

**Pflichtenheft  
für  
das Projekt  
Helpdesk mit Ticket System**

**Gruppe B**

**Teammitglieder:**

**Matthias Feil**

**Kevin Klose**

**Regina Richter**

## Inhaltsverzeichnis

1. Ausgangssituation .....	4
2. Software-Architektur.....	5
3. Use-Case-Diagramm und Datenbank-Schema .....	6
4. Wireframes.....	7
4.1 Ticket erstellen .....	7
4.2 Ticket lösen.....	8
4.3 Statistik einsehen .....	8
5. Konkrete Aufgaben.....	9
5.1 Aus der Sicht des Endnutzers .....	9
5.1.1 Innerhalb von 15 Minuten kann ein Ticket erstellt werden.....	9
5.1.2 Auf das Ticketsystem kann jederzeit innerhalb des Intranets zugegriffen werden .....	9
5.1.3 Mit Eingabe der E-Mail werden die Kontaktdaten automatisch ergänzt.....	9
5.1.4 Es gibt Vorauswahlmöglichkeiten in Form von Drop-Down-Feldern .....	9
5.1.5 Es gibt ein Freitextfeld, um das Problem zu schildern .....	9
5.1.6 Es gibt ein Feld zum Hochladen von Screenshots .....	9
5.1.7 Der richtige IT-Mitarbeiter wird mittels der Schlagwörter bestimmt.....	9
5.1.8 Schrittweiser Aufbau des Formulars .....	9
5.2 Aus der Sicht des IT-Mitarbeiters .....	9
5.2.1 Die empfangene E-Mail kommt von einer Funktionsmail-Adresse.....	9
5.2.2 Die E-Mail enthält alle wichtigen Informationen .....	9
5.2.3 Folgende Felder sollen implementiert sein .....	10
5.2.4 Eskalationsstufen.....	10
5.2.5 Über einen generierten Link in der E-Mail wird die Lösung mitgeteilt .....	10
5.2.6 Ticketsystem Übersicht .....	10
5.2.7 Ticketstatus auf gelöst setzen .....	10
5.3 Aus der Sicht des IT-Abteilungsleiters .....	11
5.3.1 Das Ticketsystem soll jederzeit erreichbar sein .....	11
5.3.2 Es gibt eine statistische Auswertung der Ticket-Daten.....	11
5.3.3 Für das Ticketsystem ist keine Schulung nötig.....	11
5.3.4 Es werden gängige Programmiersprachen verwendet .....	11
6. Constraints /Nicht-Funktionale Anforderungen.....	12
6.1.1 Robustheit: Ticketsystem soll bei falschen Eingaben nicht abstürzen.....	12
6.1.2 Verfügbarkeit: Zugriff soll von Überall über das Intranet möglich sein .....	12
6.1.3 Änderbarkeit: Quellcode soll ausführlich kommentiert sein, um nachträgliche Veränderungen vornehmen zu können .....	12

6.1.4 Zuverlässigkeit: Ticketsystem muss den Mitarbeitern zu jeder Uhrzeit (24x7) zur Verfügung stehen .....	12
6.1.5 Benutzbarkeit: Orientierung bei Ticketeingabe an Kontaktformularen.....	12
6.1.6 Sicherheit: Konformität im Umgang mit personenbezogenen Daten nach Art. 5 Abs. 1 DSGVO .....	12
7. Abgrenzung, was das Ticketsystem nicht können muss .....	13

## **1. Ausgangssituation**

Viele kleinere Behörden haben kein eigenes Ticketsystem für IT-Angelegenheiten. Wenn bei Mitarbeitern Probleme auftreten oder diese Fragen haben, müssen diese dann versuchen jemanden aus der IT-Abteilung telefonisch oder per E-Mail zu erreichen. Oftmals ist hierbei schon die erste Herausforderung den richtigen Ansprechpartner zu finden. Zudem kann es vorkommen, dass diese Anfragen im normalen Tagesgeschäft untergehen und der Mitarbeiter sich mehrmals melden muss, damit sein Anliegen bearbeitet wird. Dies kann auf beiden Seiten zu Spannungen und einer geringeren Produktivität führen, welche mit einem Ticketsystem für den Erstkontakt vermeidbar wären.

