Konzept - GPS-Tracker

Im Rahmen des Praktikum Software Engineering soll eine Applikation zur Verwaltung von „GPS-Tracks“ erstellt werden. Das Ziel dieser Applikation ist es Nutzern zu ermöglichen ihre aufgezeichneten “GPS-Tracks” zu organisieren, anzusehen und auszuwerten.

Die Applikation wird dabei in drei Sprints umgesetzt und soll spätestens am 19. Juli 2022 vollständig fertiggestellt werden. Der Auftraggeber des Projektes ist DI Dr. Johannes Sametinger und das Team zur Umsetzung bilden Ozan Akkoyun, Nuray Seker und Natalie Punz.

Die Applikation ist für sportbegeisterte Personen vorgesehen, welche ihre Aktivitäten elektronisch (ergo in einem der unterstützten Dateiformate) aufzeichnen und diese in einem Programm verwalten möchten. Die Nutzung der Applikation ist als “Single User”-Applikation bestimmt, weshalb nur ein Nutzer pro Applikationsinstallation gedacht ist. Zur Umsetzung soll in der Programmiersprache Java eine entsprechende Applikation entwickelt, getestet und schließlich veröffentlicht werden.

# Funktionale Anforderungen

1. Die Applikation ermöglicht das Einlesen von Dateien im Format “.tcx” und “.gpx” (folglich GPS-Dateien genannt).
   1. Wenn die Applikation gestartet wird, sollen alle GPS-Dateien im ersten Unterverzeichnis des beim Programmstart vom Nutzer gewählten Verzeichnis durch die Applikation eingelesen werden.
      1. Falls sich im gewählten Verzeichnis kein weiteres Unterverzeichnis befindet, soll die Applikation keine GPS-Dateien einlesen.
   2. Wenn der Nutzer das Aktualisieren der Daten auswählt, soll die Applikation alle im aktuell gewählten Unterverzeichnis befindlichen GPS-Dateien einlesen.
      1. Falls sich im Unterverzeichnis eine Datei mit einem “GPS-Track”, welche schon in der Applikation vorhanden ist, befindet, dann soll die Applikation das Einlesen dieser Datei nicht durchführen.
      2. Die Bestimmung ob jener “GPS-Track” sich schon in der Applikation befindet, erfolgt über den Namen der entsprechenden GPS-Datei.
   3. Wenn eine Datei geparst wird, soll die Applikation folgende Informationen, falls vorhanden, über den darin enthaltenen “GPS-Track” in der Applikation speichern: Name des “GPS-Tracks”, Datum der Datenaufzeichnung, Startzeitpunkt der Datenaufzeichnung, zurückgelegte Distanz, benötigte Zeit, durchschnittliche Pace, durchschnittliche Geschwindigkeit, durschnittlicher Herzschlag pro Minute, maximaler Herzschlag pro Minute, positiver Höhenunterschied und die enthaltenen GPS-Datenpunkte.
      1. Der Name des “GPS-Tracks” ist ident dem Namen der dazugehörigen Datei.
      2. Wenn die GPS-Datenpunkte eingelesen werden, soll die Applikation ebenso, falls vorhanden, pro GPS-Datenpunkt eine fortlaufende Nummer innerhalb des “GPS-Tracks”, die zurückgelegte Distanz, die benötigte Zeit, die durchschnittliche Pace, die durchschnittliche Geschwindigkeit, der durschnittlicher Herzschlag pro Minute, der maximaler Herzschlag pro Minute und den positiver Höhenunterschied in der Applikation speichern.
   4. Falls ein Fehler während dem Parsen einer Datei auftritt, soll die Applikation das Einlesen dieser Datei abbrechen und mittels Fehlermeldung den Nutzer informieren. Weitere Dateien im gewählten Verzeichnis sollen jedoch weiter eingelesen werden.
   5. Annahme: Es werden nur Dateien im Format .tcx (Schemaversion 2) und .gpx (Schemaversion 1.1) akzeptiert.
      1. Jede Datei bildet dabei einen einzulesenden GPS-Track.
2. Die Applikation soll eine Listenansicht von den einzelnen eingelesenen „GPS-Tracks“ unterstützen.
   1. Während die Listenansicht der einzelnen eingelesenen „GPS-Tracks“ aktiv ist, soll die Applikation pro “GPS-Track”, falls vorhanden, den Namen, das Datum der Datenaufzeichnung, den Startzeitpunkt der Datenaufzeichnung, die zurückgelegte Distanz, die benötigte Zeitdauer, die durchschnittliche Pace, die durchschnittliche Geschwindigkeit, der durchschnittliche Herzschlag pro Minute, der maximale Herzschlag pro Minute und den positive Höhenunterschied anzeigen.
