

Kontrakte

Nummerierung nach Use Cases und interne System Sequenz Diagramm Nummer
(Beispiel: 1.2.1 -> Use Case 1.2 und SSD 1. Schritt)

1. Allgemein

Kontrakt 1.1 AnzeigenGesamtstatus

Anmerkung: Da es diverse Kontrakte zur Anzeige einer bestimmten Übersicht gibt, werden diese für den Gesamtstatus exemplarisch komplett ausformuliert. In allen anderen Anzeigen werden nur noch Operation und Ergebnisse und Querverweise des Anzeigenkontrakts definiert. Dafür werden aber die Ergebnisse ausführlicher beschrieben um ein Bild der Rückgabeinformationen zu bekommen.

Operation: AnzeigenGesamtstatus()

Beschreibung: Die Übersicht Gesamtstatus anzeigen

Vorbedingung: -

Nachbedingung: -

Ergebnisse: Eine Anzeige ist zu sehen, die Folgende Informationen beinhaltet:

- beantragten Services (Reperaturen und Reinigungen): int (Je anzahl der bevorstehenden Reperaturen und Reinigungen)
- die eingegangenen Kundenfeedbacks: int (Je Anzahl der offenen Feedbacks für Kategorie General, Vehicle, Line und Stop)
- die anstehenden Veranstaltungen: Name, Startzeitpunkt (Für die bevorstehenden Veranstaltungen)
- Fahrten denen kein Fahrzeug hinzugefügt wurde: Id der Tour, Linie und Start- und Endzeitpunkt(Jeweils pro Fahrt)
- Den Status des Netzwerks, bestehend aus:
 - Anzahl kritischer Probleme auf Linien sowie Haltstellen pro Linie: int
 - Anzahl "warnungs" Probleme auf Linien sowie Haltstellen pro Linie: int
- den Status des Fuhrparks
 - Anzahl kritischer Fahrzeuge (Busse und U-Bahnen) : int
 - Anzahl Warnungen für Fahrzeuge (Busse und U-Bahnen) : int

Querverweise: Use Case "Gesamtstatus anzeigen (1.1)" Weitere Übersichten: -

Kontrakt 1.2.1 AnzeigenKundenfeedback

Operation: AnzeigenKundenfeedback()

Ergebnisse: Eine Anzeige ist zu sehen, die Folgende Informationen beinhaltet:

- Anzahl der offenen und bearbeiteten Feedbacks: int

- Wichtigsten Informationen zu jedem angezeigtem Feedback:
 - Betroffener Typ (General, Stop: Haltestellenname, Vehicle: Fahrzeugname, Line:Linienname)
 - Grund für das Feedback (Accident, Broken, Dirty, General, Obstacle, Overflow)
 - Eingangszeit: Zeit
- Detailansicht für ein gewähltes Feedback (enthält zusätzlich zu den oberen Punkten noch):
 - Beschreibstext des Feedbacks vom Nutzer: String

Querverweise: Use Case "Kundenfeedback verarbeiten (1.2)"

Anmerkung: Weitere Kontrakte des Use Case unter 3.1 zu finden

2. Netz

Kontrakt 2.1 AnzeigenNetzstatus

Operation: AnzeigenNetzstatus()

Ergebnisse: Eine Anzeige ist zu sehen, die Folgende Informationen beinhaltet:

- Netzplan der Linien mit Punkten für die Haltestellen und farbig markierten Kanten für die einzelnen Linien und farblich markierten Anomalien
- Gesamtstatus der einzelnen Linien, bestehend aus:
 - Name: String
 - Status: StatusEnum (OK, Warnung oder Kritisch)
- Aktuelle Probleme des Netzes (Linien und HaltestellenProbleme: Broken, Dirty, General, Obstacle, Overflow)

Querverweise: Use Case "Netzstatus anzeigen (2.1)"

3. Meldungen und Service Requests

Kontrakt 3.1.1 AnzeigenÜbersichtServiceRequests

Operation: AnzeigenÜbersichtServiceRequests()

Ergebnisse: Eine Anzeige in Tabellenform ist zu sehen, die folgende Informationen beinhaltet:

- Ein Servicerequest ist eine Zeile mit folgenden Informationen:
 - Name des Requests: String
 - Priorität des Requests: 1 - low, 2 - normal, 3 - urgent, 4 - immediate
 - Zeitpunkt bis der Request bearbeitet sein soll: Date
 - Status: int (Offen/In Arbeit/Erledigt)
 - Zielobjekt: String Name der Haltestelle/ des Fahrzeug
 - Request Type: Maintenance oder Cleaning

Querverweise: Use Case "Service Requests verwalten (3.1)"

Kontrakt 3.1.4 AnlegenServiceRequestsFuerHaltestelle

Operation: AnlegenServiceRequestsFuerHaltestelle(WartungOderReinigung: serviceRequestEnum, Name: String, Haltestelle: Haltestelle, Beschreibung: String, Priorität : priorityEnum, Fälligkeitstermin: date, ListeVonReferenziertenFeedbacks: List)

Beschreibung: Eine ServiceRequest für eine Wartung oder Reinigung wird erzeugt und abgespeichert

Vorbedingung: -

Nachbedingung: Ein Service Request Object mit Referenz auf die Haltestelle wurde gespeichert. Des Weiteren wurden die Attribute Name, Beschreibung, Priorität und Fälligkeitstermin mit den zugehörigen Werten belegt. Zusätzlich wurden alle referenzierten Feedbacks auf abgearbeitet gesetzt.

Ergebnisse: Das System informiert den Operator über das erfolgreiche Hinzufügen des Service Request für die Haltestelle

Querverweise: Use Case "Service Requests verwalten (3.1)" und Use Case "Kundenfeedback verarbeiten(1.2)"

Anmerkung: Analog anlegenServiceRequestsFuerLinien(nicht ausformuliert)

Kontrakt 3.2 AnzeigenMeldungen

Operation: AnzeigenMeldungen(Linie: Linie)

Ergebnisse: Eine Anzeige in Tabellenform ist zu sehen, die folgende Informationen beinhaltet:

- Eine Zeile ist eine Meldung mit folgenden Spalten:
 - Beschreibung: String
 - Betroffene Haltestellen: List Querverweise: Use Case "Meldungen von Linie und Haltestelle anzeigen (3.2)"

Kontrakt 3.3.1 AnzeigenHaltestelle

siehe Kontrakt 8.1.1

Querverweise: Use Case "Meldungen anlegen (3.3)"

Kontrakt 3.3.2 AnlegenMeldung

Operation: anlegenMeldung(Linie: Linie, HaltestellenListe: List, Meldung: String)

Beschreibung: Alle Information zur Identifikation der Haltestelle hinzufügen(Linie, Haltestelle). Den Text, der als Meldung angezeigt werden soll eingeben und bestätigen.

Vorbedingung: -

Nachbedingung: Eine Meldungsobjekt wurde erzeugt mit einer Referenz auf die Haltestellen und Linie. Das Attribut Meldung wurde mit einem Text in String Format belegt.

Ergebnisse: Das System informiert den Operator über das erfolgreiche Hinzufügen der Meldung für die Haltestelle

Querverweise: Use Case "Meldungen anlegen (3.3)"

4. Events

Kontrakt 4.1.1 AnzeigeEvents

Operation: AnzeigeEvents(bevorstehend: boolean)

Ergebnisse: Eine Anzeige in Tabellenform ist zu sehen, die folgende Informationen beinhaltet:

- Eine Zeile ist ein Event mit folgenden Spalten:
 - Titel: String
 - Startzeit: Date
 - Endzeit: Date
- Detailansicht für ein gewählte Veranstaltung (enthält zusätzlich zu den oberen Punkten noch):
 - Ort: String
 - Beschreibung: String

Querverweise: Use Case "Veranstaltungen einsehen (4.1)"

Kontrakt 4.2 AnlegenVeranstaltung

Operation: anlegenVeranstaltung(Titel: String, Start: date, Ende: date, ZugehörigePartei: partysEnum, Ort: String, Beschreibung: String)

