

# **Dokumentation**

## **für**

### **das Projekt**

#### **Helpdesk mit**

##### **Ticket System**

**Gruppe B**

**Teammitglieder:**

**Matthias Feil**

**Kevin Klose**

**Regina Richter**

**Inhaltsverzeichnis**

Einführung ..... 3

Installation und Konfiguration ..... 3

Ticketerstellung ..... 4

Ticketverwaltung ..... 4

Unit Tests ..... 5

Häufig auftretende Probleme in Tests ..... 6

# 1. Einführung

**Diese Dokumentation beschreibt das Ticketssystem und soll als Leitfaden für Benutzer dienen, die damit arbeiten möchten.**

**Das Ticketssystem ist eine webbasierte Anwendung, die es ermöglicht, Supportanfragen zu verfolgen und zu verwalten. Es wurde entwickelt, um Benutzern zu helfen, Anfragen schnell und effizient zu bearbeiten und zu lösen, indem sie Tickets erstellen, bearbeiten und verfolgen können.**

**Diese Dokumentation ist für alle Benutzer des Ticketsystems gedacht, einschließlich Supportmitarbeiter, Kunden und andere interessierte Personen. Sie soll als Referenzmaterial dienen und die wichtigsten Funktionen und Eigenschaften des Ticketsystems beschreiben.**

**In dieser Dokumentation werden Sie lernen, wie Sie das Ticketssystem installieren und konfigurieren, Benutzerkonten verwalten, und Tickets erstellen und verfolgen. Wir hoffen, dass diese Dokumentation Ihnen hilft, das Ticketssystem erfolgreich zu nutzen.**

## 2. Installation und Konfiguration

**Um das Ticketsystem zu installieren befolgen Sie bitte folgende Schritte:**

- 1. XAMPP installieren**
- 2. Repository downloaden**
- 3. Dateien unter 'htdocs' im XAMPP-Ordner speichern**
- 4. XAMPP Apache und Mysql auf Control Panel starten**
- 5. Im Webbrowser 'localhost/phpmyadmin' aufrufen**

- 6. Auf dieser Seite eine neue Datenbank erstellen mit den Namen 'helpdesk'**
- 7. Datei helpdesk.sql aus dem GitHub-Projekt importieren**
- 8. Im Browser 'localhost/Helpdesk' aufrufen**

### **3. Ticketerstellung**

**Um ein Ticket aufzugeben befolgen Sie bitte folgende Schritte:**

- 1. Rufen Sie den Helpdesk im Browser auf.**
- 2. Geben Sie Ihre dienstliche Emailadresse in das Feld „E-Mail“ ein.**
- 3. Das Ticketsystem prüft ob Ihre Angaben bereits bekannt sind.**

**Falls nicht tragen Sie bitte Ihre persönlichen Kontaktdaten in die entsprechenden Felder ein.**

- 4. Geben Sie Ihre PC-Nummer im Feld „PC-Nummer“ an. Die Nummer Ihres PCs finden Sie auf dem gelben Aufkleber auf der Ober oder Innenseite Ihres Rechners.**

- 5. Beschreiben Sie mithilfe des Dropdown Menüs wo das Problem auftritt.**

- 6. Beschreiben Sie mithilfe des Dropdown Menüs um welche Art des Problems es sich handelt.**

**Dies hilft Sie direkt an den richtigen Ansprechpartner zu vermitteln.**

- 7. Beschreiben Sie mithilfe des Textfeldes was das Problem ist und welche Schritte Sie bereits zur Lösung des Problems unternommen haben.**

