrap, mit der die Applikation erstellt wurde, die finale Bootstrap-Version.

2.4.4 Wieso Bootstrap benutzt wird

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

2.5 Planung der Implementierung

Das Ticketingsystem besteht aus drei Komponenten. Da diese Applikationsteile aufeinander aufbauen, müssen sie 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 Benutzerverwaltung implementiert. Erst am Schluss wird das Erstellen und Bearbeiten von Tickets ermöglicht, da diese die Benutzerverwaltung und das Login voraussetzen.

2.5.1 Login

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

2.5.2 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.

2.5.3 Die Verwaltung von Tickets

Auf der Ticketübersicht sind alle aktiven Tickets publiziert. Die Tickets 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.

2.5.4 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 erhalten die Administratoren eine E-Mail-Benachrichtigung. Das Ticket wird danach 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.

3 Design

3.1 Datenbank

3.1.1 Vor der Implementierung

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

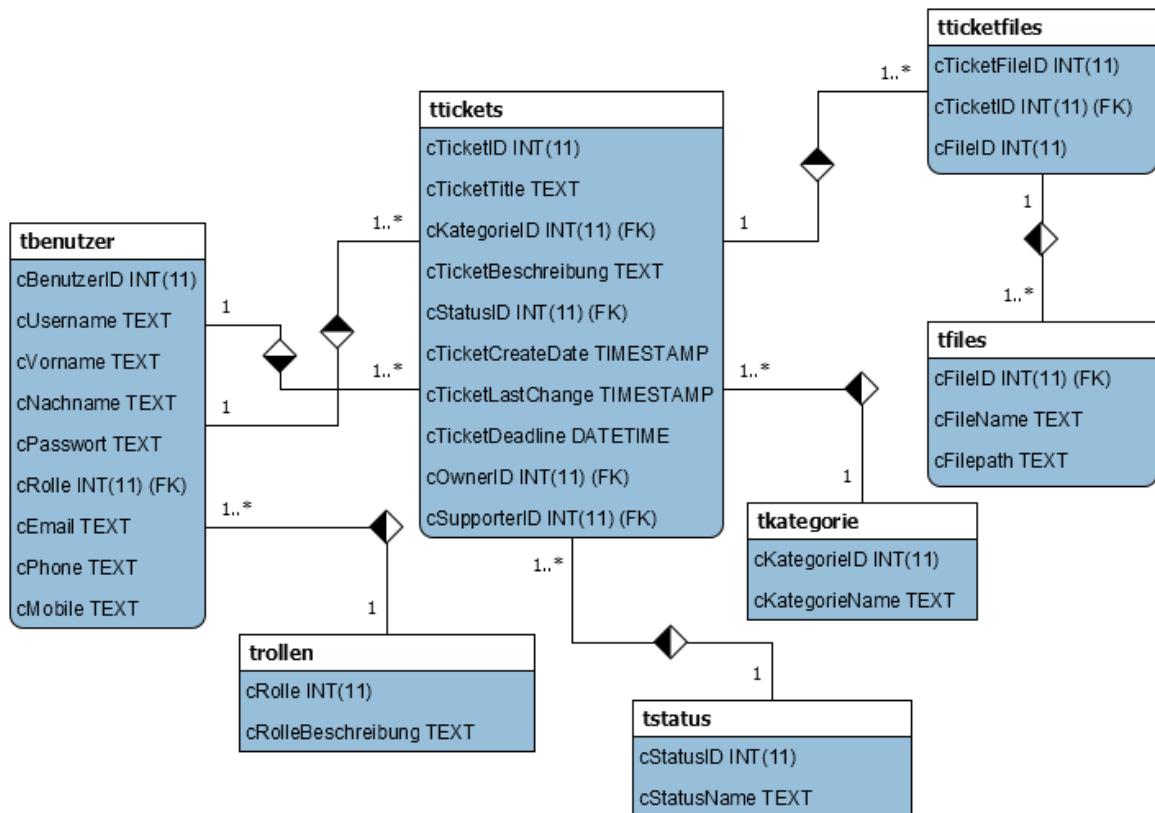


Abbildung: Datenbank vor der Erstellung der Applikation

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 hochgeladenen Dateien, welche über den Pfad in der Tabelle „tfiles“ referenziert werden. Diese gespeicherten Dateien 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 wird mit dem Feldtyp „TIMESTAMP“ und „DATETIME“ gespeichert.

3.1.2 Nach der Entwicklung der Applikation

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

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 deaktiverter Benutzer kann sich nicht mehr einloggen.

Die veränderte Datenbank hat die folgende Struktur:

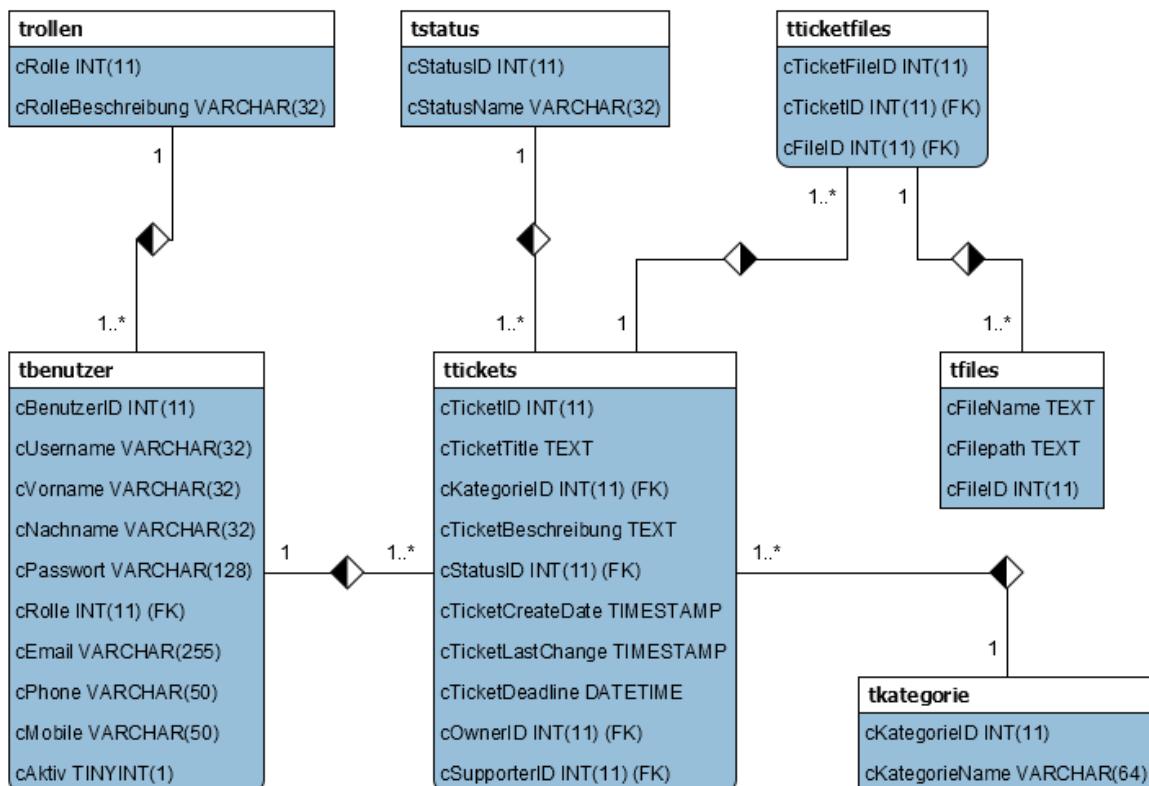


Abbildung: Neue Datenbankstruktur

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

4 Implementierung

4.1 Standard GUI

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

4.1.1 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.

4.1.1.1 getHeader()

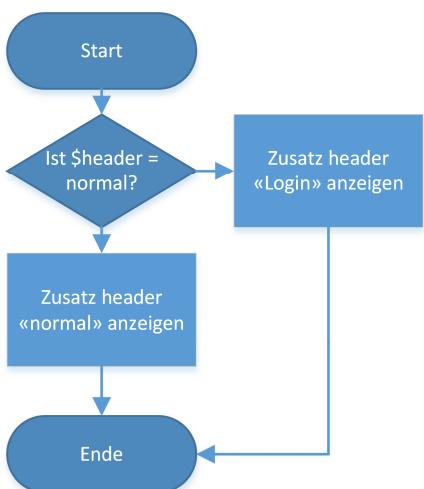


Abbildung: getHeader

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

4.1.2 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 zusätzlichen 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 zum Benutzerhandbuch. Die Navigation wird durch eine Variabel bestimmt, die in der PHP-Datei `checkauth.inc.php` festgelegt wird.

Da die Navigation responsiv ist, wird die Navigation bei einer grössse von unter 1150 Pixel eingeklappt.

