

High-fidelity Prototyp der Trip-Planner App

Team 5, Gruppe 2

MileStone 3, 2023S Mensch-Computer-Interaktion, 051041

Milenko Dukic (01636420), Malte Klaes (01650623), Martina Ovcharova (11922120), and
Felix Windhager (12009495)

University of Vienna, 1010 Vienna, Austria

1 Gesamtbeschreibung und Hinweise zur Testbenutzung

Die App besteht insgesamt aus fünf Bildschirmen. Die erste Seite (Landingpage/MainActivity) beinhaltet eine erste Aufteilung; zum einen lässt sich hier in den Suchzweig wechseln, zum anderen kann man von hier aus den Bookmark-Screen aufrufen. Innerhalb des Bookmark-Screen lassen sich alle gebookmarkten Reiseverbindungen einsehen und zudem auch durch klicken/tippen auf den jeweiligen Stern löschen. Der nächste Screen (MainScreen) beinhaltet die Eingabe der Rahmenbedingungen, diese werden durch Klicken auf "Start Search" als Filter für den darauf folgenden Screen (SortScreen) verwendet, um mit deren Hilfe in der Datenbank nach geeigneten Verbindungen zu suchen. Hier sei angemerkt, dass die Datenbank (DataBase-MockUp.class) momentan gemockt ist und basierend auf den Sucheingabe im MainScreen situationsgerecht neu erstellt wird. Nach Auswahl eines oder im Falle von Hin- und Rückreise zweier Reisemitteln gelangt man auf den letzten Screen, dem Ergebnisscreen (ResultScreen), der die zuvor ausgewählten Reisemittel noch einmal übersichtlich darstellt und die Möglichkeit bietet, gewisse Verbindung in den Bookmarks zu speichern. **Zur Testnutzung:** Prinzipiell kann jegliche Stadt als Start eingegeben werden, da die Datenbank basierend auf der Eingabe immer wieder situationsabhängig neu generiert und gemockt wird. Allerdings ist im Suchscreen (MainScreen) aus einer vorgegebene Städteliste auszuwählen, die man mit Klick/Tippen in das Suchfeld ("Select Start", "Select Destination") und durch Eingabe eines Buchstabens angezeigt bekommt. Tippt man also "H" ein, so bekommt man beispielsweise "Hamburg" vorgeschlagen. Gibt man selber eine Stadt ein, die nicht vorgeben ist, so bekommt man dies durch ein Warning "City not found!" signalisiert.

2 Screen 1 und Screen 2 - MainActivity und MainScreen2

Beim Starten der App wird dem Benutzer der Startbildschirm angezeigt. Diese Seite ist als Home-Seite und für den einfachen Zugriff auf die Hauptfunktionen der App gedacht. Der Nutzer kann bequem über die interaktiven Schaltflächen navigieren, um eine Neue Suche für eine Reise zu starten, oder über den Lesezeichen-Button direkt zu seinen gespeicherten Routen gelangen. Hier haben wir den Schwerpunkt auf eine einfache Bedienung gelegt, um zu vermeiden, dass der Benutzer durch eine überfüllte oder komplizierte Benutzeroberfläche überfordert wird. Wie bereits in unserer Benutzeranalyse erwähnt, soll die App erschwinglich und für jeden Benutzertypen leicht verständlich und nutzbar sein.

Der zweite Screen verwaltet die Suche. Im oberen Abschnitt befinden sich drei Buttons zur Orientierung und Navigation des Gesamtablaufs. Die derzeitige Position (Search) wird schwarz hervorgehoben. Die anderen beiden Buttons (Sort und Search) sind heller gehalten. Durch das Klicken der Buttons kommt man zum jeweiligen Screen. Darunter folgen zwei Eingabefelder für den Startpunkt und die Destination. Die Eingabefelder haben eine autocomplete-Funktion, wobei in der angezeigten Liste im Voraus gespeicherte Städte vorgeschlagen werden. Rechts neben den Eingabefeldern ist ein switch-Button, der Startpunkt und Destination vertauscht. Als nächstes gibt es zwei Auswahlfelder mit "single" oder "return run", wodurch Hin- oder Hin- und Zurückfahrt ausgewählt werden kann. Beim Klicken des einen wird das andere abgewählt. Ist "single" ausgewählt, werden nur die Eingabefelder für Hinfahrtsdatum und Hinfahrtsuhrzeit angezeigt. Beim Auswählen von "return run" scheinen auch die Eingabefelder für Rückfahrtsdatum und Rückfahrtsuhrzeit

