



# Anforderungsspezifikation BLS Frequenzerhebung

**Projektteam**

**Silas Stulz  
Yannick Stebler**

**V1.1**

**26.09.2018**

# Inhaltsverzeichnis

1	Zweck des Dokuments	3
2	Vision	3
3	Projektziele	3
4	Systemabgrenzung	4
4.1	Prozessumfeld	4
4.2	Systemumfeld	5
4.3	Use-Case & Userstories	6
4.3.1	Use-Case: Zu- und Aussteigende Personen zählen	6
4.3.2	Use-Case: Gesammelte Informationen eingeben	7
4.3.3	Use-Case: Maximale Personenanzahl im Auge behalten	7
4.3.4	Use-Case: Kontrolle und Übermittlung der Informationen	8
4.3.5	Use-Case: GeoTracking – Automatische Ankunfts- und Abfahrtszeit	8
4.3.6	Use-Case: GeoTracking – Automatische Erkennung der Anlegestelle	9
4.3.7	Use-Case: Extra und Dienstfahrten	9
5	Anforderungen	10
5.1	Quellen und Vorgehen	10
5.1.1	Meeting auf BLS Schiff 23.11.2018	10
5.2	Funktionale Anforderungen	11
5.3	Qualitätsanforderungen	11
5.4	Randbedingungen	12
5.5	Datenmodell	12
6	Technische Lösung und Architektur	14
6.1	Lösung	14
6.1.1	Entwicklungsumgebung	14
6.1.2	Verwendete Versionen	14
6.2	Architektur	14
6.2.1	Datenbank	14
6.2.2	Fragment Views	15
6.2.3	Screen Sizes	15
6.2.4		16
7	Glossar	17
8	Anhang	17
8.1	Definition of Ready – Checklist	17
9	Versionskontrolle	18

# 1 Zweck des Dokuments

Dieses Dokument beschreibt die Ziele und Anforderungen für das Projekt BLS Frequenzerhebung.

## 2 Vision

Der Prozess der Frequenzerhebung, anhand der Daten der Frequenzerhebung kann die BLS Rückforderungen für erbrachte Leistungen bei der SBB machen (Benutzung des GA's), bei der BLS Schifffahrt soll für alle beteiligten Personen vereinfacht werden. Damit dies geschehen kann wird er veraltete Prozess, der über Papier fungiert hat abgeschafft und durch eine zeitgemässe Softwarelösung ersetzt. Mithilfe dieser Softwarelösung wird erreicht, dass die Daten direkter verfügbar sind. Zudem können hiermit verschiedenen Fehlerquellen eliminiert werden. Des Weiteren können die Kosten und Ressourcen für die BLS AG verringert werden, da sowohl die monatlichen Betriebskosten wie auch der personelle Aufwand reduziert werden kann.

## 3 Projektziele

### Liste der relevanten Stakeholder:

- BLS
- Entwickler,
- User (Kapitän)
- Abteilung, welche das System warten/unterhalten muss

### BLS:

Die BLS will ihr bestehendes System erneuern und so Prozesskosten einsparen. Ausserdem möchte die BLS so Personal einsparen und das System automatisieren, damit keine manuellen Überprüfungen der Kennzahlen mehr nötig ist.

### User:

Für den Kapitän wird der Prozess vereinfacht, so dass er die Zahlen nur noch einmal in ein Android Tablet eingeben muss und sich nicht mehr darum kümmern muss wohin das Papier verschwindet.

### Entwickler:

Das Ziel der Entwickler ist es, dass Projekt zu einem guten Abschluss zu bringen und eine Grundlage für ein produktives System zu schaffen.

