

Checkliste - Fahrplan App Projekt

Abgabe

Jede Gruppe hat für die Abgabe einen FTP-Account erhalten.

Ordnerstruktur Abgabe

- **Canvas** <https://canvanizer.com/canvas/weLrLICynp93z>
Beinhaltet einen Screenshot von eurem Lean-Canvas (png, jpg, ..)
- **Vision Box**
Beinhaltet Bilder eurer Vision-Box (png, jpg, ...)
- **Wireframes**
Beinhaltet Fotos/Scans/exporte eure Wireframes (png, jpg, ...)
- **Code**
 - Android Studio > File > Export to Zip File
 - Lauffähes APK
- **Checkliste**
Dieses Dokument ausgefüllt

Gruppen-Informationen

Gruppe	Gruppe 2
Name 1. Person	Mattle Armin
Name 2. Person	Mele Luca

(Bitte Gruppennummer und Namen der Gruppenmitglieder einfüllen)

Eure FTP-Zugangsdaten für die Abgabe:

Server: sl49.web.hostpoint.ch
Username: Gruppe2@androidworkshop.ch
Passwort: HDD5E2N6

UX-Checkliste

- **GUI-Design ist konform mit Android-Styleguide**
Wir zählen nicht die Pixel, aber der Eindruck muss mit dem Android-Styleguide übereinstimmen. Die App fühlt sich an wie eine gute Android App.
- **Navigation ist konform mit dem Android Standard**
Platzierung von Actions in App/Action Bar vs. Overflow-Menu etc.
- **Die richtigen GUI-Elemente werden verwendet**
Textfield vs. Textarea, Radio buttons vs. Checkboxes etc.
- **Die richtigen Dialog-Typen werden eingesetzt**
Modal-Dialog vs. Fullscreen-Dialog etc.
- **Es werden die richtigen Progress-Indikatoren verwendet**
Determinate indicators vs. Indeterminate indicators etc.
- **Die Darstellung von Information und Interaktions-Elementen ist übersichtlich und logisch für den Benutzer**
GUI-Element und Inhalt sind überlegt platziert und folgen einer logischen Struktur.

Source-Checkliste

“Räumt euren Code auf”

Allgemein

- **Einheitliche Sprache für Code und Kommentare**
Kein Mischmasch aus Englisch und Deutsch
- **Kein auskommentierter oder toter Code**
- **Einheitlicher “Coding Style”**
Klassen- und Methodennamen, Klammern, Abstände, ...
- **Single Responsibility**
Klassen und Methoden haben wo möglich nur eine Aufgabe.
- **Erkennbare Architektur**
Es wird eine sinnvolle, projektweite Architektur eingesetzt.
- **Keine unnötigen Kommentare**
Code sollte selbsterklärend sein

Android

- **Keine Verwendung von statischen Measurement-Units (px, in, ...)**
- **Sinnvolle Benennung der Ressourcen**
- **Keine Strings direkt in den Layouts**

Minimal-Anforderungen

Wenn die Artefakte der Minimal-Anforderungen nicht vorhanden sind, gilt das Projekt als nicht bestanden.

Nr.	Anforderung	Umgesetzt (Ja/Nein)
1	Lean Canvas und Vision Box	ja
2	Wireframes	ja
3	Umsetzung	
3.1	Suche nach den drei nächsten Verbindungen von A nach B	
3.2	A und B können vom Benutzer der App gewählt werden	
3.3	Verbindungen werden so angezeigt, dass man weiss wann und wo der Zug/Bus/... abfährt und wann man wo ankommt.	
3.4	Anbindung der API transport.opendata.ch	
3.5	About-Screen mit Namen und E-Mail Adressen der Entwickler	

Mögliche zusätzliche Anforderungen

Nr.	Anforderung	Maximal Punkte	Umgesetzt (Ja/Nein)
5	Erweiterte Such-Funktionalität		
5.1	Autocomplete einer Stationseingabe	1	
5.2	„Umgekehrte Richtung“ – Einfache Möglichkeit Von und Nach zu vertauschen. z.B. ein Button	0.25	
5.3	Suche der Verbindung zu einem spezifischen Zeitpunkt	0.5	
5.4	Angabe ob gewählte Zeitpunkt die Ankunfts- oder Abfahrts-Zeit ist	0.5	
5.5	Spätere Verbindungen anzeigen für gesuchte Verbindung	0.5	
5.6	Nächste Station finden von der Position des Users	1	
6	Favoriten		
6.1	Eine Verbindungssuche kann als Favorit gespeichert und einfach aufgerufen werden	1	
6.2	Ein Favorit kann wieder gelöscht werden	0.5	
7	Reminder		
7.1	Der Benutzer kann einen Reminder setzen auf einer spezifischen Verbindung. Die App meldet sich n Minuten vor der Abfahrt	1	
8	Settings		
8.1	Festlegen der gewünschten Transportmittel	0.5	
8.2	Take me Home-Adresse festlegen	0.5	
8.3	Festlegen der Belegung für eine gewünschte Klasse	0.5	
9	Verbindung sharen		
9.1	Verbindung kann an jemanden gesendet werden (als Text). z.B. per WhatsApp, Mail, SMS,... Dabei sind die wichtigsten Informationen der Verbindung enthalten (Abfahrt, Ankunft, ...)	0.5	

10	Usability / User Experience / Design		
10.1	Während der Suche wird dem User angezeigt, dass etwas passiert. z.B. ein „Bitte Warten“-Dialog oder ein einfacher Indicator.	0.25	
10.2	App-Icon ist gesetzt	0.25	
10.3	Übersetzung in zwei Sprachen	0.5	
10.4	Optimierte Darstellung auf einem Tablet. Damit die volle Punktzahl vergeben werden kann muss das UI mit Fragments aufgebaut sein. Andere Optimierungen geben nur Teilpunkte.	1	
11	Verbindungs-Ansicht		
11.1	Anzeigen der Verkehrsmittel <i>Die volle Punktzahl wird nur gegeben, wenn eine grafische Darstellung der verwendeten Verkehrsmittel umgesetzt wird.</i>	0.5	

Eigenen Ideen und Anforderungen

Bitte meldet euch vor der Umsetzung bei uns, damit wir euch mitteilen können wie viele Punkte wir euch für diese Features geben können.

Nr.	Anforderung	Umgesetzt (Ja/Nein)

Bemerkungen

Zusätzliche Bemerkungen und Erklärungen:

•

1

•

•

•

•

1

•

•

•

•

•

•

1