## 2. Software-Architektur

Es wird eine klassische drei-Schichten-Architektur verwendet. Diese ist den Entwicklern gut bekannt und für unsere Zwecke ausreichend.

Presentation Layer:     HTML, CSS, Javascript, Bootstrap     Webbrowser (z.B. Firefox)

Applikation Layer:     PHP     Apache Server

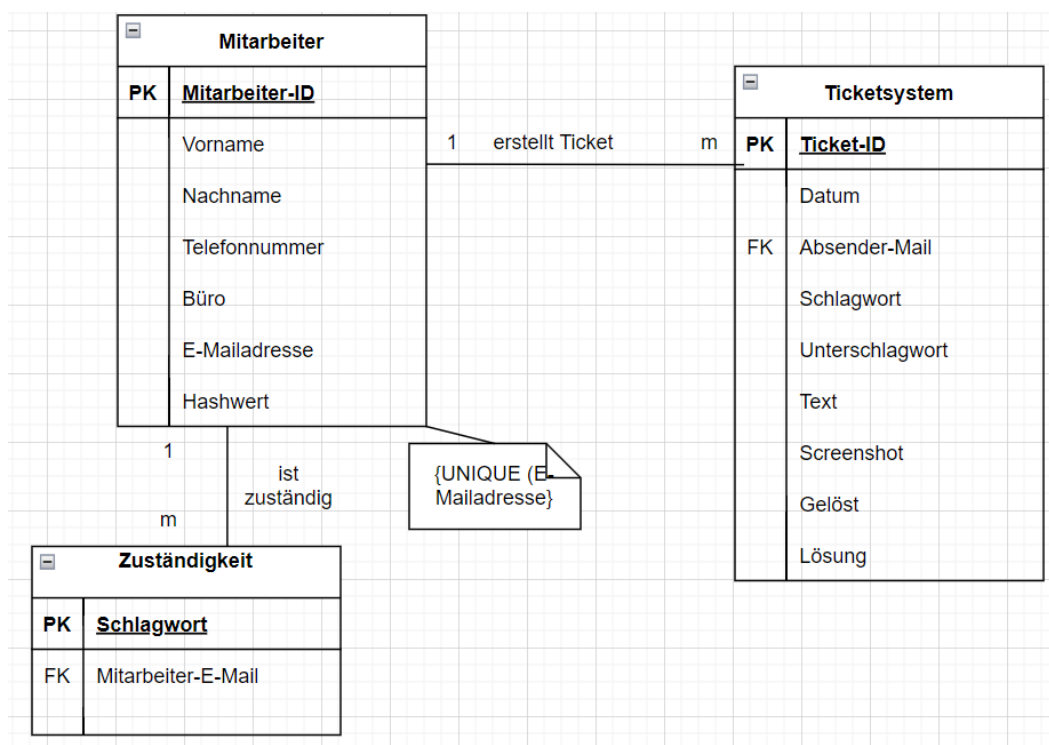
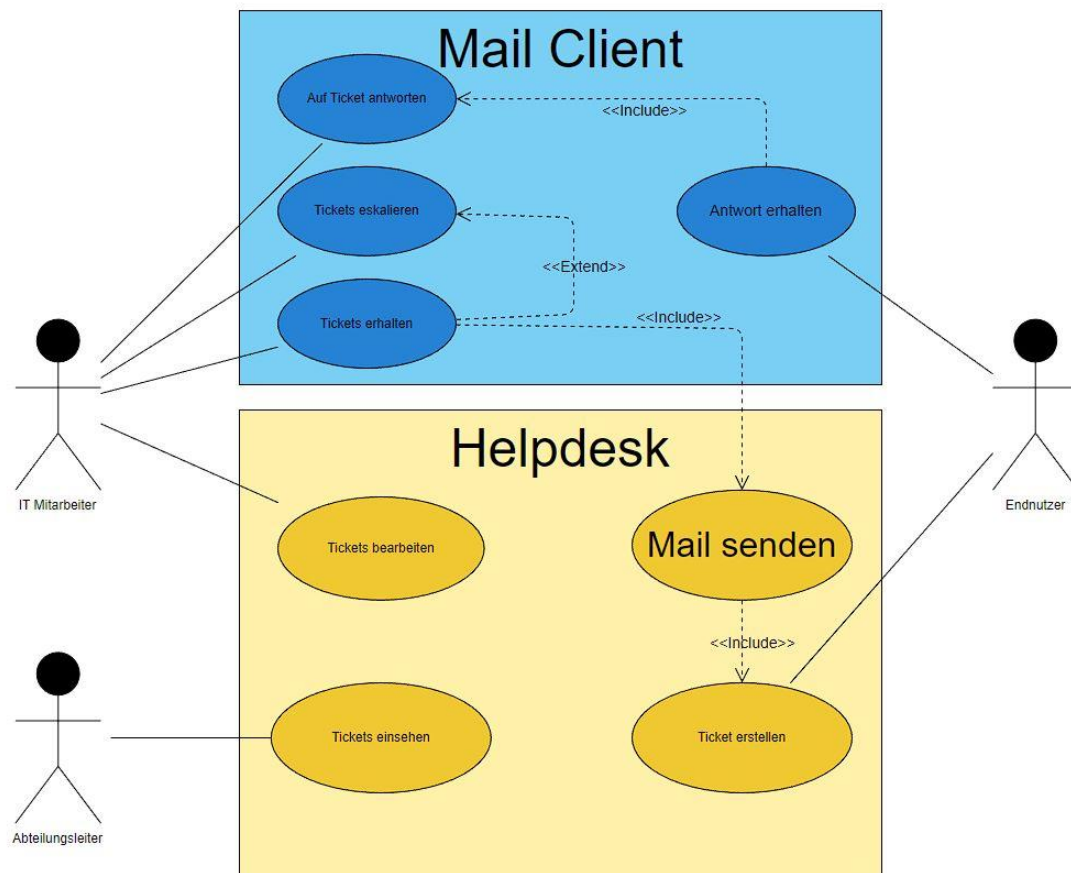
Data Layer:     SQL     MySQL

