



Protokoll zu TourPlanner – Intermediate

Von Benjamin Lampart & Miriam Unger

SWEN 2 - 2025

Inhaltsangabe

Beschreibung der Intermediate.....	3
Diagramm (UML, Use-Case, Wireframe).....	5
Ordnerstruktur & Tests.....	7
Probleme, Fehler (und ihre Lösungen).....	8
GitHub-Link.....	9
Was haben wir bis jetzt draus gelernt?.....	9
Was sind die nächsten Schritte?	9
Investierte Zeit.....	10

Beschreibung der Intermediate

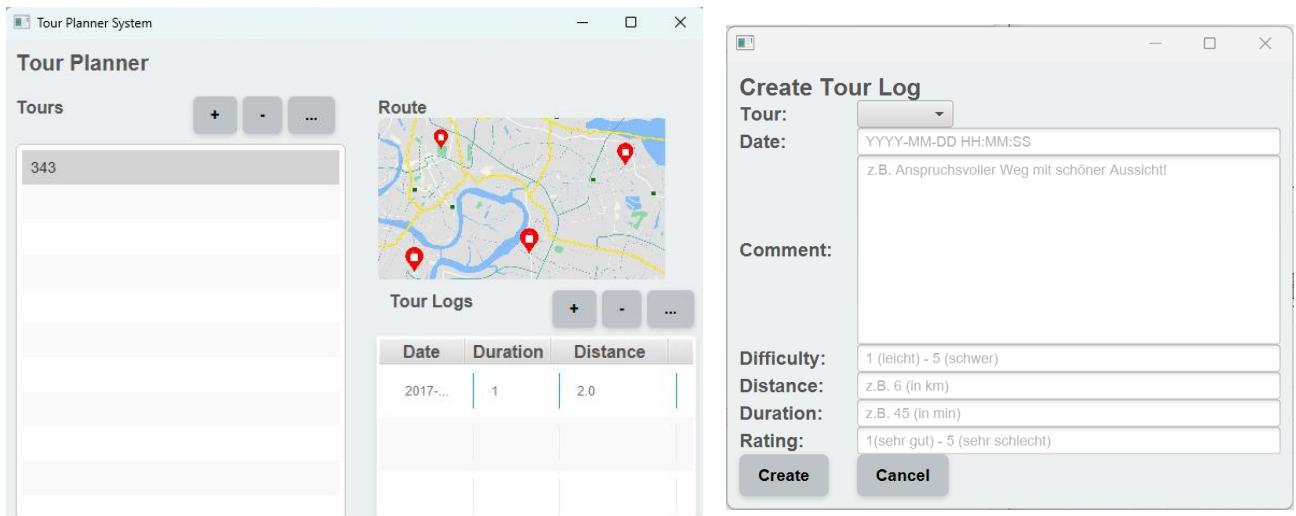
Das Intermediate Hand-In konzentriert sich auf die Kernfunktionen der Anwendung, insbesondere auf die Implementierung der grafischen Benutzeroberfläche (GUI) und des MVVM-Musters. In dieser Phase haben wir die grundlegenden UI-Funktionalitäten entwickelt und sichergestellt, dass sie benutzerfreundlich sind. Dies soll eine solide Basis für die spätere Integration der Business-Logik schaffen.

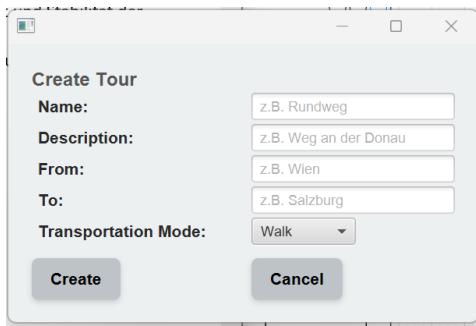
Bei der Gestaltung der Benutzeroberfläche haben wir uns eng an die in der Spezifikation vorgegebene Skizze gehalten. Unser Ziel war es, etwas ähnliches nachzubauen, da es sehr schön anzusehen war. Allgemein, war es so, dass man nicht viel Freiraum hatte, was Entscheidungen betrifft, was aber auch von Vorteil für die Gruppendynamik war.

