

Projekthandbuch
Tickify
001

Version 0.1
Projektleiter/in: Adnan Bajric
Datum: 07.02.2025

Inhalt

1	Projektpläne.....	5
1.1	Projektauftrag	5
1.2	Projektzieleplan	7
1.3	Projektorganigramm	8
1.4	Projektstrukturplan	9
1.5	Arbeitspaket-Spezifikationen	10
1.6	Projektmeilensteinplan	24
1.7	Projektbalkenplan	25
1.8	Projektkostenplan	26
1.9	Projekt-„Spielregeln“	27
1.10	Projektrisikoplan	28
1.11	Projektdokumentation	29
2	Projektstart.....	30
2.1	Protokolle – Projektstart	30
3	Projektkoordination	31
3.1	Abnahme Arbeitspakete	31
3.2	Protokolle – Projektkoordination.....	31
4	Projektcontrolling	32
4.1	Aktueller Projektfortschrittsbericht.....	32
4.2	Weitere Projektfortschrittsberichte	33
4.3	Protokolle – Projektcontrolling	33
5	Projektabschluss.....	34
5.1	Projektabschlussbericht.....	34
5.2	Protokolle – Projektabschluss.....	35

Änderungsverzeichnis

Versions- nummer	Datum	Änderung	Ersteller

1 Projektpläne

1.1 Projektauftrag

Tickify 001		PROJEKT-AUFTRAG							
Projektstartereignis: • 4.2.2025		Projektstarttermin: • 4.2.2025							
Inhaltliches Projektendereignis: • Formales Projektendereignis: •		Projektendtermine: • 03.03.2025 Zwischen Beurteilung 1 • 18.03.2025 Zwischen Beurteilung 2 • 14.03.2025 Doku Rohkonzept, • 28.03.2025 Doku PDF, 01.04.2025 Doku gebunden • 02.04 – 07.04 Präsentationen • 08.04 und 09.04 Endnote							
Projektziele: • Gestaltung einer benutzerfreundlichen UI • Sicherstellung der Systemstabilität durch Tests und Code-Reviews • KI-Integration • Entwicklung eines Rollensystems mit Berechtigungen		Nicht-Projektziele: • Integration von externen Zahlungsdiensten. • Werbung oder Marketingmaßnahmen. • Keine externe Authentifizierung							
Hauptaufgaben (Projektphasen): • • • •		Projektressourcen und –kosten*: <table border="1"> <thead> <tr> <th>Ressourcen-/Kostenart</th> <th>Mengen-einheit</th> <th>Kosten (in Euro)</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		Ressourcen-/Kostenart	Mengen-einheit	Kosten (in Euro)			
Ressourcen-/Kostenart	Mengen-einheit	Kosten (in Euro)							
ProjektauftraggeberIn: •		ProjektleiterIn: • Adnan Bajric							
Projektteam: • Adnan Bajric • Laurentiu Iordachi • •									

.....
Vorname Nachname, (ProjektauftraggeberIn)

.....
Vorname Nachname, (ProjektleiterIn)

* Kategorien für das Budget des gesamten Projektes möglich:

Kategorie A: bis 0,3 Mio €

Kategorie B: bis 1 Mio €

Kategorie C: bis 10 Mio €

Kategorie D: über 10 Mio €

1.2 Projektzieleplan

<div> <div> standard projekthandbuch 001 </div> <div> PROJEKTZIELE- PLAN </div> </div>		
Zielart	Projektziele	Adaptierte Projektziele per ...
Ziele: <ul style="list-style-type: none"> Hauptziele Zusatzziele 	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Nicht-Ziele	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">

1.3 Projektorganigramm

<div> <div>standard projekthandbuch 001</div> <div>PROJEKT- ORGANISATION</div> </div>		
Projektrolle	Aufgabenbereiche/Skills	Name
ProjektauftraggeberIn		
ProjektleiterIn		
Projektteam- mitglieder		
ProjektmitarbeiterInnen		