auf. Beim Klicken der Eingabefelder für das Datum erscheint eine Kalender-GUI, in welcher das gewünschte Datum ausgewählt wird. Für die Zeiteingabe erscheint eine Uhr-GUI in der die Zeit eingestellt oder in Zahlen eingegeben werden kann. Zum Schluss sind die Buttons "Home" und "StartSearch" mit denen entweder zur Home-Seite zurückgekehrt oder zum SortScreen weitergeschaltet wird. Mit "StartSearch" werden die eingegebenen Daten auf Korrektheit geprüft und anschließend an den SortSearch Screen weitergeschickt. Es wird überprüft, ob gültige Städte eingegeben wurden, ob bei Hin- und Zurückverbindungen die jeweiligen Datum- und Zeitangaben zusammenpassen. Bei einem Fehler wird dieser in einem Dialog-Screen eingeblendet und der User gebeten diese Daten zu überprüfen. Der Dialog wird mit "Ok" bestätigt. Für die Entwicklung dieses Bildschirms haben wir eng mit den Ergebnissen unserer Aufgabenanalyse zusammengearbeitet, da wir zu dem Schluss gekommen sind, dass die Such- und Präferenzfunktionen zu den am häufigsten verwendeten Funktionen gehören und daher im Grunde die Hauptfunktionalität darstellen. Mit Blick auf unsere Zielgruppe, aber auch auf kleinere, aber konkretere Nutzer, beispielsweise ältere Menschen, haben wir die Suchfelder und Schaltflächen in der entsprechenden Größe so gestaltet, dass sie für beide Usertypen komfortabel sind.