Schwerpunkt:

- Touren-Management
 - o Touren:
 - Erstellung
 - Bearbeitung
 - Löschen
 - o Tour Logs
 - Erstellung
 - Bearbeitung
 - Löschen
 - o User Validation: Sicherstellung korrekter Eingaben und Benutzerführung
 - o ButtonMediator: Eine zentrale Steuerung zur Verwaltung von UI-Interaktionen
 - o UnitTests: Erste Tests zur Sicherstellung der Funktionalität und Stabilität der Anwendung
 - o Design: Fokus auf ein modernes und übersichtliches Layout

Allgemeines Aussehen und Beschreibung:





So sieht derzeit unsere Applikation aus. Die besteht aus mehreren Fenstern, die wir auch fürs Edit wieder verwenden, wie abgebildet. Hier wird sich aber noch ein bisschen etwas ändern, vor allem was die Bilder betrifft. Aus diesem Grund sieht man unten unsere WireFrame (Skizze) welche derzeit die Pläne veranschaulichen.

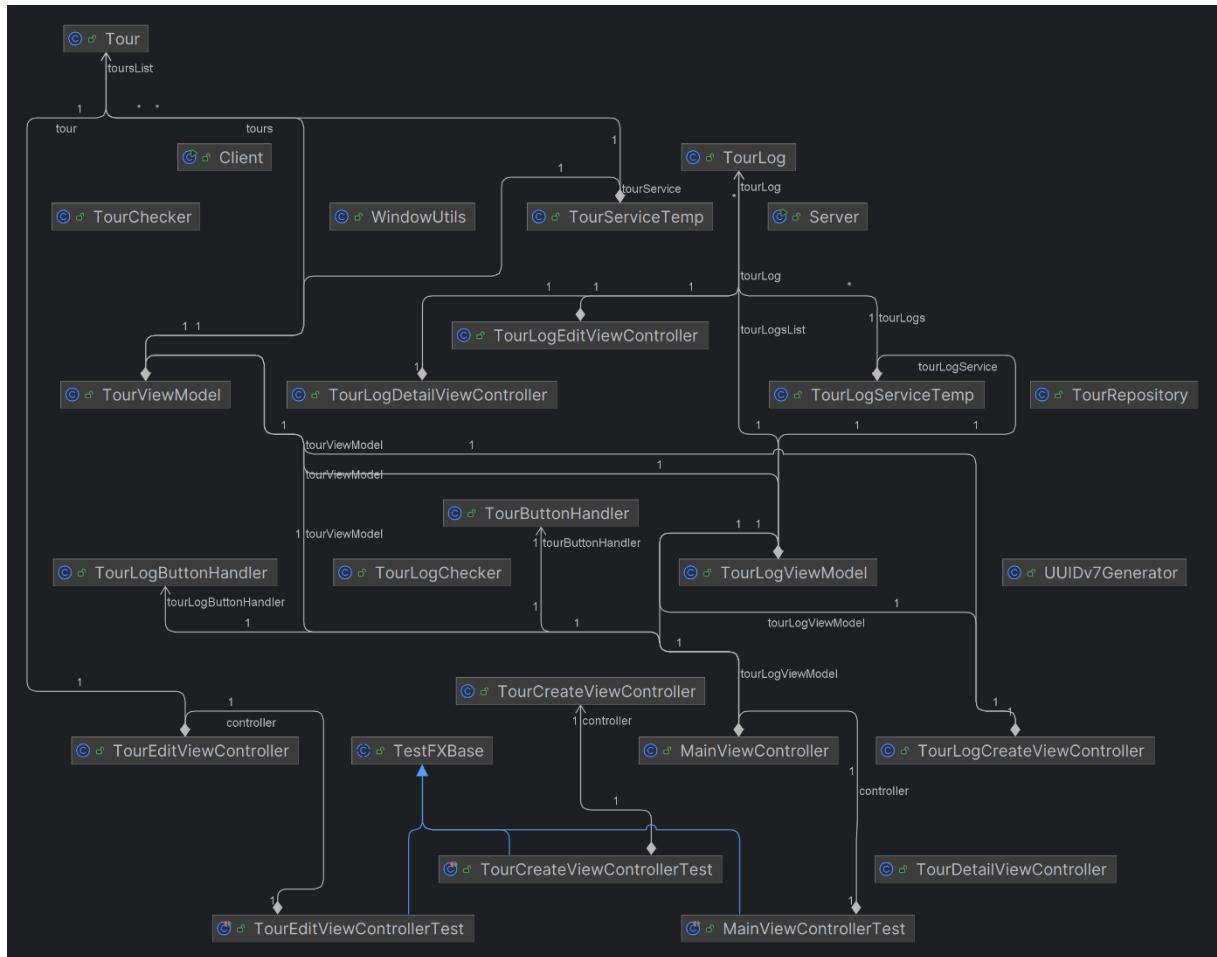
Aber weitere Informationen die man auf den Bildern nicht sieht:

Es gibt Buttons welche ausgegraunt werden, damit man sie nicht andrücken kann bei bestimmten Fällen. Wenn man bei Eingabefeldern etwas vergisst, dann bekommt man entweder Fehlermeldungen oder einen Roten Rand um den bestimmten Bereich. Wenn eine Tour gelöscht wird, werden auch die dazugehörigen TourLogs gelöscht. Und noch mehr, welche mir/uns gerade nicht mehr einfallen 😊 .

Diagramm (UML, Use-Case, Wireframe)

UML (Klassen):

Das UML-Klassendiagramm stellt die Struktur unseres Programms dar, indem es die Klassen, (und normalerweise ihre Attribute, Methoden und die Beziehungen zwischen den Klassen zeigt). Dies wäre nur für das Bild viel zu viel auf einmal.



Sequenz-Diagramm:

Folgt in Final

Use-Case:

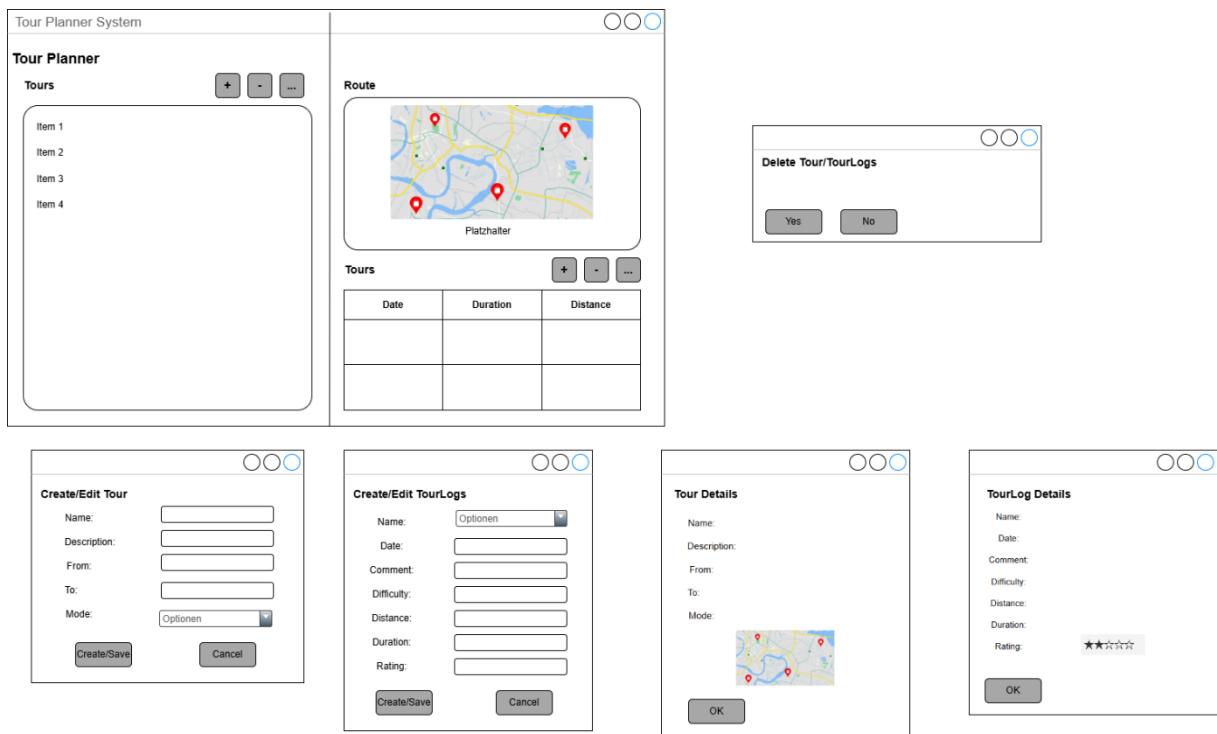
Das Use Case Diagramm zeigt die Interaktionen zwischen Nutzern und unserem System. Es stellt dar, welche Funktionen oder Anwendungsfälle das System ausführt, basierend auf den Anforderungen der Benutzer.