1.4 Projektstrukturplan

1.5 Arbeitspaket-Spezifikationen

Tickify 001	ARBEITSPAKET-SPEZIFIKATIONEN
4.0 Entwicklung Allgemein	<div data-bbox="520 483 1508 1357"> <p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • 4.1 Konzeptentwicklung: Entwicklung eines strukturierten Konzepts für das gesamte Ticketsystem, einschließlich der Benutzerinteraktionen und Workflow-Definitionen. • 4.2 Design: Erstellung eines ansprechenden und funktionalen Designs für die Benutzeroberfläche unter Berücksichtigung der Usability Prinzipien. • 4.3 Backend Plan: Planung der Backend Architektur, einschließlich Datenbankstruktur, API-Schnittstellen und Sicherheitsrichtlinien. • 4.4 APIs: Implementierung von API-Schnittstellen zur Kommunikation zwischen Frontend, Backend und externen Diensten. • 4.5 Dokumentation: Erstellung einer umfassenden technischen Dokumentation, die Architektur, API-Endpunkte und Funktionsweise beschreibt. • 4.6 Deadlines checken: Regelmäßige Überprüfung von Meilensteinen und Deadlines, um die Projektzeitpläne einzuhalten. • 4.7 Bugs fixen: Identifikation und Behebung von Fehlern im System zur Sicherstellung einer stabilen und sicheren Anwendung. • 4.8 Deployment: Bereitstellung der Anwendung in einer produktiven Umgebung unter Berücksichtigung der Skalierbarkeit und Performance. • 4.9 Hosting: Einrichtung und Verwaltung des Hostings, einschließlich Serverkonfiguration, Sicherheitseinstellungen und Wartung. </div> <div data-bbox="520 1357 1508 1603"> <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung von Features, die nicht im ursprünglichen Projektumfang enthalten sind. • Erstellung von Marketingmaterialien oder Werbung. • Aufbau eines separaten CRM-Systems. </div> <div data-bbox="520 1603 1508 2092"> <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein vollständiges, funktionierendes System mit klarer Architektur und definierten Workflows. • Ein konsistentes und benutzerfreundliches Design, das die Usability verbessert. • Eine dokumentierte und getestete API für die Kommunikation zwischen den Systemkomponenten. • Eine umfassende technische Dokumentation für Entwickler und Administratoren. • Ein stabiles, fehlerfreies und produktionsbereites System. • Ein gehostetes System, das für die Nutzer zugänglich ist. </div>

	<p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Abschluss von Konzept- und Designphasen gemäß den definierten Projektmeilensteinen. • Erfolgreiche Implementierung und Tests der API-Schnittstellen. • Regelmäßige Statusmeetings zur Überprüfung der Einhaltung von Deadlines. • Durchführung und Dokumentation von Code-Reviews und Fehleranalysen. • Bereitstellung des Systems auf der produktiven Umgebung und Überprüfung der Funktionsfähigkeit nach dem Deployment. • Überwachung der Hosting-Umgebung zur Sicherstellung der Verfügbarkeit und Performance.
<p>4.1 Konzeptentwicklung</p>	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung eines strukturierten und skalierbaren Konzepts für das Ticketsystem basierend auf dem MVC-Modell. • Definition der Benutzerrollen und Interaktionsmöglichkeiten. • Erstellung eines Workflows für Ticket-Erstellung, Bearbeitung und Abschluss. • Berücksichtigung von Sicherheitsaspekten und Datenschutzrichtlinien. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung von Prototypen oder UI-Designs. • Implementierung von Backend- oder Frontend-Komponenten. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein vollständiges Konzeptsdokument mit detaillierten Workflows und einer klaren MVC-Struktur. • Klare Definition der Benutzerrollen und Rechte. • Strukturierte Prozessabläufe für Ticketverwaltung. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Abnahme des Konzeptsdokuments durch das Team. • Durchführung von Feedbackrunden zur Optimierung des Konzepts. • Dokumentierte Änderungen und Anpassungen am Konzept.
<p>4.2 Design</p>	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung eines ansprechenden und funktionalen UI-Designs. • Nutzung von UX-Prinzipien zur Verbesserung der Benutzererfahrung. • Erstellung von Mockups und Prototypen. • Definition von Farb- und Typografiekonzepten. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Implementierung der Designs im Code. • Erstellung von Animationen oder komplexen UI-Interaktionen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein vollständiges Designkonzept mit Mockups und Prototypen. • Konsistente visuelle Identität für das Ticketsystem. • Dokumentierte Designrichtlinien für die Entwicklung. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Validierung der Mockups durch Stakeholder. • Durchführung von Usability-Tests mit Testnutzern. • Dokumentierte Designentscheidungen und Anpassungen.

