

Pflichtenheft

Tour-Management-System

Tour66

Bearbeiter:



Betreuer:

Ruth Breu

Datum:

08.04.2008 (Version 1.0)

Dokumentenhistorie

Version	Datum	Bearbeiter	Kommentar	Status
0.5	16.03.2008	alle	Erste Version (im Projekt-wiki)	WIP
0.7	19.03.2008	MB	Konsolidiert, geringe Unstimmigkeiten, einige Details müssen noch geklärt werden	WIP
0.8	27.03.2008	MB	Ergänzung Zielgruppe, Zustandsmodelle	WIP
0.9	01.04.2008	MB	Umsetzungs-Details, Zeitplan	WIP
1.0	08.04.2008	MB	Finalisierung	STABLE

Inhaltsverzeichnis

1	ANFORDERUNGEN UND ZIELE	1
1.1	Überblick	1
1.2	Details Muss-Anforderungen	1
1.3	Details Kann-Anforderungen	2
1.4	Zielgruppe und Anforderungen an die Benutzerführung	3
2	USE CASES.....	4
2.1	Akteure	4
2.2	Use Case Diagramm	5
2.3	Use Case „Alle Touren anzeigen“	6
2.4	Use Case „Touren suchen“	7
2.5	Use Case „Tour betrachten“	8
2.6	Use Case „Ältere Versionen betrachten“	9
2.7	Use Case „Tour drucken“	10
2.8	Use Case „Registrieren“	11
2.9	Use Case „Anmelden“	12
2.10	Use Case „Abmelden“	13
2.11	Use Case „Tour erstellen“	14
2.12	Use Case „Tour bearbeiten“	15
2.13	Use Case „Tourbeschreibung erstellen/aktualisieren“	17
2.14	Use Case „Bilder hochladen/verwalten“	19
2.15	Use Case „GPS Track laden“	20
2.16	Use Case „Daten löschen“	22
2.17	Use Case „Touren verwalten“	23
2.18	Use Case „Benutzer verwalten“	24
2.19	Use Case „Kommentare verwalten“	25
3	KLASSENMODELL	26
3.1	Klassendiagramm	26
3.2	Zustandsmodelle	27
4	UMSETZUNGS-DETAILS	30
4.1	Zielplattform	30
4.2	Datenhaltung	30

4.3 Oberfläche	30
4.4 Systemarchitektur (grob)	30

Abbildungsverzeichnis

Abbildung 1: Use Case Diagramm	5
Abbildung 2: Fachliches Klassendiagramm	26
Abbildung 3: Zustandsdiagramm der Klasse Tour.....	27
Abbildung 4: Zustandsdiagramm der Klasse TourDescription	28
Abbildung 5: Zustandsdiagramm der Klasse Comment	28
Abbildung 6: Zustandsdiagramm der Klasse User.....	29

1 Anforderungen und Ziele

1.1 Überblick

Es soll ein System zur Verwaltung von Touren (z.B. Skitouren, Mountainbiketouren, Bergtouren) entwickelt werden. Die Datenhaltung soll in einer zentralen Datenbank realisiert werden, die Benutzer greifen über einen Java-Client (keine Web-Applikation!) auf das System zu.

Folgende Funktionen müssen durch das System abgedeckt werden:

- Touren erfassen (inklusive Versionierung)
- Tourliste anzeigen (inklusive suchen / filtern /sortieren)
- Tourdetails anzeigen
- Tour drucken
- GPS Track in Google Earth erstellen
- Touren kommentieren

Eine der folgenden Kann-Anforderungen muss erfüllt werden

- Pinboard
- Tourenbuch

Die Anforderungen werden nachfolgend detailliert.

1.2 Details Muss-Anforderungen

1.2.1 Touren erfassen

Das System soll das Erfassen von Touren durch registrierte Benutzer erlauben. Zu jeder Tour soll ein Tourname angegeben werden. Darüber hinaus sollen Informationen wie Charakter der Tour, Anreise, Ausgangspunkt, Aufstieg/Auffahrt und Abfahrt/Abstieg vermerkt werden. Falls vorhanden sollen zu einer Tour Bilder eingegeben werden können. Weiters besitzt jede Tour Tourdaten. Bei Ski- und Bergtouren sind das Zeit, Höhendifferenz, Exposition, Schwierigkeit, Starthöhe, Endhöhe, beste Tourenzeit und Gebirge. Bei Mountainbiketouren sind das bspw. Routennummer, Fahrzeit, Höhendifferenz, Abfahrtsdifferenz, Schwierigkeit, Länge, Wegbeschaffenheit, Starthöhe, Endhöhe, beste Tourenzeit und Gebirge. Schlussendlich sollen Touren auch einen Link auf einen in Google Earth erstellten GPS Track besitzen können. Optional sollen zu Touren Karten und ein Höhenprofil angegeben werden können.

1.2.2 Touren versionieren

Da Tourinformationen von allen registrierten Benutzern in kollaborativer Art und Weise erfasst werden können, sollen die Tourdaten versioniert werden. Jede Version soll verzeichnen, wann und durch welchen Benutzer eine Änderung vorgenommen wurde.

1.2.3 Tourliste anzeigen

Das System soll das Anzeigen einer Gesamtliste mit Touren erlauben. Zu jeder Tour soll der Name, die Höhe, die Zeit, die Höhendifferenz, die Schwierigkeit, das Gebirge und das Gebiet angegeben werden. Außerdem soll vermerkt sein, ob es für die entsprechende Tour GPS Daten gibt. Für Skitouren und Bergtouren soll darüber hinaus die Exposition angezeigt werden. Für Radtouren falls vorhanden die Routennummer.

1.2.4 Touren suchen / filtern

Die Software soll die Suche von Touren unterstützen. Der Benutzer soll dabei angeben können, ob er nach Skitouren, Mountainbiketouren oder Bergtouren suchen möchte. Die Suche kann entweder über die Eingabe eines Suchbegriffs erfolgen oder durch die Eingabe von Filterkriterien wie beispielsweise Bundesland oder Gebirge.

1.2.5 Touren sortieren

Touren sollen nach verschiedenen Kriterien sortiert werden können (Dauer, Schwierigkeit, Höhendifferenz,...).

1.2.6 Tourdetails anzeigen

Der Benutzer durch Auswählen einer Tour die genauen Tourdetails anschauen können.

1.2.7 Tour drucken

Das Drucken der Tourdetails soll unterstützt werden. Wenn zu den Touren Landkarten bzw. Höhenprofile vorhanden sind, sollen diese optional mit-eingebunden werden können.

1.2.8 GPS Track in Google Earth erstellen

GPS Tracks zu Touren sollen in Google Earth erstellt werden. Dabei sollen die Touren als Pfad (nicht als einzelne Punkte) dargestellt werden. Start- und Endpunkte sollen separat gekennzeichnet werden. Bilder können entlang der Strecke eingebunden werden.

1.2.9 Touren kommentieren

Registrierte Benutzer sollen die Möglichkeit haben, Kommentare zu einzelnen Touren zu verfassen. Kommentare enthalten den Namen der Tour, das Datum, den Nickname des Benutzers, eine kurze Beschreibung über die Verhältnisse und optional Bilder.