   2. Unter der Listenansicht sollen alle “GPS-Tracks” hinsichtlich ihrer Attribute (siehe Punkt 1.3) kombiniert werden und entsprechend eine Summe oder ein Durchschnitt pro Attribut aller “GPS-Tracks” angezeigt werden.
      1. Name, Datum der Datenaufzeichnung und Startzeitpunkt der Datenaufzeichnung sind nicht zu kombinieren.
3. Die Applikation ermöglicht das Filtern der “GPS-Tracks” nach einer vom Nutzer kreierten Filterliste.
   1. Die möglichen Filter-Einträge basieren dabei auf der Verzeichnisstruktur des aktuell aktiv gewählten Verzeichnis.
      1. Ein Filter-Eintrag entspricht dabei einem Unterverzeichnis aus dem aktuell aktiv gewählten Verzeichnis.
   2. Wenn der Nutzer einen Filter-Eintrag auswählt, soll die Applikation alle aktuellen “GPS-Tracks” durch jene in dem entsprechendem Verzeichnis ersetzen.
   3. Während die Applikation ausgeführt wird, ist ein Filter-Eintrag immer aktiv. Der voreingestellte Filter-Eintrag bei Applikationsstart ist, falls vorhanden, das erste Verzeichnis.
   4. neue Unterverzeichnis bei programmausführung?
4. Die Applikation soll eine zweite Listenansicht unterstützen.
   1. Wenn der Nutzer einen einzelnen “GPS-Tracks” in der Listenansicht auswählt, sollen die in diesem “GPS-Track” enthaltenen einzelnen aufgezeichneten GPS-Datenpunkte laut einem Segmentierungsargument von der Applikation gruppiert und dargestellt werden.
      1. Die Segmentierung kann dabei mittels 1 Meter, 10 Meter, 100 Meter, 400 Meter, 500 Meter, 1000 Meter, 5000 Meter, 10000 Meter, ein Viertel eines Marathons, ein halber Marathon und pro gespeichertem GPS-Datenpunkt erfolgen. Bei Auswahl einer Distanz sollen die GPS-Datenpunkte hinsichtlich ihrer Distanz addiert werden bis das gewünschte Segmentierungsargument am nächsten erreicht wird. Diese GPS-Datenpunkte bilden gemeinsam dann ein Segment und es sollen alle Informationen pro Segment, wie bei Punkt 4.1.2 erläutert, errechnet werden. Dies wird fortgesetzt bis alle GPS-Datenpunkte einem Segment zugewiesen sind. Während die Applikation ausgeführt wird, ist ein Segmentierungsargument immer aktiv. Das voreingestellte Argument bei Applikationsstart ist GPS-Datenpunkt.
         1. Annahme: Die möglichen Segmentierungsargumente sind zur Laufzeit fixiert und können nicht erweitert werden.
      2. Wenn der Nutzer einen einzelnen “GPS-Tracks” in der Listenansicht auswählt, sollen folgende Informationen, falls vorhanden, präsentiert werden: Nummer des Segmentes, kombinierte zurückgelegte Distanz, kombinierte benötigte Zeit, die durchschnittliche Pace, die durchschnittliche Geschwindigkeit, der durchschnittliche Herzschlag pro Minute, der maximale Herzschlag pro Minute und der positive Höhenunterschied.
      3. Falls ein Fehler auftritt, soll die Applikation die zweite Listenansicht als unverändert anzeigen.
   2. Wenn innerhalb der Listenansicht kein Eintrag gewählt worden ist, soll die Applikation die zweite Listenansicht als leer anzeigen.
5. Die Applikation soll das Entfernen von geladenen „GPS-Tracks“ ermöglichen.
   1. Wenn der Nutzer das Aktualisieren der Daten auswählt, soll die Applikation “GPS-Tracks”, welche keine zugehörige Datei im gewählten Unterverzeichnis aufweisen, entfernen.
   2. Falls ein Fehler beim Entfernen auftritt, soll die Applikation mittels einer Fehlermeldung den Nutzer darüber informieren.
6. Die Applikation soll das Gruppieren von „GPS-Tracks“ nach Zeiträumen ermöglichen.
   1. Diese Gruppierung kann nach dem vollem Datum, wöchentlich (innerhalb von Jahr), monatlich (innerhalb von Jahr) oder jährlich geschehen.
      1. Annahme: Die Gruppierungsargumente sind fixiert und können zur Laufzeit nicht erweitert werden.
   2. Wenn der Nutzer die Gruppierung auswählt, soll die Applikation die “GPS-Tracks” entsprechend dem Gruppierungsargument gruppieren und in der Listenansicht (siehe Punkt 2) als Gruppen präsentieren.