Folgt in Final

WireFrame:

Ein Wireframe-Diagramm ist eine schematische Darstellung der Benutzeroberfläche (UI) einer Website oder App. Es zeigt die grundlegende Struktur und Layouts von Seiten, ohne

detaillierte Design-Elemente wie Farben oder Grafiken. Ziel ist es, die Anordnung der Inhalte und Funktionen klar darzustellen, um das Benutzererlebnis zu planen.

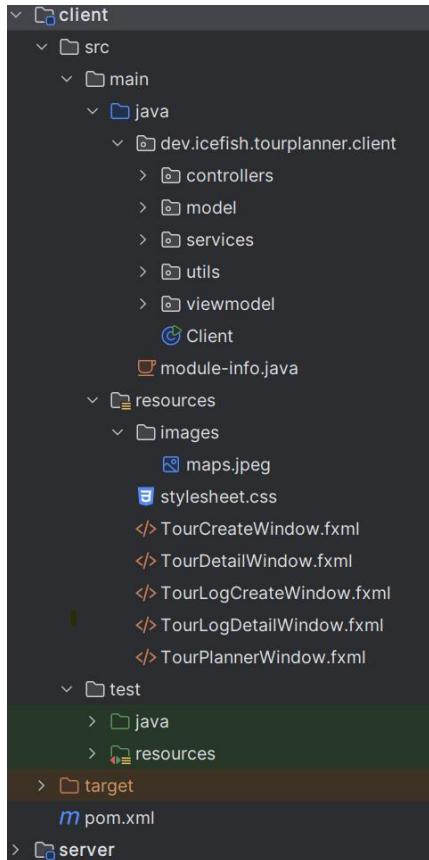


WireMock:

Folgt in Final

Ordnerstruktur & Tests

Das Projekt ist in eine Client- und eine Server-Komponente unterteilt. Die Codebasis folgt einer Architektur mit klar getrennten Verantwortlichkeiten mit der Verwendung von dem MVVM Pattern.



In unserem Projekt wird das MVVM (Model-View-ViewModel)-Pattern verwendet, um eine Trennung zwischen der Benutzeroberfläche (View), der Geschäftslogik (Model) und der Präsentationslogik (ViewModel) zu gewährleisten.

UI Tests:

Wir haben insgesamt derzeit 6 Tests welches das User-Interface und die zugehörigen Methoden testet.

- testDeleteTour()
- testDeleteTourLogs()
- testCreateTour()
- testEditTour()
- testCreateTourLog()
- testEditTourLog()

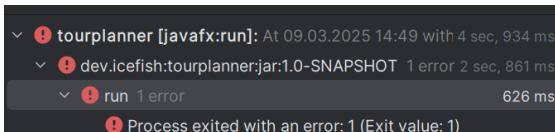
Und wie die einzelnen Namen schon sagen, testen sie genau das. Ausführen kann man mit maven -> Lifecycle -> test.