•SwissBanking

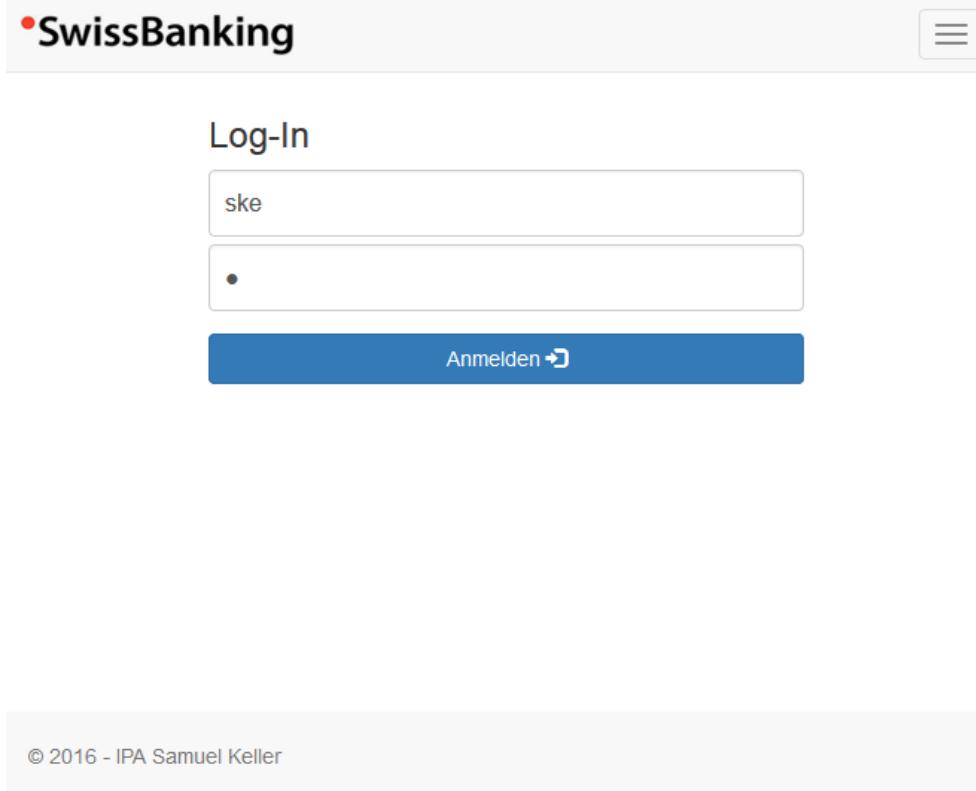


Abbildung: Die Loginseite mit eingeklappter Navigation

4.1.3 footer.inc.php

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

4.2 Autorisierungs-Seiten

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

4.2.1 checkauth.inc.php

Diese Prüfung wird auf jeder internen Ticketingseite durchgeführt. Es wird geprüft, ob der User, der auf die Ticketingapplikation zugreifen möchte, eingeloggt ist. Falls dies nicht der Fall ist, wird 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.

4.2.2 checkrole.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 Administratorfunktionen 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 des Funktionverlaufs und Code-Ausschnitt der Seite checkrole.inc.php

4.3 Loginseiten

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

4.3.1 login.php

Die Loginseite prüft nach der Eingabe von Benutzernamen und Passwort die Richtigkeit der Daten. Falls Benutzernamen oder Passwort falsch sind, wird eine Fehlermeldung ausgegeben. Ebenfalls wird diese ausgegeben, wenn der Benutzer, der sich einloggen will, deaktiviert ist. Sind die Daten richtig, werden die BenutzerID, der Vornamen, Nachnamen, Benutzernamen und die Rolle in eine Session gespeichert. Diese Session kann von anderen Funktionen aufgerufen werden. Beispielsweise wird die Navigation aufgrund der Rolle in der Session gewählt.

4.3.2 logout.php

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

4.3.2.1 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();

4.4 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, welcher noch nicht implementiert war, PDO zu verwenden, da dies die Implementierung wesentlich erleichtert und beschleunigt.

4.4.1 dbsettings.inc.php

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

4.4.2 dbconnection.inc.php

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

4.4.3 pdosettings.inc.php

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

4.4.4 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 Codeteile angepasst.

```
class dbConn
{
    ...
    catch (PDOException $e) {
        /** Database 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 in der dbConn Klasse von John Richardson

4.5 Funktionsseiten

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

4.5.1 functioncontroller.inc.php

Hier befinden sich fast alle Funktionen, die mehrmals in der Applikation verwendet werden. Es handelt 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-Versandfunktion, welche in der sendmail.inc.php zu finden ist.

4.5.2 sendmail.inc.php

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

4.5.2.1 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 der Empfänger mitgegeben. Jedes E-Mail erhält Standardinformationen, wie den Titel des E-Mails. Aufgrund des E-Mailtyp-Parameters wird dann die Art des E-Mails definiert. Diese bestimmt den weiteren Inhalt des Emails und schreibt diesen in Variablen.

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

```
/**  
 * build the whole email from variables  
 */  
$nachricht = '  
<html>  
' . $mailHead .  
<body>  
' . $mailTitel .  
' . $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“ im E-Mail gelangt der Administrator oder Benutzer mit wenigen Klicks zum Ticket.

Ein neues Ticket wurde erfasst
Automatische Ticketbenachrichtigung <samuel.keller@sba.ch>
Gesendet: Mi 13.04.2016 17:18
An: Keller Samuel

Automatische Ticket-Benachrichtigung
Soeben wurde ein neues Ticket von Herr Rolf Keller mit dem Titel "23_RK_Ticket-RK8" erfasst.
Bitte teilen Sie dem Ticket einen Supporter zu.
[Zum Ticket](#)

Weitere Ticketinformationen:
Nummer: 17
Titel: 23_RK_Ticket-RK8
Erfasser: rkeb
Name: Rolf Keller
Aktion: Ticket erfasst

Freundliche Grüsse
Das Administratoren-Team

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

4.6 Ticketverwaltungsseiten

4.6.1 index.php

Die Indexseite ist das Herzstück des Ticketingsystems. Hier sind alle existierenden Tickets zu finden. Beim Öffnen der Ticketübersicht werden alle offenen Tickets angezeigt. Diese können aber gefiltert werden.

4.6.1.1 ticketfilter.inc.php

Ticketübersicht

Kategorie	Status	Ersteller	Supporter	Filtern
<input type="button" value="Alle"/>	<input type="button" value="Alle Aktiven"/>	<input type="button" value="Alle"/>	<input type="button" value="Alle"/>	<input type="button" value="▼"/>

Abbildung: Der Ticketfilter mit den Filteroptionen

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 der Datenbank zu erhalten, wird anstatt einer Zahl das Zeichen „%“ in das SQL Skript geschrieben. Nach dem Zeichen „%“ darf jedes andere Zeichen folgen. Dies führt dazu, dass alle Tickets zurückgegeben werden. Die Sonderregelung für den Ticketstatus ist nicht einfach zu lösen. Die Angabe „Alle Aktiven“ schliesst nur die gelöschten, geschlossenen und archivierten Tickets aus der Sortierung aus. Die Filterung muss somit mehr als nur einen Wert aus der Datenbank abfragen. Mit der Hilfe des Users Sammaye von stackoverflow.com, konnte ein PDO IN()-Statement erstellt werden, welches man mit einem Array befüllen kann. So kann genau vorgeben werden, welche der Stati bei der Option „Alle Aktiven Tickets“ aufrufen werden sollen.

Dieser Codeteil zeigt das IN()-Statement in der Query.

```
...
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()

Mit diesem Code werden die Werte für die Kriterien Kategorie, Ersteller und Supporter auf „Alle anzeigen“ gesetzt. Das Statuskriterium erhielt eine Sonderregelung.

