PS Software Engineering

WS 2000/2001 2. Februar 2001

Softwaresystem zur Erfassung, Archivierung und Präsentation von archäologischen Daten

Gruppe 7

Zehentner Hildegard hzehent@cosy.sbg.ac.at

Röck Andrea aroeck@cosy.sbg.ac.at

Osterrieder Christian coster@cosy.sbg.ac.at

Mayr Manfred mmayr@cosy.sbg.ac.at

Angulanza Florian fangulan@cosy.sbg.ac.at

Projektleiter: Mag. Dipl.-Ing. Schwaiger Roland

Institut für Computerwissenschaften
Universität Salzburg

Korrespondenz an:

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

Zusammenfassung

Unser Unternehmen, die SE AG, soll ein Softwaresystem zur Erfassung, Archivierung und Präsentation von archäologischen Daten entwickeln. Dabei soll es sich um eine zentrale Datenbank handeln, die über ein Web-Frontend gepflegt und abgefragt werden kann. Präsentiert werden sollen die Daten verpackt in sogenannte virtuelle Museen.

Inhaltsverzeichnis

Kapitel 1

Ausgangssituation und Zielsetzung

1.1 Projektbeschreibung

Das Softwareprodukt soll dem Auftraggeber ermöglichen, archäologische Daten wie Siedlungen, Gräber oder Artefakte über ein Web-Frontend in einer zentralen Datenbank zu erfassen, zu verwalten und zu präsentieren. Für die Präsentation kann der Internetuser ein virtuelles Museum besuchen, wo er nach archäologischen Daten suchen und diese betrachten kann. Es kann mehrere virtuelle Museen geben, eines ist jedoch Pflicht. Von diesem virtuellen Museum können CD-ROMs (z. B. Sonderschauen) erstellt und Kataloge gedruckt werden.
Für die Verwaltung der Daten wird XML verwendet, als Grundlage für die Systementwicklung wird der Rational Unified Process verwendet.
Das Softwareprodukt soll multilingual konzipiert sein. Später soll es insgesamt Versionen in Englisch, Spanisch, Italienisch, Deutsch und Chinesisch geben.

1.2 Datenhaltung

Die Datenhaltung soll zentral sein, d. h. unser Unternehmen tritt als Application Service Provider auf. Von unserem Unternehmen werden geeignete Werkzeuge bereitgestellt, um die Datenbank zu warten, Backups zu erstellen, aber auch um die Programme zu warten, neue Versionen zu erstellen oder Fehler zu beheben.

1.2.1 Daten

Ein archäologisches Objekt besitzt Grunddaten wie Inventarnummer, Bezeichnung, Alter, Fundort, Material, Dimensionen des Objekts und Freitext. Zusätzlich können Bilddateien und Videos, Publikationen, Interpretationen oder Literaturverweise einem Objekt zugeordnet werden. Auch können Verbindungen zu anderen Objekten aufgebaut werden. Wenn sich etwa mehrere Artefakte in einem Grab befinden, kann eine Verbindung von den Objekten zu dem Grab aufgebaut werden.
Die eingegeben Daten sollen ab dem Zeitpunkt der Zuordnung über das Web-Frontend verfügbar sein.
Es können einzelne Daten oder ganze Datensätze eines archäologischen Objekts als *"sichtbar"* oder *"nicht sichtbar"* für den Internetuser gekennzeichnet werden.
Publikationen und Interpretationen werden als pdf-Dateien gespeichert. Für Videodateien wird das real audio Format vorgegeben und als Bilddateien kann man gif- und jpeg-Dateien abspeichern.
Die Daten werden multilingual abgespeichert. Es soll eine internationale Datenbank eingerichtet werden. Die Daten werden von einem Dolmetscher in die entsprechenden Sprachen übersetzt.

Die Daten müssen permanent verfügbar sein, da jederzeit Anfragen an die Datenbank über das Internet kommen können. Die Daten sollen in XML verwaltet werden, um die Inhalte auf einer breiten Palette von Endgeräten präsentieren zu können.

1.3 Gruppen

1.3.1 Objektgruppen

Der Verwalter kann Objektgruppen erstellen. Mögliche Objektgruppen wären etwa alle Objekte zu einer Epoche, zu einem Entdecker oder zu einem bestimmten Grab.
Bei der Erstellung eines neuen Objektes wird zuerst eine Gruppe ausgewählt, zu der das Objekt gehört, bevor die Daten eingegeben werden können.
Jedes Objekt muss zu mindestens einer Gruppe gehören, es kann aber auch zu mehreren Gruppen gehören. Ein Objekt kann nachträglich noch einer weiteren Gruppe zugewiesen werden oder eine Gruppe kann gelöscht werden, wobei aber immer eine Gruppe übrig bleiben muss.

Über die Zugehörigkeit eines Objekts zu einer Gruppe wird geregelt, welche Mitarbeiter das Recht haben, dieses Objekt zu bearbeiten.

1.3.2 Benutzergruppen

1.3.2.1 Internetuser

Der Internetuser hat die wenigsten Rechte. Er kann das virtuelle Meseum besuchen oder sich bestimmte Daten über die Datensuche anzeigen lassen. Er kann jedoch nur solche Daten sehen, die vorher von der Gruppe der Erfasser bzw. wissenschaftlichen Erfasser freigegeben worden sind.

1.3.2.2 Verwalter

Es gibt die Gruppe des Verwalters, der alle Rechte besitzt. In unserem Fall wird der Verwalter durch den Auftraggeber repräsentiert.
Er hat insbesondere das Recht, Funktionalitäten zu erstellen und zu verändern, Benutzergruppen und Objektgruppen zu erstellen und zu löschen und neue Museen zu erzeugen oder zu löschen.

1.3.2.3 Mitarbeiter

Es gibt keine vorgefertigten Gruppen der Mitarbeiter. Es gibt nur Funktionalitäten. Der Verwalter kann aus diesen Funktionalitäten und den Objektgruppen Benutzergruppen anlegen. Diesen Benutzergruppen werden dann Mitarbeiter zugewiesen. Die Rechte einer erstellten Benutzergruppe kann nachträglich geändert und erweitert werden. Einige mögliche Funktionalitäten wären z. B.:

- Grunddaten eingeben
- Grunddaten bearbeiten
- Grunddaten löschen
- Publikationen erstellen
- Räume erstellen
- Vitrinen gestalten
- Vitrinen löschen
- CD-ROM erstellen

- CD-ROM Daten auswählen
- etc.

Die Unterteilung der einzelnen Rechte kann sehr fein sein. Es können jederzeit neue Rechte vom Verwalter definiert werden. Falls Rechte verändert oder gelöscht werden und diese Rechte in Benutzergruppen verwendet werden, so erscheint eine Warnung. Jede Benutzergruppe muss zu jedem Zeitpunkt mindestens 1 Berechtigung besitzen.
Es wird ein Tool bereitgestellt, welches in einfacher Weise ermöglicht, neue Funktionalitäten zu erstellen oder alte zu verändern oder zu löschen. Ebenfalls wird ein Tool bereitgestellt, welches in einfacher Weise ermöglicht, eine Benutzergruppe zu erstellen. Die vorhandenen Funktionalitäten werden ebenso wie die vorhandenen Objektgruppen bei der Erstellung einer neuen Benutzergruppe angezeigt, so dass man sie ganz einfach auswählen kann.

1.4 Benutzerschnittstellen, Präsentation der Daten

Die Datenverarbeitung, wie Museum, Katalog oder CD-ROM erstellen, und die Datenverwaltung, wie Eingabe, Änderung und das Löschen der Daten sowie der Verweise auf andere Daten oder wissenschaftliche Ergänzungen, erfolgt über eine andere Internetseite als jene, die dem Internetuser zur Verfügung steht.

1.4.1 Web-Frontend für Datenverwaltung und Datenverarbeitung

Will man auf die Seite für die Datenverwaltung und Datenverarbeitung gelangen, muss sich die entsprechende Person mittels Passwort identifizieren.

1.4.1.1 Datenverwaltung

Es können Daten mit Hilfe dieses Teils des Programmes eingegeben, geändert oder gelöscht werden. Eine Eingabe erfolgt immer in Bezug auf eine Objektgruppe (siehe 1.3.1). Jeder Mitarbeiter darf nur im Bereich seiner Berechtigungen handeln.

1.4.1.2 Verwaltung

Über diesen Teil des Programmes kann der Verwalter neue Funktionalitäten eingeben, alte verändern oder löschen.
Er kann neue Objektgruppen erstellen, alte verändern oder löschen.
Er kann aus den Funktionalitäten und den Objektgruppen neue Benutzergruppen erstellen.
Er kann einen Mitarbeiter anmelden, ihm einen Benutzernamen und ein vorläufiges Passwort geben. Ebenfalls kann er helfen, wenn ein Mitarbeiter sein Passwort vergessen haben sollte oder wenn er gesperrt wurde, indem er sein Passwort 3 mal falsch eingegeben hat. Er kann einen Mitarbeiter zu einer oder mehreren Benutzergruppen zuordnen.
Der Verwalter hat ebenfalls die Möglichkeit, neue Templates für die Gestaltung des Museum hereinzuladen oder alte zu löschen.
Der Verwalter hat das Recht, ein neues Museum zu erstellen oder ein altes zu löschen. Es muss aber immer mindestens 1 Museum existieren.

1.4.1.3 Museumsgestaltung

Ein Mitarbeiter, welcher über die entsprechenden Rechte verfügt, kann in einem Museum Schauobjekte (Stockwerke, Räume, Vitrinen) erstellen, verändern oder löschen.
Für die Gestaltung der Stockwerke, Räume und Vitrinen wird eine Art Wizard bereitgestellt, welcher den Mitarbeiter durch die Erstellung führt. Der Mitarbeiter kann für die Gestaltung aus mehreren Templates auswählen, welche jeweils einer bestimmten Gestaltungsvariante entsprechen. Diese Templates können vom Verwalter jederzeit nachgeladen oder gelöscht werden.

1.4.1.4 CD-ROM

Es sollen Werkzeuge entwickelt werden, die es ermöglichen, eine 1:1 Abbildung des virtuellen Museums über das Web-Frontend auf eine CD-ROM zu schreiben. Das Tool ermöglicht es, jene Bereiche des Museums zu markieren, welche auf CD-ROM gebrannt werden sollen. Das viruelle Museum auf der CD-ROM funktioniert wie das des Web-Frontends, nur dass die Daten, die angezeigt werden, sich auf der CD-ROM befinden und nicht aus der Datenbank geholt werden. Es werden auf der CD-ROM nur jene Daten gespeichert, welche für die Abbildung des Museums benötigt werden und die für den Internetuser als sichtbar gekennzeichnet sind.

1.4.1.5 Katalog

Es werden Werkzeuge entwickelt, die es dem Auftraggeber (Verwalter) ermöglichen, Kataloge zu den virtuellen Museen zu erstellen. Die Werkzeuge müssen es ermöglichen, Daten für den Druck aktiv auszuwählen. Es sind in den Werkzeugen Funktionen integriert, die automatisch die für den Druck geeigneten Informationen selektieren und entsprechend aufbereiten. Es wäre unsinnig zu versuchen, Videodateien in gedruckter Form zu präsentieren, andererseits können kurze Interpretationen oder Zusammenfassungen von Publikationen, die eine bestimmte Maximumsgrösse nicht überschreiten, zum Druck geeignet sein.
Das Text- bzw. Grafikformat für den Katalog ist das pdf-Format.

1.4.2 Web-Frontend für den Internetuser

1.4.2.1 Das virtuelle Museum

Aus den erfassten Daten soll ein virtuelles Museum aufgebaut werden, welches über die Homepage des Auftraggebers aus besucht werden kann. Es kann mehrere Museen geben, eines ist aber Pflicht. Jedes Museum besteht aus Stockwerken. Es kann etwa ein Stockwerk geben, in welchem sich nur Fibeln befinden. Zu jedem Stockwerk gibt es eine kurze Beschreibung zum Inhalt. Als erstes findet man die Auflistung der einzelnen Stockwerke mit deren jeweiligen Kurzbeschreibungen. Wählt man ein Stockwerk aus, kommt man zur Auflistung der Räume, welche sich in diesem Stockwerk befinden. In jedem Stockwerk muss sich mindestens ein Raum befinden, es können sich natürlich auch mehrere Räume in einem Stockwerk befinden. Zu jedem Raum gibt es eine kurze Beschreibung, was der Raum enthält. Ein Raum kann z. B. einer Epoche gewidmet sein. Die Beschreibung der Räume erscheint ebenfalls bei der Auflistung der Räume. Jeder Raum enthält Vitrinen. Ein Raum kann auch nur eine Vitrine enthalten. Jede Vitrine kann eine kurze Beschreibung über ihren Inhalt enthalten, muss aber nicht. Wählt man eine Vitrine aus, werden verschiedene Objekte mit ihren Bezeichnungen und optional mit einem kleinem Bild dargestellt.
Wählt man ein bestimmtes Objekt aus, werden die für den Internetuser als sichtbar gekennzeichneten Daten dargestellt. Sehr große Bilder oder Videos, sowie Verbindungen zu anderen Objekten, Publikationen, Interpretationen oder Literaturverweise, werden, wenn die entsprechenden Daten in der Datenbank verfügbar und als sichtbar gekennzeichnet sind, als Verbindungen dargestellt. Diese Verbindungen kann man ebenfalls auswählen, um sich darüber genauer zu informieren bzw. die Daten (Bild oder Video) zu betrachten oder herunterzuladen, wenn dies erlaubt ist. Die Daten, die angezeigt wer-

den, sind nicht starr, sondern werden laufend aus der Datenbank generiert.

1.4.2.2 Datensuche

Neben dem Besuch des virtuellen Museums soll es dem Internetuser auch möglich sein, die Datenbank nach bestimmten archäologischen Objekten zu durchsuchen. Auch bei der Suche werden nur die als sichtbar gekennzeichneten Daten der archäologischen Objekte berücksichtigt.
Es gibt 2 Suchebenen:

- einfache Suche: Der Internetuser gibt z.B. den Suchbegriff Schwert ein; es wird ihm alles aufgelistet, wo dieser Begriff vorkommt. Dabei wird allerdings nur die Ebene 1 durchsucht, d. h. es werden nur die festen Attribute und keine Verweise auf Literatur oder andere Objekte durchsucht.

- erweiterte Suche: Die erweiterte Suche soll es dem Internetbesucher erlauben, seine Suche weiter einzugrenzen. Er soll z. B. nur Beschreibungen durchsuchen oder nur Speere zu seinem eingegebenen Suchbegriff suchen.

Alle eingegebenen Suchbegriffe können boolsch verknüpft werden (also mit UND, ODER, NICHT).
Die Ergebnisse der Suche werden als Liste dargestellt bzw. im virtuellen Museum markiert. Es wird ebenfalls der Pfad zu den markierten Objekten angezeigt, d. h. es wird das Museum, das Stockwerk, der Raum und die Vitrine, in dem sich das Objekt befindet, ebenfalls markiert. Der Benutzer kann ein Ergebnis auswählen. Ihm werden dann alle für den Internetuser als sichtbar gekennzeichneten Attribute angezeigt. Große Bilder oder Videos sowie Verbindungen zu anderen Objekten, Publikationen, Interpretationen oder Literaturverweise werden, wenn die entsprechenden Daten in der Datenbank verfügbar und als sichtbar gekennzeichnet sind, als Verbindungen dargestellt. Diese Verbindungen kann man ebenfalls auswählen, um sich darüber genauer zu informieren bzw. die Daten (Bild oder Video) zu betrachten oder herunterladen, wenn dies erlaubt ist. Die Daten, die angezeigt werden, sind nicht starr, sondern werden laufend aus der Datenbank generiert.

Kapitel 2

Systemeinsatz und Systemumgebung

2.1 Systemeinsatz

Das Softwareprodukt ermöglicht die Eingabe, Verwaltung und Präsentation von archäologischen Daten über ein Web-Frontend.
Es gibt eigentlich 2 Web-Frontends:

- Eines für den Internetuser, der das virtuelle Museum betrachten und die Daten der Datenbank durchsuchen kann.
- Eines für die Mitarbeiter.
  Um dieses Web-Frontend nutzen zu können, muss man sich erst identifizieren. Anschließend kann man seinen Rechten entsprechend arbeiten. (Daten eingeben, das Museum gestalten oder Kataloge erstellen)

Das Produkt und die verwendete Datenbank müssen durchgehend in Betrieb sein, d.h. das Arbeiten mit dem Produkt soll zu jeder Zeit möglich sein.

2.1.1 Zielgruppen

Es gibt 3 Zielgruppen:

- Der Internetuser als Betrachter der ihm gebotenen Daten
- Die Mitarbeiter, welche die Daten eingeben und verwalten und aus ihnen Museum, Kataloge und CD-ROMs gestalten.
- Der Verwalter, welcher mit Hilfe dieses Produkts die Museen verwaltet, neue Funktionalitäten und neue Gruppen erstellt und verwaltet und Mitarbeitern Gruppen zuweist.

2.2 Systemumgebung

2.2.1 Hardware

Es muss sowohl für uns, als Server und Datenbankverwalter, als auch beim Verwalter und bei den Museumsgestaltern eine durchgehende breitbandige Internetanbindung vorhanden sein.
Auch die Mitarbeiter müssen über einen geeigneten Internetanschluss verfügen.
Es muss genügend Speicherplatz vorhanden sein, um alle archäologischen Daten sowie Bilder und Videos abzuspeichern. Es muss einfach sein, den vorhandenen Speicherplatz zu erweitern.

2.2.2 Software

Die Benutzer dieses Softwareprodukts müssen über einen Java 1.01 fähigen Browser verfügen. Es wäre wünschenswert, wenn die Browser XML unterstützen würden (heutzutage noch selten).
Es müssen entsprechende Tools vorhanden sein (serverseitig), um ein XML Dokument darzustellen (entweder direkt, wenn der Browser dies kann, oder die Seiten müssen in HTML umgewandelt werden)
Alle Anwender des Software-Produkts müssen mit den Dateiformaten gif, jpeg, real audio und pdf arbeiten können oder wenigstens betrachten können, um einen problemlosen Gebrauch des Software-Produkts zu gewährleisten. Es muss sichergestellt werden, dass nur berechtigte Personen Daten eingeben, ändern oder löschen können.

2.2.3 Datenbank

Die Datenbank muss permanent verfügbar sein und auch noch bei bis zu mehreren hundert Anfragen gleichzeitig die Datenkonsistenz garantieren.

Kapitel 3

Funktionale Anforderungen

3.1 Internetuser

- FI10: Der Internetuser kann ein Museum auswählen, welches er näher betrachten will.
  Priorität: 1

- FI20: Der Internetuser kann ein Sockwerk auswählen, welches er näher betrachten will.
  Priorität: 1

- FI30: Der Internetuser kann einen Raum auswählen, welchen er näher betrachten will.
  Priorität: 1

- FI40: Der Internetuser kann eine Vitine auswählen, welche er näher betrachten will.
  Priorität: 1

- FI50: Der Internetuser kann ein archäologisches Objekt in einer Vitrine auswählen.
  Priorität: 1

- FI60: Der Internetuser kann für die einfache Suche ein Suchojekt oder mehrere durch boolsche Verknüpfungen verbundene Suchobjekte eingeben. Dach diesem Suchobjekt oder Verbindung von Suchobjekten wird in den Grundattributen der Datenbank geforscht.
  Priorität: 2

- FI70: Der Internetuser kann in der erweiterten Suche spezifizieren, in welchen Bereichen nach seinem Suchobjekt oder Verknüpfung von

Suchobjekten geforscht werden soll.
Priorität: 3

- FI80: Der Internetuser kann ein Objekt aus der Liste auswählen, welche nach dem Suchvorgang erscheint.
  Priorität: 1

- FI90: Der Internetuser kann Verbindungen von Objekten verfolgen, solange die Objekte sichtbar geschalten sind.
  Priorität: 2

3.2 Museum

- FM10: Der Verwalter kann ein neues Museum erstellen.
  Priorität: 1

- FM20: Der Verwalter kann ein Museum löschen.
  Priorität: 3

- FM30: Das Museum kann nur gelöscht werden, wenn noch ein anderes Museum vorhanden ist.
  Priorität 3

- FM40: Ein Stockwerk kann erzeugt werden.
  Priorität: 1

- FM50: Ein Stockwerk kann gelöscht werden.
  Priorität: 2

- FM60: Ein neuer Raum kann in einem Stockwerk erstellt werden.
  Priorität: 1

- FM70: Ein Raum kann gelöscht werden.
  Priorität: 2

- FM80: Eine Vitrine kann in einem Raum erstellt werden.
  Priorität: 1

- FM90: Eine Vitrine kann gelöscht werden.
  Priorität: 2

- FM100: Es muss ein Datensatz ausgewählt werden, um ein Objekt in einer Vitrine darzustellen.
  Priorität: 1

- FM110: Es muss ein Gestaltungs-Template zur Darstellung eines Objekts in einer Vitrine ausgewählt werden.
Priorität: 2

- FM120: Es kann ein Objekt aus einer Vitrine gelöscht werden.
Priorität: 2

- FM130: Es kann ein neues Template für die Gestaltung der Vitrinen hereingeladen werden.
Priorität: 3

- FM140: Es kann ein Template für die Gestaltung der Vitrinen gelöscht werden.
Priorität: 4

- FM150: Abfrage, ob in jedem Stockwerk sich mindestens ein Raum befindet.
Priorität: 4

- FM160: Abfrage, ob sich in jedem Raum mindestens eine Vitrine befindet.
Priorität: 4

- FM170: Abfrage, ob sich in jeder Vitrine mindestens ein Objekt befindet.
Priorität: 4

- FM180: Es kann ein beschreibender Text zu einem Stockwerk bearbeitet bzw. eingegeben werden.
Priorität: 2

- FM190: Es kann ein beschreibender Text zu einem Raum bearbeitet bzw. eingegeben werden.
Priorität: 2

- FM200: Es kann ein beschreibender Text zu einer Vitrine bearbeitet bzw. eingegeben werden.
Priorität: 2

3.3 Dateneingabe und Datenverwaltung

- FD10: Es kann zu einer Objektgruppe ein neues Objekt erstellt werden. Es muss eine Objektgruppe ausgewählt sein.
Priorität: 1

- FD20: Wurde ein Objekt erstellt, müssen die Grunddaten der Objekte eingegeben werden.
Priorität: 1

- FD30: Es können die Grunddaten der Objekte geändert werden
Priortiät: 2

- FD40: Es kann ein Objekt gelöscht werden.
Priorität: 2

- FD50: Es kann die Verbindung zu einem anderen Objekt hergestellt werden.
Priorität: 1

- FD60: Es kann die Verbindung zu einem anderen Objekt gelöscht werden.
Priorität: 2

- FD70: Es kann von einem Objekt eine Verbindung zu einer Datei, wie etwa einer Bild- oder Videodatei, einer Interpretation oder einer Publikation hergestellt werden.
Priorität: 2

- FD80: Es kann von einem Objekt eine Verbindung zu einer Datei, wie etwa einer Bild- oder Videodatei, einer Interpretation oder einer Publikation gelöscht werden.
Priorität: 3

- FD90: Es kann ein Literaturverweis zum Objekt hinzugefügt werden.
Priorität: 2

- FD100: Es kann ein Literaturverweis von einem Objekt gelöscht werden.
Priorität: 3

- FD110: Ein Objekt kann einer zusätzlichen Objektgruppe zugeordnet werden.
Priorität: 5

- FD120: Wenn ein Objekt zu mehr als einer Objektgruppe gehört, kann die Zugehörigkeit zu einer Objektgruppe gelöscht werden.
Priorität: 5

- FD130: Bevor ein Objekt bearbeitet werden kann muss, geprüft werden, ob der die Bearbeitung ausführende Mitarbeiter die Berechtigung für die bestimmte Tätigkeit und das bestimmte Objekt hat.
  Priorität: 2

- FD140: Die Datensätze werden in den gewünschten Sprachen abgespeichert.
  Priorität: 2

3.4 Verwaltung

- FV10: Bevor auf der Seite der Datenverwaltung und Datenverarbeitung gearbeiten werden kann, muss man sich über einen Usernamen und ein Passwort identifizieren.
  Priorität: 1

- FV20: Es kann eine neue Objektgruppe erstellt werden.
  Priorität: 2

- FV30: Es kann eine Objektgruppe gelöscht werden, wenn sie keine Objekte enthält.
  Priorität: 3

- FV40: Es kann eine neue Funktionalität erstellt werden.
  Priorität: 2

- FV50: Es kann eine Funktionalität gelöscht werden, wenn es keine Benutzergruppe gibt, die diese Funktionalität enthält.
  Priorität: 3

- FV60: Es kann eine Benutzergruppe erstellt werden. Dafür muss mindestens eine Funktionalität bzw. eine Objektgruppe ausgewählt werden.
  Priorität: 2

- FV70: Es kann eine Benutzergruppe gelöscht werden, wenn es keine Mitarbeiter gibt, welche dieser Benutzergruppe zugewiesen sind.
  Priorität: 3

- FV80: Es kann ein Mitarbeiter einer Benutzergruppe zugeordnet werden.
  Priorität: 2

- FV90: Die Zuordnung eines Mitarbeiters zu einer Benutzergruppe kann gelöscht werden, wenn der Mittarbeiter mehr als einer Benutzergruppe zugewiesen ist.
  Priorität: 3

- FV100: Es kann ein neuer Mitarbeiter im System erstangemeldet werden. Dafür muss er mindestens einer Benuztergruppe zugewiesen werden (FV90) und ihm muss erstmals ein Benuztername (FV110) und ein Passwort (FV120) zugewiesen werden.
  Priorität: 2

- FV110: Es kann ein neuer Benuztername in das System eingetragen werden.
  Priorität: 2

- FV120: Zu einem Mitarbeiter (Benutzernamen) kann ein Passwort zugewiesen werden und somit der Mitarbeiter für die seinen Funktionalitäten entsprechenden Arbeiten freigegeben werden. Dies kann eine erstmalige oder eine erneute Freigabe sein.
  Priorität: 2

- FV130: Ein Mitarbeiter kann gelöscht werden.
  Priorität: 3

- FV140: Ein Mitarbeiter kann sein Passwort ändern.
  Priorität: 2

- FV150: Jeder Mitarbeiter wird gezwungen, alle 4 Monate sein Passwort zu ändern, ansonsten wird er gesperrt.
  Priorität: 5

3.5 Katalog

- FK10: Es kann ein Datensatz aus dem Museum für den Katalog ausgewählt werden.
  Priorität: 2

- FK20: Es kann direkt ein Datensatz aus der Datenbank für den Katalog ausgewählt werden.
  Priorität: 2

- FK30: Bevor der Katalog erstellt wird, muss mindestens 1 Datensatz für die Erstellung des Katalogs ausgewählt sein.
Priorität: 2