Mit diesen Frameworks lässt sich XAMPP verwenden. Damit kann dann lokal auf den eigenen Rechnern entwickelt und getestet werden, wodurch die Beschaffung eines Servers wegfällt.

Als Versionsverwaltungstool wird GitHub verwendet. Für das Versenden der Mails wird die API formsubmit verwendet.

- Die Daten werden dauerhaft auf der MySQL-Datenbank gespeichert, da sie auch für statistische Zwecke ausgewertet werden sollen.
- Um sämtliche Kontaktinformationen des Antragstellers zu erhalten, kann das Unternehmen ein Active-Directory oder Datenbank einbinden. Hier wird zu Testzwecken eine einfache Datenbank verwendet.
- Der Benutzer verwendet eine Weboberfläche um seine Anfrage zu stellen.
- Das Projekt ist nur für den deutschsprachigen Raum ausgelegt. Eine Mehrsprachigkeit ist nicht vorgesehen, da sie in der Implementierung zu aufwendig ist und kaum genutzt wird, denn in deutschen Behörden werden Deutschkenntnisse vorausgesetzt und die Amtssprache ist Deutsch.
- Da die Anfragen nur über das Intranet gestellt werden, für dessen Zugang man sich vorher Authentifizieren muss, werden die Daten unverschlüsselt übertragen.
- Eine Protokollierung und Ausnahmebehandlung ist nicht vorgesehen, da sie zu aufwendig sind. Lediglich die E-Mail-Adresse wird überprüft, ob sie in dem Active-Directory vorliegt, damit eine E-Mail versendet werden kann.

### 3. Use-Case-Diagramm und Datenbank-Schema



## 4. Wireframes

### 4.1 Ticket erstellen

E-Mail

\*

Vorname Nachname Telefonnummer Büro

PC-Nummer

\*

Wo tritt das Problem auf? Art des Problems ?

☐ Behörde  
☐ Home Office  
☐ Außendienst

Bildschirm ▼

**Hardware**  
Bildschirm  
PC  
Drucker  
Handy  
Sonstige

**Netzwerk**  
Netzwerk

**Software**  
Office  
Adobe  
Sonstiges

**IT-Sicherheit**  
Spam-Mail  
Sonstiges

**Beratung**  
Beratung zu IT-spezifischen  
Fragen

**Weitere Anliegen**  
Sonstiges

Problembeschreibung und bisherige Lösungsansätze ?