```
/**
 * 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 = "%";
}
```

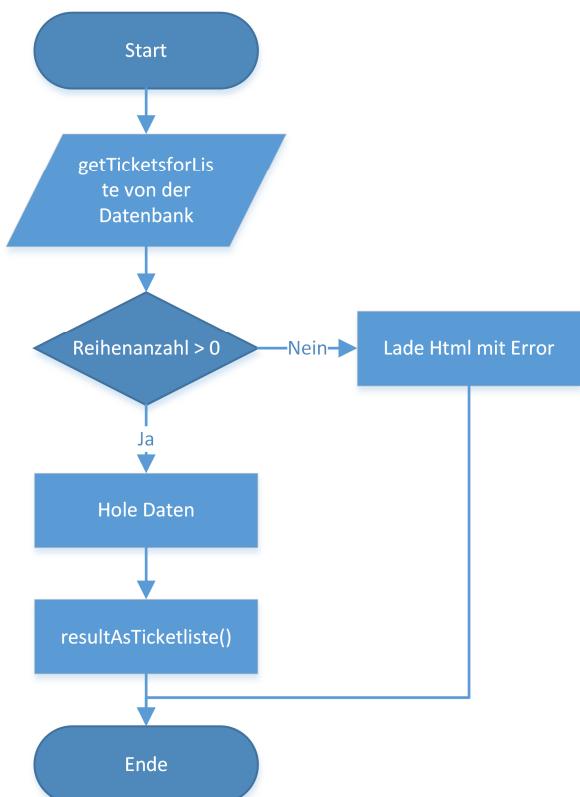
Abbildung: Codeausschnitt aus der Ticketfilter

4.6.2 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 Tickets 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



4.6.3 neuesticket.php

Neues Ticket

Titel *

Thema *

Beschreibung *

Endtermin

Datei Keine Datei ausgewählt.

Abbildung: Ticket Eingabemaske

Auf der Seite „neuesticket.php“ können Benutzer des Ticketingsystems neue Tickets erstellen. Damit das Ticket erfolgreich erstellt werden kann, muss mindestens ein Titel, ein passendes Thema und eine Beschreibung angegeben werden. Das Ticket kann nicht ohne die Eingabe dieser Werte validiert werden.

Zusätzlich zu den erforderlichen Eingaben kann eine Deadline für das Ticket angegeben werden. Diese Deadline kann nicht vor dem Erstellungsdatum des Tickets liegen. Die Eingabe des Datums wird nicht manuell getätigt, sondern wird über das Bootstrap 3 Plug-In „Datepicker“ von Eonasdan gemacht. Das erleichtert die Eingabe.

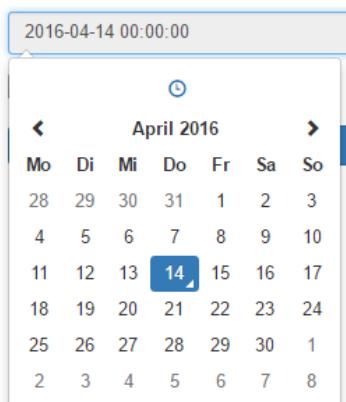


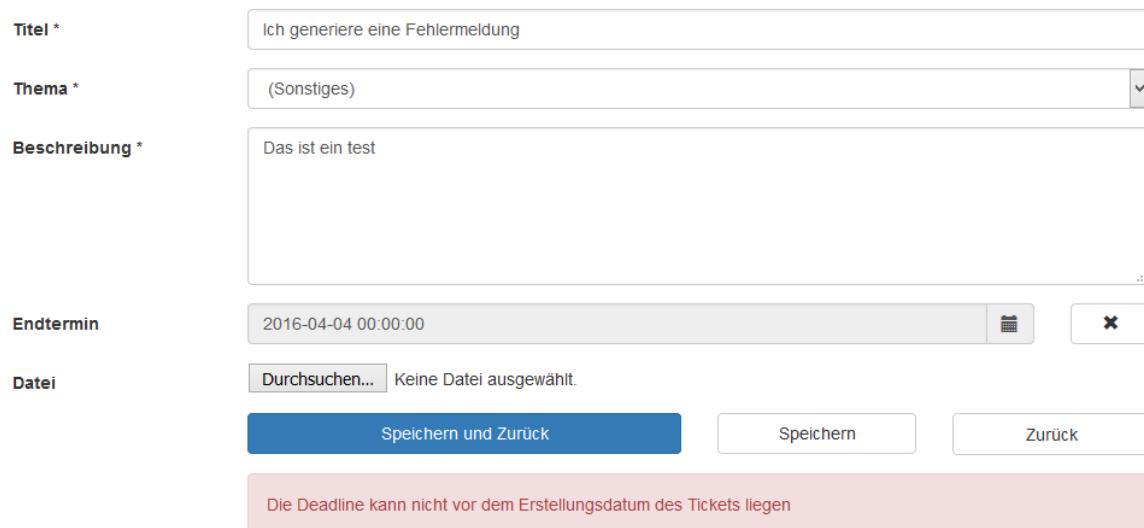
Abbildung: Plug-In „Datepicker“

Dem Ticket kann auch eine Datei beigefügt werden. Diese muss vom Typ JPEG, GIF, PNG, ZIP oder PDF sein. Die Datengröße darf 5MB nicht überschreiten.

Das Ticket wird zuerst auf die Deadline überprüft. Ist diese in Ordnung, wird als nächstes die, falls angehängte, Datei überprüft. Wird auch nur eine Fehlermeldung ausgegeben, wird das Ticket nicht erstellt und der Ticketerfasser wird auf die Fehler hingewiesen. Das Ticket wird über mehrere Funktionen erstellt, die Functioncontroller liegen.

Folgend Beispiele der Fehlermeldungen:

Neues Ticket



The screenshot shows a form titled 'Neues Ticket' with the following fields:

- Titel ***: Ich generiere eine Fehlermeldung
- Thema ***: (Sonstiges)
- Beschreibung ***: Das ist ein test
- Endtermin**: 2016-04-04 00:00:00
- Datei**: Durchsuchen... Keine Datei ausgewählt.
- Buttons**: Speichern und Zurück (highlighted in blue), Speichern, Zurück

A red error message box at the bottom states: 'Die Deadline kann nicht vor dem Erstellungsdatum des Tickets liegen' (The deadline cannot be earlier than the creation date of the ticket).

Abbildung: Eine Fehlermeldung, die durch die Eingabe eines falschen Datums ausgelöst wurde.

Upload Fehler:
· Es sind nur die Formate JPG, JPEG, PNG, GIF, PDF und ZIP erlaubt.

Abbildung: Eine Fehlermeldung, die durch einen falschen Dateityp ausgelöst wurde.

Über den Kopf „Speichern“ wird der Ticketerfasser, falls keine Fehlermeldung ausgelöst wird, zu der Detailübersicht des Tickets weitergeleitet. Nach der Erstellung wird ein E-Mail an die Web Abteilung mit dem Hinweis, dass man dem Ticket ein Supporter zuweisen soll, gesendet. Wenn eine Fehlermeldung entsteht, wird das Ticket nicht erfasst.

Bevor das Ticket erstellt gespeichert wird, muss es Validiert werden.

```

if ($_SERVER[ 'REQUEST_METHOD' ] === 'POST')
{
    $ticketTitle = $_POST[ "ticket-title" ];
    $ticketkategorieid = $_POST[ "ticket-thema" ];
    $ticketDesc = $_POST[ "ticket-beschreibung" ];
    $ticketDeadline = $_POST[ "ticket-endtermin" ];
    $ticketTitle = strip_tags($ticketTitle);
    $ticketDesc = strip_tags($ticketDesc);
    $currentDate = date('Y-m-d H:i:s');
    $checkDeadline = $_POST[ "ticket-endtermin" ];
    $dateOk = 1;

    if(empty($checkDeadline)) {
        $dateOk = 1;
    }
    elseif ($checkDeadline < $currentDate) {
        $dateError = "<div class='alert alert-danger'>Die Deadline
                    kann nicht vor dem Erstellungsdatum des Tickets
                    liegen</div>";
        $dateOk = 0;
    }
    if ($dateOk == 1) {
        $con=dbConn::getConnection();
        if(isset($_POST[ 'save-ticket' ]))
        {
            $ticketError = createNewTicket($con, $_FILES);
            if (empty($ticketError)) {
                $stmt = $con->prepare('SELECT * FROM ttickets
                WHERE cOwnerID=:ticketowner');
                $stmt->bindParam(':ticketowner', $userid,
                PDO::PARAM_STR);
                $last_ticketid = $con->lastInsertId();
                $stmt->execute();
                header("Location: ticket-
                details.php?id=$last_ticketid");
            }
        }
        if(isset($_POST[ 'save-goback' ])) {
            $ticketError = createNewTicket($con, $_FILES);
            if (empty($ticketerror)) {
                header("Location: index.php");
            }
        }
    }
}

```

Abbildung des Codes, welcher das Ticket Validiert und an die Ticketerstellungsfunktionen weiter gibt.

Zusätzlich zu der Validierung der Datei, wird diese zu einem Hash mit Salt umbenannt und so in den lokalen Order „uploads“ gespeichert. Die Kombination von Zahlen und Buchstaben sagen dem Benutzer nichts über den Inhalt der Datei aus, weshalb der originale Dateiname zusammen mit dem Dateipfad in der Datenbank abgespeichert wird. So kann beim Ticket trotzdem der richtige Dateiname angegeben werden.

•SwissBanking

Die Datei kann nicht mit dem Originalnamen abgespeichert werden. Ansonsten können Dateinamen nur einmal benutzt werden, was nervig und nicht benutzerfreundlich ist. Deswegen wird der Name der Datei vor dem Speichern geändert.