   3. Wenn der Nutzer ein Gruppierungsargument auswählt, soll die Applikation folgende Informationen pro Gruppe in der Listenansicht anzeigen: Zeitraum der Gruppierung, Anzahl der Elemente in der Gruppe, kombinierte zurückgelegte Distanz, kombinierte benötigte Zeit, durchschnittliche Pace, durchschnittliche Geschwindigkeit, durschnittlicher Herzschlag pro Minute, maximaler Herzschlag pro Minute und kombinierter positiver Höhenunterschied.
   4. Wenn der Nutzer eine Gruppe in der Listenansicht auswählt, soll die Applikation alle darin enthaltenen “GPS-Tracks” dieser Gruppe in der zweiten Listenansicht anzeigen (siehe Punkt 4). Die anzuzeigenden Informationen pro “GPS-Track” sind dieselben wie bei Punkt 2.1.
   5. Falls ein Fehler auftritt, soll die Applikation die Listenansicht beziehungsweise zweite Listenansicht unverändert angezeigt werden.
7. Die Applikation soll ein Balkendiagramm zum Vergleich der “GPS-Tracks” unterstützen.
   1. Die Argumente auf der y-Achse können zurückgelegte Distanz, benötigte Zeit, durchschnittliche Pace, durchschnittliche Geschwindigkeit, durschnittlicher Herzschlag pro Minute, maximaler Herzschlag pro Minute und positiver Höhenunterschied sein.
      1. Wenn der Nutzer eine Einheit für die y-Achse auswählt, soll die Applikation das Balkendiagramm entsprechend dem gewählten Argument anpassen.
      2. Während die Applikation ausgeführt wird, ist ein y-Achse-Argument immer aktiv. Das voreingestellt Argument ist zurückgelegte Distanz.
      3. Annahme: Die y-Achse-Argumente sind fixiert und können nicht zur Laufzeit erweitert werden.
   2. Während die Applikation ausgeführt wird, soll ein Balkendiagramm angezeigt werden, welches sich der Auswahl des Nutzers zur Laufzeit anpasst.
      1. Während sich die Applikation in der Listenansicht ohne Gruppierung befindet, sollen alle “GPS-Tracks”, die sich in der Listenansicht befinden, laut Argument der y-Achse dargestellt werden. Die Namen der “GPS-Tracks” bilden die Inhalte der x-Achse.
      2. Während sich die Applikation in der Listenansicht ohne Gruppierung aber mit ausgewähltem “GPS-Track” befindet, sollen alle Segmente (siehe Punkt 4.1) laut Argument der y-Achse dargestellt werden. Die Nummer der Segmente bilden die Inhalte der x-Achse.
      3. Während sich die Applikation in der Listenansicht mit Gruppierung befindet, sollen alle Gruppierungen laut Argument der y-Achse dargestellt werden. Der Zeitraum der Gruppierungen bilden die Inhalte der x-Achse.
      4. Während sich die Applikation in der Listenansicht mit Gruppierung und ausgewählter Gruppe befindet, sollen die in der Gruppe befindlichen “GPS-Tracks” laut Argument der y-Achse dargestellt werden. Die Namen der “GPS-Tracks” in der Gruppe bilden die Inhalte der x-Achse.
   3. Falls ein Fehler auftritt, soll die Applikation das Balkendiagramm unverändert anzeigen.
8. Die Applikation soll einen jährlichen Vergleich der “GPS-Tracks” unterstützen.
   1. Wenn der Nutzer mindestens zwei Jahre (maximal alle vorhandenen Jahre) auswählt, sollen diese Jahre anhand des aktuellen Gruppierungsargument auf dem Balkendiagramm mit dem aktiven y-Achse-Argument (siehe Punkt 7.1) verglichen werden.
      1. Falls kein Gruppierungsargument aktiv ist, soll die Applikation den Vergleich mittels Gruppierungsargument “Monat” durchführen.
      2. Die möglichen wählbaren Jahre sind dabei abhängig von den eingelesenen “GPS-Tracks”.
9. Die Applikation soll die Suche nach den Namen der “GPS-Tracks” ermöglichen.
   1. Wenn der Nutzer in das Suchfeld einen Suchbegriff eingibt, soll die Applikation die zu der Zeit aktiv angezeigte Liste in der Listenansicht (siehe Punkt 2) nach den entsprechenden Namen der “GPS-Tracks” filtern. Es sollen nur jene “GPS-Tracks” angezeigt werden, welche diesem Suchbegriff entsprechen.
   2. Falls ein Gruppierungsargument aktiv ist bei der Suche, soll die Applikation jene Gruppen präsentieren, welche “GPS-Tracks”, die dem Suchbegriff entsprechen, innehaben .
10. Die Applikation soll das Schließen der Applikation ermöglichen.
    1. Wenn der Nutzer die Applikation schließt, soll die Applikation alle offenen Fenster schließen.