1.3 Details Kann-Anforderungen

1.3.1 Pinboard

Über das Pinboard können Partner für eine Tour gefunden werden. Dabei werden eine Kurzbeschreibung, die Art der Tour, die Gegend und das Da-

tum, Nickname und Emailadresse angegeben. Interessierte können nach erfolgter Registrierung per Email Kontakt aufnehmen.

1.3.2 Tourenbuch

Registrierte Benutzer können sich einen Katalog mit allen Touren anlegen, die sie in einem bestimmten Zeitraum gemacht haben. Im Katalog sollen nicht nur die gemacht Touren gelistet werden, sondern auch Auswertungen wie beispielsweise die gesamten bewältigten Höhenmeter.

1.4 Zielgruppe und Anforderungen an die Benutzerführung

Die potentiellen Benutzer des Tour-Management-Systems sind Outdoor-Aktivisten mit zumindest grundlegenden Kenntnissen in der Bedienung von Computerprogrammen. Spezial-Kenntnisse können von dieser Zielgruppe nicht vorausgesetzt werden.

Prinzipiell können zwei unterschiedliche Benutzergruppen unterschieden werden.

1.4.1 Interested Viewers

Diese Benutzer zeichnen sich primär dadurch aus, dass sie die eingetragenen Touren durchsuchen und Detailinformationen zu einzelnen Touren einsehen. Vereinzelt werden Touren kommentiert, das Anlegen oder Bearbeiten von Touren wird hingegen seltener durchgeführt.

Die Tourensuche sowie die einzelnen Funktionen zur Anzeige von Touren und zugehörigen Daten müssen bereits bei der ersten Benutzung des Programms intuitiv bedienbar sein.

1.4.2 Enthusiastic Editors

Diese Nutzergruppe ist aktiv an der Erstellung und Wartung der Tourinformationen beteiligt. Ihr primäres Interesse liegt in der Verbesserung der gespeicherten Touren und persönliche Profilierung in der Community. Diese Gruppe ist geübt im Umgang mit Google Earth und/oder GPS-Software.

Die Funktionen zur Erstellung und Bearbeitung von Touren müssen den gewohnten Mechanismen entsprechen und sollten mit Unterstützung einer Hilfsfunktion beim erstmaligen Benutzen der Software möglich sein.

2 Use Cases

Kapitel 2.2 beschreibt die Akteure, Kapitel 2.2 zeigt das vollständige Use-Case-Diagramm, die Kapitel 2.3 bis 2.19 beschreiben die einzelnen Use Cases auf Basis der Anforderungen aus Kapitel 0.

2.1 Akteure

2.1.1 Gast

Die Applikation ist unter diversen Einschränkungen auch für Personen benutzbar welche sich nicht ausweisen (Registrierung/Anmeldung). Diese Personen fallen unter die Akteurgruppe "Gast".

Dieser Akteur entspricht im wesentlichen der Zielgruppe „Interested Viewers“. Ein Gast kann alle (Touren-)Informationen anzeigen, die Bearbeitung oder das Kommentieren von Touren ist nicht möglich.

2.1.2 Benutzer

Registrierte Personen (Use Case „Registrieren“) können sich im System anmelden (Use Case „Anmelden“) und zählen ab diesem Zeitpunkt zur Akteurgruppe "Benutzer". Achtung: Ein registrierter User kann auch als Gast agieren (wenn er nicht angemeldet ist).

Dieser Akteur entspricht im wesentlichen der Zielgruppe „Enthusiastic Editors“. Neben allen Funktionen die dem Akteur Gast zur Verfügung stehen, kann ein Benutzer zusätzlich bestehende Informationen (Tourbeschreibung, Bilder, GPS-Track) bearbeiten und Touren kommentieren.

2.1.3 Administrator

Zu dieser Akteurgruppe gehören Benutzer mit besonderen Rechten. Dieser (wahrscheinlich sehr eingeschränkte) Personenkreis wird sämtliche Administrationsarbeiten erledigen (siehe Use Cases).

