Planung und Implementierung eines Kunden- und Ticketsystem

Lasten- und Pflichtenheft

Name: Elias Eyrisch

Klasse: WI24Z1

Betreuer: Herr Lukas

Datum: 09.01.2026

Inhalt

1. Einführung

Die HelpDesk Solutions GmbH ist ein fiktives mittelständisches Dienstleistungsunternehmen, das IT- und Serviceleistungen für Geschäftskunden anbietet. Im täglichen Betrieb fallen zahlreiche Kundenanfragen und Supportfälle an, die strukturiert bearbeitet werden müssen. Das Projektumfeld erfordert eine zentrale Softwarelösung, um Transparenz, Nachvollziehbarkeit und Effizienz in der Bearbeitung von Kundenanliegen sicherzustellen.

2. Ist-Situation

Aktuell werden Kundenanfragen über verschiedene Kanäle wie E-Mail, Telefon oder Excel-Listen verwaltet. Eine zentrale Übersicht über offene, laufende oder abgeschlossene Tickets existiert nicht. Dies führt zu Doppelbearbeitungen, fehlender Nachverfolgbarkeit und erhöhtem Verwaltungsaufwand.

3. Soll-Situation

3.1 Soll-Zustand

Nach Abschluss des Projekts steht eine WPF-Desktopanwendung zur Verfügung, die mit einer relationalen Datenbank verbunden ist. Kunden, Tickets und Benutzer können zentral verwaltet werden. Der Bearbeitungsstatus von Tickets ist jederzeit einsehbar.

Mehrwert für den Auftraggeber:

- Zentrale Verwaltung aller Kundenanfragen
- Klare Zuständigkeiten und Statusübersicht
- Zeitersparnis und bessere Servicequalität
- Nachvollziehbarkeit durch Ticket-Historie

3.2 Funktionale Anforderungen + Aufwandsschätzung

Funktion	Beschreibung	Priorität	Aufwand (h)
Benutzerlogin	Anmeldung mit Benutzername und Passwort	MUSS	8
Rollenverwaltung	Unterscheidung Kunde/Mitarbeiter/Admin	MUSS	8
Kunden anlegen/bearbeiten	Verwaltung von Kundendaten	MUSS	16
Ticket anlegen	Erstellung von Supporttickets	MUSS	12
Ticket bearbeiten	Status ändern, Kommentar hinzufügen	MUSS	12
Ticketliste	Übersicht aller Tickets mit Filter	MUSS	10
Dashboard	Übersicht offener Tickets	MUSS	12
GUI (WPF)	Bedienung über grafische Oberfläche	MUSS	20

3.3 Nicht-funktionale Anforderungen + Aufwandsschätzung

Anforderung	Beschreibung	Aufwand (h)
Benutzerfreundlichkeit	Intuitive, übersichtliche Oberfläche	10
Performance	Reaktionszeit < 3 Sekunden	6
Datensicherheit	Passwort-Hashing, Zugriffsbeschränkung	6
Stabilität	Fehlerbehandlung, keine Abstürze	8

3.4 Schnittstellen

Es sind keine externe Schnittstellen vorgesehen. Die Anwendung nutzt ausschließlich eine interne relationale Datenbank.

3.5 Risiken + Alternativen

Risiken bestehen insbesondere in Zeitüberschreitung und technischen Schwierigkeiten bei der Umsetzung von WPF oder der Datenbankanbindung. Als Gegenmaßnahme werden Kernfunktionen priorisiert und optionale Funktionen ggf. Weggelassen

4. Abnahmekriterien

4.1 Muss-Kriterien

1. Benutzer können sich anmelden
2. Kunden können angelegt und bearbeitet werden
3. Tickets können erstellt, bearbeitet und abgeschlossen werden
4. Daten werden Dauerhaft gespeichert