4.3 Backend Plan	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Planung der Backend-Architektur und Datenbankstruktur. Definition der API-Schnittstellen für die Kommunikation mit dem Frontend. Festlegung von Sicherheitsrichtlinien und Authentifizierungsmechanismen. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Entwicklung des Frontends. Implementierung von Features, die nicht zum Kern des Systems gehören. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Eine detaillierte Backend-Architektur mit klar definierten Schnittstellen. Ein sicheres Authentifizierungs- und Berechtigungskonzept. Dokumentierte Datenbankstruktur und API-Spezifikationen. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Validierung der Architektur durch Code-Reviews. Erstellung von API-Tests zur Sicherstellung der Funktionalität. Durchführung von Performance-Tests zur Überprüfung der Skalierbarkeit.
4.4 APIs	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Implementierung von API-Schnittstellen zur Kommunikation zwischen Frontend, Backend und externen Diensten. Sicherstellung der API-Sicherheit durch Authentifizierung und Autorisierung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Entwicklung von APIs ohne Sicherheitsrichtlinien. Implementierung von nicht dokumentierten Endpunkten. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Vollständig dokumentierte und getestete APIs. Sicherer Datenaustausch zwischen Systemkomponenten. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Durchführung von API-Tests. Validierung der API-Funktionalität mit Mock-Daten. <p>Dokumentation der API-Endpunkte und deren Nutzung.</p>
4.5 Dokumentation	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Erstellung einer umfassenden technischen Dokumentation. Beschreibung der Architektur, API-Endpunkte und Funktionsweise. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <p>Erstellung von Benutzerhandbüchern für Endanwender.</p> <ul style="list-style-type: none"> Dokumentation von nicht implementierten Features. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Vollständig ausgearbeitete und verständliche Dokumentation. Klare Übersicht über Architektur und API-Endpunkte. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Abnahme der Dokumentation durch das Team. Durchführung von Feedbackrunden zur Verbesserung der Verständlichkeit

4.7 Bugs fixen	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Identifikation und Behebung von Fehlern. • Durchführung von Tests zur Fehlervermeidung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Einführung von neuen Features während der Fehlerbehebung. • Temporäre Workarounds ohne langfristige Lösungen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein stabiles und fehlerfreies System. • Dokumentierte Fehlerbehebungen für Nachvollziehbarkeit. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführung und Auswertung von Fehlerberichten. • Erfolg der Fehlerbehebung durch Re-Tests.
4.8 Deployment	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Bereitstellung der Anwendung in einer produktiven Umgebung. • Überwachung und Optimierung der Performance. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Bereitstellung ohne vorherige Tests. • Manuelles Deployment ohne Automatisierung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein live geschaltetes, performantes System. • Dokumentierte Deployment-Prozesse. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Erfolg der Bereitstellung durch System-Monitoring. • Durchführung von Last- und Performance-Tests.
5.0 Main Page Entwicklung	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • 5.1 Form: Entwicklung eines einfachen und benutzerfreundlichen Formulars zur Ticketerstellung. • 5.2 Design: Gestaltung einer klaren UI mit intuitiver Navigation und optimierter Nutzerführung. • 5.3 Backend Dev: Implementierung der Backend-Logik zur Verarbeitung von Ticketanfragen. • 5.4 Frontend Dev: Umsetzung des Frontends mit responsivem Design und dynamischen UI-Elementen. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Integration von externen Zahlungsdiensten. • Werbung oder Marketingmaßnahmen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Funktionierende Hauptseite mit Ticketformular.