# Nicht-Funktionale Anforderungen

## Zur Klärung der Nicht-Funktionalen Anforderungen orientieren wir uns an den ISO-9126 Standard.

Wartbarkeit

Die Applikation soll möglichst wartbar gestaltet werden. Dafür sollen entsprechende “Best-Practices” der Programmiersprache Java aber auch allgemeinere “Best-Practices” der objekt-orientierten Programmiersprachen eingehalten werden.

Pro 100 Stunden Betriebszeit soll höchstens ein Absturz vorkommen Eine hohe Stabilität ist daher von enormer Wichtigkeit.

Zur Sicherstellung der Wartbarkeit sollen ebenso nach einer Analyse des Programms “SonarQube” weder “critical” noch “major” Issues erkannt werden.

## Benutzbarkeit

Die Applikation soll klar und verständlich für den Nutzer sein. Ein Nutzer sollte die Applikation während der erstmaligen Nutzung vollstens bedienen können ohne die Benutzerdokumentation zu konsultieren. Die Nutzung muss daher möglichst intuitiv sein und für den Nutzer einem bereits bekannten Nutzungsstil entsprechen.

Die Benutzeroberfläche muss daher möglichst modern, ästhetisch aber auch funktionell sein.

## Effizienz

Die Applikation soll möglichst ressourcenarm in der Ausführung sein. Es sollen während der Ausführung bei einer Anzahl von 250 eingelesenen “GPS-Tracks” maximal 200 Megabyte Arbeitsspeicher verwendet werden. Beim Start der Applikation soll das Einlesen und Anzeigen von 250 “GPS-Tracks” (mit einer kombinierten Speichergröße von 70 Megabyte) höchstens eine Sekunden benötigen. Das Ändern eines Gruppierungsargument und der damit einhergehenden Umstellung der graphischen Oberfläche darf bei 250 “GPS-Tracks” höchstens eine halbe Sekunde benötigen. Das Ändern des Filter-Eintrags oder des gewählten Verzeichnisses und der damit einhergehenden Umstellung der graphischen Oberfläche darf bei 250 “GPS-Tracks” (mit einer kombinierten Speichergröße von 70 Megabyte) höchstens eine halbe Sekunde benötigen. Das Aktualisieren der vorhandenen “GPS-Tracks” während der Laufzeit und der damit einhergehenden Umstellung der graphischen Oberfläche darf bei 250 “GPS-Tracks” (mit einer kombinierten Speichergröße von 70 Megabyte) höchstens eine halbe Sekunde benötigen.