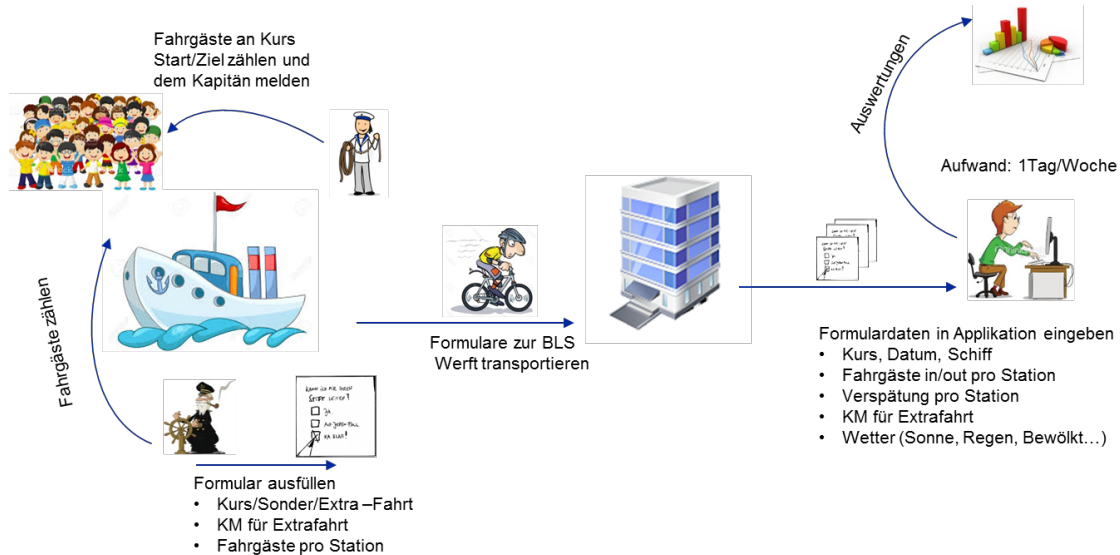
### Nicht-Ziele:

Bewusste Ziele sind nicht die Änderung des bestehenden Prozesses im eigentlichen Sinn. Da das System schon tief verankert ist, insbesondere bei den Mitarbeitern der BLS. Deshalb soll der bestehende Prozess vereinfacht und modernisiert werden.

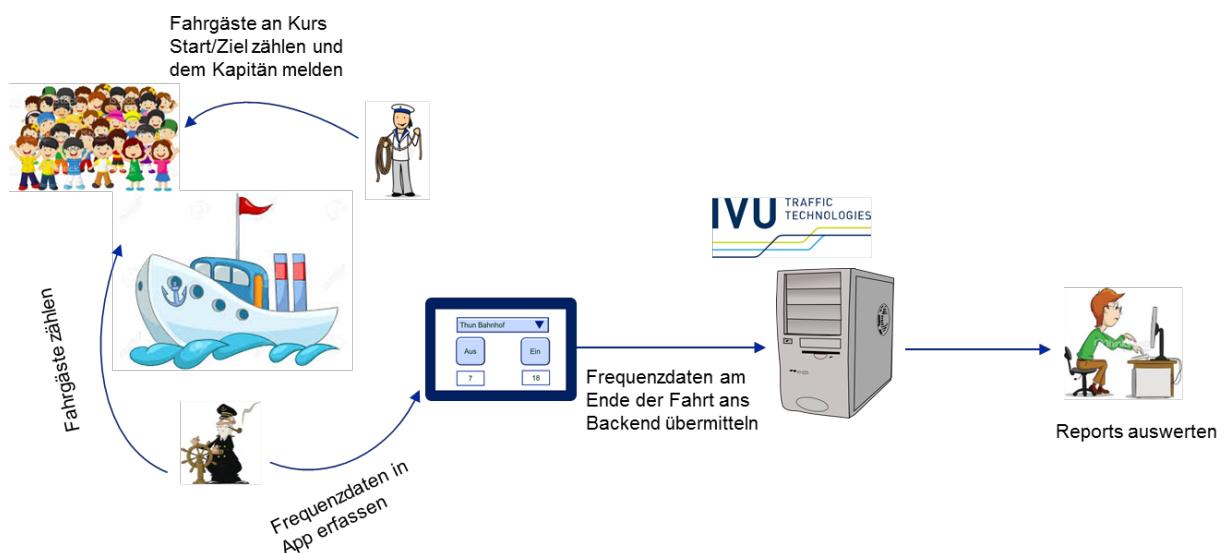
## 4 Systemabgrenzung

### 4.1 Prozessumfeld

Momentaner Prozess:

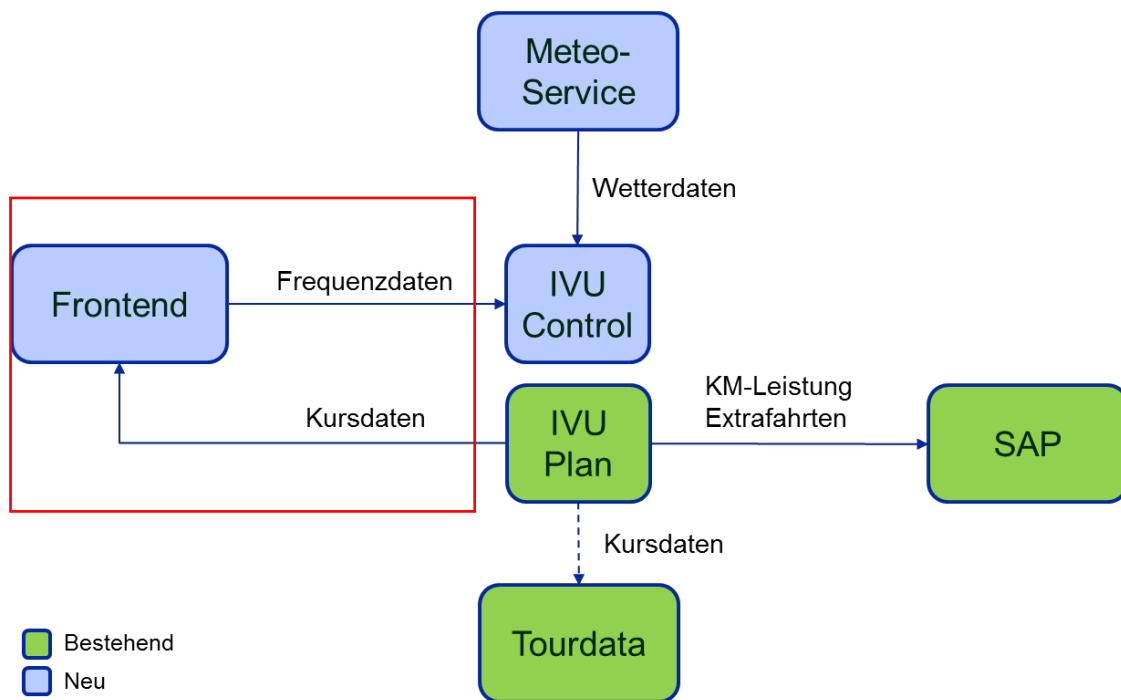


Neuer Prozess:



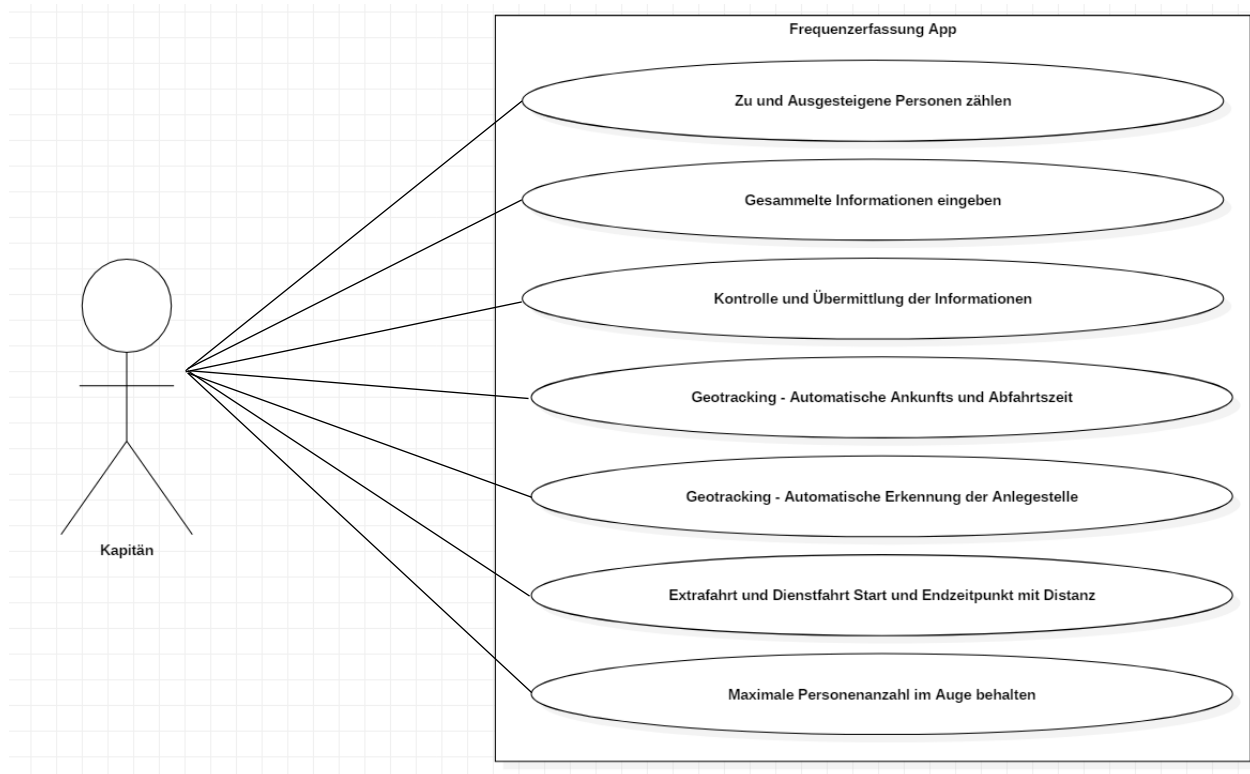
Das Formular, das vom Kapitän bis anhin mit den erforderlichen Daten ausgefüllt wurde und anschliessend zu einer Person im Backoffice transportiert wurde, die die Daten eingetragen hat, wird nun direkt über die Eingabe der Daten in die Softwarelösung und die Übermittlung der Daten aus der Softwarelösung ins Backend abgelöst.

## 4.2 Systemumfeld



Dieses Projekt konzentriert sich auf das Frontend (Rot umrandet) des Prozesses. In direktem Zusammenhang mit dem Frontend, also der Android Applikation stehen das IVU Control und das IVU Plan. Durch das IVU Plan erhält unsere Applikation die nötigen Daten (Fahrplan, Kapitän, etc.). Diese werden von der Applikation erweitert und zurück ins IVU Control gespeist. Von dort aus haben wir keinen Einfluss mehr auf die verarbeiteten Daten. Unsere Applikation steht in keinem anderen Zusammenhang mit anderen Systemteilnehmern.

## 4.3 Use-Case & Userstories



### 4.3.1 Use-Case: Zu- und Aussteigende Personen zählen

Der User (Kapitän) muss die Möglichkeit haben zu- und aussteigende Personen in der App zu registrieren. Hierfür wird vom User(Kapitän) mithilfe eines mechanischen Zählers die zu- und aussteigenden Personen gezählt und anschliessend in der App erfasst.

#### 4.3.2 Use-Case: Gesammelte Informationen eingeben

Frequenzerhebung Blüemlisalp

Kurs

131

Station

Interlaken Ost

An/ab

12:15/12:20

Passagiere

213

Aussteiger

Einsteiger

Angefahren

☒

Grund

Schlechte Sicht

Tastatur

Der Kapitän kann die wichtigsten Informationen in der App überprüfen (Schiff, Linie, Haltestelle), anschliessend kann er die im vorhergehenden manuellen Prozess erhaltenen Informationen in die dafür vorgesehen Felder eingeben und diese speichern.

#### 4.3.3 Use-Case: Maximale Personenanzahl im Auge behalten

Der Kapitän muss zu jeder Zeit die Möglichkeit haben, sich über die momentane Anzahl an Personen an Bord zu informieren. Dies da die Schiffe eine maximale Anzahl an Personen aufnehmen können und es vor allem im Winter bei den kleineren Schiffen dazu kommen kann, dass die maximale Kapazität überschritten werden könnte.

#### 4.3.4 Use-Case: Kontrolle und Übermittlung der Informationen

Frequenzerhebung Blüemlisalp

Kurs

131

Station	Ein	Aus
Thun(See)	102	0
Hünibach(See)	15	10
Hinterfingen(See)	5	15
Oberhofen am Thunersee	20	5
Gunten(See)	10	65
Spiez Schiffstation	0	57
Total	152	152

Der Kapitän hat die Möglichkeit am Ende einer Fahrt eine Kontrolle aller erfassten Informationen anzuschauen (überprüfen) anschliessend kann er diese Daten in das Backoffice Übertragen. Eine Korrektur der erfassten Informationen durch den Kapitän ist im ersten Schritt nicht möglich.

#### 4.3.5 Use-Case: GeoTracking – Automatische Ankunfts- und Abfahrtszeit

Der Kapitän soll die Ankunfts- und Abfahrtszeit nicht mehr manuell eingeben müssen, daher wird über GeoTracking geschaut wann das Schiff eine Geschwindigkeit von 0 erreicht und somit angelegt hat und wann die Geschwindigkeit wieder erhöht wird und somit das Schiff abgelegt hat.



#### 4.3.6 Use-Case: GeoTracking – Automatische Erkennung der Anlegestelle

Die App soll mittels GeoTracking automatisch die momentane Anlegestelle erkennen. Somit wird die Fehlermöglichkeit weiter verringert.

#### 4.3.7 Use-Case: Extra und Dienstfahrten

Frequenzerhebung Blüemlisalp

Kurs

131

---

Startzeit


22.01.2019 14:06

StartStation

Thun (See)

Distanz zurückgelegt

25km



Der Kapitän hat die Möglichkeit bei einer Dienstfahrt oder eine Extrafahrt (Schiffe von Firmen oder Privatpersonen gemietet) speziell zu erfassen. Sobald er eine Extra- oder Dienstfahrt auslöst wird der Startzeitpunkt festgelegt. Die Distanz wird zu einem späteren Zeitpunkt über GeoTracking errechnet. Sobald er die Dienstfahrt beendet, wird ein Eintrag für die Extra/Dienstfahrt in der Datenbank vorgenommen.

## 5 Anforderungen

### 5.1 Quellen und Vorgehen

Zur Definition der Anforderungen haben wir mit dem Dokument, dass wir von der BLS AG erhalten haben gearbeitet. Nach einem ersten Entwurf haben wir mit Herr Welti (BLS AG) und Herr Lange die Anforderungen genauer spezifiziert.

Wir haben uns mit den Beteiligten Personen der BLS AG darauf geeinigt, dass wir schon ziemlich früh im Projekt uns daran machen einen Prototypen zu entwickeln. Dieser Prototyp wurde auf den Ideen und Voraussetzungen, die wir von den Herrn Welti und Lange erhalten haben in einer ersten Version erstellt. Das Ziel des ersten Prototypen wird sein, dass wir eine erste verwendbare App haben, die den Kapitänen der BLS Schifffahrt auf dem Thuner und Brienzersee gezeigt und erklärt werden kann. Damit wir anschliessend eine Besprechungsgrundlage haben, und mit den Kapitänen die Anforderungen noch genauer spezifizieren und eventuelle Änderungswünsche berücksichtigen können.

#### 5.1.1 Meeting auf BLS Schiff 23.11.2018

Am 23.11.2018 haben wir dann mit der ersten richtigen Prototypen Version einen Ausflug nach Interlaken und auf ein Schiff der BLS gemacht. Hier haben wir dem Kapitän des Schiffes den Prototypen vorgestellt und konnte mit ihm noch weitere wichtige Punkte, die in die nächste Version des Prototypen einfliessen werden besprechen.

Das wichtigste Ziel der Applikation wird es sein ein möglichst einfaches Benutzerinterface zur Verfügung zu stellen, dies war das Hauptaufmerksamkeitspunkt der uns beim Besuch auf dem BLS Schiff aufgefallen ist.