- FK40: Es werden die für den Katalog relevanten Daten aus dem Datensatz des Objekts ausgewählt. (Sichtbare Daten, nicht zu große Textdateien, etc.)
Priorität: 2

- FK50: Es muss für die Erstellung des Katalogs eine Gestaltungsvariante ausgewählt sein.
Priorität: 2

- FK60: Es wird aus den ausgewählten und für den Katalog relevanten Daten eine pdf-Datei erzeugt.
Priorität: 2

- FK70: Es kann eine Auswahl eines Datensatzes für den Katalog rückgängig gemacht werden.
Priorität: 3

- FD80: Es kann die Auswahl eines Datensatz aus dem Museum für den Katalog rückgängig gemacht werden.
Priorität: 3

3.6 CD-ROM

- FC10: Es kann ein Teil (Stockwerk, Raum, Vitrine oder Objekt in einer Vitrine) eines Museums für die CD-ROM ausgewählt werden.
Priorität: 2

- FC20: Es kann die Auswahl eines Teils (Stockwerk, Raum, Vitrine oder Objekt in einer Vitrine) des Museums für die CD-ROM aufgehoben werden.
Priorität: 2

- FC30: Es werden die für die CD-ROM relevanten Daten aus einem ausgewählten Datensatz herausgefiltert (nur sichtbare Daten).
Priorität: 2

- FC40: Um eine CD-ROM zu erstellen, muss mindestens ein Datensatz für die CD-ROM ausgewählt sein.
Priorität: 2

- FC50: Die ausgewählten (FC10) und kontrollierten (FC30) Daten werden in geeigneter Form zum CD-Brenner gelierfert.
Priorität: 2

- FC60: Es kann die Auswahl von allen für die CD-ROM ausgewählten Teile des Museums gelöscht werden (z.B. wenn die CD-ROMs fertig erstellt wurden).
Priorität: 4

3.7 Datenbankpflege

- FDB10: Es muss in einem bestimmten zeitlichen Abstand ein Backup von der Datenbank erstellt werden.
Priorität: 1

- FDB20: Es kann der zeitliche Abstand, in dem die Backups erstellt werden, eingestellt werden.
Priorität: 3

- FDB30: Es muss in einem bestimmten zeitlichen Abstand ein Backup von den Museen gemacht werden.
Priorität: 2

- FDB40: Es kann der zeitliche Abstand, in dem die Backups von den Museen gemacht werden, eingestellt werden.
Priorität: 3

- FDB50: Bei Ausfall der Datenbank muss auf das letzte Backup der Museen zurückgegriffen werden.
Priorität: 2

- FDB60: Bei Zerstörung der Datenbank muss eine funktionstüchtige Datenbank aus einem Backup erstellt werden.
Priorität: 1

- FDB70: Bei Verlust von Daten muss mit Hilfe der Backups eine datenkonsistente Version der Datenbank wieder hergestellt werden können, wobei alle vom Datenverlust nicht betroffenen Daten erhalten bleiben müssen.
Priorität: 1

Kapitel 4

Nicht funktionale Anforderungen

4.1 Zuverlässigkeit

- NZ10: Die Datenbank muss zu jeder Zeit verfügbar sein, da zu jeder Zeit eine Anfrage oder ein neuer Eintrag über das Internet kommen kann.
  Priorität: 1

- NZ20: Die beiden Web-Frontends (für den Internetsuer und für die Datenverwaltung und die Datenverarbeitung) müssen zu jeder Zeit verfügbar sein, da man jederzeit damit arbeiten können muss.
  Priorität: 1 (außer bei Ausfall der Datenbank)

- NZ30: Die Daten in der Datenbank müssen zu jedem Zeitpunkt konsistent sein.
  Priorität: 1

- NZ40: Ein Datenverlust darf nie vorkommen.
  Priorität: 1

4.2 Portabilität

- NP10: Auf alle Funktionen des Software Produkts muss über das Internet zugegriffen werden können.
  Priorität: 1

- NP20: Sonderschauen auf CD-ROM sollen auf allen gängigen Betriebssystemen lauffähig sein.
  Priorität: 2

- NP30: Es werden für die Darstellung der Daten gängige Formate verwendet. Bilder (jpeg, gif), Text wie Katalog, Publikationen oder Interpretationen (pdf), Videos (real audio), welche von den meisten Browsern unterstützt werden.
Priorität: 2

4.3 Antwortzeiten

- NA10: Die Antwortzeit der Datenbank darf 1000 ms nicht überschreiten.
Priorität: 1

- NA20: Die einzelnen Funktionen des Produkts dürfen nicht länger als 60 Sekunden dauern. (ausgenommen, die Erstellung der CD-ROM oder des Katalogs oder der Datensicherung bzw. Datenrekonstruktion)
Priorität: 2

4.4 Oberfläche

- NO10: Die Gestaltung der Benutzeroberfläche soll ansprechend und leicht verständlich sein.
Priorität: 1

- NO20: Alle Programme sollen eine grafische Benutzeroberfläche haben.
Priorität: 1

- NO30: Alle Funktionen sollen auch ohne Maus ausführbar sein.
Priorität: 1

Kapitel 5

Fehlerverhalten

Erst wenn ein Mitarbeiter alle Daten eines archäologischen Objekts vollständig eingegeben hat, werden alle Daten auf einmal in die Datenbank eingelesen.
Betrachtet gerade ein Internetuser im Museum einen Datensatz, der gerade geändert wurde, so erscheint kurz eine Mitteilung "Daten werden geändert", anschließend erscheint der Teil des Museums mit den neuen Daten. Wird im Museum etwas geändert (Stockwerk, Raum oder Vitrine), so wird dieser Teil des Museums für die Dauer der Arbeiten für den Internetuser als nicht betrachtbar geschalten, damit keine Probleme mit plötzlich ändernden Verhältnissen entstehen.
Hat der Internetuser über die Suche einen Datensatz gefunden, der gerade geändert wurde, so wird in der Liste nichts verändert, sobald aber der geänderte Datensatz ausgewählt wird, erscheinen die neuen Daten. Betrachtet der Internetuser gerade den bestimmten Datensatz, nachdem er ihn bei der Suche ausgewählt hat, der geändert wurde, so wird verfahren, wie wenn im Museum ein gerade betrachteter Datensatz geändert wird.
Hat der Internetuser einen Datensatz, der gerade gelöscht wird, nach der Suche ausgewählt oder betrachtet er ihn gerade im Museum, so erscheint eine Informationsmitteilung, dass dieser Datensatz gerade gelöscht wurde, mit Verweis auf die davor besuchte Seite bzw. auf die Anfangsseite des Web-Frontends für dem Internetuser.
Wird ein Datensatz oder ein Teil des Museums von einem Mitarbeiter ausgewählt, um diesen zu ändern, so wird dieser Datensatz oder Teil des Museums für alle weiteren Personen gesperrt, so dass es nicht passieren kann, dass 2 Personen gleichzeitig einen Datensatz verändern.
Fällt die Datenbank aus, so ist die Datensuche nicht möglich. Von den Museen existiert aber eine statische Abbildung, welche immer nach einem bestimmten Zeitraum von der aktuellen Datenbank erzeugt wird (z. B. 1 mal

im Monat). Die Abspeicherung der Daten erfolgt auf die selbe Weise, wie wenn die Daten auf CD-ROM gespeichert werden, um ein Abbild des Museums darzustellen. Die Abbildung muss aber nicht zu jedem Zeitpunkt mit dem aktuellen Stand der Datenbank "ubereinstimmen. Ist die Datenbank ausgefallen, so wird beim Besuch des Museums auf diese Abbildung zurückgegriffen. Das Museum funktioniert dann so wie auf der CD-ROM, nur dass die Daten nicht auf CD-ROM, sondern auf einem externen Speicher liegen. Zur Begrenzung der abzuspeichernden Datenmenge werden nur die Daten gespeichert, welche im Museum angezeigt werden. Ist die Datenbank ausgefallen, so ist es nicht möglich, Daten einzugeben, zu verändern oder zu löschen. Die Museen können ebenso nicht verändert werden. Es kann eine CD-ROM oder ein Katalog von der statischen Abbildung des Museums gemacht werden.
Um einen Datenverlust zu vermeiden werden regenlmäßig von unserer Firma Backups von der Datenbank gemacht.
Kann sich ein Mitarbeiter 3 mal hintereinander nicht korrekt identifizieren, so wird sein Account gesperrt und er muss erneut vom Verwalter autorisiert werden.
Bei Datenverlust können aus einem der vorhandenen Backups die verlorenen Daten so wiedergewonnen werden, dass die Konsistenz der Daten wieder gewährleistet ist. Es werden dabei nur Daten geändert, welche von dem Datenverlust direkt betroffen sind. Bei totaler Zerstörung der Datenbank kann aus einem Backup wieder eine lauffähige und den Anschprüchen unseres Produkts entsprechende Datenbank generiert werden.
Kann ein Browser die XML oder den eventuell daraus generierten HTML Code nicht korrekt anzeigen, so soll eine erklärende Mitteilung erscheinen.
Tritt ein schwerer Systemfehler auf, so wird dieser so gut als möglich protokolliert und anschließend wird der Server neugestartet.

Kapitel 6

Dokumentationsanforderungen

6.1 Dokumentation des Aufbaus

Die laufende Dokumentation des Aufbaus des Software Produkts einschließlich der Klassendiagramme wird über das Program Rose von der Firma Rational ausgeführt. Die dort enthaltenen Daten sollen laufend in der Entwicklungsphase an den momentanen Zustand des Software Produkts angepasst werden.

6.2 Dokumentation der Datenbank

Die Struktur und Arbeitsweise der Datenbank wird ausführlich in einem Dokument festgehalten, um ein effizientes Arbeiten und Warten der Datenbank zu ermöglichen.

6.3 Benutzerhandbücher

6.3.1 Internetuser

Das Benuterhandbuch für den Internetuser steht in Form einer elektronischen Hilfefunktion des Programmes zu Verfügung. Es existiert eine kontextbezogene Hilfefunktion, ein hierachischer Hilfebaum und ein kleines Tutorial, welches ihn durch die Benutzung des Museums und der Datensuche leitet.

6.3.2 Mitarbeiter

Der Teil des Programmes, welcher für die Mitarbeiter vorbehalten ist, hat ebenfalls eine kontextbezogene Hilfefunktion und einen hierachischen Hilfebaum. Zusätzlich existiert für jede einzelne Gruppe der Mitarbeiter (Erfasser, wissenschaftliche Erfasser, Museumsgestalter, Verwalter) eine eigenes Benutzerhandbuch in gedruckter Form.

Kapitel 7

Abnahmekriterien

Der Auftraggeber fordert für die Phase 1 als zentrale Anforderungen alle funktionalen und nicht funktionalen Anforderungen mit Priorität 1 und 2. Auf jeden Fall müssen die Berechtigungsgruppen erstellt werden können und das Virtuelle Museum soll aufgebaut werden können. Kein Muss für die Phase 1 sind die Erstellung der CD-ROMs und des Katalogs.

7.1 Funktionale Anforderungen

7.1.1 Virtuelles Museum

Das Virtuelle Museum soll aufgebaut werden können. d. h. das Museum wird erzeugt aus Stockwerken, Räumen und Vitrinen. Auch die Gestaltung dieser soll möglich sein. Das Ändern bzw. Löschen des Museums, der Stockwerke, Räume und Vitrinen muss noch nicht realisiert sein.
Die Internetuser können dann die Stockwerke, Räume und Virtrinen betrachten und die darin befindlichen Objekte betrachten. Die Suche wird in dieser Phase noch nicht gefordert.
Die Erstellung der CD-ROM und des Katalogs wird zu diesem Zeitpunkt der Entwicklung noch nicht gefordert.

7.1.2 Daten

Die Neuerfassung von archäologischen Objekten soll möglich sein. Auch sollen die Beziehungen eingegeben werden können. Das Ändern und Löschen dieser Datensätze muss noch nicht vorhanden sein.
Die Daten werden vorerst nur in einer Sprache abgespeichert.

7.1.3 Verwaltung

Der Verwalter muss die Berechtigungsgruppen, das sind Benutzergruppen und Objektgruppen, erzeugen können. Wichtig ist auch die Vergabe von Usernamen und Passwörtern für die Mitarbeiter. Auch hier muss das Ändern und Löschen noch nicht möglich sein.

7.1.4 Datenbankpflege

Das Backup der Datenbank soll schon in dieser frühen Phase der Systementwicklung erstellt werden können.

7.2 Nicht funktionale Anforderungen

7.2.1 Zuverlässigkeit

Die in diesem Bereich definierten Anforderungen sind absolutes Muss.

- Die Datenbank muss jederzeit verfügbar sein.
- Die Web-Frontends müssen jederzeit verfügbar sein.
- Die Daten müssen konsistent sein.
- Ein Datenverlust darf nie vorkommen.

7.2.2 Portabilität

Auf alle Funktionen des Softwareprodukts muss über das Internet zugegriffen werden können.

7.2.3 Antwortzeiten

Die vereinbarten Antwortzeiten müssen eingehalten werden.

7.2.4 Oberfläche

Die in diesen Punkten festgelegten Bereiche müssen erfüllt sein:

- Die Gestaltung der Oberfläche soll ansprechend und einfach verständlich sein.
- Alle Programme sollen eine grafische Oberfläche haben.

- Alle Funktionen sollen auch ohne Maus ausführbar sein.

Kapitel 8

Risikoabschätzung

8.1 Erfüllungsrisiken

8.1.1 technische Ursachen

- Zugesagte Zugriffszeiten sind nicht einhaltbar: Auf Grund von Hardware oder Softwareproblemen. Eventuell zu viele Benutzer.
- Zugesagte Benutzerdichten (Benutzer pro Minute) sind nicht einhaltbar.
- Die vorgesehene Rechnerkapazität für den Server bzw. die Datenbank ist zu klein.
- Vorgesehener Plattenspeicher für die Datenbank ist zu klein.
- Vorgesehene Zuverlässigkeit ist nicht erreichbar.
- Vorgesehene Verfügbarkeit ist nicht erreichbar.
- Der Hardware-Lieferant geht Bankrott.
- Die verwendete eingekaufte Software ist fehlerhaft und erfüllt nicht die versprochenen Kriterien.

8.1.2 personelle Ursachen

- Das erforderliche Personal ist nicht verfügbar und kann nicht beschafft werden.

- Ausfall von wichtigen Know-How-Trägern wegen Krankheit oder anderen nicht vermeidbaren Gründen. Nicht so großes Problem, da unsere Firma relativ gross ist und daher leichter ein Nachfolger gefunden werden kann. Eine Einführungszeit ist aber immer einzuberechnen.

- Gleichzeitiges Ausfallen von mehreren Mitarbeitern (z. B. Grippeepedemie). Nicht so großes Problem, da unsere Firma relativ gross ist und daher leichter ein Nachfolger gefunden werden kann. Eine Einführungszeit ist aber immer einzuberechnen.

- Unerwartete Unproduktivität von Mitarbeitern auf Grund von unzureichender Ausbildung oder schlechten Arbeitsbedingungen.

- Die Einarbeitung in ein neu verwendetes Softwareprodukt braucht länger als erwartet.

8.1.3 rechtliche Ursachen

- Unterauftraggeber wie Hardware- oder Softwarelieferanten liefern nicht termingerecht.

- Leistung kann wegen Behördenauflagen oder Gesetztesänderungen nicht erbracht werden (z. B. Datenerfassung wegen Datenschutz)

- Gesetzliches Verbot der Veröffentlichung bestimmter Daten.

8.2 Umgebungsrisiken

- Wirtschaftliche oder geschäftliche Turbulenzen können das Projekt zum Erliegen bringen.

- instabile Randbedingungen, z. B. wechselnder Ansprechpartner mit wechselnden Wünschen.

8.3 Risiken bei Anwendung, Einführung

- Der Internetuser interessiert sich nicht für das ihm gebotene Angebot.

- Der Internetuser findet das Web-Frontend nicht angenehm

- Die Ansprüche des Benutzers an die Performance des Programms ist bei größerer Auslastung nicht machbar.

8.4 Kostenrisiken

- Eine Menge von versteckten Kosten treten auf, wie zusätzliche Dokumentationen, welche im Prototyp noch nicht notwendig waren.
- Durch Wechselkurse erhöhte Kosten für benötigte Resourcen.
- Erhöhter Zeitaufwand durch neue Anforderungen.
- Erhöhter Zeitaufwand durch Fehler in der Software.

8.5 Auslandsrisiko

Es wird relevant, da wir auf Grund der multilingualen Eigenschaft des Software Produkts auch mit Experten der entsprechenen Ländern zusammenarbeiten, um eine den Ländern gerecht werdende Lösung zu finden.

- Die politische Situation gefährdet ausländische Mitarbeiter.
- Die Zusammenarbeit mit ausländischen Mitarbeitern kann auf Grund von der politischen und wirtschaftlichen Situation erschwert oder gefährdet werden.
- Die rechtlichen Gegebenheiten in den anderen Ländern erlauben den Datenaustausch nur in eingeschränkter Form.

Kapitel 9

Durchführbarkeitsstudie

9.1 Vollständigkeit der Anforderungen

Muss vom Auftraggeber bestätigt werden.

9.2 Konsistenz der Anforderungen

Alle gestellten Anforderungen sind bis jetzt konsistent.

9.3 Prüfung der technischen Vorraussetzungen

Die für den Server benötigte Hardware ist verfügbar. Für die Datenbank wird so aufgebaut, dass, falls größere Resourcen benötigt werden, eine Erweiterung leicht möglich ist. Von unserer Seite aus ist eine entsprechende breitbandige Internetverbindung verfügbar. Der Auftraggeber und seine Mitarbeiter müssen sicherstellen, dass sie über einen entsprechende Verbindung verfügen. Von den Zulieferfirmen bzw. Filialen unserer Firma wurde uns bestätgt, dass die von uns bestellten Hardware- und Softwaresegmente allen Anforderungen entsprechen.

9.4 Prüfung der personellen Vorraussetzungen

Es ist für die Realisierung dieses Projekts eine Hauptgruppe von 5 Mitarbeitern bereitgestellt (ein Datenbankexperte, drei Programmierer sowie ein Netzwerkspezialist). Zusätzlich gibt es eine Gruppe von weiteren 3 Personen, welche Routinearbeiten, sowie die Betreuung des laufenden Produkts erledigen.

Auf Grund der Größe unserer Firma ist es kein Problem, beim Ausfall von Personal in einer relativ kurzen Zeit Ersatz bereitzustellen.
Es m"ussen noch Sprachexperten gesucht werden, die unsere Firma bei der Gewährleistung der Mehrsprachigkeit unterst"utzen.

9.5 Ökonomische Rechtfertigung

Das Programm erleichtert die Verwaltung der archäologischen Daten. Durch das Web-Frontend können die Mitarbeiter von lokal getrennten Orten die Daten verwalten. Somit spart man sich einiges an Fahrzeit und -kosten.
Durch die Präsentation der Daten in ansprechender Form erhählt der Auftraggeber einen Imagezuwachs für sein Unternehmen.
Durch Werbung für den Museumsshop des Auftragsgebers wir zusätzlich das Software-Produkt rentabel.

Kapitel 10

Actors

10.1 Identifikation von actors

- Internetuser
- Mitarbeiter
- Verwalter
- Drucker
- CD-Brenner
- Dolmetscher
- Systemadministrator
- Datenbank

10.2 Beschreibung der actors

10.2.1 Internetuser

Der Internetuser ist eine Person, die das Virtuelle Museum über einen Webbrowser besucht. Er kann dort die zur Schau gestellten Objekte betrachten. Es steht ihm auch eine Suche zu Verfügung.

10.2.2 Mitarbeiter

Der Mitarbeiter ist eine Person, die vom Verwalter verschiedenste Rechte zugeteilt bekommen kann. Er kann einer oder mehreren Benutzergruppen zugeordnet werden und somit indirekt Rechte erlangen. Solche Rechte sind:

- Daten erfassen
- Daten ändern und löschen
- Beziehungen festlegen
- Publikationen und Literaturverweise eingeben
- Katalog drucken
- CD-ROM erstellen
- Stockwerke, Räume, Vitrinen im Virtuellen Museum erzeugen
- Museum gestalten
- usw.

Der Mitarbeiter muss sich beim System identifizieren, um von seinen Rechten Gebrauch nehmen zu können.

10.2.3 Verwalter

Der Verwalter ist eine Person, die die Rechte den Mitarbeitern zuteilt. Er selbst hat alle Rechte. Der Verwalter kann Objektgruppen erstellen und auch Mitarbeiter in Gruppen einteilen. Er vergibt die Rechte entweder einer Objektgruppe oder einer Gruppe, die Rechte können aber auch nur an einzelne Personen verteilt werden. Will sich ein neuer Mitarbeiter anmelden, muss er dafür sorgen, dass dieser einen Usernamen und ein Passwort bekommt, um sich beim System anmelden zu können. Der Verwalter kann ein Virtuelles Museum löschen oder bei Bedarf ein neues Museum eröffnen.

10.2.4 Drucker

Der Drucker ist ein Gerät, das die mittels eines Werkzeuges erzeugte pdf-Datei auszudrucken und somit den Katalog erstellen kann.

10.2.5 CD-Brenner

Der CD-Brenner ist ein Gerät, das das erstellte Abbild des Virtuellen Museums auf eine CD-ROM brennt.

10.2.6 Dolmetscher

Der Dolmetscher ist eine Person, der unser Projekt in die vom Kunden geforderten Sprachen übersetzt.

10.2.7 Systemadministrator

Unserer Firma, die SE AG, tritt als Systemadministrator auf. Wir haben dafür zu sorgen, dass ein Backup der Daten in der Datenbank erstellt wird, um bei einem möglichen Ausfall den Schaden gering zu halten. Es muss auch dafür gesorgt werden, dass die Programme einwandfrei funktionieren. Zusätzlich soll die Datenbank gewartet werden.

Kapitel 11

Anwendungsfälle (use cases)

11.1 Löschen von Artefakten

11.1.1 Kurzbeschreibung

Der use case wird von einem Mitarbeiter gestartet. Der Mitarbeiter muss die Rechte zum Löschen von Artefakten haben. Will er einen bereits bestehenden Datensatz löschen, so wird zunächst danach gesucht. Letztlich entfernt der Mitarbeiter die eingegebenen Daten.

11.1.2 Actors

- Mitarbeiter
- Datenbank

11.2 ArtefakteEingeben

11.2.1 Kurzbeschreibung

Der Use Case wird von einem Mitarbeiter gestartet. Der Mitarbeiter muss die Rechte zum Erfassen von Artefakten haben. Der Mitarbeiter kann ein Artefakt neu erfassen und alle Attribute der unteren Ebene (Grundeigenschaften) eingeben. Will er einen entsprechenden Datensatz verändern, so wird zunächst danach gesucht. Der Mitarbeiter sichert oder verwirft die eingegebenen Daten.

11.2.2 Actors

Erfasser

11.2.3 Priority

1

11.2.4 Status

Filled

11.2.5 Preconditions

1. Der Mitarbeiter muss am System angemeldet sein.
2. Der Mitarbeiter muss ausserdem die Rechte für die Eingabe von Artefakten haben.
3. Die Verbindung zur Datenbank muss bestehen.

11.2.6 Postconditions

Das neu eingegebene oder modifizierte Artefakt ist in der Datenbank gespeichert.

11.2.7 Extension Points

Fehlerbehandlung

11.2.8 "Used" Use Cases

Suchen von Objekten

11.2.9 Flow of Events

1. Wenn der Benutzer die Aktion ArtefakteEingeben startet, beginnt der Use Case.
2. Der Benutzer wird aufgefordert, eine Inventarnummer einzugeben. Will er ein bestehendes Artefakt bearbeiten, so gibt er an dieser Stelle die Inventarnummer des zu ändernden Artefakts ein. Will er ein neues

Artefakt einpflegen, so gibt er eine noch nicht vorhandene Inventarnummer ein.

3. Inventarnummer eingegeben, der Benutzer bestätigt die Eingabe.

4. Das System prüft, ob die Inventarnummer bereits in der Datenbank vergeben ist.

5. if (Inventarnummer in der Datenbank gefunden)

    (a) Das System meldet, dass es die Inventarnummer in der Datenbank gefunden hat. Es stellt die Frage, ob das der gefundenen Inventarnummer entsprechende Artefakt bearbeitet werden soll.

    (b) if (Benutzer will Artefakt bearbeiten)

        i. Hole das Artefakt mit der eingegebenen Inventarnummer aus der Datenbank.

        ii. Kopiere die Attribute des gefundenen Artefakts in die Felder des Dialogfensters zur Attributspflege.

    (c) if (Benutzer will Artefakt nicht bearbeiten)

    (d) Gehe zu 2.

6. if (Inventarnummer in der Datenbank nicht gefunden)

    (a) Das System meldet, dass es die Inventarnummer in der Datenbank nicht gefunden hat. Es stellt die Frage, ob der Benutzer ein neues Artefakt anlegen will.

    (b) if (Benutzer will neues Artefakt eingeben)

        i. Das System legt temporär ein neues Artefakt mit der eingegebenen Inventarnummer an.

        ii. Die Felder des Dialogfensters zur Attributspflege werden leer gesetzt.

    (c) if (Benutzer will kein neues Artefakt eingeben)

        i. Gehe zu 2.