☐ Ich stimme der Speicherung meiner E-Mailadresse für die Kontaktaufnahme zu

Absenden

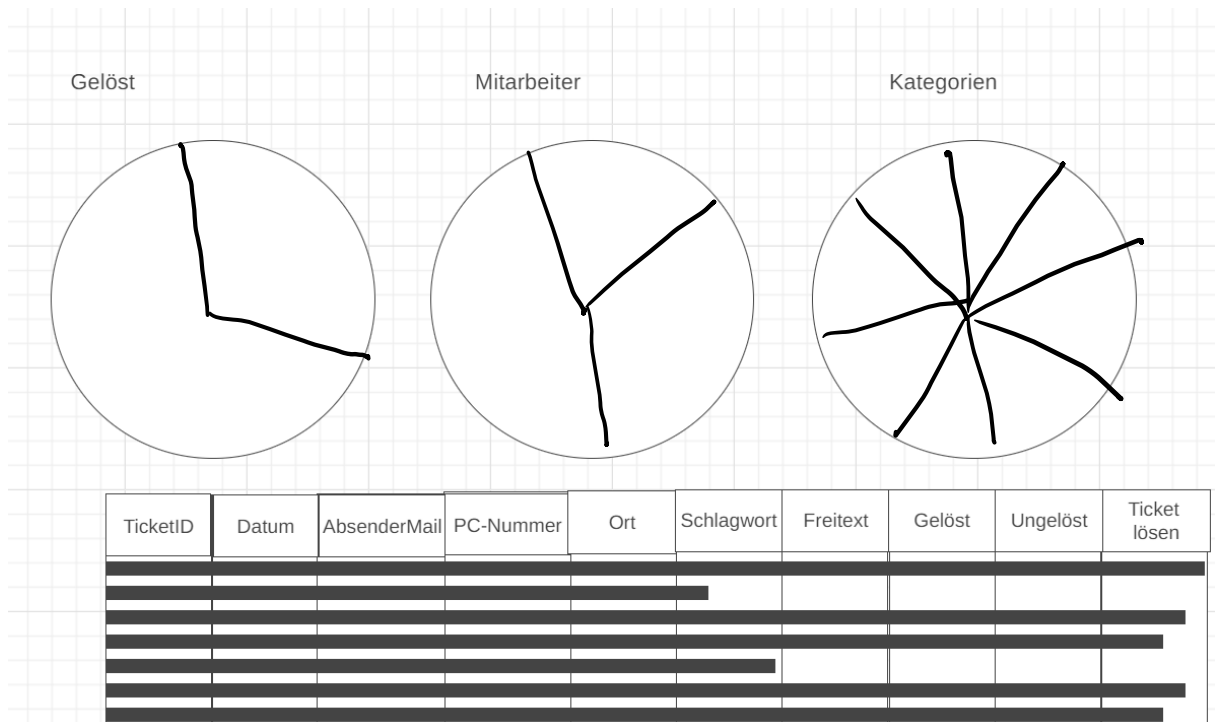
<https://wireframe.cc/mbCan1>

## 4.2 Ticket lösen

A wireframe for a 'Ticket lösen' (Solve Ticket) form. It features a 'Ticket Nummer' label above a text input field. Below this is a 'Lösung' label above a large text area. At the bottom, there is a checked checkbox labeled 'Gelöst' and a 'Senden' button.

<https://wireframe.cc/wmx8q9>

## 4.3 Statistik einsehen



<https://wireframe.cc/cN9OoE>



## **5. Konkrete Aufgaben**

### **5.1 Aus der Sicht des Endnutzers**

#### **5.1.1 Innerhalb von 15 Minuten kann ein Ticket erstellt werden**

Die Ticket Erstellung soll mit wenigen Klicks und ohne Vorwissen für den normalen Mitarbeiter möglich sein.

#### **5.1.2 Auf das Ticketsystem kann jederzeit innerhalb des Intranets zugegriffen werden**

Der Webserver muss 24/7 aus dem Intranet zugänglich sein.

#### **5.1.3 Mit Eingabe der E-Mail werden die Kontaktdaten automatisch ergänzt**

Dies geschieht mit einer Datenbank-Abfrage im Hintergrund über einen XMLHttpRequest. Die Felder Name, Telefonnummer, Büro werden gefüllt, wenn eingegeben E-Mail-Adresse mit Daten aus Mitarbeiterdatenbank übereinstimmen.