## 5.2 Funktionale Anforderungen

ID	Status	Prio	Beschreibung
F1.1	Entwurf	M	Erfassen der Ein- und Aussteigenden Personen. Die Eingabe soll über die Softwarelösung erfolgen. Das Zählen der Personen wird weiterhin manuell durch die zuständigen Personen durchgeführt. Anschliessend können die Zahlen in die dafür vorgegebenen Eingabefelder eingegeben werden.
F1.2	Entwurf	M	Übertragung der Daten ans Backend (Frequenzdaten, evt. Zusätzliche Informationen. Die Daten der Frequenzerfassung dürfen zwischen den Eingestiegenen und Ausgestiegenen Personen keine Differenzen aufweisen, da die Daten vom Backend sonst nicht eingelesen werden können.
F1.3	Entwurf	M	Übernehmen der für den aktuellen Tag geplanten Kursfahrten pro Schiff. Die App kann diese Kursdaten über das Backend beziehen und in die Datenbank einspeisen.
F1.4	Entwurf	P1	<p>Automatische Erkennung der Anlegestelle, in einen späteren Verlauf wird es möglich sein über die vom Backend bezogenen Informationen (diese Informationen werden die Stationen inkl. der Geokoordinaten beinhalten) die Stationen mithilfe von GPS automatisch zu erkennen.</p> <p>Die Automatische Erkennung der Anlegestelle wird nicht benötigt. Da wir durch den Server der BLS für jeden Tag die aktuellen Fahrpläne beziehen können wird der Benutzer die Haltestellen danach abarbeiten</p>
F1.5	Entwurf	P2	<p>Automatische Erkennung der An- und Abfahrtszeit, die Softwarelösung soll erfassen, wann ein Schiff bei einer Anlegestelle ankommt und wann es wieder losfährt. Es muss geklärt werden ob die An- und Abfahrtszeit per Knopfdruck in der Softwarelösung durch den User bestätigt wird, oder ob dies über eine GPS / Geschwindigkeitsmessung geschehen soll.</p> <p>In der ersten Version der Lösung wird die An- Abfahrtszeit noch nicht über GPS erfasst werden</p>
F1.6	Entwurf	P3	Anpassungen am Fahrplan (Anlegeort wird nicht bedient, neuer Anlegeort, temporärer Fahrplanwechsel)
F1.7	Entwurf	P9	Möglichkeit der User Authentifizierung – Es wird keine Authentifizierung in der App notwendig sein. Das Tablet wird durch einen PIN/Muster durch die BLS AG geschützt.
F1.8	Entwurf	M	Anzahl Personen die Momentan an Bord sind
F1.9	Entwurf	M	Es wird eine Auswahl an möglichen Gründen gegeben sein, wieso eine Haltestelle nicht angelaufen werden kann
F.1.10	Entwurf	M	Zu jeder Haltestelle sollen die Plan-Ankunfts und Abfahrtszeiten angezeigt werden.

## 5.3 Qualitätsanforderungen

ID	Status	Prio	Beschreibung
Q1.1	Entwurf	M	Die Applikation muss sehr intuitiv und einfach zu bedienen sein. Der Benutzer soll über eine möglichst kleine Anzahl an Eingaben und Klicks durch die App navigieren können.
Q1.2	Entwurf	M	Es soll eine Versionskontrolle des Quellcodes via GIT stattfinden
Q1.3	Entwurf	M	Der User soll sich nicht authentifizieren müssen
Q1.4	Entwurf	M	Die Applikation soll die geforderten Daten exakt und ohne Verluste ins Backend liefern
Q1.5	Entwurf	M	Die Applikation soll für die Benutzer attraktiv und einladend sein.

## 5.4 Randbedingungen

ID	Status	Prio	Beschreibung
R1.1	Entwurf	M	Die Applikation sollte für den Gebrauch auf Tablets optimiert sein
R1.2	Entwurf	M	Die Applikation soll den bestehenden Prozess modernisieren aber nicht grundlegend verändern.
R1.3	Entwurf	M	Die Daten müssen in dem exakt vorgegebenen Format des IVU ans Backend übermittelt werden

## 5.5 Datenmodell

Modell der wichtigsten Entitäten, soweit es für das Verständnis der Anforderungen notwendig ist.

Merkmal	Beschreibung	Typ	Länge	Erforderlich
<b>Betreiber</b>	Kürzel des Betreibers	Text	8	Nein
<b>Linie</b>	Name der Linie	Text	32	Nein
<b>Fahrtnummer</b>	Fahrtnummer der Fahrt	Text	10	Ja
<b>Datum</b>	Betriebstags-Datum der Zählung. Format: dd.mm.yyyy	Datum	10	Ja
<b>Fahrzeugnummer</b>	Nummer des Fahrzeugs	Text	32	Nein
<b>Haltestelle</b>	Kürzel der Haltestelle der Erfassung. Bei Ringfahrten gefolgt von «#» und der Passagenummer (z.B. KUERZEL #2 für die zweite Passage der Haltestelle)	Text	8	Ja
<b>Ankunftszeit</b>	Ankunftszeit an der Haltestelle. Uhrzeit im Format 00:00:00 (HH:MM:SS). Die Uhrzeit bezieht sich auf den Kalendertag, nicht auf den Betriebstag der Fahrt (01:00:00 statt 25:00:00)	Zeit	8	Nein
<b>Abfahrtszeit</b>	Abfahrtszeit an der Haltestelle. Uhrzeit im Format 00:00:00 (HH:MM:SS). Die Uhrzeit bezieht sich auf den	Zeit	8	Nein



## 6 Technische Lösung und Architektur

### 6.1 Lösung

#### 6.1.1 Entwicklungsumgebung und Source Control

Als Entwicklungsumgebung für die App diente uns Android Studio. Es wurde für die Entwicklung der App von uns keine zusätzlichen API installiert.