```
function checkFile($con, $datei)
{
    /** variables for check */
    $ticketerror = "";
    $fileerror = "";
    $target_dir = "uploads/";
    $filename = $datei["fileToUpload"]["name"];
    $file_ext = substr($filename, strpos($filename, '.')); // get
    file name
    $file_ext = strtolower($file_ext);

    /** hash filename */
    $hashfilename = md5($filename);

    /** generate salt */
    $salt = mt_rand();

    $target_file = $target_dir.$salt.$hashfilename.$file_ext;
    $imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);
    $imageFileType = strtolower($imageFileType);
    $uploadOk = 1;
    ...
}
```

Abbildung: Codeausschnitt der Funktion checkFile(), welche die Datei umbennt und überprüft.

Zuerst wird der Speicherort der Datei gewählt. Danach wird die Dateiendung vom Dateinamen getrennt. Aus dem Dateinamen (ohne Dateiendung) wird ein md5 Hash generiert. Dieser wird zusammen mit einer zufallsgenerierten Zahl in den abzuspeichernden Dateinamen geschrieben.

```
function saveFile($target_file, $con, $datei, $filename)
{
    /** moves file */
    if (move_uploaded_file($datei["fileToUpload"]["tmp_name"], $target_file))
    {
        /** saves file to database */
        fileToDb($con, $target_file, $filename);
    }
    else
    {
        /** error */
        $ticketerror = '<div class="alert alert-danger">Es gab einen Fehler
        beim Upload der Datei';
    }
}
```

Abbildung: Codeausschnitt der Funktion saveFile(), welche die Datei abspeichert und überprüft.

Die Funktion saveFile() verschiebt über den Befehl “move_uploaded_file (\$datai[“fileToUpload”][“tmp_name”], \$target_file)” die Datei in den “uploads” Ordner und nennt die Datei in den generierten Namen um.

```
function createTicket($con) {  
    /** variables */  
    $tickettitle = $_POST[ "ticket-title" ];  
    $ticketkategorieid = $_POST[ "ticket-thema" ];  
    $ticketdesc = $_POST[ "ticket-beschreibung" ];  
    $ticketdeadline = $_POST[ "ticket-endtermin" ];  
    $owner = $_SESSION[ "user_id" ];  
    $ticketusername = $_SESSION[ "user_username" ];  
    $ticketnachname = $_SESSION[ "user_nachname" ];  
    $ticketvorname = $_SESSION[ "user_vorname" ];  
  
    $stmt = $con->prepare('INSERT INTO ttickets SET cTicket-  
Title=:tickettitle, cKategorieID=:kategorieid, cTicket-  
beschreibung=:ticketdesc, cTicketDeadline=:ticketdeadline, cOwn-  
erID=:ticketowner');  
    $stmt->bindParam(':tickettitle', $tickettitle, PDO::PARAM_STR);  
    $stmt->bindParam(':kategorieid', $ticketkategorieid,  
PDO::PARAM_STR);  
    $stmt->bindParam(':ticketdesc', $ticketdesc, PDO::PARAM_STR);  
    $stmt->bindParam(':ticketdeadline', $ticketdeadline,  
PDO::PARAM_STR);  
    $stmt->bindParam(':ticketowner', $owner, PDO::PARAM_STR);  
    $stmt->execute();  
    $last_ticketid = $con->lastInsertId();  
    sendMail($last_ticketid, $tickettitle, $ticketusername, "Neues  
Ticket", $ticketvorname, $ticketnachname, "", "", "ske@sba.ch");  
};
```

Abbildung: Die Funktion createTicket(), welche das Ticket in die Datenbank speichert und den Mail-Versand auslöst.

Diese Funktion speichert nur die Ticketinformationen ab. Die hinzugefügte Datei wird separat gespeichert. Da die Ticketinformationen nur ausgelesen werden, wenn Sie bereits validiert worden sind, können die Informationen aus dem Post-back der Ticketseite genommen werden. Zusätzlich werden Informationen aus der Session geladen. Dann wird das Ticket in die Datenbank gespeichert und das Mail versendet.

4.6.4 ticket_details.php

Ticket Detailübersicht

ID	17
Titel *	23_RK_Ticket-RK8
Thema *	(Sonstiges)
Beschreibung *	falsche Deadline eingeben
Erfasser	rkeb
Supporter *	Nicht Zugewiesen
Erstellungsdatum	2016-04-13 17:18:22
Endtermin	2016-04-16 00:00:00
Last Change	2016-04-14 11:43:45
Status *	Offen
Datei	Keine Datei vorhanden

Speichern und Zurück Speichern Zurück

Abbildung: Administratorenansicht der Ticketdetails

Die Detailübersicht zeigt alle Informationen eines Tickets an. Hier kann ein Administrator die Ticketdaten verändern.

Durch die Zuweisung eines Administrators und der Veränderung des Ticketstatus, wird ein erneuter E-Mail versand an Supporter und Ticketerfasser ausgelöst. So kann der Ticketerfasser immer auf dem aktuellen Status des Tickets gehalten werden.

4.7 Userverwaltung

4.7.1 profil.php

Profil

Vorname	<input type="text" value="Samuel"/>
Nachname	<input type="text" value="Keller"/>
E-Mail Adresse *	<input type="text" value="samue.keller@sba.ch"/>
Telephonnummer	<input type="text" value="0000000000000000"/>
Handynummer	<input type="text" value="00410000000000"/>
Passwort ändern:	
Neues Passwort	<input type="text"/>
<input type="button" value="Speichern"/> <input type="button" value="Zurück"/>	

Im Profil kann der 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, dass kein Username und keine E-Mail-Adresse zweimal verwendet werden darf. Vorname und Nachname dürfen 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 geprüft. Falls die eingegebenen Daten nicht den Kriterien entsprechen, werden Fehlermeldungen erstellt. Alle Fehlermeldungen werden in eine Variable gesammelt und zusammen ausgegeben. Nur wenn die Validierung korrekt durchgeführt worden ist, wird das Profil aktualisiert.

```
/**  
 * checks if 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

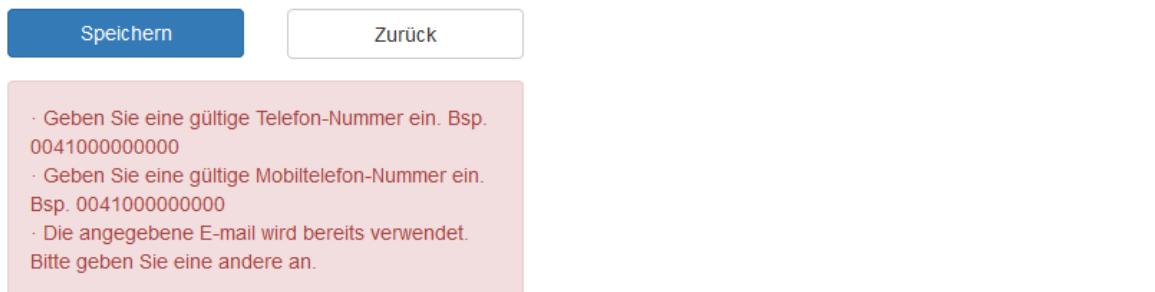


Abbildung: Eine Fehlermeldung, ausgelöst durch die Eingabe einer bereits verwendeten E-Mail-Adresse und falscher Telefon- und Handynummer.

4.7.2 usermanagement.php

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

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

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

Usermanager

The screenshot shows a table titled 'Usermanager' with the following columns: ID, Username, Name, and Rolle. At the top right are buttons for 'User erfassen' (Create User) and '+ User erfassen' (Create User). The table data is as follows:

ID	Username	Name	Rolle	
1	ske	Samuel Keller	Administrator	<button>Bearbeiten</button>
2	rke	Rolf Keller	Administrator	<button>Bearbeiten</button>
3	rkeb	Rolf Keller	Benutzer	<button>Bearbeiten</button>
4	jar	Jonas Arab	Administrator	<button>Bearbeiten</button>
5	jarb	Hans Muster	Benutzer	<button>Bearbeiten</button>
8	jet	joe tester jetzt als user	Benutzer	<button>Bearbeiten</button>
9	JoArax	Johannesx Araberx	Administrator	<button>Bearbeiten</button>

Abbildung: Die Benutzerübersicht im Usermanager

4.7.3 neuernutzer.php

Neuer Nutzer

Vorname *	<input type="text"/>
Nachname *	<input type="text"/>
Username *	<input type="text"/>
E-Mail Adresse *	<input type="text"/>
Telephonnummer	0041000000000
Handynummer	0041000000000
Passwort *	<input type="password"/>
Rolle *	<input type="text"/>

Speichern **Zurück**

Über diese Eingabemaske kann ein Administrator einen neuen Benutzer für das Ticke-
tingsystem erstellen. Ohne Eingabe der Pflichtfelder kann kein Benutzer erstellt wer-
den. Die Validierung der eingegebenen Daten wird über die gleiche Validierungsfunkti-
on wie das Profil abgewickelt. Somit kann die E-Mail und der Benutzernamen nicht
bereits vorhanden sein, der Vor- und Nachname darf nicht länger als 32 Zeichen lang
sein. Ebenfalls muss die E-Mail korrekt sein. Zusätzlich muss dem Benutzer erstmalig
ein Passwort zugewiesen werden.