Jegliche andere Interaktionen sollen innerhalb von einer halben Sekunde abgeschlossen sein.

Diese Werte sind auf einem handelsüblichen Laptop (2-Kern-Prozessor ab Intel i5-6200U oder Ähnlichem) zu erreichen.

## Funktionalität

Die Applikation soll nur jene Funktionen anbieten, die in ihrem Einsatzkontext auch Sinn ergeben. Diese Funktionen müssen richtig (ergo den Anforderungen entsprechend) implementiert werden.

Die Applikation interagiert mit keinen anderen Systemen und muss dafür entsprechend keine Unterstützung anbieten.

Die Applikation interagiert nicht mit anderen Diensten und hat ebenso keine direkten Netzwerkverbindungen. Jegliche Daten werden ebenso nur während der Ausführung in der Applikation gehalten und werden (außerhalb des Aufgabengebietes der Applikation) persistent in durch vom Betriebssystem verwaltete Dateien gehalten. Der Sicherheitsaspekt ist daher zu vernachlässigen.

Die Applikation soll Daten korrekt verarbeiten. Die Gruppierung, Sortierung und Filterung von “GPS-Tracks” ist korrekt durchzuführen. Beim Einlesen der GPS-Dateien sollen ebenso alle notwendigen Informationen korrekt und vollständig ausgelesen werden.

Übertragbarkeit

Durch die Nutzung von Java als Programmiersprache ist eine hohe Übertragbarkeit von Beginn an gegeben. Die Applikation soll auf “java-fähigen” (ab Version 11) Computern ausführbar sein. Die Interaktion mit dem Nutzer findet über Maus und Tastatur statt. Die Applikation ist mit jedem Betriebssystem kompatibel, welches Java Programme ausführen kann. Eine Installation der Applikation ist nicht notwendig.

Eine möglichst gute Koexistenz mit anderen Programmen soll gegeben sein. Die Nutzung von etwaigen Ressourcen, wie zum Beispiel den GPS-Dateien beim Einlesen, soll nach Abschluss oder Fehlermeldung wieder freigegeben werden.