**Dies hilft Ihrem Ansprechpartner sich schneller in das Problem einzuarbeiten.**

**8. Bestätigen Sie mithilfe der Checkbox die Übermittlung Ihrer persönlichen Kontaktinformationen.**

**9. Versenden Sie Ihr Ticket mithilfe der Schaltfläche „Absenden“.**

**Das Ticket wird unmittelbar an die entsprechende Kontaktperson als E-Mail versendet.**

## **4. Ticketverwaltung**

**Um Tickets einzusehen und zu lösen befolgen Sie folgende Schritte:**

**1. Rufen Sie das Ticketsystem im Browser auf (localhost/ticketsystem.php).**

**2. Wählen Sie das zu lösende Ticket mit einem Klick auf „Ticket“ unter dem Reiter „Ticket Lösen“.**

**3. Geben Sie optional eine Lösung an.**

**4. Bestätigen Sie die Lösung des Tickets mithilfe der Schaltfläche „Absenden“.**

**Das Ticket wurde nun auf „Gelöst“ gesetzt.**

## **5. Unit Tests**

**Unit Tests sind kleine, isolierte Tests, die verwendet werden, um sicherzustellen, dass ein bestimmter Teil einer Software (eine "Unit") korrekt funktioniert. Im Kontext eines Ticketssystems könnten Unit Tests beispielsweise verwendet werden, um sicherzustellen, dass die Funktionen zum Erstellen, Bearbeiten und Verfolgen von Tickets ordnungsgemäß funktionieren.**

**Die Verwendung von Unit Tests hat mehrere Vorteile. Zunächst können sie dazu beitragen, Fehler frühzeitig zu identifizieren und zu beheben, was die Qualität der Software verbessert. Zweitens können sie dazu beitragen, den Code besser verständlich und wartbar zu machen, indem sie sicherstellen, dass jede Einheit eine klar definierte Funktionalität hat. Drittens können sie dazu beitragen, die Entwicklungszeit zu verkürzen, indem sie sicherstellen, dass Änderungen an einem Teil der Software keine unerwarteten Auswirkungen auf andere Teile haben.**

**Im Allgemeinen empfiehlt es sich, Unit Tests für wichtige Funktionen des Ticketssystems zu schreiben, um sicherzustellen, dass diese ordnungsgemäß funktionieren. Dies kann dazu beitragen, die Qualität und Zuverlässigkeit des Systems zu verbessern und das Risiko von Fehlern zu minimieren.**

**Das Ticketsystem konnte hierbei allerdings schlecht in einzelne Units geteilt werden. Grund dafür ist die enge Zusammenarbeit von Elementen, sowie der Datenbankanbindung und der verwendeten APIs.**

**Das größte Problem bei der Erstellung von Testfällen war in diesem Projekt das händische Ausfüllen der Form. Um nach einer Änderung die Funktionalität testen zu können wurde der Unit Test Button und die Unit Test Funktion in den Code implementiert. Die Schaltfläche Unit Test ist immer sichtbar. Bei der Betätigung dieser Schaltfläche wird die Unit Test**

**Funktion aufgerufen. Diese Funktion versendet gleichzeitig drei Testfälle. Einzeln gedrückt oder in schneller Reihenfolge wiederholt kann so die Funktionalität sowie Belastungsgrenze des Systems getestet werden.**

## **6. Häufig auftretende Probleme in Tests**

**Das System ist funktionsfähig. Sowohl manuelle als auch automatisierte Testfälle werden zuverlässig, schnell und korrekt in die Datenbank gespeichert. Bei den Labortests sind jedoch zwei Probleme, bezogen auf externe Server, wiederholt aufgefallen:**

**1. Da die verwendete API von FormSubmit nicht opensource und somit nicht einsehbar ist konnte die genaue Funktionsweise der API analysiert werden. Zudem kann eine hohe Auslastung der FormSubmit Server zu Performanceeinbußen führen.**

**2. Die derzeitigen Wartungsarbeiten an den Webservern der Fachhochschule Hof führten zu erheblichen Wartezeiten bei der Ankunft der Tickets an die Zielperson. Ein Verlust der Tickets war aufgrund des Arbeiten und eventueller Spam Filter nicht auszuschließen. Dies erschwerte die zuverlässige Ausführung der Tests enorm.**

**Es wird jedoch davon ausgegangen dass bei hinreichendem Zugang zu den Webserver Kontrollstrukturen einer Behörde in der dieses System implementiert werden soll diese Fehler nicht auftreten werden.**