4.7.4 benutzer_details.php

In der Benutzerdetailansicht können die Administratoren alle Benutzerdaten ändern. So
kann einem Benutzer ein anderer Usernamen gegeben werden, was der Benutzer
selbst nicht machen kann. Falls ein Benutzer das Passwort vergessen hat, kann über
diese Funktion ein neues vergeben werden.

Fall ein User nicht mehr benötigt wird, kann dieser über den Knopf „User deaktivieren“
deaktiviert werden. Ein deaktiverter Nutzer kann sich nicht mehr Einloggen und wird
nicht mehr in der Userübersicht angezeigt.

Samuel Keller's Profil

ID	1
Vorname	Samuel
Nachname	Keller
Username *	ske
E-Mail Adresse *	samue.keller@sba.ch
Telephonnummer	00000000000000
Handynummer	0041000000000
Rolle *	Administrator
Neues Passwort	
<input style="background-color: #0072BD; color: white; padding: 5px 20px; border: none; border-radius: 5px; width: 150px; height: 30px; margin-right: 10px;" type="button" value="Speichern"/> <input style="border: 1px solid #ccc; padding: 5px 20px; border-radius: 5px; width: 150px; height: 30px;" type="button" value="Zurück"/>	
<input style="background-color: #E74C3C; color: white; padding: 5px 20px; border: none; border-radius: 5px; width: 150px; height: 30px;" type="button" value="User deaktivieren"/>	

Abbildung der Detailansicht eines Benutzers

```

if(isset($_POST['deactivate-user']))
{
    $stmt = mysqli_stmt_init($conn);
    mysqli_stmt_prepare($stmt, "UPDATE tbenutzer SET cAktiv=? WHERE
cBenutzerID=?");
    mysqli_stmt_bind_param($stmt, "ii", $notactive, $detailId);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_free_result($stmt);
    mysqli_stmt_close($stmt);
    header('Location: usermanagement.php');
}

```

Abbildung des Codeteils, mit dem man einen Benutzer deaktivieren kann.

5 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. Dank der Nutzung von Bootstrap sind keine Funktions- oder Darstellungsprobleme entstanden.

5.1 Endresultat

Wie auf den Testprotokollen zu sehen ist, gab es fast keine Probleme mit der Applikation. Alle Testfälle bis auf den Testfall #24 konnten „Wie erwartet“ abgeschlossen werden. Dieser Fehler kann in einer Version 2.0 oder einem Patch behoben werden.

5.2 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.sbgv.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.sbgv.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.sbgv.eu/usermanager.php http://ipa.sbgv.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	<p>Für das Speichern der Profil Daten muss jede Eingabe korrekt validiert werden. Die Eingabemaske gibt einen Fehler aus, wenn:</p> <ul style="list-style-type: none"> - 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	<p>Für das Speichern der Profil Daten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:</p> <ul style="list-style-type: none"> - 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	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.

#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 eingegebenen 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: <ul style="list-style-type: none"> - 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, Username, 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	<p>Für das aktualisieren der Benutzerdaten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:</p> <ul style="list-style-type: none"> - 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 benutzte E-Mail-Adresse angegeben wurde, - die Telefonnummern nicht dem angezeigten Format entsprechen.
Test	<p>Mit einem Klick auf den Button „Bearbeiten“ im Usermanager beim zu editierenden User wird dieser in der Detailübersicht geöffnet.</p> <p>Geben Sie falsche Daten in Vorname, Nachname, Usernamen, E-Mail, Telefonnummern ein.</p> <p>Rolle entweder „Benutzer“ oder „Administrator“.</p> <p>Profil speichern.</p>
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	<p>Für das aktualisieren der Benutzerdaten muss jede Eingabe korrekt sein. Die Eingabemaske gibt einen Fehler aus, wenn:</p> <ul style="list-style-type: none"> - 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	<p>Mit einem Klick auf den Button „Bearbeiten“ im Usermanager beim zu editierenden User wird dieser in der Detailübersicht geöffnet.</p> <p>Geben Sie korrekte Daten in Vorname, Nachname, Usernamen, E-Mail, Telefonnummern ein.</p> <p>Rolle entweder „Benutzer“ oder „Administrator“.</p> <p>Profil speichern.</p>
Erwartetes Resultat	Das Profil wird aktualisiert.

5.3 Testprotokolle

5.3.1 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.sbgv.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.sbgv.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.sbgv.eu/neuesticket.php)

•SwissBanking

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

The screenshot shows the SwissBanking Admin interface. At the top, there is a header with the logo, a 'Ticketübersicht' link, a 'Ticket erfassen' button, and user navigation links for 'Hilfe', 'Rolf Keller', 'Usermanager', and 'Log-Out'. Below the header, the main title 'Ticketübersicht' is displayed. On the right side of the header, there is a link 'Neuer Nutzer Übersicht'.

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

The screenshot shows the SwissBanking User interface. At the top, there is a header with the logo, a 'Ticketübersicht' link, a 'Ticket erfassen' button, and user navigation links for 'Hilfe', 'Rolf Keller', and 'Log-Out'. Below the header, the main title 'Ticketübersicht' is displayed.

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

The screenshot shows the 'Ticketübersicht' (Ticket Overview) page. At the top, there are several filter dropdowns: 'Kategorie' (Alle), 'Status' (Offen), 'Ersteller' (Alle), 'Supporter' (Alle), and a 'Filtern' button. Below the filters, there is a table with columns: 'ID', 'Titel', 'Kategorie', 'Erfasser', 'Supporter', 'Status', and 'Deadline'. One row is visible, showing ID 9, Titel 'Test-Ticket', Kategorie 'Grundbildung SwissBanking Future', Erfasser 'ske', Supporter 'Nicht Zugewiesen', Status 'Offen', and Deadline. A 'Ansehen' (View) button is at the bottom right of the table.

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.

•SwissBanking

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)

•SwissBanking

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.

•SwissBanking

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.

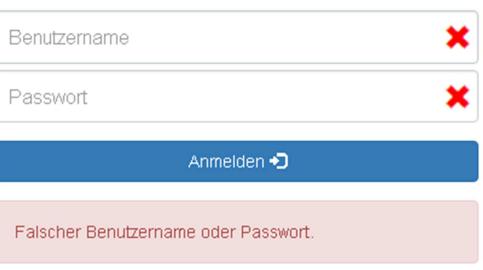
5.3.2 Testprotokoll #2

Tester: Jonas Arab

Testmittel:

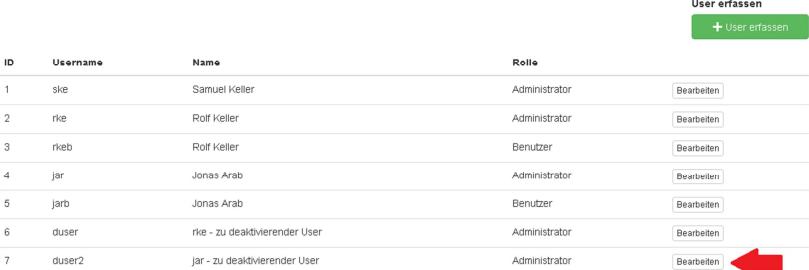
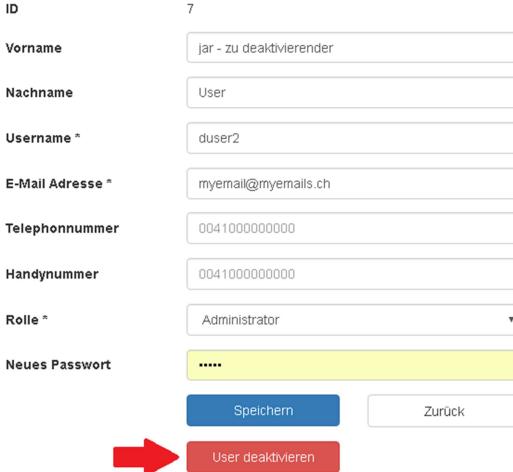
Windows 7 Pro

Google Chrome (Version 49.0.2623.112m)

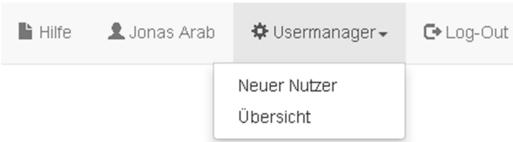
#01	Login mit falschen Daten
Testdaten	Benutzer: invalid; Passwort: abc123
Aktionen	<p>Die oben genannten Daten wurden in die jeweiligen Textboxen eingegeben.</p> <p>Log-In</p> 
Resultat	<p>Es wird eine Fehlermeldung ausgegeben.</p> <p>Log-In</p> 

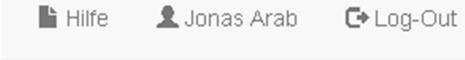
#02	Zugriff auf die Applikation ohne Login
Testdaten	URL: ipa.sbgv.eu/index.php
Aktionen	Es wurde versucht, per Link auf die Detailübersicht zuzugreifen ohne Login
Resultat	Die Applikation verweist auf den Login-Screen.