2.2 Use Case Diagramm

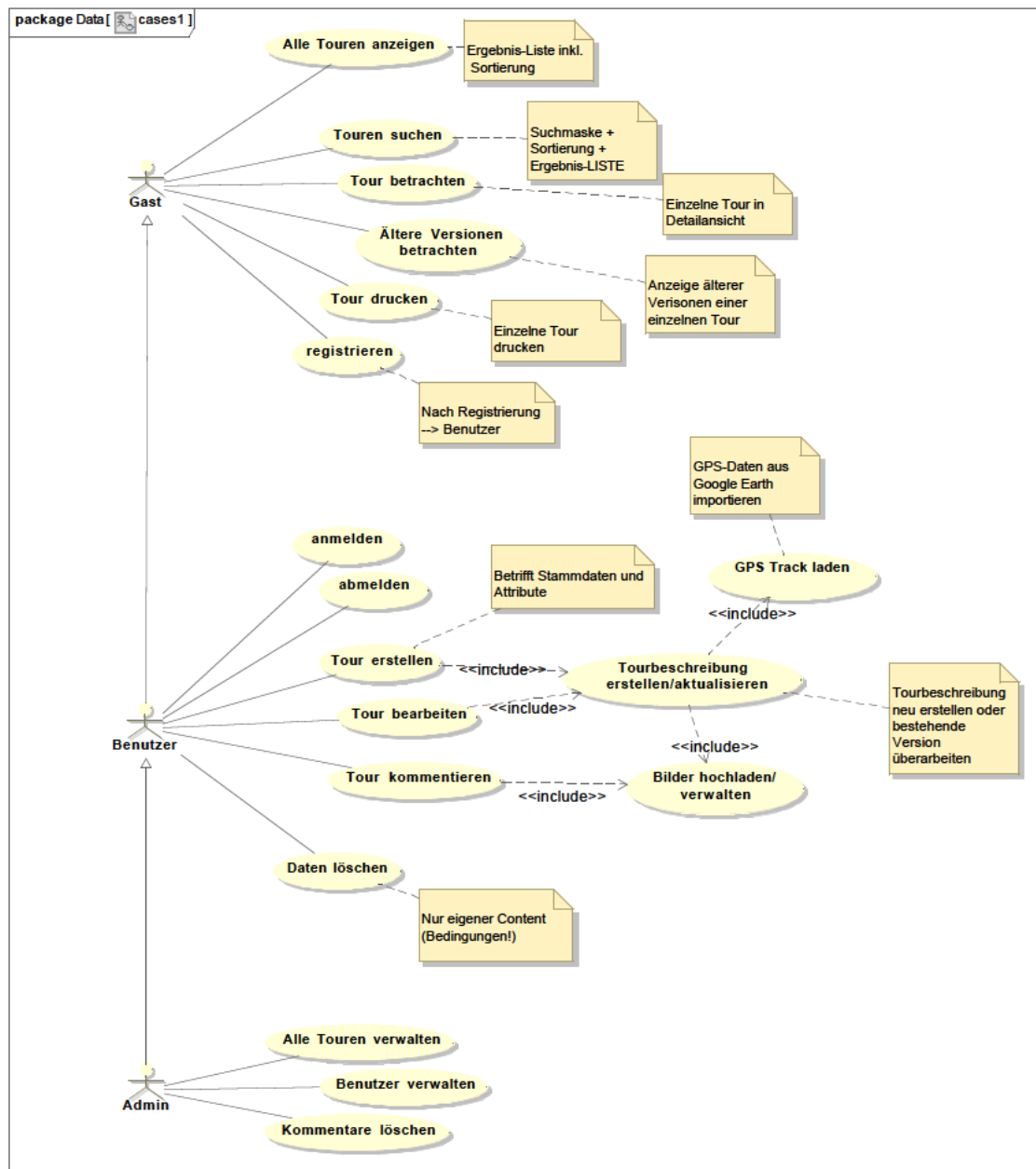


Abbildung 1: Use Case Diagramm

2.3 Use Case „Alle Touren anzeigen“

Kurze Beschreibung

Der Gast lässt sich eine Liste von Touren anzeigen, deren Sortierung optional angepasst werden kann.

Auslöser

Der Gast wechselt in den Touren-Übersichts-Bereich

Akteur(e)

Gast

Vorbedingung(en)

System bereit

Schritte

1. Das System zeigt dem Gast eine Auflistung der Touren, sortiert nach den Standard-Einstellungen, an.
2. Falls nicht alle Touren zusammen auf der Oberfläche angezeigt werden könne, so wird dem Gast eine Navigation zur Verfügung gestellt, mit welcher er die restlichen Touren seitenweise ansehen kann.

Alternative Schritte

1. a. Der Gast ändert die Sortier-Einstellung und das System aktualisiert die angezeigten Ergebnisse.
b. Das System enthält keine Touren und zeigt dem Gast dies mit einer Hinweis-Meldung an.

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

Tour, TourType, TourDescription, TourStatus, TourAttribs

2.4 Use Case „Touren suchen“

Kurze Beschreibung

Der Gast gibt Suchkriterien der gewünschten Touren ein und das System liefert die Ergebnisse zurück.

Auslöser

Der Gast wechselt in den Suchbereich.

Akteur(e)

Gast

Vorbedingung(en)

System bereit

Schritte

1. Eine Suchmaske wird angezeigt, in welcher der Gast seine Suchkriterien eingibt und anschließend die Suche bestätigt.
2. Das System liefert das Resultats der Suchanfrage, sortiert nach den Standard-Einstellungen, als Liste zurück.
3. Falls nicht alle Touren zusammen auf der Oberfläche angezeigt werden könne, so wird dem Gast eine Navigation zur Verfügung gestellt, mit welcher er die restlichen Touren seitenweise ansehen kann.

Alternative Schritte

2. a. Der Gast ändert die Sortier-Einstellung und das System aktualisiert die angezeigten Ergebnisse.
b. Das System konnte keine entsprechenden Touren finden und zeigt dem Gast dies mit einer Hinweis-Meldung an. Der Gast kann die Suchkritieren korrigieren und die Suche wiederholen.

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

Tour, TourType, TourDescription, TourStatus, TourAttribs

2.5 Use Case „Tour betrachten“

Kurze Beschreibung

Dem Gast werden Detailinformationen über eine ausgewählte Tour angezeigt.

Auslöser

Der Gast wechselt in die Detail-Ansicht einer Tour.

Akteur(e)

Gast

Vorbedingung(en)

System bereit

Schritte

1. Der Gast lässt sich eine Auflistung von Touren anzeigen.
2. Eine Tour wird ausgewählt und in der Detail-Ansicht geöffnet.
3. Das System gibt dem Gast eine detaillierte Beschreibung der Tour aus, welche folgendes enthält:
 - Titel der Tour
 - Typ der Tour
 - Region
 - Strecke
 - Tour-Eigenschaften bezogen auf den Tour-Typ
 - aktuelle Beschreibung der Tour
 - Status der Beschreibung der Tour
 - Kommentare zur Beschreibung der Tour
 - (optional) GPS-Track Export für GoogleEarth
 - (optional) Bilder zur Beschreibung

Alternative Schritte

-

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

Tour, TourType, TourDescription, TourStatus, TourAttribs, Comment, Content, Image

2.6 Use Case „Ältere Versionen betrachten“

Kurze Beschreibung

Der Gast kann jede Version einer Tour-Detail-Beschreibung ansehen.

Auslöser

Der Gast lässt sich die Navigation der Versionen einer Touren-Detail-Beschreibung anzeigen.

Akteur(e)

Gast

Vorbedingung(en)

Gast befindet sich in der Detailansicht einer Tour

Schritte

1. Dem Gast wird eine Navigation angezeigt, mit der er aus allen Versionen der Detail-Beschreibung eine Version wählen kann.
2. Das System gibt dem Gast eine detaillierte Beschreibung der Tour aus, welche folgendes enthält:
 - Titel der Tour
 - Typ der Tour
 - Region
 - Strecke
 - Tour-Eigenschaften bezogen auf den Tour-Typ
 - aktuelle Beschreibung der Tour
 - Status der Beschreibung der Tour
 - Kommentare zur Beschreibung der Tour
 - (optional) GPS-Track Export für GoogleEarth
 - (optional) Bilder zur Beschreibung

Alternative Schritte

2. a. Der Gast wählt keine Version aus, sondern beendet die Versionen-Navigation.

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

Tour, TourType, TourDescription, TourStatus, TourAttribs, Comment, Content, Image

2.7 Use Case „Tour drucken“

Kurze Beschreibung

Der Gast kann die Detail-Ansicht einer Tour drucken.

Auslöser

Der Gast betätigt die Druckfunktion.

Akteur(e)

Gast

Vorbedingung(en)

Gast befindet sich in der Detailansicht einer Tour

Schritte

1. Dem Gast wird eine Druckvorschau angezeigt und Schaltflächen zum Bestätigen oder Abbrechen des Druckvorgangs.
2. Mit dem Bestätigen des Druckvorgangs wird dem Gast ein Druckdialog angezeigt, indem er Druckeinstellungen festlegen und den Druckvorgang starten kann.
3. Nach dem Starten des Druckvorgangs werden wieder sämtliche Druck-Fenster und Dialoge geschlossen und der Gast befindet sich wieder in der Detail-Ansicht der Tour.
4. Das Drucksystem bearbeitet das neue Dokument.

Alternative Schritte

2. a. Wenn der Druckvorgang abgebrochen wird, wird die Druckvorschau geschlossen und der Gast befindet sich wieder in der Detail-Ansicht der Tour.
3. a. Wenn der Druckvorgang abgebrochen wird, wird der Druckdialog geschlossen und der Gast befindet sich wieder in der Druckvorschau.
4. a. Das Drucksystem kann das neue Dokument nicht bearbeiten und gibt eine entsprechende Fehlermeldung aus.

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

Tour, TourType, TourDescription, TourStatus, TourAttribs, Comment, Content, Image

2.8 Use Case „Registrieren“**Kurze Beschreibung**

Um das System mit eigenen Tourempfehlungen mitgestalten zu können ist eine Registrierung nötig. Der Interessent (momentan noch Gast) gibt seine Daten an (siehe unten) und bestätigt die Geschäftsbedingungen. Das System speichert diese und erzeugt einen Account.

Auslöser

Gast klickt auf "Registrieren"

Akteur(e)

Gast

Vorbedingung(en)

System bereit, Hinzufügen von Accounts ist erlaubt (kann von Admin blockiert werden)

Schritte

1. Gast gibt alle nötigen Daten ein (Vorname, Nachname, Adresse, Telefon, Geburtsdatum, eMail-Adresse, Nickname)
2. Gast klickt auf den Button "registrieren"
3. System validiert die Eingabe
4. System generiert ein Zufallspasswort und speichert alle Daten
5. System versendet ein Mail an den Gast (Willkommenstext, Passwort) und informiert ihn darüber
6. Gast kann sich nun mit seinem Nickname und dem generierten Passwort anmelden (Use Case „Anmelden“)

Alternative Schritte

1. a. i. Werden Eingabefehler erkannt (Mail-Adresse ist ungültig oder leer, Vor-, Nachname ist leer, ...) dann werden die entsprechenden Felder markiert, eine Fehlermeldung ausgegeben und die Korrektur des Akteurs abgewartet
- ii. Wird ein Nickname gewählt welcher bereits vergeben ist (NICHT CASE-SENSITIVE!), dann erscheint ebenfalls eine Fehlermeldung bzw. ein Hinweis

Nachbedingung(en)

Der Account wird mit dem Status "registered" gespeichert. Meldet sich der neue Benutzer das erste Mal an wird der Status auf "active" abgeändert -

andernfalls wird der Account nach einer bestimmten Zeit automatisch gelöscht.

Inkludierte Use Cases

-

Involvierte Klassen

User(c)

2.9 Use Case „Anmelden“

Kurze Beschreibung

Der User muss sich bei jedem Besuch ausweisen um erweiterte Rechte (vs. Gast) zu erhalten (evtl. Mechanismus wie z.B. cookies einpflegen?). Dazu muss er seine persönlichen Login-Daten eingeben.

Auslöser

User klickt (im Gast-Modus) auf den (immer sichtbaren) "Login/Logout"-Button. Alternative: bei Login-Mechanismus ist keine Aktion nötig (User wird vom System erkannt und automatisch mit erweiterten Rechten ausgestattet).

Akteur(e)

Benutzer

Vorbedingung(en)

User besitzt einen aktiven Account (Use Case „Registrieren“), System bereit

Schritte

1. User gibt seine Login-Daten ein [2. User wählt aus ob er vom aktuellen Rechner aus immer eingeloggt bleiben möchte]
2. User klickt auf den Button "Login"
3. System überprüft die Daten (beim ersten Login des Users: System ändert den Account-Status von „registered“ auf „active“)
4. System erteilt dem User erweiterte Rechte und leitet ihn auf die Startseite (oder auf die zuletzt gesehene bzw. angeforderte Seite) weiter

Alternative Schritte

4. a. Daten stimmen nicht überein: User bekommt entsprechende Meldung (kein Hinweis darauf welche Eingabe falsch war)

Nachbedingung(en)

User ist als Benutzer angemeldet und besitzt erweiterte Rechte, Loginvorgang ist im System gespeichert (Last Login)

Inkludierte Use Cases

-

Involvierte Klassen

User

2.10 Use Case „Abmelden“**Kurze Beschreibung**

"Abmelden" beendet eine User-Session. Ein angemeldeter User (Usecase: Anmelden) hat nach diesem Vorgang nur noch Gast-Rechte im System.

Auslöser

User klickt auf "Logout". Alternative: User schließt die Applikation

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, User angemeldet

Schritte

1. angemeldeter User klickt auf einen (im System immer sichtbaren) "Login/Logout"-Button
2. System nimmt die nötigen Änderungen vor um die Session zu beenden

Alternative Schritte

1. a. angemeldeter User schließt die Applikation

Nachbedingung(en)

User ist abgemeldet (nur mehr Gast-Rechte), Sitzungsdaten im System gespeichert (z.B. Dauer der Sitzung, ...)

Inkludierte Use Cases

-

Involvierte Klassen

User

2.11 Use Case „Tour erstellen“

Kurze Beschreibung

Anlegen einer neuen Tour (Stammdaten und erste Tour-Beschreibung).

Auslöser

Benutzer klickt auf "Neue Tour anlegen"

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, Benutzer angemeldet

Schritte

1. Der Benutzer gibt eine Bezeichnung ein und wählt den Typ (z.B. Wanderung, Skitour, etc.)
2. Das System aktualisiert (erst nach Eingabe des Tour-Typs) die Eingabemaske und erlaubt die Eingabe der Tour-Typ-abhängigen Attribute
3. Der Benutzer gibt die weiteren Stammdaten (Region, Charakter, Anreise, Startpunkt, Endpunkt) ein
4. Der Benutzer trägt die Tour-Attribute (siehe 2.) ein
5. Das System öffnet die Maske zur Eingabe der Tourbeschreibung -> Use Case "Tourbeschreibung erstellen/aktualisieren"
6. Der Benutzer speichert die neue Tour mit Klick auf „Speichern“

Alternative Schritte

1. a. Der Benutzer gibt einen Bezeichner ein, der bereits vergeben ist. Das System informiert den Benutzer und verlangt die Eingabe einer neuen Bezeichnung
2. a. Der Benutzer ändert nachträglich den Tour-Typ. Das System aktualisiert die Eingabemaske und zeigt die für den neuen Tour-Typ relevanten Attribute an. Werte werden - soweit möglich - übernommen
6. a. Der Benutzer bricht die Eingabe ab (Button „Abbrechen“ oder Fenster schließen). Das System fragt mit einem Dialog nach, ob die eingegebenen Daten wirklich verworfen werden sollen

Nachbedingung(en)

Die eingegebene Tour wird im System gespeichert

Inkludierte Use Cases

Tourbeschreibung erstellen/aktualisieren, GPS Track laden

Involvierte Klassen

User, Tour(c), TourType, TourAttribs(c), TourDescription(c), Content(c)

2.12 Use Case „Tour bearbeiten“

Kurze Beschreibung

Ein Benutzer bearbeitet (ändert) eine bestehende Tour (Stammdaten) und/oder aktualisiert die Tourbeschreibung.

Auslöser

Benutzer klickt auf "Tour bearbeiten"

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, Benutzer angemeldet

Schritte

1. Der Benutzer führt einen oder mehrere der Schritte 2 bis 6 in beliebiger Reihenfolge aus
2. (optional) Benutzer ändert Stammdaten (Region, Charakter, Anreise, Startpunkt, Endpunkt), Bezeichnung und Tour-Typ können nicht verändert werden
3. (optional) Benutzer lädt aktualisiertes Kartenmaterial über einen Upload-Dialog hoch
4. (optional) Benutzer aktualisiert das Höhenprofil über einen Upload-Dialog
5. (optional) Benutzer aktualisiert den GPS-Track -> Use Case "GPS Track hochladen"
6. (optional) Benutzer aktualisiert die Tourbeschreibung (ggf. inkl. Bildmaterial) -> Use Case "Tourbeschreibung erstellen/aktualisieren"
7. Abschließend klickt der Benutzer auf "Änderungen speichern"

Alternative Schritte

- 2. a. i. Nach Änderung von Start- oder Endpunkt wird der Benutzer in einem Dialog gefragt, ob er die bestehende Tour ändern will, oder eine neue Tour mit den gerade eingegebenen Start/End-Punkt anlegen will
- ii. Folgende Optionen können ausgewählt werden: Bestehende Tour ändern, Neue Tour (leer), Neue Tour (Daten übernehmen)
- iii. Wenn der Benutzer "Neue Tour (leer)" oder "Neue Tour (Daten übernehmen)" auswählt wird die bestehende Tour nicht verändert und die Eingabemaske für eine neue Tour angezeigt (ggf. mit den übernommenen Daten) -> Use Case "Tour anlegen"
- iv. Wenn der Benutzer "Tour ändern" auswählt wird die bestehende Tour aktualisiert
- 7. a. i. Der Benutzer klickt auf "Abbrechen"
- ii. Das System fragt über einen Dialog nach, ob die eingetragenen Änderungen wirklich verworfen werden sollen.
- iii. Wenn der Benutzer auf "Ja" klickt werden die Änderungen verworfen

Nachbedingung(en)

Die veränderten Daten sind in der zentralen Datenbank aktualisiert

Inkludierte Use Cases

Tourbeschreibung erstellen/aktualisieren, GPS Track laden, Bilder laden/verwalten

Involvierte Klassen

User, Tour(m), TourType, TourAttribs(m), TourDescription(c), Content(c)

2.13 Use Case „Tourbeschreibung erstellen/aktualisieren“

Kurze Beschreibung

Der Benutzer schreibt eine detaillierte Beschreibung der Tour (ggf. auf Basis der bestehenden Beschreibung).

Auslöser

Benutzer klickt auf "Tour beschreiben"

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, Benutzer angemeldet, Stammdaten der Tour erfasst

Schritte

1. Der Benutzer gibt in einem Textfeld eine detaillierte Beschreibung der Tour an. Dabei stehen ihm unterschiedliche Formatierungsanweisungen (Fett, Kursiv, Unterstrichen, Aufzählung und unterschiedliche Schriftgrößen) zur Verfügung. Die Anzahl verwendeter Zeichen ist nicht beschränkt. Hinweis: Wenn bereits eine Beschreibung für die ausgewählte Tour gespeichert ist, kann diese erweitert werden
2. (Optional) Benutzer lädt Fotos zur besseren Beschreibung hoch -> Use Case "Bilder hochladen/verwalten"
3. (Optional) Benutzer lädt Kartenmaterial in Form eines oder mehrerer Bilder hoch -> Use Case "Bilder hochladen/verwalten"
4. (Optional) Benutzer lädt GPS-Track aus Google Earth hoch -> Use Case "GPS Track laden"
5. (Optional) Benutzer kommentiert die aktuelle Tourbeschreibung mit Text und Bild(er) -> Use Case "Tour kommentieren"
6. Der Benutzer speichert die aktualisierte Tourbeschreibung mit "Tourbeschreibung speichern"

Alternative Schritte

1. a. i. Der Benutzer gibt keine Beschreibung für eine neue Tour ein aber ein oder mehrere der optionalen Daten
 - ii. Das System informiert den Benutzer, dass ein Beschreibungstext eingetragen werden muss
 - iii. Der Benutzer kann entweder eine Beschreibung angeben (Schritt 1) oder die optionalen Daten werden nicht gespeichert
 - b. i. Der Benutzer löscht die bestehende Beschreibung
 - ii. Das System informiert den Benutzer per Dialog das der bestehende Beschreibungstext nicht vollständig gelöscht werden darf
 - iii. Der Benutzer kann die letzte Version des Textes wieder einfügen oder eine neue Beschreibung eintragen
6. a. i. Der Benutzer klickt auf "Abbrechen"
 - ii. Das System fragt über einen Dialog nach, ob die eingetragenen Änderungen wirklich verworfen werden sollen
 - iii. Wenn der Benutzer auf "Ja" klickt werden die Änderungen verworfen

Nachbedingung(en)

Die neuen/veränderten Daten sind in der zentralen Datenbank gespeichert

Inkludierte Use Cases

Bilder laden/verwalten, GPS Track laden

Involvierte Klassen

User, Tour(m), TourType, TourAttribs(m), TourDescription(c), Content(c)

2.14 Use Case „Bilder hochladen/verwalten“

Kurze Beschreibung

Der Benutzer verwaltet Bilder und/oder lädt weitere hoch jeweils für Tourbeschreibungen (Kartenmaterial, Fotos) oder Kommentare.

Auslöser

- Benutzer klickt auf "Kartenmaterial hochladen/verwalten" in der Maske für Tourbeschreibung
- Benutzer klickt auf "Fotos hochladen/verwalten" in der Maske für Tourbeschreibung
- Benutzer klickt auf "Fotos hochladen" in der Maske für Kommentare

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, Benutzer angemeldet, Tourbeschreibung/Kommentar ausgewählt

Schritte

1. Das System zeigt die zugehörigen Bilder in einer Listenansicht mit Thumbnails an. Wenn zu einem Bild eine Beschreibung angegeben wurde, wird dies durch ein kleines Icon symbolisiert
2. Der Benutzer kann die Bildreihenfolge in dieser Ansicht verändern (drag&drop)
3. Durch einen Doppelklick auf ein Bild in der Listenansicht wird der Bearbeitungsmodus aktiviert. Darin kann der Benutzer das Bild austauschen (Upload eines neuen Bildes), die Bildbeschreibung (WYSIWYG) bearbeiten und einfache Bildmanipulationen vornehmen. Die Änderungen werden mit einem Klick auf "Übernehmen" übernommen bzw. bei Klick auf "Abbrechen" verworfen
4. Durch einen Klick auf "Bild einfügen" kann ein weiteres Bild hochgeladen werden. Hierfür wird ebenfalls der Bearbeitungsmodus (siehe 3.) benutzt
5. Der Benutzer klickt auf „übernehmen“ um die Änderungen zu übertragen

Alternative Schritte

2. a. i. Der Benutzer löscht ein oder mehrere Bilder in der Listenansicht (Bild markieren und Entfernen-Taste oder per Kontextmenü)
- ii. Das System zeigt einen Dialog an und fragt nach, ob die Bilder wirklich gelöscht werden soll
- iii. Bestätigt der Benutzer mit "Ja" werden die markierten Bilder entfernt
3. a. i. Beim Bild-Upload wählt der Benutzer ein nicht unterstütztes Dateiformat aus
- ii. Das System weist den Benutzer darauf hin und verhindert den Upload
- b. i. Der Benutzer löscht eine vorhandene Bildbeschreibung (entfernen des gesamten Textes)
- ii. Das System zeigt einen Dialog an und fragt nach, ob die Beschreibung wirklich entfernt werden soll
- iii. Bestätigt der Benutzer mit "Ja" wird die Bildbeschreibung entfernt, anderenfalls beibehalten

Nachbedingung(en)

Änderungen an den Bildern und Bilddaten werden lokal zwischengespeichert. Erst beim Speichern der jeweiligen Tourbeschreibung oder des Kommentars werden die Änderungen in die zentrale Datenbank übernommen.

Inkludierte Use Cases

-

Involvierte Klassen

User, Tour, TourType, TourAttribs(m), TourDescription, Content(c), Image(c)

2.15 Use Case „GPS Track laden“**Kurze Beschreibung**

Der Benutzer lädt einen zuvor in Google Earth erstellten GPS Track.

Auslöser

Klick auf "GPS Track laden" in der Detailansicht einer Tour

Akteur(e)

Benutzer

Vorbedingung(en)

System bereit, Benutzer angemeldet, Tour ausgewählt (Use Case "Tour anzeigen")

Schritte

1. Der Benutzer gibt die gewünschte KML-Datei an
2. Das System lädt die Datei und speichert eine lokale Kopie im Cache-Verzeichnis
3. Ein eigener Dialog zeigt die relevanten Daten des KML-Files als Baumstruktur an (Wegpunkte, zusätzliche Beschreibungen/Bilder)
4. Wenn im KLM-File nur Wegpunkte gespeichert sind, kann der Benutzer diese durch "Wegpunkte zu Pfad verbinden" in eine Strecke konvertieren
5. Der neue GPS Track wird lokal zwischengespeichert und erst beim Speichern der Tour in die zentrale Datenbank übernommen

Alternative Schritte

1. a. i. Die ausgewählte KML-Datei enthält keine Daten
ii. Der Benutzer wird per Dialog informiert und gebeten eine andere KML-Datei auszuwählen
- b. i. Die KML-Datei kann nicht korrekt gelesen werden
ii. Der Benutzer wird per Dialog informiert und gebeten eine andere KML-Datei auszuwählen

Nachbedingung(en)

-

Inkludierte Use Cases

-

Involvierte Klassen

User, Tour, GPSTrack(c), Content(c)

2.16 Use Case „Daten löschen“

Kurze Beschreibung

Der Benutzer löscht Daten aus dem System.

Auslöser

Benutzer

Akteur(e)

Benutzer

Vorbedingung(en)

Seine Tourbeschreibung ist die letzte Beschreibung der Tour (Stack)

Schritte

1. Benutzer meldet sich an
2. Benutzer sucht die gewünschte Tour
3. Benutzer wählt Tour aus
4. Benutzer löscht seine Tourbeschreibung
5. wenn es nicht die letzte Beschreibung der Tour ist (Stack) erscheint ein Fenster, dass ihn darauf hinweist, dass er die Beschreibung nicht löschen kann.

Alternative Schritte

-

Nachbedingung(en)

Beschreibung ist gelöscht

Inkludierte Use Cases

-

Involvierte Klassen

User, Tour(m), TourDescription(m), Comment(m)

2.17 Use Case „Touren verwalten“

Kurze Beschreibung

Der Administrator bearbeitet/löscht Daten aus dem System.

Auslöser

Administrator wechselt in die Admin-Ansicht

Akteur(e)

Administrator

Vorbedingung(en)

System bereit, Administrator angemeldet

Schritte

1. Gewünschte Tour suchen (Use Case „Touren suchen“) und auswählen
2. ENTWEDER Tour bearbeiten (Use Case „Tour bearbeiten“)
3. ODER Tourbeschreibungen löschen (=ältere Version wiederherstellen)
4. ODER Tour löschen (inkl. Abhängiger Daten - kaskadierend)

Alternative Schritte

-

Nachbedingung(en)

Vorgenommene Änderungen sind dauerhaft in der zentralen Datenbank gespeichert

Inkludierte Use Cases

Tour bearbeiten

Involvierte Klassen

User, Tour(m), TourDescription(m)

2.18 Use Case „Benutzer verwalten“

Kurze Beschreibung

Der Administrator bearbeitet/löscht Benutzer aus dem System.

Auslöser

Administrator wechselt in die Admin-Ansicht

Akteur(e)

Administrator

Vorbedingung(en)

System bereit, Administrator angemeldet

Schritte

1. System zeigt die vollständige Benutzerliste an
2. Administrator kann mit Such- und/oder Sortierfunktion die gewünschten Benutzer selektieren
3. Selektierte Benutzer können gelöscht bzw. bearbeitet werden

Alternative Schritte

-

Nachbedingung(en)

Vorgenommene Änderungen sind dauerhaft in der zentralen Datenbank gespeichert

Inkludierte Use Cases

-

Involvierte Klassen

User (m)

2.19 Use Case „Kommentare verwalten“

Kurze Beschreibung

Der Administrator bearbeitet/löscht Kommentare aus dem System.

Auslöser

Administrator wechselt in die Admin-Ansicht

Akteur(e)

Administrator

Vorbedingung(en)

System bereit, Administrator angemeldet

Schritte

1. Gewünschte Tour suchen (Use Case „Touren suchen“) und auswählen
2. Kommentare dieser Tour durchsuchen und gewünschtes löschen

Alternative Schritte

-

Nachbedingung(en)

Vorgenommene Änderungen sind dauerhaft in der zentralen Datenbank gespeichert

Inkludierte Use Cases

Tour bearbeiten

Involvierte Klassen

User, Tour, TourDescription, Comment(m)

3 Klassenmodell

3.1 Klassendiagramm

Nachfolgende Grafik zeigt das fachliche Klassendiagramm für das TourManagementsystem.