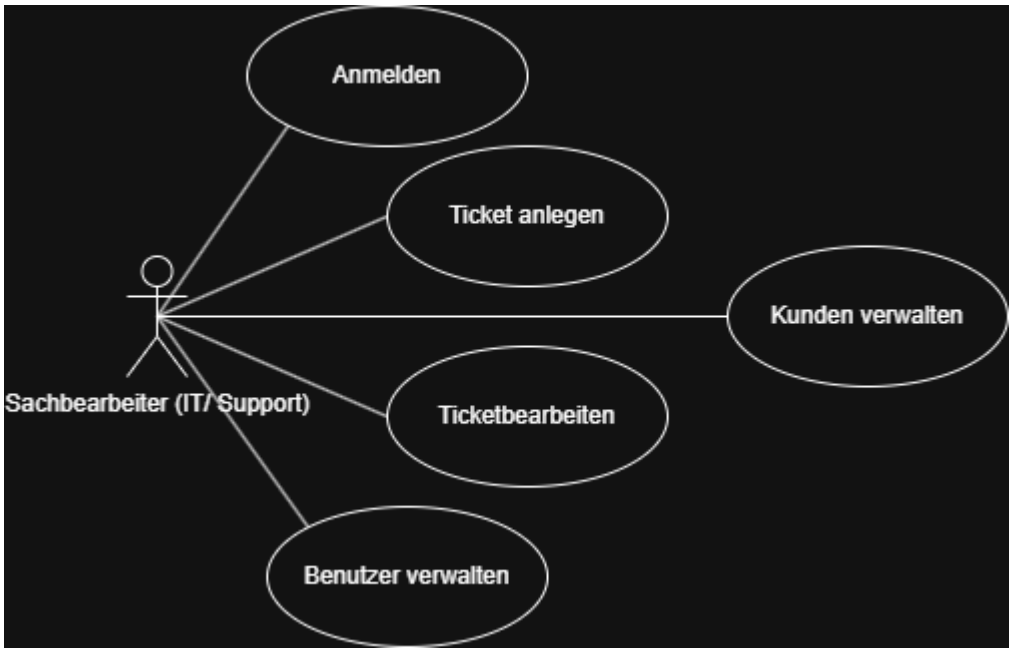
4.2 Kann-Kriterien

1. Statistische Auswertungen
2. Erweiterte Filterfunktionen

5. Use-Case-Diagramm und Use-Case-Beschreibung

Akteure: Sachbearbeiter(IT/Support)

Use-Case-Diagramm:



5.1 Tabellarische Detailbeschreibung zentraler Use-Cases

Name	Akteur	Vorbedingung	Ablauf (kurz)	Ergebnis / Nachbedingung
Anmelden	Sachbearbeiter (IT/Support)	Anwendung ist gestartet, Benutzer existiert	Benutzername und Passwort eingeben → Login bestätigen → Zugangsdaten werden geprüft	Benutzer ist erfolgreich angemeldet und hat Zugriff auf das System
Ticket anlegen	Sachbearbeiter (IT/Support)	Benutzer ist angemeldet	„Neues Ticket“ auswählen → Titel, Beschreibung und Kunde eingeben → Speichern	Ticket ist in der Datenbank gespeichert und in der Ticketübersicht sichtbar
Ticket bearbeiten	Sachbearbeiter (IT/Support)	Ticket existiert, Benutzer ist angemeldet	Ticket auswählen → Status oder Beschreibung ändern → Speichern	Ticket wurde aktualisiert und Änderungen sind gespeichert
Kunden verwalten	Sachbearbeiter (IT/Support)	Benutzer ist angemeldet	Kunden anlegen, bearbeiten oder löschen → Eingaben speichern	Kundendaten sind korrekt in der Datenbank gespeichert
Benutzer verwalten	Sachbearbeiter (IT/Support)	Benutzer ist als Administrator berechtigt	Benutzer anlegen, bearbeiten oder löschen → Rolle festlegen → Speichern	Benutzerkonten und Rollen sind korrekt gespeichert

6. Projektplan (Gantt-Diagramm)

Projektphase	KW 1	KW 2	KW 3	KW 4	KW 5	KW 6	KW 7	KW 8
Analyse & Lastenheft								
Pflichtenheft & Entwurf								
DB - Entwurf								
Implementierung Kernfunktionen								
Implementierung Benutzer- & Ticketverwaltung								
Testphase								
Dokumentation / Abschlussarbeit								
Pufferzeit								

7. Produktumgebung

Programmiersprache

- **C# (.NET)**
  Die Anwendung wird mit der Programmiersprache C# umgesetzt. C# eignet sich besonders für die Entwicklung von Windows-Desktopanwendungen und bietet eine gute Integration in das .NET-Framework.

