

# WorkFlow Assist

## Mobile-Applikation zur Erfassung und Verwaltung von Arbeitsplatz-Problemen

ÜK Modul 335 – Mobile-Applikation mit dem Ionic-Framework (Angular) realisieren

Name: Halil Cankilic

Klasse: INA23C

---

|   |    |
|---|----|
| <u>WorkFlow Assist</u>  | 1  |
| <u>Mobile-Applikation zur Erfassung und Verwaltung von Arbeitsplatz-Problemen</u> | 1  |
| <u>    1.1 Ausgangslage / Problemstellung</u>                                     | 1  |
| <u>    1.2 Zielsetzung</u>  | 2  |
| <u>    1.3 Zielgruppen</u>  | 2  |
| <u>2. User Stories</u>  | 2  |
| <u>3. Storyboard und UI-Elemente</u>  | 3  |
| <u>Seite: Home</u>  | 3  |
| <u>Seite: Ticket-Liste</u>  | 4  |
| <u>Seite: Ticket erstellen</u>  | 5  |
| <u>Seite: Ticket-Details</u>  | 6  |
| <u>Seite: Einstellungen</u>   | 7  |
| <u>4. Datenbank-Modell</u>  | 8  |
| <u>5. Technische Umsetzung</u>  | 9  |
| <u>6. Fazit</u>   | 10 |

---

**!!!Readme file lesen für Installation!!!**

<https://github.com/Eiermitsucuk/Modul335>

# **1. Projektbeschreibung**

## **1.1 Ausgangslage / Problemstellung**

In vielen Ausbildungsbetrieben werden technische, infrastrukturelle oder ergonomische Probleme über unstrukturierte Kanäle gemeldet – zum Beispiel per E-Mail oder Zuruf. Probleme entstehen durch Medienbrüche, fehlende Transparenz, lange Reaktionszeiten und Doppelspurigkeiten. Lernende haben zudem oft keinen PC zur Hand, was die Hemmschwelle erhöht.

## **1.2 Zielsetzung**

Entwicklung einer mobilen Ionic/Angular-Applikation zur schnellen und strukturierten Meldung von Arbeitsplatz-Problemen.

Zentrale Ziele:

- Standardisierung des Meldeprozesses
- Mobile-first Nutzung
- Transparenz über Status und Bearbeitung
- Reduzierte Reaktionszeit
- Reduktion von Medienbrüchen

## **1.3 Zielgruppen**

- Lernende
- IT-Support / Facility Management
- Ausbildungsverantwortliche

## **2. User Stories**

### **US1: Ticket erstellen**

Als Lernender möchte ich schnell ein Problem melden, damit es zeitnah behoben wird.

Akzeptanzkriterien: Titel, Beschreibung, Kategorie, GPS oder manueller Standort, Validierung, Offline-Speicherung, Speicherung in Datenbank.

### **US2: Tickets anzeigen**

Als Lernender möchte ich meine gemeldeten Tickets einsehen, damit ich den Überblick behalte.

Akzeptanzkriterien: Liste, Filter, Status-Badges, Detailansicht.

### **US3: Ticket-Details einsehen**

Als Lernender möchte ich Details sehen, um den Kontext zu verstehen.

Akzeptanzkriterien: Beschreibung, Standort, Karte, Zeitstempel, Löschen, Status.

### **US4: Ticket-Status ändern**

Als Support-Mitarbeiter möchte ich Status ändern, damit der Fortschritt sichtbar bleibt.

### **US5: GPS erfassen**

Als Lernender möchte ich meinen Standort automatisch erfassen können.

### **US6: Offline arbeiten**

Als Lernender möchte ich Tickets auch offline erstellen können.

### **US7: Benachrichtigungen**

Als Lernender möchte ich über wichtige Ereignisse informiert werden.

### **US8: Dark Mode**

Als Benutzer möchte ich zwischen hellem und dunklem Design wechseln.