	<ul style="list-style-type: none"> • Vollständige Projektdokumentation. • Behebung gemeldeter Fehler. • Testergebnisse zur Validierung der UI/UX-Funktionalität.
5.1 Form	<p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Wöchentliche Code-Reviews. • Usability-Tests zur Optimierung der Benutzerfreundlichkeit. • Einhaltung geplanter Entwicklungsmeilensteine. • Abschluss der Dokumentation und interne Freigabe. <p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung des Ticketformulars mit PHP für die Verarbeitung von Anfragen. • Serverseitige Validierung der Pflichtfelder zur Fehlervermeidung. • Anbindung des Formulars an die MySQL-Datenbank über POST-Request. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Integration mit externen Ticketing-Systemen. • Keine Speicherung unverschlüsselter Benutzerdaten. • Keine Automatisierung ohne Nutzerinteraktion. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein funktionales und sicheres Formular für die Ticketerstellung. • Erfolgreiche Datenvalidierung und Speicherung. • Verbindung mit der Datenbank zur Verwaltung der Tickets. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Manuelle Tests zur Überprüfung der Validierung. • Test der Verbindung zwischen Formular und Backend. • Review der Funktionalität durch verschiedene Testnutzer.
5.2 Design	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Gestaltung der Benutzeroberfläche mit Tailwind CSS. • Sicherstellung der Responsivität für verschiedene Bildschirmgrößen. • Anwendung moderner UI/UX-Prinzipien zur intuitiven Nutzung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine individuellen Anpassungen für einzelne Nutzer. • Keine vollständige Branding-Strategie. • Keine unnötigen Animationen ohne Mehrwert. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine optisch ansprechende und gut strukturierte Hauptseite.

	<ul style="list-style-type: none"> • Responsive Design, das auf verschiedenen Geräten gut funktioniert. • UI-Komponenten nach aktuellen Usability-Standards. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Überprüfung der UI/UX-Standards. • Responsiveness-Tests auf unterschiedlichen Geräten. • Nutzerfeedback zur Optimierung des Designs.
5.3 Backend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Implementierung der Backend-Logik zur Verarbeitung von Ticketanfragen. • Erstellung sicherer API-Endpunkte zur Datenübertragung. • Speicherung und Verwaltung der Tickets in der MySQL-Datenbank. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Entwicklung eines komplexen Berechtigungssystems. • Keine Integration mit Drittanbieter-Diensten. • Keine Verarbeitung von Zahlungsinformationen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine funktionale Backend-Logik zur Verarbeitung von Tickets. • Sichere Speicherung der Tickets in der Datenbank. • Dokumentierte API-Schnittstellen für zukünftige Erweiterungen. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführung von API-Tests zur Überprüfung der Endpunkte. • Sicherstellung der Datenbankintegrität durch Unit-Tests. • Review der Backend-Sicherheit durch Testdurchläufe.
5.4 Frontend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung der Benutzeroberfläche mit Tailwind CSS. • Implementierung interaktiver UI-Elemente zur Ticketanzeige. • Sicherstellung einer flüssigen Navigation und Nutzerführung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Implementierung unnötiger visueller Effekte. • Keine Integration von Werbeanzeigen. • Keine externe Plugin-Nutzung ohne Validierung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine nutzerfreundliche und interaktive Hauptseite. • Optimierte UI-Komponenten für eine einfache Bedienung.