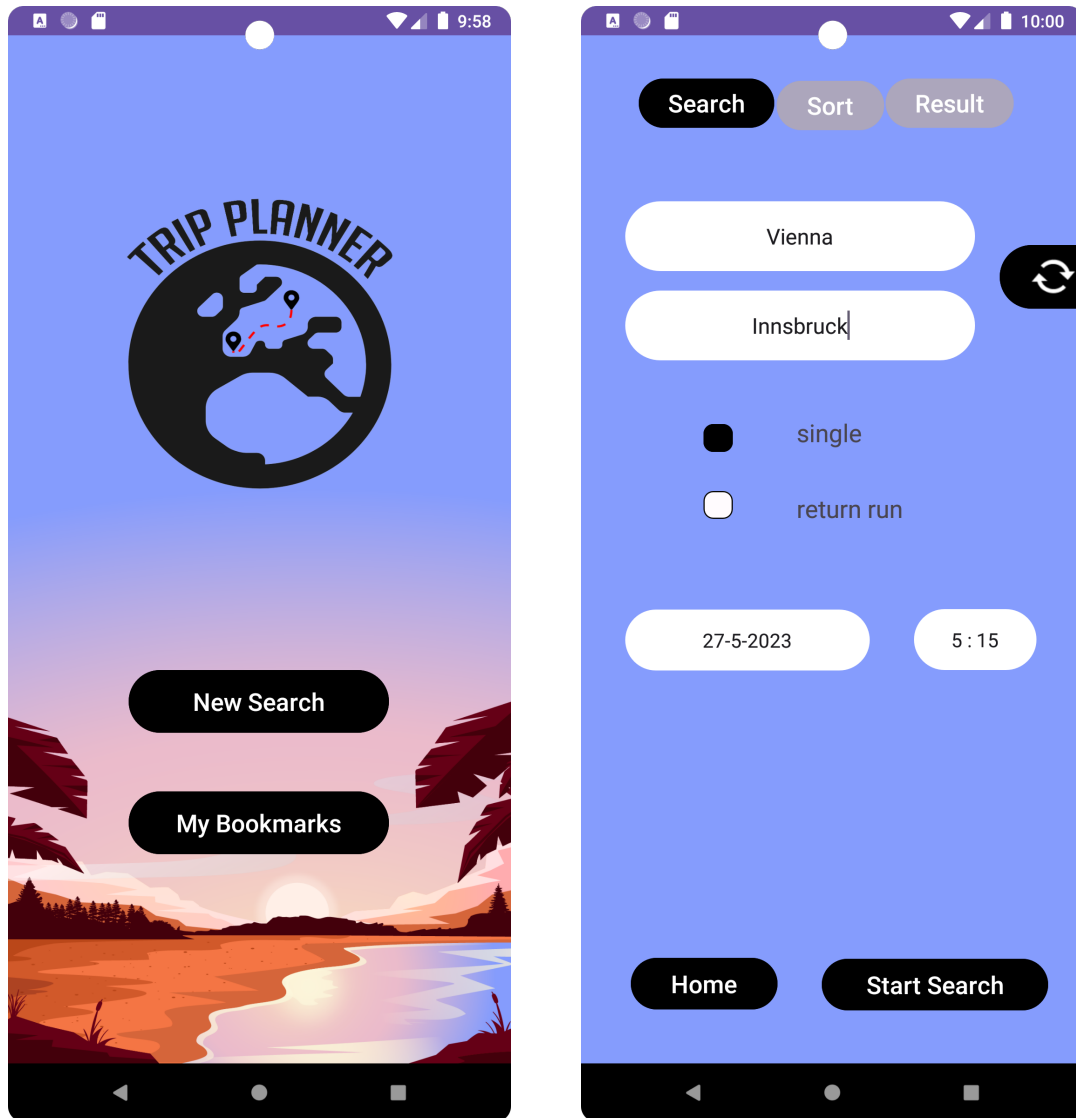


Fig. 1. Screenshots: Screen 1 and Screen 2 - MainActivity and MainScreen

3 Screen 3 - SortScreen

Der dritte Screen ist insgesamt gesehen in vier Großabschnitte aufgeteilt.

Im oberen Teil befinden sich erneut die drei Navigationsbuttons, die schon in Screen2 angesprochen wurden. In diesem Screen wird dann der Button "Sort" farblich hervorgehoben. Darunter ermöglicht ein Filterfeld (Spinner) dem User die unterschiedlichen Ergebnisse nach Präferenzen ("eco-friendly", "fast", "reliable", "comfortable", "cheap", "few stops") zu filtern (M1, main feature "Präferenzfunktion"). Die Ergebnisse werden dann in einem geteiltem Feld (ViewPager mit zwei Fragments und jeweils integrierter RecyclerView und customized Row-Elementen) nach Reisetransporttypen