7. Der Dialog zur Attributspflege wird sichtbar.

8. Der Benutzer gibt die Daten ein.

9. Der Benutzer bestätigt ober verwirft seine Eingaben.

10. if (Eingabe speichern)

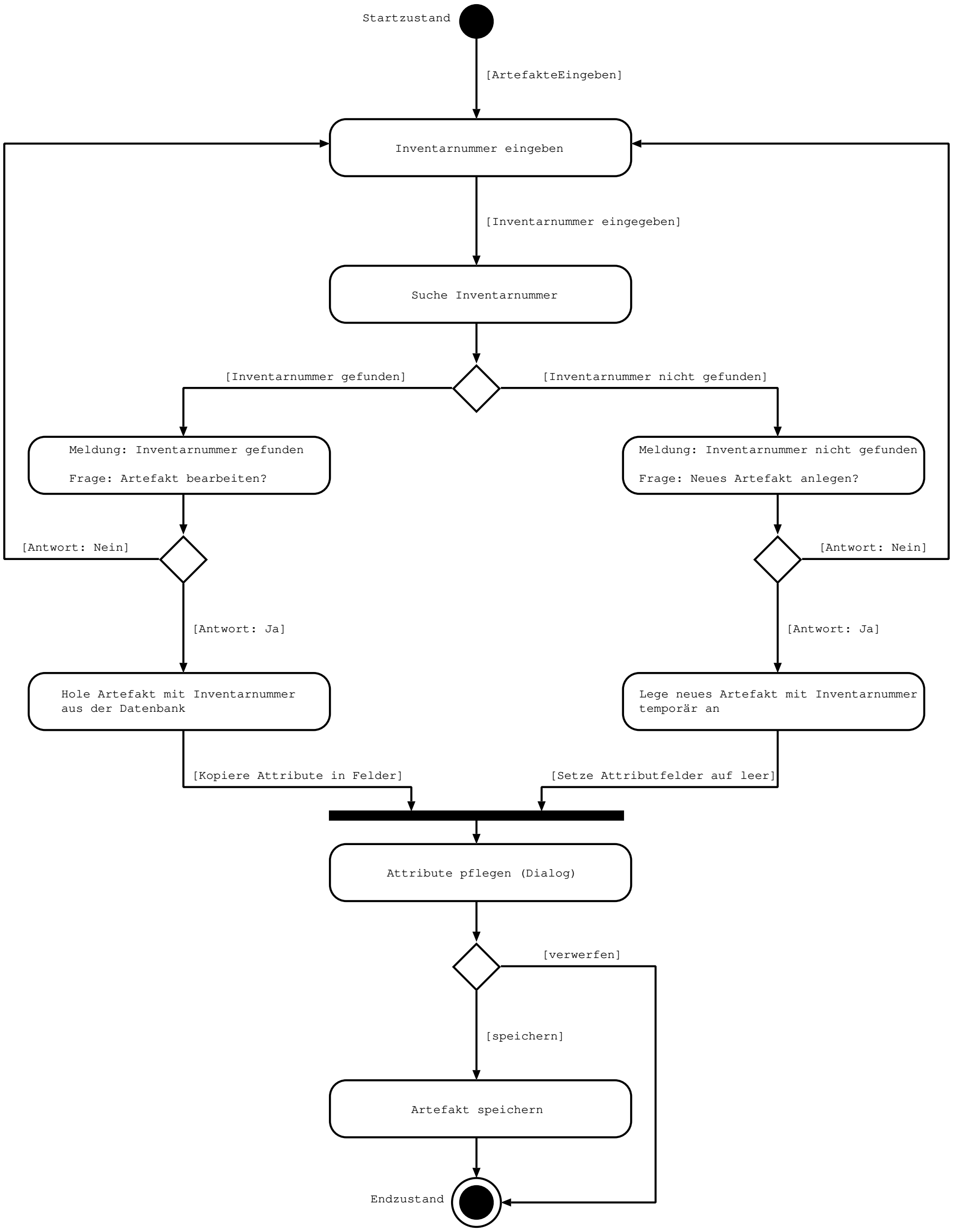
    (a) Das Artefakt wird mit den eingegebenen Attributen in der Datenbank gespeichert.

11. if (Eingabe verwerfen)

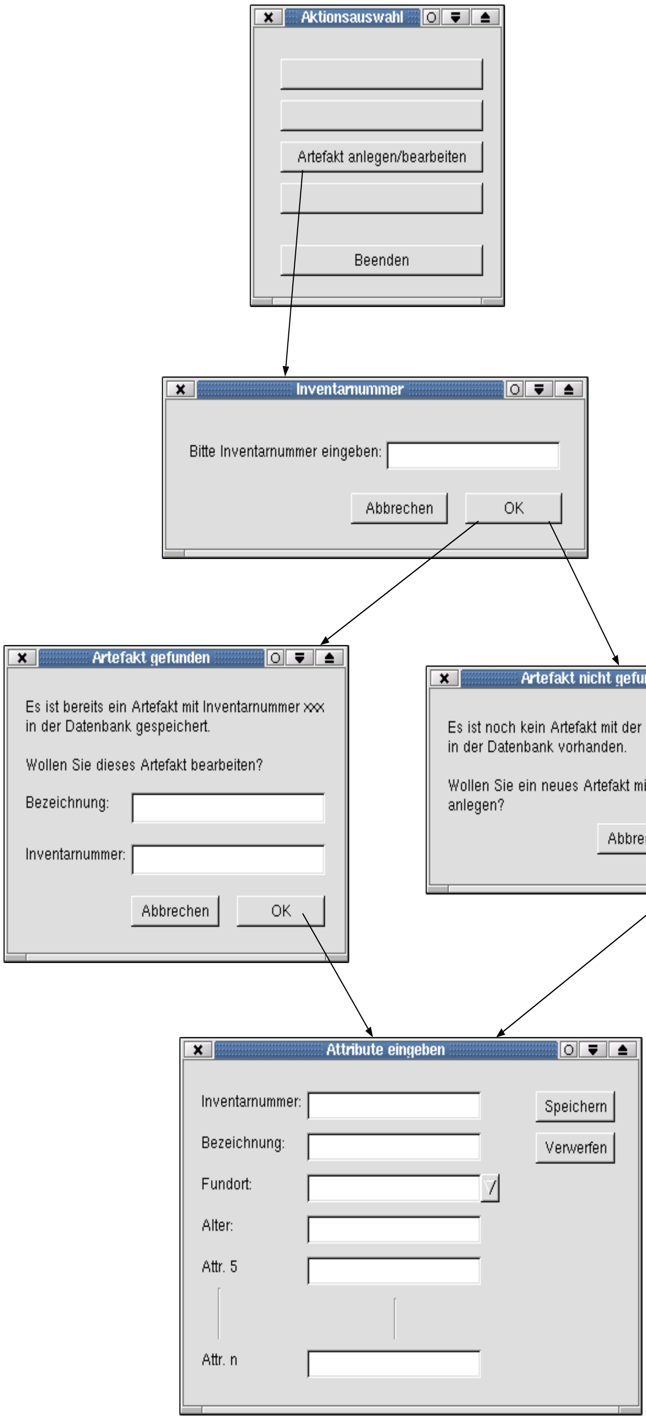
    (a) Die Eingabe wird abgebrochen, die eingegebenen Daten aus 8. gehen verloren.

12. Der Use Case wird beendet.

11.2.10 Activity Diagram



11.2.11 User Interface



11.2.12 Secondary Scenarios

Der Benutzer kann jede Aktion jederzeit mit der ABBRECHEN Taste abbrechen. Damit kehrt er zum Startzustand zurück.

11.3 Katalog erstellen

11.3.1 Kurzbeschreibung

Der use case wird von einem Mitarbeiter mit den entsprechenden Rechten gestartet. Der Mitarbeiter wählt von den vorhandenen Daten die gewünschten Daten aus. Mittels eines Werkzeugs wird dann eine pdf-Datei damit erstellt. Der Drucker erzeugt schließlich den Katalog.

11.3.2 Actors

- Mitarbeiter mit entsprechenden Rechten
- Drucker
- Datenbank

11.4 CD-ROM erzeugen

11.4.1 Kurzbeschreibung

Der use case wird von einem Mitarbeiter mit diesen Rechten gestartet. Er soll ein Abbild des Virtuellen Museums auf CD-ROM erstellen. Der Mitarbeiter hat die Möglichkeit, das ganze Museum abzubilden oder nur einen Teil davon. Ein Werkzeug unterstützt ihn bei der Arbeit. Die davon erstellte Datei wird nun mit Hilfe des CD-Brenners auf die CD-ROM gebracht.

11.4.2 Actors

- Mitarbeiter mit Berechtigung
- CD-Brenner
- Datenbank

11.5 Suchen von Objekten

11.5.1 Kurzbeschreibung

Der use case wird vom Internetuser gestartet. Er will in den von den Mitarbeitern sichtbaren Objekten nach etwas bestimmten suchen. Es wird ihm dazu eine einfache Suche nach Grundattributen und eine erweiterte Suche auch in den Publikationen und der Literatur zur Verfügung gestellt. Das Ergebnis wird in einer Liste dargestellt, von der er direkt die Objekte betrachten kann und die Informationen dazu erhält. Wählt er aus der Liste ein Objekt aus, werden zusätzlich die Suchergebnisse im Virtuellen Museum markiert, indem das Stockwerk, dann der Raum und die Vitrine markiert werden.

11.5.2 Actors

- Internetuser
- Datenbank

11.5.3 Priority

Priority: 1

11.5.4 Status

Filled

11.5.5 Preconditions

- Ein Museum muss erstellt worden sein.
- Objekte müssen erfasst worden sein.
- Objekte im Museum müssen als "sichtbar" gekennzeichnet sein.

11.5.6 Postconditions

- Suchergebnisse sind im Virtuellen Museum markiert.

11.5.7 Extension Points

- ErrorHandling

11.5.8 Flow of Events

1. Der use case beginnt mit der Selektion "Objekte suchen" auf der Homepage des Virtuellen Museums
2. Der Internetuser kann einen Suchbegriff für die einfache Suche eingeben
3. Der Internetuser kann einfache Suche (default) oder erweiterte Suche auswählen
4. Der Internetuser bestätigt die Eingabe
5. if (einfache Suche ausgewählt)
    (a) Das System durchsucht die Grundattribute nach dem eingegebenen Begriff
    (b) if (System hat Begriff gefunden)
        i. Darstellung einer Liste mit Objekten in denen der Suchbegriff vorkommt
        ii. Markierung der gefundenen Objekte im Virtuellen Museum
    (c) if (System findet kein Objekt zu dem eingegebenen Suchbegriff)
        i. Ausgabe einer Meldung "Suchbegriff nicht gefunden - Suche wiederholen? ja/nein"
        ii. if (Suche wiederholen)
            A. System zeigt die Eingabemaske der Suche erneut an.
        iii. if (Suche nicht wiederholen)
            A. Das System verlässt die Eingabemaske. Das System beendet den use-case
6. if (Erweiterte Suche ausgewählt)
    (a) Das System durchsucht die Grundattribute, Literatur und Publikationen nach dem eingegebenen Begriff
    (b) if (System hat Begriff gefunden)
        i. Darstellung einer Liste mit Objekten in denen der Suchbegriff vorkommt
        ii. Auswahl eines Objektes aus der Liste
        iii. Markierung der gefundenen Objekte im Virtuellen Museum
    (c) if (System findet kein Objekt zu dem eingegebenen Suchbegriff)

i. Ausgabe einer Meldung "Suchbegriff nicht gefunden - Suche wiederholen? ja/nein"

7. if (Suche wiederholen)

   (a) System zeigt die Eingabemaske der Suche erneut an.

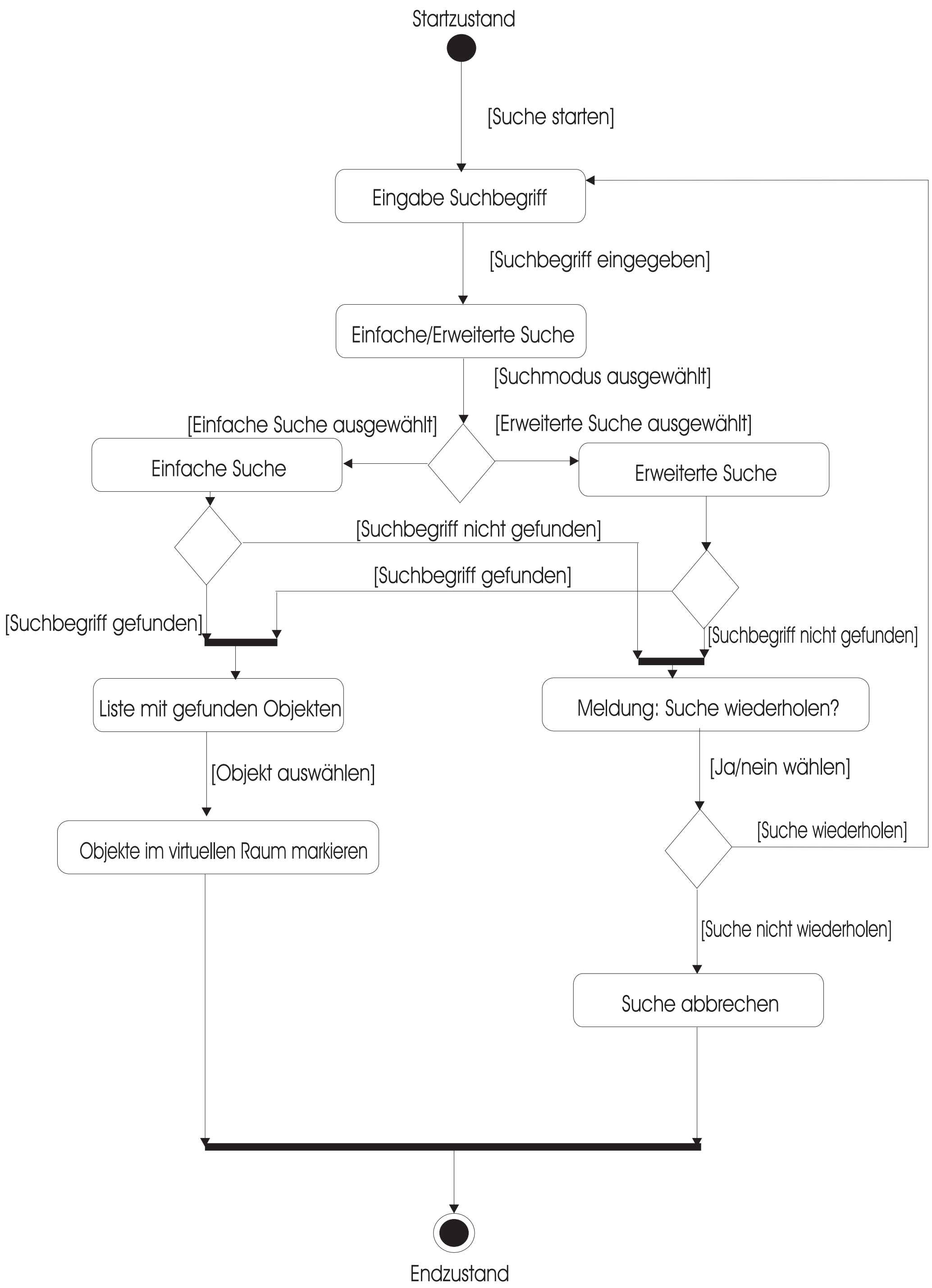
8. if (Suche nicht wiederholen)

   (a) Das System verlässt die Eingabemaske. Das System beendet den use-case

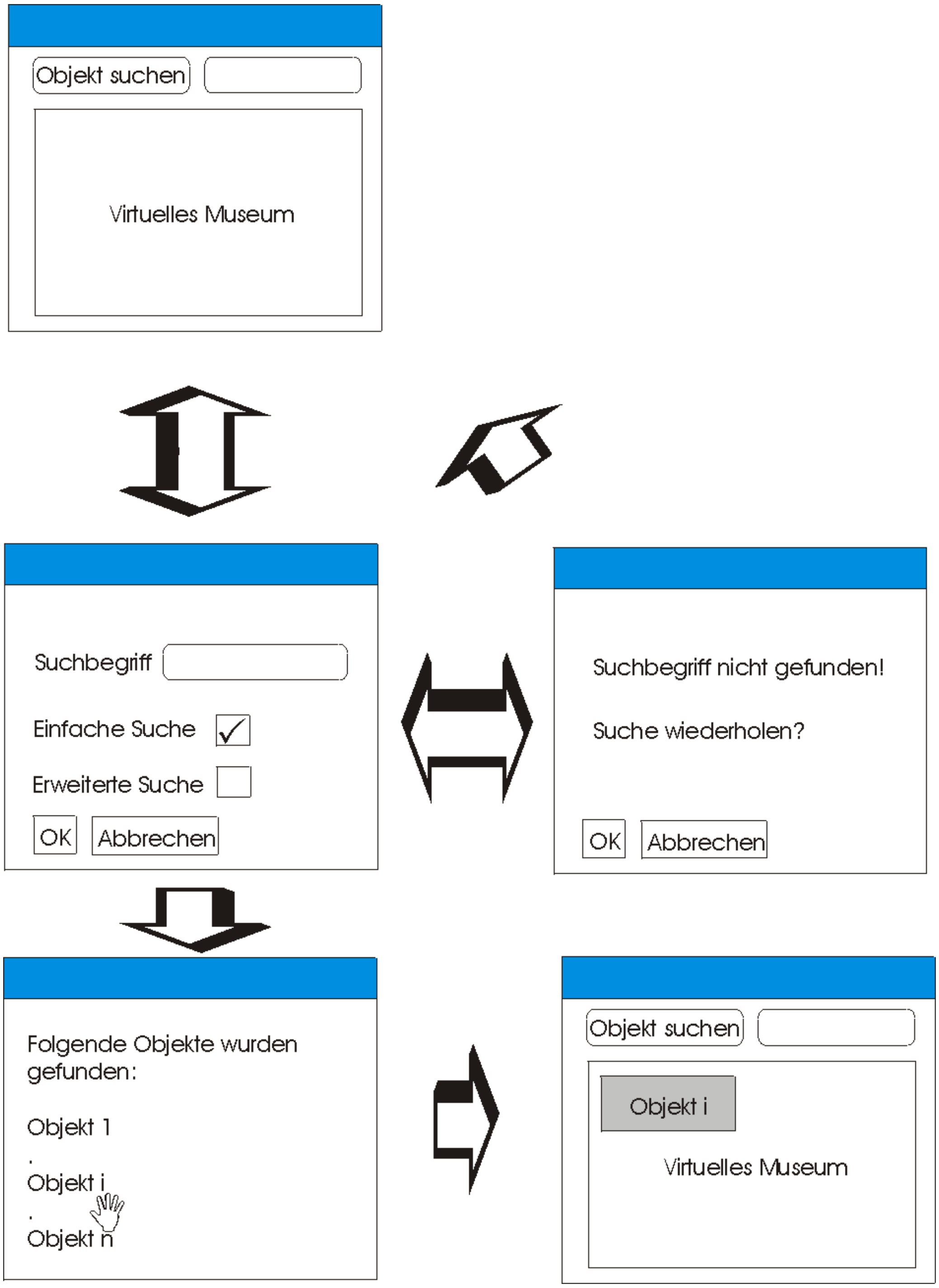
9. if (Benutzer bricht Verarbeitung ab)

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

11.5.9 Activity Diagram



11.5.10 User Interface



11.5.11 Secondary Scenarios

Der Benutzer kann zu jeder Zeit den Vorgang durch schließ en des Fensters oder durch die Abbrechen-Taste abbrechen und kehrt damit zum Ausgangspunkt zurück.

11.6 MuseumBesuchen

11.6.1 Kurzbeschreibung

Der use case wird vom Internetbesucher gestartet. Der Internetbesucher kann das Virtuelle Museum nun durchwandern. Das ganze ist hierarchisch aufgebaut, d.h. er wählt ein Stockwerk, dann einen Raum uund schlussendlich auch noch eine Vitrine mit den Objekten darin.

11.6.2 Actors

- Internetuser
- Datenbank

11.6.3 Priority

Der use case MuseumBesuchen hat für unser Projekt Priority **1**, da ein Hauptteil des Projekts in der Gestaltung dieses Museum besteht.

11.6.4 Status

Filled

11.6.5 Preconditions

Ein Museum muss bestehen.

11.6.6 Postconditions

Es gibt in diesem Sinn keinen Endzustand, der Internetbesucher kann das Museum immer wieder durchwandern.

11.6.7 Extension Points

- ErrorHandling

11.6.8 “Used” Use Cases

- DatenSuchen

11.6.9 Flow of Events

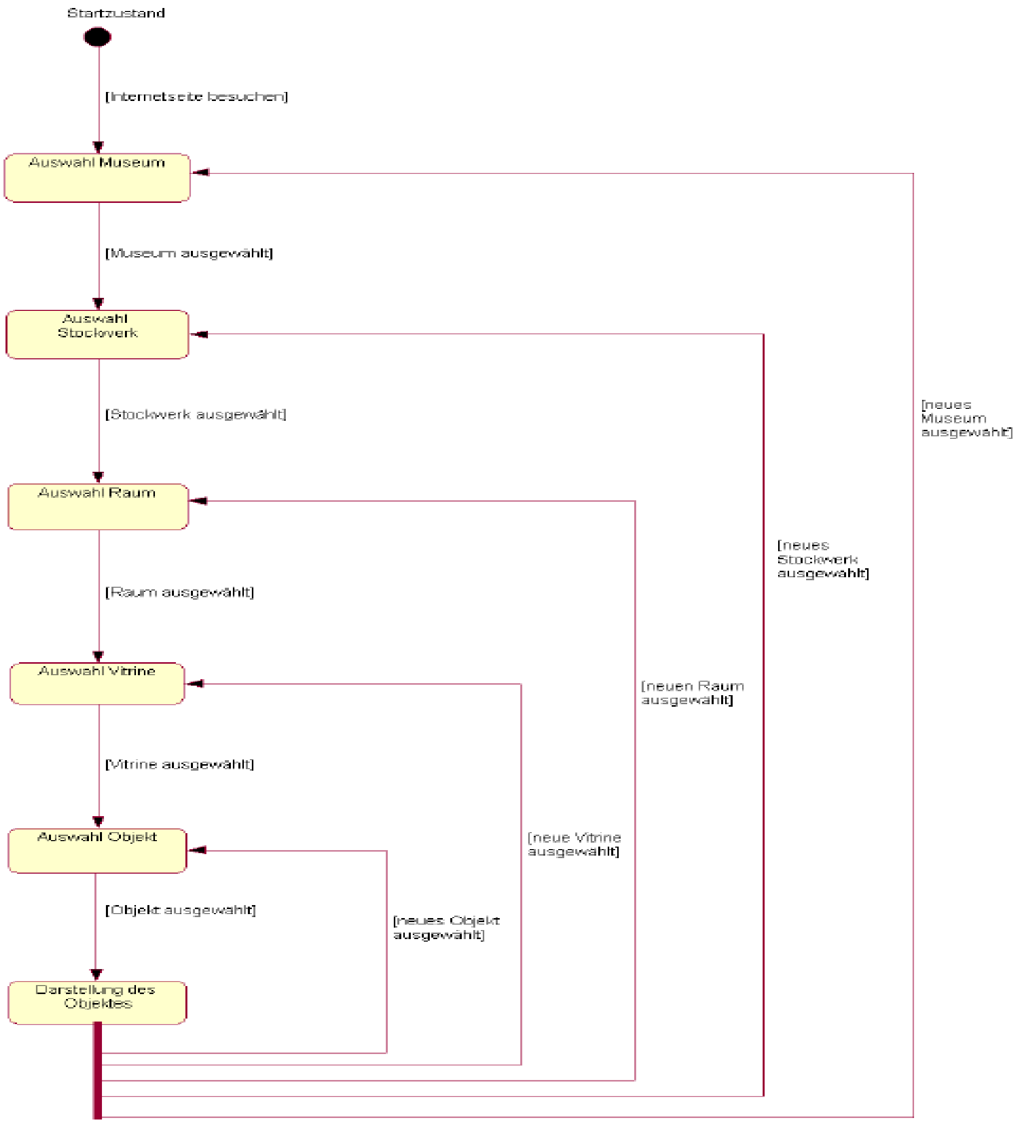
1. Der use case beginnt mit dem Einwählen des Internetusers auf diese Seite im Internet.
2. Der Internetuser wählt ein Museum aus, das er sehen will.
3. Es erscheint ein Plan des ausgewählten Museums. Der Internetuser wählt ein Stockwerk aus.
4. Der Plan wird spezifiziert. Die Räume eines Stockwerkes werden mit einer Kurzbeschreibung angezeigt. Der Internetuser wählt einen Raum aus.
5. In diesem Raum befinden sich Vitrinen. Diese werden wieder in einem Plan dargestellt und mit einer Kurzbeschreibung versehen. Der Internetuser wählt eine Vitrine aus.
6. Letztlich wählt der Internetuser dann das gewünschte Objekt aus der Vitrine aus.
7. Das geählte Objekt wird mit den dazugehörigen Informationen dargestellt.
8. Der Internetuser kann jetzt wieder

   (a) nur ein neues Objekt aus der Vitrine
   (b) eine neue Vitrine
   (c) einen neuen Raum
   (d) eine neues Stockwerk
   (e) ein neues Museum

   wählen.

Figure 11.1: Activity Diagram für MuseumBesuchen

11.6.10 Activity Diagram



siehe Abbildung 11.1

Figure 11.2: User Interface: Museen

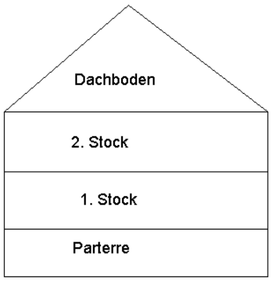


Figure 11.3: User Interface: Stockwerke eines Museum

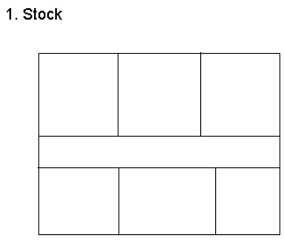
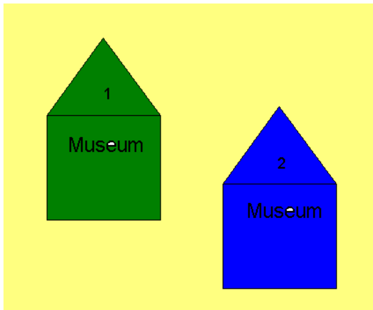


Figure 11.4: User Interface: Räume eines Stockwerkes

11.6.11 User Interface



siehe Abbildung 11.2 bis Abbildung 11.7

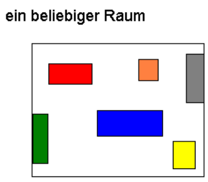


Figure 11.5: User Interface: Vitrinen eines Raumes

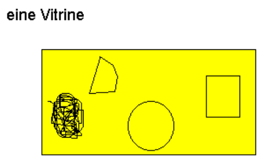


Figure 11.6: User Interface: Artefakte einer Vitrine

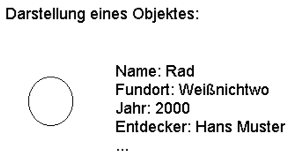


Figure 11.7: User Interface: Darstellung eines Objektes

11.6.12 Secondary Scenarios

Der Internetuser hat immer die Möglichkeit auf die Startseite mittels eines Navigators zu gelangen.
Der Internetuser kann mittels einer “Abbrechen”-Funktion in der Hierarchie immer um eine Stufe weiter nach oben gelangen. Ist er z.B. in einem Raum, kommt er mittels “Abbrechen” zur Auswahl eines Stockwerkes.

11.7 Datensicherung

11.7.1 Kurzbeschreibung

Dieser use case wird von unserer Firma, der SE AG, gestartet. In bestimmten Abständen soll von der Datenbank ein Backup erstellt werden, um bei möglichen Ausfällen der Datenbank den Schaden möglichst gering zu halten. Das Virtuelle Museum kann mit diesem Backup weiterlaufen, nur Suche und die Aufgaben der Mitarbeiter können bis zur Reparatur nicht durchgeführt werden.

11.7.2 Actors

- Systemadministrator

11.8 Objektgruppen erstellen

11.8.1 Kurzbeschreibung

Der use case wird vom Verwalter gestartet. Er kann damit die Rechte der Mitarbeiter auf eine Objektgruppe einschränken. Er soll die Daten in der Datenbank suchen können. Er kann diese Objektgruppen natürlich verändern und sogar wieder löschen. Das Löschen der Objektgruppen ist aber nur möglich, wenn sie keinen Mitarbeitern mehr zugeordnet sind, deren Rechte darauf beschränkt sind.