Zuverlässigkeit

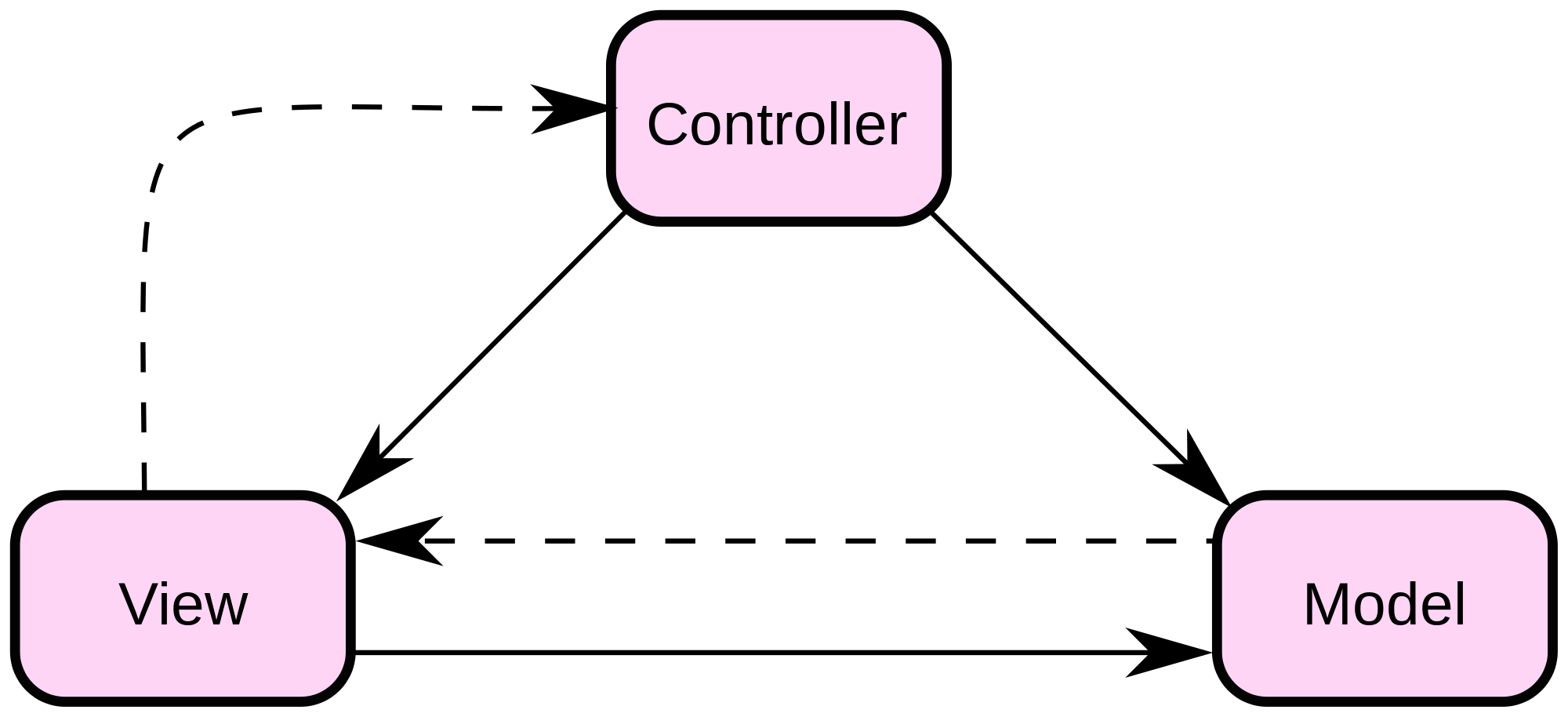
Die Applikation soll auch beim Auftreten von Fehlermeldungen, welche innerhalb des Adressierungsbereich des Entwicklers liegen (Fehler in der “Java Virtual Machine” als Gegenbeispiel), behandeln können und seine Tätigkeiten bestmöglich fortsetzen.

Besonders beim Einlesen von GPS-Dateien darf es aufgrund von zum Beispiel einer fehlerhaften Datei zu keinem Absturz kommen.

Bei einem Absturz der Applikation soll nach erneutem Start und erneutem Einlesen der GPS-Dateien die Nutzung wieder fortgefahren werden können.

# Architektur

Zur Umsetzung dieser Applikation wird Java Version 11 verwendet. Zur Realisierung der graphischen Benutzeroberfläche wird das Framework “JavaFX” in der Version 18 verwendet. Die Applikation wird mittels einem “MVC-Pattern” realisiert, da “JavaFX” fundamentall dafür strukturiert und ebenso für die Aufgabenstellung passend ist. Das “MVC-Pattern” besteht aus einem Modell, einem oder mehreren “Views” und der Anzahl der “Views” entsprechenden “Controllers”.

Das Modell enthält die für die Applikation notwendigen Daten und die entsprechende Geschäftslogik. Das sind zum Beispiel Methoden um Elemente hinzuzufügen oder sie auch wieder zu entfernen. In unserem Projekt verwenden wir das Modell um alle “GPS-Tracks” und weitere notwendige Informationen zu verwalten. Zum Interagieren mit dem Modell und dem Liefern von Informationen an die “Views” wird ein “Controller” verwendet. Dieser definiert welche Aktion beim Drücken von zum Beispiel einem Knopf auf der “View” ausgelöst werden soll. Ebenso bestimmt der “Controller”, welche Daten aus dem Modell auf der “View” zu sehen sind. Unser Projekt besteht aus einem “Controller”. Dieser “Controller” übernimmt die Steuerung der Listenansichten, des Diagramms und der allgemeinen Interaktion mit dem Nutzer. Die “View” übernimmt die Darstellung der Interaktionsmöglichkeiten und der Daten. Sie kommuniziert dazu mit dem “Controller” und interagiert ebenso mit dem Modell. In unserem Projekt interagiert die “View” mit dem zuvor angesprochenen Modell um die Daten für die Darstellung der Listenansichten und dem Balkendiagramm zu beschaffen.