Beschreibung: Eine neue Veranstaltung wird angelegt

Vorbedingung: -

Nachbedingung: Eine Veranstaltungsobjekt wurde gespeichert. Es besteht aus:

- Titel: String
- Start: Zeit (Anfangsdatum mit Uhrzeit)
- Ende: Zeit (Enddatum mit Uhrzeit)
- ZugehörigePartei: partysEnum
- Ort: String (Adresse oder aussagekräftige Beschreibung des Veranstaltungsortes)
- Beschreibung: String

Ergebnisse: Das System informiert den Operator über das erfolgreiche Hinzufügen der Veranstaltung

Querverweise: Use Case "Veranstaltungen anlegen (4.2)"

5. Fuhrpark

Kontrakt 5.1 AnzeigenFuhrpark

Operation: AnzeigenFuhrpark()

Ergebnisse: Eine Anzeige in Tabellenform ist zu sehen, die folgende Informationen beinhaltet:

- Eine Zeile stellt ein Fahrzeug dar und beinhaltet folgende Informationen:
 - FahrzeugTyp: vehicleEnum
 - FahrzeugID: String
 - Verfügbar: boolean
 - Status: String(Rot,Grün,Gelb)
 - Probleme: List
- Detailansicht für ein gewähltes Fahrzeug (enthält zusätzlich zu den oberen Punkten noch):
 - Feedbacks: List
 - ServiceRequests: List

Querverweise: Use Case "Fuhrparkinformationen einsehen (5.1)"

Kontrakt 5.2.1 HinzufügenFahrzeug

Operation: hinzufügenFahrzeug(Name: String, Typ: vehicleEnum)

Beschreibung: Daten für einen Fahrzeug eingeben und bestätigen

Vorbedingung: -

Nachbedingung: Ein Fahrzeugobjekt inklusive eingegebenen Namen wurde dem Fuhrpark hinzugefügt

Ergebnisse: Das System informiert den Operator über das erfolgreiche Hinzufügen des Fahrzeugs zum Fuhrpark

Querverweise: Use Case "Hinzufügen eines Fahrzeugs zum Fuhrpark" (5.2)

Kontrakt 5.3 entfernenFahrzeug

Operation: entfernenFahrzeug(Fahrzeug: Fahrzeug)

Beschreibung: Fahrzeug aus dem Fuhrpark entfernen und bestätigen

Vorbedingung: Das zu entfernende Fahrzeug existiert im Fuhrpark

Nachbedingung: Das Fahrzeug, das aus dem Fuhrpark entfernt werden sollte, wurde im Fuhrpark als "gelöscht" markiert. Dies bewirkt, dass das Fahrzeug nicht mehr Fahrten zugeordnet werden kann, jedoch noch inklusive seiner referenzierten Historie einsehbar ist.

Ergebnisse:

- Das System informiert den Operator über das erfolgreiche Löschen-Markieren des Fahrzeugs
- Das System zeigt dem Operator den aktualisierten Fuhrpark an

Querverweise:

Use Case "Fahrzeug aus Fuhrpark entfernen" (5.3)

6. Fahrzeug

Kontrakt 6.1 AnzeigenFahrzeughistorie

Operation: AnzeigenFahrzeughistorie(Fahrzeug: Fahrzeug)

Ergebnisse: Eine Anzeige ist zu sehen, die folgende Informationen beinhaltet:

- Tabelle der Servicevorgänge(Analog zu: Servicevorgänge anzeigen nur mit Parameter Fahrzeug: Fahrzeug)
- Tabelle der Kundenfeedbacks zu den Fahrzeug(Analog zu Kundenfeedback anzeigen nur mit Parameter Fahrzeug: Fahrzeug)

Querverweise: Use Case "Fahrzeughistorie einsehen (6.1)"

Kontrakt 6.2 AnzeigenServiceterminFuerFahrzeug

Operation: AnzeigenServiceterminFuerFahrzeug (ServiceId:int)

Ergebnis: Ein Anzeige ist zu sehen, der folgende Informationen beinhaltet:

- Name des Requests: String
- Priorität des Requests: 1 - low, 2 - normal, 3 - urgent, 4 - immediate
- Zeitpunkt bis der Request bearbeitet sein soll: Date
- Status: int (Offen/In Arbeit/Erledigt)
- Zielobjekt: String Name der Haltestelle/ des Fahrzeug
- Request Type: Maintenance oder Cleaning

Querverweise: Use Case "Servicetermin für ein Fahrzeug einsehen (6.2)"

Kontrakt 6.3 AnzeigenLinienInfo

Operation: AnzeigenLinienInfo(Linie: Linie)

Ergebnisse: Ein Anzeige ist zu sehen, der folgende Informationen beinhaltet:

- In- und Outbound der Linie mit derzeit fahrenden Fahrzeugen
- Fahrzeug und deren jeweilige Verspätung

Querverweise: Use Case "Planmäßigkeit einer Linie prüfen (6.3)"

Kontrakt 6.4.1 AnzeigenEinsatzplanFahrzeugs

Operation: AnzeigenEinsatzplanFahrzeugs(Fahrzeug: Fahrzeug)

Ergebnisse: Eine Kalenderansicht ist zu sehen, die folgende Informationen beinhaltet:

- Fahrten: List(Fahrten bestehen aus Anfangs- und Endzeit, wird visualisiert)

Querverweise: Use Case "Einsatzpläne verwalten (6.4)"

Kontrakt 6.4.2 LöschenEinsatz

Operation: löschenEinsatz(Fahrt: Fahrt)

Beschreibung: Einen bestimmten Einsatz (s. Glossar) vom Einsatzplan eines Fahrzeugs löschen

Vorbedingung: Es liegt ein Einsatz für das entsprechende Fahrzeug vor

Nachbedingung: Der Einsatz, der entfernt werden sollte, wurde aus dem Einsatzplan des Fahrzeugs gelöscht

Ergebnisse: Das System zeigt dem Operator den aktualisierten Einsatzplan des Fahrzeugs an. (Die Referenz von der Fahrt auf das Fahrzeug wurde gelöscht)

Querverweise: Use Case "Einsatzpläne verwalten" (6.4, Alternativszenario 1)

Kontrakt 6.4.3 HinzufügenFahrt

Operation: hinzufügenFahrt(Fahrt: Fahrt)

Beschreibung: Eine Fahrt zu einem Fahrzeug zuordnen

Vorbedingungen:

- Ein nicht-defektes Fahrzeug existiert im Fuhrpark
- Dieses Fahrzeug hat noch keinen vollen Einsatzplan
- Ein Fahrzeug ist ausgewählt.

Nachbedingung: Ein Fahrtobjekt mit Referenz auf das Fahrzeugs wurde gespeichert.

Ergebnisse: Das System zeigt dem Operator den aktualisierten Einsatzplan des Fahrzeugs an

Querverweise:

- Use Case "Einsatzpläne verwalten" (6.4, Alternativszenario 2)
- Use Case "Fahrpläne verwalten" (7.3)

7. Linie

Kontrakt 7.1. AnzeigenLinie

Operation: AnzeigenLinie(Linie: Linie)

Ergebnisse: Ein Anzeige ist zu sehen, die folgende Informationen beinhaltet:

- Haltestellen der Linie: List(visualisiert)
- Aktuell auf der Linie fahrende Fahrzeuge: List
- Aktuelle Position dieser Fahrzeuge: List
- Meldungen der Linie(wie in Meldung anzeigen beschrieben)
- Feedback der Linie(wie in Feedback der Linie anzeigen beschrieben, nur mit Parameter Linie:Name)
- Service Requests der Stops der Linie(wie in Service Request anzeigen beschrieben, nur mit Parameter Linie:Name)

Querverweise: Use Case "Linienstatus prüfen (7.1)"

Kontrakt 7.2.1 HinzufügenFahrzeugZuFahrt

Operation: hinzufügenFahrzeugZuFahrt(Fahrt: Fahrt, Fahrzeug: Fahrzeug)

Beschreibung: Fahrzeug das zur ausgewählten Fahrt hinzugefügt werden soll auswählen und bestätigen

Vorbedingung: -

Nachbedingung: Das Fahrtobjekt referenziert das Fahrzeug

Ergebnisse: Das System informiert den Operator über das erfolgreiche Hinzufügen des Fahrzeugs zur Fahrt

Querverweise: Use Case "Fahrzeug einer Fahrt verwalten" (7.2)

Kontrakt 7.2.2 EntferneFahrzeugVonFahrt

Operation: entferneFahrzeugVonFahrt(Fahrt: Fahrt, Fahrzeug: Fahrzeug)

Beschreibung: Fahrt wählen, Löschen-Funktion auswählen und bestätigen

Vorbedingung: -

Nachbedingung: Das Fahrtobjekt referenziert auf keine Fahrzeug mehr

Ergebnisse: Das System informiert den Operator über das erfolgreiche Löschen des Fahrzeugs von der Fahrt

Querverweise: Use Case "Fahrzeug einer Fahrt verwalten" (7.2)

Kontrakt 7.3.1 AnzeigenFahrplan (Nur Fahrplan anschauen)

Operation: AnzeigenFahrplan(Linie: Linie)

Ergebnisse: Ein Anzeige in Tabellenform ist zu sehen, der folgende Informationen beinhaltet:

- Eine Zeile ist eine Fahrt mit folgenden Werten:
 - TourID: String
 - Fahrzeug: Fahrzeug
 - Datum: Datum
 - Ankunftszeit für jede Haltestelle: List

Querverweise: Use Case "Fahrpläne verwalten (7.3)" , Use Case „Fahrzeug einer Fahrt verwalten(7.2)“

Kontrakt 7.3.2 HinzufügenFahrtZuFahrplan

Operation: hinzufügenFahrtZuFahrplan(Startzeit: date, Inbound: boolean, Fahrzeug: Fahrzeug)

Beschreibung: Daten für eine Fahrt eingeben und bestätigen.

Vorbedingung: Eine Linie ist ausgewählt

Nachbedingung: Ein Fahrtojekt wurde angelegt und mit allen angegeben Attributen initialisiert. Es referenziert auf ein Fahrzeuge.

Ergebnisse: Das System informiert den Operator über das erfolgreiche hinzufügen eines Fahrzeugs zur Fahrt

Querverweise: Use Case "Fahrpläne verwalten" (7.3)

Kontrakt 7.3.3 EntfernenFahrtVonFahrplan

Operation: entfernenFahrtVonFahrplan(Fahrt: Fahrt)

Beschreibung: Fahrt von Fahrplan entfernen und bestätigen

Vorbedingung: -

Nachbedingung: Das Fahrtojekt wude gelöscht

Ergebnisse: Das System informiert den Operator über das erfolgreiche Entfernen der Fahrt vom Fahrplan

Querverweise: Use Case "Fahrpläne verwalten" (7.3)

8. Haltestellen

Kontrakt 8.1.1 AnzeigenHaltestelle

Operation: AnzeigenHaltestelle(Haltestelle: Haltestelle)

Ergebnisse: Ein Anzeige ist zu sehen, der folgende Informationen beinhaltet:

- Name: String
- Status: String(ok, Warnung, Defekt)
- Services: List(Wie in Services anzeigen nur mit Parameter HaltestellenID:int)
- Feedbacks: List(Wie in Feedbacks anzeigen nur mit Parameter HaltestellenID:int)
- Meldungen: List(Wie in Meldungen anzeigen nur mit Parameter HaltestellenID:int)

Querverweise: Use Case "Haltestellenstatus verwalten (8.1)"

Kontrakt 8.1.2 SperreHaltestelle

Operation: sperreHaltestelle(Haltestelle: Haltestelle)

Beschreibung: Haltestelle sperren und bestätigen

Vorbedingung: -

Nachbedingung: Das Attribut gesperrt des Haltestellenobjekts wurde auf wahr gesetzt.

Ergebnisse: Das System informiert den Operator über das erfolgreiche Sperren der Haltestelle

Querverweise: Use Case "Haltestellenstatus verwalten" (8.1)