Die Source Control Verwaltung wurde über eine GitHub Repository gelöst. Das Repository ist private. Somit ist es nur für die Personen die daran beteiligt sind verfügbar und steht der Öffentlichkeit nicht zur Verfügung.

#### 6.1.2 Verwendete Versionen

Nachfolgend ist eine Auflistung der Versionen der einzelnen Komponenten, die wir für die Entwicklung der App benötigt haben:

- Android Studio 3.2.1
- JRE: 1.8.0\_152-release-1136-b06 amd64
- JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

### 6.2 Architektur

#### 6.2.1 Datenbank

Gearbeitet wurde mit einer SQLite Datenbank direkt auf dem Android Device. Da jede App selber Daten anlegen und verwalten soll. Wir haben für die Datenbankbindung auf einen OR-Mapper verzichtet.

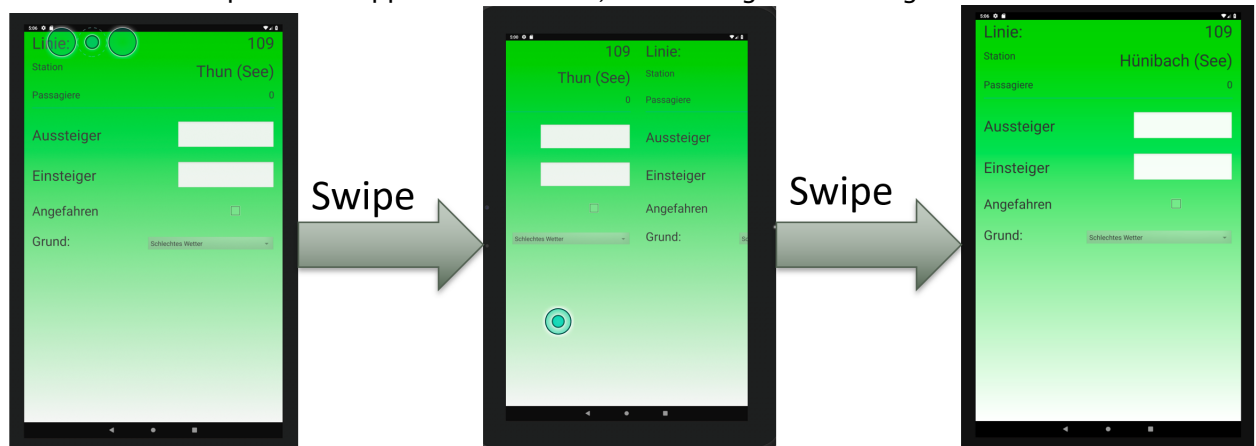
Jede Tabelle wurde als Entity abgebildet und alle Tabellen werden über die DBHelper-Klasse erstellt und verwaltet. Alle CRUD (Create, Read, Update, Delete) Anfragen sind in der DBHelper-Klasse festgehalten und können aufgerufen werden.

Um die Datenbank komplett zurück zu setzen kann, die DATABASE\_VERSION inkrementiert werden. Sobald der DBHelper feststellt, dass die Nummer höher ist, wird die Datenbank per Drop gelöscht und anschliessend neu generiert.

```
public class DBHelper extends SQLiteOpenHelper {  
  
    // Database Version  
    private static final int DATABASE_VERSION = 15;  
  
    // Database Name  
    public static final String DATABASE_NAME = "bls_boats";  
  
    public DBHelper(Context context) { super(context, DATABASE_NAME, factory: null, DATABASE_VERSION); }  
  
    //Creating Tables  
    ...  
}
```

### 6.2.2 Fragment Views

Damit der Userswipe für die App erkennbar wird, wurden sogenannte Fragment Views erstellt.



Die Haltestellen werden somit als eigenstehende Views erstellt. Sobald der User einen Swipe ausführt wird überprüft ob eine Haltestelle für den User sichtbar ist. Hierfür dient die folgende Methode in der StationFragment Klasse.

```
@Override
public void setUserVisibleHint(boolean isVisibleToUser) {
    super.setUserVisibleHint(isVisibleToUser);
    if (db != null) {
        if (isVisibleToUser) {
            updateTextPersonCountOnShip(calculatePersonCountOnShip());
        } else {
            saveErfassung();
        }
    }
}
```

Hier wird überprüft ob die Haltestelle für den User sichtbar ist und anschliessend das Speichern der Daten ausgelöst.