Durch die Auswahl von “MVC-Pattern” sind die Komponenten der Applikation eine Main-Klasse zum Start der Applikation, ein Modell, mehrere “Views” und entsprechende “Controllers”.

Die graphische Benutzeroberfläche wird, wie schon angesprochen, mittels “JavaFX” und “FXML” umgesetzt. Der Vorteil durch den Einsatz von “FXML” ist die Trennung zwischen der graphischen Komponente (View) und der Logik, welche mit der Grafik interagiert (Controller). Dies ermöglicht es die graphische Benutzeroberfläche anzupassen ohne Änderungen an der Logik vornehmen zu müssen.

Die entsprechenden Komponenten der Applikation sind aus dem UML-Klassendiagramm zu entnehmen.

# Kernfunktionen (Must-haves)

Folgende Funktionen sind basierend auf den Anforderungen als “must-haves” kategorisiert worden. Die Applikation soll daher folgende Funktionen spätestens mit dem finalen Release umgesetzt haben.

## Start

Die Applikation ist auf Computern, welche Java Version 15 unterstützen, ausführbar. Beim Start der Applikation sollen alle zuvor geladenen „GPS-Tracks“ einzeln in einer Listenansicht wiederhergestellt werden. Dazu muss die Applikation beim Start die zuvor schon geladenen “GPS-Tracks” einlesen. Wenn keine „GPS-Tracks“ vorhanden sind, ist die Liste leer. Das Diagramm in der Applikation soll nach dem Start die zurückgelegte Distanz pro Monat anzeigen. Die detaillierte Listenansicht ist leer nach dem Start der Applikation.

## Einlesen und Persistieren

Die Applikation unterstützt das Einlesen und Persistieren von neuen „GPS-Tracks“, welche im Format „.tcx“ (Schema Version 2) und „.gpx“ (Schema Version 1.1) vorliegen. Diese Dateien sollen dabei vom Programm ausgelesen und die relevanten Informationen innerhalb der Applikation gespeichert werden. Relevante Informationen sind dabei der Name, das Datum der Datenaufzeichnung, der Startzeitpunkt, die Kategorie und die einzelnen aufgezeichneten GPS-Datenpunkte des “GPS-Tracks”. Im Rahmen des Einlesens kann der Nutzer die Kategorie, das Datum und den Startzeitpunkt anpassen. Der Name entspricht dem zugehörigen Dateinamen. Die einzelnen GPS-Datenpunkte werden jeweils mit der zurückgelegten Distanz und der benötigten Zeit relativ von der Startzeit oder falls vorhanden dem vorangegangen Datenpunkt abgespeichert. Falls das Einlesen und/oder Abspeichern der „GPS-Tracks“ nicht möglich ist, soll eine Fehlermeldung, die den User über den Fehler informiert, präsentiert werden.

Das Einlesen der “GPS-Tracks” kann dabei entweder einzeln oder per Ordner geschehen.

Löschen

Die Applikation soll es Nutzern ermöglichen “GPS-Tracks” zu entfernen. Diese entfernten “GPS-Tracks” dürfen bei einem erneuten Start der Applikation nicht erneut gezeigt werden.

## Listenansicht

Die “GPS-Tracks” sollen jeweils einzeln mittels einer Liste angezeigt werden. Anzuzeigende Informationen sind der Name, das Datum, die Startzeit, die zurückgelegte Distanz, die benötigte Zeit, die Pace und die Geschwindigkeit.

## Detaillierte Listenansicht

Durch die Auswahl eines einzelnen Tracks soll eine detaillierte Listenansicht dieses Tracks angezeigt werden. Die detaillierte Listenansicht zeigt dabei folgende Informationen pro aufgezeichneten GPS-Datenpunkt dieses “GPS-Tracks” an: Reihenfolgenummer, die zurückgelegte Distanz, die benötigte Zeit, die Pace und die Geschwindigkeit.

## Gruppieren

Der Benutzer kann die “GPS-Tracks” nach Zeiträumen gruppieren. Diese Gruppierungen sind jeweils auf Basis von Wochen, Monaten oder Jahren möglich. Bei der Gruppierung sollen in der Listenansicht die Anzahl der “GPS-Tracks” als Spalte angezeigt werden, die Kombination der Distanz und Zeit aller “GPS-Tracks” in der Gruppierung als separate Spalten, die durchschnittliche Pace und die durchschnittlichen Geschwindigkeit der Gruppe. Ebenso sollen die Akkumulation der Höhenunterschiede angezeigt werden.

## Sortieren

Im Rahmen der Listenansicht und detaillierten Listenansicht sollen Sortierungen nach den Werten pro Spalte absteigend oder aufsteigend möglich sein.

Diagrammansicht

Neben der Listenansicht soll ebenso eine Diagrammansicht angezeigt werden. Diese Diagrammansicht ist dabei ein Balkendiagramm, welches je nach der aktuellen Selektion und dem gewählten Vergleichswert (zurückgelegte Distanz, benötigte Zeit, Pace oder Geschwindigkeit) an die gegebenen Werte anpasst. Die Selektion kann dabei keine Selektion sein, wobei das Diagramm in diesem Fall die entsprechenden Werte der Listenansicht widerspiegelt. Bei Selektion eines einzelnen “GPS-Tracks” werden die Werte laut Segmentierung gespiegelt. Bei Selektion einer Gruppierung sollen die Werte der “GPS-Tracks” in dieser Gruppierung verwendet werden.

Filtern

Die Applikation ermöglicht es die Tracks nach ihrem Aktivitätstypen zu filtern. Die möglichen Aktivitäten sind dabei Biking, Driving, Flying, Hiking, Running und Skiing. Diese Filterung gilt dabei sowohl übergreifend über die Listenansicht als auch der Diagrammansicht.

Schließen

Die Applikation ermöglicht es die die Applikation zu schließen.

Zusatzfunktion (nice-to-haves)

Vergleich

Ein Vergleich der “GPS-Tracks” wird durch die Applikation ebenfalls unterstützt. Der Vergleichen von “GPS-Tracks” ist dabei über Jahre möglich. Die Anzahl der zu vergleichenden Jahre ist dabei frei wählbar.

Trackpoint:

* Datum (fix bei tcx & optional bei GPX)
* Zeitpunkte (fix bei tcx & optional bei GPX)
* zurückgelegte Distanz (optional bei tcx & optional bei GPX)
* benötigte Zeit (fix bei tcx (über Zeitpunkte) & optional bei GPX)
* pace (optional bei tcx & optional bei GPX)
* speed (optional bei tcx & optional bei GPX)
* bpm (optional bei tcx & bei GPX leer)
* max bpm (optional bei tcx & bei GPX leer)
* elevation (optional bei tcx & optional bei GPX)

Track:

* Trackname (fix bei tcx & optional bei GPX)
* Datum (fix bei tcx (aus Lap) & optional bei GPX)
* zurückgelegte Distanz (fix bei tcx (über lap) & optional bei GPX)
* benötigte Zeit (fix bei tcx (über lap) & optional bei GPX)
* pace (fix bei tcx (über oben) & optional bei GPX)
* speed (fix bei tcx (über oben) & optional bei GPX)
* bpm (optional bei tcx & bei GPX leer)
* max bpm (optional bei tcx & bei GPX leer)
* elevation (optional bei tcx & optional bei GPX)

GPS-Tracks

Diese Tracks bestehen aus

* Name
* Datum
* Startzeit
* Entfernung
* Dauer
* Geschwindigkeit
* Herzschlag pro Minute
* Maximaler Herzschlag
* Höhe

Grafik

Die Applikation soll die Möglichkeit bieten Tracks mithilfe von Grafiken miteinander zu vergleichen.

Zum Vergleich kann ausgewählt werden:

* Distanz
* Dauer
* Erhebungen
* Geschwindigkeit
* Herzschlag

# Komponente

Working with inconsistent xml files

<https://gist.github.com/bennadel/9759130#file-code-3-cfm-L62>