Grafische Benutzeroberfläche

- **Microsoft WPF (Windows Presentation Foundation)**
  Für die Benutzeroberfläche wird WPF verwendet. WPF ermöglicht die Entwicklung moderner, datenbindungsbasierter Oberflächen und unterstützt eine klare Trennung von Oberfläche und Logik.

Architektur

- **MVVM (Model–View–ViewModel)**
  Die Anwendung folgt dem MVVM-Architekturmuster. Dieses Muster sorgt für eine saubere Trennung zwischen Benutzeroberfläche, Geschäftslogik und Datenmodellen und erhöht die Wartbarkeit und Erweiterbarkeit der Software.

Datenbank

- **SQL Server Express oder SQLite**
  Die Daten werden in einer relationalen Datenbank gespeichert. Abhängig vom Entwicklungsstand wird entweder SQL Server Express oder SQLite verwendet. Beide Datenbanksysteme sind für Desktopanwendungen geeignet und kostenlos nutzbar.

Entwicklungsumgebung

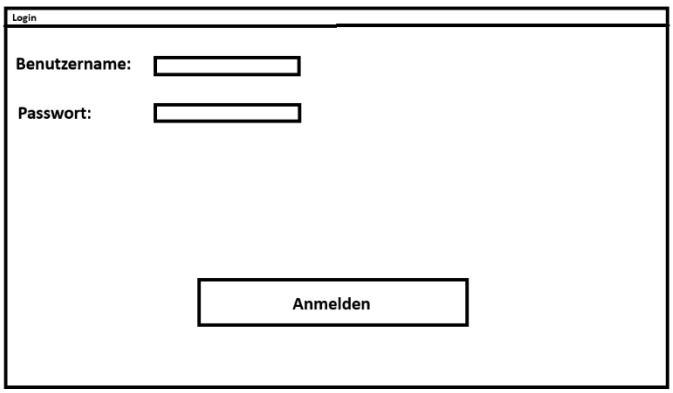
- **Visual Studio**
  Als Entwicklungsumgebung wird Microsoft Visual Studio eingesetzt. Visual Studio bietet umfangreiche Werkzeuge für die Entwicklung, das Debugging und das Testen von .NET-Anwendungen.

Versionsverwaltung

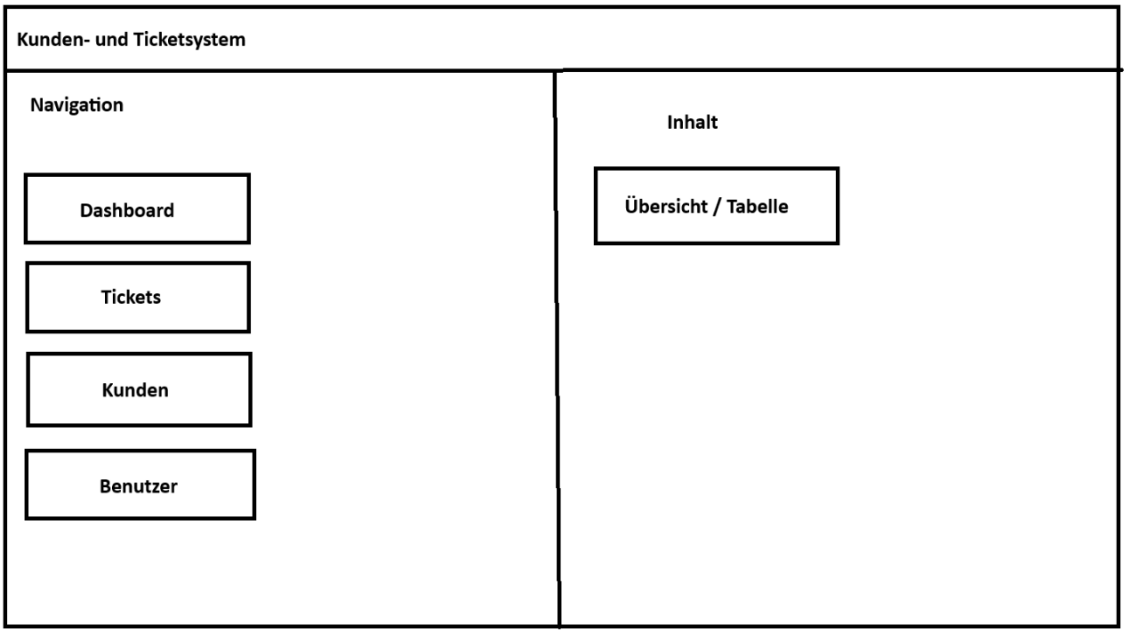
- **Git (GitHub)**
  Zur Versionsverwaltung des Quellcodes wird Git verwendet. Der Quellcode wird in einem GitHub-Repository verwaltet, um Änderungen nachvollziehbar zu dokumentieren und den Entwicklungsstand zu sichern.

8. Skizze der GUI

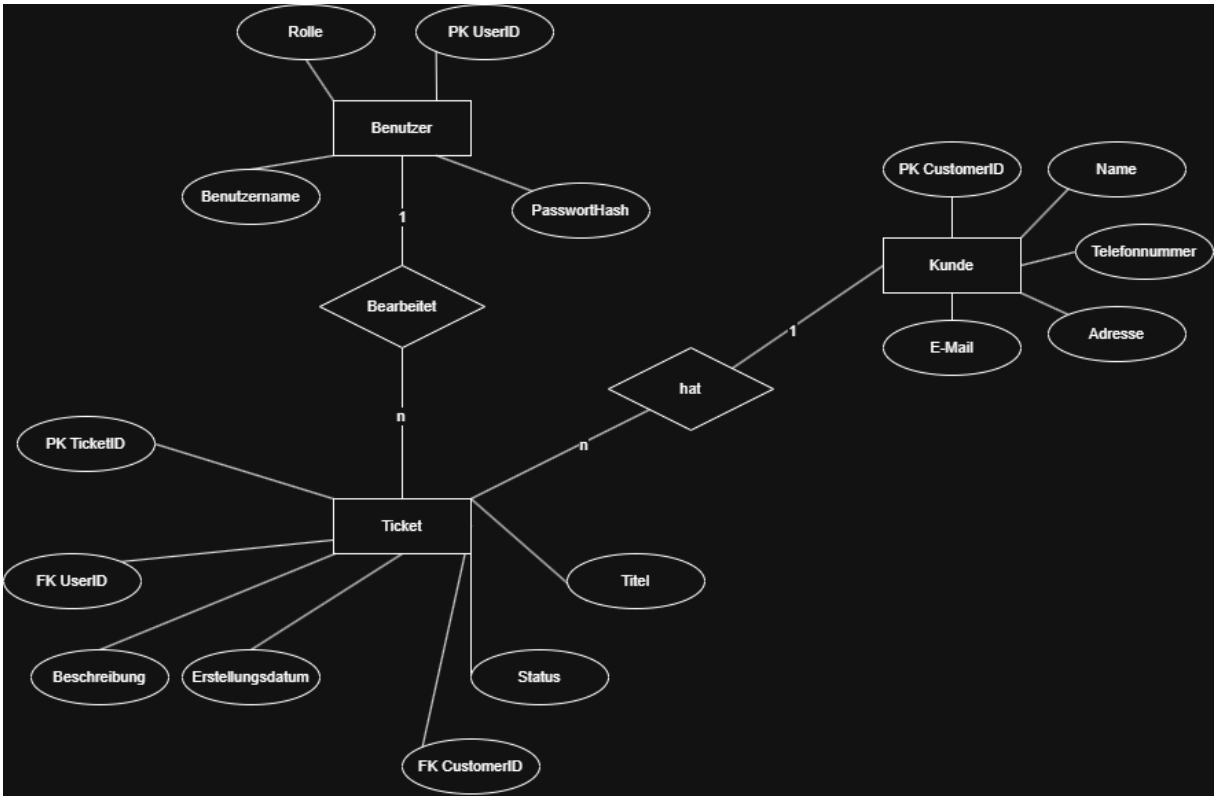
Login:



Hauptmenü:



9. DB-Entwurf (ER-Diagramm)



9.1 Beschreibung der wichtigsten Entitäten

Entität: Benutzer

Die Entität **Benutzer** repräsentiert die Sachbearbeiter (IT/Support), die mit der Anwendung arbeiten.
Benutzer können sich am System anmelden, Tickets bearbeiten und verwalten sowie – abhängig von ihrer Rolle – weitere administrative Aufgaben übernehmen.

Wesentliche Attribute:

- **UserID (Primärschlüssel):** Eindeutige Identifikation eines Benutzers
- **Benutzername:** Wird für die Anmeldung verwendet
- **PasswortHash:** Speicherung des verschlüsselten Passworts
- **Rolle:** Definiert die Berechtigungen des Benutzers (z. B. Sachbearbeiter, Administrator)

Ein Benutzer kann mehrere Tickets bearbeiten.

Entität: Kunde

Die Entität **Kunde** speichert alle relevanten Informationen zu den Kunden des Unternehmens.
Kunden sind die Auftraggeber bzw. Betroffenen der erfassten Supporttickets.

Wesentliche Attribute:

- **CustomerID (Primärschlüssel):** Eindeutige Identifikation eines Kunden
- **Name:** Name des Kunden
- **Adresse:** Anschrift des Kunden
- **E-Mail:** Kontaktadresse
- **Telefonnummer:** Telefonnummer des Kunden

Ein Kunde kann mehrere Tickets besitzen.

Entität: Ticket

Die Entität **Ticket** bildet das zentrale Element des Systems und stellt eine Service- bzw. Supportanfrage dar.
Ein Ticket ist immer genau einem Kunden zugeordnet und wird von einem Benutzer bearbeitet.

Wesentliche Attribute:

- **TicketID (Primärschlüssel):** Eindeutige Identifikation eines Tickets
- **Titel:** Kurze Beschreibung des Anliegens
- **Beschreibung:** Detaillierte Beschreibung des Problems
- **Status:** Aktueller Bearbeitungsstand (z. B. offen, in Bearbeitung, erledigt)
- **Erstellungsdatum:** Zeitpunkt der Ticketanlage
- **UserID (Fremdschlüssel):** Verweis auf den bearbeitenden Benutzer
- **CustomerID (Fremdschlüssel):** Verweis auf den zugehörigen Kunden

Tickets stellen die Verbindung zwischen Benutzern und Kunden dar und ermöglichen eine strukturierte Bearbeitung von Anfragen.

10. Link zum gehosteten Git-Repository

Link zum Git-Repository:

https://github.com/E-Eyrisch/Planung-und-Implementierung-eines-Kunden--und-Ticketsystem

11. Testplan

Der Testplan beschreibt die vorgesehenen Testfälle zur Überprüfung der Funktionalität des Kunden- und Ticketsystems.
Die Tests orientieren sich an den definierten Abnahmekriterien und stellen sicher, dass die wichtigsten Funktionen der Anwendung korrekt arbeiten.

Die Tests werden überwiegend manuell durchgeführt. Zusätzlich sind einfache Unit-Tests für ausgewählte logische Komponenten vorgesehen.

11.1 Testfälle

Testfall-Nr.	Testziel	Voraussetzung	Testablauf	Erwartetes Ergebnis
T1	Benutzeranmeldung prüfen	Benutzer existiert	Benutzername und Passwort eingeben → Anmelden	Benutzer wird erfolgreich angemeldet
T2	Ticket anlegen	Benutzer ist angemeldet	Neues Ticket anlegen → Daten eingeben → Speichern	Ticket wird gespeichert und angezeigt
T3	Ticket bearbeiten	Ticket existiert	Ticket auswählen → Status ändern → Speichern	Ticketstatus wird aktualisiert
T4	Kunden anlegen	Benutzer ist angemeldet	Neuen Kunden anlegen → Daten eingeben → Speichern	Kunde wird gespeichert
T5	Kunden bearbeiten	Kunde existiert	Kundendaten ändern → Speichern	Änderungen werden gespeichert
T6	Benutzer verwalten	Benutzer hat Admin-Rechte	Benutzer anlegen oder bearbeiten	Benutzer wird korrekt gespeichert
T7	Datenpersistenz prüfen	Daten wurden angelegt	Anwendung neu starten	Daten sind weiterhin vorhanden

11.2 Teststarten

1. **Manuelle Tests:**
   Die meisten Testfälle werden manuell durchgeführt, indem die Anwendung bedient und das Ergebnis überprüft wird.
2. **Unit-Tests (optional):**
   Für ausgewählte Teile der Geschäftslogik (z. B. Validierungen oder Statuswechsel) können Unit-Tests vorgesehen werden.
   Die Unit-Tests dienen der Überprüfung einzelner Methoden unabhängig von der Benutzeroberfläche.