#### **5.1.4 Es gibt Vorauswahlmöglichkeiten in Form von Drop-Down-Feldern**

Die Vorauswahlmöglichkeiten sollen folgende alltägliche Schlagwörter sein.

##### **Netzwerk**

**Software:** Office, Adobe, Sonstiges

**Hardware:** PC, Bildschirm, Drucker, Handy, Sonstiges

**IT-Sicherheit:** IT-SpamMails, Sonstiges

**Beratung zu IT-spezifischen Fragen**

**Sonstiges**

#### **5.1.5 Es gibt ein Freitextfeld, um das Problem zu schildern**

Zudem gibt es einen Hinweis-Button, der eine Formulierungshilfe gibt.

#### **5.1.6 Es gibt ein Feld zum Hochladen von Screenshots**

Wird nicht implementiert, da Aufwand und Kosten für das Hochladen und Abspeichern von Bildern deutlich größer sind als der Nutzen.

#### **5.1.7 Der richtige IT-Mitarbeiter wird mittels der Schlagwörter bestimmt**

Je nach Schlagwort, wird an den zuständigen Mitarbeiter eine E-Mail versendet. Dies geschieht über einen XMLHttpRequest.

#### **5.1.8 Schrittweiser Aufbau des Formulars**

Formular wird schrittweise aufgebaut, wenn ein Mitarbeiter seine Eingaben macht für eine leichtere Bedienbarkeit in folgender Reihenfolge:

- Persönliche Daten: Name, Email, Telefonnummer, Büronummer
- Angaben zum Gerät: PC-Nummer
- Problembeschreibung

### **5.2 Aus der Sicht des IT-Mitarbeiters**

#### **5.2.1 Die empfangene E-Mail kommt von einer Funktionsmail-Adresse**

Hierfür wird die API Formsubmit verwendet. Somit kommt diese auch von einer einheitlichen E-Mail-Adresse namens [submissions@formsubmit.co](mailto:submissions@formsubmit.co).

#### **5.2.2 Die E-Mail enthält alle wichtigen Informationen**

Die Felder aus dem Ticket-Formular werden formatiert und in die E-Mail hineingeschrieben. Zudem wird noch für den Mitarbeiter ein Link in der Mail mit versendet, über welchen dieser direkt auf die Ticketsystemübersicht zugreifen kann.

### **5.2.3 Folgende Felder sollen implementiert sein**

- Persönliche Daten: Name, Email, Telefonnummer, Büronummer
- Angaben zum Gerät: PC-Nummer
- Problembeschreibung
  - Wo tritt das Problem auf: Behörde, Homeoffice, Außendienst
  - Art des Problems: siehe 5.1.4
  - Problembeschreibung und bisherige Lösungsansätze

### **5.2.4 Eskalationsstufen**

Da HelpDesk für normale Mitarbeiter ohne IT-Kenntnisse gedacht ist. Würden hier mehrere Eskalationsstufen nur zu Verwirrung führen bzw. würde jeder Mitarbeiter sein Problem als schwerwiegend einstufen um eine schnellere Bearbeitung zu erzwingen. Deswegen wird darauf verzichtet dies zu implementieren.

### **5.2.5 Über einen generierten Link in der E-Mail wird die Lösung mitgeteilt**

Auf diese Funktionalität wird verzichtet, da das Ticket vom IT-Mitarbeiter auf gelöst gesetzt wird, wenn er das Problem für den Mitarbeiter gelöst hat. Daher kennt der Mitarbeiter mit dem Problem die Lösung bereits bzw. bei komplexen Problemen hat ein Mitarbeiter mit wenig IT-Kenntnissen hieraus keinen Mehrwert und wird im schlimmsten Fall noch verwirrt.

Alternativ wird nun eine Tabelle bei der Statistikübersicht erstellt, in welcher alle Tickets aufgelistet werden und man das Ticket lösen kann.