11.8.2 Actors

- Verwalter

11.9 Rechte festlegen

11.9.1 Kurzbeschreibung

Der use case wird vom Verwalter gestartet. Der Verwalter muss den Mitarbeitern Rechte vergeben. Er kann die Rechte an Gruppen von Mitarbeitern vergeben. Der Verwalter hat aber auch die Möglichkeit, sogenannte Objektgruppen zu erstellen und die Rechte der Mitarbeiter dann zusätzlich auf eine solche Objektgruppe beschränken.

11.9.2 Actors

- Verwalter

11.10 Benutzergruppe erstellen

11.10.1 Kurzbeschreibung

Benutzergruppe anlegen: Der Use Case wird durch den Verwalter gestartet. Es kann eine Benutzergruppe gesucht werden, eine vorhandene kopiert werden und eine neue Benuztergruppe erstellt werden. Für die Erstellung einer Benutzergruppe müssen eine oder mehrere Funktionalitäten und ein oder mehrere Objektgruppen ausgewählt werden. Zum Abschluß werden die eingegebenen Daten gesichert oder vom Verwalter verworfen.

11.10.2 Actors

Verwalter

11.10.3 Priorität

1

11.10.4 Status

Filled

11.10.5 Precondition

- Der Veralter muss sich identifiziert und am System angemeldet haben.
  - Es muss mindestens eine Funktionalität existieren.
  - Es muss mindestens eine Objektgruppe existieren.

11.10.6 Postcondition

Die Benutzergruppe wurde in der Datenbank gespeichert. Es wurde in der Benutzergruppe mindestens eine Funktionalität und eine Objektgruppe ausgewählt.

11.10.7 Extension Points

ErrorHandling

11.10.8 "used" Use Cases:

- Benutzergruppe suchen
  - Objektgruppe suchen
  - Funktionalität suchen

11.10.9 Flow of Events:

1. Der Verwalter wählt die Funktion "Benutzergruppe erstellen".
   (a) Der Verwalter gibt die Bezeichnung für die Benutzergruppe ein.
   (b) Der Verwalter bestätigt die Eingabe.
   (c) Das System sucht, ob der eingegebene Bezeichner für die Benutzergruppe bereits existiert.
   (d) if (Bezeichnung für Benutzergruppe existiert bereits)
      i. System meldet, dass Bezeichnung für Benutzergruppe schon existiert.
      ii. while(kein neuer Bezeichner für Benutzergruppe wurde eingegeben)
         A. Der Verwalter gibt eine weitere Bezeichnung für eine Benutzergruppe ein.
         B. Der Verwalter bestätigt die Eingabe.

C. Das System sucht, ob der Bezeichner für die Benutzergruppe schon existiert.

iii. end

iv. Die als erstes eingegeben Benutzergruppe wird kopiert und unter den als letztes eingegebenen Bezeichner, der ja noch nicht existiert, abgelegt.

v. Es wird das Feld zur Änderung der Daten der Benutzergruppe angezeigt.

(e) if(Bezeichnung für Benutzergruppe existiert noch nicht)

i. Es wird das Feld zur Änderung der Daten der Benutzergruppe angezeigt.

(f) if(der Verwalter will die Benutzergruppe abspeichern)

i. Das System kontrolliert, ob mindestens 1 existierende Objektgruppe **und** 1 existierende Funktionalität ausgewählt sind.

ii. while((es wurde keine existierende Objektgruppe ausgewählt) OR
(es wurde keine existierende Funktionalität ausgewählt)

A. Es wird das Feld zur Änderung von Daten der Benutzergruppe angezeigt. (Rücksprung auf 1(e)i)

iii. end

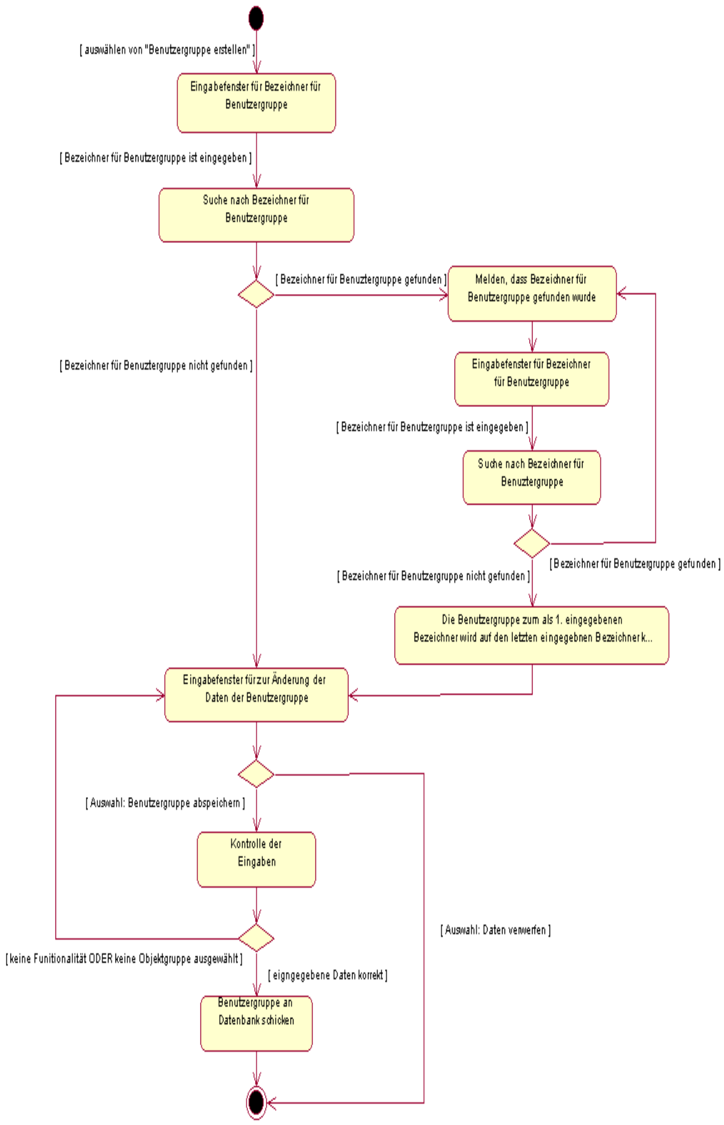
iv. Die Benutzergruppe wird an die Datenbank geschickt.

v. Das System geht wieder in den Grundzustand zurück.

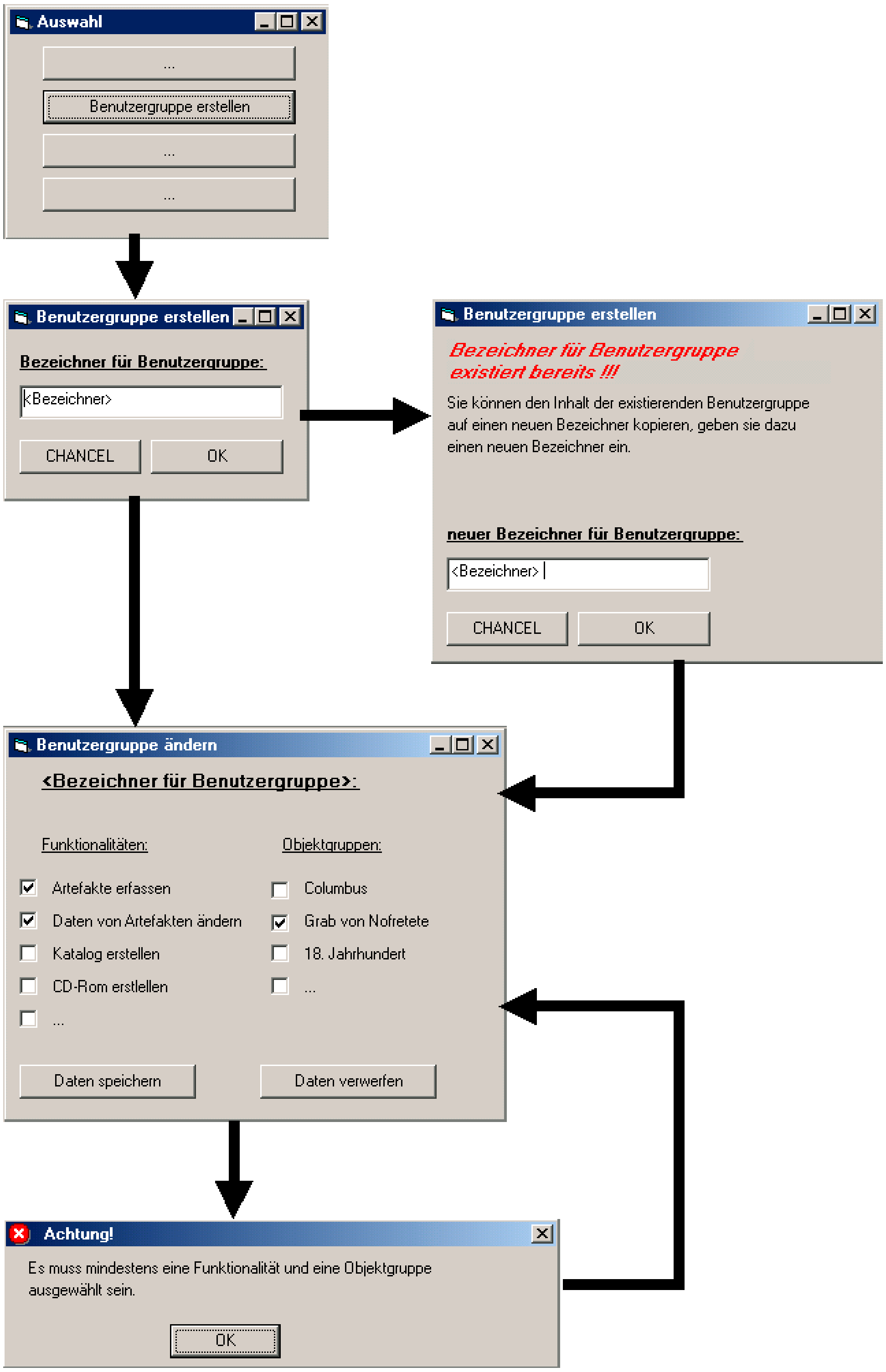
(g) if(der Verwalter will die Benutzergruppe verwerfen)

i. Die Benutzergruppe wird nicht gespeichert und das System geht wieder in den Grundzustand zurück.

11.10.10 Activity Diagram:



11.10.11 User Interface



11.10.12 Secondary Scenarios

Der Verwalter kann jederzeit durch Betätigen der CHANCEL Taste die Anwendung abbrechen und zum Ausgangszustand zurückkehren.

11.11 Identifizierung eines neuen Mitarbeiters

11.11.1 Kurzbeschreibung

Der use case wird vom Mitarbeiter gestartet. Möchte sich ein neuer Mitarbeiter beim System anmelden, erhält er vom Verwalter einen Username und ein Passwort. Der Verwalter gibt ihm auch sofort seine Rechte, die jederzeit erweitert, aber auch eingeschränkt werden können bei Missbrauch der Rechte. Loggt sich ein Mitarbeiter beim System ein, so kann er in seinem erlaubten Bereich agieren. Die Anmeldung ist erforderlich für die Nutzung der Rechte.

11.11.2 Actors

- Mitarbeiter
- Verwalter

11.12 Museumsgestaltung

11.12.1 Kurzbeschreibung

Der use case wird von einem Mitarbeiter gestartet, der vom Verwalter die Rechte dazu erhalten hat. Der Mitarbeiter darf nun im Virtuellen Museum ein Stockwerk und einen Raum gestalten. Unterstützt wird er von einer Art Wizard, der bei der Gestaltung eine Hilfe sein soll. Er hat aber nicht die Möglichkeit, Stockwerke oder Räume selbst zu erstellen. Dazu benötigt er zusätzliche Rechte.

11.12.2 Actors

- Mitarbeiter, der die Rechte dazu erhalten hat

11.13 VitrineErstellen

11.13.1 Kurzbeschreibung

In der folgenden Beschreibung wird der Begriff Benutzer stellvertretend für die Benutzergruppen Verwalter und Mitarbeiter verwendet. Unter Mitarbeiter wird nur jener Personenkreis verstanden, der auch das Zugriffsrecht hat, eine Vitrine zu erstellen.

Beschreibung der Funktionalitäten, die geboten werden sollen:
Ein Mitarbeiter, welcher über die entsprechenden Rechte verfügt, kann in einem Museum Vitrinen erstellen. Für die Gestaltung der Vitrinen wird eine Art Wizard bereitgestellt, welcher den Mitarbeiter durch die Erstellung führt. Der Mitarbeiter kann für die Gestaltung aus mehreren Templates auswählen, welche jeweils einer bestimmten Gestaltungsvariante entsprechen. Diese Templates können vom Verwalter jederzeit nachgeladen oder gelöscht werden.
Jeder Raum enthält Vitrinen. Ein Raum kann auch nur eine Vitrine enthalten. Jede Vitrine kann eine kurze Beschreibung über ihren Inhalt enthalten, muss aber nicht.

11.13.2 Actors

Actors die mit dem Use Case kommunizieren, sind:

- Mitarbeiter
- Verwalter
- Datenbank

11.13.3 Priority

Die Wichtigkeit, diesen Use Case zu implementieren, ist 2, d.h. Prioritä ist 2. (Priorität: [1..10]. Priorität 1 bezeichnet die höchste - und 10 die niedrigste Priorität.)

11.13.4 Status

In [5] wird zwischen vier Statusausprägungen der Entwicklung von Use Cases unterschieden.

- Facade
  Beschreibung auf hohem Abstraktionslevel
- Filled
  Design des kompletten Use Case.
- Focused
  Narrowing und Prunning des Use Case.
- Finished
  Fine tuning des Use Case.

Dieser Use Case besitzt den Status Filled.

11.13.5 Preconditions

In der Vorbedingung wird der Systemzustand spezifiziert, der eine notwendige Voraussetzung für ein erfolgreiches Ausführen des Anwendungsfalls darstellt. Ist die Vorbedingung nicht erfüllt, so kann der Anwendungsfall in der Regel nicht ausgeführt werden.
Die Vorbedingungen sind:
Es muss der Verwalter eingelogged sein oder es muss der Mitarbeiter eingelogged sein. Außerdem muss der Mtarbeiter das Zugriffsrecht VitrineErstellen besitzen. Unter „eingelogged"wird verstanden, dass der Benutzer sich erfolgreich mittels Passwort und Username identifiziert hat.

11.13.6 Postconditions

Analog dazu wird in der Nachbedingung der Systemzustand nach dem erfolgreichen Ausführen des Anwendungsfalls beschrieben. Die Nachbedingung muss wahr sein, egal welcher Weg durch den Use Case gewählt wird.
Die Nachbedingungen sind:
Die erstellte Vitrine ist in der Datenbank gespeichert.

11.13.7 Extension Points

Folgende Extension Points gibt es:

- VitrinePositionieren
  Die Vitrine kann, aber muss nicht in einem Raum positioniert werden.

- ObjektPositionieren
  Ein Objekt kann in einer Vitrinen positioniert werden, muss aber nicht. Das kann Sinn machen, wenn Vitrinen zu bestimmten Kategorien angelegt werden und erst später Objekte in die Vitrinen gestellt werden. Hier gibt es wieder folgende Extension Points:

  - ObjektEinstellen
    Hier wird ein Objekt an einem bestimmten Platz in der Vitrine gestellt.
  - ObjektLöschen
    Hier wird ein bestimmtes Objekt aus der Vitrine gelöscht.
  - ObjekteUmordnen
    Hier werden keine neue Objekte in die Vitrine gestellt oder schon vorhandene Objekte gelöscht, sondern gewisse Objekte werden an einem anderen Platz in der Vitrine gestellt.

- ObjektdarstellungFestlegen
  über einen Wizard kann der Benutzer die Darstellung der Objekte festlegen. Dies macht Sinn, da, wenn noch keine Objekte in die Vitrine gestellt wurden, auch nicht unbedingt die Darstellungsform der Objekte festgelegt werden muss.

11.13.8 „Used“ Use Cases

Folgende „Used“ Use Cases existieren:

- VitrinendarstellungFestlegen
  Es ist klar, wenn eine Vitrine erstellt wird, dass der Benutzer festlegen muss, wie die Vitrine dargestellt wird. Die Darstellungsform kann der Benutzer über einen Wizard festlegen. Der Benutzer kann Dinge festlegen, wie z.B. Form und Aussehen der Vitrine.

- VitrinenbeschreibungEingeben
  Der Benutzer der eine Vitrine erstellt, muss auch diese Vitrine beschreiben. Was eingegeben werden muss: Welche Kategorie diese Vitrine repräsentiert
  Optional: historische - und politsche Hintergründe, Motivation, Daten über den Vitrinenersteller, ...

11.13.9 Flow of Events

Ist eine Serie von deklarativen Statements, die die Schritte eines Use Case beschreiben. Alternativen können hier mittels Verzweigungen angeführt werden oder alternativ unter dem Punkt „Secondary Scenarios“ aufgelistet werden. Normalerweise wird das Wort if benutzt, um einen alternativen Pfad anzudeuten. Für Wiederholungen, also wenn gewisse Schritte mehrfach ausgeführt werden sollten, werden i.A. die Wörter for bzw. while benutzt. Das Ende einer Schleife wird durch das Wort end angezeigt. Zu for und while können zusätzlich logische Bedingungen angegeben werden.

1. Der Vorgang beginnt mit der Selektion „Vitrine erstellen“.
2. Festlegung wie die Vitrine dargestellt werden soll.
3. Beschreibung der Vitrine
4. if (Vitrine Raum zuweisen)
   (a) Vitrine Raum zu weisen.
5. else
6. if (Objekt in Vitrine positionieren)
   (a) while (weitere Objekte positionieren)
      i. Objektpositionierung
   (b) end
7. else
8. if (Speichern)
   (a) Der ganze Vorgang „Vitrine erstellen“ wird in der Datenbank abgespeichert.
   (b) Ende des Vorgangs „Vitrine erstellen“.
9. if (Abbrechen)
10. Ende des Vorgangs Vitrine erstellen, ohne den Vorgang in der Datenbank zu speichern.

11.13.10 Activity Diagram

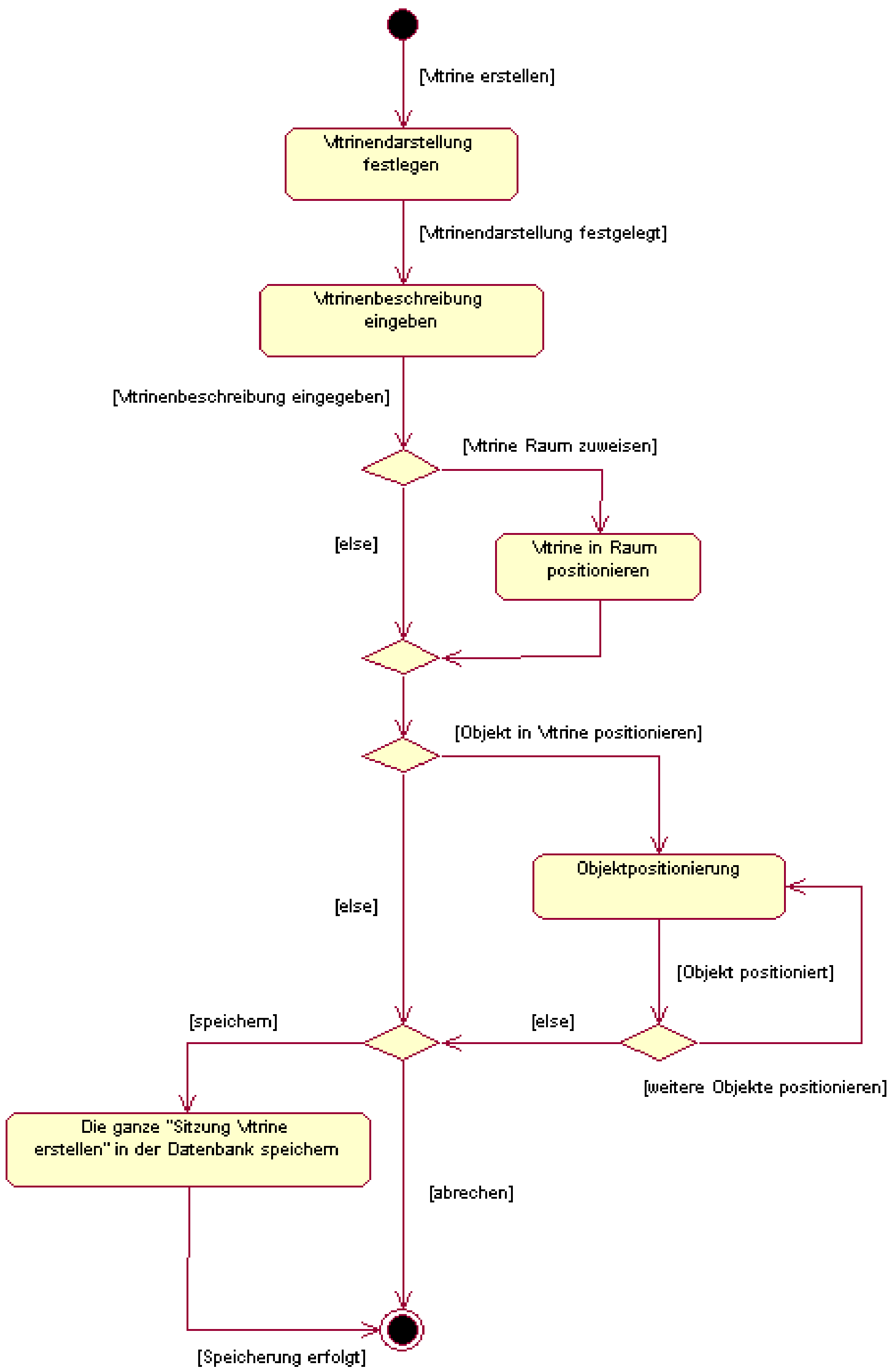
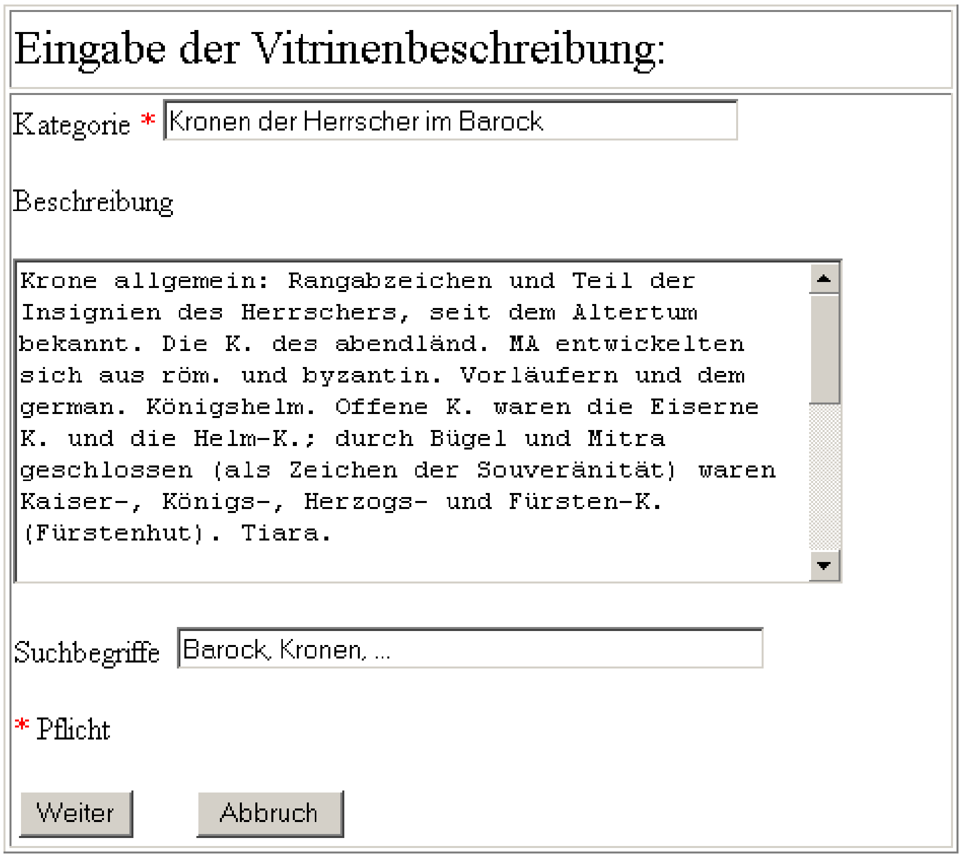


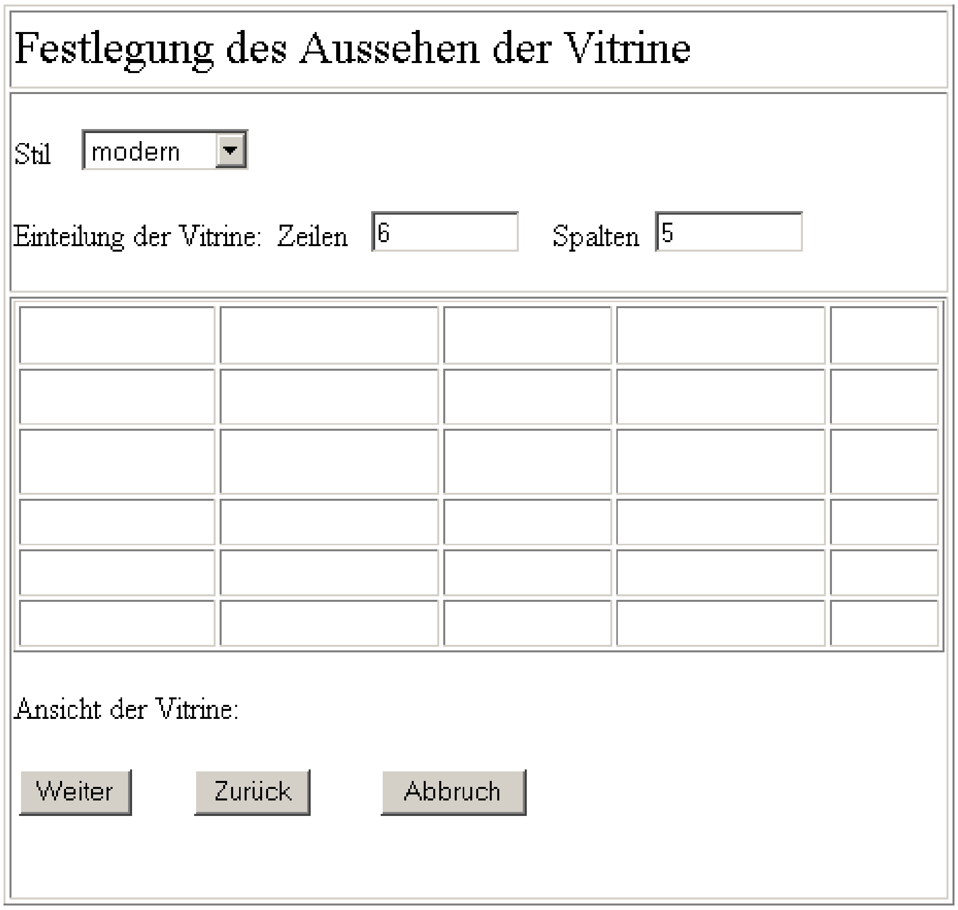
Figure 11.8: Aktivitätsdiagramm

11.13.11 User Interface

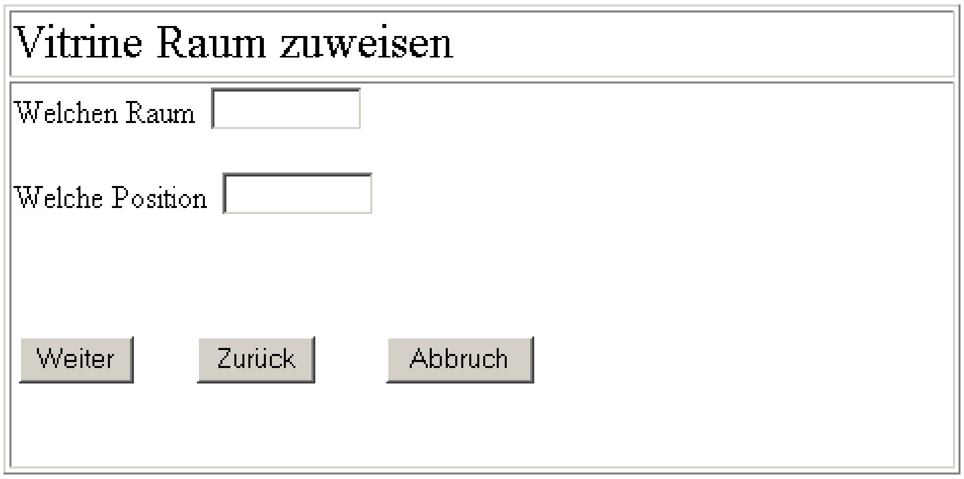
Bei Projekten, die sehr viel mit Kommunikation zwischen Benutzer und System auf GUI Ebene operieren, ist es angebracht, um die Benutzerakzeptanz zu erhöhen, Beispiele für mögliche GUIs zu präsentieren. Durch diese Beispiele können sich u.U. neue Anforderungen ergeben. Mittels der Story Board Technik werden dem Benutzer Folgen von Screens präsentiert, als ob er das System benutzen würde. Diese Technik wurde aus der Filmindustrie übernommen.