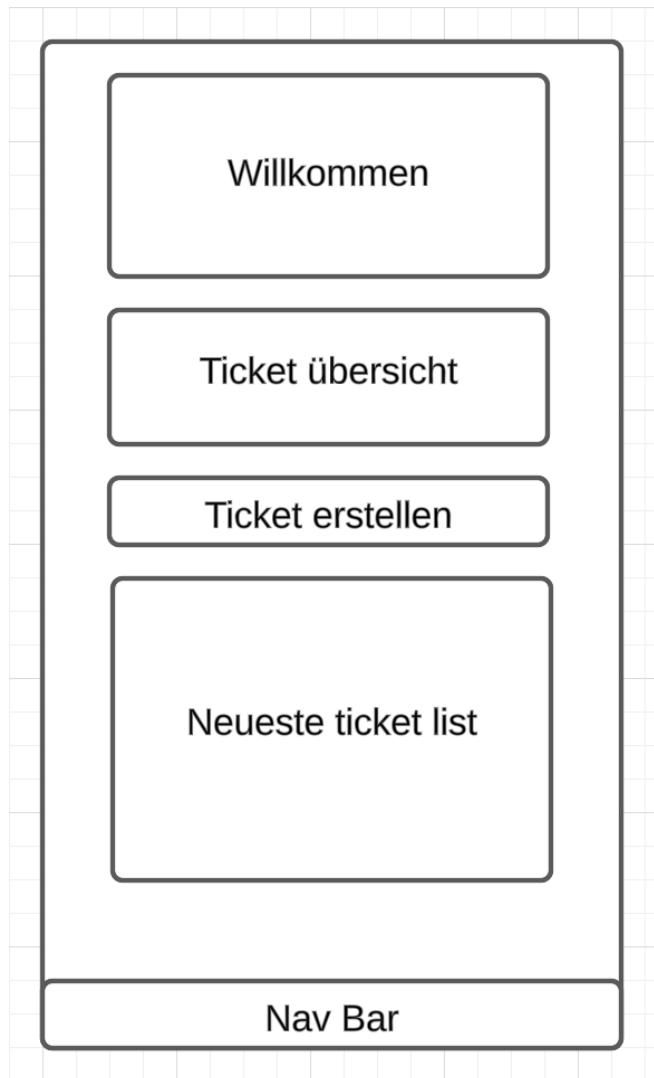
### **US9: Statistiken**

Als Ausbildungsverantwortlicher möchte ich Ticket-Trends sehen.

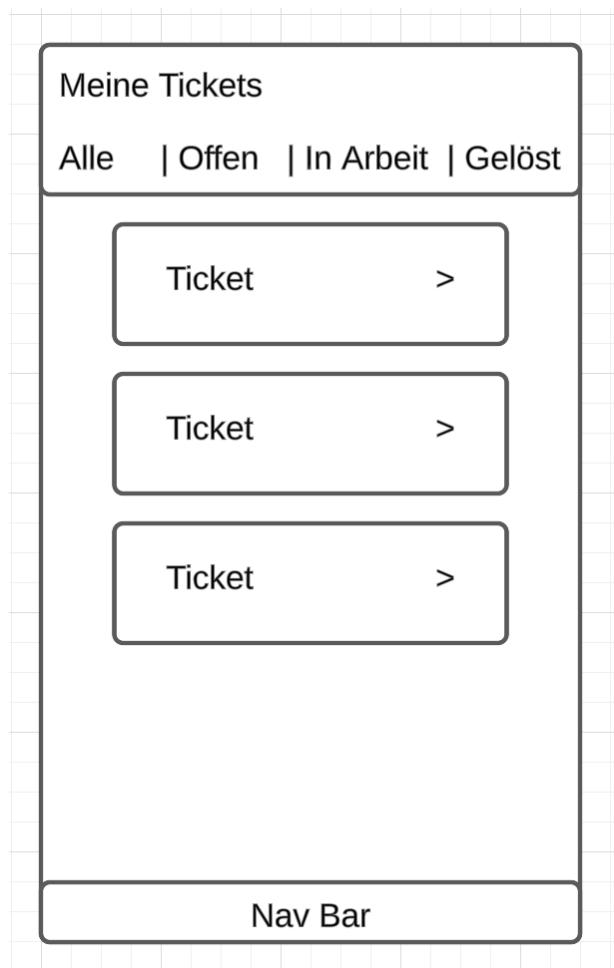
### 3. Storyboard und UI-Elemente

Die App basiert auf einer Tab-Navigation: Home, Tickets, Erstellen, Settings.

#### Seite: Home



## Seite: Ticket-Liste



## Seite: Ticket erstellen

A wireframe diagram of a 'Ticket erstellen' (Create Ticket) page. The page has a light gray header bar with a dark gray border containing the text '← Neues Ticket'. Below this are five input fields: 'Titel' (Title), 'Kategorie' (Category), 'Beschreibung' (Description), and 'Standort' (Location). At the bottom is a green rectangular button labeled 'Ticket erstellen' (Create Ticket). The entire form is set against a background of a light gray grid.