### **5.2.6 Ticketsystem Übersicht**

Es werden tabellarisch alle Tickets aufgelistet, wobei neueste Tickets zuerst dortstehen. In dieser sind folgende Informationen aufgelistet:

- TicketID
- Datum
- AbsenderMail
- PC-Nummer
- Ort
- Schlagwort
- Freitext
- Gelöst
- Lösung
- Ticket lösen

Über Ticket lösen wird der Mitarbeiter auf ein Eingabeformular weitergeleitet, in welches der Mitarbeiter den Ticketstatus auf gelöst setzen und seine Lösung angeben kann. Bei der Spalte gelöst wird zur besseren Lesbarkeit der Hintergrund gelb/rot angezeigt bei Tickets die noch nicht gelöst wurden und bei Tickets mit Lösung ist der Hintergrund grün.

### **5.2.7 Ticketstatus auf gelöst setzen**

Hier kann der Mitarbeiter seine Lösung angeben, welche dann in der Datenbank gespeichert wird und im Ticketsystem der Status auf gelöst gesetzt wird und die Lösung in der Tabelle angezeigt wird. Sollte es hierbei ein Problem beim abspeichern der Datenbank geben, wird

dies dem Mitarbeiter mitgeteilt. Ansonsten kriegt der Mitarbeiter eine Erfolgsmeldung und kann über einen Link wieder zur Ticketübersicht zurückkehren.

### **5.3 Aus der Sicht des IT-Abteilungsleiters**

#### **5.3.1 Das Ticketsystem soll jederzeit erreichbar sein**

Wird sichergestellt, indem Ticketsystem über einen Webserver 24/7 erreichbar ist.

#### **5.3.2 Es gibt eine statistische Auswertung der Ticket-Daten**

Alle Tickets werden in einer Datenbank gespeichert mit Datum, wann das Ticket erstellt wurde. Es wird ausgewertet nach den Lösungsstatus, Problembereich, Zuständige Mitarbeiter.

#### **5.3.3 Für das Ticketsystem ist keine Schulung nötig**

Orientierung an Kontaktformularen aus dem Internet und Hinweise bei schwierigen Feldern, damit Mitarbeiter ohne IT-Kenntnisse diese ausfüllen können.

#### **5.3.4 Es werden gängige Programmiersprachen verwendet**

JavaScript, PHP, CSS und HTML. Auf eine gute Dokumentation des Codes wird geachtet.

## **6. Constraints /Nicht-Funktionale Anforderungen**

### **6.1.1 Robustheit: Ticketsystem soll bei falschen Eingaben nicht abstürzen**

Bei falschen Eingaben wird eine Fehlermeldung ausgegeben. Wenn beim Ticketformular der Mitarbeiter eine E-Mail-Adresse angibt, die sich nicht in der Mitarbeiter-Datenbank befindet wird beim Absenden des Tickets eine Fehlermeldung ausgegeben und das Ticket wird nicht in der Datenbank gespeichert, da der Mitarbeiter über die E-Mail-Adresse identifiziert wird.

### **6.1.2 Verfügbarkeit: Zugriff soll von Überall über das Intranet möglich sein**

Seite muss nur im Behördennetz implementiert werden. Hierbei ist zu beachten, dass das Ticketsystem und das Lösen von Tickets in einem getrennten Bereich (z.B. backend) implementiert wird, auf welchen nur IT-Mitarbeiter Zugriff haben. Und das Formular in einem Bereich (z.B. frontend) auf den alle Mitarbeiter zugreifen können.

### **6.1.3 Änderbarkeit: Quellcode soll ausführlich kommentiert sein, um nachträgliche Veränderungen vornehmen zu können**

### **6.1.4 Zuverlässigkeit: Ticketsystem muss den Mitarbeitern zu jeder Uhrzeit (24x7) zur Verfügung stehen**

### **6.1.5 Benutzbarkeit: Orientierung bei Ticketeingabe an Kontaktformularen**

Hierbei wurde sich an gängigen Kontakformularen von Anbietern wie z.B. EON, Otto, Lichtblick

### **6.1.6 Sicherheit: Konformität im Umgang mit personenbezogenen Daten nach Art. 5 Abs. 1 DSGVO**

Mitarbeiter kann nur ein Ticket abschicken, wenn er zustimmt, dass seine Daten zur Kontaktaufnahme gespeichert werden dürfen.

## **7. Abgrenzung, was das Ticketsystem nicht können muss**

Dieses Ticket-System soll nur dabei helfen den Erstkontakt zwischen Benutzern und der IT-Abteilung herzustellen. Es ist nicht für andere Bereiche in einer Behörde z.B. Fuhrpark, Hausmeister, etc. gedacht.