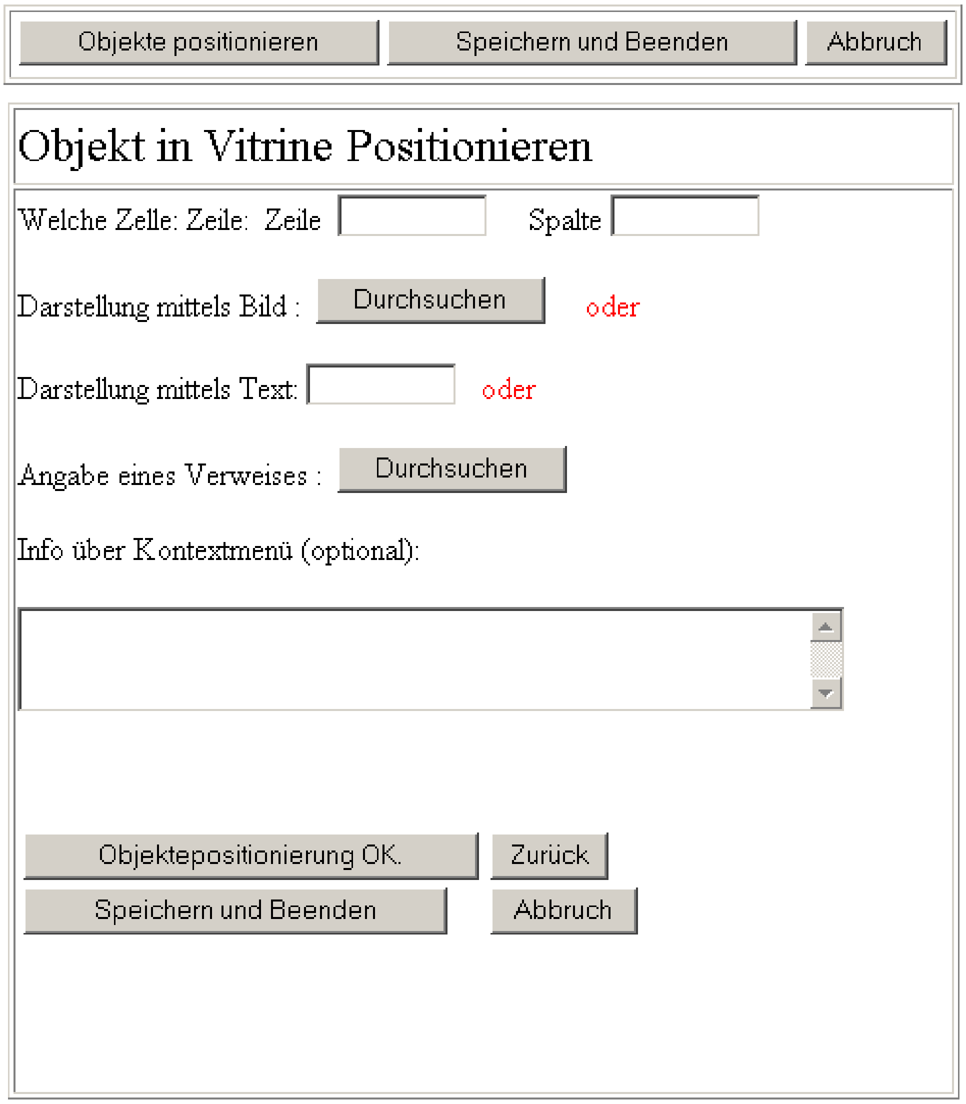
Über den Button in Abb. 11.13.11 kommt der Benutzer in das Dialogfenster abgebildet in Abb. 11.13.11. In den folgenden Screens bedeutet das Drücken des Buttons *Abbruch* den Abbruch des Vorgangs und Rückkehr in den Ausgangszustand. *Weiter* bedeutet, dass der nächste Screen aufgerufen wird. Analog dazu bedeutet *Zurück*, dass der letzte Screen aufgerufen wird.
Mit *Weiter* in Abb. 11.13.11 kommt der Screen in Abb. 11.13.11.
Mit *Weiter* in Abb. 11.13.11 kommt der Screen in Abb. 11.13.11.
Mit *Weiter* in Abb. 11.13.11 kommt der Screen in Abb. 11.13.11 und mit *Vitrine Raum zuweisen* der Screen in Abb. 11.13.11.
Mit *Weiter* in Abb. 11.13.11 kommt der Screen in Abb. 11.13.11.
In Abb. 11.13.11 wird mit *Speichern und Beenden* der Vorgang „Vitrinen erstellen " beendet und mit *Objekte positionieren* wird der Screen in Abb. 11.13.11 dargestellt.
In Abb. 11.13.11 wird mit *Objektpositionierung* eine Objektpositionierung in der Datenbank gespeichert und ein neues Objekt kann positioniert werden und mit *Speichern und Beenden* der Vorgang „Vitrinen erstellen " beendet.
Der Screen in Abb. 11.13.11 dient als Hilfe für die Positionierung der Objekte in Abb. 11.13.11.

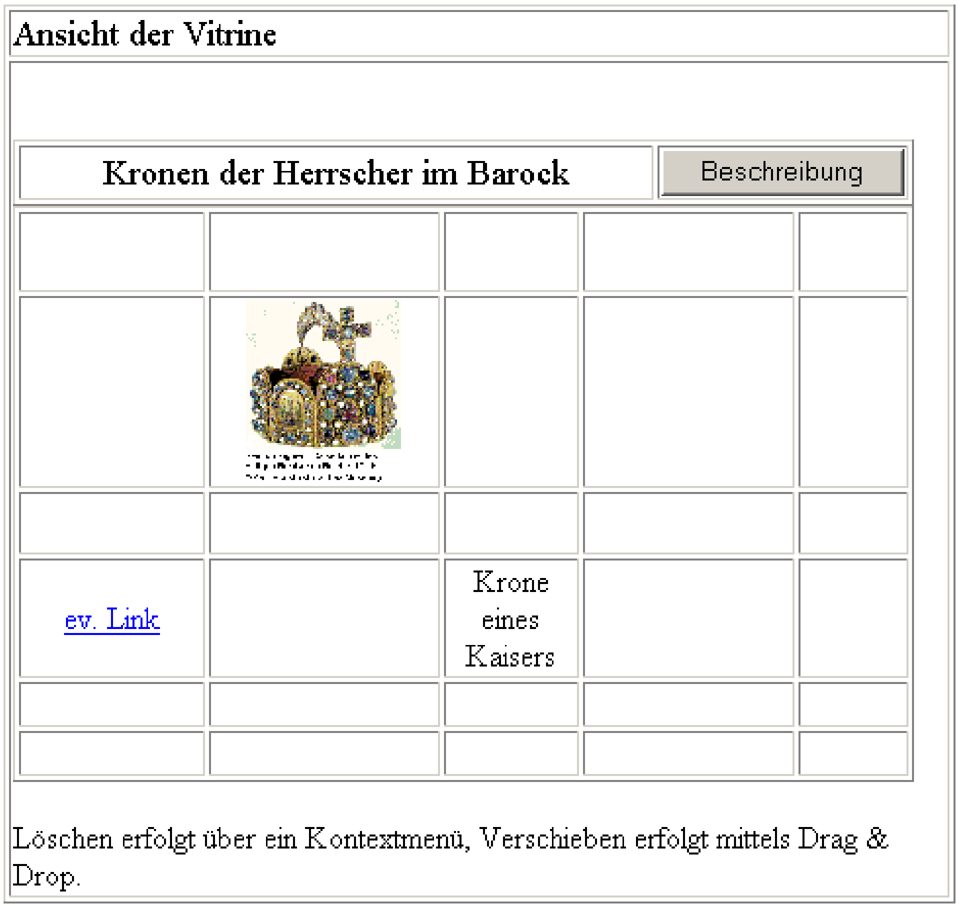

Eingabe der Vitrinenbeschreibung:
Kategorie * Kronen der Herrscher im Barock
Beschreibung
Krone allgemein: Rangabzeichen und Teil der Insignien des Herrschers, seit dem Altertum bekannt. Die K. des abendländ. MA entwickelten sich aus röm. und byzantin. Vorläufern und dem german. Königshelm. Offene K. waren die Eiserne K. und die Helm-K.; durch Bügel und Mitra geschlossen (als Zeichen der Souveränität) waren Kaiser-, Königs-, Herzogs- und Fürsten-K. (Fürstenhut). Tiara.
Suchbegriffe Barock, Kronen, ...
* Pflicht
Weiter
Abbruch


Festlegung des Aussehen der Vitrine
Stil
modern
Einteilung der Vitrine: Zeilen
6
Spalten
5
Ansicht der Vitrine:
Weiter
Zurück
Abbruch


Vitrine Raum zuweisen
Weiter
Zurück
Abbruch


Vitrine Raum zuweisen
Welchen Raum
Welche Position
Weiter
Zurück
Abbruch


Objekte positionieren
Speichern und Beenden
Abbruch
Objekt in Vitrine Positionieren
Welche Zelle: Zeile: Zeile
Spalte
Darstellung mittels Bild :
Durchsuchen
oder
Darstellung mittels Text:
oder
Angabe eines Verweises :
Durchsuchen
Info über Kontextmenü (optional):
Objektepositionierung OK.
Zurück
Speichern und Beenden
Abbruch



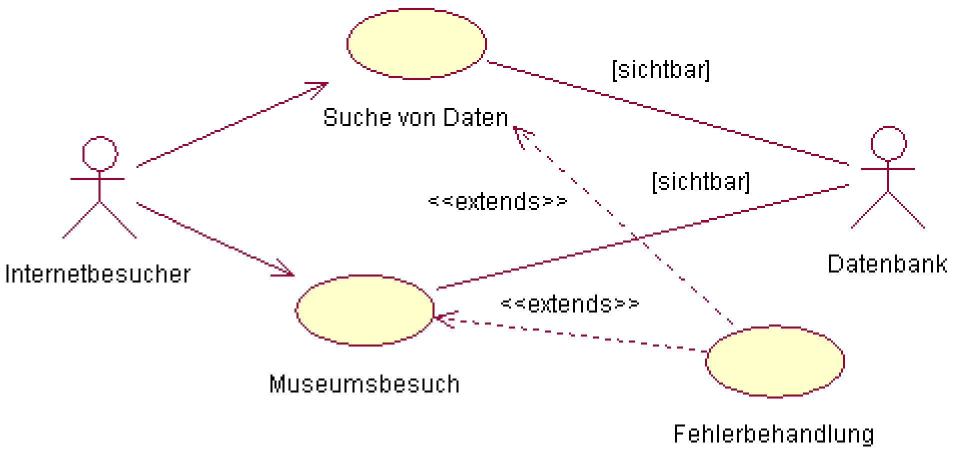
11.13.12 Secondary Scenarios

Der Benutzer kann zu jeder Zeit mit *Abbrechen* den Vorgang abbrechen und kehrt zum Ausgangszustand zurück.

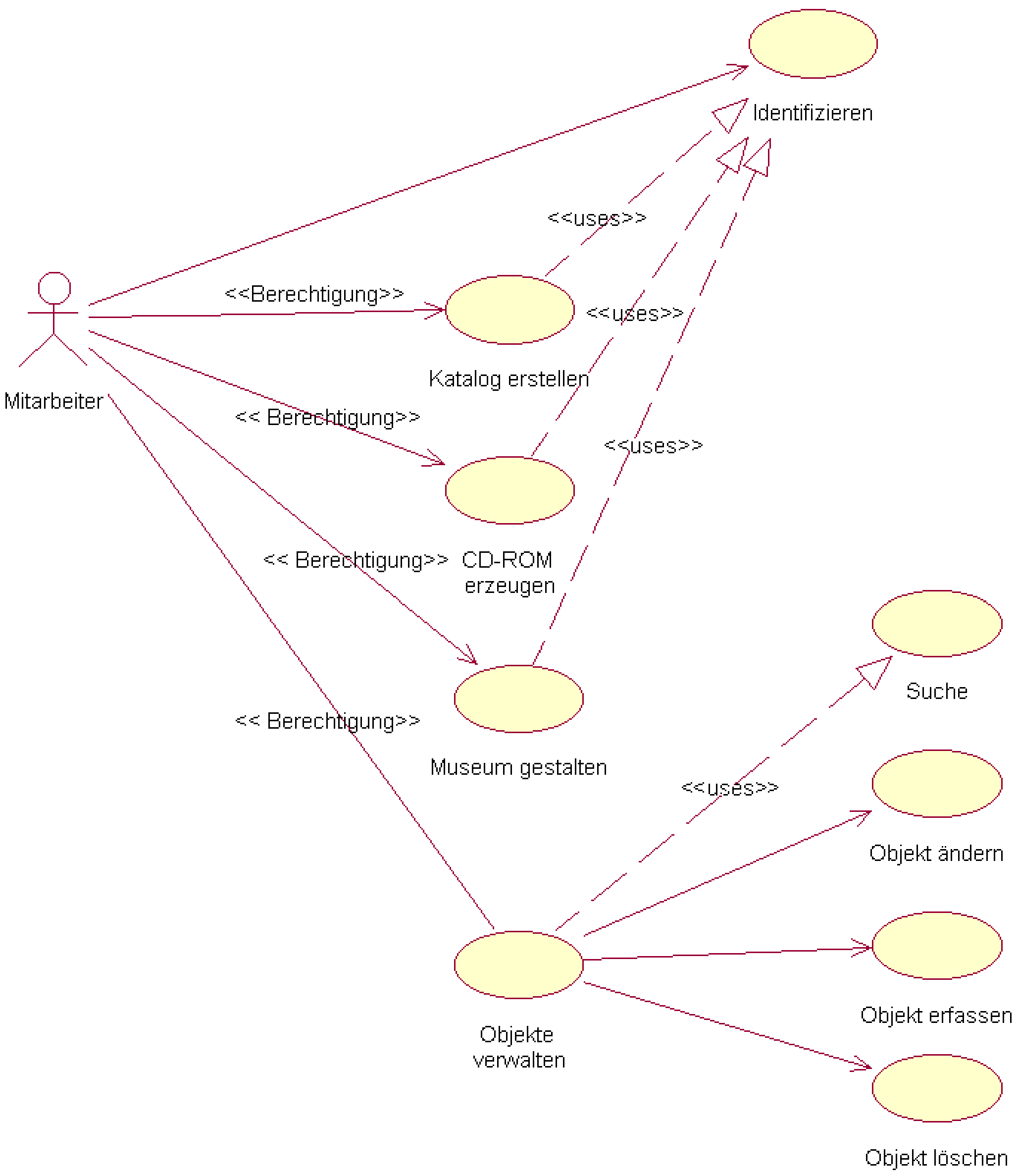
Kapitel 12

Use Case Diagramm

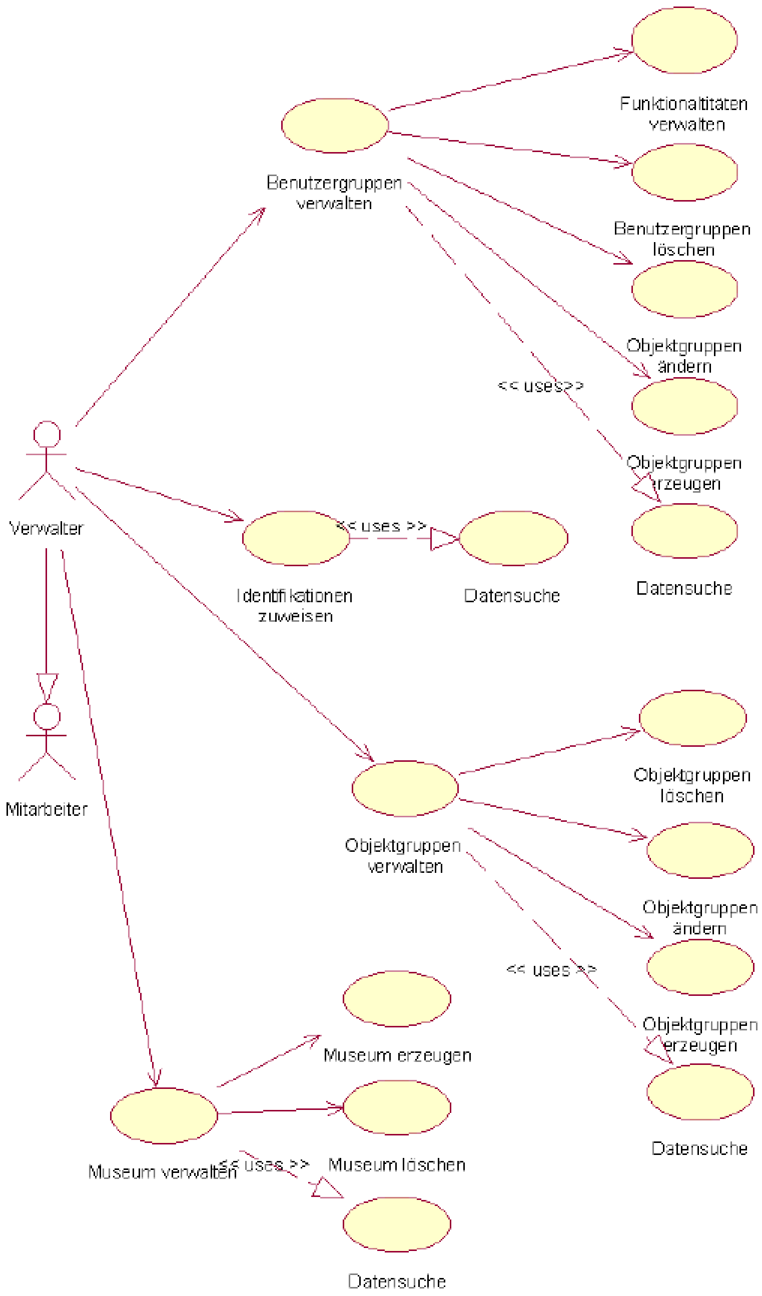
12.1 Internetbesucher



12.2 Mitarbeiter



12.3 Verwalter



Kapitel 13

Aufwandsabschätzung

13.1 UUCP: unadjusted use case points

Die Schätzung der Dauer des Projekts wird mit Hilfe von [7] gemacht.

13.1.1 Total actor weight for order processing

Die Actors werden je nach Komplexität gewichtet. Dabei wird berücksichtigt, wie sie mit dem System kommunizieren.

Actor Typ	Beschreibung	Faktor
simple	Programm-Schnittstelle	1
average	interaktive oder protokollgesteuerte Schnittstelle	2
complex	graphische Schnittstelle	3

Es gibt folgende Aktuere mit Gewichtung in unserem System:

- Verwalter: Gewicht 3
- Systemadministrator: Gewicht 2
- Mitarbeiter: Gewicht 3
- Internetbesucher: Gewicht 3
- CD-Brenner: Gewicht 1
- Drucker: Gewicht 1
- Datenbank: Gewicht 1

- Backupmedium: Gewicht 1

Es wird nun gezählt, wieviele Actors von jedem Typ vorhanden sind. Die jeweilige Anzahl wird mit einem Faktor multipliziert. Die Summe der Produkte ist die Gewichtung der Actors.

Total actor weight for order processing $= 3 * 3 + 1 * 2 + 4 * 1 = 15$

13.1.2 Total use case weight for order processing

Unter einem Transaktionstep versteht man alle Transaktionen, inklusive solcher von sekundären Szenarien, wobei eine Transaktion eine atomare Menge von Aktionen darstellt, welche entweder vollständig oder überhaupt nicht ausgeführt wird.

Use-Case Typ	Beschreibung	Faktor
simple	3 oder weniger Transaktionen	5
average	4 bis 7 Transaktionen	10
complex	mehr als 7 Transaktionen	15

- **Erfassung, Ändern und Löschen von Artefakten - complex**
  Transaktionsteps:

  1. Erstellen eines neuen Objekts zu einer Objektgruppe.
  2. Eingeben und Ändern der Daten eines Objekts. (auch ob Daten sichtbar sind)
  3. Ein Objekt löschen.
  4. Eingeben und Ändern von Verweisen zu Dateien und anderen Objekten.
  5. Ein Objekt suchen.
  6. Ändern der Eigenschaften eines Objekts, etwa welchen Objektgruppen sie zugeordnet ist, ob das Objekt sichtbar ist, etc.
  7. Eingegebene Daten an Datenbank schicken.
  8. Daten von Objekten von der Datenbank empfangen.
  9. Fehlerhandling, wenn die Kommunikation mit der Datenbank nicht funktioniert.

- **Katalog erstellen - average**
  Transaktionsteps:

  1. Daten für den Katalog kennzeichnen
  2. Ausgewählte Daten auf Sichtbarkeit und Darstellbarkeit prüfen.
  3. pdf-File erzeugen
  4. pdf-File an Drucker schicken
  5. Fehlerhandling, Fehler beim pdf-File erzeugen
  6. Fehlerhandling, Fehler beim pfd-File an den Drucker schicken

- **CD-ROM erzeugen - average**
  Transaktionsteps:

  1. Museum, welches auf CD-ROM abgebildet werden soll, auswählen.
  2. Dateistruktur und Programm, welches dem ausgewählten Museum entspricht, erzeugen
  3. Dateistruktur und Programm an den CD-Brenner schicken.
  4. Fehlerhandling, Fehler beim Erzeugen der Dateistruktur
  5. Fehlerhandling, Fehler beim Schicken der Dateistruktur

- **Suchen von Objekten - complex**
  Transaktionsteps:

  1. Eingegebene Suchbegriffe in entsprechende Anfrage an die Datenbank umwandeln.
  2. Anfrage an die Datenbank schicken.
  3. Ergebnis von der Datenbank empfangen.
  4. Ergebnis entsprechend darstellen.
  5. Daten eines ausgewählten Objekts aus dem Ergebnis aus der Datenbank beantragen.
  6. Daten des ausgewählten Objekts von der Datenbank empfangen.
  7. Ausgwähltes Objekt mit seinen Daten darstellen (nur sichtbare Daten).
  8. Fehlerhandling, Kommunkation mit der Datenbank funktioniert nicht.
  9. Fehlerhandling, Probleme bei der Darstellung der Daten.

- **Museumsbesuch - complex**
  Transaktionsteps:

  1. Auswahl eines Museums.
  2. Auswahl eines Stockwerks.
  3. Auswahl eines Raums.
  4. Auswahl einer Vitrine.
  5. Auswahl eines Objekts in einer Vitrine.
  6. Daten zu einem Objekt aus der Datenbank holen.
  7. Daten den Darstellungsvorschriften entsprechend darstellen.
  8. Fehlerhandling, wenn Zugriff auf Daten nicht funktioniert.

- **Datensicherung - average**
  Transaktionsteps:

  1. Sichern der Daten der Datenbank.
  2. Sichern eines Museums.
  3. Beim Ausfall der Datenbank auf das gesicherte Museum zurückgreifen.
  4. Rekonstruktion der Datenbank aus den Backups.
  5. Fehlerhandling, falls bei der Sicherung der Daten Probleme auftreten
  6. Fehlerhandling, falls bei der Sicherung des Museums Probleme auftreten.

- **Rechte verwalten - average**
  Transaktionsteps:

  1. Erzeugen einer neuen Berechtigung.
  2. Löschen einer Berechtigung.
  3. Eine Berechtigung suchen.
  4. Feststellen, welche Benutzergruppen diese Berechtigungen enthalten.

- **Objektgruppen verwalten - average**
  Transaktionsteps:

1. Eine Objektgruppe erstellen.
2. Eine Objektgruppe löschen.
3. Eine Objektgruppe suchen.
4. Eine Objektgruppe ändern (Die Eigenschaften einer Objektgruppe ändern.)
5. Ermitteln, welche Benutzergruppe, eine Objektgruppe verwendet.
6. Ermitteln, welche Objekte in einer Objektgruppe enthalten sind.

- **Benutzergruppen verwalten - average**
  Transaktionsteps:

  1. Erzeugen einer neuen Benutzergruppe.
  2. Löschen einer Benuztergruppe.
  3. Eine Benutzergruppe suchen.
  4. Eigenschaften einer Benutzergruppe änderen (welche Objektgruppen und welche Berechtigungen sie enthält)
  5. Festellen, welche Mitarbeiter einer Benutzergruppe zugewiesen sind.
  6. Einen Mitarbeiter einer Benutzergruppe zuweisen.
  7. Die Zuweisung eines Mitarbeiters zu einer Benuztergruppe aufheben.