Probleme, Fehler (und ihre Lösungen)

Wir müssen ehrlich zugeben, Fehler kann man das bei unserem Programm eigentlich gar nicht nennen. Wenn etwas nicht funktioniert hat, lag es nicht unbedingt an einem Fehler im klassischen Sinne, sondern eher daran, dass die gewählte Weise oder Technik einfach nicht zur derzeitigen Aufgabe gepasst hat. Es war mehr eine Frage der Anpassung und Optimierung, als dass wir von wirklichen "Fehlern" sprechen konnten. Und ein weiterer großer Punkt ist die Sache, dass wir ja noch Lernen, es gehört dazu Sachen komplizierter zu machen als sie eigentlich sind, dafür sind wir doch hier, denn irgendwann mit der Zeit wird es einfacher.

Dennoch gab es mehrere Herausforderungen, die wir überwinden mussten:

- Der Aufbau mit FXML war anfangs echt schwer: FXML war eine große Hürde, die uns vor allem am Anfang zum Verzweifeln gebracht hat.
- . Besonders bei der Implementierung von responsiven Layouts hatte ich (Miriam) viele „Aussetzer“, und ich hoffe das es jetzt so passt, sicher bin ich mir aber nicht.
- Unnötige Fehlermeldungen von FXML: Die Fehlermeldungen von FXML sind wirklich ein Grauen. Sie sagen einem nichts Konkretes und führen einen in die Irre, sodass man ewig nach dem Fehler suchen muss, der eigentlich ganz einfach zu beheben wäre. Das war und ist immer noch mehr als frustrierend.
- Mein Perfektionismus und das Stylesheet: Ich (Miriam) bin leider etwas zu perfektionistisch, was zu einigen Schwierigkeiten geführt hat. Das Stylesheet ist mein bester Freund, aber es ist auch ein echtes Manchmal tut er einfach nicht, was ich von ihm will – egal ob es die Schriftart, die einfach nicht geladen wird, oder das Layout, das nicht passt, obwohl der Code eigentlich korrekt ist. Wenn etwas nicht 100% perfekt aussieht, treibt mich das fast in den Wahnsinn.
- Problem bei Miriam und nicht bei Benjamin: Ein weiterer Punkt war der Fehler der dann gottseidank durch unseren Lektor behoben wurde, aber die Fehlermeldung war absolut frustrierend



```
✗ ① tourplanner [javafx:run]: At 09.03.2025 14:49 with 4 sec, 934 ms
  ✓ ① dev.icefish:tourplanner:jar:1.0-SNAPSHOT 1 error 2 sec, 861 ms
    ✓ ① run 1 error
      ① Process exited with an error: 1 (Exit value: 1)
```

GitHub-Link

GitHub-Link: <https://github.com/icefishii/tourplanner>

(Wir haben aber auch die Zip abgegeben)

Was haben wir bis jetzt draus gelernt?

Das hier sind einige Punkte die uns im Laufe unseres Projektes aufgefallen sind

- Git erleichtert die Arbeit enorm, aber dennoch fragen wir einander immer ob der Kollege gerade daran arbeitet damit es keine Fehler/Probleme beim Pushen gibt.
- Die To-Do-Funktion in IntelliJ ist ein echtes Highlight, denn man kann sich die Liste immer anschauen und updaten und der andere sieht was gerade zu tun ist .

```

▼ Found 8 TODO items in 5 files
  ▼ dev.icefish.tourplanner.client 5 items
    ▼ controllers 4 items
      ▼ MainViewController.java 4 items
        [ (313, 7) //ToDo the distance, and the time should be retrieved by a REST request using the OpenRouteservice.org API
        [ (315, 7) //ToDo better Detail-Output for Tours and TourLogs
        [ (317, 7) //ToDo Keyboard-Shortcuts
        [ (319, 7) //ToDo Mandatory Feature (Language, ???)
      ▼ services 1 item
        ▼ TourService.java 1 item
          [ (14, 3) //TODO: Implement the REST API in he server
    ▼ </> TourDetailWindow.fxml 1 item
      [ (9, 5) <!-- ToDo Style auslagern-->
    ▼ </> TourLogDetailWindow.fxml 1 item
      [ (8, 5) <!-- ToDo Style auslagern-->
    ▼ </> TourPlannerWindow.fxml 1 item
      [ (120, 29) <!-- TODO: Ändern wie Daten ausgegeben werden -->