	<ul style="list-style-type: none"> • Verbindung des Frontends mit der Backend-Logik. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführung von UI-Tests mit Testnutzergruppen. • Sicherstellung der reibungslosen Kommunikation mit dem Backend. • Code-Review zur Optimierung der Frontend-Performance. •
6.0 Ticket System Entwicklung	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • 6.1 Form: Implementierung eines strukturierten Ticketformulars mit Pflichtfeldern und Validierungen. • 6.2 Design: Erstellung eines übersichtlichen Dashboards für die Ticketverwaltung. • 6.3 Rechte: Entwicklung eines Rollensystems mit spezifischen Berechtigungen (User, Moderator, Admin). • 6.4 Backend Dev: Anbindung der Ticketverwaltung an die Datenbank mit CRUD Funktionen. • 6.5 Frontend Dev: Entwicklung der Benutzeroberfläche zur Anzeige und Bearbeitung von Tickets. • 6.6 AI Integration: Implementierung von AI um die Bearbeitung von Tickets zu erleichtern/beschleunigen.
	<p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung einer mobilen App. • Automatisierte Entscheidungsfindung für Ticketbearbeitung. • Werbung oder externe Marketingstrategien.
	<p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Vollständig funktionierendes Ticket-System mit allen relevanten Funktionen. • Rollenspezifischer Zugriff und Rechteverwaltung. • Getestete und stabile Backend-Logik. • Detaillierte Dokumentation über Systemabläufe und API-Schnittstellen.
	<p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Abgleich der implementierten Features mit den ursprünglichen Anforderungen. • Automatisierte Unit- und Integrationstests zur Sicherstellung der Systemstabilität. • Code-Reviews und Performance-Optimierungen. • Finaler Deployment-Check.
6.1 Form	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung eines Formulars mit Pflichtfeldern für die Ticketerstellung.

	<ul style="list-style-type: none"> Validierung der Nutzereingaben zur Fehlervermeidung. Anbindung des Formulars an das Backend für die Ticketverwaltung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Keine Integration externer Formular-Plugins. Keine automatische Ticketbearbeitung ohne menschliche Kontrolle. Keine Nutzung externer Datenquellen zur Ticketerstellung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Ein funktionales, fehlerfreies Ticketformular. Validierte Nutzereingaben mit Fehlermeldungen bei falscher Eingabe. Verbindung des Formulars mit der Datenbank zur Speicherung der Tickets. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Durchführung von Usability-Tests. Sicherstellung der Validierungsregeln durch Testfälle. Erfolgreiche Integration mit der Ticket-Datenbank.
6.2 Design	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Erstellung eines übersichtlichen Dashboards zur Ticketverwaltung. Umsetzung einer klaren UI-Struktur mit Tailwind CSS. Sicherstellung der Barrierefreiheit und Nutzerfreundlichkeit. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Keine individuellen Designs für einzelne Nutzer. Keine Integration von UI-Elementen ohne Funktionalität. Keine vollständige Rebranding-Strategie. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Ein funktionales, benutzerfreundliches Ticket-Dashboard. Klare visuelle Hierarchie zur schnellen Bearbeitung von Tickets. Dokumentierte Designrichtlinien zur Weiterentwicklung. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Überprüfung der UI/UX mit Testnutzergruppen. Sicherstellung der Responsivität auf verschiedenen Geräten. Feedbackschleifen zur Optimierung des Designs.
6.3 Rechte	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Entwicklung eines Rollensystems mit spezifischen Berechtigungen.

	<ul style="list-style-type: none"> • Implementierung von Admin-, Moderator- und Nutzerrechten. • Sicherstellung der Zugriffskontrollen durch Validierung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine freie Rechtevergabe durch Nutzer. • Keine übermäßige Einschränkung der User-Funktionalität. • Keine Implementierung von Drittanbieter-Sicherheitsdiensten. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein getestetes und funktionierendes Berechtigungssystem. • Dokumentierte Rollen und deren Zugriffsrechte. • Sichere Verwaltung der Benutzerzugriffe. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Test der Rollenvergabe in verschiedenen Szenarien. • Durchführung von Sicherheitstests. • Review und Anpassung der Rollen bei Bedarf.
6.4 Backend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung der Ticketverwaltung in der Datenbank. • Erstellung sicherer API-Endpunkte für CRUD-Funktionen. • Implementierung von Fehler-Handling und Datenvalidierung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Speicherung sensibler Nutzerdaten ohne Verschlüsselung. • Keine Bereitstellung offener API-Endpunkte ohne Authentifizierung. • Keine Integration von Drittanbieter-Backendsystemen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein funktionierendes und sicheres Backend. • API-Schnittstellen für die Frontend-Kommunikation. • Dokumentierte Backend-Architektur. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführen von API-Tests. • Sicherstellen der Datenbankintegrität durch Tests. • Code-Review zur Optimierung der Performance.
6.5 Frontend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung der Benutzeroberfläche zur Anzeige und Bearbeitung von Tickets.

	<ul style="list-style-type: none"> • Integration der Frontend-Komponenten mit dem Backend. • Sicherstellung der Responsivität mit Tailwind CSS. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Entwicklung einer eigenständigen Mobile-App. • Keine Animationen ohne Funktionalität. • Keine externe Plugin-Integration ohne Absprache. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein funktionales und nutzerfreundliches Ticket-Frontend. • Direkte Verbindung mit dem Backend für Echtzeit-Updates. • Dokumentierte Frontend-Struktur für Erweiterungen. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • UI-Tests mit Testnutzern. • Sicherstellung der reibungslosen Kommunikation mit dem Backend. • Code-Review zur Verbesserung der Frontend-Performance.
6.6 AI Integration	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Implementierung von AI zur Erleichterung der Ticketbearbeitung. • Automatische Klassifizierung und Priorisierung von Tickets. • Machine Learning-Modelle zur Erkennung wiederkehrender Probleme. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine vollständige Automatisierung ohne menschliche Kontrolle. • Keine Nutzung externer AI-Services ohne Datenschutzprüfung. • Keine selbstlernenden Algorithmen ohne Sicherheitsprüfung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein AI-gestütztes System zur Unterstützung der Ticketbearbeitung. • Dokumentierte und getestete AI-Funktionalitäten. • Verbesserte Effizienz durch automatische Kategorisierung. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Erfolgreiche Tests der AI-Modelle. • Manuelle Validierung der AI-Vorschläge. • Performance-Monitoring und Optimierung der AI-Integration.
	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p>

7.0 Benutzerverwaltung	<ul style="list-style-type: none"> • 7.1 Form: Implementierung von Anmelde- und Registrierungsformulären mit Validierung. • 7.2 Design: Gestaltung einer nutzerfreundlichen Verwaltungsoberfläche. • 7.3 Liste mit Users Rechte: Entwicklung einer Benutzerverwaltung mit anpassbaren Rollen und Berechtigungen. • 7.4 Verwaltung: Bereitstellung von Admin-Tools zur Verwaltung von Nutzerdaten und Zugriffsrechten. • 7.5 Backend Dev: Implementierung der Benutzerverwaltung in der Datenbank. • 7.6 Frontend Dev: Umsetzung des Frontends zur einfachen Verwaltung von Benutzern und Rollen. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Integration von externen Zahlungsdiensten. • Werbung oder Marketingmaßnahmen. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein funktionierendes System zur Verwaltung von Benutzern, inklusive Rollenrechten. • Eine vollständige und aktuelle Dokumentation. • Erfolgreiche Durchführung von Sicherheitstests zur Überprüfung der Zugriffsbeschränkungen. • Implementierung einer Administrationsansicht für Admins und Moderatoren. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Abschluss der Implementierung nach den vordefinierten Anforderungen. • Durchführung von Sicherheitstests und Überprüfung auf potenzielle Sicherheitslücken. • Regelmäßige interne Tests zur Benutzerfreundlichkeit und Performance. • Code-Reviews und finale Freigabe zur Produktivsetzung.
7.1 Form	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung eines sicheren Anmelde- und Registrierungsformulars. • Implementierung von Validierungen zur Fehlervermeidung. • Integration von Passwortverschlüsselung. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine externe Authentifizierung über Drittanbieter. • Keine Speicherung von Klartext-Passwörtern. • Keine Entwicklung einer App-Version. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Ein funktionierendes, sicheres Formular für Anmeldung und Registrierung. • Validierte Nutzereingaben mit klarer Fehleranzeige.

	<ul style="list-style-type: none"> • Verschlüsselte Speicherung der Passwörter in der Datenbank. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Überprüfung der Funktionalität durch Unit-Tests. • Manuelle Tests durch verschiedene Testnutzer. • Sicherstellung der Einhaltung von Datenschutzrichtlinien.
7.2 Design	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Gestaltung einer nutzerfreundlichen Verwaltungsoberfläche. • Erstellung eines konsistenten UI-Designs mit Tailwind CSS. • Implementierung eines responsiven Layouts für verschiedene Endgeräte. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Kein Fokus auf Branding oder Marketingdesign. • Keine Implementierung komplexer Animationen. • Keine eigenständige mobile App. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine intuitive, optisch ansprechende Verwaltungsoberfläche. • Konsistente UI-Elemente mit klarer Nutzerführung. • Responsive Anpassung an verschiedene Bildschirmgrößen. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführung von UI/UX-Tests. • Überprüfung der responsiven Darstellung. • Feedback aus Testnutzungen zur Verbesserung des Designs.
7.3 Liste mit Users Rechte	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Entwicklung einer Benutzerverwaltung mit anpassbaren Rollen. • Erstellung einer Benutzerliste mit Such- und Filteroptionen. • Implementierung von Zugriffsrechten basierend auf Nutzerrollen. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Integration in externe Systeme. • Keine manuelle Verwaltung von Berechtigungen durch Nutzer. • Keine komplexe Berechtigungslogik mit KI-Unterstützung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine vollständige Benutzerverwaltung mit anpassbaren Rollen. • Zugriffskontrolle durch klar definierte Berechtigungen.

	<ul style="list-style-type: none"> Nutzerfreundliche Listenansicht mit Such- und Filterfunktionen. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Test der Benutzerrechte durch verschiedene Nutzerrollen. Sicherstellung der korrekten Funktion der Zugriffskontrollen. Code-Review und Sicherheitsprüfung der Berechtigungen.
7.4 Verwaltung	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Bereitstellung von Admin-Tools zur Verwaltung von Nutzerdaten. Möglichkeit zur Sperrung und Löschung von Benutzerkonten. Verwaltung von Rollen und Berechtigungen durch Administratoren. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Keine automatische Löschung von Konten ohne Admin-Intervention. Keine Integration mit sozialen Netzwerken. Keine Echtzeit-Kollaboration für Admins. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Eine funktionierende Administrationsoberfläche. Klare Trennung der Admin- und Benutzerrechte. Protokollierung von Verwaltungsaktionen für Nachvollziehbarkeit. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> Test der Admin-Funktionen durch verschiedene Szenarien. Sicherstellung der Datenintegrität nach Änderungen. Überprüfung der Zugriffskontrollen durch Sicherheitstests.
7.5 Backend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> Implementierung der Benutzerverwaltung in der Datenbank. Erstellung sicherer API-Endpunkte zur Benutzerverwaltung. Verschlüsselung und sichere Speicherung von Benutzerdaten. <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> Keine Integration externer Authentifizierungsdienste. Keine Verarbeitung von Zahlungsinformationen. Keine offene API ohne Authentifizierung. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> Ein sicher implementiertes Backend für die Benutzerverwaltung. Funktionierende Schnittstellen für Anmeldung und Verwaltung.

	<ul style="list-style-type: none"> • Sicher gespeicherte Nutzerdaten mit Verschlüsselung. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Durchführen von API-Tests und Sicherheitsprüfungen. • Sicherstellen der Datenbankintegrität durch Tests. • Erfolgreiche Kommunikation mit dem Frontend.
7.6 Frontend Dev	<p>AP-Inhalt (<i>Was soll getan werden?</i>)</p> <ul style="list-style-type: none"> • Umsetzung des Frontends zur einfachen Verwaltung von Benutzern und Rollen. • Entwicklung interaktiver UI-Elemente für die Verwaltung. • Anzeige und Bearbeitung von Benutzerinformationen. • <p>AP-Nicht-Inhalte (<i>Was soll nicht getan werden?</i>)</p> <ul style="list-style-type: none"> • Keine Entwicklung einer eigenständigen Mobile-App. • Keine Integration von Drittanbieter-Plugins. • Keine erweiterte Datenanalyse. <p>AP-Ergebnisse (<i>Was liegt nach Beendigung des Arbeitspaketes vor?</i>)</p> <ul style="list-style-type: none"> • Eine funktionierende Benutzerverwaltungs-Oberfläche. • Nutzerfreundliche Interaktionen mit CRUD-Funktionalitäten. • Klare UI mit einfacher Bedienbarkeit. <p>AP-Leistungsfortschrittsmessung (<i>Wie wird der Fortschritt gemessen?</i>)</p> <ul style="list-style-type: none"> • Test der UI mit realen Nutzern. • Sicherstellung der responsiven Darstellung. • Erfolgreiche Kommunikation mit dem Backend.