- **Identifizierung eines neuen Mitarbeiters - average**
  Transaktionsteps:

  1. Erstellung eines neuen Mitarbeiters im System inkl. Zuweisung eines Usernamens (Bedingung: der Mitarbeiter bzw. der Username sind im System noch nicht vorhanden).
  2. Suche ob ein Mitarbeiter vorhanden ist.
  3. Zuweisung eines Paßwortes an einen erstellten Mitarbeiter.
  4. Löschen eines Mitarbeiters.

- **Museumsgestaltung - simple**
  Transaktionsteps:

  1. Erzeugen eines neuen Museums.
  2. Löschen eines Museums.

- **Stockwerksgestaltung - simple**
  Transaktionsteps:
  1. Erzeugen eines Stockwerks.
  2. Löschen eines Stockwerks.
  3. Ändern der Eigenschaften eines Stockwerks, wie Kommentar, Position im Museum, Darstellungsart, etc.
- **Raumgestaltung - simple**
  Transaktionsteps:
  1. Erzeugen eines Raums.
  2. Löschen eines Raums.
  3. Ändern der Eigenschaften eines Raums, wie Kommentar, Position im Stockwerk oder Darstellungsart.
- **Vitrinengestaltung - average**
  Transaktionsteps:
  1. Erzeugen einer Vitrine in einem Raum.
  2. Löschen einer Vitrine.
  3. Einfügen eines Objekts in eine Vitrine und Festlegung dessen Darstellung
  4. Löschen eines Objekts in einer Vitrine.
  5. Darstellung eines Objekts in einer Vitrine verändern (Position in der Vitrine, Darstellungsart etc.)
  6. Ändern der Eigenschaften einer Vitrine, wie Kommentar, Position im Raum, Darstellungsart etc.
- **Prüfung von Berechtigungen - average**
  Transaktionsteps:
  1. Benutzergruppen, denen der Mitarbeiter angehört, suchen.
  2. Anzeigen des Ergebnisses
  3. Kontrollieren, ob die gewünschte Berechigung in einer dieser Gruppen enthalten ist.
  4. Fehlerhandling, falls etwas bei der Überprüfung schiefläuft.

Total use case weight for order processing $= 3*5+9*10+3*15 = 150$
UUCP $= 15+150 = 165$

13.2 Weighting Technical Factors

13.2.1 TCF: technical complexity factor

Die Gewichtung 0 bedeutet, dass der Faktor irrelevant ist und die Gewichtung 5 bedeutet, dass der Faktor wesentlich ist.

Faktor-Nummer	Faktor-Beschreibung	Gewicht
T1	Verteiltes System	2
T2	Antworts- und Zugriffszeit ausschlaggebend	1
T3	Endbenutzer-Effizienz (online)	1
T4	Komplexe interne Berechnungen	1
T5	Code muss wiederverwendbar sein	1
T6	Einfach zu installieren	0,5
T7	Einfach zu benutzen	0,5
T8	Portabel	2
T9	Einfach zu verändern	1
T10	Gleichzeitige Nutzung	1
T11	Beinhaltet spezielle Sicherheitskonzepte	1
T12	Bietet direkten Zugang für Dritte	1
T13	Spezielle Benutzungsschulungen werden benötigt	1

Gewichtung:

Faktor	Gewicht	Wert	Grund
T1	2	3	Alle Tools greifen auf zentrale Datenbank zu
T2	1	5	mehrere Anfragen gleichzeitig
T3	1	5	Teilweise langsame Internetanbindung der Internetbesucher
T4	1	3	z. B. Anfrage an Datenbank und Auswertung
T5	1	3	mehrere Klassen, Routinen können wieder verwendet werden
T6	0,5	1	Installation durch SE AG
T7	0,5	5	Internetbesucher müssen sich leicht zurechtfinden
T8	2	5	versch. Browser u. Versionen, Datenbank ist auswechselbar
T9	1	3	Auftraggeber ändert hin und wieder seine Meinung
T10	1	5	Durch Internetbesucher gleichzeitig
T11	1	3	Berechtigungsvergabe durch Verwalter
T12	1	5	Internetbesucher haben Zugang
T13	1	2	Für Mitarbeiter: Schulungen

TFactor $= 6 + 5 + 5 + 3 + 3 + 0,5 + 2,5 + 10 + 3 + 5 + 3 + 5 + 2 = 53$
TCF $= 0,6 + (0,01 * 53) = 1.13$

13.2.2 EF: environmental factor

Faktor-Nummer	Faktor-Bescheibung	Gewicht
F1	Vertraut mit dem Rational Unified Process	1,5
F2	Applikationserfahrung	0,5
F3	Objektorientierte Erfahrung	1
F4	Kompetent in Analyse (Leiter)	0,5
F5	Motivation	1
F6	Stabilität Voraussetzungen	2
F7	Teilzeitarbeiter	-1
F8	Schwere Programmiersprache	-1

- F1 - F4: 0 bedeutet, es gibt keine Erfahrung in diesem Bereich, 5 bedeutet Experte.
- F5: 0 bedeutet keine Motivation, 5 bedeutet sehr motiviert.
- F6: 0 bedeutet sehr unstabile Bedingungen, 5 bedeutet unveränderliche Bedingungen.
- F7: 0 bedeutet keine Teilzeitarbeiter, 5 bedeutet nur Teilzeitarbeiter.
- F8: 0 bedeutet einfach zu verwendende Programmiersprache, 5 bedeutet eine komplizierte Programmiersprache.

Faktorr	Gewicht	Wert	Grund
F1	1,5	3	Schon beim letzten Projekt teilw. benutzt
F2	0,5	5	Team hat schon andere Applikationen erst.
F3	1	5	sehr erfahren
F4	0,5	4	recht erfahren
F5	1	4	hoch motiviert und gut bezahlt
F6	2	2	Teilw. Änderungen der Vorraussetzungen
F7	-1	0	Keine
F8	-1	3	durchschnittlich schwere Programmiersprachen

EFactor $= 4,5 + 2,5 + 5 + 2 + 4 + 4 + 0 - 3 = 19$
EF $= 1,4 + (-0,03 * 19) = 0,83$

13.3 UCP: use case points

UCP $= 165 * 1,13 * 0,83 = 154,75 \approx 155$

13.4 Schätzung der Dauer des Projekts

Um die Mann-Stunden zu bekommen, muss jetzt noch mit einem Faktor multipliziert werden.
Aus der Untersuchung von F1 bis F6 und F7 bis F8 geht hervor, dass der Faktor 20 ist.
Mann-Stunden $= 155 * 20 = 3100\ Stunden/Mann$.
Unter Annahme von 38 Stunden/Woche ergibt sich:
Mann-Wochen $= 3100/38 = 81,57\ Wochen/Mann$
Es arbeiten 8 Personen an diesem Projekt, daher ergibt sich:
Team-Wochen $= 81,57/8 = 10.2 \approx 11\ Wochen/Team$.
Es werden von unserer Firma 2 Wochen zur Sicherheit miteinkalkuliert, daher:
Geschätzte Dauer des Projekts: 14 Wochen.

Kapitel 14

Kostenabschätzung

14.1 Software

Unsere Firma verrechnet 1200 Schilling/Stunde.
Daher ergibt sich der Preis: $14(Wochen) * 8(Personen) * 38(Stunden) * 1200(Schilling/Stunde) = 5.107.200$
Gerundete Kosten softwareseitig = <u>ATS 5.110.000.-</u>

14.2 Hardware

- Serverkosten: ATS 850.000.-
- Errichtung der Standleitung: ATS 100.000.-
- sonstige Kosten ATS 100.000 .-

Ergibt in Summe: ATS 1.050.000.- **Kosten hardwareseitig = <u>ATS 1.050.000.-</u>**

14.3 Gesamtpreis

Gesamtpreis = 5.110.000 + 1.050.000 = <u>ATS 6.160.000.-</u>

14.4 Laufende Kosten

- Kosten für Standleitung: ATS 50.000.-
- Wartungsvertrag: ATS 30.000.-, incl 1.000 Schilling/Stunde - ungefähr 25 Stunden/Monat ergibt: ATS 55.000.-

Ergibt: **Laufende Kosten** = **105.000 Schilling/Monat**

Chapter 15

Architektur

15.1 Vorgehensweise

Um die Architektur des Systems zu bestimmen, wird es in mehrere Teile (Subsysteme) zerlegt. Die Subsysteme bestehen aus einzelnen Modulen, die miteinander in Beziehung stehen können. Die Beziehungen zwischen den Subsystemen werden über Interfaces definiert.

Es gibt nun verschiedene Möglichkeiten, das System in Subsysteme zu zerlegen. Dazu bietet es sich an, von vordefinierten patterns (Mustern) auszugehen, die jeweils als Vorlagen dienen, das System zu definieren, zu unterteilen und zu organisieren.

Unsere Vorgehensweise besteht nun darin, zuerst die wichtigsten Use Cases zu gruppieren. Diese Gruppierungen liefern uns eine Partitionierung der Gesamtfunktionalität des Systems, sodass in der Entwicklungsphase einzelne Subsysteme möglichst unabhängig voneinander (Verbindungen zu anderen Subsystemen sind ja über die Interfaces wohldefiniert) in mehreren Iterationen entwickelt werden können.

Unsere Vorgehensweise bei der Zerteilung in Subsysteme und Module wird von folgenden Vereinbarungen bestimmt:

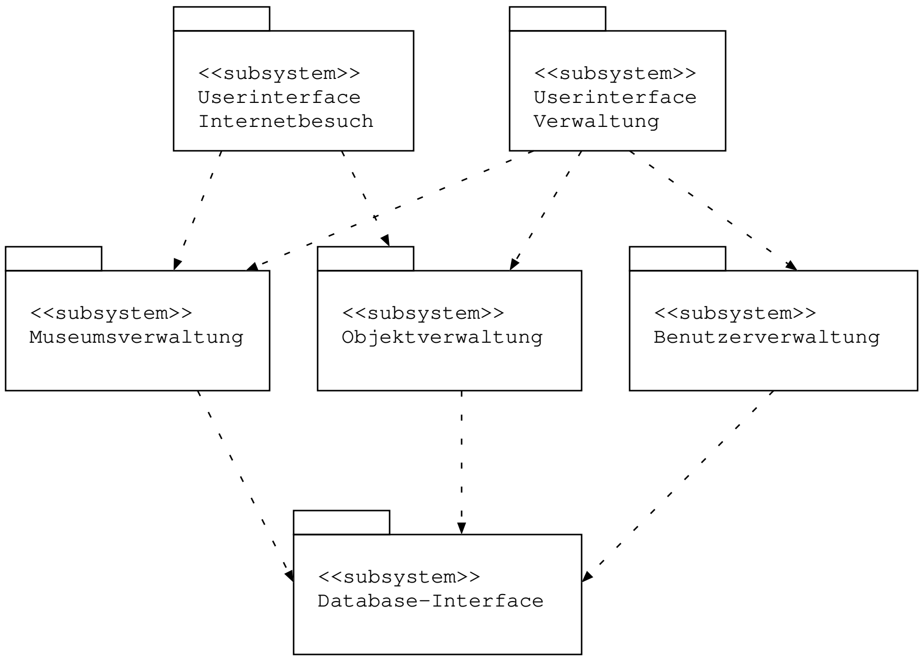
1. Verwendung des Object-Oriented-Pattern
2. Drei Schichten-Modell:
    - User-Interface
    - Business Layer
    - Database
3. Verwendung von Java

4. XML wird zur Speicherung von Daten verwendet

15.2 Subsysteme

Ausgehend von den 3 Schichten User-Interface, Business-Layer und Database wird das System in Subsysteme aufgeteilt, die jeweils wieder aus Modulen bestehen.

Die Interfaces zwischen den Schichten werden weitgehend durch die Module (Klassen) des Business-Layer bestimmt, sodass der **Business-Layer eine zentrale Rolle in der Architektur** einnimmt.



15.2.1 Subsysteme des User-Interface

User-Interface Internetbesuch

User-Interface Verwaltung

Die Module der Subsysteme des User-Interface sind die Fenster, die im vorigen Kapitel dargestellt worden sind. Ihre Beziehungen untereinander sind bereits in den zugehörigen Diagrammen dargestellt worden.

Ihre Beziehungen zu den Subsystemen und Modulen des Business-Layers werden über Verknüpfungen der OK-, Cancel- etc. Buttons mit den Methoden der Module im Business-Layer hergestellt. Direkte Beziehungen zu den Modulen der Database sind nicht vorgesehen.

Implementiert werden die Subsysteme des User-Interface als **Java-Applets**.

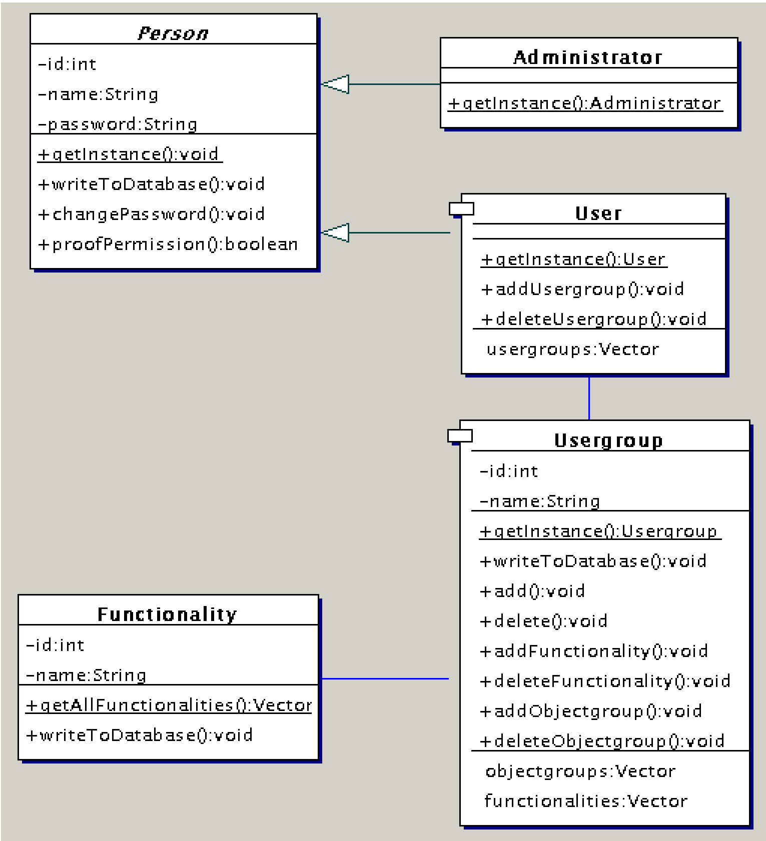
15.2.2 Subsysteme des Business-Layer

Die Module der Subsysteme, die zum Business Layer gehören, werden als Klassen spezifiziert. Die Interfaces der Klassen werden durch ihre Methoden, die Beziehungen untereinander durch Verweise auf andere Instanzen von den jeweiligen Klassen hergestellt.

Implementiert werden die Klassen des Business-Layers in **Java-Servlets**.

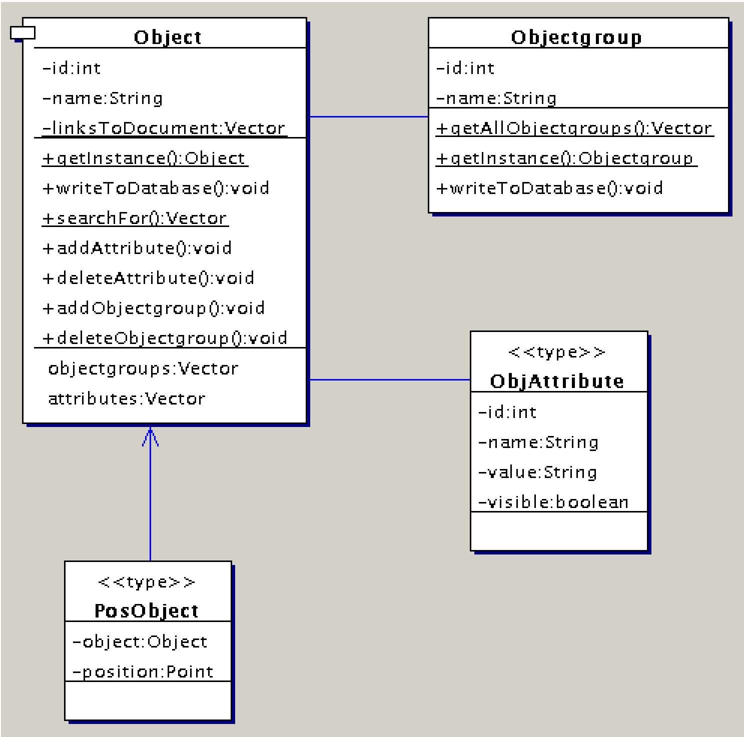
Die Kommunikation zu den User-Interfaces erfolgt über die definierten Methoden.

Benutzerverwaltung



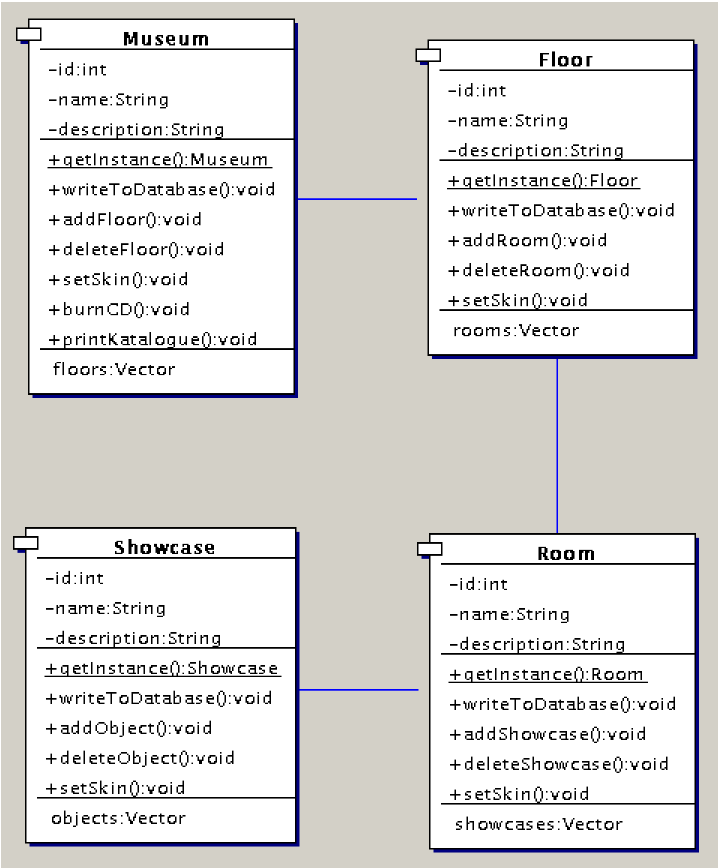
Verknüpfung zur Objektverwaltung: Usergroup-Objectgroup

Objektverwaltung



Verknüpfung zur Museumsverwaltung: PosObject-Showcase

Museumsverwaltung



15.2.3 Subsysteme der Database-Schicht

Das Database-Interface

Der Datenaustausch des Database-Interface mit den Subsystemen und Modulen des Business-Layers erfolgt mittels Austausch von SQL-Statements.

15.3 Durchwanderung der Use Cases in der Architektur

15.3.1 BenutzergruppeErstellen

Um das Anwendungsfeld der einzelnen Operationen zu verdeutlichen wird teilweise ein Pseudocode verwendet.

1. *Verwaltungs UserInterface:* Auswahl: Benutzergruppe erstellen
2. *Verwaltungs UserInterface:* Eingabe eines Bezeichners für eine Benutzergruppe
3. *Usergroup.getInstance(<Bezeichner>)*
   Es wird überprüft, ob schon eine Benuztergruppe mit dieser Bezeichnung existiert. Wenn nicht, wird null zurückgegeben, ansonsten wird eine Referenz auf dieses Objekt übergeben.
4. if (usergr ist ungleich null)

   (a) *Verwaltungs UserInterface:* Es wird mitgeteilt, dass diese Gruppe schon existiert, aber wenn eine Kopie von dieser Usergroup erstellt werden will, soll ein neuer Bezeichner eingegeben werden.

   (b) *newUsergr = Usergroup.getInstance(<neuer Bezeichner>)*
   Es wird wird überprüft, ob eine Usergroup mit diesem Bezeichner existiert.

   (c) if(newUsergr ist nicht null)

      i. *Verwaltungs UserInterface:* Erneute Eingabe des Bezeichners.
      ii. springe zu Punkt 4b

   (d) if(newUsergr ist null)

      i. *newUsergr = new Usergroup(<neuer Bezeichner>)*
      Erzeugen eines neuen Objekts von Usergroup
      ii. *funct = usergr.functionalities()*
      for(i=1 to funct.length())

         - *newUsergr.addFunctionality(funct[i])*

      Alle Funktionalitäten, welche in usergr enthalten sind werden, in newUsergr eingetragen.
      iii. *ogroups = usergr.opjectgroups()*
      for(i=1 to ogroups.length()

         - *newUsergr.addObjectgroup(ogroups[i])*

      Alle Objektgruppe, welche in usergr enthalten sind, werden in newUsergr eingetragen.

iv. usergr = newUsergr Ab jetzt wird mit der neu erzeugten und mit den Daten der alten Usergroup geladenen Usergroup weitergearbeitet.

v. end if

(e) if(usergr ist null)

i. *usergr = new Usergroup(<Bezeichner>)*
Ein neues Objekt von Usergroup wird erzeugt und unter usergr abgelegt

ii. end if

(f) end if

5. *Objectgroup.getAllObjectgroups()*
Es werden alle allgemein vorhandenen Objektgruppen ermittelt.

6. *Objectgroup.getAllFunctionalities()*
Es werden alle allgemein vorhandenen Funktionalitäten ermittelt.

7. *usergr.objectgroups()*
Es werden alle beim Usergroup Objekt eingetragenen Objektgruppen ermittelt.

8. *usergr.functionalities()*
Es werden alle beim Usergroup Objekt eingetragenen Funktionalitäten ermittelt.

9. *Verwaltungs UserInterface:* Alle allgemein vorhandenen Objektgruppen und Funktionalitäten werden angezeigt und jene, welche schon beim Objekt eingetragen wurden, als aktiviert angezeigt.

10. *Verwaltungs UserInterface:* Es können Objektgruppen und Funktionalitäten aktiviert und deaktiviert werden.

11. *Verwaltungs UserInterface:* Auswahl von Objektgruppe speichern.

12. if(mindestens 1 Funktionalität und mindestens 1 Objektgruppe sind aktiviert)

    (a) *usergr.save()* Die Daten des Usergroup Objekts werden in der Datenbank abgespeichert.

13. else

    (a) Gehe zurück zu Punkt 9.

14. end if

15.3.2 ObjekteSuchen

1. Internetuser startet Applet für ObjekteSuchen im Browser durch Mausklick auf den entsprechenden Button
2. Internetuser gibt in die Maske Suchbegriff(e) und erweiterte/einfache Suche ein.
3. Applet übergibt der statischen Suchfunktion searchFor() am Server die Parameter Suchbegriffe(e) und erweiterte/einfache Suche
4. Suchfunktion durchsucht über das Datenbankinterface die Datenbanken nach entsprechenden Objekten
5. Datenbankinterface gibt Objekte mit Verweis auf die jeweilige Vitrine zurück
6. Suchfunktion testet gefundene Objekte auf Mehrfachvorkommen und auf Sichtbarkeit
7. Suchfunktion gibt Liste mit gefundenen und geprüften Objekten mit Referenzen auf die Datenbank an das Applet retour
8. Applet stellt gefundene Objekte in Form einer Liste im Browser dar
9. Internetuser wählt Objekt aus Liste aus
10. Vitrine mit Objekt wird dargestellt

15.3.3 MuseumBesuchen

1. Museum auswählen
   (a) Der Internetuser hat die Möglichkeit, ein Museum bzw. aus mehreren Museen eines auszuwählen.
2. Stockwerk auswählen
   (a) "Museum" hat eine Liste, in dem alle möglichen Stockwerke aufgelistet sind. Der Internetuser wählt ein Stockwerk aus und kommt so dorthin.
3. Raum auswählen

(a) Hier wird wieder aus einer Liste aller verfügbaren Räume der gewünschte ausgewählt.

4. Vitrine auswählen

   (a) Die Liste mit allen Vitrinen aus "Room" steht wieder zur Verfügung. Eine Vitrine wird ausgewählt.

5. Objekt auswählen

   (a) Aus der Liste der in einer Vitrine dargestellten Objekte aus "Showcase" wird das darzustellende Objekt ausgewählt.

   (b) Die Position und die Informationen zu einem Objekt aus "Object" werden geholt.

15.3.4 VitrineErstellen

Mit der Durchwanderung des Use Case VitrineErstellen im Klassendiagramm wird überprüft, ob die zur Verfügung gestellten Funktionalitäten des Klassendiagramms ausreichend sind, damit das Use Case vollständig implementiert ist. Mit Hilfe des schon oben genannten Flow of Events für den Use Case VitrineErstellen wird der Use Case durchwandert und dabei wird betrachtet, ob jeder Punkt im Flow of Events implementiert ist und was die jeweiligen Klasssen und Methoden dazu beitragen.

1. Der Vorgang beginnt mit der Selektion „Vitrine erstellen".
   Im Klassendiagramm:
   Mit der Selektion „Vitrine erstellen" wird die Methode *getInstance():Showcase* in der Klasse *Showcase* aufgerufen. Mit dieser Methode(=Konstruktor) wird eine Instanz von Showcase (=Vitrine) gebildet und die Attribute *id* (=Primärschlüssel der Vitrine), *name* (Titel der Vitrine) und description (=Beschreibung der Vitrine) initalisiert.

2. Festlegung, wie die Vitrine dargestellt werden soll.
   Im Klassendiagramm:
   Mit der Methode *setSkin(): void* in der Klasse *Showcase* wird das Aussehen der Vitrine festgelegt.

3. Beschreibung der Vitrine
   Im Klassendiagramm:
   Wie schon oben erwähnt, wird mit der Methode *getInstance():Showcase* das Attribut description initalisiert.

4. if (Vitrine Raum zuweisen)

    (a) Vitrine Raum zu weisen.

    Im Klassendiagramm:
    Um einer Vitrine einen Raum zu zuweisen, wird die Mehtode *addShowcase():void* von der Klasse *Room* aufgerufen. Das Löschen einer Zuweisung einer Vitrine zu einem Raum erfolgt mit *deleteShowcase():void* aus der Klasse *Room*. Im Vector *showcases* in der Klasse *Room* wird gespeichert, welche Vitrinen einem Raum zugewiesen sind.