#03	Login mit richtigen Daten
Testdaten	Benutzer: jar; Passwort: admin
Aktionen	<p>Die oben genannten Daten wurden in die jeweiligen Textboxen eingegeben.</p> <p>Log-In</p> 
Resultat	Die Applikation verweist auf die Ticketübersicht.

#04	Deaktivieren eines Benutzers durch den Administrator in der Userdetailübersicht.
Testdaten	Benutzer „duser2“ in der Benutzerübersicht
Aktionen	<p>Als Administrator wurde im Usermanager das Profil „duser2“ bearbeitet.</p> <p>Usermanager</p>  <p>Anschliessend wird auf „User deaktivieren“ geklickt</p> <p>jar - zu deaktivierender User's Profil</p> 
Resultat	Der User ist nicht mehr im Usermanager sichtbar.

#05	Login mit übergebenen URL
Testdaten	URL: http://ipa.sbgv.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. 

#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. 

#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. 

#09	Ticketfilterung																								
Testdaten	Kategorie: SBC; Status: Offen; Ersteller: ske; Supporter: Nicht Zugewiesen																								
Aktionen	Die oben genannten Kriterien wurden selektiert.																								
Resultat	<p>Die Ticketübersicht zeigt nun das gefilterte Resultat an.</p> <p>Ticketübersicht</p> <table border="1"> <thead> <tr> <th>Kategorie</th> <th>Status</th> <th>Ersteller</th> <th>Supporter</th> <th>Filtern</th> </tr> </thead> <tbody> <tr> <td>SBC</td> <td>Offen</td> <td>ske</td> <td>Nicht Zugewiesen</td> <td><input type="button" value="▼"/></td> </tr> <tr> <th>ID</th> <th>Titel</th> <th>Kategorie</th> <th>Erfasser</th> <th>Supporter</th> <th>Status</th> <th>Deadline</th> </tr> <tr> <td>1</td> <td>Test-Ticket ohne Deadline und ohne Datei</td> <td>SBC</td> <td>ske</td> <td>Nicht Zugewiesen</td> <td>Offen</td> <td><input type="button" value="Ansehen"/></td> </tr> </tbody> </table>	Kategorie	Status	Ersteller	Supporter	Filtern	SBC	Offen	ske	Nicht Zugewiesen	<input type="button" value="▼"/>	ID	Titel	Kategorie	Erfasser	Supporter	Status	Deadline	1	Test-Ticket ohne Deadline und ohne Datei	SBC	ske	Nicht Zugewiesen	Offen	<input type="button" value="Ansehen"/>
Kategorie	Status	Ersteller	Supporter	Filtern																					
SBC	Offen	ske	Nicht Zugewiesen	<input type="button" value="▼"/>																					
ID	Titel	Kategorie	Erfasser	Supporter	Status	Deadline																			
1	Test-Ticket ohne Deadline und ohne Datei	SBC	ske	Nicht Zugewiesen	Offen	<input type="button" value="Ansehen"/>																			

#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	<p>Die Applikation zeigt eine Fehlermeldung und verweist auf die Ticketübersicht.</p> <p>•SwissBanking</p> <p>Keine Berechtigung für diese Seite</p> <p>Bitte melden Sie sich an.</p>

•SwissBanking

#12	Individuelles Benutzerprofil
Testdaten	Benutzer: jarb; Passwort: benutzer
Aktionen	Auf der Navigation wurde auf den Benutzernamen geklickt.
Resultat	<p>Das Benutzerprofil wurde aufgerufen.</p> <p>Profil</p> <p>Vorname <input type="text" value="Jonas"/></p> <p>Nachname <input type="text" value="Arab"/></p> <p>E-Mail Adresse * <input type="text" value="jemandhalt@yahoo.de"/></p> <p>Telefonnummer <input type="text" value="0041000000000"/></p> <p>Handynummer <input type="text" value="0041000000000"/></p> <p>Passwort ändern:</p> <p>Neues Passwort <input type="text"/></p> <p><input type="button" value="Speichern"/> <input type="button" value="Zurück"/></p>

#13	Individuelles Benutzerprofil
Testdaten	Vorname: Ai Nachname: 33x „A“ Telefonnummer: 123 Mobilnummer: 456
Aktionen	<p>Es wurden ungültige Eingaben in der Benutzerbearbeitung eingegeben.</p> <p>Profil</p> <p>Vorname <input type="text" value="Ai"/></p> <p>Nachname <input type="text" value="aaa"/></p> <p>E-Mail Adresse * <input type="text" value="jemandhalt@yahoo.de"/></p> <p>Telefonnummer <input type="text" value="123"/></p> <p>Handynummer <input type="text" value="456"/></p>
Resultat	<p>Für die jeweiligen Felder gibt die Applikation eine Fehlermeldung aus.</p> <ul style="list-style-type: none"> · Der Vorname ist zu kurz. Er muss mindestens 3 Zeichen enthalten. · Der Nachname ist zu lang. Er darf maximal 32 Zeichen enthalten. · Geben Sie eine gültige E-mail Adresse an ein. · Geben Sie eine gültige Telefon-Nummer ein. Bsp. 004100000000 · Geben Sie eine gültige Mobiltelefon-Nummer ein. Bsp. 0041000000000

#14	Benutzerprofil aktualisieren
Testdaten	<p>Vorname: Hans Nachname: Muster E-Mail: jemandhalt@yahoo.de Telefonnummer: 0041614823216 Handynummer: 0041764083216</p>
Aktionen	<p>Die oben genannten Daten wurden in die jeweiligen Textfelder eingegeben.</p> <p>Profil</p> <p> Vorname <input type="text" value="Hans"/> Nachname <input type="text" value="Muster"/> E-Mail Adresse * <input type="text" value="jemandhalt@yahoo.de"/> Telefonnummer <input type="text" value="0041614823216"/> Handynummer <input type="text" value="0041764083216"/> </p>
Resultat	<p>Die Daten werden ohne Fehler übernommen. (Überprüfung mittels Admin)</p> <p>Hans Muster's Profil</p> <p> ID 5 Vorname <input type="text" value="Hans"/> Nachname <input type="text" value="Muster"/> Username * <input type="text" value="jarb"/> E-Mail Adresse * <input type="text" value="jemandhalt@yahoo.de"/> Telefonnummer <input type="text" value="0041614823216"/> Handynummer <input type="text" value="0041764083216"/> Rolle * <input type="text" value="Benutzer"/> </p>

#15	Detailansicht des Tickets als Benutzer																						
Testdaten	Benutzer: jarb; Passwort: benutzer																						
Aktionen	Auf der Ticketübersicht wurde bei einem Test-Ticket auf „Ansehen“ geklickt.																						
	<p>9 Test-Ticket Grundbildung SwissBanking Future ske Nicht Zugewiesen Offen Ansehen</p>																						
Resultat	<p>Der Benutzer wird auf die Detailübersicht weitergeleitet.</p> <p>Ticket Detailübersicht</p> <table> <tr> <td>ID</td><td>9</td></tr> <tr> <td>Titel</td><td>Test-Ticket</td></tr> <tr> <td>Thema</td><td>Grundbildung SwissBanking Future</td></tr> <tr> <td>Beschreibung</td><td> <p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p> </td></tr> <tr> <td>Erfasser</td><td>ske</td></tr> <tr> <td>Supporter</td><td>Nicht Zugewiesen</td></tr> <tr> <td>Erstellungsdatum</td><td>2016-04-13 13:24:06</td></tr> <tr> <td>Endtermin</td><td></td></tr> <tr> <td>Last Change</td><td>2016-04-13 13:26:20</td></tr> <tr> <td>Status</td><td>Offen</td></tr> <tr> <td>Datei</td><td>Keine Datei vorhanden</td></tr> </table> <p>Zurück</p>	ID	9	Titel	Test-Ticket	Thema	Grundbildung SwissBanking Future	Beschreibung	<p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p>	Erfasser	ske	Supporter	Nicht Zugewiesen	Erstellungsdatum	2016-04-13 13:24:06	Endtermin		Last Change	2016-04-13 13:26:20	Status	Offen	Datei	Keine Datei vorhanden
ID	9																						
Titel	Test-Ticket																						
Thema	Grundbildung SwissBanking Future																						
Beschreibung	<p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p>																						
Erfasser	ske																						
Supporter	Nicht Zugewiesen																						
Erstellungsdatum	2016-04-13 13:24:06																						
Endtermin																							
Last Change	2016-04-13 13:26:20																						
Status	Offen																						
Datei	Keine Datei vorhanden																						
#16	Detailansicht des Tickets als Administrator																						
Testdaten	Benutzer: jar, Passwort: admin																						
Aktionen	Auf der Ticketübersicht wurde bei einem Test-Ticket auf „Ansehen“ geklickt.																						
	<p>9 Test-Ticket Grundbildung SwissBanking Future ske Nicht Zugewiesen Offen Ansehen</p>																						
Resultat	<p>Der Admin wird auf die Detailübersicht weitergeleitet und kann zusätzlich die Daten bearbeiten.</p> <p>Ticket Detailübersicht</p> <table> <tr> <td>ID</td><td>9</td></tr> <tr> <td>Titel *</td><td><input type="text" value="Test-Ticket"/></td></tr> <tr> <td>Thema *</td><td><input type="text" value="Grundbildung SwissBanking Future"/></td></tr> <tr> <td>Beschreibung *</td><td> <p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p> </td></tr> <tr> <td>Erfasser</td><td>ske</td></tr> <tr> <td>Supporter *</td><td><input type="text" value="Nicht Zugewiesen"/></td></tr> <tr> <td>Erstellungsdatum</td><td><input type="text" value="2016-04-13 13:24:06"/></td></tr> <tr> <td>Endtermin</td><td><input type="text" value="YYYY-MM-DD HH:MM:SS"/></td></tr> <tr> <td>Last Change</td><td>2016-04-13 13:26:20</td></tr> <tr> <td>Status *</td><td><input type="text" value="Offen"/></td></tr> <tr> <td>Datei</td><td>Keine Datei vorhanden</td></tr> </table> <p>Speichern und Zurück Speichern Zurück</p>	ID	9	Titel *	<input type="text" value="Test-Ticket"/>	Thema *	<input type="text" value="Grundbildung SwissBanking Future"/>	Beschreibung *	<p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p>	Erfasser	ske	Supporter *	<input type="text" value="Nicht Zugewiesen"/>	Erstellungsdatum	<input type="text" value="2016-04-13 13:24:06"/>	Endtermin	<input type="text" value="YYYY-MM-DD HH:MM:SS"/>	Last Change	2016-04-13 13:26:20	Status *	<input type="text" value="Offen"/>	Datei	Keine Datei vorhanden
ID	9																						
Titel *	<input type="text" value="Test-Ticket"/>																						
Thema *	<input type="text" value="Grundbildung SwissBanking Future"/>																						
Beschreibung *	<p> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p>																						
Erfasser	ske																						
Supporter *	<input type="text" value="Nicht Zugewiesen"/>																						
Erstellungsdatum	<input type="text" value="2016-04-13 13:24:06"/>																						
Endtermin	<input type="text" value="YYYY-MM-DD HH:MM:SS"/>																						
Last Change	2016-04-13 13:26:20																						
Status *	<input type="text" value="Offen"/>																						
Datei	Keine Datei vorhanden																						

#17	Verändern des Tickets in der Detailansicht durch den Administrator
Testdaten	<p>Titel: 123 Thema: One Voice Beschreibung: Neue Beschreibung Supporter: jar Status: In Beobachtung</p>
Aktionen	<p>Die oben genannten Daten werden in die jeweiligen Felder eingegeben.</p> <p>Ticket Detailübersicht</p> <p>ID: 9 Titel: 123 Thema: One Voice Beschreibung: Neue Beschreibung Erfasser: ske Supporter: jar Erstellungsdatum: 2016-04-13 13:24:06 Endtermin: YYYY-MM-DD HH:MM:SS Last Change: 2016-04-13 13:26:20 Status: In Beobachtung Datei: Keine Datei vorhanden <input type="button" value="Speichern und Zurück"/> <input type="button" value="Speichern"/> <input type="button" value="Zurück"/></p>
Resultat	Die Daten werden korrekt übernommen.

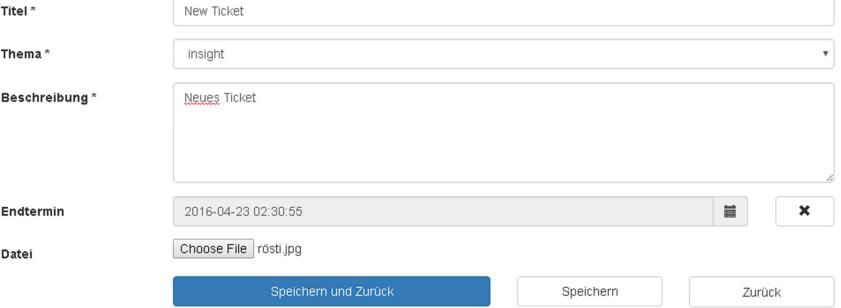
Testfall-Nummer (#18)	Verändern des Ticketstatus in der Detailansicht durch den Administrator
Testdaten	Status: In Bearbeitung
Aktionen	Der Ticketstatus „In Bearbeitung“ wurde aus dem Dropdown-Menü selektiert.
Resultat	Der Status wird korrekt übernommen.

#19	Verändern des Supporter in der Detailansicht durch den Administrator
Testdaten	Supporter: rke
Aktionen	Der Supporter „rke“ wurde aus dem Dropdown-Menü selektiert.
Resultat	Der Supporter wird korrekt übernommen.

#20	Ticket erfassen
Testdaten	-
Aktionen	Auf der Navigation wurde auf „Ticket erfassen“ geklickt
Resultat	<p>Der „Neues Ticket“-Dialog wird aufgerufen.</p> <p>Neues Ticket</p> <p>Titel * <input type="text"/></p> <p>Thema * <input type="text"/>(Sonstiges)</p> <p>Beschreibung * <input type="text"/></p> <p>Endtermin <input type="text"/> YYYY-MM-DD HH:MM:SS <input type="button"/> <input type="button"/></p> <p>Datei <input type="button"/> Choose File No file chosen</p> <p><input type="button"/> Speichern und Zurück <input type="button"/> Speichern <input type="button"/> Zurück</p>

#21	Ticket Validierung
Testdaten	-
Aktionen	Es wurde versucht, ein neues Ticket zu erstellen ohne irgendwelche Eingaben zu betätigen.
Resultat	<p>Die Maske weist darauf hin, dass die Felder ausgefüllt werden sollen.</p> <p>Neues Ticket</p> <p>Titel * <input type="text"/></p> <p>Thema * <input type="text"/>(Sonstiges) Please fill out this field.</p> <p>Beschreibung * <input type="text"/></p> <p>Endtermin <input type="text"/> YYYY-MM-DD HH:MM:SS <input type="button"/> <input type="button"/></p> <p>Datei <input type="button"/> Choose File No file chosen</p> <p><input type="button"/> Speichern und Zurück <input type="button"/> Speichern <input type="button"/> Zurück</p>

•SwissBanking

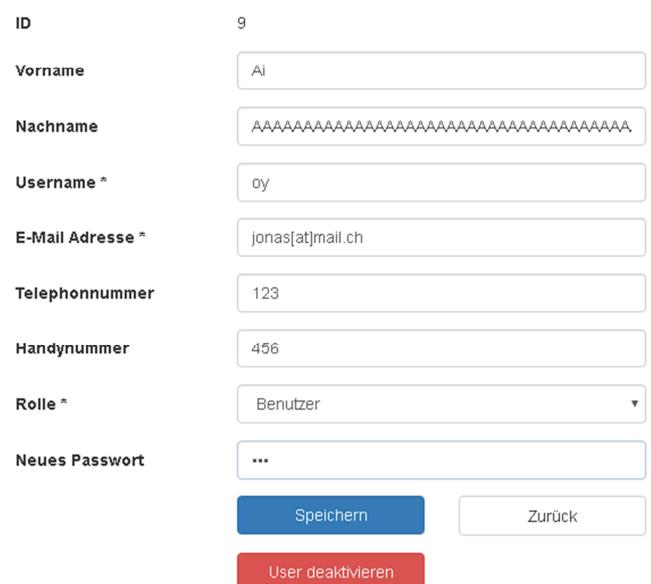
#22	Ticket-Deadline Validierung							
Testdaten	<p>Titel: New Ticket Thema:insight Beschreibung: Neues Ticket Endtermin: 2016-04-23 (02:03:55) Datei: rösti.jpg</p>							
Aktionen	<p>Die obengenannten Daten wurden im „Neues Ticket“-Dialog eingegeben.</p> <p>Neues Ticket</p> 							
Resultat	<p>Das neue Ticket wurde erfolgreich erstellt.</p> <table border="1"> <tr> <td>14 New Ticket</td> <td>insight</td> <td>jar</td> <td>Nicht Zugewiesen</td> <td>Offen</td> <td>2016-04-23 02:30:55</td> <td>Ansehen</td> </tr> </table>	14 New Ticket	insight	jar	Nicht Zugewiesen	Offen	2016-04-23 02:30:55	Ansehen
14 New Ticket	insight	jar	Nicht Zugewiesen	Offen	2016-04-23 02:30:55	Ansehen		