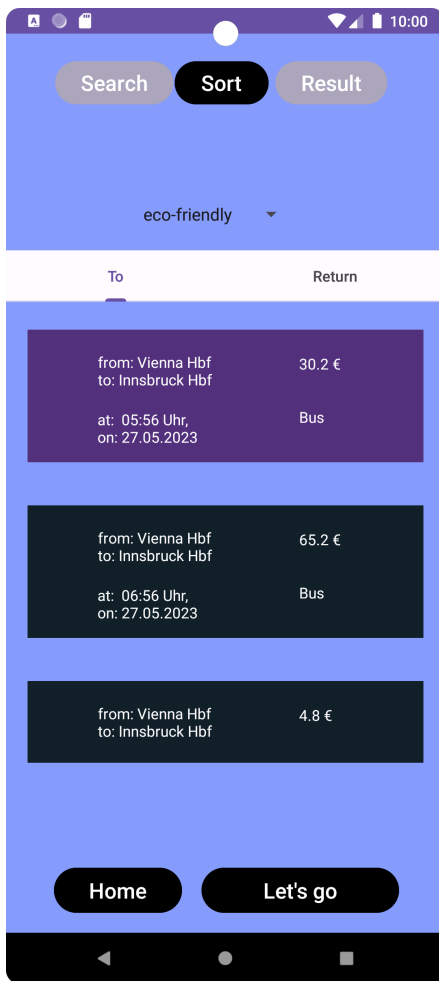


Fig. 2. Screenshot: Screen 3 - SortScreen

("Car_Sharing", "Bus", "Ship", "Train", "Plane", "Mix") alphabetisch aufsteigend sortiert (M1, main feature "Sortierfunktion") angezeigt.

Das Ergebnis wird auf einem Row-Element mit Start und Ziel, Startzeit und Startdatum, Preis (randomisiert) und Transporttyp angezeigt. Aus der Liste an angebotenen Ergebnissen kann der User dann eines durch klicken/tippen auf eines der Elemente auswählen oder seine Auswahl mit tippen auf ein anderes Feld korrigieren. Wird kein Ergebnis gefunden, so wird "nothing found!" angezeigt, wird nur ein Hinweg ausgewählt und der User tippt trotzdem auf das Feld "Return", so wird lediglich "only one way was chosen!" angezeigt.

Den letzten Abschnitt bilden die beiden Hauptfunktion-Buttons "Home" und "Let's go", wobei ersterer wieder den User auf die Suchfunktionsseite samt zuvor eingegebener Suchparameter zurückleitet und letzterer die angeklickte/angetippte Auswahl an den Ergebnisscreen schickt. Hat der User nichts oder nur teilweise etwas ausgewählt, so ist es nicht möglich mit "Let's go" auf den Ergebnisbildschirm zu wechseln und stattdessen bekommt der User den Hinweis "No or not enough journeys were chosen. Please check your traveltransport-choice!", womit verhindert wird, dass der User unnötig hin- und herklicken muss, um seine unvollständige Eingabe zu korrigieren.

Insgesamt sollen durch den Aufbau mittels Prozess-Übersicht durch die oberen drei Buttons, das schnelle Filtern für Hin- und Rückweg auf einer Seite mittels Filterauswahl und relativ kleingestalteten, dafür übersichtlicheren Row-Elementen die primären Personas, also junge Erwachsene und junge Familien angesprochen werden. Das Planen insgesamt, Umentscheidungen während dem Planen und

eine detaillierte Übersicht über Zwischenergebnisse soll durch diese Funktionen unterstützt werden.

4 Screen 4 - ResultScreen

Nachdem man im "Sort-Screen" seine gewünschte Reise ausgewählt hat, gelangt man über den "Let's go"-Button zum "Result-Screen", wo mehrere Details zur gewählten Reise angegeben werden.