5. if (Objekt in Vitrine positionieren)

    (a) while (weitere Objekte positionieren)

        i. Objektpositionierung

    (b) end

    Im Klassendiagramm:
    Mit den Methoden *addObjekt():void* bzw. *deleteObject():void* von der Klasse *Showcase* kann ein Objekt in einer Vitrine positioniert bzw. gelöscht werden. Welche Objekte einer Vitrine zugeordnet sind, wird im Vector *objects* gespeichert. Die Position des Objekts in der Vitrine wird im Typ *PosObjekt* festgehalten.

6. if (Speichern)

    (a) Der ganze Vorgang „Vitrine erstellen“ wird in der Datenbank abgespeichert.

    (b) Ende des Vorgangs „Vitrine erstellen“.

7. if (Abbrechen)

8. Ende des Vorgangs „Vitrine erstellen“ ohne den Vorgang in der Datenbank zu speichern.

Im Klassendiagramm:
Immer, wenn ein Vorgang abgeschlossen ist, wie z.B. Objekt positionieren, wird das Ergebnis in der Datenbank abgespeichert. D.h. *Speichern* bedeutet einfach die Sitzung ist abgeschlossen. Mit *Abbrechen* wird mittels einer Hilfsmethode in *Showcase* alles, was in dieser Sitzung „Vitrine erstellen“ in der Datenbank gespeichert wurde, wieder gelöscht.

Chapter 16

Projektplan für die Construction Phase

16.1 Kategorisierung der Use Cases

Wichtigkeit und Risiko werden jeweils in Stufen von 1-10 (10 = höchste Wichtigkeit bzw. größ tes Risiko), Entwicklungszeit in geschätzten Mannwochen angegeben.

16.1.1 Objektverwaltung

16.1.1.1 Artefakte eingeben

- Wichtigkeit: 10
- Risiko: 10
- Entwicklungszeit: 15

16.1.1.2 Artefakte löschen

- Wichtigkeit: 10
- Risiko: 5
- Entwicklungszeit: 1

16.1.1.3 Objektgruppen erstellen

- Wichtigkeit: 10

- Risiko: 6
- Entwicklungszeit: 5

16.1.2 Benutzerverwaltung

16.1.2.1 Benutzergruppe erstellen

- Wichtigkeit: 10
- Risiko: 10
- Entwicklungszeit: 16

16.1.2.2 Rechte festlegen

- Wichtigkeit: 10
- Risiko: 7
- Entwicklungszeit: 5

16.1.2.3 Identifizierung eines neuen Mitarbeiters

- Wichtigkeit: 10
- Risiko: 6
- Entwicklungszeit: 1

16.1.3 Museumsverwaltung

16.1.3.1 Museum erstellen

- Wichtigkeit: 10
- Risiko: 6
- Entwicklungszeit: 5

16.1.3.2 Vitrine erstellen

- Wichtigkeit: 10
- Risiko: 8
- Entwicklungszeit: 12

16.1.4 Sonstige

16.1.4.1 Katalog erstellen

- Wichtigkeit: 5
- Risiko: 1
- Entwicklungszeit: 5

16.1.4.2 CD-ROM erzeugen

- Wichtigkeit: 5
- Risiko: 1
- Entwicklungszeit: 5

16.1.4.3 Suchen von Objekten

- Wichtigkeit: 8
- Risiko: 3
- Entwicklungszeit: 5

16.1.4.4 Datensicherung

- Wichtigkeit: 10
- Risiko: 1
- Entwicklungszeit: 2

16.2 Iterationen

Aufgrund der oben geschätzten Entwicklungszeiten, Wichtigkeiten und der Größe des Entwicklungsteams wird eine Iterationslänge von 2 Wochen festgelegt.

Insgesamt wird es 7 Iterationen geben.

Iteration 1

Artefakte eingeben und löschen, Objektgruppe erstellen
Teil des User-Interface Verwaltung
Objektverwaltung

Iteration 2

Kommunikation zur Datenbank

Datenbank-Interface

Iteration 3

Museum erstellen, Vitrine erstellen, Museumsbesuch

Museumsverwaltung
Teil des User-Interface Verwaltung

Iteration 4

Darstellung des Museums im Internet

User-Interface Internetbesuch

Iteration 5

Benutzergruppe erstellen, Identifizierung eines neuen Mitarbeiters, Rechte festlegen

Benutzerverwaltung
User-Interface Verwaltung fertigstellen

Iteration 6

Suchen von Objekten + Markierung der gefundenen Objekte im Museum

Alle Subsysteme des Business-Layer fertigstellen

Iteration 7

Datensicherung, CD-ROM erzeugen, Katalog erstellen

Chapter 17

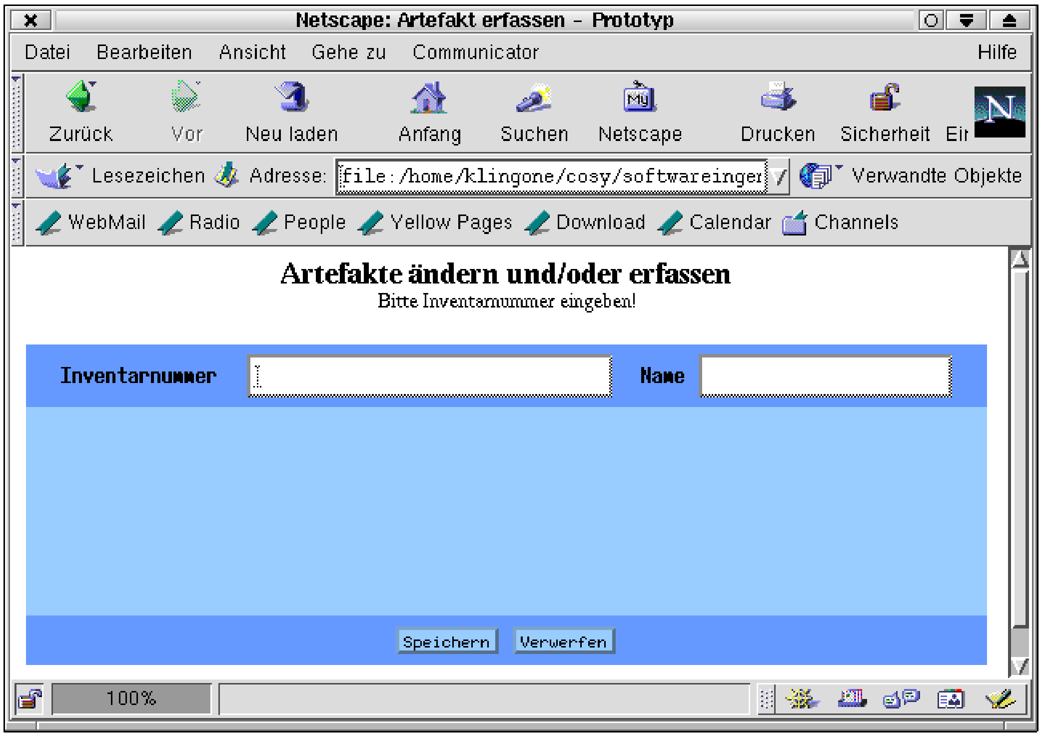
Prototyp

Es wird ein Protoyp erstellt, um die größten Risiken abzufangen.
Wir haben uns entschlossen, die Eingabe von Daten als Prototyp zu gestalten.
Das Grundprinzip dieser Funktion: ein Objekt erstellen, ein Objekt suchen, ein Objekt abspeichern und die entsprechende Zusammenarbeit mit der Datenbank findet man überall im Projekt wieder. Wenn hier beim Grundkonzept Probleme auftreten, haben diese sehr große Auswirkungen.
Im Prototyp wird realisiert:
Über eine Eingabemaske ist es möglich, Daten über ein Artefakt einzugeben und auch Daten über ein Artefakt wieder auszulesen.
Ziel ist es, nicht eine perfekte Implementierung zu erzielen, z.B. können beim Prototypen noch nicht beliebig viele Attribute eingeben werden, sondern die wichtigsten Funktionalitäten zu implementieren. Wir wollen Daten aus der Datenbank lesen und in die Datenbank schreiben, ebenso wollen wir mit den Daten im Programm effektiv arbeiten und gewährleisten, dass die Datenbank konsistent bleibt.
Wir haben uns dazu entschlossen, dass wir im Programm mit Objekten arbeiten, welche beim Erzeugen ihre Daten aus der Datenbank lesen.
Am Schluß der Bearbeitung werden die Daten des Objekts wieder in die Datenbank geschrieben.
Kleinere Änderungen werden direkt auf das Objekt ausgeführt, was schneller ist, als wenn jedesmal die Datenbank angesprochen wird.
Die Konsistenz der Daten kann man in den Griff bekommen, indem man sich merkt, wer die Daten gelesen hat, und nur dieser die Daten auch neu abspeichern darf (noch nicht im Prototypen realisiert, da wir erst 1 Benutzer haben).
Beim Prototypen wird über eine JDBC-Schnittstelle mit einer MySql-Datenbank kommuniziert, welche die entsprechenden Daten enthält.

Figure 17.1: Startseite

17.1 Handlungsablauf

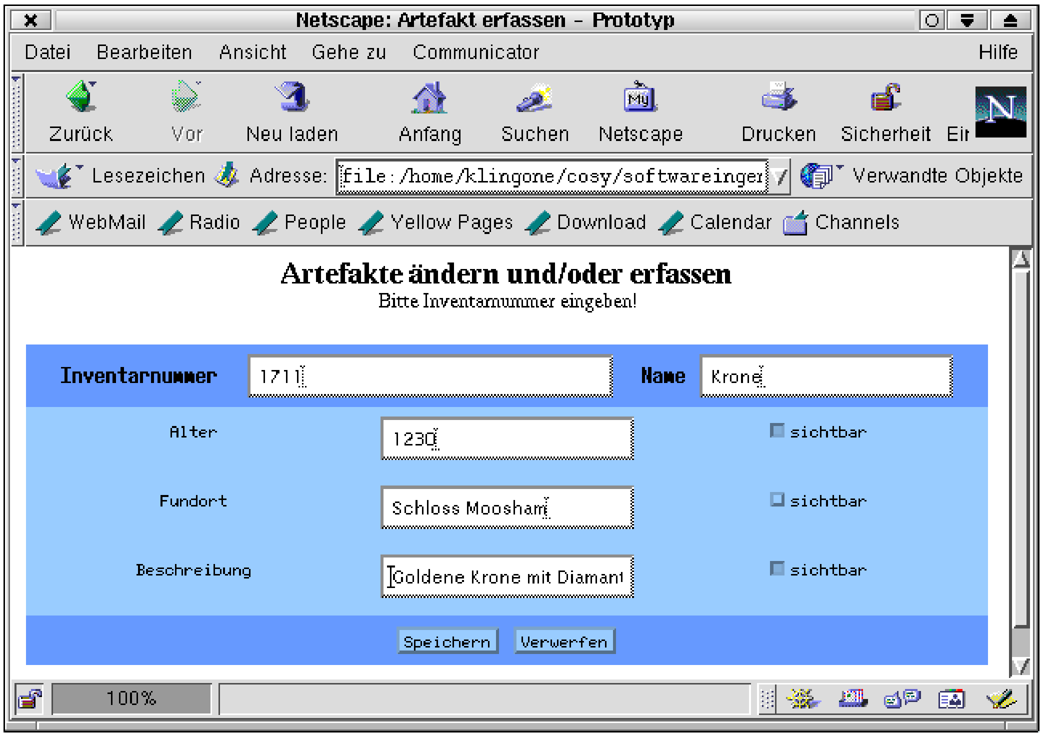
Das Programm wird von der html Seite index.htm gestartet. Als 1. erscheint die Startseite:



Nur das Feld für die Inventarnummer ist aktiv. Der Name kann noch nicht eingegeben werden.
Als nächstes muss der Benutzer eine Inventarnummer eingeben. Es wird in der Datenbank gesucht, ob schon ein Artefakt mit dieser Inventarnummer existiert.

Figure 17.2: Ändern der Attribute eines vorhandenen Artefakts

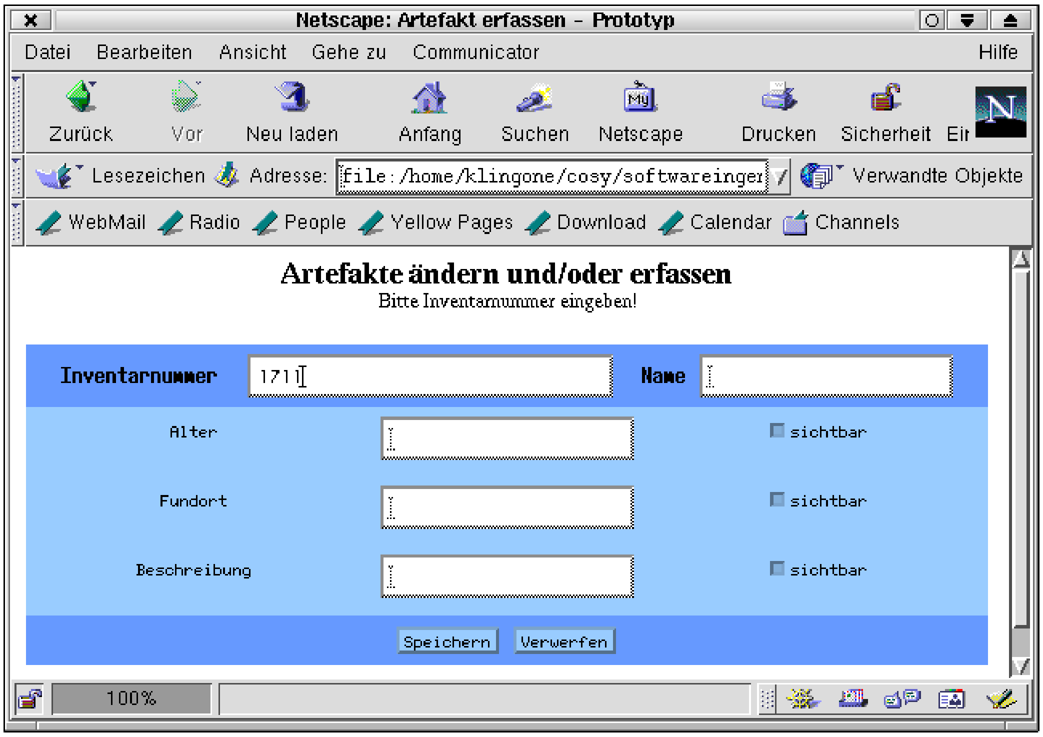
- Wenn bereits eine Artefakt existiert, wird als nächstes die Seite zur Änderung der Attribute aufgerufen, wobei in den Attributsfeldern die Daten des Artefakts mit der eingegebenen Inventarnummer stehen.



Es können nun die Werte des Artefakts geändert werden.
Nach Abschluß der Arbeit kann man durch Betätigen von *Speichern* die neuen Daten des Artefakts in die Datenbank schreiben.
Wird *Verwerfen* gewählt, so werden die Daten nicht abgespeichert und man gelangt wieder zur Startseite (siehe Abb. 17.1) und kann ein neues Artefakt bearbeiten.

Figure 17.3: Neues Artefakt wurde angelegt

- Ist in der Datenbank noch kein Artefakt mit der eingegebenen Inventarnummer vorhanden, so erscheint ein leeres Fromular zum Ändern der Attribute.
  Es kann nun in alle Felder etwas eingegeben werden. Auch das Feld zur Namenseingabe ist jetzt freigegeben.
  Die Eingabe des Namens ist optional, die Eingabe der anderen Attribute ist optional.



Wird nach der Eingabe der Attribute *Speichern* ausgewählt, so wird in der Datenbank ein neues Artefakt mit den entsprechenden Daten angelegt.
Bei *Verwerfen* werden die Daten nicht weiterverarbeitet und der Benutzer gelangt zur Startseite (siehe Abb. 17.1) und kann ein neues Artefakt bearbeiten.

17.2 Funktionsweise

Beim Öffnen der Startseite versucht das Applet, über JDBC eine Verbindung zur Datenbank herzustellen.
Nachdem die Inventarnummer eingegeben wurde, wird in der Datenbank gesucht, ob schon ein Objekt mit dieser Inventarnummer existiert. Wenn dies der Fall ist, wird ein Objekt mit den Daten des Artefakts aus der Datenbank erzeugt. Mit diesem Objekt wird nun in dem Applet weitergearbeitet.
Ist noch kein Artefakt mit dieser Inventarnummer vorhanden, so wird ein Objekt ohne Attribute (bis auf die Inventarnummer) erzeugt und mit diesem weitergearbeitet.
Bei *Speichern* werden in beiden Fällen die Daten des Objekts, mit welchem in dem Applet gearbeitet wurde, in der Datenbank gespeichert.
Bei *Verwerfen* wird in beiden Fällen das Objekt verworfen.

17.2.1 Datenstruktur

In der Datenbank existiert eine Entität ARTIFACTS, welche die Inventarnummer (ID) und den Namen des Artefakt (NAME) enthält.
Ebenso existiert eine Entität ARTATTRIBUTES, welche einem optionalen Attribut entspricht. Es wird durch seinen Namen (NAME), seinem Wert (VALUE), seiner Sichtbarkeit (VISIBLE) und natürlich dem Verweis, zu welchem Attribut es gehört, (ART_ID) bestimmt.
Durch die Methode, alle Attribute eines Artefakts, aus der der Inventarnummer und dem Namen, auszulagern, hat man die Möglichkeit, jedem Artefakt genau jene Anzahl und Art der Attribute zu geben, welche es am besten beschreiben.

17.2.2 Performance

Um eine möglichst schnelle Bearbeitung und Verwendung der Daten zu erreichen, haben wir uns entschlossen, nicht bei jeder Datenänderung auf die Datenbank zuzugreifen, sondern die Daten in Objekten zwischenzuhalten. Dies hat den Preis, dass zusätzlich geregelt werden muss, dass nicht 2 Personen gleichzeitig an einem Artefakt arbeiten und somit die Konsistenz der Datenbank gefährdet wird.
Wir haben uns für die Verwendung von Applets entschieden, da dies die schnellste Variante ist. Nur beim Holen der Daten und beim Abspeichern der Daten muss auf den Server und somit die Datenbank zugegriffen werden. Alle anderen arbeiten werden vom Applet direkt beim Benutzer erledigt. Es muss jedoch auch darauf geachtet werden, dass der Benutzer ohne grössere

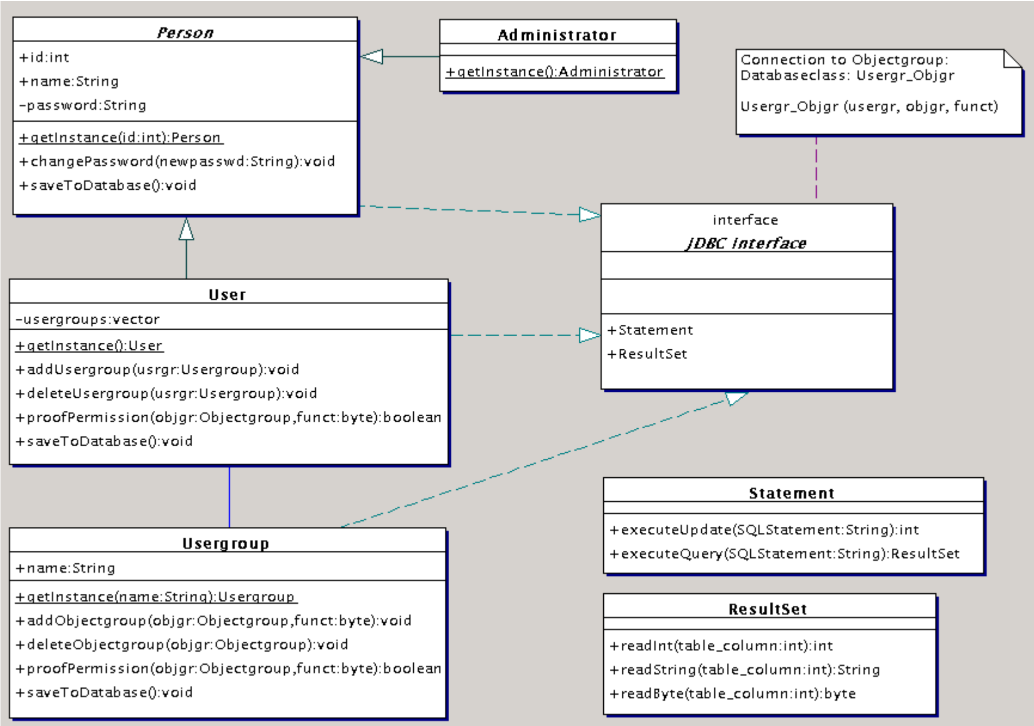
Probleme und ohne langes Herunterladen von benötigten Libaries arbeiten kann.

Kapitel 18

Construction Phase

Wir haben uns in dieser Phase mit der Iteration 5 beschäftigt.

18.1 Klassendiagramm

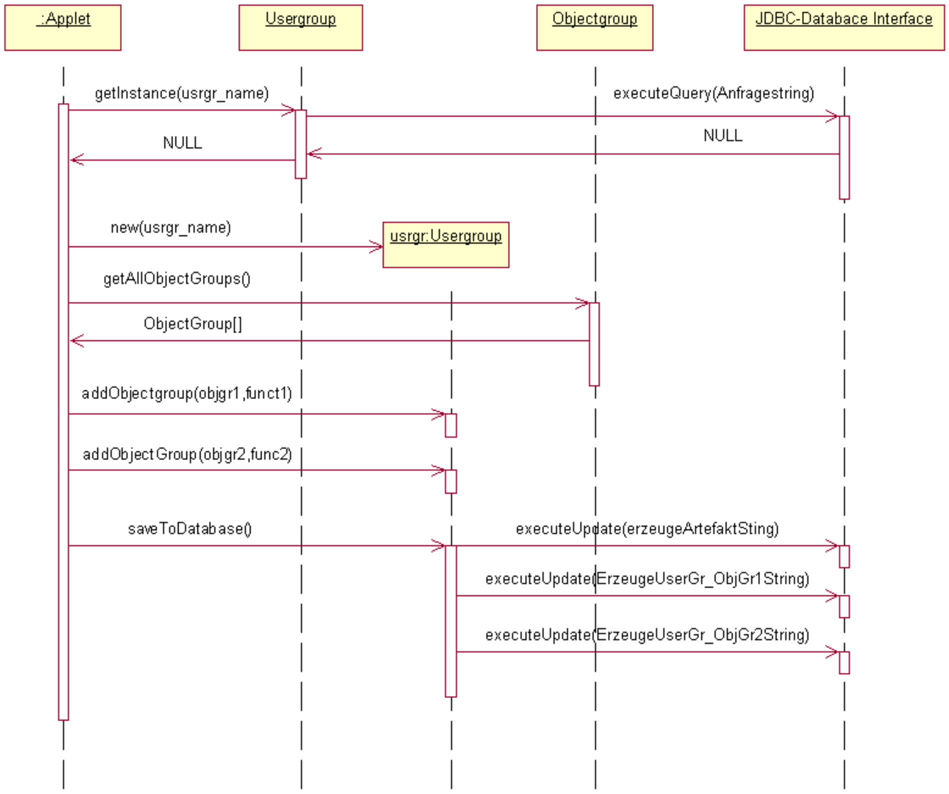


18.2 Interaktionsdiagramme

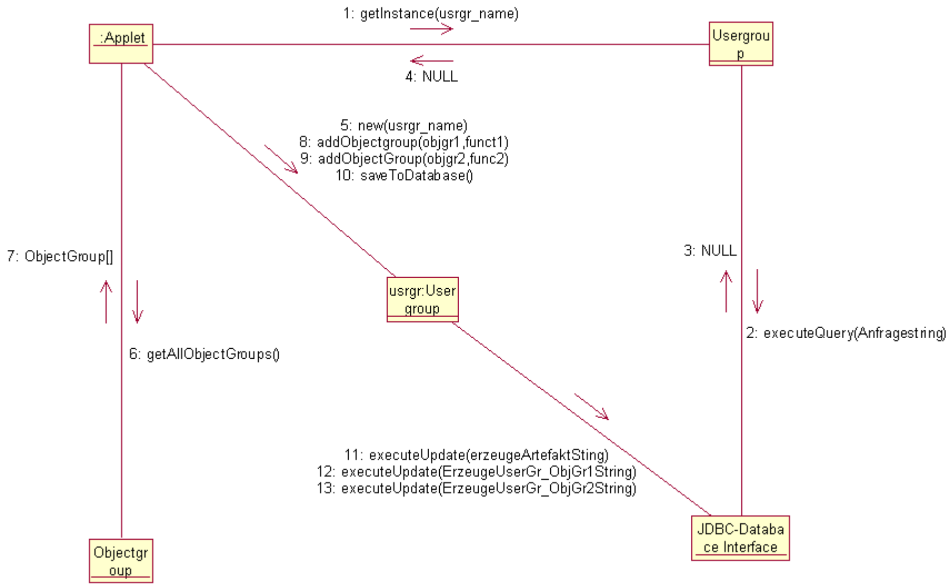
18.2.1 Benutzergruppe erstellen

Es wird der Fall durchgespielt, dass eine Benuztergruppe neu angelegt wird. Der neue Bezeichner für die Benutzergruppe existiert noch nicht. Dadurch wird beim Aufruf von getInstance NULL zurückgegeben. Mit new wird anschließend ein neues Objekt von UserGroup erzeugt. Im nächsten Schritt werden mit der Klassenmethode getAllObjectGroups alle momentan vorhandenen ObjectGroups abgefragt. Somit können alle möglichen ObjectGroups im Applet zur Auswahl zur Verfügung gestellt werden. Nun kann der Benutzer sich beliebige ObjectGroups aussuchen und zu jeder ausgewählten einen Satz von Funktionalitäten bestimmen. Für einen Satz von Funktionalitäten kann er aus allen fix vorgegebenen Funktionalitäten, all jene die man wünscht auswählen oder nicht auswählen. Der gesamte Satz enspricht einer Variablen (Byte). An die neue UserGroup werden nun alle ausgewählten ObjectGroups samt ihrem dazugehörigen Satz an Funktionalitäten weitergegeben. In dem durchgespieltem Fall werden 2 ObjectGroups ausgewählt. Am Schluß werden die Daten des neuen UserGroup Objekts in der Datenbank abgespeichert.