### 6.2.3 Screen Sizes

Um die Anforderung, dass die App für verschiedene Geräte mit unterschiedlichen Bildschirmgrössen funktionieren soll, zu erfüllen. Haben wir auf die von Android Studio vorgegebene Möglichkeit zurückgegriffen. Es werden pro View (Android Resource File) für die jeweiligen Grössen und Orientations ein File erstellt.

Es gibt folgende Grössen:

- Small
- Normal
- Large
- XLarge

Ausserdem gibt es folgende Orientations:

- Portrait
- Landscape
- Square

Die App wählt je nach Bildschirmgrösse und Orientation die jeweilige View aus und zeigt diese an. Es besteht die Möglichkeit über das Android Manifest die erlaubten Screensizes einzustellen.

```

<supports-screens
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:xlargeScreens="true"
/>

```

```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="BLS_Boats"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat">
    <activity
        android:name=".Journey.JourneyActivity"
        android:screenOrientation="portrait" />
    <activity android:name=".MenuActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".ExtraJourney.ExtraJourneyActivity" />
    <activity android:name=".Settings.SettingsActivity" />
    <activity android:name=".Journey.SpecialJourneyActivity"/>
</application>

```



## 7 Glossar

Bezeichnung	Beschreibung
<b>Android</b>	Mobiles Betriebssystem für Smartphones und Tablets
<b>App</b>	App kurz für Applikation
<b>Backend</b>	Als Backend wird der Teil der Software bezeichnet welcher sich im Hintergrund, meist versteckt für den Benutzer, mit der Verarbeitung von Daten beschäftigt.
<b>BLS</b>	Ist eine Schweizer Bahngesellschaft, welche verschieden Strecken (mehrheitlich im Kanton Bern) betreibt
<b>Frontend</b>	Als Frontend wird meistens die Software bezeichnet die lokal auf dem Gerät des Benutzers läuft, oder darauf dargestellt wird. Dies beinhaltet meistens eine graphische Benutzeroberfläche
<b>GA</b>	General Abonnement, erlaubt die freie Benützung der meisten öffentlichen Verkehrsmittel in der Schweiz
<b>GIT</b>	GIT ist eine freie Software zur verteilten Versionsverwaltung von Dateien
<b>IVU</b>	Das IVU ist eine spezielle Softwarelösung für den öffentlichen Verkehr
<b>SAP</b>	SAP ist ein deutsches Softwareunternehmen das verschiedene Softwarelösungen für Unternehmen produziert.
<b>SBB</b>	Die Schweizerischen Bundesbahnen sind die staatliche Eisenbahngesellschaft der Schweiz

## 8 Anhang

### 8.1 Definition of Ready – Checklist

Es müssen die Zu- und Ausgestiegenen Personen ohne Differenzen, eventuell mit Differenkkorrektur, an das Backend der BLS AG übermittelt werden können.

## 9 Versionskontrolle

Version	Datum	Beschreibung	Autor
X0.1	26.09.2018	Dokument erstellt	Silas Stulz, Yannick Stebler
X0.2	03.10.2018	Dokument überarbeitet	Silas Stulz, Yannick Stebler
X0.3	17.10.2018	Dokument überarbeitet	Silas Stulz, Yannick Stebler
X0.4	24.10.2018	Dokument überarbeitet	Silas Stulz, Yannick Stebler
X0.5	9.11.2018	Dokument überarbeitet (Datenmodell)	YS
X0.6	16.11.2018	Dokument überarbeitet	Silas Stulz, Yannick Stebler
X0.7	19.11.2018	Anpassungen bezüglich Kapazität Klasse 1 & 2 wurden bei Datenmodell entfernt	Yannick Stebler
X0.8	28.11.2018	Glossar, überarbeitet	Silas Stulz
X0.9	28.11.2018	Informationen bezüglich Meeting auf dem BLS Schiff hinzugefügt und einige Teile überarbeitet	Yannick Stebler
V1.0	30.11.2018	Abschliessende Arbeiten am Dokument	Silas Stulz, Yannick Stebler
V1.1	22.01.2019	Überarbeiten des Dokumentes und hinzufügen technische Dokumentation	Yannick Stebler