

Moderne Software- entwicklung

27. OKTOBER

889684

Marcell Wolf de Lêu



Übung 3: Systemarchitektur Ihres Projektes modellieren

Systemarchitektur für das Projekt „Mitfahrten-Management für Zugpersonal“

1. Event Storming durchführen

Um ein gemeinsames Verständnis der Domäne zu schaffen, habe ich eine Event Storming-Session durchgeführt und die wichtigsten Events in meinem System identifiziert. Diese sind:

- Zugpersonal registriert: Das Zugpersonal wird durch berechtigte Rollen im System neu registriert.
- Zugpersonal Stammdaten bearbeitet: Änderungen an den Stammdaten des Zugpersonals, durchgeführt von berechtigten Nutzern.
- Mitfahrt Gruppenleiter durchgeführt: Ein Gruppenleiter führt eine Mitfahrt durch und dokumentiert diese im System.
- Mitfahrt BOStrab durchgeführt: Eine BOStrab-Mitfahrt, durchgeführt von der Betriebsaufsicht, die zusätzliche Anforderungen und Erfassungen beinhaltet.
- Mitfahrten auswerten und filtern/sortieren: Nutzer haben die Möglichkeit, Daten zu Mitfahrten nach bestimmten Kriterien zu filtern und auszuwerten.

Ziel

Die Event Storming-Session ermöglicht eine klare Darstellung der Geschäftsprozesse und hilft dabei, alle wesentlichen Events des Systems zu erfassen.

2. Domänenmodell erstellen

Auf Basis der Events habe ich ein Domänenmodell entworfen, das die Kernkonzepte meines Projekts beschreibt und deren Beziehungen zueinander verdeutlicht. Die wichtigsten Entitäten sind:

- Zugpersonal: Diese Entität repräsentiert alle relevanten Stammdaten des Zugpersonals, einschließlich Informationen zu ihrem Profil (Klein- oder Großprofil) und den zugeordneten Bezirken.
- Mitfahrt: Jede Mitfahrt enthält Informationen zum Datum, zur Uhrzeit, zum Typ (Gruppenleiter- oder BOStrab-Mitfahrt) und zum Verantwortlichen. Die BOStrab-

Mitfahrten enthalten zusätzliche Details wie Fahrzeuge und Bemerkungen zur Fahrweise.

- Verantwortlicher: Eine Entität für die Person, die eine Mitfahrt durchführt (entweder Gruppenleiter oder Betriebsaufsicht), inklusive deren Rolle und Bezirk.

Ziel

Das Domänenmodell bietet eine klar strukturierte Darstellung der Kernkonzepte und zeigt, wie sie miteinander verbunden sind.

3. Bounded Contexts identifizieren

Basierend auf der Analyse der Domäne habe ich verschiedene Bounded Contexts definiert, um die Komplexität zu reduzieren und Verantwortlichkeiten klar zu trennen.

- Verwaltung Zugpersonal: Zuständig für das Erfassen und Bearbeiten von Zugpersonal-Daten.
- Mitfahrten-Management: Verwaltet alle Mitfahrten-Informationen, einschließlich der spezifischen Anforderungen für Gruppenleiter- und BOStrab-Mitfahrten.
- Reporting und Dashboard: Ermöglichen die Filterung und Auswertung der Mitfahrtdaten sowie den CSV-Export für spätere Analysen.

Ziel

Die Bounded Contexts schaffen eine klare Trennung der Verantwortlichkeiten und vereinfachen die spätere Implementierung des Systems.

4. Entitäten und Aggregates definieren

In jedem Bounded Context habe ich die relevanten Entitäten und Aggregates identifiziert und deren Attribute und Methoden definiert.

- Zugpersonal: Umfasst Attribute wie Name, Profil, Bezirkszuordnung und das Datum der letzten BOStrab-Mitfahrt.
- Mitfahrt: Umfasst Details wie Datum, Uhrzeit, Typ (Gruppenleiter- oder BOStrab-Mitfahrt), sowie die spezifischen Anforderungen bei BOStrab-Mitfahrten (Fahrzeuge, Beurteilung der Fahrweise).
- Verantwortlicher: Enthält Informationen zum Namen und zur Rolle (Gruppenleiter oder Betriebsaufsicht) und wird mit Mitfahrten verknüpft.

Ziel

Ein detailliertes Verständnis der Datenstruktur und Logik in meiner Anwendung.

5. Domain Services und Repositories

Für die Verwaltung der Entitäten habe ich notwendige Domain Services und Repositories identifiziert, die die Geschäftslogik und Datenpersistenz gewährleisten.

Domain Services

- ZugpersonalService: Verwalten der Registrierung und Bearbeitung der Zugpersonal-Daten.
- MitfahrtenService: Verwaltung und Dokumentation aller Mitfahrten (Gruppenleiter- und BOStrab-Mitfahrten).
- ReportingService: Bereitstellung von Auswertungen und CSV-Exportmöglichkeiten.

Repositories

- ZugpersonalRepository: Verwaltet die Speicherung und den Zugriff auf Zugpersonal-Daten.
- MitfahrtenRepository: Speichert und verwaltet die Daten zu allen Mitfahrten.

Ziel

Klare Definition der Services und Repositories, die die Daten persistieren und Geschäftslogik abwickeln.

6. Implementierungsstrategie

Für die Implementierung der Domain-Modelle habe ich eine Strategie entwickelt, die die Aspekte der Testbarkeit und Modularität berücksichtigt.

- Entitäten und Aggregates umsetzen: Jede Entität wird als separate PHP-Klasse implementiert, die alle Attribute und Methoden zur Verarbeitung der Geschäftslogik enthält.
- Domain Services implementieren: Für jeden Domain Service wird eine Service-Klasse erstellt, die mit den Repositories zusammenarbeitet und die Geschäftslogik ausführt.
- Repositories implementieren: Jedes Repository wird als Interface definiert, das Methoden zur Speicherung und zum Abrufen der Entitäten enthält.

Ziel

Eine strukturierte Herangehensweise an die Codierung, die die Prinzipien von Domain-Driven Design berücksichtigt.