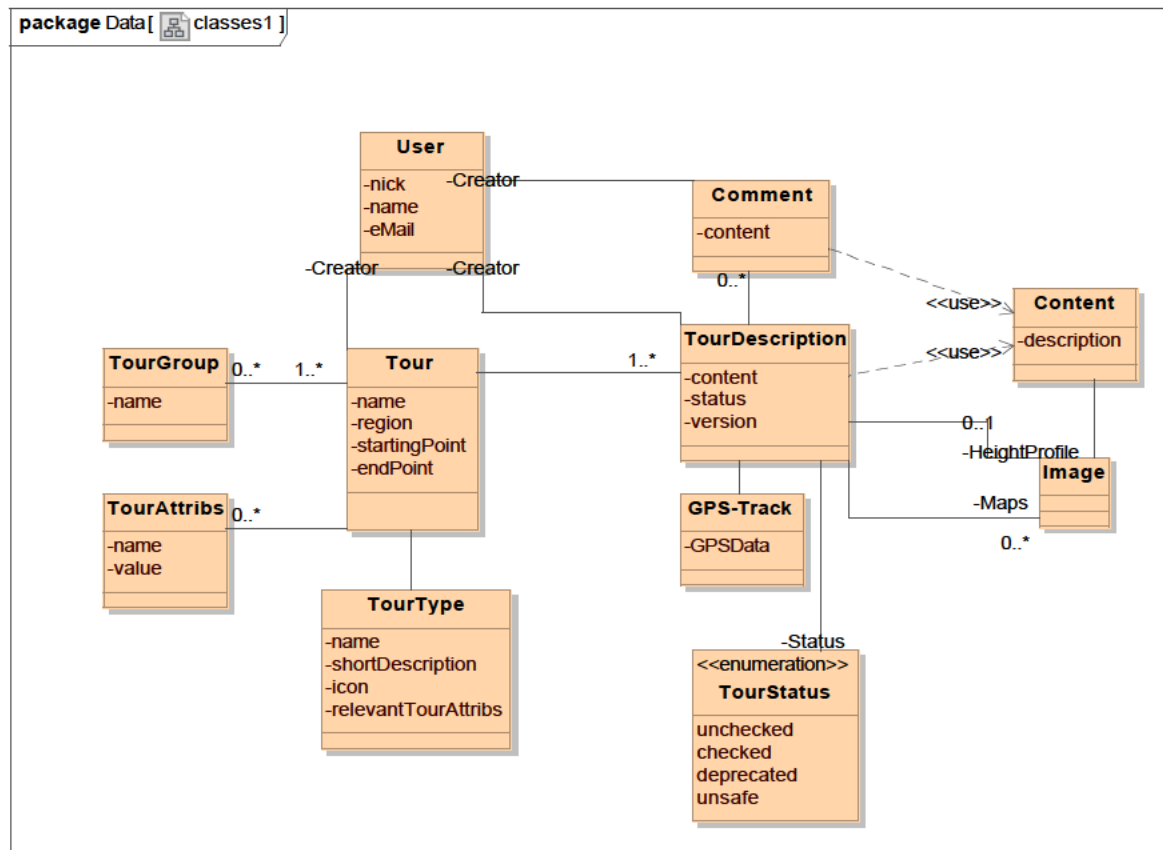


Abbildung 2: Fachliches Klassendiagramm

Bemerkungen zu Abbildung 2:

In diesem Diagramm wird nicht zwischen Gast, Benutzer und Administrator unterschieden – diese Ausprägungen werden über eine Rechteverwaltung umgesetzt.

Die Daten einer Tour sind folgendermaßen organisiert:

- Das Objekt Tour enthält die Stammdaten
- Die vom jeweiligen Tour-Typ abhängigen Attribute werden direkt mit der Tour (also den Stammdaten verknüpft)
- Die eigentliche Beschreibung wird im Objekt TourDescription verwaltet. Dieses Objekt wird zur Umsetzung der Versionierung herangezogen
- Kommentare (Objekt Comment) werden jeweils mit der zum Erstellungszeitpunkt gültigen Version der Tour-Beschreibung verknüpft

3.2 Zustandsmodelle

3.2.1 Klasse Tour

Eine neue Tour wird mit Status „active“ angelegt. Wenn ein Benutzer eine Tour als „deprecated“ markiert bleibt Sie in diesem Zustand, kann nach wie vor eingesehen werden, jedoch nicht mehr bearbeitet werden. Eine gelöschte Tour (Status „deleted“) wird im System nicht mehr angezeigt, bleibt jedoch zu Archivierungszwecken in der Datenbank gespeichert. Eine Tour kann von einem Admin in den Status „locked“ und wieder zurück gesetzt werden. Eine gesperrte Tour kann eingesehen jedoch nicht bearbeitet werden.

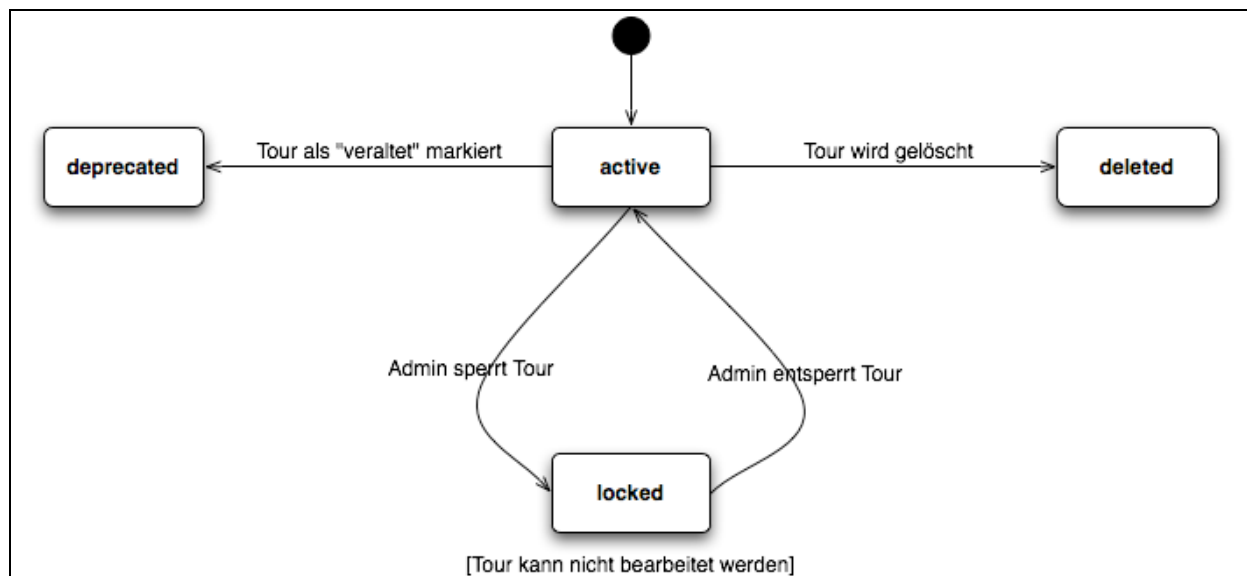


Abbildung 3: Zustandsdiagramm der Klasse Tour

3.2.2 Klasse TourDescription

Eine neue Tourbeschreibung wird mit Status „active“ angelegt. Eine gelöschte Beschreibung wird mit Status „deleted“ markiert und lediglich zu Archivierungszwecken in der Datenbank gehalten, eine Anzeige ist nicht mehr möglich. Wird eine Tourbeschreibung aktualisiert, wird die alte Version auf Status „deprecated“ gesetzt und die neue Beschreibung mit Status „active“ gespeichert.

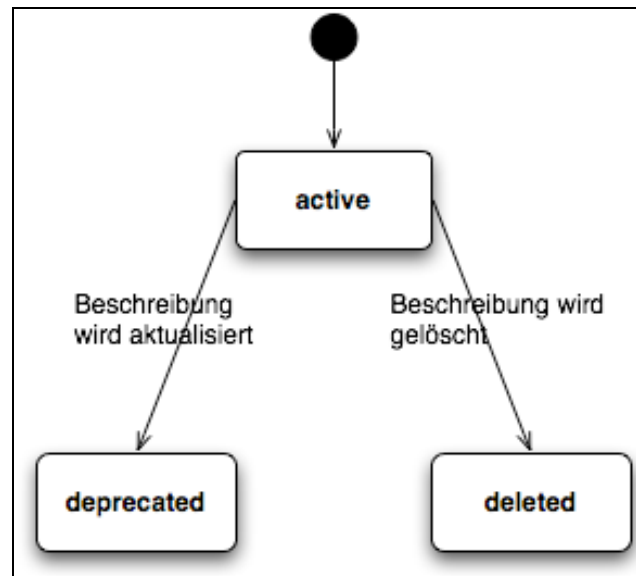


Abbildung 4: Zustandsdiagramm der Klasse *TourDescription*

3.2.3 Klasse Comment

Ein neues Kommentar wird mit Status „open“ angelegt. Solange ein Kommentar diesen Status besitzt kann sie vom Ersteller noch verändert werden. Nach 24 Stunden wird der Status automatisch auf „closed“ gesetzt. In diesem Status ist eine Bearbeitung nicht mehr möglich. Kommentare werden – wie die anderen Datensätze auch – nicht in der Datenbank gelöscht, sondern mit dem Status „deleted“ markiert.

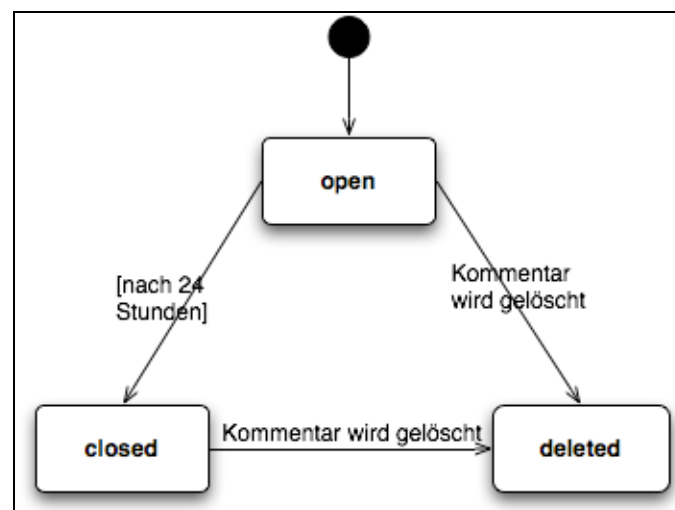


Abbildung 5: Zustandsdiagramm der Klasse *Comment*

3.2.4 Klasse User

Ein Benutzer wird mit Status „active“ angelegt. Bei Verletzung der Nutzungsbedingungen (z.B. Beleidigung in Kommentaren) kann der Nutzer durch einen Administrator blockiert werden. Dieser Status verhindert eine Anmeldung am System und der Benutzer kann nur mehr als Gast mit dem System interagieren.

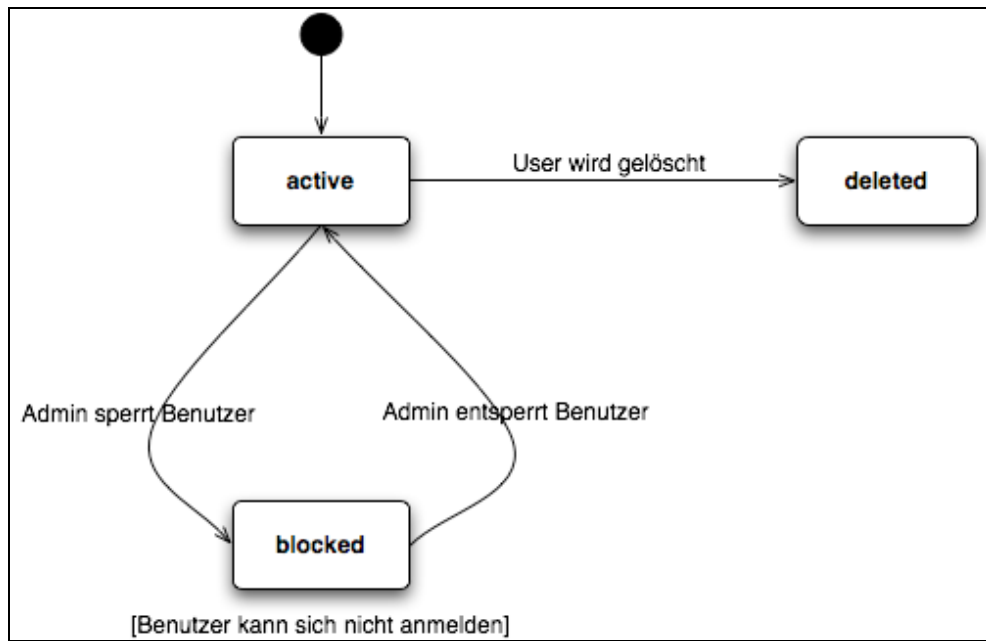


Abbildung 6: Zustandsdiagramm der Klasse User

4 Umsetzungs-Details

4.1 Zielplattform

Die Tour-Management-Software Tour66 wird als Java-Applikation umgesetzt. Zielplattform ist Java 5.0 SE und damit alle gebräuchlichen Betriebssysteme. Besondere Hardware-Anforderungen werden nicht gestellt, auf ältere Computermodelle wird jedoch nur bedingt Rücksicht genommen. Eine Bildschirmauflösung von mindestens 1024*786 Pixel wird – bedingt durch den Charakter der zu erfassenden bzw. anzuzeigenden Tourdaten – vorausgesetzt.

4.2 Datenhaltung

Die Datenhaltung erfolgt in einer zentralen Datenbank. Während der Entwicklung wird primär Oracle (Version tbd) fokussiert. Die Einbindung weiterer Datenbanksysteme wird durch Einsatz von Hibernate bei Implementierung der Persistenz-Schicht prinzipiell unterstützt.

4.3 Oberfläche

Die Oberfläche wird mit Swing implementiert, NetBeans 6.0.1 wird als GUI-Builder eingesetzt. Für größtmögliche Flexibilität betreffend Look&Feel sowie Benutzerführung wird auf den Einsatz von SWT/JFace oder Eclipse RCP verzichtet. Stattdessen wird bei Bedarf auf entsprechende Bibliotheken (nicht kommerziell, Open-Source-Lizenzen) zurückgegriffen.

Beim Design der Oberflächenkomponenten wird auf größtmögliche Übersichtlichkeit, klare Benutzerführung und intuitive Bedienbarkeit geachtet. Zusätzlich wird ein „Farbleit-System“ eingesetzt, dass unterschiedliche Farben für bestimmte Anwendungsfälle vorsieht.

4.4 Systemarchitektur (grob)

Eine eigene Server-Komponente (als Teil der Anwendungsschicht) ist nicht vorgesehen. Aus heutiger Sicht wäre der Nutzen einer solchen Lösung verhältnismäßig gering. Dem gegenüber steht ein nicht unerheblicher Mehraufwand bei der Implementierung und in späterer Folge beim Betrieb des Systems. Daher wird Tour66 als Client-Applikation mit integrierter Datenbank-Anbindung umgesetzt.

* * * * *

* * * * * 09.04.2008 * * * * *