PS Software Engineering

PS-SWE-WS2000 1. Februar 2001

Archäologisches Managementsystem

Report

Matthias Ausweger
maus@cosy.sbg.ac.at

Werner Weiser
wweiser@cosy.sbg.ac.at

Franz Wilhelmstötter
fwilhelm@cosy.sbg.ac.at

Kasbi Mohammed Jamal
jkasbi@cosy.sbg.ac.at

Mohamed Guemboura
mguem@cosy.sbg.ac.at

Academic Supervisor Mag. Dipl.-Ing. Schwaiger Roland

Department of Computer Science
University of Salzburg

Correspondence to:
Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Archäologisches Managementsystem

Mohamed Guemboura, Franz Wilhelmstötter, Matthias Ausweger,
Werner Weiser, Kasbi M. Jamal
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
mguem@cosy.sbg.ac.at, fwilhelm@cosy.sbg.ac.at, maus@cosy.sbg.ac.at
wweiser@cosy.sbg.ac.at, jkasbi@cosy.sbg.ac.at

Inhaltsverzeichnis

Zusammenfassung

Erstellung eines Software- Packetes, das zur Erfassung und Verwaltung von archäologischen Daten verwendet wird. Die Erfassung, sowie die Bearbeitung und Darstellung der Daten soll webbasierend sein. Weiters wird über das Internet die Suche nach archäologgischen Daten (Funden) durch eine Suchmaschine ermöglicht. Eine bestimmte Auswahl an Informationen kann, durch die Erfasser aufbereitet, über das Internet in einem *virtuellen Museum* (VM) betrachtet werden.
Die Erstellung einer CD- ROM aus einem VM und die Zusammenstellung einer Auswahl der archäologischen Daten zu einem Fund- Katalog soll ebenfalls möglich sein.

Kapitel 1

Anforderungsdefinition

Die folgende Problembeschreibung soll eine ersten Überblick über das gesamte Projekt vermitteln.

1.1 Ausgangssituation und Zielsetzung

Erstellung einer Software zur Suche nach, Erfassung und Verwaltung von archäologgischen Daten über ein Web- Frontend, sowie die Darstellung dieser Daten in einem virtuellen Museum. Diese Daten sollen auch wahlweise in Form einer CD- Rom oder eines druckbaren Katalogs dargestellt werden können.

1.1.1 Mußkriterien

- Erfassung und Bearbeitung von archäologischen Grunddaten.
- Erfassung und Bearbeitung von wissenschaftlichen Berichten, sowie Publikationen und Interpretationen der archäologischen Funde.
- Die Erfassten Artifakten sollen ab der Zeitpunkt der Erfassung im Internet zu verfühgung stehen,sobald sie freigegeben worden sind.
- Suchen nach bereits erfassten archäologischen Daten anhand verschiedener Suchkriterien.
- Erstellung und Gestalltung eines VM, das vom Internet aus besucht werden kann.
- Erstellung und Gestalltung von Sonderschauen.
- Für den Zugriff auf die archäologischen Daten sind Benutzergruppen mit entsprechenden Rechten und Einschränkungen einzurichten.

1.1.2 Produkteinsatz

- Der Zugang zum archäologgischen Datenbank-System erfolgt über Web- Browser (IE, Netscape, ...). Dadurch wird eine Plattformunabhängigkeit gewährleistet.

Anwendungsbereich

1. Eingabe und Verwaltung von archäologgischen Daten.
2. Eingabe und Verwaltung von Personen- Daten (User).
3. Eingabe und Verwaltung von weiterführender Literatur und Informationen (Bild, Ton, Video).
4. Suche nach Detail- Informationen für Erfasser.
5. Erstellung und Bearbeitung eines VM.
6. Erstellung und Bearbeitung von Sonderschauen.
7. Suche nach Detail- Informationen durch Internet- Benutzer.
8. Besichtigung des VM und der Sonderschauen (veröffentlicht) durch den Internet-Besucher.
9. Erstellung einer CD- ROM mit dem VM.
10. Erstellung eines archäologgischen Kataloges.

Die Reihenfolge der angeführten Punkte entspricht einer von uns gewählten Prioritätenreihenfolge.

Betriebsbedingungen

- Zugang über Internet mittels gewöhlichem Arbeitsplatzrecher (PC, Laptop).
- Browser
- Server zentral bei Service-Provider mit Fernwartung.

1.1.3 Produktumgebung

Software

- Daten in Relationalem DBMS (wahlweise DB2 - IBM).
- Application- Server
- Web- Server

Hardware

Client: Es genügt ein gewöhnlicher Arbeitsplatzrechner (PC, Laptop) mit Internet-Zugang.

Server: Vorzugsweise ein UNIX-Server.

Backup-Server Wie Server.

1.1.4 Produktdaten

Suchmaschine

Es gibt eine Suchmaschine, mit der, anhand von bestimmten Kriterien, nach archäologischen Daten gesuchen werden kann. War eine Suche erfolgreich und wurden mehrere Ergebnisse gefunden, so werden diese in einer Ergebnis- Auflistung angezeigt. Aus dieser Auflistung kann man dann ein einzelnes Datum auswählen und davon die Detail- Informationen anzeigen lassen.
Die angezeigten Informationen und die Kriterien der Suche sind beim Internet- Besucher auf eine gewisse Grundauswahl beschränkt. Der Erfasser hat Zugang zu allen Informationen und auch eine größere Auswahl an Such- Kriterien.

Virtuelles Museum

Im gesamten System gibt es nur ein VM. Ein VM ist unterteilt in Stockwerke, welche Themenbereichen entsprechen. Die Stockwerke sind wieder unterteilt in verschiedene Räume. Die Räume enthalten Vitrinen, die ein oder mehrere Artefakte enthalten. Wird ein Artefakt zur Detail- Ansicht ausgewählt, so werde die entsprechenden eingeschränkten Daten bezüglich des Artefaktes übersichtlich dargestellt. Der Verwalter (siehe Benutzergruppen) organisiert die Gestaltung des Gesamtmuseums.
Die Gestalltung eines Raumes übernimmt ein Ausgrabungsleiter, der anhand verschiedener Templates (Vorlagen) eine Auswahl an Gestalltungsmöglichkeiten für die Anordnung von Vitrinen, Artefakten und Detail- Informationen zu den Artefakten hat. Weiters kann der Erfasser zusätzliche 'Schautafel- Informationen' für Raum, Vitrine und Artefakte erstellen. Die Erstellung einer Schautafel- Information für das gesamte Museum und der Stockwerke obliegt dem Verwalter.

Sonderschau

Jeder Ausgrabungsleiter ist berechtigt, eine Sonderschau zu erstellen, welche im Wesentlichen wie ein Raum im VM aufgebaut ist. Bei der Erstellung und Gestalltung hat er die Möglichkeiten wie bei der Erstellung eines Raumes im Museum (siehe dort).

Datenmodell

Einen genaueren Überblick über die aufzunehmenden Daten, bietet folgende Auflistung.

DATA-ITEMS für archäologgische Daten:

- Katalog- Nummer.
- Abmessungen (Größe, Länge, Gewicht, ...)
- Zustand
- Material (Zusammensetzung)
- Alter
- Epoche

- Beschreibung
- Bezeichnung
- Kommentar (z.B. zur Grabeigenschaft)
- Finder
- Fundort (absolute Koordinaten?)
- Fundzeitpunkt
- Kategorie (Waffe, Kleidung, Tier, Gefäß)
- Bild-, Video-, Ton- Material (Bsp: Lageskizze)

DATA-ITEMS für wissenschaftliche Daten :

- Verfasser
- Titel
- Beziehungen (zu archäologgischen Daten)
- Kategorie (Publikation, Interpretation, Video, Ton, Bild, ...)
- Daten an sich.

DATA-ITEMS für Personen-Daten (Erfasser):

- Anschrift, Adresse, ...

1.1.5 Abgrenzungskriterien

- Es erfolgt keine Kontrolle über die inhaltliche Richtigkeit der eingetragenen Daten seitens der Software.
- Es gibt nur ein VM, das mehrere Themenbereiche (Stockwerke) mit mehreren Räumen enthält. Darüber hinaus sind Sonderschauen zu aktuellen Themen durch das Portal des VM zugänglich.
- Der Internet-Besucher hat nur eingeschränkten Zugriff zu den Daten (keine oder nur ausgewählte Videos, Ton-Beispiele und weitereführende Details), damit keine unübersichtliche Informationsflut entsteht und die Resourcen nicht zu schnell überlastet werden (DB).
- Es entstehen keine Beziehungen zwischen den Artifakten auf Fakten-Basis,da die Fakten-Basis nur als Grund-Daten zu betrachten sind.
- Ein noch nicht freigegebener Artifakt wird dem Internet-Besucher(Anonymous) nicht angezeigt.

1.2 Benutzerschnittstellen

Die konkrete Beschreibung der Benutzerschnittstellen verlangt die Entwicklung eines Prototypes, deshalb siehe später.
Das Layout der Gesamt Benutzerschnittstellen soll einheitlich sein,mit ausnahme des VM und Sonderschauen die vom Anwender(Verwalter,Erfasser)selbst festgelegt wird.
Die Benutzerschnittstellen unterteilen sich in 5 Hauptschnittstellen:

- Erfassungsmaske und Bearbeitungsmaske. Für die Daten der Artifakten.
- Erfassungsmaske und Bearbeitungsmaske. Für die Wissenschaftlichen Arbeiten.
- Suchmaske.
- Erstellungsmaske und Gestaltungsmaske. Für das VM und Den Sonderschauen.
- DB-Anfagemaske. Für die verschiedenen Anfragen an der DB vom Verwalter und teilweise vom Erfasser.

1.3 Funktionale Anforderungen

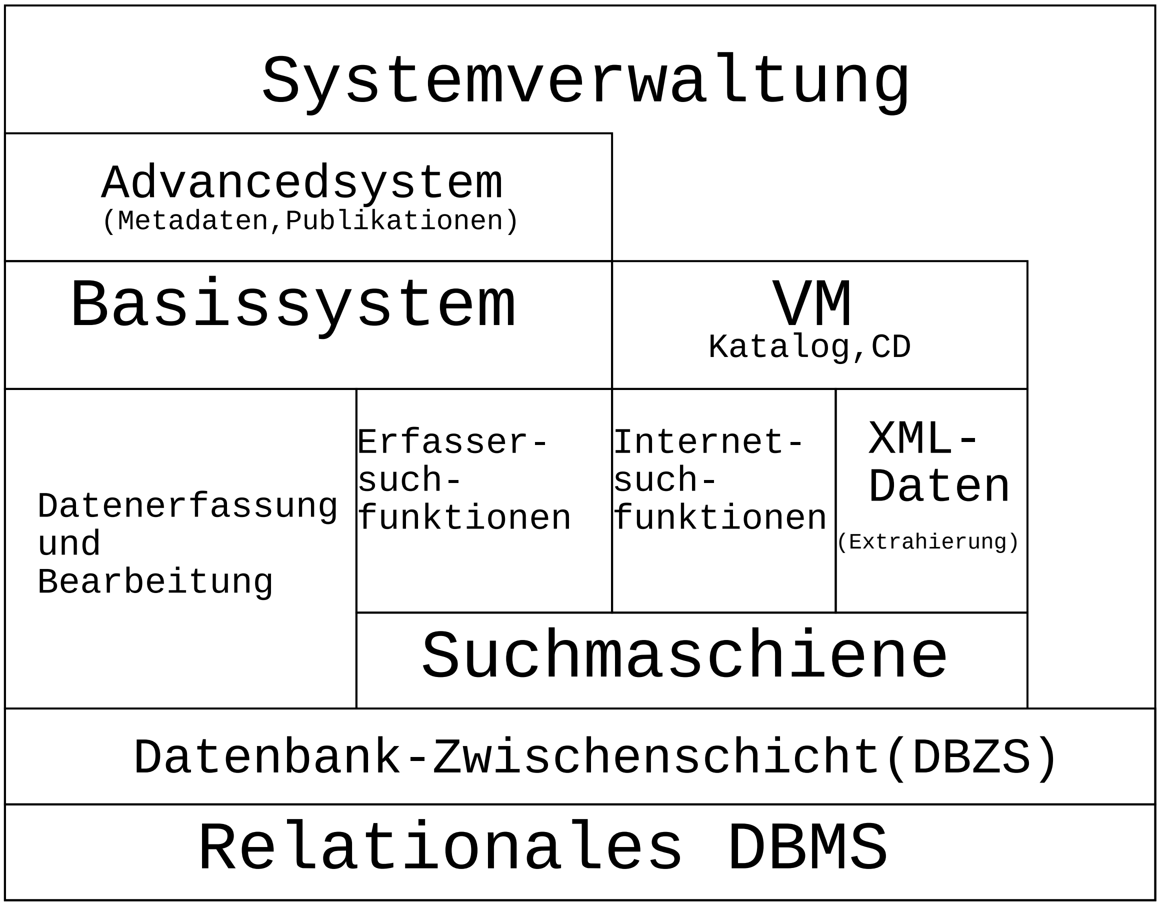


Abbildung 1.1: Systemdiagramm

Das Diagramm ist so aufgebaut, daß Elemente, welche als erstes Implementiert werden müssen sich weiter unten befinden. Die unterste Schicht stellt die Datenbank, bzw. das DBMS dar. Hier werden (ausnahmslos) alle für das System relevanten Daten (einschließlich der User- Verwaltung) gespeichert. Über der Datenbankschicht ist eine Zwischenschicht eingefügt, über welche man die Datenbankzugriffe realisiert. Diese Schicht hat den Sinn, eine bessere Skalierbarkeit der Datenbank zu erreichen. D.h. sollte es, durch den großen Erfolg des Systems, eine Verteilung der Suchabfragen auf mehrere Maschinen (Prozessoren) notwendig sein, so muß nur diese Zwischenschicht angepaßt werden. Für alle darüberligenden Layer ist diese Zwischenschicht transparent. Die Datenerfassung, Datenbearbeitung und die Suchdienste (Suchmaschine) setzt auf den Diensten auf, die durch die DBZS (Datenbank-Zwischenschicht) zur Verfügung gestellt werden. Das Basissystem besteht aus der Datenerfassung, Datenbearbeitung und den Erfassersuchfunktionen. Die Erfasser- Suchfunktionen nutzen die Dienste des Suchmaschinen- Layers. Über dem *Basissystem* liegt das *Erweiterte System (Advanced System)*. Hier wird die Verwaltung der Metadaten, Interpretationen und Publikationen realisiert. Das *virtuelle Museum (VM)* ist vom Basissystem unabhängig. Die Suchfunktionen für den Internetbenutzer werden auch vom VM genutzt. Die eigentliche Datenbasis des VM ist nicht direkt die Datenbank, sondern eine Menge von XML- Datein, welche aus der Datenbasis des RDBMS extrahiert werden. Die *System- Verwaltung* (Benutzerverwaltung, Rechteverwaltung usw.) überspannt das gesamte System.
Anhand dieser Gliederung ist es möglich eine Aufgabenverteilung vorzunehen. Weiters ist es möglich eine Aufwandsschätzung (Zeit, Kosten) vorzunehmen.
Ausgehend vom Systemdiagramm (Abbildung 1.1), werden die funktionalen Anforderung, in den dort definierten Teilen, gegliedert beschrieben.

1.3.1 Datenbank

Sämtliche Ausgrabungs- und Artefaktdaten werden in der DB abgelget. Eine Ausgrabung beschreibt eine archäologische 'Expedition', die zu einer bestimmten Zeit unter einem bestimmten Ausgrabungsleiter in einem bestimmten Gebiet stattgefunden hat. Tabelle 1.1 listet die, eine Ausgrabung bestimmenden Attribute auf.
Ein Ausgrabungsort besitzt ein oder mehrere Fundgegenstände. Der Fundgegenstand ist hirachisch eingeteilt. Beginnend auf der untersten Ebene, dem Artefakt, bis zum Ausgrabungsort. Z.B. ein Fundgegenstand 'Schwert', dieses Schwert ist Teil des Fundgegenstandes 'Grab' usw. Damit läßt sich eine Hierachie bis zur Ausgrabung aufbauen - welche aus mehreren Fundgegenständen bestehen kann, auf oberster Ebene. Weiters wird am Ausgrabungsort ein Koordinatenursprung definiert. Die Koordinaten aller Funde werden relativ zu diesem angegeben. Der Ursprung ist durch die geographische Länge, Breite und Höhe (festzustellen mittels GPS) eindeutig festgelegt. Tabelle 1.2 ist eine Auflistung der Attribute eines Fundgegenstandes.

Allen Bild-, Ton- und Videodokumente erhalten ein 'Bezeichnung', 'Beschreibung' und ein zusätzliches Attribut, welches festlegt, ob es für den Internetbenutzer zugänglich ist (um z.B. eine Überlastung des Sytems, durch das Abspielen langer Videosequenzen, zu verhindern). Dieses Attribut legt fest, ob es sich dabei um das 'Hauptbild' oder um ein 'Nebenbild' handelt (ein 'Hauptbild' ist für den anonymen Benutzer sichtbar). Die *bounding Box* eines Fundgegenstandes gibt dessen größte Ausdehnung an (in der Lage in der er gefunden wurde). Alle bounding Boxes sind Quader.

Attributname	Type	mandatory
Bezeichnung	text	ja
Beschreibung	text	nein
Dauer von	datum	ja
Dauer bis	datum	nein
Kosten	decimal	nein
Ausdehnung	bounding Box	ja
Koordinatenursprung der A.	koordinaten	ja
Projektziel	text	nein
Projektergebnis	text	nein
Staat	text	nein
Land	text	nein
Kommentar	text	nein
Projektleiter	perosn	ja
Projektteilnehmer	{person}	nein
Person teilgenommen von	datum	ja
Person teilgenommen bis	datum	nein
Fundobjekte	{fundgegenstant}	nein
Bilder zur Ausgrabung	{blob}	nein
Video zur Ausgrabung	{blob}	nein
Audioaufnahmen z. A.	{blob}	nein

Tabelle 1.1: Ausgrabungsattribute

Da alle Benutzer ebenfalls in der DB vorhanden sein müssen, listet Tabelle 2.4 die Attribute der Personen bzw. *User* auf. Jede Person, die je ein Ausgrabung geleitet, bzw. einen Fundgegenstand eingetragen hat, wird hier erfaßt. Verläßt eine Person ein Ausgrabungsteam (sodaß ihm keine Eintrag- Rechte mehr zustehen), werden ihm nur die entsprechenden Benutzerrechte entzogen; als Person bleit der Eintrag in der DB bestehen (da in der DB (möglicherweise) bereits Referenzen auf diese Person, z.B. als Ausgrabungsleiter, existieren).

1.3.2 DB- Zwischenschicht (DBZS)

Diese Zwischenschicht dient dazu, alle darüberliegenden Systemteile einen direkten Datenbankzugriff unnötig zu machen. Daraus ergibt sich unter anderem der Vorteil, nicht für jede Benutzeranfrage eine eigen DB- Verbindung aufbauen zu müssen (Connection- Pool). [1] Die Schicht kann auch dazu dienen, den Wechsel auf ein anderes RDBMS zu erleichtern. In diesem Fall muß nur die DBZS ausgetauscht werden. Weiters ist auch ein feineres 'Load- Ballancing' und eine zusätzliche Fehlerprotokollierung möglich (falls gewünscht oder erforderlich). Die DBZS kann in weiter kleiner Teile unterteilt werden. Jeder dieser Teile übernimmt eine spezielle (wie oben angedeutet) Aufgaben.

Aufgaben der DBZS:

[1] Durch mehrere offene Verbindungen wird die DB nicht unbedingt schneller.

Attributname	Type	mandatory
Bezeichnung	text	ja
Beschreibung	text	nein
Katalognummer	text	nein
Fundzeit	time	ja
pos. Länge	decimal	ja
pos. Breite	decimal	ja
pos. Höhe	decimal	ja
Größe des Gegenstandes	bounding box	ja
Masse	decimal	ja
Teil von Fundgegenstand	fundgegenstand	nein
Alter	integer	nein
Material	{text}	nein
Fundortbeschreibung	text	ja
Finder	{person}	ja
Bild	{blob}	nein
Bildattribut	{text}	nein
Audio	{blob}	nein
Audioattribut	{text}	nein
Video	{text}	nein
Videoattribut	{text}	nein

Tabelle 1.2: Attribute der Fundgegenstände

Attributname	Type	mandatory
Vorname	text	nein
Nachname	text	ja
Titel / Anrede	text	nein
Benutzerrechte	{rechte}	ja

Tabelle 1.3: Attribute zur Person (User)

Query- Manager: Liegt direkt über der Datenbank. Nimmt die Abfragen aus den oberen Schichten entgegen, stellt sie in eine Warteschlange und leitet sie an eine oder mehrere offenen Datenbankverbindungen (parametrisierbar; je nach dem wieviel gleichzeitige DB- Verbindungen sinnvoll sind - hängt u.a. von der Anzahl der Prozessoren der DB-Maschine ab). Das Abfrageergebnis wird an den aufrufenden Systemteil zurückgegeben.

1.3.3 Basissystem (BS)

Das Basissystem stellt den Kern der Anwendung dar. Das BS bietet den Erfassern (Ausgrabungsleitern, Assistent) die Funktionalität für das Erfassen, Ändern und Suchen von Artefakten. Die Erfassung von Metadaten, wie Interpretationen und Verweisen zu anderen Publikationen, werden vom BS nicht unterstützt.

1.3.4 Verwaltung

Die Verwaltungsschicht hat folgende Aufgaben:

- native Systemverwaltung der Datenbank
- Rechte,- Benutzererwaltung
- Verwaltung des VM

Rechte-, Benutzerverwaltung

Jeder, der Artefakte erfassen will, muß dem System als Benutzer bekannt sein und über die notwendigen Rechte verfügen. Eingetragen Benutzer erhalten aus diesem Grund eine Menge von Rechten, die durch sogenannte Rollen zusammengefasst werden. Die einzelnen Rechte sind atomar. Es werden folgende Rechte definiert:

Suchrecht: Das Suchrecht ist das fundamentalste aller Rechte. Dieses Recht besitzt auch der Internetbenutzer (Anonymous).

Erstellerrecht: Der Inhaber dieses Rechts ist in der Lage neue Einträge in der Datenbank zu erstellen. Welche Art von Einträgen erstellt werden kann, hängt davon ab, welcher Benutzergruppe der Eintragende angehört. Nur einem Ausgrabungsleiter ist es gestattet einen neuen *Ausgrabungseintrag* zu erstellen. Dem Ausgrabungsassistenten ist dies nicht erlaubt. Er darf ledeglich einzelne Artefakt- Einträge erstellen.

Änderungsrecht: Dieses Recht gestattet es dem Rechtinhaber, bestehende Einträge zu ändern. Der Gültigkeitsbereich (Scope) dieses Rechts ist von der Benutzergruppe und dem Eigentümer eines Eintrags abhängig.

Freigaberecht: Bevor ein Artefakt dem normalen Internetbenutzer zugänglich ist, muß es vom Ausgrabungsleiter - oder einem anderen Berechtigten - freigegeben werden. Dieses Recht gestattet es noch unvollständige Einträge zu *verbergen*.

Rechtevergabe- Recht: Der Rechteinhaber kann die Menge der Rechte (Roles) der Benutzer ändern. Hat das Recht neue Benutzer anzulegen. Auch hier bestimmt die Benutzergruppe das Ausmaß des Rechts. Ein Ausgrabungsleiter hat das Recht, neue Assisten in das System einzutragen. Es ist aber nur gestattet, ihnen das Recht zum Eintragen von Artefakten zu gewähren.

Mit der Zuordnung einer Menge von Rechten zu einem Benutzer [2] , ist jedoch noch keineswegs der Gültigkeitsbereich dieser Rechte festgelegt. Zu diesem Zweck müssen noch *Benutzergruppen* eingeführt werden. Es werden folgende Benutzergruppen definiert:

Verwalter (Superuser): Angehörgie dieser Gruppe haben uneingeschränkte Rechte auf alle Einträge der Datenbank.

Ausgrabungsleiter: Ein Ausgrabungsleiter kann Ausgrabungseinträge erstellen, Artefakte erstellen, Artefakte löschen (solange noch keine Freigabe darauf existiert) und Artefakte freigeben. Wenn ein Ausgrabungsleiter eine Beziehungen zwischen Funden herstellen will, so muß er der Besitzer von mindestens einem der Funde sein.

Ausgrabungsassistent: Ein Assisten ist in der Lage einzelne Artefakte in eine bereits bestehende Ausgrabung einzutragen. Solange noch keine Freigabe darauf existiert, kann er sie beliebig bearbeiten (auch löschen - aber nur 'seine').

Anonymous: Als anonym gilt der normale Internetbenutzer. Er hat nur das Recht zu Suchen (auf allen in der Datenbank freigegebenen Artefakten).