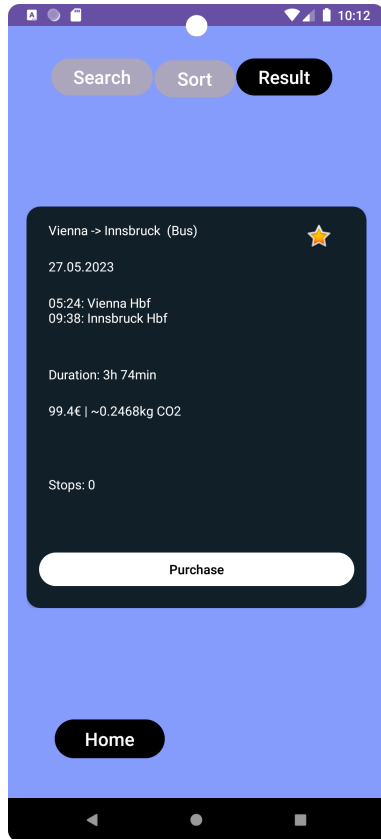


Fig. 3. Screenshot: Screen 4 - ResultScreen

Hat man zuvor zusätzlich ausgewählt, auch eine Rückreise buchen zu wollen, so werden Hin- und Rückreise angezeigt. Für jede Reise wird anschließend das Datum, die Abfahrts- und Ankunftszeit und die Gesamtdauer der Reise angezeigt. Des Weiteren werden Informationen bezüglich des Preises, CO2-Emissionen und der Anzahl an Umstiegen bereitgestellt. Gibt es bei der Reise mehrere Umstiege, so werden die einzelnen Reisen mitsamt Informationen über Ortsangaben, Ankunfts- und Abfahrtszeiten aufgeschlüsselt. In den Klammern wird immer das Reisetransportmittel angegeben. Für jede einzelne aufgezählte Reise gibt es nun außerdem noch die Funktion, die Reise als Lesezeichen zu speichern. Dazu klickt man auf den Stern in der rechten, oberen Ecke. Dieser färbt sich anschließend gelb und die ausgewählte Reise ist für später in der Lesezeichen-Kategorie gespeichert. Klickt man den Stern erneut, um das Lesezeichen wieder zu entfernen, so wird der User zur Sicherheit danach gefragt, ob er das Lesezeichen wirklich entfernen möchte. Erst nach der Bestätigung dieser Sicherheitsmaßnahme wird das Lesezeichen auch wirklich entfernt. Der User kann den Vorgang auch abbrechen, indem er auf "NO" klickt. Über den "Purchase"-Button wird der User nun noch auf eine Website geleitet, auf der die Tickets erwerbbar sind (in unserer App wird immer nur beispielhaft auf eine Website mit derartigen Angeboten geleitet). Die vier restlichen Buttons ("Search", "Sort", "Result", "Home") dienen zum Hin- und Herspringen zwischen den einzelnen Screens. Der Navigations-Button "Result" wird dabei farblich hervorgehoben.

Da sich unsere App vermehrt auf jüngere, reisebegeisterte Menschen fokussiert, setzen wir eine gewisse Vertrautheit mit "App-Standards" voraus. Der Stern als Speichersymbol ist in der Appentwicklung weit verbreitet und wird daher von jungen Nutzern mit dieser Funktionalität verbunden. Für ältere Nutzergruppen könnte diese Funktion nicht von vornherein klar sein.

5 Screen 5 - BookmarkScreen

Der Lesezeichenbildschirm soll dem Benutzer dabei helfen, einige für ihn interessante Routen leicht zu finden und darauf zuzugreifen. Diese Funktion bietet Komfort, da zum Speichern von Fahrten kein Konto erforderlich ist. Wenn eine Route gespeichert ist, kann sie entweder für einen späteren Kauf angeklickt oder gelöscht werden, wenn sie nicht mehr gewünscht wird. Die gesamte Lesezeichenliste wird auf dem Bildschirm angezeigt. Anschließend kann der Benutzer mit der Home-Taste zurück zur Startseite navigieren. Wie in unserer Aufgabenanalyse besprochen, besteht das Ziel dieser Funktionalität darin, dass der User seine gebuchten Verbindungen speichern kann und mit nur einem Klick leicht wieder auf die Informationen zu seinen ausgewählten Verbindungen zugreifen kann.

