# Helpdesk, ein Ticketierungssystem

#### Projekt der Oberstufe

Informationstechnischen Assistent(inn)en am

TBK-Solingen

Gruppenmitglieder

[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

Um den internen Bedarf und die Dokumentation sowie Verwaltung von Tickets zu verbessern, wird ein Ticketierungssystem entwickelt. In erster Linie soll den Mitarbeitern der Firma Easitix ermöglicht werdet Tickets effizient zu erstellen, zu bearbeiten und abzurechnen. Wärend einer vorläufigen Diskussionrunde wurden die grundlegenden Strukturen der Datenbank ausgearbeitet (siehe begleitendes [Diagramm](#_2._Entity_)).  
  
Weitere Besprechnungen ergaben nach Entwicklung eines rudimentären Login Systems das während dieses ersten Prototypen auch die Möglichkeit beigelegt werden kann, dass Kunden sowie deren Mitarbeiter sich anmelden können, um ohne Vermittelung der Firma Easitix Tickets eingereicht werden können.  
  
So sollte der erste Prototyp dieses Systems folgende Funktionen umfassen:

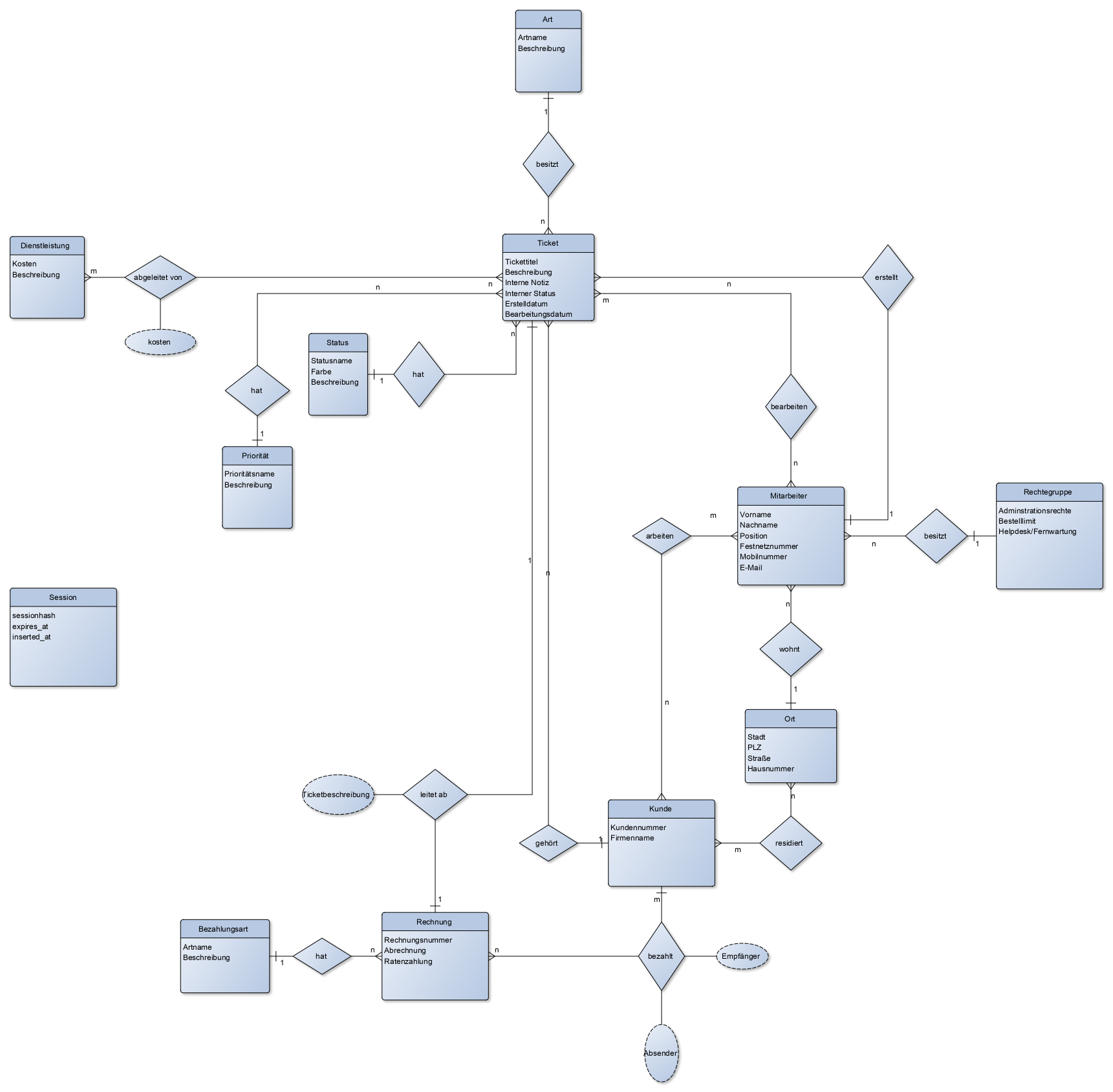
|  |  |  |
| --- | --- | --- |
| **Aufgabe** | **Beschreibung** | **Priorität** |
| **Ticket-Erstellung ermöglichen** | Benutzer müssen neue Tickets für verschiedene Anfragen oder Probleme erstellen können. | Hoch |
| **Statusverwaltung von Tickets** | Tickets sollten unterschiedliche Status (z. B. offen, 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 |
| **Erstellung von Tickets durch Kunden** | Möglichkeit Tickets zu erstellen ohne einen Mitarbeiter anzufragen diesen zu erstellen | Niedrig |

Zusätzlich wurden nach Rückmeldungen der Kunden folgende weitere Anforderungen definiert:

|  |  |  |
| --- | --- | --- |
| **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

abgeleitet: *FK\_Dienstleistung, FK\_Ticket*

arbeitet: *FK\_Mitarbeiter, FK\_Kunde*

art: PK\_Art - Artname, Beschreibung

bearbeitet: *FK\_Mitarbeiter, FK\_Ticket*

bezahlungsart: PK\_Bezahlungsart - Artname, Beschreibung

dienstleistung: PK\_Dienstleistung - Beschreibung, Kosten

kunde:  
PK: PK\_Kunde  
Attribute: Kundennummer, Firmenname, Email, Passwort  
FK: -

rechnung:  
PK: Rechnungsnr  
Attribute: Ticketbeschreibung, Abrechnung, Ratenzahlung  
FK: FK\_Bezahlungsart, FK\_Kunde, FK\_Ticket

mitarbeiter:  
PK: PK\_Mitarbeiter  
Attribute: Vorname, Nachname, Position, Festnetznummer, Mobilnummer, Email, Passwort  
FK: FK\_Ort, FK\_Rechtgruppe

ort: PK\_Ort - Stadt, PLZ, Straße, Hausnummer

priorität: PK\_Priorität - Prioritätsname, Beschreibung

rechtegruppe: PK\_Rechtgruppe, - Administrationsrechte, Bestelllimit, Helpdesk\_Fernwartung

residiert: *FK\_Kunde, FK\_Ort*

session: id - sessionhash, expires\_at, inserted\_at

status: PK\_Status - Statusname, Farbe, Beschreibung

ticket: PK\_Ticket - Ticket\_Titel, Beschreibung, InterneNotiz, InternerStatus, Erstelldatum, Bearbeitungsdatum - *FK\_Status, FK\_Priorität, FK\_Art, FK\_Mitarbeiter, FK\_Kunde*

### 4. Physischer Datenbankentwurf