# Helpdesk

#### Projekt der Oberstufe

Informationstechnischen Assistent(inn)en am

TBK-Solingen

[Timon Wegner](mailto:taw.kontakt@gmail.com), [Lia Kaup](mailto:lia@xomo.de), [Jadon Sill](mailto:jadonnoah06@gmail.com), [Timo Szadkowski](mailto:timosz0103@gmail.com), [Maurice Severin](mailto:schueler.mseverin@tbk-solingen.de), [Leon Wilde](mailto:l.wilde@mail.de)

[Helpdesk 1](#_Toc1)

[Projekt der Oberstufe 1](#_Toc2)

[Abkürzungsverzeichnis 3](#_Toc3)

[Abbildungsverzeichnis 4](#_Toc4)

[1. Anforderungskatalog 5](#_Toc5)

[2. Entity Relationship Diagramm 6](#_Toc6)

[3. Relationales Datenbankmodell 7](#_Toc7)

[4. Physischer Datenbankentwurf 8](#_Toc8)

## Abkürzungsverzeichnis

## Abbildungsverzeichnis

### 1. Anforderungskatalog

Im Rahmen von eigen Bedarf und zu besseren Dokumentation von Ticket, soll ein Ticketierungssystem entwickelt werden, über welches primär Mitarbeiter der Firma Easitix, Ticket erstellen, bearbeiten, sowie abrechnen könnten. Aus einigen Diskussionsrunden wurden, dann die grundlegenden Strukturen der Datenbank ausgelegt (zu sehen hier). Aus weiteren Diskussionsrunden, einem Ansatz für ein Login System wurde und Rücksprache mit einigen Kunden wurde die Bitte angelegt auch ohne Mittelsmann der Firma Easitix, Tickets zu erstellen aus eigen Bedarf der Kunden.

Zum Schluss sollten folgende Funktionen und Strukturen hier zusammengefasst in einer Tabelle mit Prioritäten im ersten Prototypen des Systems existieren.

|  |  |  |
| --- | --- | --- |
| **Aufgabe** | **Beschreibung** | **Priorität** |
| **Ticket-Erstellung ermöglichen** | Benutzer müssen neue Tickets für verschiedene Anfragen oder Probleme erstellen können. | Hoch |
| **Statusverwaltung von Tickets** | Tickets sollten unterschiedliche Status (z. B. offen, in Bearbeitung, geschlossen) haben. | Hoch |
| **Detaillierte Beschreibung erfassen** | Jedes Ticket sollte eine detaillierte Problembeschreibung enthalten. | Hoch |
| **Interne Notizen hinzufügen** | System sollte ermöglichen, dass Mitarbeiter interne Notizen zu Tickets hinzufügen können. | Mittel |
| **Prioritäten festlegen** | Tickets müssen priorisiert werden können (z. B. hoch, mittel, niedrig). | Hoch |
| **Zuweisung an Mitarbeiter** | Tickets sollten an bestimmte Mitarbeiter zur Bearbeitung zugewiesen werden können. | Hoch |
| **Kundeninformationen erfassen** | Das System sollte Kundendaten wie Name, E-Mail, Telefonnummer und Kundennummer speichern. | Hoch |
| **Kommunikation mit Kunden** | Das System sollte die Kommunikation zwischen Mitarbeitern und Kunden über das Ticket ermöglichen. | Hoch |

|  |  |  |
| --- | --- | --- |
| **Zeiterfassung und -berechnung** | Es sollte eine Funktion zur Erfassung der Bearbeitungszeit für jedes Ticket geben. | Mittel |
| **Standort- und Firmeninformationen** | Kundenstandorte und Firmenzugehörigkeiten sollten gespeichert werden können. | Niedrig |
| **Ticketabrechnung ermöglichen** | Das System sollte eine Abrechnung der erbrachten Leistungen pro Ticket unterstützen. | Mittel |
| **Rechnungsstellung integrieren** | Rechnungen für bearbeitete Tickets sollten erstellt und versendet werden können. | Mittel |
| **Ratenzahlung verwalten** | Möglichkeit zur Verwaltung von Ratenzahlungen für Rechnungen. | Niedrig |

Zusätzlich wurden nach diesen Funktionen auf Feedback der Kunden einige weitere Anforderungen gestellten.

|  |  |  |
| --- | --- | --- |
| **Aufgabe** | **Beschreibung** | **Priorität** |
| **Erstellung von Berichten** | Das System sollte Berichte über Tickets, deren Status und Bearbeitungszeiten generieren können. | Mittel |
| **E-Mail-Benachrichtigungen** | Benutzer und Kunden sollten E-Mail-Benachrichtigungen bei Ticket-Aktivitäten erhalten können. | Hoch |
| **Such- und Filterfunktionen** | Tickets sollten nach verschiedenen Kriterien wie Status, Priorität und Mitarbeiter gefiltert und durchsucht werden können. | Mittel |
| **Rollen- und Rechteverwaltung** | Es sollten unterschiedliche Benutzerrollen und Zugriffsrechte im System verwaltet werden können. | Hoch |

### 2. Entity Relationship Diagramm

### 3. Relationales Datenbankmodell

### 4. Physischer Datenbankentwurf