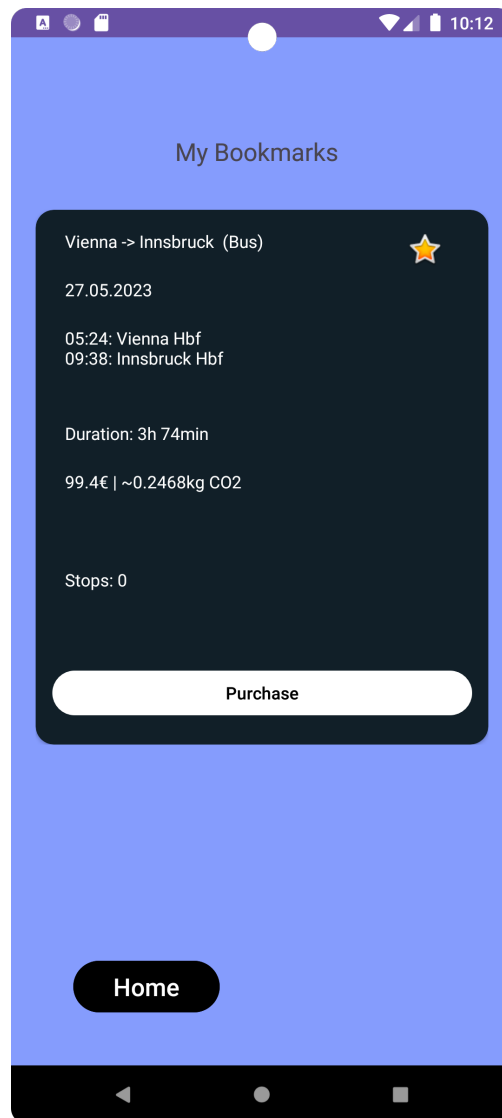


Fig. 4. Screenshot: Screen 5 - BookmarkScreen

6 Aufgabenverteilung

Im Folgenden sind die einzelnen Unterpunkte aufgelistet, an welche die einzelnen Projektmitglieder gearbeitet haben. Insgesamt standen insbesondere drei Screens (mainScreen2, sortScreen3, resultScreen4) und das Managen des Datenflusses zwischen den Klassen und die Stabilisierung der App im Vordergrund.

[1] **Milenko Dukic:**

- (a) **Hauptaufgabe:** mainScreen2.java, gesamte Hintergrundgestaltung (Logo, Hintergrundsbild, Farben, Elementanordnung und -gestaltung (u.a. Schriftart und -größe, Buttonart und -größe, etc.))
- (b) **Nebenaufgaben:** Datentyperstellung (DataBaseMockUp.java (allg. Logik), DataConnection.java, DataDate.java, DataTime.java, DataEnumTransportProperties.java, DataEnumTransport.java, DataEnumTimeComparison.java), Debugging und Testing

[2] **Malte Klaes:**

- (a) **Hauptaufgabe:** sortScreen3.java, SortScreen3fragment1.java, SortScreen3fragment2.java, SortScreen3RecyclerViewAdapter.java, ViewPagerAdapter.java
- (b) **Nebenaufgaben:** Datentyperstellung (DataBaseMockUp.java (allg. Logik), DataConnection.java, DataDate.java, DataTime.java, DataEnumTransportProperties.java, DataEnumTransport.java, DataEnumTimeComparison.java), Debugging und Testing

[3] **Martina Ovcharova:**

- (a) **Hauptaufgabe:** Datenübermittlung und -speicherung zwischen den Screens (Search/Sort/Result Buttons, Home/Start Search/Let's Go Button) (safe states und App-Stabilisierungsmaßnahmen), MainActivity.java, bookmarkScreen5.java, git-Repository Versionsverwaltung
- (b) **Nebenaufgaben:** Datentyperstellung (DataBaseMockUp.java (allg. Logik und Erstellung der konkreten Datengenerierung, DataConnection.java, DataDate.java, DataTime.java, DataEnumTransportProperties.java, DataEnumTransport.java, DataEnumTimeComparison.java), Debugging und Testing

[4] **Felix Windhager:**

- (a) **Hauptaufgabe:** resultScreen4.java, StopAdapter.java, ResultAdapter.java, insbesondere bookmark-Logik mit BookmarkAdapter.java
- (b) **Nebenaufgaben:** Datentyperstellung (DataBaseMockUp.java (allg. Logik), DataConnection.java, DataDate.java, DataTime.java, DataEnumTransportProperties.java, DataEnumTransport.java, DataEnumTimeComparison.java), Debugging und Testing