1.6 Projektmeilensteinplan

<div> <div>standard projekthandbuch 001</div> <div>PROJEKT- MEILENSTEINPLAN</div> </div>				
PSP-Code	Meilenstein	Basis- termine	Aktuelle Plantermine	Ist Termine

*Termine chronologisch nach Planterminen reihen!

1.7 Projektbalkenplan

1.8 Projektkostenplan

<div> <div>standard projekthandbuch</div> <div>001</div> <div>PROJEKT-KOSTENPLAN</div> <div></div> </div>					
PSP-Code, AP-Bezeichnung	Kostenart	Plankosten	Adaptierte Plankosten per	Istkosten	Kostenabweichung
	• Personal				
	• Material				
	• Fremdleistungen				
	• Sonstige				
	Gesamt				
	• Personal				
	• Material				
	• Fremdleistungen				
	• Sonstige				
	Gesamt				
	• Personal				
	• Material				
	• Fremdleistungen				
	• Sonstige				
	Gesamt				
Projektkosten					

1.9 Projekt-„Spielregeln“

1.11 Projektdokumentation

Bereich	Beschreibung
Ablage	
Zugriffs- berechtigung	
Namenskonvention	
Spielregeln	

2 Projektstart

2.1 Protokolle – Projektstart

2.1.1 Projektstart-Workshop

2.1.2 Follow-up-Workshop

2.1.3 Projektauftraggeber-Sitzung

3 Projektkoordination

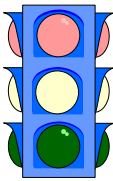

3.1 Abnahme Arbeitspakete

<div> <div> standard projekthandbuch 001 </div> <div> ABNAHME ARBEITSPAKETE </div> </div>					
PSP-Code	Arbeitspaket	AP-Verantw.	Datum	Abnahme durch	Unterschrift

3.2 Protokolle – Projektkoordination

4 Projektcontrolling

4.1 Aktueller Projektfortschrittsbericht

standard projekthandbuch 001		PROJEKT- FORTSCHRITTSBERICHT per	
 Projektkrise Projekt in Schwierigkeiten Projekt planmäßig 		1) Gesamtstatus •	
2) Status Ziele • •		Maßnahmen: • •	
3) Status Leistungsfortschritt • •		Maßnahmen: • •	
4) Status Termine • •		Maßnahmen: • •	
5) Status Ressourcen/Kosten • •		Maßnahmen: • •	
6) Status Kontext • •		Maßnahmen: • •	
7) Status Organisation/Kultur • •		Maßnahmen: • •	

4.2 Weitere Projektfortschrittsberichte

4.3 Protokolle – Projektcontrolling

4.3.1 Projektcontrolling-Sitzungen

4.3.2 Projektauftraggeber-Sitzungen

5 Projektabschluss

5.1 Projektabschlussbericht

standard projekthandbuch 001		PROJEKT- ABSCHLUSSBERICHT	
1) Gesamteindruck		2) Reflexion: Zielereichung	
3) Reflexion: Leistungen/Termine			
4) Reflexion: Ressourcen/Kosten			
5) Reflexion: Interne Organisation/ Umweltbeziehungen			
6) Leistungsbeurteilung (ProjektauftraggeberIn, ProjektleiterIn, ProjektmitarbeiterIn)		7) Lessons learned (Zusammenfassende Erfahrungen und Verbesserungsvorschläge)	
8) Planung Nachprojektphase, Restaufgaben			
To-Do		Zuständigkeit	Termin
9) Projektabnahme			
..... Vorname Nachname, (ProjektauftraggeberIn)	 Vorname Nachname, (ProjektleiterIn)	

5.2 Protokolle – Projektabschluss

5.2.1 Projektabschluss-Workshop