#23	Ticket-Deadline Validierung
Testdaten	Endtermin: 2016-03-24 (00:00:00)
Aktionen	Es wurde versucht, ein Ticket mit bereits vergangenem Endtermin zu erstellen.
Resultat	<p>Die Applikation gibt eine Fehlermeldung aus.</p> <div style="background-color: #f8d7da; padding: 10px; border-radius: 5px; margin-top: 10px;"> <p>Die Deadline kann nicht vor dem Erstellungsdatum des Tickets liegen</p> </div>

#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	<p>Bei „Speichern“ gibt die Applikation eine Fehlermeldung aus.</p> <div style="background-color: #f8d7da; padding: 10px; border-radius: 5px; margin-top: 10px;"> <p>Upload Fehler:</p> <ul style="list-style-type: none"> · Die Datei darf maximal 5MB gross sein. · Es sind nur die Formate JPG, JPEG, PNG, GIF, PDF und ZIP erlaubt. </div> <p>Bei „Speichern und zurück“ gibt die Applikation keine Fehlermeldung aus. Sie verweist lediglich auf die Ticketübersicht zurück. Das Ticket wird jedoch nicht erstellt.</p>

#25	Neuer Nutzer erfassen als Administrator
Testdaten	<p>Vorname:ai Nachname: 33x „A“ Username: Oy E-Mail: jonas[at]mail.ch Telefon: 123 Handy: 456 Passwort: 123 Rolle: Benutzer</p>
Aktionen	<p>Es wurde versucht, als Administrator einen neuen Benutzer mit ungültigen eingaben zu erstellen.</p> <p>Neuer Nutzer</p> <p>Vorname * <input type="text" value="ai"/></p> <p>Nachname * <input type="text" value="AA"/></p> <p>Username * <input type="text" value="Oy"/></p> <p>E-Mail Adresse * <input type="text" value="jonas[at]mail.ch"/></p> <p>Telefonnummer <input type="text" value="123"/></p> <p>Handynummer <input type="text" value="456"/></p> <p>Passwort * <input type="text" value="***"/></p> <p>Rolle * <input type="text" value="Benutzer"/></p> <p>Speichern</p> <p>Zurück</p>
Resultat	<p>Die Applikation gibt für jede ungültige Eingabe eine Fehlermeldung aus.</p> <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;"> <ul style="list-style-type: none"> • Der Vorname ist zu kurz. Er muss mindestens 3 Zeichen enthalten. • Der Nachname ist zu lang. Er darf maximal 32 Zeichen enthalten. • Geben Sie eine gültige E-mail Adresse an ein. • Geben Sie eine gültige Telefon-Nummer ein. Bsp. 004100000000 • Geben Sie eine gültige Mobiltelefon-Nummer ein. Bsp. 004100000000 </div>

•SwissBanking

#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. Johannes Araber's Profil 
Resultat	Es werden keine Änderungen übernommen. Die Applikation gibt eine Fehlermeldung aus. <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;"><ul style="list-style-type: none">• Der Vorname ist zu kurz. Er muss mindestens 3 Zeichen enthalten.• Der Nachname ist zu lang. Er darf maximal 32 Zeichen enthalten.• Geben Sie eine gültige E-mail Adresse an ein.• Geben Sie eine gültige Telefon-Nummer ein. Bsp. 004100000000• Geben Sie eine gültige Mobiltelefon-Nummer ein. Bsp. 004100000000</div>

#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.

•SwissBanking

#28	Veränderung der Benutzerdaten durch den Administrator in der Userdetailübersicht.
Testdaten	Der zuvor erstellte Benutzer „JoAra“
Aktionen	<p>Der Vorname, Nachname, Username und die E-Mail wurden mit einem „x“ ergänzt.“</p> <p>Die letzte Zahl der Telefonnummern wurde mit der Nummer 2 ausgetauscht.</p> <p>Das Passwort wurde auf 456123 umgeändert.</p> <p>Die Rolle wurde von „Benutzer“ auf „Administrator“ umgeändert.</p> <p>Es wurde versucht, die Änderungen zu speichern.</p> <p>Johannes Araber's Profil</p> <p>ID 9</p> <p>Vorname Johannesx</p> <p>Nachname Araberx</p> <p>Username * JoArax</p> <p>E-Mail Adresse * mail@mail.chx</p> <p>Telefonnummer 0041614823212</p> <p>Handynummer 0041764083212</p> <p>Rolle * Administrator</p> <p>Neues Passwort</p> <p>Speichern Zurück</p> <p>User deaktivieren</p>
Resultat	Die Änderungen wurden erfolgreich übernommen.

6 Abschlussbericht

6.1 Rückblick

Rückblickend muss ich sagen, dass der Anfang des Projektes sehr gut verlaufen ist. Jedoch wurde der Zeitmangel in der zweiten Woche nach jedem verstrichenen Tag immer wie offensichtlicher und der Druck war gegen Ende (vor allem am Freitag) kaum auszuhalten. Am Freitag ging es mir körperlich und vor allem psychisch nicht sonderlich gut.

6.2 Lessons learned & Was ist gut gelaufen

Bis zu der Implementation der Applikation lief alles gut und mir hat das Projekt viel Spass bereitet. Ich habe gelernt, dass man bei einem IT-Projekt nicht nur den Fokus auf der Applikation selbst, sondern auch auf dessen Dokumentation haben muss. Dabei muss man die Zeit immer im Blickfeld haben und flexibel sein.

6.3 Was schief gelaufen ist

Ich habe die Zeit, die für die Dokumentation benötigt wird, total unterschätzt und bin gegen Ende der Arbeit in einen extremen Zeitmangel gekommen. Ich konnte bis zu der Implementierung den Zeitplan einhalten, hatte dann aber nicht genügend Zeit um während der Implementation die Dokumentation dazu zu schreiben. Durch die Fokus auf die Applikation, denn man will ja keine fehlerhafte Applikation abliefern, anstatt die Dokumentation, ist ein grosser Zeitmangel entstanden, an welchem die Dokumentation stark. Die Angst der nicht Fertigstellung wurde immer wie grösser. Ich habe mich dann ausschliesslich auf die Dokumentation Fokussiert und die Tagesjournale immer wie mehr vernachlässigt.

6.4 Resultat

Die Applikation ist toll geworden und ich bin stolz auf das erreichte Endresultat. Ich freue mich, die ersten Aufträge über die Applikation bearbeiten zu können und die Applikation mit Benutzerfeedback zu verbessern.

6.5 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.

6.5.1 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 Hash 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.

6.5.2 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 bei den Tickets) kann dies umgangen werden.

6.5.3 Passwort ändern Funktion auf eine eigene Seite auslagern

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

6.5.4 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, der das Ticket bearbeitet, soll dem Ticket einen Text hinzufügen können, mit welchem er dem Ticketersteller antworten kann. Der Kommentar soll per E-Mail auch an den Ticketersteller gesendet werden.

6.6 Schlusswort

Ich bin froh, den ersten Teil der Abschlussprüfung beendet zu haben. Ich habe viel aus meinen Fehlern gelernt und freue mich auf die Präsentation und das Fachgespräch.

7 Quellenverzeichnis

7.1 Versionierung

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

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

7.2 Risikoanalyse

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

- Als Referenz für den Text benutzt

7.3 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

7.4 Weitere Quellen

Wikipedia

Stackoverflow.com

8 Glossar

Administrator

Benutzer mit erweiterten Rechten

Cookie

Eine Textinformation, die von der besuchte Website auf dem Rechner des Betrachters speichert

CSS

Cascading Style Sheets ist eine Stylingsprache für HTML

Footer

Die Fusszeile der Webseite

Framework

Ein Programmiergerüst auf dem aufgebaut werden kann

•SwissBanking

Front-End

Teil der Anwendung mit welcher der Benutzer kommuniziert.

GIT

Versionsverwaltungssoftware

GitHub

Webbasierte Entwicklungssoftware mit Versionsverwaltung

GUI

Die grafische Oberfläche einer Applikation

Header

Die Kopfzeile einer Webseite

HTML

Hypertext Markup Language ist eine auf Text basierende Auszeichnungssprache, die im Internet benutzt wird

IPERKA

Projektmanagementmethode

JavaScript / JS

Skriptsprache für dynamisches HTML

Mobile-first

Auf mobile Endgeräte fokussiert

PHP

Serverseitig interpretierte Skriptsprache

Prototyping

Erstellen von Prototypen für schnelles Feedback vor der Entwicklung der Applikation

Repository

Digitales Archiv

Responsives Webdesign

Gestaltungsart einer Webseite, die auf die Grösse des Webbrowsers reagiert und sich anpasst.

Session

Verbindung eines Clients mit einem Server

Update

Aktualisierung einer Software

•SwissBanking

XAMPP

Zusammenstellung von Software für das einfache Installieren und Konfigurieren eines Webservers mit Datenbank.

Salt

Eine zufällig gewählte Zeichenfolge

Hash

Nicht zurück verfolgbare Zeichenreihenfolge

MySQLI

MySQL Improved ist eine verbesserte Version von MySQL

PDO

PHP Data Objects, objektorientierte Datenbankschnittstelle