← Neues Ticket

Titel

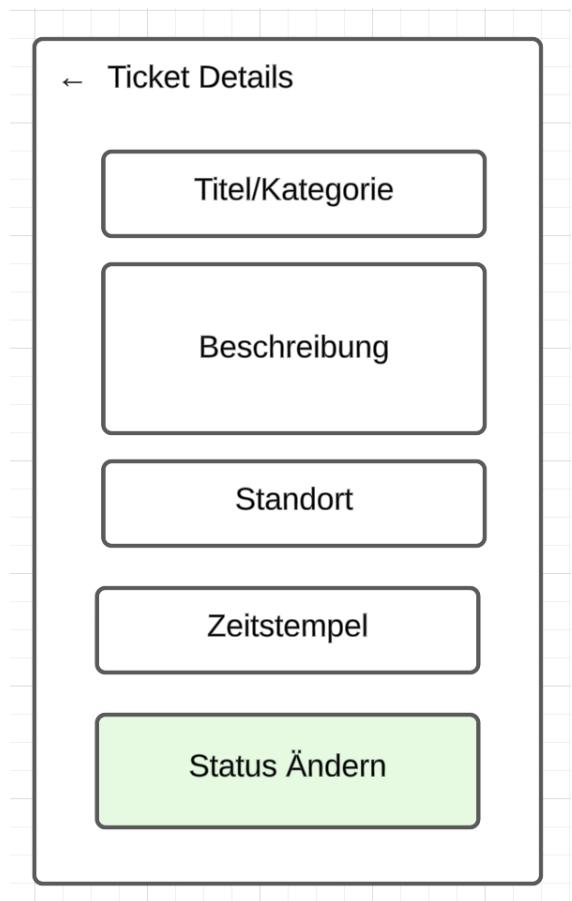
Kategorie

Beschreibung

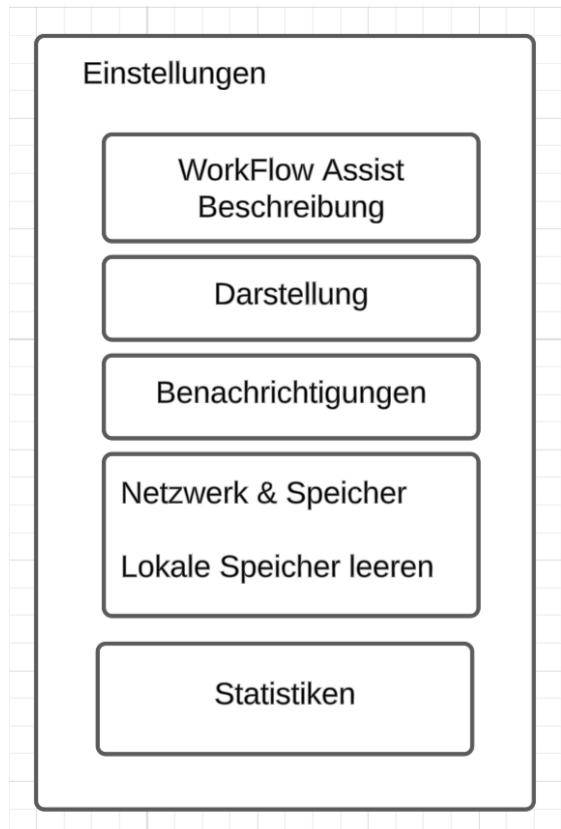
Standort

Ticket erstellen

## Seite: Ticket-Details



## Seite: Einstellungen



## 4. Datenbank-Modell

| tickets   |             |
|---|-------------|
| <b>id</b>   | UUID        |
| <b>title</b>   | TEXT NN     |
| <b>description</b>   | TEXT NN     |
| <b>category</b>    | TEXT NN     |
| <b>status</b>    | TEXT NN     |
| <b>location</b>    | TEXT        |
| <b>user_id</b>   | UUID        |
| <b>created_at</b>    | TIMESTAMPTZ |
| <b>updated_at</b>    | TIMESTAMPTZ |

## 5. Technische Umsetzung

**Frontend:** Ionic 8, Angular 20, SCSS.

**Backend:** Supabase (PostgreSQL + Storage + Realtime).

**Mobile:** Capacitor 7.4.

### Plugins:

- Geolocation
- Local Notifications
- Network

**Weitere Features:** Offline-Mode, Filter, Dark Mode, GPS, Statistik-Dashboard.

### Projektstruktur:

- Models
- Services (Supabase, GPS, Storage, Network, Notifications, Theme)
- Pages (Ticket-Create, Ticket-List, Ticket-Detail, Settings, Home, Tabs)
- Theme (variables.scss)

## **6. Fazit**

WorkFlow Assist erfüllt alle Anforderungen des ÜK Modul 335 vollständig. Die App ermöglicht intuitive Problemerfassung mit CRUD, GPS-Integration, Offline-Funktionalität und Dark Mode. Während der Entwicklung sammelte ich wertvolle Erfahrungen mit Capacitor-Plugins, Offline-First Strategien und Supabase-Integration. Besonders herausfordernd waren GPS-Implementierung und Online/Offline-Synchronisation. Das Projekt zeigt, dass mit Ionic und Angular professionelle, plattformübergreifende Apps effizient entwickelt werden können. Die App ist produktionsreif für den Einsatz in Ausbildungsbetrieben.