18.2.1.1 Sequenz-Diagramm

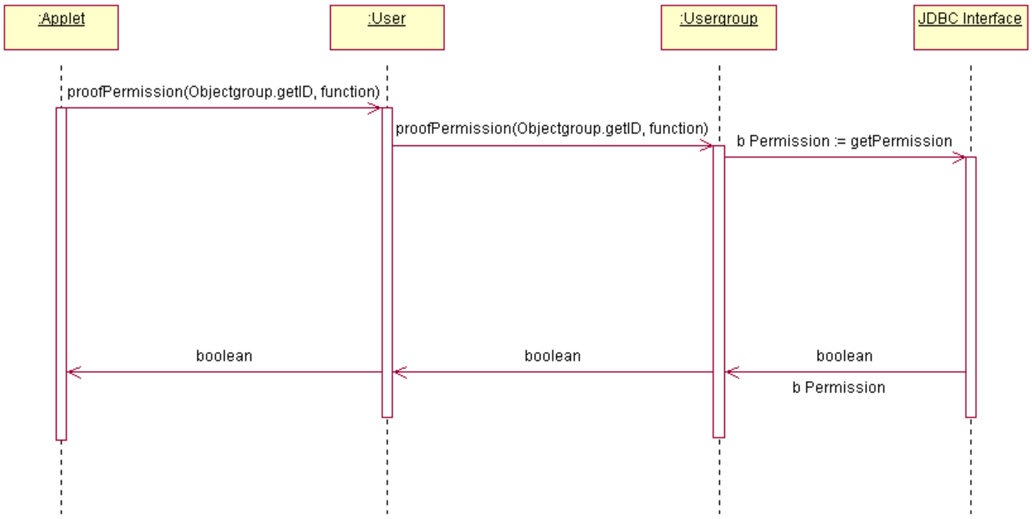


18.2.1.2 Kollaborations-Diagramm

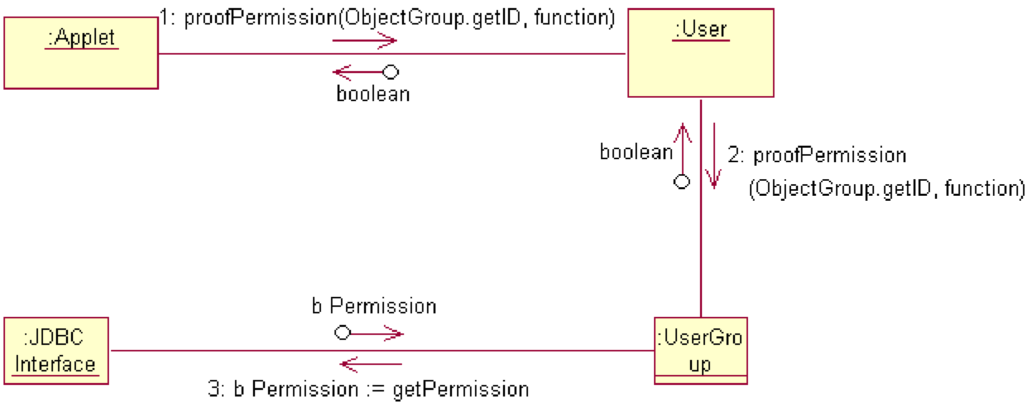


18.2.2 Rechte prüfen

18.2.2.1 Sequenz-Diagramm



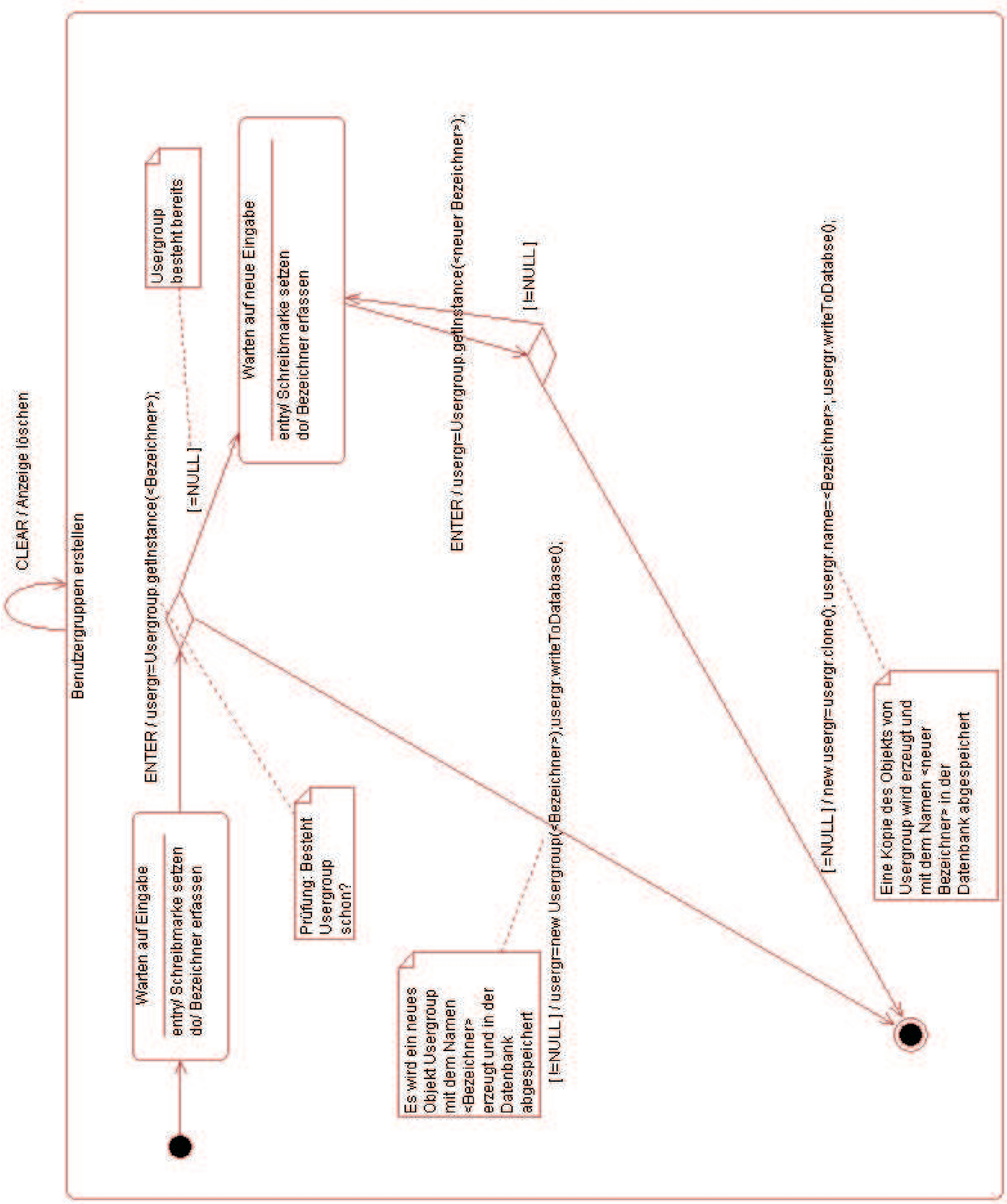
18.2.2.2 Kollaborations-Diagramm



18.3 Zustandsdiagramm

State Diagram f"ur die Klasse User mit dem Zustand Benutzergruppen erstellen

Abbildung 18.1: Zustand: Benutzergruppen erstellen



Es gibt zwei Zustände. Beim Zustand *Warten auf Eingabe* werden das Eingabefeld aktiv geschalten und die Tastatureingaben konsumiert.

Das Eingabefeld wird aus der Sicht des Benutzers durch Drücken der Eingabetaste (=ENTER) verlassen.
Dann erfolgt eine Prüfung, ob eine Usergroup mit dem Namen „Bezeichner“ schon in der Datenbank existiert. Die Methode getInstance() liefert entweder NULL, wenn noch keine Usergroup mit diesem Namen existiert oder sie liefert eine Referenz auf das Objekt Usergroup zurück.
Falls noch keine Usergroup mit diesem Namen vorhanden ist, wird eine neue Usergroup mit dem Namen „Bezeichner“ angelegt und dann mit der Methode *writeToDatabase()* in der Datenbank abgespeichert.
Falls schon eine Usergroup mit diesem Namen existiert, so muss ein „neuer Bezeichner“ eingegeben werden.
Mit der Eingabetaste wird dann wieder das Eingabefeld verlassen und es wird wieder geprüft, ob schon eine Usergroup diesen Namen hat.
Wenn nein, so wird die bestehende Usergroup „Bezeichner“ kopiert und unter dem Namen „neuer Bezeichner“ in der Datenbank abgespeichert.
Falls eine Usergroup schon den Namen „neuer Bezeichner“ hat, so muss nochmals ein „neuer Bezeichner“ eingegeben werden, solange, bis es keine Usergroup unter diesem Namen gibt; wenn dies der Fall ist, wird wie oben beschrieben fortgefahren.
Mit der Taste CLEAR soll der Benutzer die Möglichkeit haben, das Eingabefeld wieder zu löschen.

Kapitel 19

Glossar

Verwalter: Gruppe der Verwalter, siehe 1.3.2.2

Internetuser: Gruppe der Internetuser, siehe 1.3.2.1

Mitarbeiter: Alle Personen, die für den Auftraggeber arbeiten. Dies umfasst den Verwalter und alle Personen, welche einer Benutzergruppe zugeordnet sind, siehe 1.3.2.3

archäologisches Objekt: Artefakte, Siedlungen, Gräber, etc.

Objekt: archäologisches Objekt

Datensatz: alle Daten, die zu einem Objekt gehören

Funktionalitäten: Die Funktionen bzw. die Berechtigung zu Funktionen, welche Mitarbeiter ausführen können.

virtuelles Museum: siehe 1.4.2.1

Stockwerk: siehe 1.4.2.1

Raum: siehe 1.4.2.1

Vitrine: siehe 1.4.2.1

CD-ROM: Abbildung eines Museum auf einer CD-ROM, siehe 1.4.1.4

Katalog: Katalog zu bestimmten archäologischen Daten, siehe 1.4.1.5

sichtbar: Dem Internetuser werden Daten die als sichtbar gekennzeichnet und sind im Museum und bei der Datensuche angezeigt.

nicht sichtbar: Dem Internetuser werden Daten, die als nicht sichtbar gekennzeichnet sind, weder im Museum, noch bei der Datensuche angezeigt.

Suchobjekt: Suchbegriff, kann boolsch (UND, ODER, NICHT) mit anderen Suchbegriffen verknüpft werden

Kapitel 20

Protokolle

20.1 18. Oktober 2000, 15.15-15.45 Uhr

1. Protokollführer: Florian Angulanza
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Florian Angulanza
3. Besprochene Themen: Projektbeschreibung: Diskussion offener Fragen

Aus der Diskussion gewonnene Informationen:

- Der Kunde hat bereits eine Homepage, auf die aufgebaut werden kann.
- Der Kunde soll die Möglichkeit haben, als Verwalter Rechte für die Datenhaltung zu vergeben. Dabei soll ihm von uns (der SE AG) ein entsprechendes Tool bereitgestellt werden, das die Administration der Rechte erlaubt.
- Die Rechte werden auf Benutzergruppen vergeben. Eine Benutzergruppe stellt z. B. die Gruppe der Erfasser dar. Die Rechte einer Benutzergruppe beziehen sich auf Objektgruppen, d. h. es kann z. B. das Recht vom Verwalter erteilt werden, dass die Gruppe der Erfasser neue Artefakte einpflegen darf, die einem bestimmten Grab zugeordnet sind.
- Die Gruppe der wissenschaftlichen Erfasser hat auf der einen Seite alle Rechte der Erfasser, hat aber darüber hinaus noch zusätzliche Berechtigungen, wie etwa das anlegen von Literaturverweisen oder Verweisen auf Publikationen. Die Rechte des wissenschaftlichen Erfassers umfassen also die Rechte des “normalen” Erfassers.
  Es gibt eine Skala, auf der die Berechtigungen für die jeweilige Benutzergruppe festgelegt werden können.

Ein Erfasser kann die Möglichkeit haben, Daten einzugeben, ein anderer darf diese zusätzlich im virtuellen Museum zuordnen

- Ergänzungen zum virtuellen Museum:
  - Es ist die Abbildung eines Hauses im Netz, das aus Stockwerken besteht (natürlich keine 3D-Welt, sondern einfach)
  - Der 2. Stock besteht z. B. aus lauter Fibeln
  - Ein Raum könnte eine bestimmte Epoche repräsentieren
  - In einem Raum kann auch nur eine Vitrine stehen
  - In der Vitrine werden die archäologischen Objekte durch Texte und multimediale Daten beschrieben:
    Es werden keine auditiven Daten präsentiert, sondern nur visuelle, also Bilder, eventuell Videos
- es gibt eventuell mehrere Museen, eines ist aber Pflicht
- Die Daten sollen ab dem Zeitpunkt der Zuordnung im virtuellen Museum im Internet verfügbar sein.
- Die Pflege der Daten erfolgt über eine andere Internetseite, also nicht die Seite, die für den Besucher vorgesehen ist.
- Suche für den Internetbesucher:
  - gibt der Besucher z. B. ein Schwert ein, so wird ihm alles aufgelistet, wo dieser Begriff vorkommt. Dabei wird allerdings nur die Ebene 1 durchsucht (also feste Attribute, keine Verweise auf Literatur usw.).
  - Die Suchergebnisse werden als Liste dargestellt bzw. im virtuellen Museum markiert.
  - Es soll eine erweiterte Suche geben, die es erlaubt, z. B. nur Beschreibungen zu durchsuchen oder nur nach Speeren zu suchen
- Die CD-ROM ist eine Abbildung des virtuellen Museums, es sind ausgewählte Objekte darauf enthalten. Die CD-ROMs sollen vom Kunden erstellt werden können.
- Der Erfasser, der die Objekte bearbeitet, legt fest, welche Daten sichtbar sind.

- Zu den "Grundattributen" eines Objektes zählen: Inventarnummer, Fundort, Dimensionen, Material, Freitext
- Das Text- bzw. Grafikformat für den Katalog ist das pdf-Format
- Das nächste Treffen findet in ca. 2 Wochen statt

20.2 31. Oktober 2000, 16.45 - 17.5 Uhr

1. Protokollführer: Röck Andrea
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Röck Andrea
3. Besprochene Themen: Anforderungsdefinition, Diskussion offener Fragen

Aus der Diskussion gewonnene Informationen:

- **Auslastung:** Dem Aussagen des Kunden nach muss man mit dem Eintragen von 10 Artefakten pro Tag rechnen und mit etwa 100 gleichzeitigen Zugriffen von Internetuserns auf das Museum bzw. die Datensuche.
- **Kosten:** Die Verwendung des Programms kostet den Internetuser nichts. Es wird über die Imageaufbesserung des Auftraggebers und die Werbung für den Museumsshop des Kunden rentabel.
- **Gestaltung des Museums:** Für die Gestaltung des Museums soll eine Art Wizard existieren. Für diesen Wizard gibt es verschiedene Templates, welche jeweils einer bestimmten Gestaltungsvariante entsprechen. Es können jederzeit neue Templates hinzugefügt werden und alte gelöscht werden. Die Gestaltung und die Erstellung der Stockwerke, Räume, Vitrinen und die Darstellung des Inhalts der Vitrinen soll über den Wizard und die Template geschehen.
- **Abnahmekriterien:** Der Kunde muss sich die Anforderungsdefinition erst genauer durchlesen, bevor er ein Kommentar zu den Abnahmekriterien abgibt.
- **Inventarnummer:** Die Inventarnummern müssen explizit eingegeben werden, da schon Objekten auf Papier Inventarnummern zugewiesen wurden. Die können also nicht automatisch vergeben werden.
- **Datenformate:** verwendete Datenformate:

  - Bildformate: j-peg und gif
  - Textformate (Katalog, Puplikationen, etc.): pdf
  - Vidoeformate: real audio

- **Objektgruppen:** Es können vom Verwalter Objektgruppen erzeugt werden (z. B. eine Epoche, ein Entdecker, ein Fondort, ein Grab oder ähnliches). Der Erfasser kann dann eine Gruppe auswählen, für die er die Berechtigung besitzt und ein neues Objekt zu dieser Gruppe erstellen.

- **Verteilung der Rechte:** Es gibt keine vordefinierten Gruppen (bis auf den Verwalter der alle Rechte insbesondere administrative hat). Es gibt Rechte bzw. Funktionalitäten (z. B. Daten erfassen, Museum aufbauen, CD-ROM erstellen, etc.). Die Unterteilung soll eher fein sein. Es muss die Möglichkeit bestehen, soche Rechte zu verändern und neue hinzuzufügen.
  Der Verwalter kann nun eine Gruppe erzeugen und für diese Gruppe Rechte und Objektgruppen auswählen. (z. B. wieder eine Art Wizard mit Rechten zum anklicken) Er aknn dann Personen diesen Gruppen zuweisen. Somit ist die Gestaltung der Rechte und er Gruppen sehr dynamisch und flexibel.

- **Sprache:** Das Programm soll multilingual einsetzbar sein. Es soll die Sprachen Deutsch, Englisch, Fanzosisch, Italienisch und zwingend Chinesisch unterstützen.

- **Sicherheit:** Die Sicherheit der Datenübergabe und die Identifikation der Mitarbeiter soll nur über Username und Passwort und http gewährleistet werden.

20.3 8. November 2000, 17.00-17.30 Uhr

1. Protokollführer: Hildegard Zehentner
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Hildegard Zehentner
3. Besprochene Themen: Diskussion offener Fragen, Durchsicht der Actors und der Use Case Diagramme

Aus der Diskussion gewonnene Informationen:

- **UML**
  Bei uns wird die UML 1.1 verwendet, erst später, wenn es ausdrücklich erwähnt wird, verwenden wir UML 1.3.

- **Tool für Use case Diagramme**
  Für die Use Case Diagramme wird Rational Rose verwendet, es steht uns aber auch frei ein anderes Tool (z. B. ARGO-UML oder Together) zu verwenden.

- **Browser**
  Wir können nicht darauf vertrauen, dass die Browser beim ersten Einsatz unseres Systems schon alle XML unterstützen. Deshalb muss für die notwendige Konvertierung von XML auf HTML gesorgt werden.

- **Gruppen- und Rechteverteilung**
  Da die Gruppen- und Rechteverteilung beim letzten Treffen von Röck Andrea und Mag. Dipl.-Ing. Roland Schwaiger überarbeitet wurden, holte ich mir letzte Bestätigung, dass das neue System nun das gewünschte ist.

- **Daten**
  Die Daten sollen multilingual abgespeichert werden, d. h. der Einsatz eines Dolmetscher wird unvermeidlich.

- **CD-ROM**
  Auf der CD-ROM soll eine 1:1 Abbildung des Virtuellen Museums erzeugt werden. Es muss nicht möglich sein, nur gewünschte Objekte auszuwählen und für diese ein Virtuelles Museum zu erzeugen, welches nie im Web exisitiert.

- **Actors**
  Es soll besser der Begriff Systemadministrator anstelle von SE AG verwendet werden.

- **Use case Diagramme**

  - Genauere Unterteilung
  - Vorbedingungen (guards) angeben, falls nötig
  - usw.

Die genauen Änderungen werden im Kapitel 12 bei den Use Case Diagrammen selbst geändert.

- **Abnahmekriterien**
  Der Kunde hat sich schon Gedanken gemacht über die Abnahmekriterien:
  Zentrale Anforderungen für die Phase 1 sind die funktionalen und nicht funktionalen Anforderungen mit Priorität 1 - 2 und die Musskriterien. Auf jeden Fall müssen die Berechtigungsgruppen erstellt werden können und das Virtuelle Museum soll aufgebaut werden können. Keine Muss für die Phase 1 sind die Erstellung der CD-ROM und des Katalogs.

20.4 17. November 2000, 10:30-10.45 Uhr

1. Protokollführer: Manfred Mayr
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Manfred Mayr
3. Besprochene Themen: Diskussion und Durchsicht der Kosten - und Aufwandsabschätzung, Beantwortung noch offener Fragen

Aus der Diskussion gewonnene Informationen:

- CD-ROM-Erstellung
  Es wird angenommen, dass die 1:1 Abbildung des Museums auf die CD immer Platz hat.
- Portable
  Es wird festgelegt, dass eine transparente Schicht über die Datenbankebene gelegt wird, damit wird erreicht, dass die Datenbank ausgewechselt werden kann.
- Stundenlohn
  Die Empfehlung der Wirtschaftskammer: 1200 - 1400 Schilling/Stunde
- Die Schwierigkeit einer Schätzung. In diesem Stadium ist eine Kosten - und Aufwandsabschätzung sehr schwierig, aber trotzdem sehr wichtig. Man sieht es schon daran, wie unterschiedlich die Schätzungen in anderen Gruppen ausgefallen sind.
- Transaktionen:
  SAP fasst eine Transaktion ein wenig anders auf. z. B. ist bei SAP eine Datenbanktransaktion, der Übergang der Datenbank von einem konsistenten Zustand zum nächsten konstistenten Zustand.

20.5 06. Dezember 2000, 16:30 - 17:00 Uhr

1. Protokollführer: Christian Osterrieder
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Christian Osterrieder
3. Besprochene Themen: Diskussion und Durchsicht der Activity-Diagramme und User-Interfaces

Aus der Diskussion gewonnene Informationen:

- Eine genauere "Zerteilung" der Use-Cases braucht in dieser Phase des Projekts noch nicht vorgenommen werden. Das durch weitere Bearbeitung der Use-Cases weitere entstehen oder umgeändert werden müssen ist eine normaler Vorgang.
- ObjekteSuchen: Suche sollte auch nach bestimmten Attributen möglich sein. Beispielsweise sollte man nur nach Ausgrabungsorten suchen können.
- Activity-Diagramme: Die Darstellung der Synchronisation in den Diagrammen sollte nochmals auf Richtigkeit überprüft werden.
- MuseumBesuchen: Die Navigation im Activity-Diagramm stellt keine Rückschritte dar. Es sollte möglich sein jeden Schritt sofort wieder rückgängig zu machen (ansonsten müsste das gesamte Museum durchwandert werden, um wieder zum Ausgangspunkt zu gelangen).
- Berechtigungen: Die derzeite Vergabe von Berechtigungen ist sehr komplex. Man sollte darüber diskutieren die Berechtigungsvergabe mehr auf den einzelnen Benutzer zu beziehen.
- Vitrine erstellen: Wünschenswert wäre eine Generierung der Suchbegriffe aus dem Beschreibungstext. Eventuell auch eine Normierung der Suchbegriffe (Vorgaben)

20.6 20. Dezember 2000, 14:30 - 15:00 Uhr

1. Protokollführer: Florian Angulanza
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Florian Angulanza
3. Besprochene Themen: Diskussion des geplanten Architektur und der Iterationsphasen für den Projektplan

Aus der Diskussion gewonnene Informationen:

- Kommunikation zwischen Business-Layer und Database zu umständlich: Die Daten müssen nicht als XML-Objekte der Datenbank übergeben werden, ebenso müssen die Daten nicht permanent als XML-Objekte in der Datenbank vorliegen.
  Es reicht, wenn das Datenbankinterface Routinen zur Verfügung stellt, die XML-Objekte aus den gespeicherten Daten liefern können (etwa für die Katalogerstellung).

- Wir können eine relationale Datenbank verwenden, mit der wir über JDBC-Interfaces kommunizieren.

- Überlegung: 2 Alternativen

  - Verwendung von Servlets im Business Layer; Vorteile:
    Instanzen liegen nur einfach vor.
    Performance liegt in unserer Hand.
  - Statt Servlets im Business-Layer durchgehend Applets verwenden. Dann muss das Datenbank-Interface selbst dafür sorgen, dass die Daten konsistent bleiben. Es muss gegenüber dem Business-Layer Sperrungen vornehmen, damit nicht mehrere Instanzen vom eigentlich selben Objekt entstehen.
    Die Performance wird bei Verwendung von Applets vorwiegend vom User bestimmt.

- Iterationslänge von vier Wochen im Projektplan ist wahrscheinlich zu lang.
  Nach jeder Iteration soll ein Prototyp vorliegen, mit dem der Kunde arbeiten kann. Er soll diesen Prototypen testen können. Dadurch gewinnen wir wiederum Erkenntnisse und Vorschläge für die nächste Iteration. Eine Iterationslänge von etwa 2 Wochen wäre ideal.

- Vorgehensweise bei Erstelltung der Iterationen:

  1. Nicht von den vorhandenen Use-Cases ausgehen und diese in Iterationen einteilen, sondern mit diesen "groben" Use-Cases die Architektur durchwandern, um so auf Teile von Use-Cases zu stossen, sozusagen verfeinerte Use-Cases, die in einer Iteration behandelt werden können.
  2. Diese verfeinerten Use-Cases so zu Iterationen gruppieren, dass in jeder Iteration Teilfunktionalitäten enthalten sind.

20.7 11. Jänner 2001, 17:00 - 17:30 Uhr

1. Protokollführer: Röck Andrea
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Röck Andrea

Aus der Diskussion gewonnene Informationen:

- Der Prototyp war fertiggestellt, aber da es nicht möglich war auf der Uni einen funktionierenden MySQL Server zu installieren konnte er nicht vorgeführt werden.
  Abgabevorgaben:
  - Abgabe des Codes
  - Abgabe eines erweiterten Dokuments samt Screenshots vom Prototypen in Aktion.
- Problem beim Applet des Prototypens:
  Man benötigt den entsprechenden Treiber für die Verbindung zwischen der Datenbank und JDBC um das Applet laufen zu lassen. Für den allgemeinen Gebrauch bedeutet dies, dass der Anwender des Programms diesen Treiber braucht um das Programm zu verwenden.

20.8 25. Jänner 2001, 16:30 - 17:00 Uhr

1. Protokollführer: Hildegard Zehentner
2. Teilnehmer: Mag. Dipl.-Ing. Roland Schwaiger, Hildegard Zehentner
3. Besprochene Themen: Durchsicht der letzten Aufgabe (Klassen-, Interaktions- und Zustandsdiagramme)

Aus der Diskussion gewonnene Informationen:

- Bei der Durchsicht der verschiedenen Diagramme wurden einige Änderungen vorgenommen. Dies wurden von uns geändert und in dieses Dokument neu eingearbeitet.
- Freitag, 2. Februar 2001 um 12.00 Uhr ist der absolut letzte Abgabetermin der im Netz geforderten Unterlagen. Aus Zeitmagel wird auf die ursprüngliche Art der Präsentation verzichtet. Es soll nun eine Internetpräsentation stattfinden. Wir haben maximal 6 Html-Seiten zur Verfügung. Layout ist frei.

Literaturverzeichnis

[1] Gustav Pomberger, Günther Blaschek: *Software Engineering: Prototyping und objektorientierte Software-Entwicklung, Hanser Verlag, 1993*

[2] Geri Schneider, Jason P. Winters: *Applying Use Cases: A Practical Guide : A Practical Guide, Addison-Wesley, 1998*

[3] Siegfried Vollmann: *Aufwandsschätzung im Software Engineering: Neue Verfahren und Arbeitshilfen, IWT Verlag, 1990*

[4] Edward Yourdon: *Modern Structured Analysis, Yourdon Press, 1989*

[5] Darly Kulak, Eamonn Guiney: *Use Cases: Requirements in Context, Addison-Wesley, 2000*

[6] Martin Hitz, Gerti Kappel: *UML@WORK: Von der Analyse zur Realisierung, dpunkt.verlag, 1999*

[7] Geri Schneider, Jason P. Winters: *Applying Use Cases: A Practical Guide : A Practical Guide, Addison-Wesley, 1998*