Es sei darauf hingewiesen, das ein Benutzer, der der Benutzergruppe *Ausgrabungsleiter* angehört, Artefakte nur freigeben und löschen kann, wenn er über die entsprechenden Rechte verfügt. Die Benutzergruppe legt nur den *Gültigkeitsbereich (Scope)* des jeweiligen Rechts fest.

1.3.5 Systematische Beschreibung der Funktionalen Anforderungen

Die einzelnen funktionale Anforderungen werden in loser (zumeist unzusammenhängender Form) aufgelistet. Die Prioritäten reichen von 1 bis 3 - 1 für höchste Priorität und 3 für die niedrigste.

Basissystem

BSF1: Die Eingabe von Artefakten soll über ein Web- Frontend erfolgen.
{Priorität 1}

BSF2: Die Suche nach Artefakten - für den Erfasser und den Internetbesucher - soll über ein Web- Frontend erfolgen.
{Priorität 1}

[2] Hier und in weiterer Folge ist mit 'Benutzer' nicht der Internetnutzer gemeint.

Datenbasis

DBMF1: Alle für das Artefakt wesentliche Daten müssen in der Datenbank gespeichert werden.
{Priorität 1}

DBMF2: Publikationen und Interpretationen von Artefakten werden in der Datenbank abgelegt.
{Priorität 2}

DBMF3: Die Rechte für das Eintragen von Artefakten ist nicht jedem Benutzer gleichermaßen gestattet. Für die Verwaltung der Datenbases werden atomare Rechte definiert. Diese werden in Benutzergruppen zusammengefaßt. Die Verwaltung der Rechte und der Gruppen erfolgt ebenfalls in der Datenbank.
{Priorität 1}

DBMF4: Das Recht, einem Benutzer Rechte zu vergeben ist eine atomares Recht.
{Priorität 1}

DBMF5: Das Recht, ein Artefakt in die Datenbank einzutragen ist ein atomares Recht.
{Priorität 1}

DBMF6: Bevor ein Artefakt dem Internetbenutzer zugängliche ist, muß es zuerst freigegeben werden.
{Priorität 1}

DBMF7: Des Recht, zur Freigabe eines Artefakts ist ein atomares Recht.
{Priorität 1}

DBMF8: Das Recht, in der Datenbank zu suchen ist ein atomares Recht.
{Priorität 1}

DBMF9: Die Benutzergruppe *Verwalter (Superuser)* besitzt alle atomaren Rechte.
{Priorität 1}

DBMF10: Die Benutzergruppe *Ausgrabungsleiter* hat die Rechte ein Artefakt einzutrage, zu ändern und freizugeben. Sie kann neu Benutzer anlegen und ihnen das Recht zum Eintragen von Artefakten geben.Ein Ausgrabungsleiter kann Sonderschauen und Teile eines VM erstellen.
{Priorität 1}

DBMF11: Die Benutzergruppe *Ausgrabungsassitent* hat das Recht Artefakte einzutragen.
{Priorität 1}

DBMF12: Das Recht, das VM zu verwalten ist ein atomares Recht.
{Priorität 1}

DBMF13: Das Recht, Teile des VM/CDs/Kataloge zu erstellen ist ein atomares Recht.
{Priorität 1}

VM

Funktionale Aspekte: Die Suche nach Informationen über archäologische Daten erfolgt in gleicher Weise wie die Erfasser- Suchfunktion.

VM-F 1: Ein Dialog zur wahlweisen Erstellung verschiedener Räumlichkeiten (Sonderschau, Raum in einer Sonderschau, Raum im VM). Als Basisgestaltungsform werden hier mehrere Templates zur Auswahl angeboten, die bereits eine Grundstruktur für den Aufbau festlegen.
{Priorität 2}
Rationale: Dies bildet den Eingangsdialog zur Erzeugung von Sonderschau/Raum im VM. Der Erstellungsweg ist modular hierarchisch aufgebaut (Schau, Raum, Vitrine, Artefakt).

VM-F 2: Ein Dialog zur Suche nach archäologischen Daten (Siedlung, Grab, Artefakt). Die Antwortzeit für eine Suche darf fünf Sekunden nicht überschreiten.
{Priorität 2}
Rationale: Eine umfassende Datenbankabfrage über Informationen kann sehr aufwendig sein. Ein Erfasser, der einen Raum / eine Sonderschau erstellen darf, kann nach einer Vielzahl von Kriterien nach den Informationen suchen, daher kann die Suche aufwendig werden. Eine längere Suchdauer ist jedoch vom Benutzer nicht akzeptabel.

VM-F 3: Der Dialog zur Suche bietet ein *Dummy-Eingabefeld* in das man einen oder mehrere Begriffe eingeben kann. Die Suche wird je nach Verknüpfung (AND, OR) über alle archäologischen Daten durchgeführt.
{Priorität 2}
Rationale: Da in einer Sonderschau/einem VM-Raum jede Art von Daten angezeigt werden kann, muss auch nach jeder Art von Kombinationen gesucht werden können.

VM-F 4: Ein Dialog, der die Suchergebnisse mit knappen Hauptinformationen in einer Liste anführt. Aus der Liste kann ein einzelnes Datums ausgewählt werden, von dem dann die Detailinformationen angezeigt werden.
{Priorität 2}
Rationale: Da viele Suchergebnisse möglich sind, muß eine Überblicksauflistung von allen gefundenen Daten mit kurzen Informationen (Bezeichnung, Beschreibung) möglich sein.

VM-F 5: Ein Dialog, der alle Detailinformationen zu einem Datum (Siedlung, Grab, Artefakt) anzeigt und eine Selektion einer einzelnen Information zuläßt, welche dann in den Einfüge-Dialog übernommen wird.
{Priorität 2}
Rationale: Die Detail-Informationen, welche in VM-Raum/Sonderschau eingefügt werden sollen, müssen aus der Auflistung selektiert werden können.

VM-F 6: Ein Dialog zur Bearbeitung der Sonderschau/Räume/Vitrinen (siehe XML-Daten).
{Priorität 2}
Rationale: Die ausgewählten Daten können wahlweise in verschiedener Weise in eine Ansicht eingetragen werden. Dazu stehen mehrere Templates zur Verfügung.

VM-F 7: Ein Dialog zur Positionierung der ausgewählten Daten im Template zur Anzeige des jeweiligen Bereiches (Raum, Vitrine, Artefakte). Hier können auch Texte als Schautafel-Information eingegeben und plaziert werden.
{Priorität 2}
Rationale: Die ausgewählten Daten und eingegebene Texte für einen Bereich können an verschiedenen Stellen und in verschiedener Weise (z.B. Hintergrundmuster, Rahmen, ...), je nach Temlate, positioniert werden. Die extra eingefügten Schautafel-Informationen werden nicht in der Datenbank, sonder nur in der XML-Struktur gespeichert (rasche Änderung möglich).

1.4 Nicht Funktionale Anforderungen

Basissystem

BSNF1: Der Dialog zum Erfassen und Bearbeiten eines Fundes (Artefakt,...) muß stätestens 5 Sekunden nach seinem Aufruf durch den Erfasser am bildschirm erscheinen. Dabei sollen alle Daten, die zu einem Fund gehören, zum zweck der Bearbeitung, angezeigt werden; das Speichern der Daten soll ebenfalls in diesem zeitrahmen bleiben.
{Priorität 2}

BSNF2: Der Dialog zum Erfassen und Bearbeiten von wissenschaftlichen Arbeiten, muß spätestens 5 Sekunden nach seinem Aufruf durch den Erfasser am Bildschirm erscheinen. Es werden nur die für die Bearbeitung nötigen Felder angezeigt.
{Priorität 2}

BSNF3: Der Dialog zum Suchen nach aräologischen Daten muß spätestens 5 Sekunden nach seinem Aufruf durch den Erfasser am Bildschirm erscheinen. Das Ergebnis einer Suchabfrage darf nicht länger als 30 Sekunden dauern.
{Priorität 1}

BSNF4: Der Dialog zum Erstellen und Bearbeiten von Sonderschauen (für das VM), muß spätestens 5 Sekunden nach dem aufruf durch den Erfasser am Bildschirm erscheinen. Das Speichern soll ebenfalls nicht länger als 5 Sekunden in Anspruch nehmen.
{Priorität 2}

Datenbasis

DBMNF1: Das Datenbanksystem sollte so aufgebaut sein, daß ein Austausch der RDBMS mit geringem Anpassungen möglich ist.
{Priorität 3}

DBMNF2: Das Datenbanksystem muß möglichekeiten zur Skalierung bieten. D.h. wenn es erforderlich ist, soll die DB- Last auf mehrere Maschinen verteilt werden können; mit geringen Anpassungsarbeiten.
{Priorität 2}

XML Daten

XML-NF 1: Templates, die eine Grundstruktur für VM/Sonderschau/Raum/Vitrine/Artefakt anbietet. Diese Templates sind in XML-Form gespeichert und werden für den jeweiligen Bereich angepaßt.
{Priorität 2}
Rationale: Damit die Gestalltung von den einzlenen Bereichen flexibel ist und dabei doch nicht ins Uferlose ausartet, werden mehrere Grundformen für die Gestaltung (einschließlich einer Mengenmäßingen Einschränkung der Teilobjekte) angeboten.

XML-NF 2: Nach Fertigstellung eines Raumes im VM oder einer Sonderschau werden diese durch ein eigenes Verfahren geparst und zu einer XML-Struktur aufgebaut, welche Infos über Aufbau und Daten einschließt.
{Priorität 3}
Rationale: Durch die Erzeugung einer XML-Struktur ist eine einfache und rasche Erzeugung einer HTML-Darstellung für den Museumsbesucher möglich. Weiter sind Änderungen der Schautafel-Informationen leicht korrigierbar, da die Informationen nur in den XML-Dateien abgelegt sind. Es ist schneller und einfacher, als die Infos in der Datenbank abzulegen.

XML-NF 3: Eine Aktualisierung von textlichen Informationen, die in der XML-Struktur gespeichert sind, dauert nicht länger als ein bis zwei Minuten.
{Priorität 3}
Rationale: Da eine Aktualisierung des VM/einer Sonderschau täglich einmal zu erwarten ist, darf der Aktualisierungsprozess nicht länger dauern, da er sonst auf Ablehnung stoßen würde.

XML-NF 4: Eine Erstellung eines VM aus Daten, die in einer XML-Struktur gespeichert sind, dauert nicht länger als fünf Minuten.
{Priorität 3}
Rationale: Da die Erstellung des VM alle halben Jahre geplant ist, kann ihr Vorgang mehrere Minuten beanspruchen. Sie sollte aber nicht länger dauern, da sie sonst auf Ablehnung stoßen würde.

XML-NF 5: Eine Erstellung eine Sonderschau aus Daten, die in einer XML-Struktur gespeichert sind, dauert nicht länger als zwei Minuten.
{Priorität 3}
Rationale: Da die Erstellung einer Sonderschau nicht täglich erfolgt, kann die Erzeugung aus einer XML-Struktur mehrere Minuten beanspruchen. Sie sollte aber nicht länger dauern, da sie sonst auf Ablehnung stoßen würde.

1.5 Usecase Dokumentation

1.5.1 System Beschreibung

Interaktives Museum

Systemname

Archäologisches Managementsystem

System-Level Use Case Diagram

Architektur Diagramm

Teilsystem Beschreibung

Actors

Verwalter: Ein Direktor hat diesselben Rechte auf ein Artefakt, wie sein Erfasser.Außerdem kann er Erfasser bearbeiten und das virtuelle Museum warten. Angehörige dieser Gruppe haben uneingeschränkte Rechte auf alle Einträge der Datenbank.

Erfasser: Dem Erfasser ist es erlaubt Artefakte einzutragen, seine Artefakte zu bearbeiten und zu löschen. Außerdem kann er die Rechtevergabe an seinen Artefakten übernehmen und eine Sonderschau erstellen. Zusätzlich kann er Interpretationen zu allen Artefakten erstellen.

Anonymous: Als anonym gilt der normale Internetbenutzer. Er hat nur das Recht zu Suchen (auf allen in der Datenbank freigegebenen Artefakten).

Die hier eingeführten Flow of Events sind Grob zu betrachten, erst nach mehreren Iterationen werden sie dann ausführlicher, am besten mit dem Activity Diagram.

1.5.2 Detailierte Usecase Beschreibung: RechteWarten

Der Use- Case *RechteWarten* spielt bei der Systemsicherheit eine zentrale Rolle. Es werden den einzelnen Benutzern (Erfassern) gerade soviel Rechte zugeteilt, wie sie für die Durchführung ihrer Arbeit benötigen. Zuviele Rechte auf zu viele unterschiedliche Benutzer verteilt, gefährden die Konsistenz der Einträge und somit die Quallität des Datenbestandes.

actors

Erfasser, Verwalter

Priorität

1

Status

Vorbedingungen

Der Erfasser (User) muß die Rechte zum Ändern der Benutzerrechte verfügen.

Nachbedingung

Konsistenz der Benutzerrechte.

Extension Points

"Used" Use Cases

Flow of Events

Der Großteil bei der Änderung der Benutzerrechte wird vom System übernommen. Der *Flow of Events* ist somit relativ kurz. In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System und überprüfung der Passwort und Rechte.
2. Passwort falsch oder Rechte nicht vorhanden dann gehe zu 1.
3. Es wird eine Seite generiert,die die Möglichkeiten des Erfassers anhand seine Rechte auflistet.Der Verwalter(Direktor) hat uneingeschränkte Rechte.
4. Abmeldung vom System.

ER- Modell

Abbildung 1.2 zeigt das ER- Modell der Benutzer- und Rechteverwaltung. Jede Person wird mit Titel, Vorname und Nachname eingetragen. Jeder Person kann eine oder mehrer Adressen besitzen. Das Attribut *Location* in der Tabelle *Address* gibt das *Institut* an, auf die sich die Adresse bezieht. Der *User* ist von *Person* abgeleitet (ein User ist eine Person).

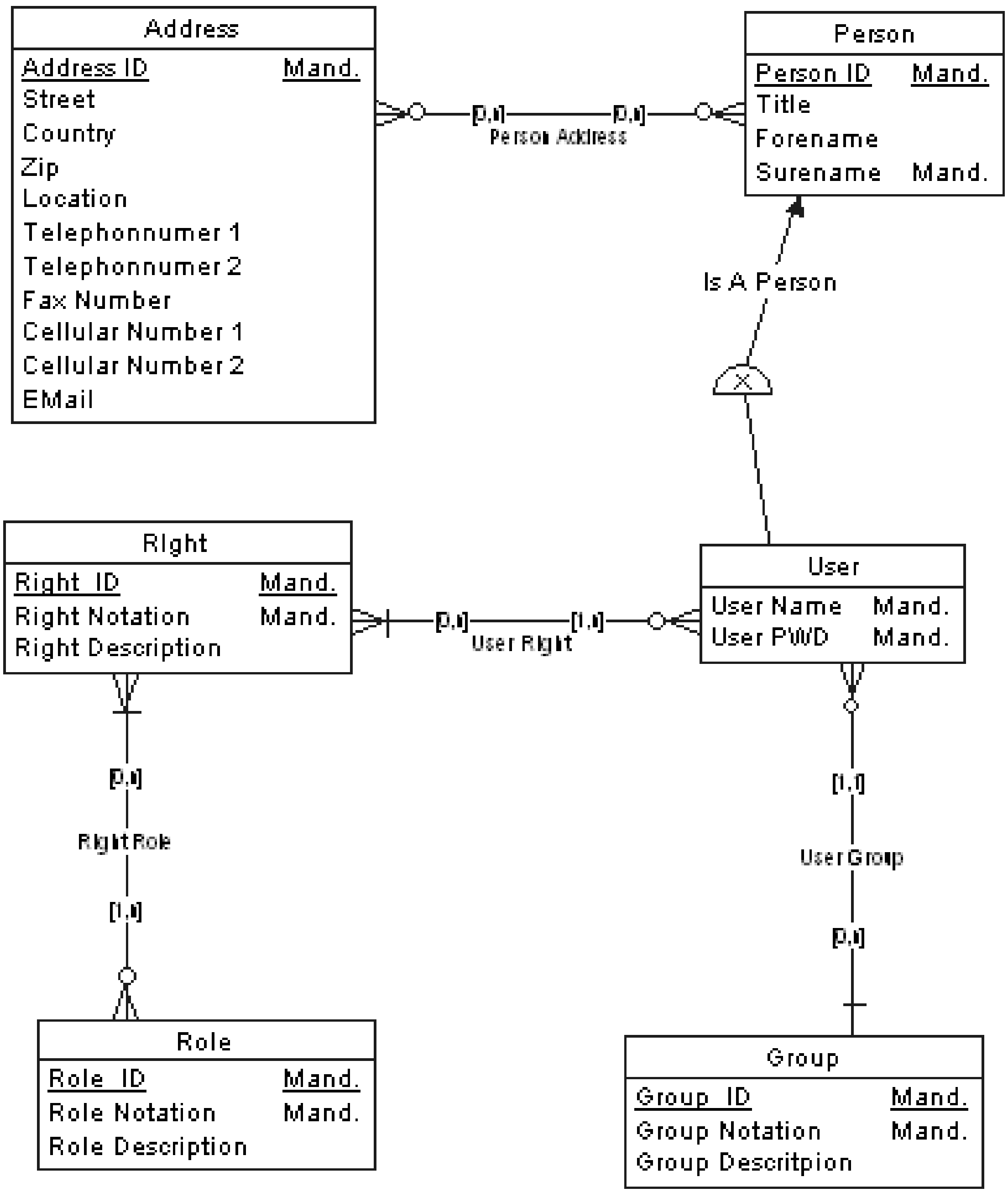


Abbildung 1.2: Das Diagramm zeigt das ER- Modell der Benutzer- und Rechteverwaltung.

Beim *User* sind nur sein Login- Name und das dazugehörige Passwort [3] von interesse. Jeder Benutzer hat mehrere Rechte, zumindest aber eines. Um die *Rights* leichter verwalten zu können, können sie in *Roles* eingeteilt werden. Eine Role besteht aus ein oder mehreren Rechten. Um den Umfang (Scope) der Benutzerrechte festzulegen, werden die User in Benutztergruppen eingeteilt - Verwalter, Ausgrabungsleiter, Assistent. Ein User ist genau einer Benutzergruppe zugeordnet.

[3] das Passwort muß verschlüsselt in der Datenbank abgelegt werden - eventuell Hashcode des Passwortes

Activity Diagram

Das Warten der Rechte beschränkt sich auf die Definition von Benutzerrollen. Abbildung 1.3 zeigt das dazugehörige Aktivitätsdiagramm.

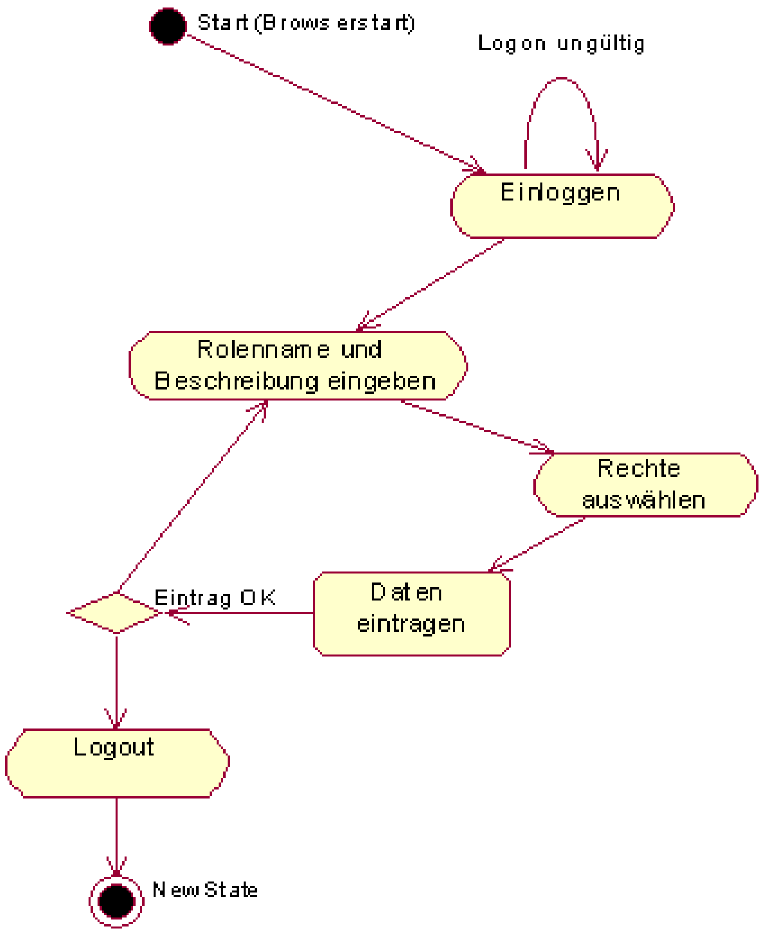


Abbildung 1.3: Aktivitätsdiagramm für die Administrierung von Benutzerrollen

User Interface

Alle verwendeten Rechte sind atomar, d.h. es können keine neuen Rechte hinzugefügt werden. Alle Rechte müssen vorher definiert, und in die Datenbank eingetragen werden. Da diese Rechte nachträglich nicht mehr geändert werden dürfen, existiert auch hierfür keine Interface. Es könne nur die Roles [4] bearbeitet werden.

Bei der Bearbeitung der Benutzerrechte wurde dacon ausgegangen, daß sich der Verwalter ordnungsgemäß angemeldet hat.

Bild 1.4 zeigt die Form für das Erstellen einer neuen Role. Die Auswahl der dazugehörigen Rechte erfolgt über die Form *Auswählen der Benutzerrechte*, Bild 1.5.

[4] Eine Rolle ist eine Zusammenfassung von mehreren Rechten

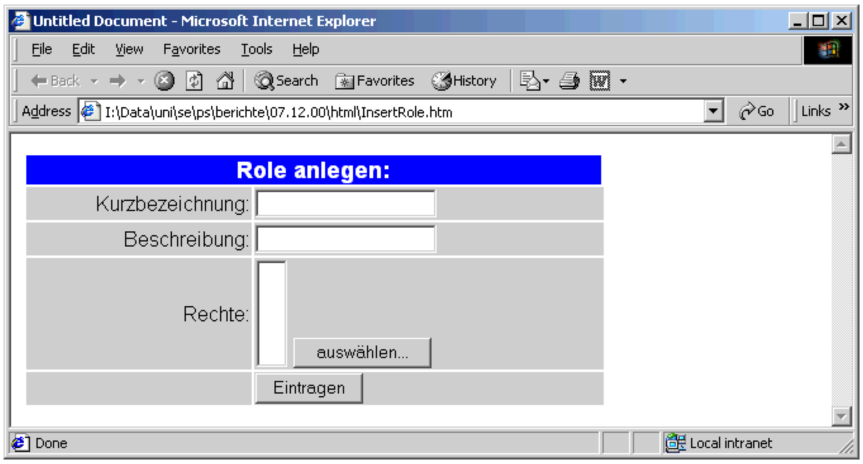


Abbildung 1.4: Screenshot der Form ”Erstellen einer neuen Role”

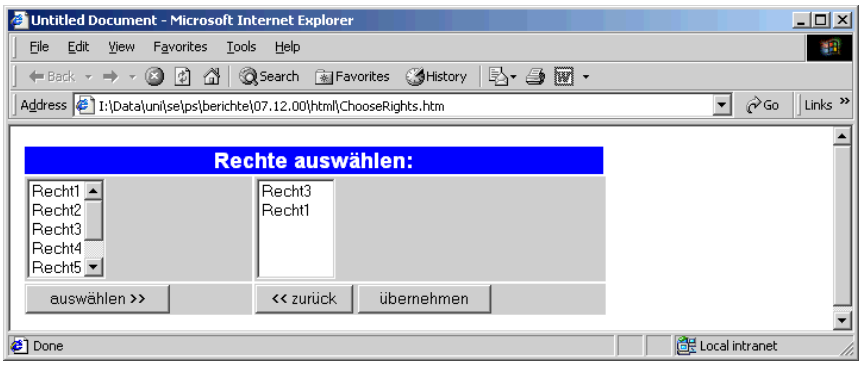


Abbildung 1.5: Screenshot der Form ”Auswählen der Benuzterrechte”

Secondary Scenarios

Sequence Diagrams

View of Participating Classes

Other Artifacts

Other Requirements

1.5.3 Detailierte Usecase Beschreibung: Erfasser warten

Erfasser warten:

Der Verwalter kann Erfasser hinzufügen, verändern und eventuell entfernen. Zum Hinzufügen oder Ändern muß er alle erforderlichen Daten eintragen. Beim Entfernen wird darauf geachtet, daß der Erfasser keine Fremdverweise besitzt. Nur dann wird er gelöscht. Ein Erfasser der jedoch ein Artefakt gefunden hat wird nach wie vor als Finder dieses Artefaktes aufscheinen. Er wird lediglich aus der Login-Berechtigung verlieren(d.h. Löschen aus dem UserTable aber nicht aus dem PersonTable).

actors

Verwalter

Priorität

1

Status

Vorbedingungen

Der Verwalter muß eingeloggt sein.

Nachbedingung

Einträge erfolgen ganz oder gar nicht.Konsistenz der DB d.h. die geänderte Daten müssen rechtig gespeichert.

Extension Points

"Used" Use Cases

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System und überprüfung der Passwort und Rechte.
2. Passwort falsch oder Rechte nicht vorhanden dann gehe zu 1.
3. Es wird eine Seite generiert,die die Möglichkeiten des Erfassers anhand seine Rechte auflistet.Der Verwalter(Direktor hat uneingäschränkte Rechte).
4. hier kann gesucht werden nach einer Personn.D.h.DB abfragen.
5. Ist die Personn gefunden, dann werden ihre Daten geändert,bearbeitet oder gelöscht,es kann aber auch sein das die Personn Acount aus dem DB gelöscht werden.
6. Einen neuen Acount erstellen.Um hierher zu gelangen wird der Verwalter die zwei vorherigen Schritte nicht brauchen.
7. Bestätigen und warten auf ein Antwort vom System.
8. Abmeldung vom System.

Abbildung 1.6: Screenshot der Form "Person einfügen".

Activity Diagram

User Interface

Da die Verwaltung der Benutzer über das Internet möglich sein muß, wird dieser Teil über html- bzw. jsp- Seiten realisiert. Eine Reihe von *Screenshots* sollen einen ersten Eindruck vermitteln. Die ersten Bilder zeigen das Eintragen einer neuen *Person* in die Datenbank. Beim Änderm und Löschen werden ähnliche Formen verwendet.

Bild 1.6 zeigt die Form für das Eintragen einer neuen Person - laut ER- Diagramm (Abbildung 1.2) muß, bevor ein neuer User (Erfasser) angelegt werden kann eine entsprechender Personeneintrag erzeugt werden.

Abbildung 1.7: Screeshot der Form "Adressen auswählen"

Die Auswahl der einen oder mehreren Adressen zu einer Person, zeigt Bild 1.7.
Ist eine Adresse noch nicht in der Liste vorhanden, was wahrscheinlich ist, kann von hier aus ein neuer Eintrag erstellt werden - Bild 1.8.

Wurde eine Person angelegt, so kann ein Benutzer (User) erzeugt werden. Diesem Benutzer können / müssen Rechte bzw. Roles zugewiesen werden. Weiters wird er einer Benutzergruppe zugeordnet.

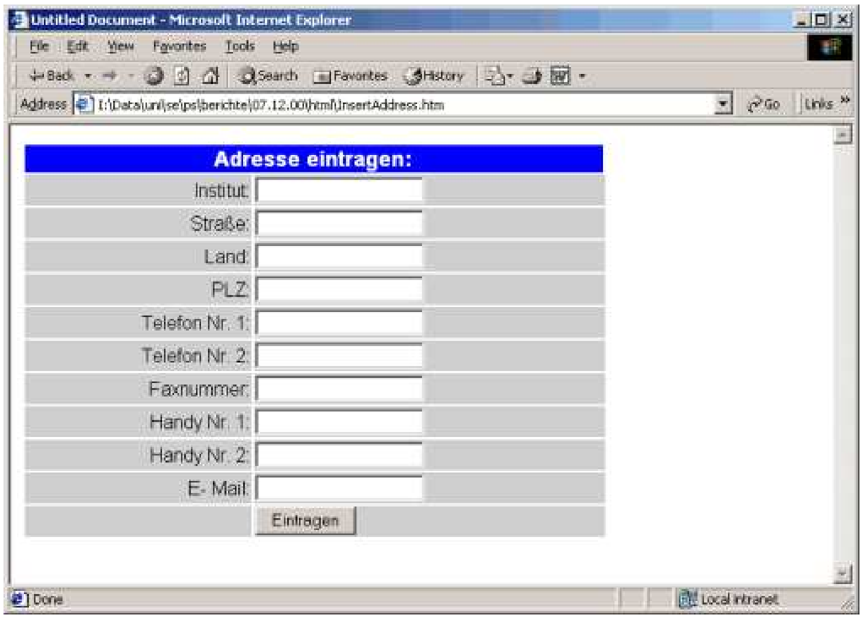


Abbildung 1.8: Screenshot der Form "Adresse eintragen"

Bild 1.9 zeigt die Form, mit der ein neuer Benutzer angeleget wird. Neben den Feldern für Username und Passwort müssen auch noch die zugehörige Person (Bild 1.10), Role (Bild 1.11) und Gruppe (Bild 1.12) ausgewählt werden.
Für das Ändern oder Löschen eines Benutzers werden ähnliche Formen verwendet.

Secondary Scenarios

Sequence Diagrams

Subusecases

- Erfasser suchen

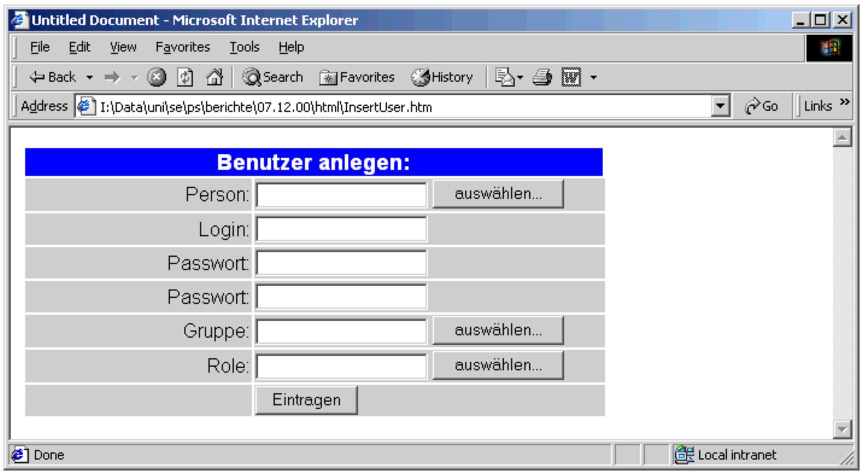


Abbildung 1.9: Screenshot der Form "Anlegen eines neuen Benutzers"

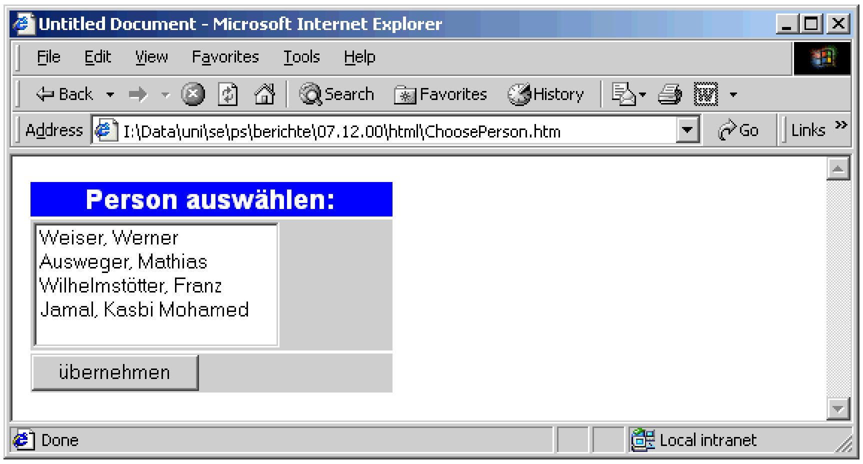


Abbildung 1.10: Screenshot der Form "Auswählen einer Person"

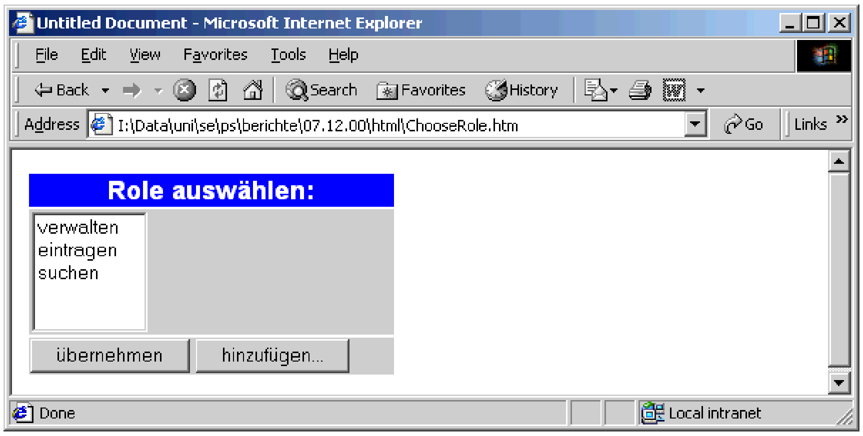


Abbildung 1.11: Screenshot der Form "Auswählen einer Role"

Abbildung 1.12: Screenshot der Form "Auswählen einer Benutzergruppe"

View of Participating Classes

Other Artifacts

Other Requirements

1.5.4 Detailierte Usecase Beschreibung: Artefakte suchen:

Artefakte suchen

Es wird hier nach vorhandenen Artefakten gesucht und dessen Daten Angezeigt. Der Internet-Besucher gelangt ausgehend vom virtuellen Museum zur Artefaktsuche, ihm werden nicht alle Daten angezeigt. Der Erfasser und der Verwalter können uneingeschrängte Suche starten und ihnen werden alle vorhandenen Daten über das Artefakt angezeigt.

actors

Internet-Benutzer, Erfasser

Priorität

1

Status

Filled

Vorbedingungen

Mindestens ein Artefakt ist in der DB vorhanden. Ein Erfasser oder der Verwalter muß eingeloggt sein, um die größere Such-Kriterienauswahl zu erhalten. Die Benutzerrechte müssen bekannt sein. Für den Internet-Besucher gelten diese Bedingungen nicht.

Nachbedingungen

Anzeige des Suchergebnisses oder der Nachricht für fehlgeschlagene Suche.

Extension Points

Error Handling (bzgl. Verbindung zum Server, Datenbankzugriff).

"Used" Use Cases

Artefaktrechte prüfen
Benutzerrechte prüfen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. if (Benutzer ist Erfasser)
   (a) Benutzer (Erfasser) durch Eingabe von Login-Name und Passwort anmelden.
   (b) Prüfung der Eingabedaten und der Benutzerrechte auf Erfasser.
   (c) if (kein gültiger Erfasser oder Verwalter)
      i. Erneut zum Anmeldefenster

   ii. Anmeldung als Normaler Benutzer
   iii. System verlassen

2. while (Benutzer sucht nach Artefakten)

   (a) Es wird eine Seite generiert,die die Möglichkeiten des Benutzers (Verwalter, Erfasser oder Internet-Benutzer)anhand seine Rechte auflistet.
   (b) Abhängig vom Art des Benutzers werden die Suchoptionen angezeigt.
   (c) Eingabe der Suchkriterien.
   (d) Die Suche wird gestartet.
   (e) Anzeige aller Suchergebnisse in Kurzbeschreibungsansicht (Bsp: Bild, Name, Kurzbeschreibung).
   (f) while (Suchergebnisse werden für Detail-Ansicht ausgewählt)
      i. Detailergebnis wird generiert und in neuer Darstellung angezeigt. Die angezeigten Ergebnisse sind abhängig von der Art des Benutzers (Erfasser sieht alles, Internet-Benutzer nur die freigegebenen Informationen - *Artefakt-Rechte*).

3. Abmelden vom System, nur für den Erfasser und Verwalter.

Activity Diagram

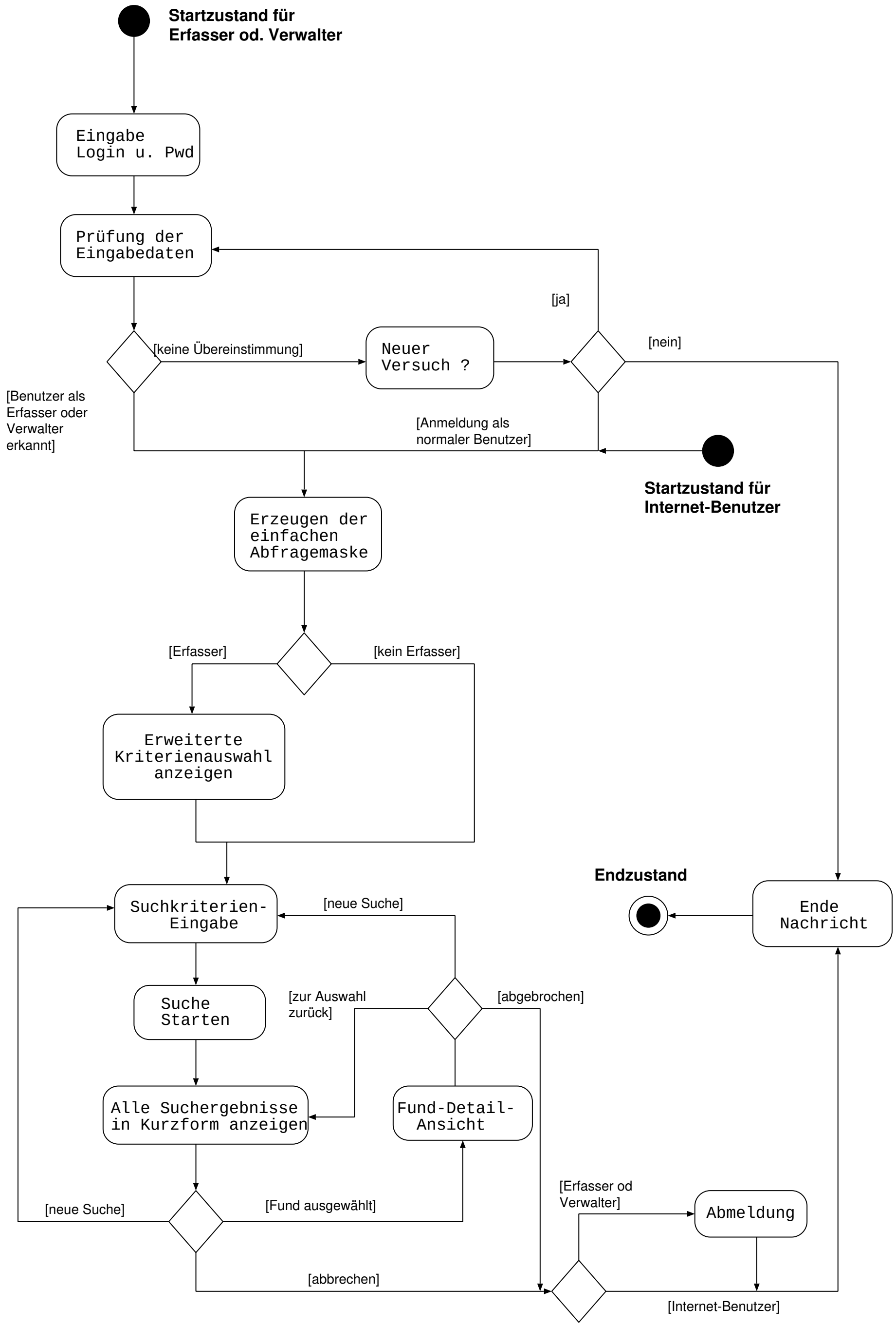


Abbildung 1.13: Aktivity-Diagram: Artefakte Suchen

User Interface

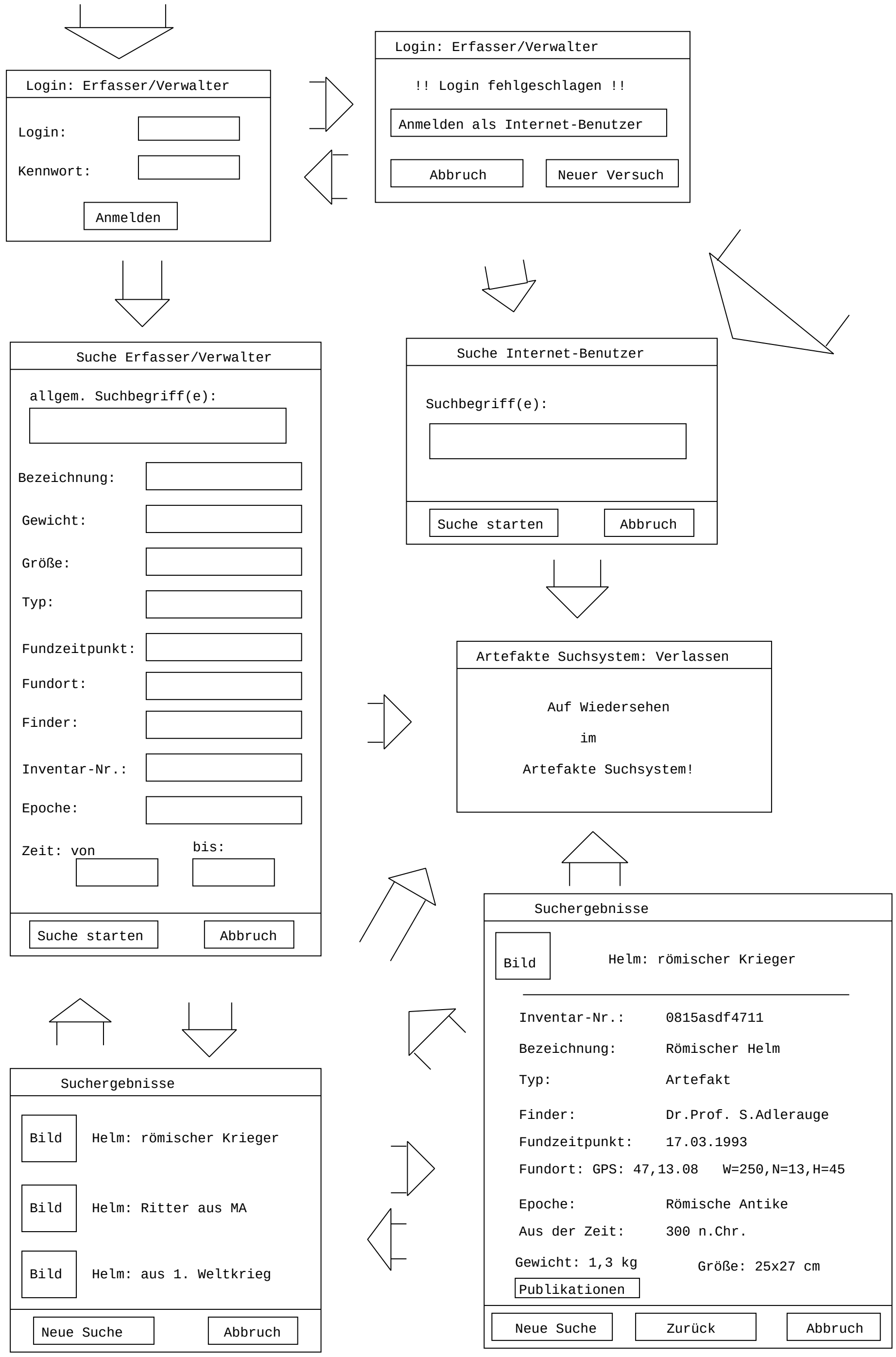


Abbildung 1.14: User Interface Diagramms:Artefakte Suchen

Secondary Scenario

1. Es besteht jederzeit die Möglichkeit, über eine *ZURÜCK* Funktion eine Hierarchie-Ebene in der Suche zurückzukehren.

2. Es gitb ebenso jederzeit eine *ABBRUCH* Funktion, über die man zu einer *Abschiedsmaske* kommt und das System verlassen kann (Erfasser und Verwalter werden zuvor abgemeldet).

3. Passiert ein Fehler bei der Kommunikation mit dem Datenbankserver, oder ist die Datenbank einfach nichtmehr erreichtbar, so wird man beim nächsten Versuch automatisch auf eine Fehlerseite weitergeleitet.

Die hier eingeführten Flow of Events sind Grob zu betrachten, erst nach mehreren Iterationen werden sie dann ausführlicher, am besten mit dem Activity Diagram.

1.5.5 Detailierte Usecase Beschreibung: Artefakte eintragen

Artefakte eintragen:

Hier werden die Grunddaten eines Artefaktes in die Datenbank aufgenommen. Dabei soll es möglich sein, Ausgrabungen und Fundorte auszuwählen bzw. zu erstellen. Außerdem soll die Möglichkeit einer Freigabe bzw. Sperre eines Artefaktes gegeben sein.

actors

Erfasser, Verwalter

Priorität

1

Status

Facade

Vorbedingungen

Der Erfasser muß eingeloggt sein. Der Erfasser muß die erforderlichen Rechte auf diese Ausgrabung besitzen. Der Fundort des Artefaktes muß bereits vorhanden sein.

Nachbedingung

Einträge erfolgen ganz oder gar nicht. Ebenso die Freigabe eines Artefaktes.

Extension Points

"Used" Use Cases

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System und Überprüfung der Passwort und Rechte.
2. if (Passwort falsch || Rechte nicht vorhanden)
   (a) gehe zu 1.
3. else
4. Der Erfasser wird aufgefordert eine Ausgrabung (die er mitbetreut) auszuwählen.
5. if (keine Ausgrabung vorhanden || Rechte nicht vorhanden)

   (a) Anlegen einer neuen Ausgrabung (damit wird der Erfasser zum Ausgrabungsleiter) || Organisieren der Rechte (können vom Verwalter oder vom Ausgrabungsleiter erteilt werden).
   (b) gehe zu 4.

6. else
7. Der Erfasser wird aufgefordert einen Fundort auszuwählen.
8. if (Fundort nicht vorhanden)
   (a) Anlegen eines neuen Fundortes.
   (b) gehe zu 7.
9. else
10. Das System zeigt die Eingabemaske zur Pflege der Attribute eines Artefaktes.
11. Der Erfasser wird aufgefordert die Artefaktdaten einzutragen.
12. Der Erfasser wird aufgefordert die Freigabe des Artefaktes festzulegen.
13. Der Erfasser wird aufgefordert das anzuzeigende layout des Artefaktes festzulegen (sonst default).
14. if (Erfasser bricht Verarbeitung ab)
    (a) Das System verlässt die Eingabemaske. Der use case ist beendet.
15. if (Erfasser sichert Artefakt)
    (a) Das System speichert das Artefakt in der Datenbank und verlässt die Eingabemaske. Der use case ist beendet.
16. Abmeldung vom System.

Activity Diagram

User Interface

Secondary Scenarios

Sequence Diagrams

Sub-Usecases

- Ausgrabung anlegen
- Ausgrabungsrechte erteilen
- Fundort erstellen
- Layout erstellen
- Artefaktrechte erteilen

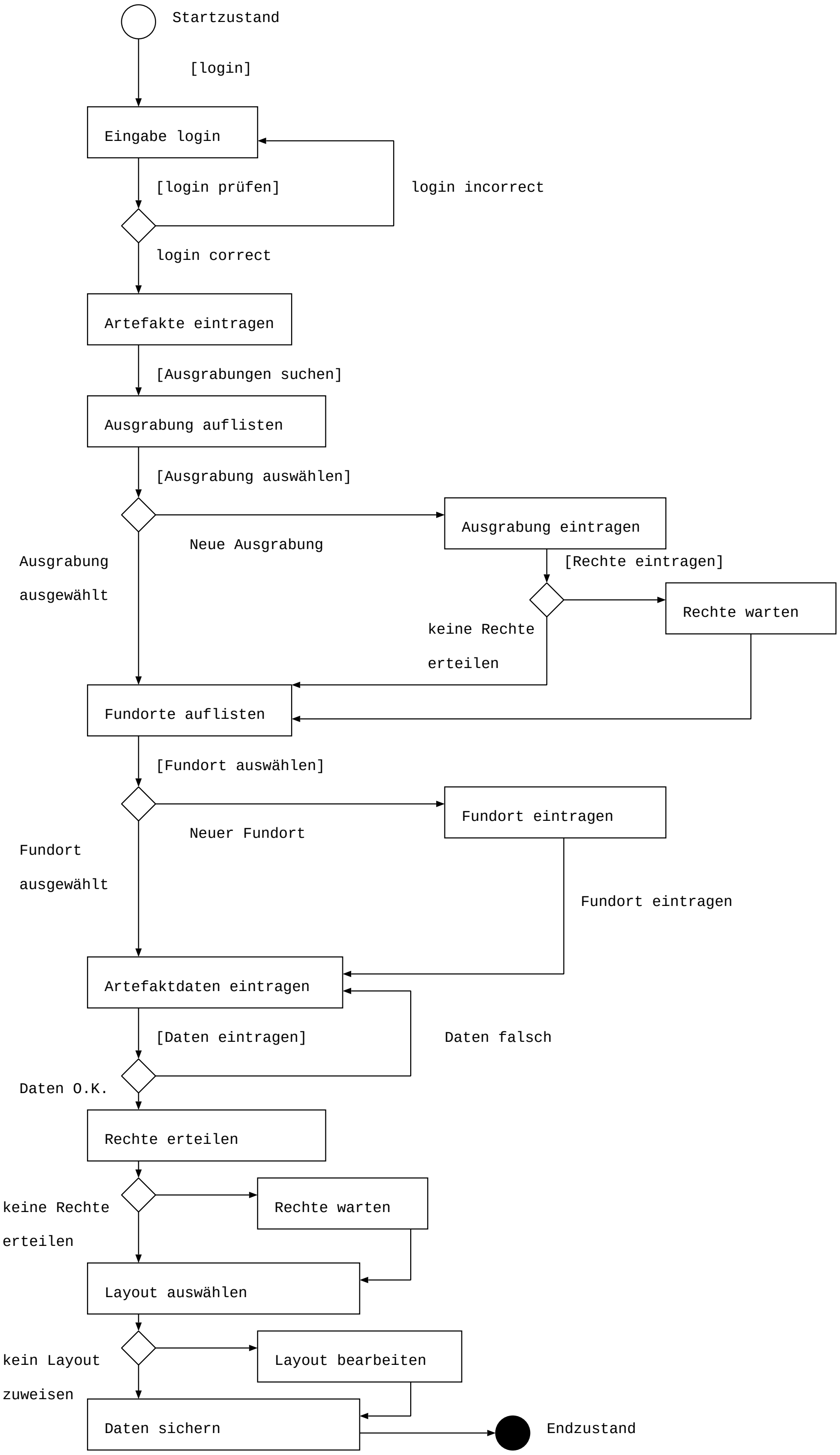


Abbildung 1.15: Artefakt eintragen - Activity Diagramm

View of Participating Classes

Other Artifacts

Other Requirements

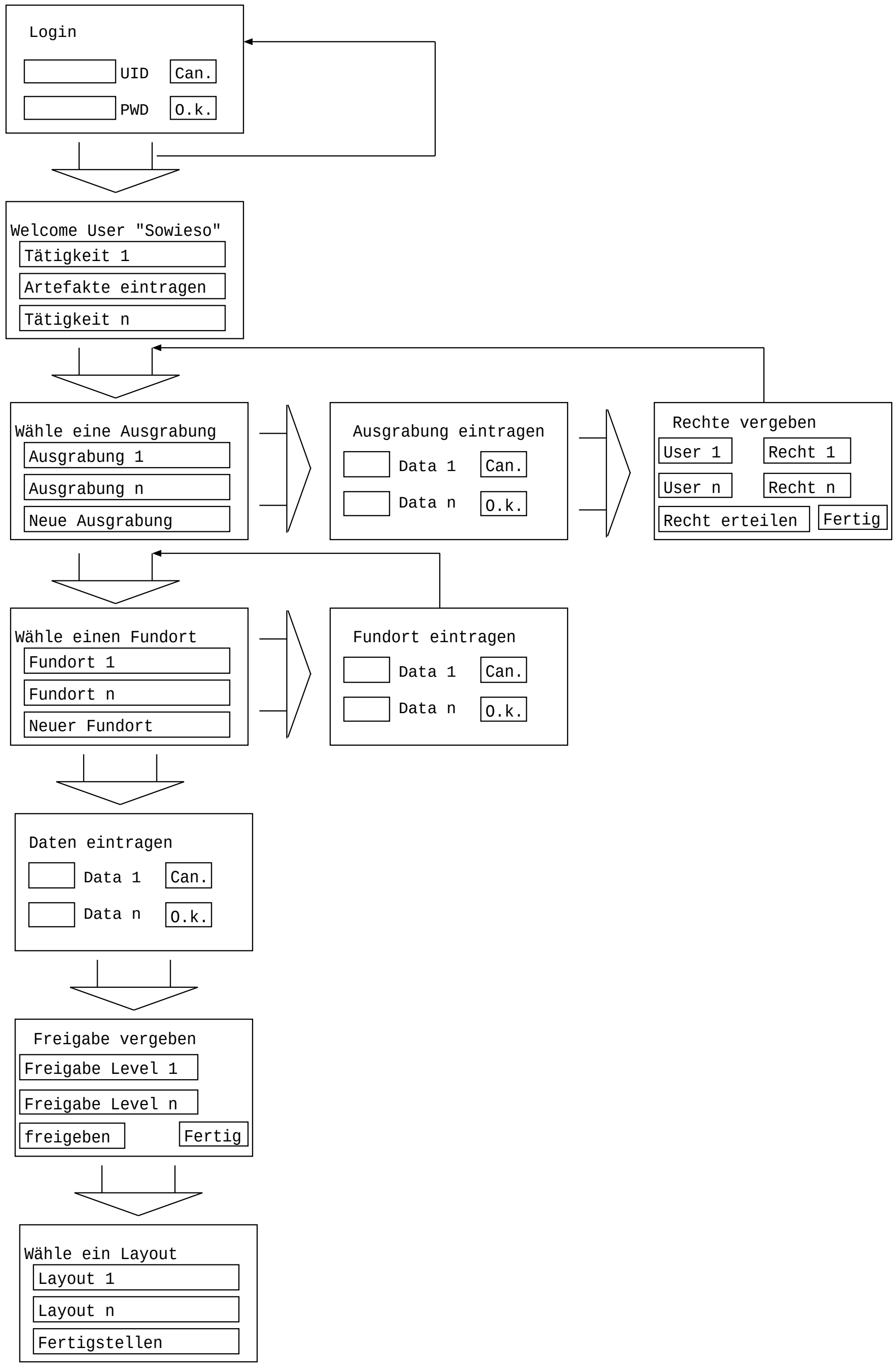


Abbildung 1.16: Artefakt eintragen - User Interface

1.5.6 Detailierte Usecase Beschreibung: Artefakte bearbeiten

Artefakte bearbeiten:

Es werden Daten eines vorhandenen Artefaktes neu bearbeitet,gelöscht oder geändert.Es soll hier auch möglich sein die Freigabe oder Sperre eines Artefaktes durchzuführen. Außerdem soll das Hinzufügen bzw. bearbeiten oder löschen einer Interpretation des Artefaktes ermöglicht werden.

actors

Erfasser, Verwalter

Priorität

1

Status

Facade

Vorbedingungen

Der Erfasser muß eingeloggt sein. Der Erfasser muß die erforderlichen Rechte auf dieses Artefakt besitzen.

Nachbedingung

Einträge erfolgen ganz oder gar nicht. Ebenso die Freigabe eines Artefaktes.

Extension Points

"Used" Use Cases

Usecases

- Artefakt suchen

Subusecases

- Layout erstellen
- Artefaktrechte erteilen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System und Überprüfung der Passwort und Rechte.
2. if (Passwort falsch || Rechte nicht vorhanden)
   (a) gehe zu 1.
3. else
4. Der Erfasser kann ein Artefakt suchen.
5. if (Artefakt nicht vorhanden || Rechte nicht vorhanden)
   (a) gehe zu 4.
6. else

7. Das System zeigt die Eingabemaske zur Pflege der Attribute eines Artefaktes.
8. Der Erfasser kann die Artefaktdaten ändern.
9. Der Erfasser kann die Freigabe des Artefaktes ändern.
10. Der Erfasser kann das anzuzeigende layout des Artefaktes ändern.
11. Der Erfasser kann das Artefakt betreffende Interpretationen erstellen.
12. Der Erfasser kann das Artefakt betreffende Interpretationen (auch solche, die er nicht erstellt hat) löschen.
13. if (Erfasser bricht Verarbeitung ab)
    (a) Das System verlässt die Eingabemaske. Der use case ist beendet.
14. if (Erfasser sichert Artefakt)
    (a) Das System speichert das Artefakt in der Datenbank und verlässt die Eingabemaske. Der use case ist beendet.
15. Abmeldung vom System.

Activity Diagram

User Interface

Secondary Scenarios

Sequence Diagrams

View of Participating Classes

Other Artifacts

Other Requirements

1.5.7 Detailierte Usecase Beschreibung: Artefakte löschen

Artefakte löschen:

Hier hat der Verwalter und der Erfasser(Erfasser nur wenn er Rechte dazu hat) die Möglichkeit ein vorhandenes Artefakt zu löschen.

actors

Verwalter, Erfasser

Priorität

2

Status

Facade

Vorbedingungen

Der Verwalter muß eingeloggt sein. Das Artefakt darf über keine Fremdverweise verfügen.

Nachbedingung

Artefakt nicht mehr in der DB vorhanden.

Extension Points

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System und Überprüfung der Passwort und Rechte.
2. if (Passwort falsch || Rechte nicht vorhanden)
   (a) gehe zu 1.
3. else
4. Der Erfasser kann ein Artefakt suchen.
5. if (Artefakt nicht vorhanden || Rechte nicht vorhanden)
   (a) gehe zu 4.
6. else
7. Das System zeigt die Eingabemaske zur Pflege der Attribute eines Artefaktes.
8. Der Erfasser kann das Artefakt löschen.
9. if (Erfasser bricht Verarbeitung ab)

(a) Das System verlässt die Eingabemaske. Der use case ist beendet.

10. if (Erfasser sichert Artefakt)

    (a) Das System speichert das Artefakt in der Datenbank und verlässt die Eingabemaske. Der use case ist beendet.

11. Abmeldung vom System.

"Used" Use Cases

Usecases

- Artefakt suchen

Activity Diagram

User Interface

Secondary Scenarios

Sequence Diagrams

Subordinate Use Cases

View of Participating Classes

Other Artifacts

Other Requirements

1.5.8 Detailierte Usecase Beschreibung: Aufbau des virtuellen Museums:

Aufbau des virtuellen Museums:

Jeder Erfasser hat das Recht virtuelle Museen(Sonderschauen) zu erstellen.
Der Erfasser kann die Sonderschau jederzeit als Katalog ausdrucken.

actors

- Direktor
- Erfasser

Priorität

3

Status

Filled
Erste konkrete Ablaufbearbeitung nach überblicksmäßiger Szenario-Beschreibung.

Vorbedingungen

- Der Erfasser muß am System angemeldet sein.
- Es muß mindestens ein Template zur Gestaltung der verschiedenen Teilbereiche existieren.

Nachbedingungen

- Sonderschau abspeichern.

Extension Points

- Fehlerbehandlung
- Abbruch

"Used" Use Cases

Usecases

- Artefakt suchen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Der Use case beginnt mit der Selektion (VM erstellen)
2. Der Erfasser logt sich am System ein
3. IF (System findet keine Berechtigung)
   1) Das System verlässt die Eingabemaske, und fordert die Berechtigung
   2) Der User gibt die Eingabe od. bricht ab (Goto Sicherung)
4. ELSE Sonderschau registrieren
5. While(Sonderschau bearbeiten)
   (a) IF (true)
      i. Auflisten der Sonderschau
      ii. Template auswaelen
      iii. individualles Layout auswählen
      iv. Beschreibung
      v. Goto Vitrine bearbeiten
   (b) Elseif
      i. VM Gestaltung
      ii. if (keine Eingabe bestaetigen)
      iii. Goto Sicherung
      iv. elseif
      v. Raueme gestalten
      vi. Goto Vitrine bearbeitung
   (c) Vitrine Bearbeitung:
      i. Vitrine einfuegen
      ii. Artefakte suchen
         A. if (keine Artefakt gefunden)
         B. weitere suchen
         C. elseif
         D. Template auflisten
         E. Artefakt einfuegen
         F. schautafel-Info gestalten
         G. weitere Bearbeitung
         H. if (true)
         I. Goto Vitrine Bearbeitung
         J. elseif
         K. Stockwerk bearbeiten

L. Raeum im Sockwerk abspeichern

M. if (abspeichern=true)
   - if (weitere Bea rbeitung=true)
   - Goto Vitine Bearbeitung)
   - elseif Exit (Goto Sicherung)

6. Sicherung :
   (a) abspeichern des aktuellen Standes
   (b) Exit

7. End while (Sonderschau Bearbeitung)

8. Logout

9. Endzustand

Activity Diagram

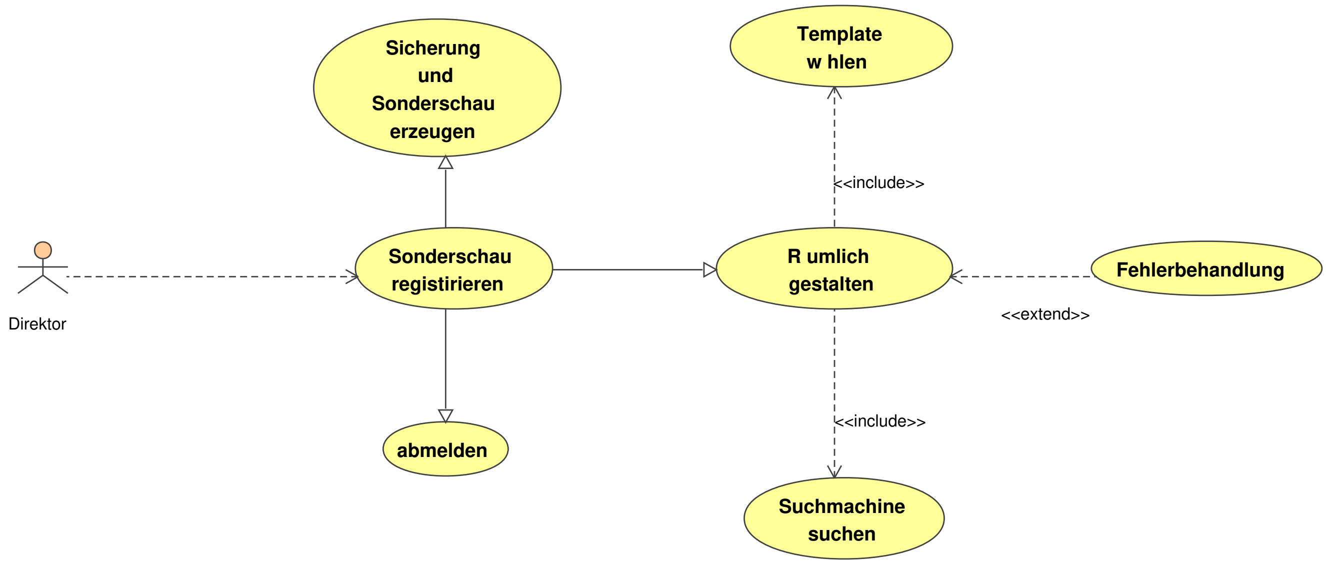


Abbildung 1.17: Sonderschau erstellen - Use Case Diagramm

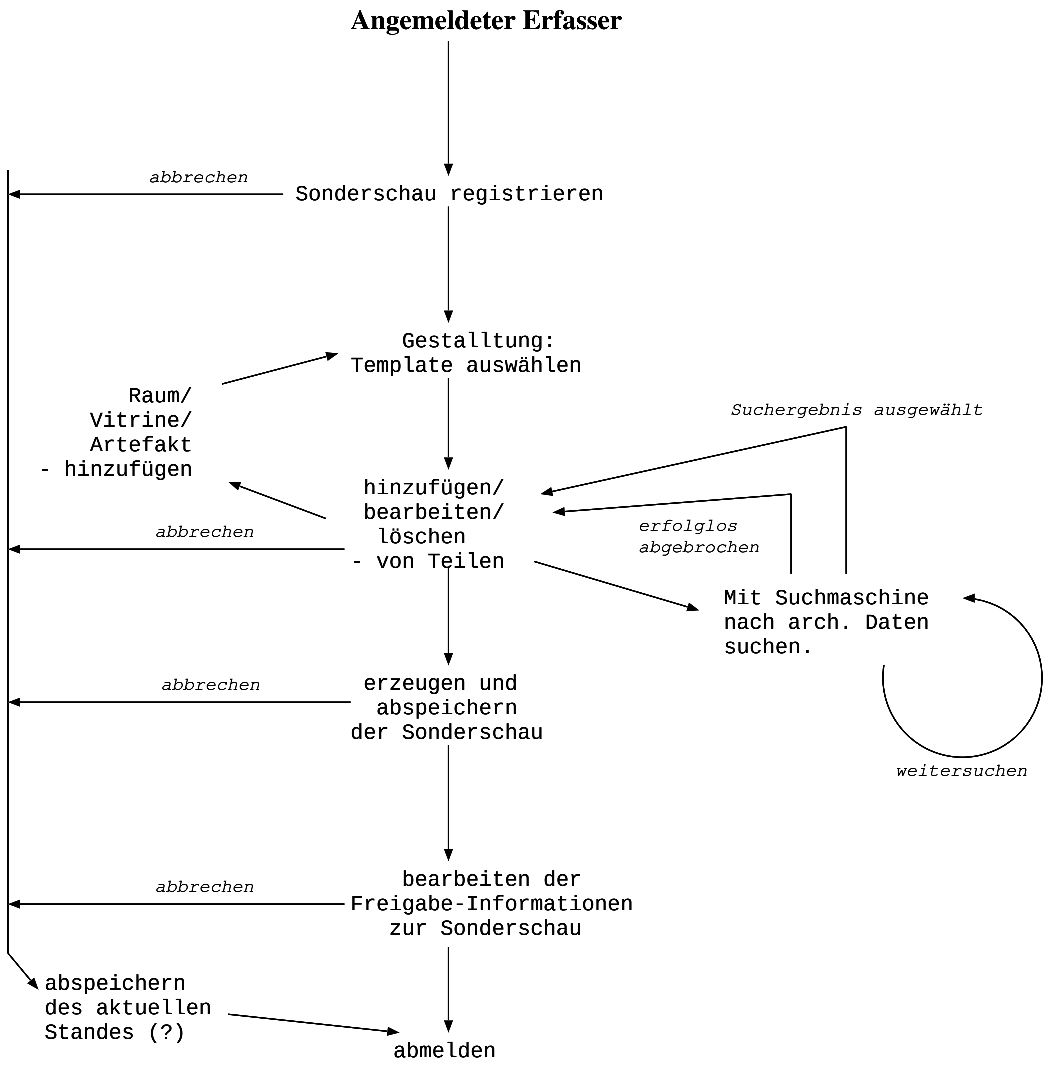
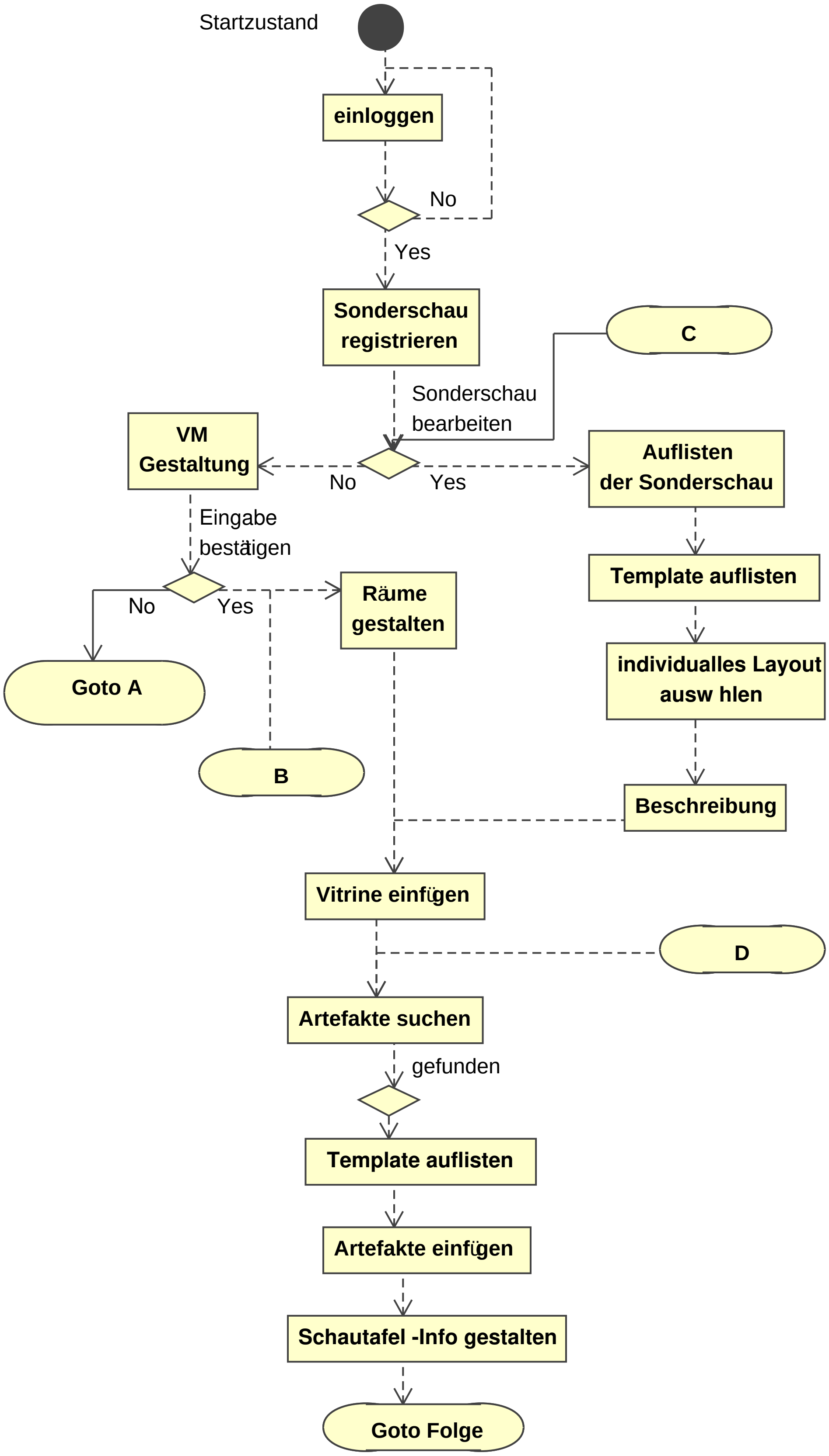


Abbildung 1.18: Sonderschau erstellen - Activity-Diagramm

Usecase Diagram



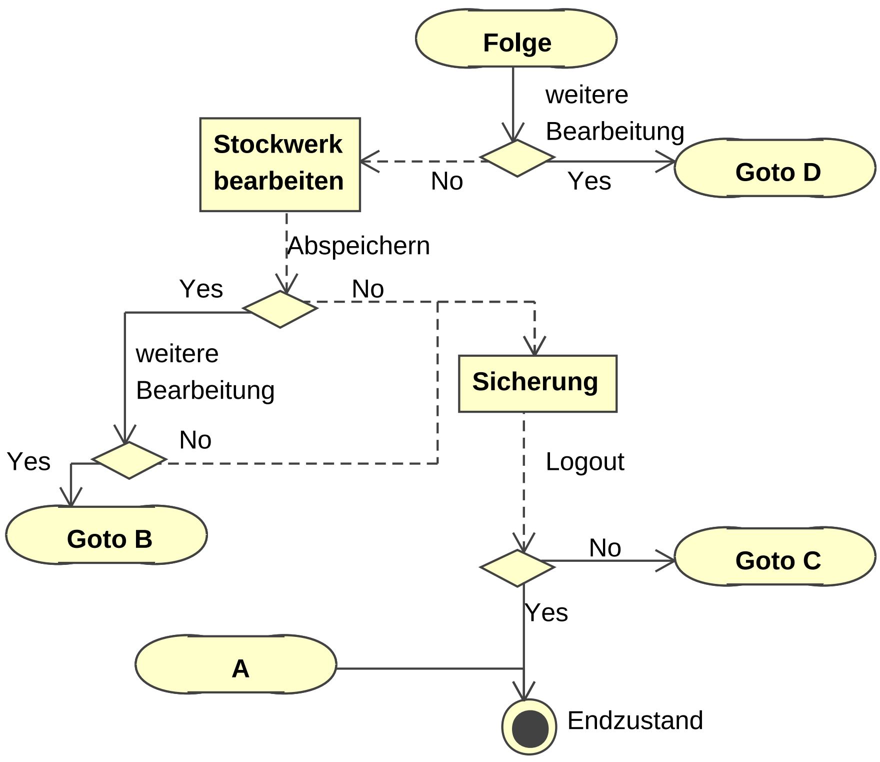


Abbildung 1.19: VM Gestaltung - Activity Diagramm

User Interface

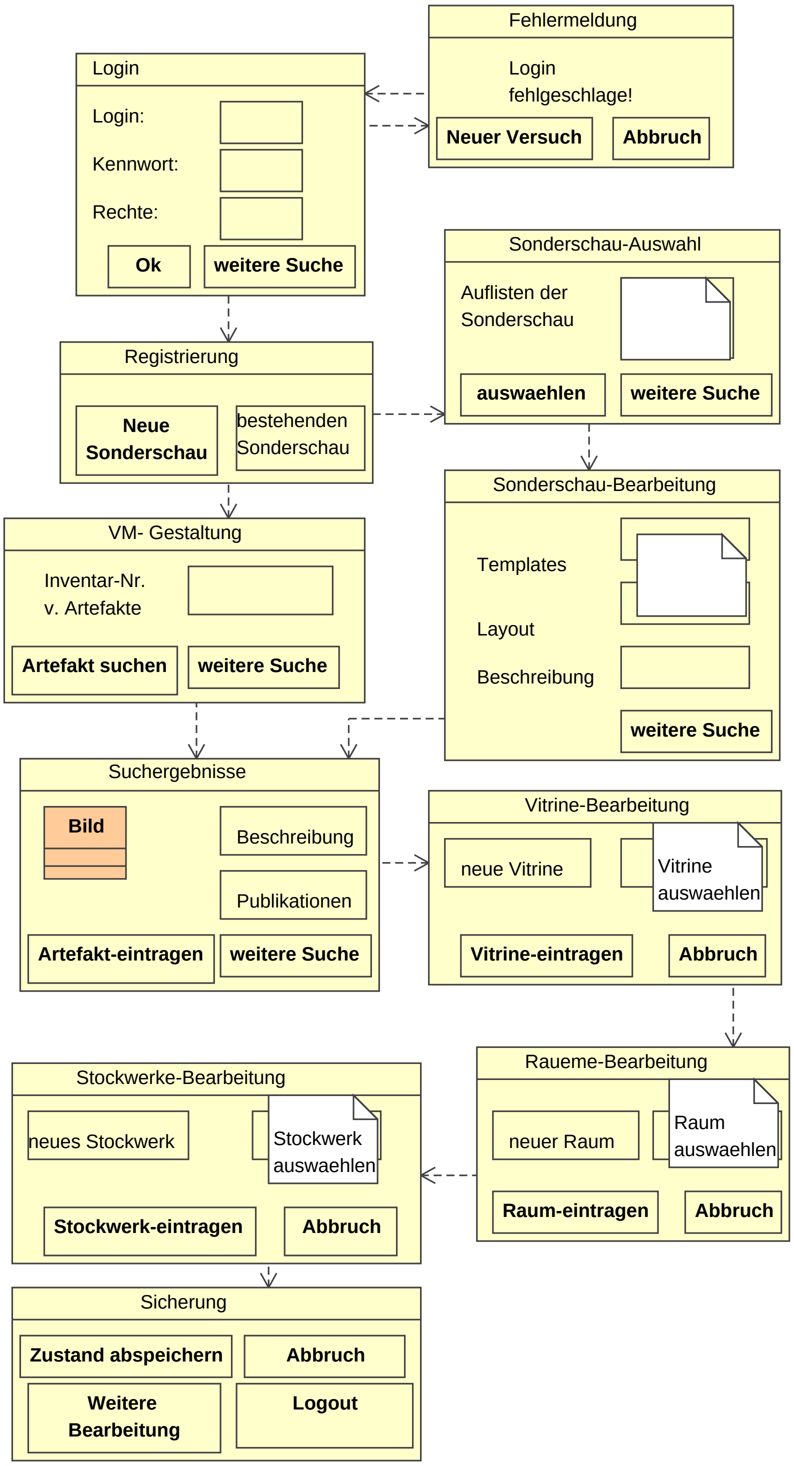


Abbildung 1.20: VM Gestaltung - User Interface

Secondary Scenarios

Sequence Diagrams

Subordinate Use Cases

View of Participating Classes

Other Artifacts

Other Requirements

1.5.9 Detailierte Usecase Beschreibung: Wartung eines virtuellen Museums:

Wartung eines virtuellen Museums:

Jeder Erfasser hat das Recht virtuelle Museum(Sonderschauen), die er erstellt hat, zu warten.
Außerdem kann er die Sonderschau für andere Erfasser freigeben.
Die Freigabe der Sonderschau für den Internetbenutzer kann jedoch nur über den Direktor durch hinzufügen der Sonderschau zum virtuellen Museum geschiehen.

actors:

- Direktor
- Erfasser.

Priorität:

5

Status:

Filled

Vorbedingungen:

- Der Erfasser muß eingeloggt sein.
- Der Erfasser muß über die erforderlichen Rechte an dieser Sonderschau verfügen

Nachbedingungen

Extension Points:

- Fehlerbehandlung

"Used" Use Cases:

- RechteWarten
- Artefakt suchen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Der Usecase beginnt mit der Selektion (VM-warten)
2. Der Erfasser logt am System ein
3. *IF* (Passwort falsch oder Rechte nicht vorhanden)

- Goto 1 or to Endzustand

4. *ELSEIF*

5. *WHILE Rechte Vorhanden=True* Do

   (a) Es wird eine Seite generiert
   (b) Sonderschau auswählen
   (c) Hierarchie der Sonderschau wird angezeigt
   (d) Sonderschau betrachten
   (e) (editieren,erstellen oder löschen eines Stockes,Raum oder Vitrine ist Optional.)
   (f) If Vitrine ausgewählt
       - Artefakt auswählen
       - **Artefakt suchen**
       - Artefakt wird angezeigt
   (g) endif
   (h) Katalog oder CD erstellen.Optional
   (i) If Sonderschau freigeben=TRUE)
   (j) Hinzufügen des Sonderschaues in des VM wird aber nur vom Verwalter durchführbar sein.
   (k) Bestätigen und warten auf eine Antwort vom System.

6. *ENDWHILE*

7. *ENDIF*

8. ENDZUSTAND :

   - Zustand abspeichern
   - Abmeldung vom System.

Activity Diagram

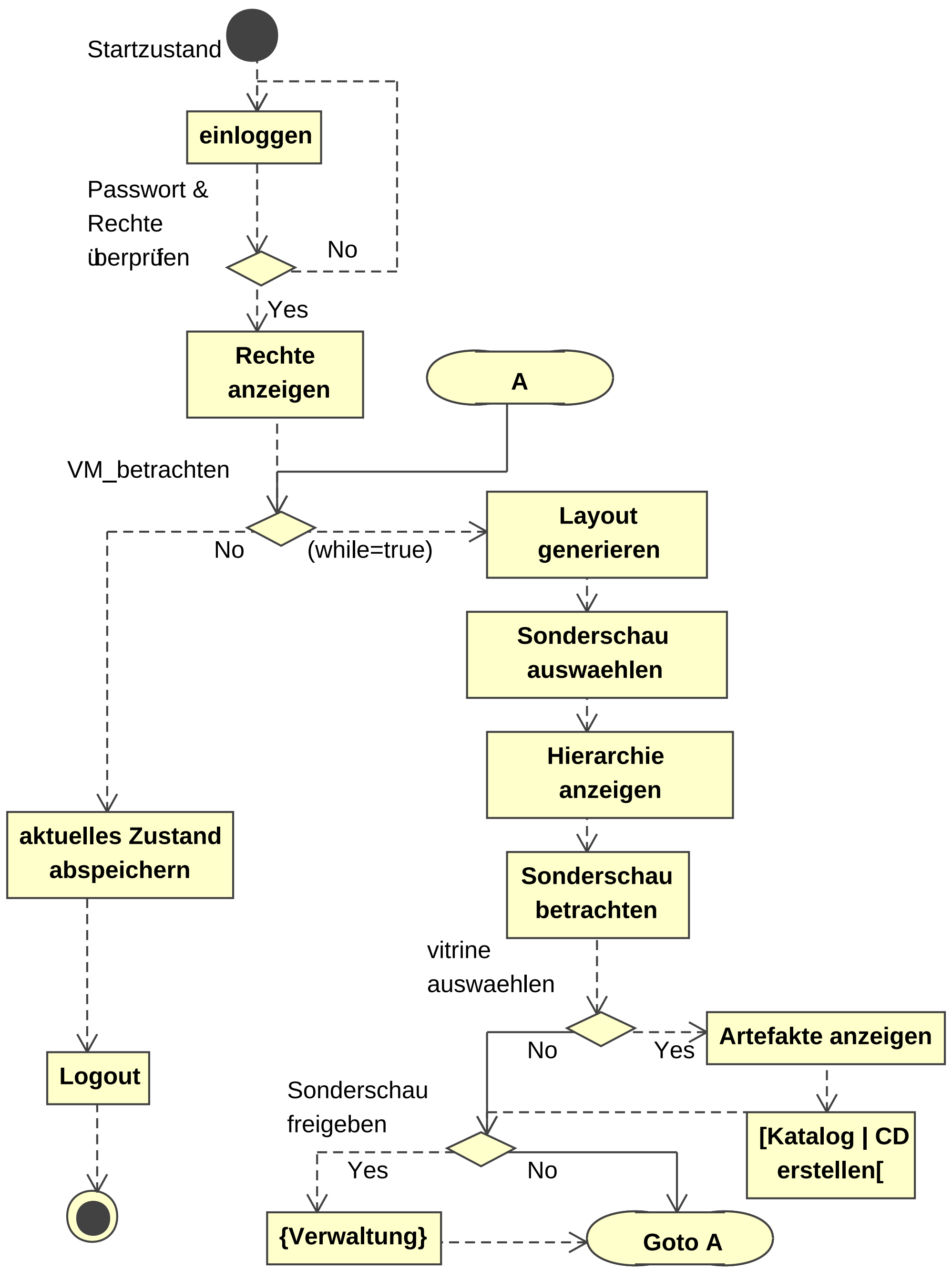


Abbildung 1.21: Aktivitätsdiagramm für die Wartung des virtuellen Museums

User Interface

Secondary Scenarios

Sequence Diagrams

Subusecases

- Museumsrechte erteilen
- Museumshierarchie erstellen
- Layout erstellen

View of Participating Classes

Other Artifacts

Other Requirements

1.5.10 Detailierte Usecase Beschreibung: Betrachtung eines virtuellen Museums:

Betrachtung eines virtuellen Museums:

Jeder Erfasser hat das Recht virtuelle Museen(Sonderschauen). Für den Internet-Besucher Erfolgt die Betrachtung anhand Verweise mit denen er ganz leich im VM navigieren kann.

actors:

- Direktor
- Erfasser
- Internet-Besucher.

Priorität:

8

Status:

Finished

Preconditions

- Der Erfasser muß eingeloggt sein.
- Der Erfasser muß über die erforderlichen Rechte an dieser Sonderschau verfügen.
- Für Das VM wird keine Anmeldung verlangt solange der User als Anonymous das VM betrachten will.

Nachbedingungen:

Einträge erfolgen ganz od. gar nicht

Extension Points:

- Fehlerbehandlung

"Used" Use Cases:

- Wissenschafltische Arbeit Suchen

Flow of Events:

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Der Usecase beginnt mit der Selektion (VM-betrachten)
2. Das System überprüft der Passwort des Users und seine Rechte
3. Falls der User ein Internet-Besucher ist dann gehe zu Punt 6.
4. IF (Passwort falsch oder Rechte nicht vorhanden)
   - Goto 1
5. ELSEIF
   (a) Es wird eine Seite generiert,die die Möglichkeiten des Erfassers anhand seine Rechte auflistet.
   (b) Der Verwalter(Direktor) hat uneingäschränkte Rechte.
   (c) Sonderschu zum Zweck der Betrachtung aus einer Liste auswählen.Optional.
6. Anhand Verweise wird es Möglich sein innerhalb des VM oder des Sonderschaues zu navigieren.
7. Dabei wird die Hierarchie des VM oder des Sonderschau angezeigt.
8. Abmeldung vom System.gilt nur für den Erfasser und der Verwalter.

Activity Diagram:

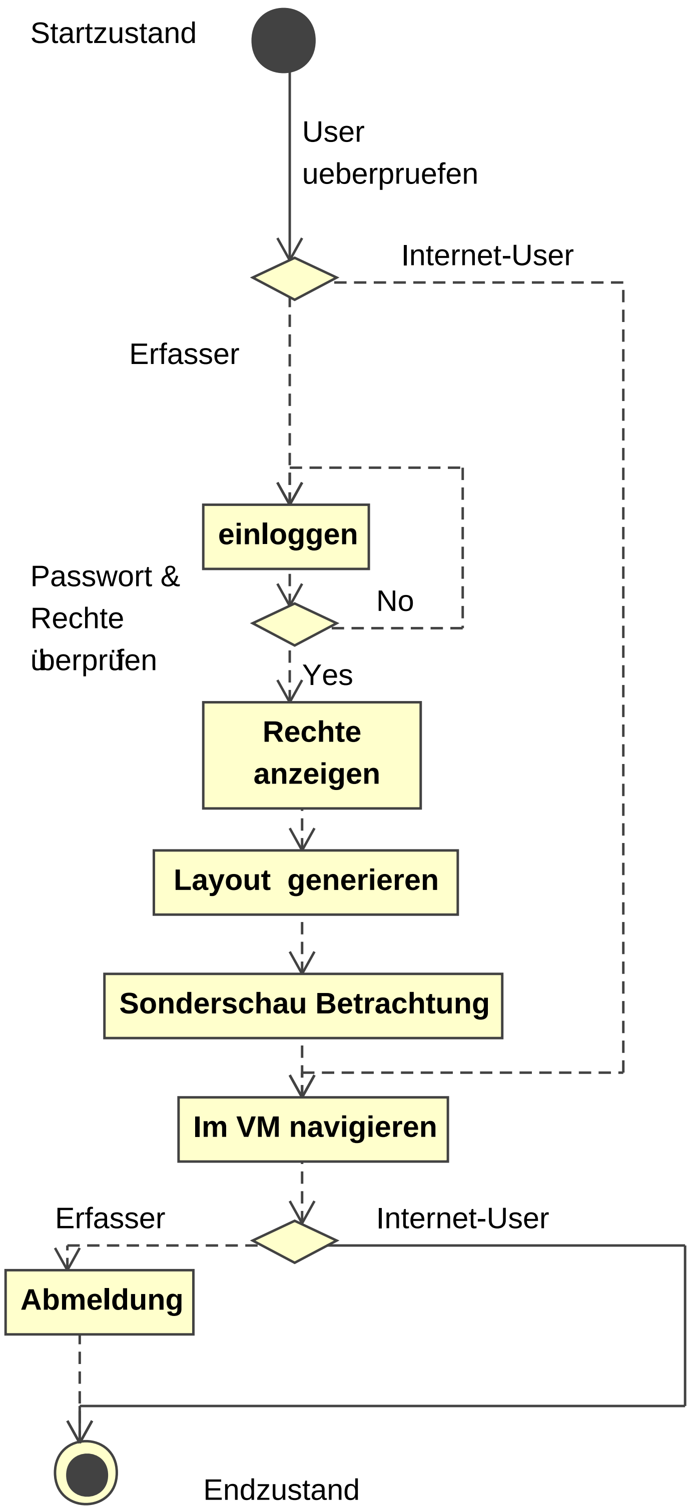


Abbildung 1.22: Aktivitätsdiagramm für die Betrachtung des virtuellen Museums

User Interface

Secondary Scenarios

Sequence Diagrams

- Museumshierarchie anzeigen

View of Participating Classes

Other Artifacts

Other Requirements

1.5.11 Detailierte Usecase Beschreibung: Wissenschaftliche Arbeit erfassen

Autor:Kasbi M. Jamal

Wissenschaftliche Arbeit erfassen:

Es werden vom Erfasser(Ausgrabungsleiter) Daten zu einem vorhandenen Artifakt in die DB eingegeben, dies ist möglich nur in dem Fall, dass der Erfasser besitzer von dem betrefenden Artifat.Aufbauend auf diese Wissenschaftliche Daten können nun Beziehungen vom Erfasser zu anderen Artifakten entstehen.Er kann dabei nch vorhandenen Artifakten suchen.Am Ende kann er die eingegebenen Daten Sichern oder verwerfen.

actors

Erfasser(Ausgrabungsleiter)

Priorität

3

Status

Filled

Vorbedingungen

Der Erfasser muß eingeloggt sein. Der Erfasser muß die erforderlichen Rechte auf den Artifakt besitzen.DB nicht leer d.h. es existiert mind. ein Artifakt in der DB.

Nachbedingung

Daten sind gespeichert.Im Fall, dass eine Beziehung entstehen soll, muss der Erfasser eine Bestaetigung fuer die Entstehung der Beziehung.

Extension Points

ErrorHandling.Passwort und Rechte überprüfen.

"Used" Use Cases

Usecases

- Artefakt suchen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System.
2. if (Acount vorhanden).
    (a) Eingangsseite zum System wird generiert.
3. if (Acount nicht vorhanden oder Passwort falsch eingegeben).
    (a) das System Verlangt wieder Acount und Passwort.
4. Der Erfasser wählt das Use Case Wissenschaftliche Arbeit erfassen.
5. Das System suchet in der DB nach vorhandenen Artifakten auf denen der Erfasser Rechte Zugriffe hat.
6. if (System findet Artifakte mit Zugrifsrechte)
    (a) Das System generiert die Seite zum Erfassen von Wissenschaftlichen Arbeiten mit einer Liste von den vorhandenen Artifakten, auf denen der Erfasser Zugrifsberechtigung hat.
7. if (System nichts gefunden)
    (a) Das System bringt die passende Meldung generiert die vorherige Seite.
8. Das System generiert die Seite zum Erfassen von Wissenschaftlichen Arbeiten mit einer Liste von den vorhandenen Artifakten, auf denen der Erfasser Zugrifsberechtigung hat.
9. Erfasser wählt einen Artifakt aus der Liste und bestätigt.
10. Das System zeigt die Grund Daten des Artifaktes.
11. Erfasser pflegt die Attribute des Wissenschaftlichen Arbeit.
12. Erfasser sichert die Eingegebenen Daten oder bricht die Verarbeitung ab.
13. if (Erfasser bricht Verarbeitung ab)
    (a) Das System verlässt die Erfassungmaske. Der use case ist beendet.
14. if (Erfasser sichert die eingegebenen Daten)
    (a) Das System speichert die Wissenschaftliche Arbeit auf die Datenbank und bringt die passende Meldung.
    (b) Das System fragt den Erfasser, ob er Beziehungen zum anderen Artifakten Pflegen will.
    (c) if (Erfasser mit ja antwortet)
        i. Das System zeigt zwei Optionen: Artifakt Suchen und Artifakt Auswählen.

ii. if (Erfasser Suche nach Artifakt auswählt)
    A. Der Erfasser gibt den Inventarnummer ein.
    B. Er Bestätigt die Eingabe.
    C. das System sucht nach dem Artifakt in der DB und bringt die passende Meldung mit Erfolg oder nicht Erfolg.Beim Erfolg zeigt es dann die Daten.

iii. if (Erfasser Artifakt auswählen auswählt)
    A. Erfasser wählt einen Artifakt aus der Liste.
    B. Erfasser bestätigt.
    C. Das System generiert die Seite um Beziehungen zu Pflegen.

(d) if (Erfasser mit nein antwortet)
    i. Das System verlässt die Erfassungsmaske. Der use case ist beendet.

15. Das System generiert die Seite um Beziehungen zu Pflegen, und zeigt die Grund Daten des Artifaktes.

16. Erfasser Pflegt die Daten Einer Beziehung

17. Erfasser sichert die Eingegebenen Daten oder bricht die Verarbeitung ab.

18. if (Erfasser bricht Verarbeitung ab)

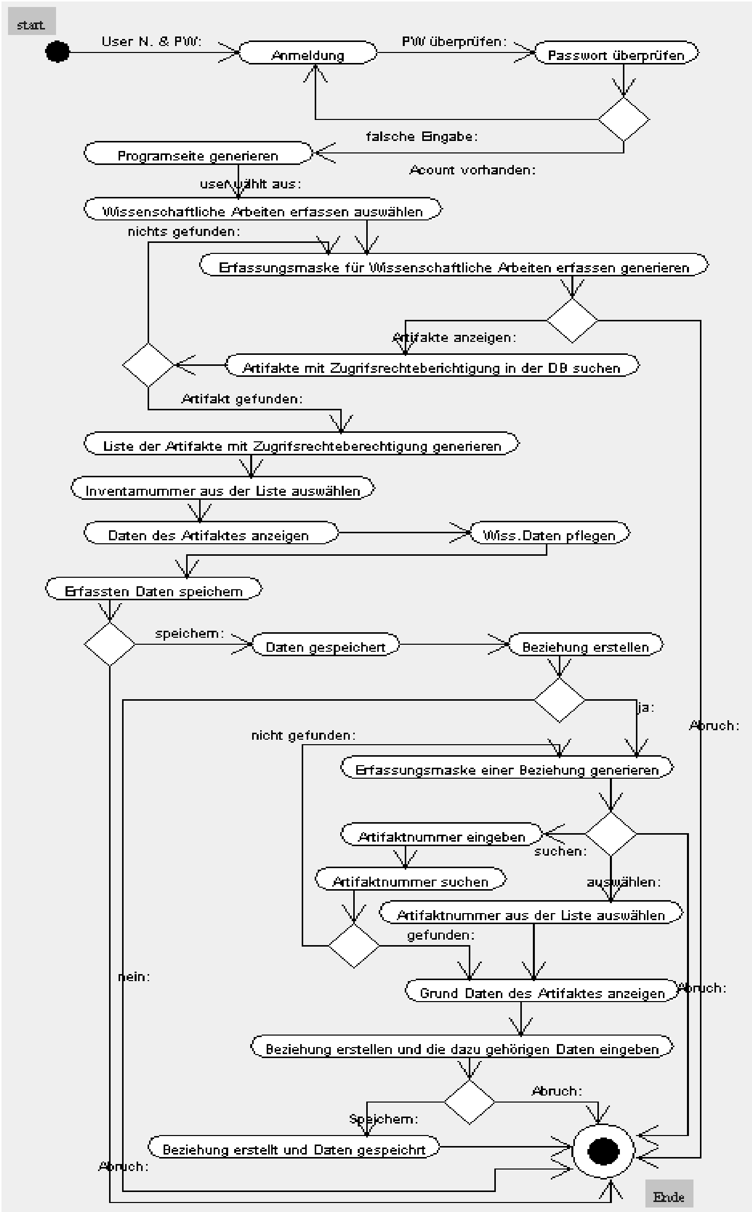
    (a) Das System verlässt die Erfassungmaske der Beziehung. Der Gesamt use case ist beendet.

19. if (Erfasser sichert die eingegebenen Daten der Beziehung)

    (a) Das System speichert die Daten der Beziehung auf die Datenbank und verlässt die Erfassungsmaske der Beziehung. Der Gesamt use case ist beendet.

20. Erfasser meldet sich vom System ab.

Activity Diagram



User Interface

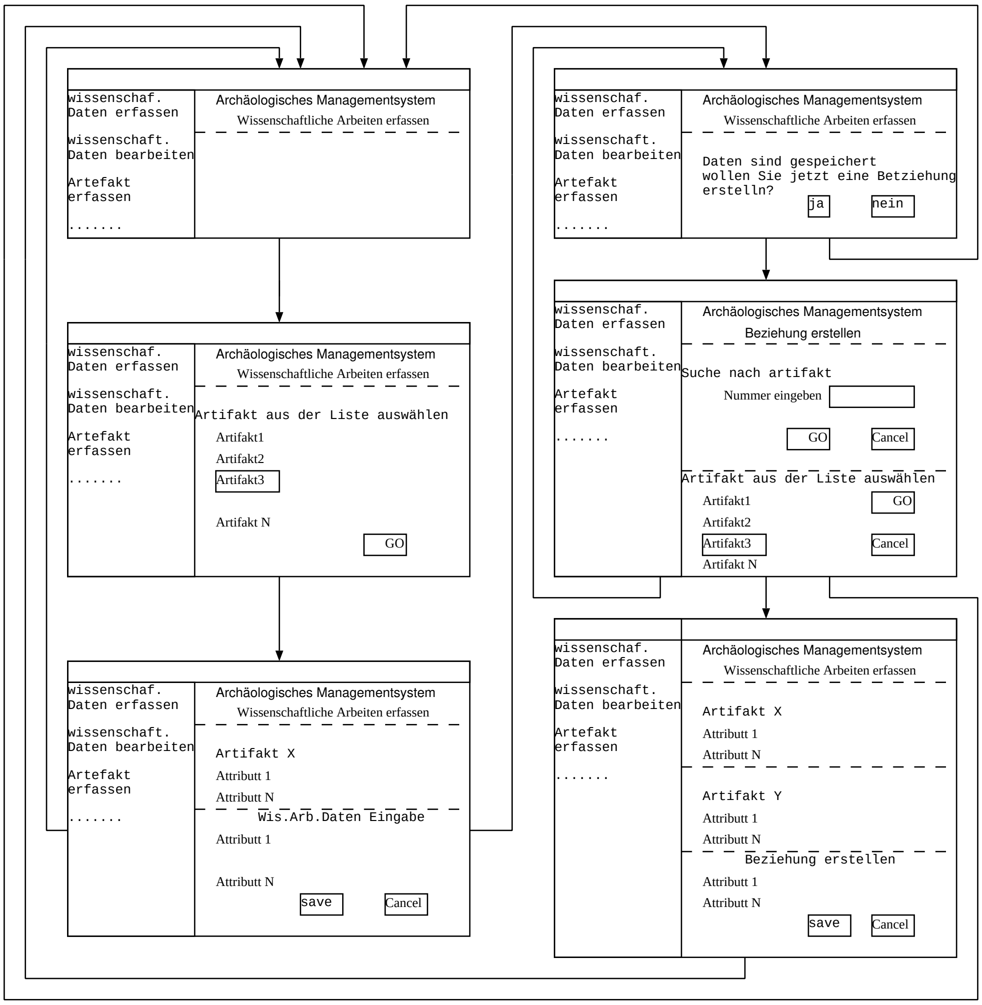


Abbildung 1.23: User Interface

Secondary Scenarios

Der User hat die Moeglichkeit jeder Zeit aus dem Use Case auszusteigen.

Sequence Diagrams

Subusecases

- Beziehung erstellen.

View of Participating Classes

Other Artifacts

Other Requirements

1.5.12 Detailierte Usecase Beschreibung: Wissenschaftliche Arbeit bearbeiten

Autor:Kasbi M. Jamal

Wissenschaftliche Arbeit bearbeiten:

Es werden vom Erfasser(Ausgrabungsleiter) Wissenschaftliche Arbeiten,auf die er Zugrifsrechte hat, bearbeitet,d.h. geaendert, geleoscht oder Beziehungen erstellt.Er kann dabei nach vorhandenen Artifakten suchen.Am Ende kann er die eingegebenen Daten Sichern oder verwerfen.

actors

Erfasser(Ausgrabungsleiter)

Priorität

4

Status

Filled

Vorbedingungen

Der Erfasser muß eingeloggt sein. Der Erfasser muß die erforderlichen Rechte auf den Wissenschaftliche Arbeitbesitzen.DB nicht leer d.h. es existiert mind. eine Wissenschaftliche Areit in der DB.

Nachbedingung

Daten sind gespeichert.Im Fall, dass eine Beziehung entstehen soll, muss der Erfasser eine Bestaetigung fuer die Entstehung der Beziehung,fuer das leoschen muss die passende Meldung auch gebracht werden.

Extension Points

ErrorHandling.Passwort und Rechte überprüfen.

”Used” Use Cases

Usecases

- Artefakt suchen

Flow of Events

In Folge wird ein Ablauf von Ereignissen aufgelistet, der in diesem Use-Case auftritt:

1. Anmeldung beim System.
2. if (Acount vorhanden).
   (a) Eingangsseite zum System wird generiert.
3. if (Acount nicht vorhanden oder Passwort falsch eingegeben).
   (a) das System Verlangt wieder Acount und Passwort.
4. Der Erfasser wählt das Use Case Wissenschaftliche Arbeit bearbeiten.
5. Das System suchet in der DB nach vorhandenen Wiss.Arb. auf denen der Erfasser Zugriffsrechte hat.
6. if (System findet Wiss.Arb. mit Zugrifsrechte)
   (a) Das System generiert die Seite zum bearbeiten von Wissenschaftlichen Arbeiten mit einer Liste von den vorhandenen Wiss.Arb., auf denen der Erfasser Zugrifsberechtigung hat.
7. if (System nichts gefunden)
   (a) Das System bringt die passende Meldung und generiert die vorherige Seite.
8. Das System generiert die Seite zum bearbeiten von Wissenschaftlichen Arbeiten mit einer Liste von den vorhandenen Wiss.Arb., auf denen der Erfasser Zugrifsberechtigung hat.
9. Erfasser wählt eine Wiss.Arb. aus der Liste und bestätigt.
10. Das System zeigt die Grund Daten der Wiss.Arb. und des Artifaktes zu dem die Arbeit gehoert.
11. Erfasser waehlt aus der drei Optionen: LOESCHEN,AENDERN,BEZIEHUNG, eine aus.
12. if (Erfasser waehlt Beziehung aus)
    (a) Das System zeigt zwei Optionen: Artifakt Suchen und Artifakt Auswählen.
    (b) if (Erfasser Suche nach Artifakt auswählt)
        i. Der Erfasser gibt den Inventarnummer ein.
        ii. Er Bestätigt die Eingabe.
        iii. das System sucht nach dem Artifakt in der DB und bringt die passende Meldung mit Er folg oder nicht Erfolg.Beim Erfolg zeigt es dann die Daten.
    (c) if (Erfasser Artifakt auswählen auswählt)
        i. Erfasser wählt einen Artifakt aus der Liste.

ii. Erfasser bestätigt.

(d) Das System generiert die Seite um Beziehungen zu Pflegen.

(e) Erfasser Pflegt die Daten Einer Beziehung.

13. if (Erfasser waehlt Aendern aus)

    (a) Das System generiert die Seite zum Aendern von Attribute deR Arbeit.

    (b) Erfasser pflegt die Attribute des Wissenschaftlichen Arbeit.

14. if (Erfasser waehlt Loeschen aus)

    (a) Das System sucht in der DB nach vorhandenen Beziehungen fuer die Wiss.Arb.

    (b) System fragt den Erfasser nach Bestaetigung des Loeschens.

    (c) if (Ja)

        i. Die Wiss.Arbeit und die Beziehungen,die zwischen zwei Artifakten besteht wird geleoscht.

        ii. System bringt die passende Meldung.Ende des Use Cases.

    (d) if (nein)

        i. Operation wird abgebroschen.Ende des Use Cases.

15. Erfasser sichert die Eingegebenen Daten oder bricht die Verarbeitung ab.

16. if (Erfasser bricht Verarbeitung ab)

    (a) Der use case ist beendet.

17. if (Erfasser sichert die eingegebenen Daten der Beziehung).

    (a) Das System speichert die Daten auf die Datenbank. Der use case is t beendet.

18. Erfasser meldet sich vom System ab.

Activity Diagram

User Interface

Secondary Scenarios

Der User hat die Moeglichkeit jeder Zeit aus dem Use Case auszusteigen.

Sequence Diagrams

Subusecases

- Beziehung erstellen.
- Wiss.Arbeit Loeschen.
- Daten Aendern.
- Wissenschaftliche Arbeit suchen.

View of Participating Classes

Other Artifacts

Other Requirements

Kapitel 2

Durchführbarkeitsstudie

2.1 Vollständigkeit der Anforderungen

Wird erst später erstellt.

2.2 Konsistenz der Anforderungen

Erst nach der zustimmung des Kunden. Bis jetzt war der Kunde zufrieden mit unser Arbeit.

2.3 Ökonomische Rechtfertigung

2.3.1 Risikoabschätzung

Bei einem Softwareprojekt dieser größe besteht die Gefahr, daß das Endergebnis mit den Wünschen der Benutzer nicht übereinstimmt. Um diese Gefahr zu minimieren, ist es empfehlenswert das Gesamtprojekt in kleinere Teile zu zerlegen. Abbildung 1.1 zeigt die Einteilung des Projektes in seine logischen Komponenten.
Anhand dieser Gliederung(Abbildung 1.1 auf Seite 5) ist es möglich eine Aufgabenverteilung vorzunehen. Weiters ist es möglich eine Aufwandsschätzung (Zeit, Kosten) vorzunehmen.

2.3.2 Geschätzter Projektumfang

Aufteilung und Gewichtung der Actors: Die bereits besprochenen Actors werden in Folge nach ihrer Komplexität gewichtet. Die Einteilung der Actors erfolgen in drei Bereiche:

einfach: Einfaches Programm-Interface.**Faktor: 1**

durchschnittlich: Interaktives, protokollbasiertes Interface. **Faktor: 2**

komplex: Graphisches Interface. **Faktor: 3**

Actorgewichttotal $= 0E * 1 + 0D * 2 + 3K * 3 = \mathbf{9}$

Aktor-Typ	Beschreibung	Faktor
Verwalter	Verwaltung des Systemes	3
Erfasser	Daten Erfassen und VM	3
Anonymous	Internet-Suche und Besuch im VM	3

Tabelle 2.1: Aufteilung der Actors.

Aufteilung und Gewichtung der Use-Cases: Auch hier werden die Use-Cases wieder nach den oberen Kriterien unterschieden. Basis der Entscheidung hier ist die Anzahl der Transaktionen im Use-Case. Für diese Betrachtung wird eine Transaktion als Set von automatisch durchgeführten Aktionen angesehen. So hat zum Beispiel ein *einfacher* Use-Case 3 (Faktor 5), *durchschnittlicher* 4 bis 7 (Faktor 10) und ein *komplexer* 7 bis 10 (Faktor 15).

Use Case-Typ	Beschreibung	Faktor	Kategorie
Artefakte suchen	Suchen nach Artefakten in der Datenbank	15	K
Arch. Daten eintragen	Grunddaten ins System eingeben - archäologische Daten	15	K
Arch. Daten bearbeiten	Bearbeitung von bereits im System bestehenden Daten.	15	K
Artefakte löschen	Löschen von bestehenden arch. Daten.	10	D
VM aufbauen	Aufbau und Erstellung eines VM oder einer Sonderschau.	15	K
VM warten	Wartung, Änderung und Gestaltung des VM oder einer Sonderschau.	15	K
VM betrachten	Betrachtung von VM oder einer Sonderschau.	10	D
Rechte bearbeiten	Benutzerrechte bearbeiten und administrieren.	5	E
Erfasser administrieren	Administration von Rechten und persönlichen Daten der Erfasser.	5	E
Erstellung wiss. Arbeiten	Erfassen von wissenschaftlichen Daten.	10	D
Bearbeitung der wiss. Daten	Bearbeiten von bereits erfaßten wissenschaftlichen Daten	10	D

Tabelle 2.2: Aufteilung der Use-Cases mit Faktorbewertung.

$$\mathbf{Actorgewichtungtotal} = 2E * 5 + 4D * 10 + 5K * 15 = \mathbf{125}$$

Durch Addition des Actor Gewichtes und des Use Case Gewichtes erhält man die unangepaßten Use Case Punkte (UUCP).

Diese werden noch durch die Komplexität des Projektes und durch die Erfahrung der Mitarbeiter beeinflußt.

$$\mathbf{UUCP} = 9 + 125 = \mathbf{134}$$

Gewichtung der technischen Faktoren: Als nächstes muß die Komplexität des Systems und die Erfahrung der Mitarbeiter berücksichtigt werden. Diese Eigenschaften gewichten die oben berechneten UUCPs.

Der Faktor für die technische Komplexität (TCF) wird durch folgende Tabelle dargestellt, wobei jeder Teilbereich mit einem Faktor zwischen 0 und 5 bewertet (0=unwichtig, 5=wesentlich).

Eigenschaft	Gewicht	Wert	Ergebnis	Grund
Verteilte Systeme	2	5	10	Es werden durchwegs verteilte Systeme eingesetzt
Antwort- oder Durchsatzleistung	1	5	5	Die Responsezeit ist bei interaktiven Systemen enorm wichtig
Endbenutzereffizienz	1	4	4	letztendlich muß nur der Enbenutzer darauf arbeiten
Komplexe interne Prozesse	1	4	4	Datenbankabfragen
Wiederverwendbarer Code	1	1	1	wurde nicht gefordert
einfach Installierbarkeit	0.5	1	0,5	wir installieren und betreiben selbst
einfache Benutzbarkeit	0.5	2	1	Gui immer ausgehend vom DAU entwerfen
Portierbarkeit	2	2	4	Server - wurde nicht gefordert/ Client - durch Browser gegeben
Einfachheit von Änderungen	1	3	3	nur im Rahmen des virtuellen Museums
gleichzeitige Zugriffe	1	5	5	sind immer vorhanden
inkludiert spezielle Sicherheitsmechanismen	1	3	3	herkömmliches Login
erteilt einem Dritten direkten Zugriff	1	2	2	nur einem anderen Erfasser
spezielle Benutzer-Trainingsinterfaces sind erforderlich	1	2	2	Hilfe-Dok.bereitstellen

Tabelle 2.3: Berechnung des technischen Aufwandes

$$\mathbf{TFactor} = \sum(Level)*(Faktor) = 10{+}5{+}4{+}4{+}1{+}0.5{+}1{+}4{+}3{+}5{+}3{+}2{+}2 = \mathbf{44.5}$$
$$\mathbf{TCF} = 0.6 + (0.01 * TFaktor) = 0.6 + (0.01 * 44.5) = \mathbf{1.045}$$

Eigenschaft	Gewicht	Wert	Ergebnis	Grund
Vertraut mit Rationel Unified Process	1.5	2	3	die meisten sind nicht damit vertraut
Anwendungserfahrung	0.5	3	1.5	alle haben schon an diversen Anwendungen programmiert
obektorient. Erfahrung	1	3	3	alle haben schon an diversen Anwendungen programmiert
Führungsqualitäten	0.5	3	1.5	zwei Leute haben bereits ein Projekt geleitet
Motivation	1	0	0	wir werden sehr schlecht bezahlt
Verfügbarkeit	1	3	3	wir studieren alle noch
Teilzeitarbeiter	-1	2	-2	Teilzeit-Tätigkeit in geringem Ausmass
Einfachheit der Programmiersprache	-1	1	-1	java;jsp;js;sql;html;xml/xslt;php

Tabelle 2.4: Abschätzung der Erfahrungswerte

Erfahrungswertabschätzung: Berechnung der Werte:

$$3 + 1.5 + 3 + 1.5 + 0 + 3 + \text{-}2 + \text{-}1 = 9$$
$$\textbf{EF} = 1.4 + (-0.03 * 9) = \mathbf{1.13}$$

Use-Case Points: Die *Use-Case Points* (UCP) werden nun aus den berechneten Werten ermittelt:

$$\text{UCP} = \text{UUCP} * \text{TCP} * \text{EF}$$
$$UCP = 125 * 1.045 * 1.13 = \mathbf{147.500}$$

Projektabschätzung: Zur Gesamtabschätzung wird nun die UCP zu Mann-Stunden pro UCP umgerechnet.
Zur Genaueren Betrachtung der Fähigkeiten sind die Erfahrungswerte näher zu untersuchen. Es sind in Tabelle 2.4 von F1 bis F6 drei unter den Wert 3 und von F7 bis F8 0 über 3 also insgesamt 3 Einträge.
Da mehr als 2 Einträge, die über den Wert 2 und unter den Wert 5 liegen, gefunden sind, wird dann die UCPs mit 28 Mann-Stunden zu berechnen.

Die endgültige Abschätzung des Zeitaufwandes wird nun ermittelt:

$$\textbf{Mann} - \textbf{Stundengesamt} = UCP * 28 = 147.500 * 28 = 4130 also \mathbf{4130 Stunden}$$

Das entspricht etwa 21 Wochen mit einem fünf Mann Team .
Rechnen wir dazu 3 Wochen für die Kommunikation dann kommen wir auf 24 Wochen.

Kostenabschätzung Berechnung der Kosten mit einem Stundensatz von 1200,– Schilling.

$$\textbf{Kosten} = Mann - Stunden * 1200.0 = (4130 + 120) * 1200.0 = \mathbf{5'100,000.00 Schilling}.$$

Bei dieser Berechnung sind nicht berücksichtigt:

- Software
- Lizenzen für Datenbank, BS, ...
- Hardware

Kapitel 3

Architektur

In einer generellen Betrachtung haben wir für die Darstellung des Systemes als Architektur den *Layer pattern* gewählt. In dieser Methode wird das System aus verschiedenen Layern aufgebaut. Die einzelnen Bestandteile haben nach innen eine starke Bindung und nach außen kommunizieren sie nur über Interfaces.

3.1 Schichten Architektur

Das System ist daher aus folgenden Schichten (Layer) aufgebaut:

Layer 7: GUI-Engine

Layer 6: Session: Kontrolle und Steuerung der gesamten Session mit Prüfung der Berechtigungen zu den Aktivitäten.

Layer 5: Datenrepräsentation: Verwaltung aller, für diese Session benötigten Daten.

Layer 4: Filterung und Steuerung des Datenbankzugriffes, eingeteilt in die Teilbereiche **Suchen, Einfügen und Ändern, Löschen**.

Layer 3: Datenbankzugriff mit Abfragen in aggregierter (komplexer) Form.

Layer 2: Aufspaltung von komplexen Abfragen in atomare Aufrufe und Kapselung des gesamten Datenbanksystems, um die Datenhaltung flexibel zu halten (Loadbalancing, mehrere Datenbanken, ...).

Layer 1: Datenbankverwaltung

Eine Darstellung dieser Architektur ist zu sehen in Abbildung 3.1.

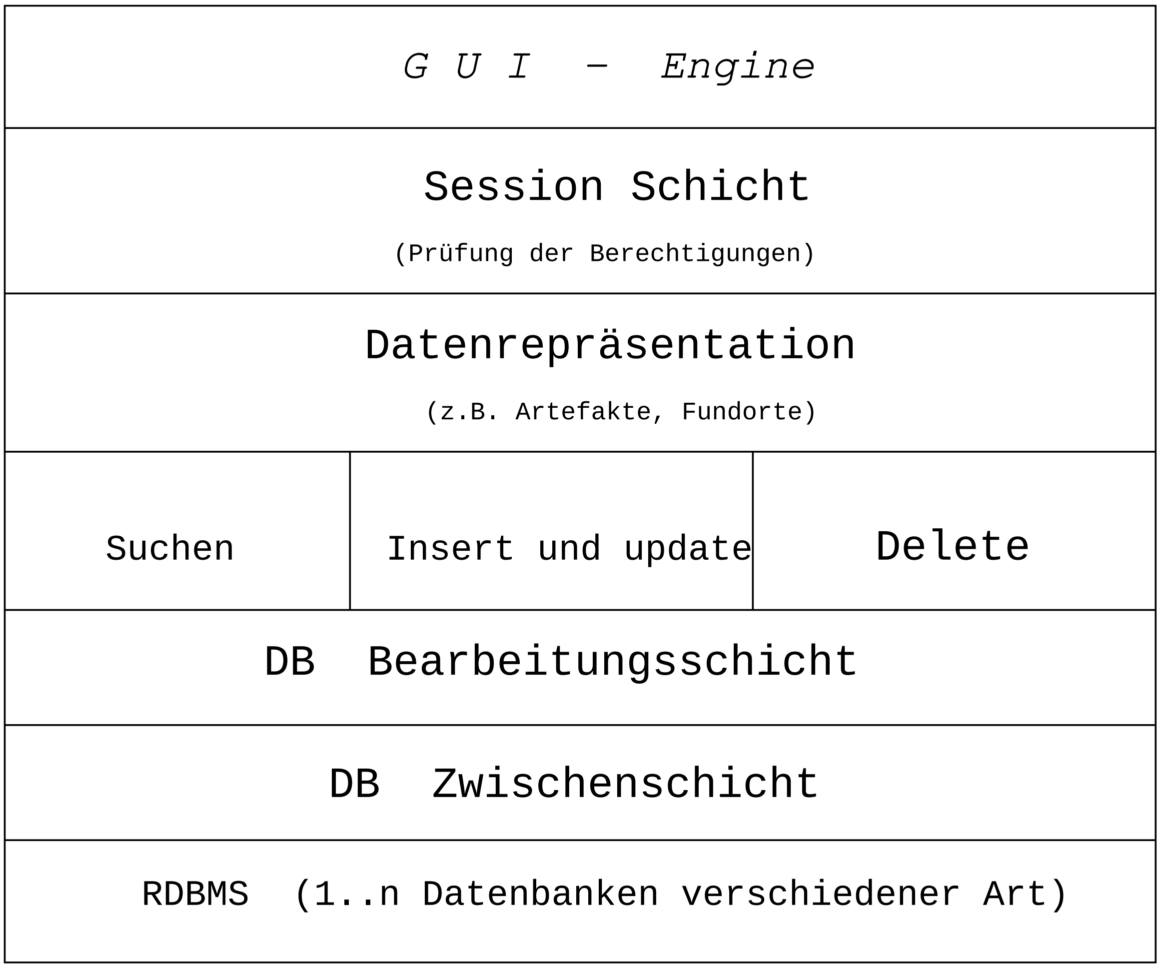


Abbildung 3.1: Architektur-Diagramm

3.2 Subsysteme

In Abbildung 3.2 sind die einzelnen Teile der Architektur aufgeteilt in *Subsysteme* dargestellt.

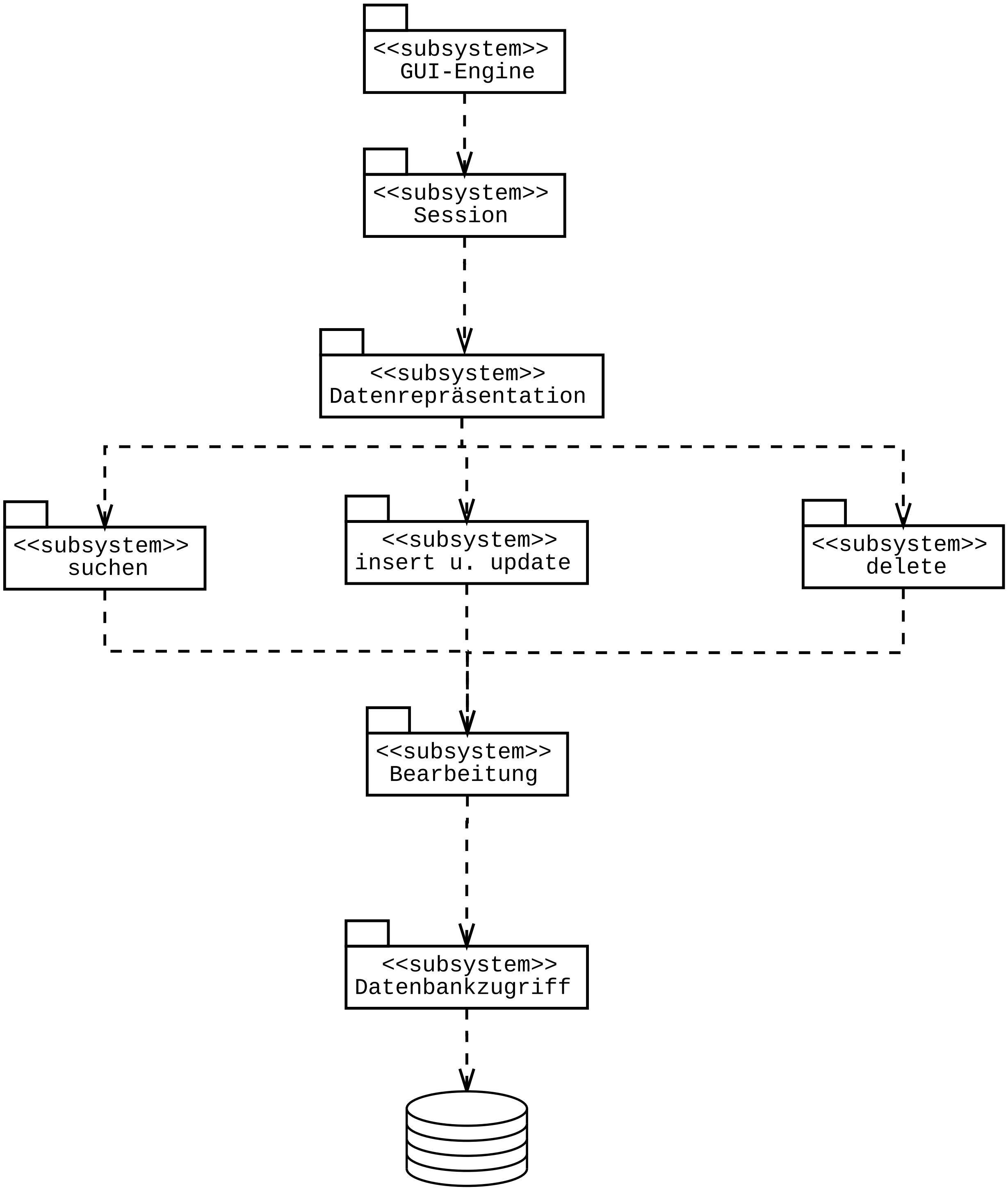


Abbildung 3.2: Diagramm von Subsystemen

Für die bisher beschriebenen Use Cases stellt diese Architektur-Darstellung einen Querschnitt der einzelnen Teilbereiche dar, die von jedem Use Case durchlaufen werden muß. Am Beispiel des Suchens durch den Erfasser:

- Die GUI-Engine erzeugt (nach korrektem Login) eine Eingabeform für den Erfasser.
- Die Session verwaltet die Berechtigungen bezüglich des Erfassers beim Suchen.
- Die Datenrepräsentation enthält alle relevanten Suchbegriffe und Ergebnisdaten der Suche.
- Bei der Suche wird im *Layer 4* die Suche aufgerufen.
- Die Frage wird weitergereicht an die darunterliegenden Layer, die nach entsprechendem Aufwand eine atomare Datenbankabfrage durchführen.
- Nach dem Durchsuchen der Datenbank werden von dort in umgekehrter Reihenfolge die einzelnen Stationen durchlaufen bis die Ergebnisse die GUI-Engine erreicht haben, die dann eine Ergebnisdarstellung aufbaut.

Die Subsysteme enthalten verschiedene Bereiche der Funktionalitäten, welche den anderen Subsystemen mittels Interfaces zur Verfügung gestellt werden.

3.2.1 Funktionalität und Interface

1. **GUI-Engine:**

   Funktionalität: Generierung einer GUI aus der Funktionalität (Use Cases), der vorhandenen Daten und der zur Verfügung stehenden Rechte.

   Interface: GUI-Oberfläche (Html-Fenster am Browser).

2. **Session:**

   Funktionalität: Auswertung, Filterung und Weiterleitung der Informationen zwischen GUI und der darunterliegenden Schichten (Daten, Rechte, ..).
   Hier befinden sich die gesamten Use-Case Repräsentationen.
   Nach den Use Cases werden die darunterliegenden Schichten entspechende Aktionen ausführen.
   Bsp: Artefakte suchen:
   Daten von Artefakte werden gespeichert und Suchaufrufe suchen nach Artefakten (nicht Personendaten). Die Überprüfung der Berechtigungen erfolgt ebenfalls hier. D.h. Funktionen der unteren Schichten werden nur dann ausgeführt, wenn auch eine Berechtigung dafür gegeben ist (z.B. Artefakte löschen).

   Interface:
   - getData: Ist eine überladene Methode, die je nach Use Case die entsprechenden Daten zur Verfügung stellt.
   - search_for_Data: Mit Suchkriterien als Parameter wird eine Suche veranlaßt.
   - insert_updateData
   - delete_Data

3. **Datenrepräsentation:**

   Funktionalität: Verwaltung und Speicherung der gesammelten Daten dieser Sitzung.

   Interface:
   - getData: Ist eine überladene Methode, die je nach Use Case die entsprechenden Daten zur Verfügung stellt.
   - search_for_Data: Mit Suchkriterien als Parameter wird eine Suche veranlaßt.
   - insert_updateData
   - delete_Data

4. **Suchen:**

   Funktionalität: Aufbereitung der Anfragen für die Suche in der Datenbank in Abhängigkeit zur Art der Suche (Artefakte suchen als Erfasser, Internet-Besucher, ...).
   Die Aufbereitung ist abhängig vom übergebenen Objekt der Suchkriterien.
   Rückgabe der Ergebnisse aus der Suche an die darüberliegende Schicht.

   Interface: search_for_Data: Der Parameter ist ein Objekt zur Identifikation der Art der Suche. Rückgegeben wird das Suchergebnis oder eine Fehlermeldung.

5. **Einfügen und Ändern:**

   Funktionalität: Ein übergebenes Objekt wird zum Insert oder Update für Aktualisierung an die untere Schicht weitergeleitet. Bestätigung des Erfolgsstatus wird an die obere Schicht zurückgegeben.

   Interface: insert_updateData

6. **Löschen:**

   Funktionalität: Informationen aus der oberen Schicht werden für die untere Schicht aufbereitet, um bestimmte Informationen aus der Datenbank zu entfernen (z.B. ganzes Artefakt entfernen, Publikation entfernen).

   Interface: delete_Data

7. **Bearbeitung:**

   Funktionalität: Stellt der oberen Schicht die Funktionen zur Verfügung, die Abfragen komplexerer Datenbankzugriffe ermoglichen und gibt diese an die untere Schicht weiter.
   Die Ergebnisse aus dem Datenbankzugriff werden in entsprechend aggregierter Form an die darüberliegende Schicht weitergegeben (Bsp Suchertgebnisse bei Suche nach Artefakten).

   Interface:
   - select: Rückgabe ist das Suchergebnis in aggregierter Form.
   - insert: Rückgabe ist eine Erfolgsmeldung oder Fehlermeldung.
   - update: wie bei insert.
   - delete: wie bei insert.

8. **Datenzugriff:**

Funktionalität: Die aggregierten Anweisungen auf die Datenbank werden von der oberen Schicht übernommen und in atomare Datenbankanweisungen aufgeteilt. Hier finden die atomaren Datenbankzugriffe statt. Die Ergebnisse des Datenbankzugriffes (in Rohform) an die obere Schicht zurückgegeben (bei Suche), oder nur die entsprechende Erfolgsrückmeldung (Bestätigung für erfolgreiches Insert, Update oder Delete).

Interface:

- select: Rückgabe ist das Suchergebnis in aggregierter Form.
- insert: Rückgabe ist eine Erfolgsmeldung oder Fehlermeldung.
- update: wie bei insert.
- delete: wie bei insert.

Kapitel 4

Project Plan

Aufgrund der Einteilung des Systems mittels Layer-Pattern ist ein Projektplan rein auf Basis der Use Cases nicht sinnvoll. Vielmehr ist die Einteilung der Implementierung in Subsysteme in Verbindung mit Use Cases zu betrachten.

4.1 Kategorisierung

Eine Kategorisierung der Use Cases bezüglich der Problemstellung ist zwar noch mit dem Kunden genauer zu definieren, jedoch wollen wir hier eine Diskussionsgrundlage anführen. In der nachfolgenden Reihung ist die Priorität höher, je kleiner die Nummer der Kategorie ist (1 ist die höchste).

Kategorie 1: Erfassung, Bearbeitung und Entfernung von archäologischen Daten.

Kategorie 2: Erfassung, Bearbeitung und Entfernung von personellen Daten.

Kategorie 3: Suche nach und Anzeige von gefundenen Daten.

Kategorie 4: Erfassung, Bearbeitung und Entfernung von wissenschaftlichen Daten.

Kategorie 5: Erzeugung, Bearbeitung und Besuch von VM und Sonderschauen.

Kategorie 6: Erzeugung eines Kataloges oder einer CD.

Bei der Kategorisierung muß jedoch auch die Implementierung bezüglich der Subsysteme berücksichtigt werden. So sind für die Implementierung einer Kategorie die Subsysteme der Layer eins bis vier Voraussetzung. Die Prioritäten der Subsysteme gehen aus der Layer-Darstellung der Architektur aus Abbildung 3.1 hervor. Die Aufteilung der Priorität bei den Layern:

1. **Layer 1-4:** Datenbank mit Datenbankmodell, DB Zwischenschicht, DB Bearbeitungsschicht, Suchen, Einfügen und Ändern, Löschen

2. **Layer 5-7:** Datenrepräsentation, Session, GUI-Engine. Hier werden die Use Case spezifischen Teile implementiert (siehe Kategorien).

4.2 Entwicklungsrisiko

Das Entwicklungsrisiko ist bei der Aufteilung der Prioritäten für die Subsysteme berücksichtigt. Größtes Risiko liegt in den Layern eins bis vier der Layer-Architektur.

4.3 Entwicklungszeit

In der ersten Durchführbarkeitsstudie war die gesamte Mann-Wochen Anzahl bei 135 Stunden. Dieses Ergebnis soll hier nocheinmal betrachtet werden.
Ziel der Einteilung der Implementierungsphase in einzelne gleichgroße Iterationsschritte ist, daß nach jedem Iterationsschritt ein Teil der Implementierung (ein Use Case zum Beispiel) dem Kunden zum Testen übergeben werden kann.
So ist ein regelmäßiges Feedback für beide Seiten möglich. Hierbei sei jedoch zu beachten, daß nach Abschluß der Implementierung der Layer eins bis vier diese bei der Implementierung der Use Cases weiter angepasst werden müssen.

Durch die Einteilung in *Kategorien* und *Subsystemen* ergibt sich eine Einteilung in sieben Iterationsschritte.
Die ersten vier Layer der Layer-Architektur sind ein wesentliches Grundgerüst für die einzelnen Kategorien, deshalb werden hier zwei Iteratiosschritte einberechten.

Mit der Orientierung auf die erste Durchführbarkeitsstudie mit einer Projektgesamtlänge von 135 Mann-Wochen sind hier für die Implementierung eine Dauer von 120 Mann-Wochen eigerechnet.
Ein Iterationsschritt hat demzufolge (acht Iterationen) eine Dauer von 15 Mann-Wochen.
Die Implementierungsreihenfolge nach Priorität ist nachfolgend angeführt:

1. **Subsysteme:** Layer eins bis vier der Layer-Architektur. - **zwei Iterat.-S.: 30 M-W)**
2. **Kategorie: 1** - ein Iterationsschritt: **15 M-W**
3. **Kategorie: 2** - ein Iterationsschritt: **15 M-W**
4. **Kategorie: 3** - ein Iterationsschritt: **15 M-W**
5. **Kategorie: 4** - ein Iterationsschritt: **15 M-W**
6. **Kategorie: 5** - ein Iterationsschritt: **15 M-W**
7. **Kategorie: 6** - ein Iterationsschritt: **15 M-W**

Implementierungsgesamtdauer: 120 Mann-Wochen

Anzahl der Iterationsschritte: 8 (2 Schritte für Layer eins bis vier).

Kapitel 5

Prototyp

Aus dem Architekturdiagramm ist zu sehen, daß es hirachisch in Schichten eingeteilt ist. Die n plus erste Schicht nutzt die Dienste der n- ten Schicht. D.h. Fehler in der untersten Schicht wirken sich zwangsläufig auf alle darüberliegenden Schichten aus. Es ist daher sinnvoll diese Schicht, in unserem Fall die DB- Zwischenschicht einem Prototyping zu unterziehen.

5.1 DB- Zwischenschicht

5.1.1 Aufgaben

Es soll ein Objektmodell entwickelt werden, welches es ermöglicht DB- Abfragen aus der darüberliegenden Schicht entgegen zu nehmen, an die Datenbank(en) weiterzuleiten, auf das Ergebnis zu warten und dies dem aufrufenden Teil zu liefern. Bei mehreren DBen oder DB- Verbindungen, sollen die Abfragen (Last) geleichmäßig verteilt werden. Bei lesenden Zugriffen stellt dies kein Problem dar. Bei schreibenden Zugriffen muß noch eine Möglichkeit der Synchronisation gefunden werden (ist nicht Teil des Prototyps).
Der Prototyp, sowie das Gesamtprojekt werden in JAVA implementiert. Der DB- Zugriff erfolgt über JDBC.

5.1.2 Modellierung

Grobunterteilung

- Objektmodell welche die Abfragen (Query) repräsentieren.
- Objekt welches das Ergebnis (Result) repräsentiert.
- Zentrales Objekt(e) welches die Querys entgegennimmt, in eine Warteschlange stellt und gleich danach die nächste Abfrage bearbeiten kann.
- Bearbeitungsobjekt (Thread) welche die Abfrage aus der Warteschlange nimmt, bearbeitet und Resultobjekt in eine "Ergebniswarteschlange" stellt.

Objektbeschreibung

Das Zentrale Objekt der Abfrage ist das *SqlQuerySpec* Objekt. Es enthält die eigentlich SQL- Abfrage (PreparedStatement String), eine Array von Abfrageparametern und ein Array von Ergebnisparameter. Zur Identifizierung der Abfrage dient ein eindeutiger Schlüssel. Die *QueryParameter* und *ResultParameter* sind abstrakte Klassen. Abbildung 5.1 zeigt das SqlQuerySpec Objekt.

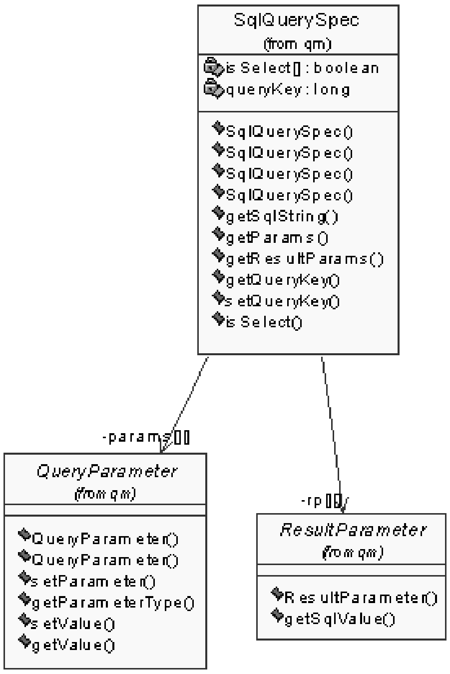


Abbildung 5.1: Objekt SqlQuerySpec: Objektrepräsentation einer SQL- Abfrage

Abbildung 5.2 zeigt die Abstrakte Klasse *QueryParameter* und die konkreten Ableitungen davon.
Abbildung 5.3 zeigt die Abstrakte Klasse *ResultParameter* und die konkreten Ableitungen davon.

Mit den oben beschriebenen Klassen wird nun eine SQL- Abfrage vollständig beschrieben.Das Ergebnis einer Abfrage wird in dem Objekt *ResultTable* (Abbildung 5.4) gespeichert und dem aufrufenden Teil übergeben.
Es muß auch notwendigerweise eine *zentrale* Stelle (Klasse) existieren, bei der die abfragenden Onjekte ihre Queries "abgeben" können. Diese Aufgabe übernimmt der *SqlDaemon*. Der SqlDaemon ist ein Objekt, welches nur einmal! existiert und beim ersten Aufruf initialisiert wird. Der Daemon nimmt Abfragen entgegen und stellt sie in eine Warteschlange (*SqlQueue*); danach ist er sofort wieder bereit die nächste Anfrage entgegen zu nehmen. Wird ein neues Query- Objekt in die Warteschlange gestellt, so wird der ausführende Thread (*ExecuteQuery*) "geweckt"; dieser holt sich dann die Abfrage aus der Queue und führt sie aus. [1] Wurde die Abfrage erfolgreich beendet, so stellt er das Ergebnis in die

[1]Existieren mehrere *Bearbeitungsthreads*, wird nach dem Motto "wer zuerst kommt, mahlt zuerst" ver-

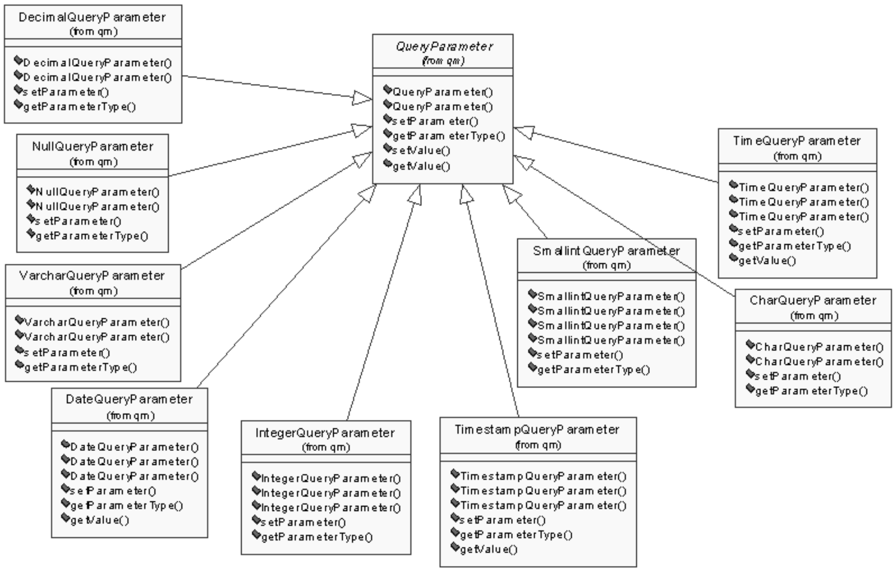


Abbildung 5.2: Abstrakte Klasse QueryParameter und deren Ableitungen

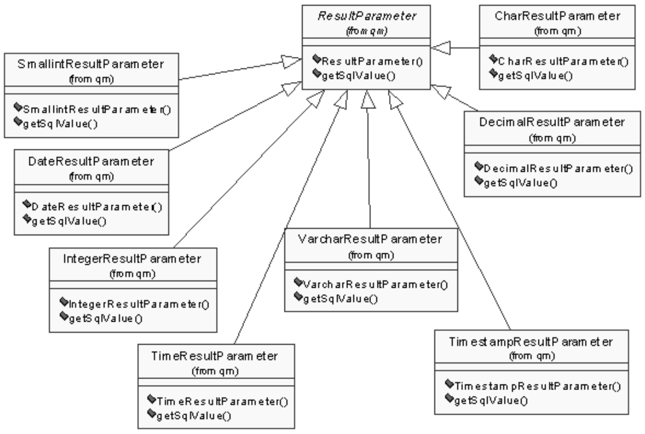


Abbildung 5.3: Abstrakte Klasse ResultParameter und deren Ableitungen

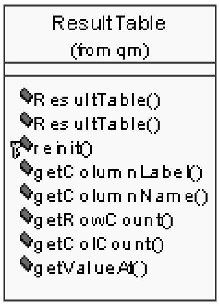


Abbildung 5.4: Abstrakte Klasse ResultParameter und deren Ableitungen

fahren.

ResultQueue. Traten bei der Abfrage Fehler auf, so wird eine Fehlerobjekt (*QueryError*) in die *QueryErrorQueue* gestellt.
Die Anzahl der ExecuteQuery- Threads kann in einem Property- File festgelegt werden, daß beim Initialisieren des SqlDaemons eingelesen wird. Der SqlDaemon erzeugt dann die entsprechende Anzahl von Threads. Abbildung 5.5 zeigt die oben beschriebenen Klassen, die für das Absetzen und Ausführen der Abfragen notwendig sind.

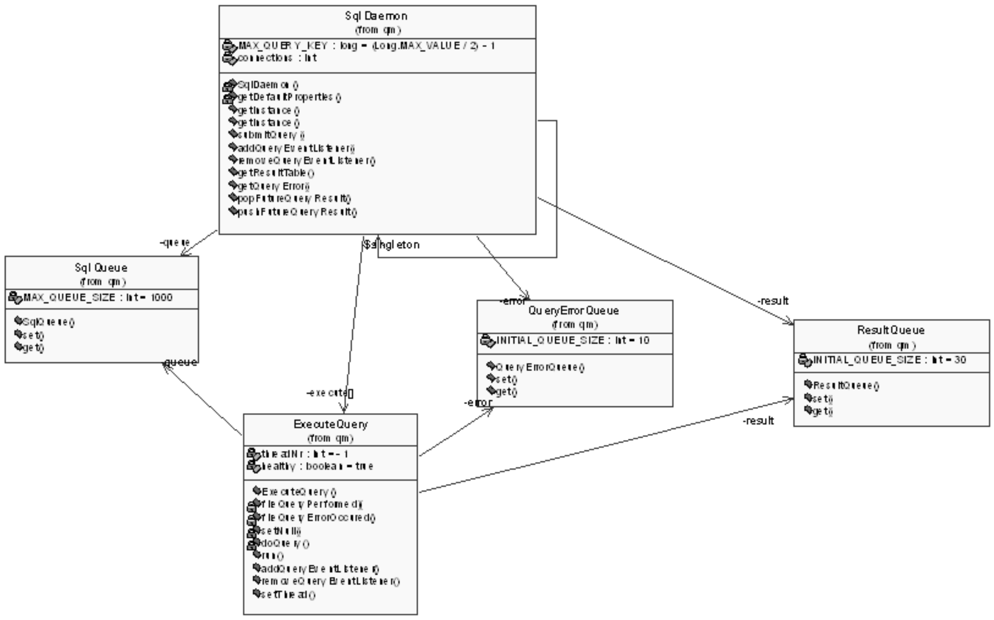


Abbildung 5.5: Modell des SqlDaemon (zentrale Abfragestelle)

Da die Abfragen asynchron ausgeführt werden, kann das aufrufende Objekt nicht feststellen, wann seine Anfrage fertig ist. Um die Abfragen für den Aufrufer wieder zu synchronisieren, wurde das *FutureQueryResult* Object eingeführt. Das FutureQueryResult wird über *EventListener*, über "Ereignissen" in den Warteschlangen informiert. Das FutureQueryResult "wartet" solange im aufrufenden Code- Teil, bis es durch das richtige Ereignis (ResultTable oder QueryError mit der passenden ID in die Warteschlange gestellt) geweckt wird. Abbildung 5.6 zeigt das Objekt mit Listener- Interface.
Durch das Absetzen eines Abfrageobjektes über den SqlDaemon erhält man automatisch ein passendes "Future- Objekt": [2]

```
SqlDaemon daemon = SqlDaemon.getInstance();
FutureQueryResult fqr = daemon.submitQuery(query);
ResultTable rt = fqr.getResultTable();
```

Die Synchronisation der Abfragen wird somit extrem vereinfacht. Den kompletten Code einer Testklasse ist im Abschnitt 5.1.3 zu finden.

[2]Das Exceptionhandling wurde hier weggelassen.

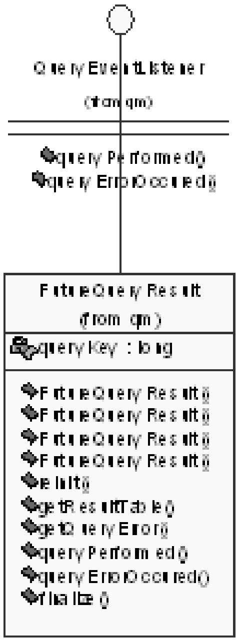


Abbildung 5.6: Synchronisations- Objekt: FutureQueryResult

5.1.3 Umsetzung

Die folgenden Beispielcode demonstriert die Benutzung des QueryMangers. Die Dokumentation der gesamten Paket liegt in Form von *JAVA- DOC* vor.

```
package com.seag.qm.test;

import java.util.*;

import com.seag.qm.*;

public class DoQueryThread extends Object implements Runnable {
    private boolean healthy = true;
    private SqlDaemon daemon = null;

   /** Creates new DoQueryThread */
   public DoQueryThread(SqlDaemon daemon) {
       this.daemon = daemon;
   }

  /**
   * Erzeugen eines Abfrageobjekts
   */
   private SqlQuerySpec createQuery(SqlDaemon daemon) {
      //Das eigentliche Abfrageobjekt
      SqlQuerySpec query;
      //Array von Parametern
      QueryParameter[] params;
      //Array der Ergebnisparameter
      ResultParameter[] resultParams;
```

```
        //Ist Abfrage ein SELECT?
        boolean isSelect;

        //Abfrage SQL- String
        String sql = "SELECT * FROM testdb WHERE id = ?";

        //Abfrage hat einen Parameter (ein Fragezeichen)
        params = new QueryParameter[1];
        params[0] = new IntegerQueryParameter(getRandomId());

        //Abfrage liefert 4 Ergebnisspalten
        resultParams = new ResultParameter[4];
        resultParams[0] = new IntegerResultParameter();
        resultParams[1] = new CharResultParameter();
        resultParams[2] = new CharResultParameter();
        resultParams[3] = new CharResultParameter();

        //Abfrage ist ein SELECT
        isSelect = true;

        //Erzeugen der Abfrage
       query = new SqlQuerySpec(sql, isSelect, params, resultParams);

       return query;
   }

     /**
      * Diese Methode setzt eine zufällig erzeugte
      * Abfrage ab.
      */
      private void doQuery(SqlDaemon daemon) {
         //Erzeugen eines Query- Objektes und Absetzen
         SqlQuerySpec query = createQuery(daemon);
         FutureQueryResult fqr = daemon.submitQuery(query);
         long queryKey = query.getQueryKey();
         System.out.println("Query " + queryKey + " abgesetzt");
         //Warten auf das Ergebnis
         ResultTable[] rt = fqr.getResultTable();
         //Ist ResultTable rt null, ist ein Fehler aufgetreten
         if (rt == null) {
             QueryError qe = fqr.getQueryError();
             //Ausgabe der Fehlermeldung
             System.out.println("Query: " + queryKey +
                                   " fehlgeschlagen." +
                                   qe.getErrorMessage());
         //ist rt != null ist Abfrage gut gegangen
         } else {
         //Abfrage brachte ein Ergebnis
             if (rt[0].getRowCount() < 0) {
                 //Holen des Ergebnisses
                 int id;
                 String text1, text2, text3;
                 for (int i = 0; i < rt[0].getRowCount(); i++) {
                     id = ((Integer)rt[0].getValueAt(i,0)).intValue();
                     text1 = (String)rt[0].getValueAt(i,1);
                     text2 = (String)rt[0].getValueAt(i,2);
```

```
                    text3 = (String)rt[0].getValueAt(i,3);
                }
            }
            System.out.println("Query: " + queryKey + " erfolgreich beendet.");
        }
        //Nach der Abfrage wird das FutureQueryResult- Objekt wieder
        //zurückgestellt; Wiederverwendung
        daemon.pushFutureQueryResult(fqr);
    }

    /**
     * Erzeugen einer zufälligen ID
     */
    private int getRandomId() {
        Random rnd = new Random(System.currentTimeMillis());
        return rnd.nextInt(100000);
    }

    /**
     * Run- Methode des Threads
     */
    public synchronized void run() {
        while (healthy) {
            doQuery(daemon);
        }
    }

    public void exit() {
        healthy = false;
    }
}
```

Kapitel 6

Konstruktion

In der Konstruktionphase wird das System in einer Serie von Iterationen gebaut. Um dies zu veranschaulichen wird nun eine Iteration aus dem Kapitel 4 gewählt und detailierter behandelt. Bei der gewählten Iteration handelt es sich um die Kategorie 3 (Suche und Anzeige von Artefaktdaten).

6.1 Entwicklung

6.1.1 Konzeptionelles Klassendiagramm

Es soll ein Objektmodell entwickelt werden, welches es ermöglicht auf eintreffende Suchanfragen zu reagieren und anhand der vorhandenen Daten und Rechte ein Ergebnis zu erstellen. Dabei sollen die wichtigsten Operationen und Attribute für die Klassen und deren Assoziationen erarbeitet werden.

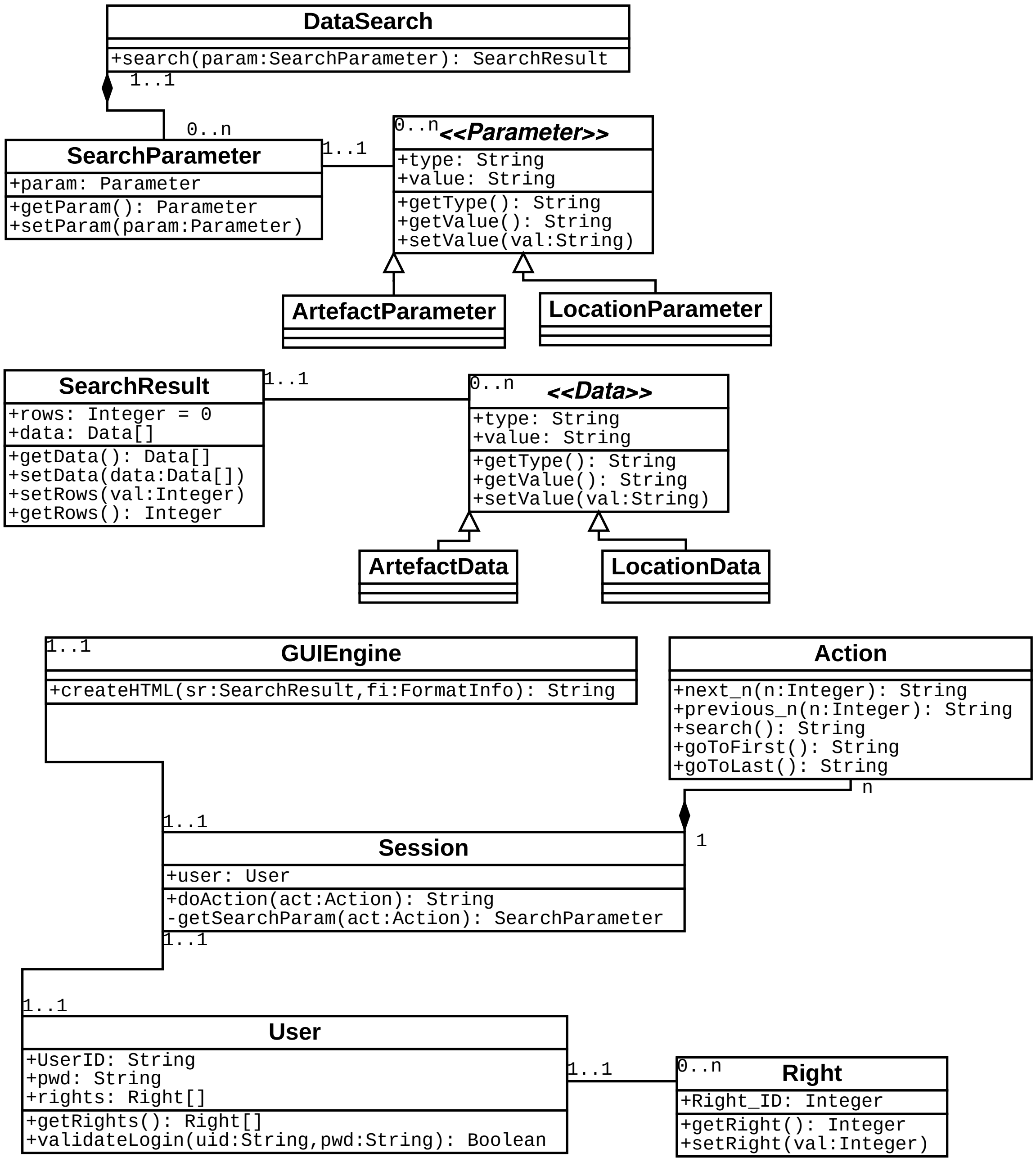


Abbildung 6.1: Konzeptionelles Klassendiagramm: Suche und Anzeige von Artefaktdaten

6.1.2 Beschreibung des Klassendiagramms

Wenn vom Browser eine Anfrage eingeht (*get* oder *post*), wird, wenn noch nicht vorhanden, ein Session-Objekt mit dem vorliegenden Benutzer und dazugehörigen Rechten erzeugt. Jede Session erhält eine eigene Instanz der GUI-Engine. Diese wird zum Parsen/Erstellen des HTTP-Requests/Response verwendet. Aus dem *Request* wird das Action-Objekt gebildet. In unserem Fall bildet das Action-Objekt eine Suchabfrage ab. Aus dem Action-Objekt wird nun ein SearchParameter-Objekt (Rechte werden hier berücksichtigt) erstellt. Daraufhin wird die *search*-Methode des DataSearch-Objekts verwendet um ein SearchResult-Objekt zu erzeugen. Dieses wird nun in der GUI-Engine zum HTTP-Response geparst und liefert die Stringausgabe der *doAction*-Methode. In der GUI-Engine wird die Formatierung des Ergebnisses durch benutzerspezifische Layout-Parameter festgelegt (z.B. als XML-Datei).

6.1.3 Interaktionsdiagramm

Hier wird vor allem der zeitliche Ablauf der ablaufenden Methoden dargestellt.

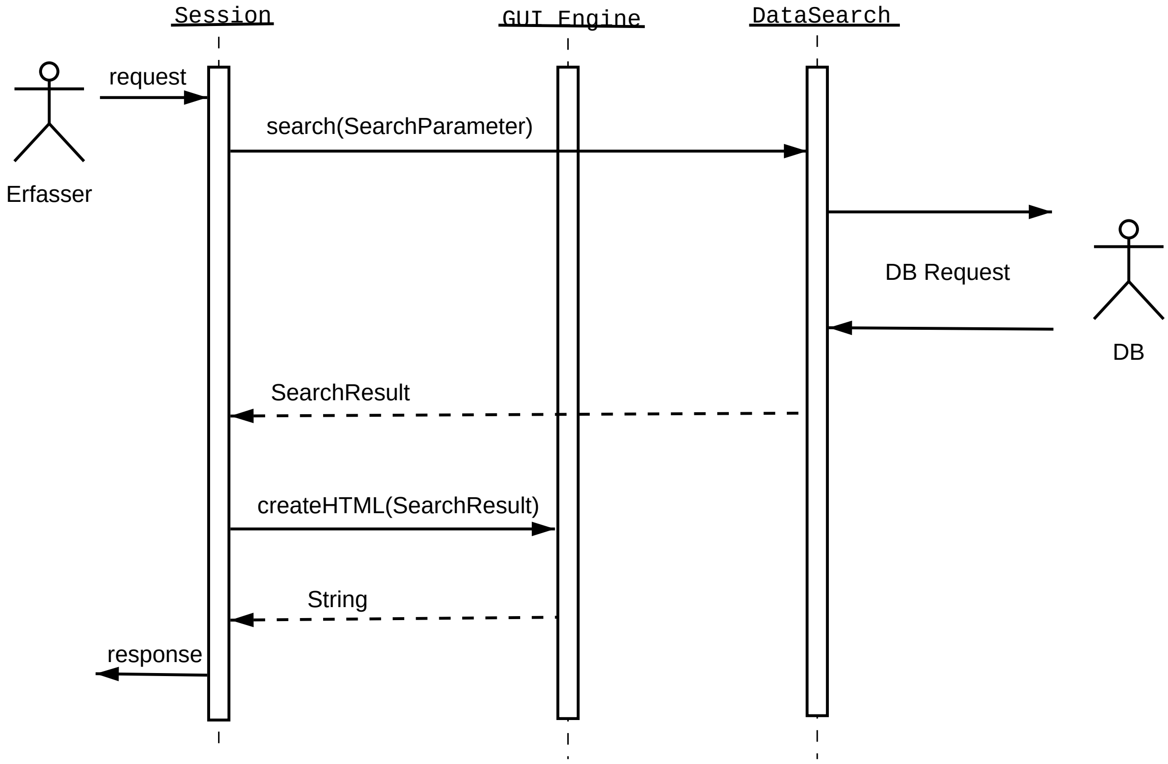


Abbildung 6.2: Sequenz und Kollaboration Diagrammme: Suche und Anzeige von Artefaktdaten

6.1.4 Beschreibung des Interaktionsdiagramms

Interaktionsdiagramme betrachten die dynamischen und zusätzlich die zeitlichen Aspekte des zu entwickelnden Systems und stellen den groben Überblick eines konkreten Programmablaufs unter Berücksichtigung von Interaktionen der Objekte dar. Es muß noch angemerkt werden, daß beim Absetzen einer Suchabfrage ein parametrisierbarer (vom Benutzer abhängig) Timer gestartet wird, der bei einer überschrittenen Response-Zeit der Datenbank die Funktion *cancel* des DataSearch-Objekts ausführt, welche die Abfrage, wenn sich noch in der Queue befindet, aus der Qeue entfernt und den Suchvorgang abbricht. Mit der privaten Methode checkResult der Klasse Session wird nun der Typ des Suchergebnisses analysiert.

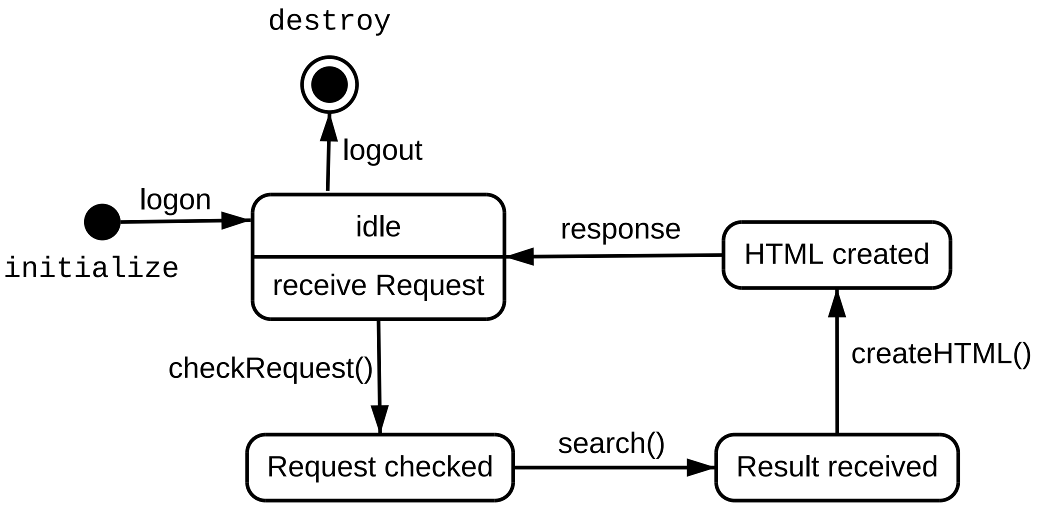


Abbildung 6.3: Statusdiagramm Klasse: Session

6.1.5 Statusdiagramm

6.1.6 Beschreibung des Statusdiagramms

Abhängig von der Art der Aktion sind im Checkzustand verschiedene Szenarien möglich, weshalb auf eine Verschmelzung dieses Zustandes verzichte wurde.

Anhang A

Protokolle

A.1 18.10.2000

1. **Zeitpunkt:** Mittwoch 18.10.2000, von 17:00 Uhr bis 17:50 Uhr.
2. Für das **Prolokoll** verantwortlich: Matthias Ausweger
3. **Teilnehmer** am Treffen:
   - Matthias Ausweger: Vertreter der Entwicklungsabteilung 2 der Fa. SE AG.
   - Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.
4. **Besprochene Themenpunkte:**

   Abgabeformlitäten: Die Arbeits-Unterlagen eines jeden Gruppenmitgliedes sind in getippter, ausgedruckter Form abzugeben.

   übergabe an den nächsten Project-Leader: Nachdem dieses Protokoll in den Report integriert wurde, erfolgt eine Gruppenbesprechung, in der der aktuelle Status-Quo besprochen wird und von dem aus jeder weiterarbeitet. Hierbei erfolgt die übergabe der Projekt-Führung an den nächsten Project-Leader. Wer der nächste Project-Leader sein wird und was die nächste Arbeitsaufgabe ist, kann man aus dem Web ab Donnerstag (19.10.2000) entnehmen.

   Antworten auf Fragen: Es folgen nun die Antworten und Erklärungen zu den im Report gestellten Fragen.

   ad Frage 1: Zielgruppe: Die Zielgruppen sind - wie erwartet - *Erfasser* und *Internet-Besucher*. Sie greifen auf das System in gleichem Ausmaß zu, d.h. vorrangiges Einsatzgebiet ist nicht Erfasser-Seite, sondern beide Gruppen gleichermaßen.

   ad Frage 2: Beziehungen: Wie in unserem Konzept festgestellt, sind Zusammenhänge, wie 'Lage eines Artefaktes in einem bestimmten Grab' Grund-Feststellungen, die als Fakten bezeichnet werden und nur vom Erfasser des Artefaktes eingegeben und bearbeitet werden können.
   Verweise, die darüber hinausgehen, gelten als Interpretationen und werden als *Meta-Daten* eingestuft. Diese Art von Beziehungen kann von jedem wissenschaftlichen Erfasser erstellt werden, sofert er der Erfasser mindestens

eines der archäologischen Objekte (Siedlung, Grab, Artefakt) ist, die in Beziehung gesetzt werden sollen.

ad Frage 3: Suchkriterien: Ein Erfasser kann als Kriterium zur Suche alle Data- Items als Suchkriterium wählen.
Für den Internet-Benutzer sind die Suchkriterien auf die Data- Items beschränkt, die für ihn sichtbar sind. Er kann also nicht nach Data-Items (Attributen) suchen, die er als Internet-Benutzer garnicht angezeigt bekommt.

ad Frage 4: Kriterien anzeigen: Antwort siehe Frage 3.

ad Frage 5: Info- Einschränkung für Internet-B esucher: Es gibt *Basis-Informationen*, die jedem Internet- Besucher angezeigt werden müßen. Diese sind *Bild, Bezeichnung.* Wird ein archäologisches Datum erfaßt, so sind die Basis- Informationen ab Erfassungszeitpunkt im Internet zugänglich. Weitere Informationen, das Datum betreffend, können wahlweise vom Erfasser für den Internet-Benutzer freigegeben werden, oder nicht. Es ist gewollt, daß dem Internet- Benutzer auch Ton-Beispiele zur Verfügung stehen können - Server Resourcen!
Als **Objekt-Daten** sind unsere Data-Items als Vorschlag akzeptiert worden.
Ein **Fundort** besteht aus: *Basis-Fixpunkt* - als GPS-Koordinate, *relative Koordinaten* zum Fixpunkt.

ad Frage 6: Erzeugung VM: Es existiert zu einem Zeitpunkt nur ein VM. Dieses VM wird etwa jedes halbe Jahr neu erstellt
(vom Grundkonzept auf ; Bsp: Stockwerke neu erstellen, entfernen, ...).
Änderungen innerhalb des VM, d.h. innerhalb der Räume können durch die Erfasser täglich erfolgen, was bedeutet, daß änderungs-Updates täglich erfolgen können müssen. Die änderugen durch die Erfasser könnten z.B. auswechseln der Schautafel- Texte im Museum sein.

ad Frage 7: Löschen von archiolog. Daten: Ist auch durch den Erfasser nicht möglich (Bearbeitung ja).

ad Frage 8: Data-Items: Die von uns vorgeschlagenen Data-Items wurden akzeptiert. Frage der Fundort- Beschreibung siehe Antwort zu Frage 5.

ad Frage 9: Anzeige der Daten für Internet-Besuscher: Die Anzeige der Informationen zu einem Datum können differieren, da der Erfasser zu jeder Information,das spezielle Artefakt betreffend, angeben kann, ob es für den Internet-Besucher angezeigt werden soll, oder nicht.

ad Frage 10: Löschen von wissenschaftlichen Arbeiten: Ist nur dem Besitzer der wissenschaftlichen Arbeit erlaubt. Will ein Erfasser z.B. eine Publikation bei seinem Artefakt entfernt haben, so mußer dieses dem Besitzer der Publikation mitteilen. Dieser kann dann die Publikation selbst entfernen, der Erfasser darf das nicht.

Suchmaschine: Es wurde gewünscht, daß dem Internet- Benutzer ein *Dummy-Eingabefeld* (ähnlich dem Eingabefeld bei einer Internet- Suchmaschine) zur Verfügung steht. Hier muß die Einschränkung der Kriterien und der Suchzeit durch die Software erfolgen, da sonst die gesamte DB bei jeder Abfrage durchsucht werden muß. Das heißt im speziellen Fall, daß bei Eingabe von [Pfeiferl]

die ganze DB nach dem String 'Pfeiferl' durchsucht wird.
Anregung durch Herrn Schwaiger: Redundante Tabellen. Als Wunschkriterium kann eine *Experten-Suche* als Alternative durchgeführt werden.

Abschließend sei zusagen, daß der Vertreter des (unbekannten) Kunden mit dem Status der Entwicklung recht zufrieden war.

A.2 31.10.2000

Inhaltlich waren alle vorgenommenen Änderungen OK. Überarbeitet wurden die Benutzerrechte und deren Einteilung in Benztergruppen. Hauptkritikpunkt war die Form der Änderungen. Es ist gefordert, die Verfeinerung der Anforderung, in der Form wie sie vorgegeben waren aufzulisten. Die Auflistung der Anforderungen wird noch nachgereicht.

A.3 08.11.2000

1. **Zeitpunkt:** Mittwoch 08.11.2000, von 16:00 Uhr bis 16:30 Uhr.
2. Für das **Prolokoll** verantwortlich: Werner Weiser
3. **Teilnehmer** am Treffen:
   - Werner Weiser: Vertreter der Entwicklungsabteilung 2 der Fa. SE AG.
   - Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.
4. **Besprochene Themenpunkte:**

   Abgabeformlitäten: Die Arbeits-Unterlagen eines jeden Gruppenmitgliedes sind in mit Latex getippter, ausgedruckter Form abzugeben. Da am Ende des Projekts ohnehin jeder seine Mitschrift abgeben muß, ist dies nur eine vorgezogene Arbeit.

   übergabe an den nächsten Project-Leader: Nachdem dieses Protokoll in den Report integriert wurde, erfolgt eine Gruppenbesprechung, in der der aktuelle Status-Quo besprochen wird und von dem aus jeder weiterarbeitet. Hierbei erfolgt die übergabe der Projekt-Führung an den nächsten Project-Leader. Wer der nächste Project-Leader sein wird und was die nächste Arbeitsaufgabe ist, kann man aus dem Web ab Donnerstag (09.11.2000) entnehmen.

   Struktur: Es ist unbedingt notwendig, zur Lösung der Aufgaben, die dafür vorgesehenen Strukturen zu verwenden, um eine Beurteilung der Arbeit zu erleichtern.

   Use Cases: Die von mir abgegebenen Kurzbeschreibungen waren wie erwartet zu lang. Da jedoch durch die folgenden Iterationsschritte (Flow of Events, Preconditions usw.) Informationen aus den Kurzbeschreibungen zu entnehmen sind, werden diese dadurch gekürzt und stehen nach diesen Iterationsschritten in kompakter Form zur Verfügung.

A.4 17.11.2000

1. **Zeitpunkt:** Freitag 17.11.2000, von 9:00 Uhr bis 9:30 Uhr.
2. Für das **Prolokoll** verantwortlich: Kasbi M. Jamal
3. **Teilnehmer** am Treffen:
    - Kasbi M. Jamal: Vertreter der Entwicklungsabteilung 2 der Fa. SE AG.
    - Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.
4. **Besprochene Themenpunkte:**

    übergabe an den nächsten Project-Leader: Nachdem dieses Protokoll in den Report integriert wurde, erfolgt eine Gruppenbesprechung, in der der aktuelle Status-Quo besprochen wird und von dem aus jeder weiterarbeitet. Hierbei erfolgt die übergabe der Projekt-Führung an den nächsten Project-Leader. Wer der nächste Project-Leader sein wird und was die nächste Arbeitsaufgabe ist, kann man aus dem Web ab Donnerstag (20.11.2000) entnehmen.

    Gewichtung der technischen Faktoren: Es wurde darauf acht genohmmen, wie wir die Wiederverwendbarkeit des Codes und die Portabilitaet des System Bewertet haben.Mit der Portabilitaet ist hier,die Geasmtanpassung des Systems zu einer anderen DB gemeint,d.h.die flexibilitaet der Systemschnittstelle.
    Die Wiederverwendbarkeit des Codes ist eine Anforderung die sich selbst aus der Anfordrungen bei der Software Entwicklung imliziert,d.h.,dass es sehr wichtig ein Code zu schreiben,dass sich leicht zu anderen Systemen portieren lässt. Ausserdem es wurde erwaehnt vom Mag.D.I.R.Schwaiger,dass so ein System no nicht im Markt gibt,desswegen soll es portierbar und wiederverwendbar.

    Ausswertung unserer Arbeit Unser Kosten- und Zeitabschaetzungen wurden als realistisch betrachtet,es ist aber darauf zu achten,dass dies nur eine grobe Abschaetzung ist,die im Monat Januar wiederholt wird. Dabei wird dem Kunde die entscheidung über das Kaufen unser Pruduktes ueberlassen.

    Zusammenfassung: Unser Arbeit wurde Positiv akzeptiert und als realistisch betrachtet,nun liegt die Entscheidung bei dem Kunde über das Kaufen unseres Produktes.

    BEMERKUNG Ich schlage vor, dass wir in Zukunft die Struktur des Dokummentes nicht mehr aendern, da der Gruppen-Leader sonst viel zu bearbeiten hat.

A.5 20.12.2000

1. **Zeitpunkt:** Mittwoch 20.12.2000, von 16:00 Uhr bis 16:30 Uhr.
2. Für das **Prolokoll** verantwortlich: Matthias Ausweger
3. **Teilnehmer** am Treffen:
    - Matthias Ausweger: Vertreter der Entwicklungsabteilung 2 der Fa. SE AG.

- Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.

4. **Besprochene Themenpunkte:**

Architektur: Beim abgegebene Dokument-Teil wurden keine Kritikpunkte geäußert.

Projektplan:
- Die Implementierung der Architektur-Schichten fünf bis sieben ist aus der Auflistung der Iterationen zu entfernen, da diese der Implementierung der Kategorien (Use Cases) entspricht.
- Da der Kunde mit uns ein schnelleres Feedback haben will, sind die Use Cases in mehrere Kategorien einzuteilen. So ist es möglich, in kürzeren Abständen Use Cases und Feedback auszutauschen. Es gibt keinen Informationsstau.
- Die Iterationslänge ist noch einmal zu überdenken, da die Implementierung z.B. eines kleineren Use Cases nicht 23 Mann-Wochen dauern wird.

Allgemein:
- Die Form des Abgabedokumentes war so weit in Ordnung.
- Kleine Überarbeitungen des Projektplan-Bereiches wurden in diesem Dokument bereits vorgenommen.
- Das Dokument der Aufgabe 6 wird als Beispiel-Dokument ins Netz gehängt.

A.6 11.01.2001

1. **Zeitpunkt:** Donnerstag 11.01.2001, von 13:00 Uhr bis 13:45 Uhr.
2. Für das **Prolokoll** verantwortlich: Franz Wilhelmstötter
3. **Teilnehmer** am Treffen:
   - Franz Wilhelmstötter: Vertreter der Entwicklungsabteilung I der Fa. SE AG.
   - Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.
4. **Besprochene Themenpunkte:**

Prototyp: Besprechung und Vorführung des Funktionsumfanges und der Funktionsweise des implementierten Prototypen. Das Gespräch war sehr zufriedenstellend

A.7 25.01.2000

1. **Zeitpunkt:** Donnerstag 25.01.2001, von 13:45 Uhr bis 14:15 Uhr..
2. Für das **Prolokoll** verantwortlich: Werner Weiser
3. **Teilnehmer** am Treffen:
   - Werner Weiser: Vertreter der Entwicklungsabteilung 2 der Fa. SE AG.
   - Mag. Dipl.-Ing. Schwaiger Roland: Vertreter des (unbekannten) Kunden.
4. **Besprochene Themenpunkte:**

   Klassendiagramm: Das Klassendiagramm wurde bis auf einige Details für korrekt befunden.

   Interaktionsdiagramm: Da das Sequenzdiagramm zum Verständnis des Ablaufs ausreichend ist, wird auf ein Kollaborationsdiagramm verzichtet.

   Statusdiagramm: Das Statusdiagramm für die Klasse Session soll sich nur auf die Methode search beschränken.