```

- Kontinuierliches Arbeiten kann zwar anstrengend sein, macht die Abgabe aber entspannter. Denkt man. Irgendwie fühlt es sich falsch an früher abzugeben und zu sagen, dass man fertig ist. Weil dann regt man sich erst recht auf, wenn etwas bei der Abgabe nicht passt.
- Auch wenn die Vorgaben anfangs unverständlich wirken, erkennt man am Ende, was gemeint war.
- Auch die Teamkommunikation wurde mit der Zeit besser, obwohl es manchmal ein paar Missverständnisse gab.

Was sind die nächsten Schritte?

- Kleinigkeiten:
 - Internationalisierung (Sprachen)
 - Keyboard-Shortcuts
 - Style anpassen
 - Images
 - und vieles mehr
- Größere Dinge:
 - API zum Berechnen der Zeit und Distance
 - REST
 - Datenbank

Investierte Zeit

Das hier ist unsere getrackte Zeit für das Projekt „TourPlanner“. Die getrackten Zeiten sollten an sich vollständig und korrekt sein, um bei uns am Ende eine Gesamtzeit von etwa 53 Stunden.

Wir tracken unsere Zeit mit Clockify :)

<https://app.clockify.me/>

Screenshots:

Februar

● TourPlanner	Add tags	\$	Miriam Unger ▾	10:24 11:19 25/02/2025	00:55:00	○ ○
● TourPlanner	Add tags	\$	Miriam Unger ▾	11:21 13:43 20/02/2025	02:22:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	16:31 16:57 17/02/2025	00:26:00	○ ○

März

● TourPlanner	Add tags	\$	Miriam Unger ▾	08:50 13:27 Today	04:36:36	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	20:18 20:18 19/03/2025	00:00:01	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	19:15 20:17 19/03/2025	01:02:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	11:30 12:30 19/03/2025	01:00:00	○ ○
● TourPlanner	Add tags	\$	Miriam Unger ▾	16:39 18:41 18/03/2025	02:02:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	10:17 12:58 16/03/2025	02:41:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	14:16 17:39 15/03/2025	03:23:00	○ ○
● TourPlanner	Add tags	\$	Miriam Unger ▾	20:30 22:13 14/03/2025	01:43:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	19:21 22:44 13/03/2025	03:23:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	16:34 16:52 13/03/2025	00:18:00	○ ○
● TourPlanner	Add tags	\$	Benjamin Lamp... ▾	15:15 16:02 13/03/2025	00:47:00	○ ○

● TourPlanner	Add tags	\$	Benjamin Lamp...	09:43 10:11 13/03/2025	00:28:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	19:41 21:10 12/03/2025	01:29:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	19:12 22:14 12/03/2025	03:02:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	14:15 15:42 12/03/2025	01:27:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	12:15 15:12 12/03/2025	02:57:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	16:10 16:27 10/03/2025	00:17:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	15:45 16:00 10/03/2025	00:15:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	14:00 15:12 10/03/2025	01:12:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	14:32 18:18 09/03/2025	03:46:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	13:46 14:27 09/03/2025	00:41:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	09:15 11:15 09/03/2025	02:00:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	17:14 18:42 08/03/2025	01:28:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	10:30 14:30 08/03/2025	04:00:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	11:27 12:41 06/03/2025	01:14:00	○○○
● TourPlanner	Add tags	\$	Benjamin Lamp...	13:22 15:37 05/03/2025	02:15:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	14:33 16:45 04/03/2025	02:12:00	○○○
● TourPlanner	Add tags	\$	Miriam Unger ▾	08:30 10:04 03/03/2025	01:34:00	○○○

Gesamt:

