Entwurf

Id: Entwurf.tex,v 1.19 2004/05/11 14:59:31 merches Exp

Phase	Verantwortlicher	eMail
Pflichtenheft	Mathias Pinzhoffer	pinzhoff@fmi.uni-passau.de
Entwurf	Bastian Merches	merches@fmi.uni-passau.de
Spezifikation	Christian Brunnermeier	brunnerc@fmi.uni-passau.de
Implementierung	Christoph Bachhuber-Haller	bachhube@fmi.uni-passau.de
Verifikation	Thomas Bernreiter	bernreit@fmi.uni-passau.de
Präsentation	Stephan R. Bayer	bayers@fmi.uni-passau.de

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument stellt den konzeptionellen Entwurf des webbasierten Raumverwaltungssystem RoomBA dar.

Es wird eine erste Übersicht über die verschiedenen Komponenten des MVC-Modells, das der Anwendung zugrunde liegt, gegeben. Dies geschieht durch eine Angabe der Klassenstruktur, der Benutzersichten in Form von Beschreibungen der einzelnen JSP-Seiten mit deren prinzipieller Verlinkung sowie der Zuteilung der auftretenden Funktionen zu einzelnen Servlets. Einzelne Funktionsaufrufe werden beispielhaft zu Illustrationszwecken in Sequenzdiagrammen dargestellt.

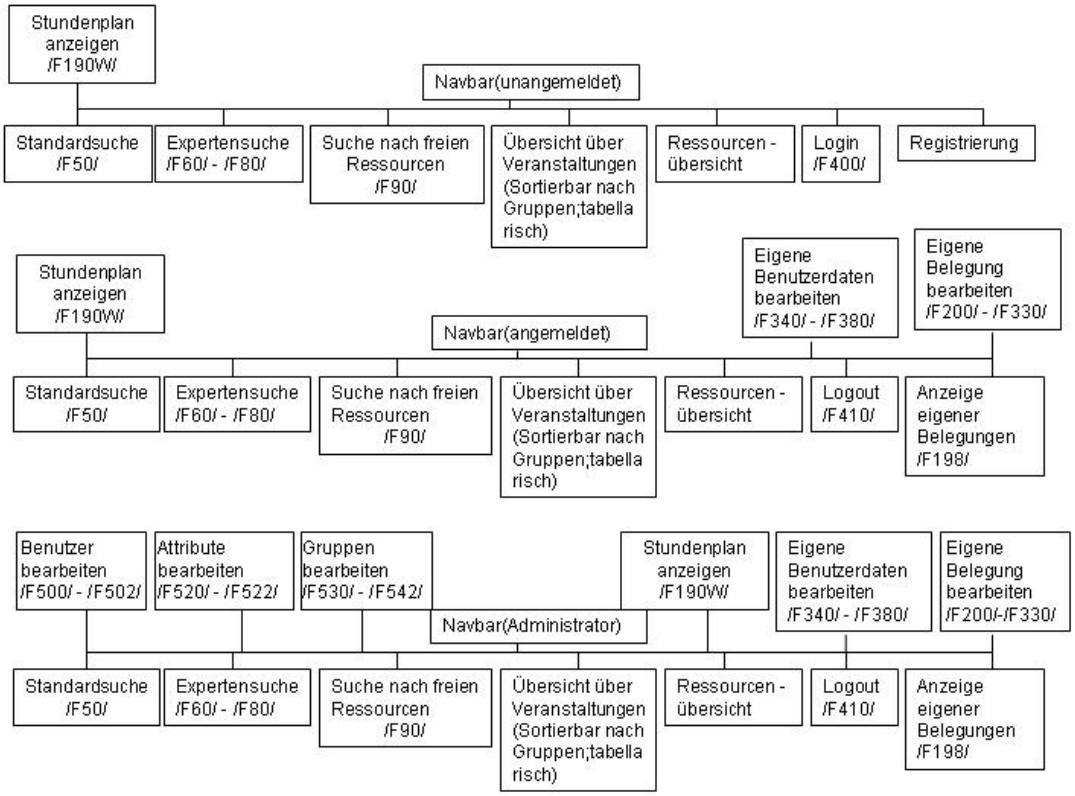
Durch das Verwenden des MVC Design Patterns und einer Konfigurationsdatei wird eine Unabhängigkeit der View-Schicht vom restlichen System erzielt, so dass die Software - wie gefordert - ohne Anpassung der Modell- und Controllerschicht auch für andere Zwecke einsetzbar ist. Auf die Verwendung von Struts wird dabei verzichtet.

Weiterhin wird hier eine erste Modellierung der zu speichernden Daten anhand eines Entity-Relationship-Diagramms gegeben. Damit wird das Datenbankschema schon weitestgehend festgelegt.

2 Analyse des Systems

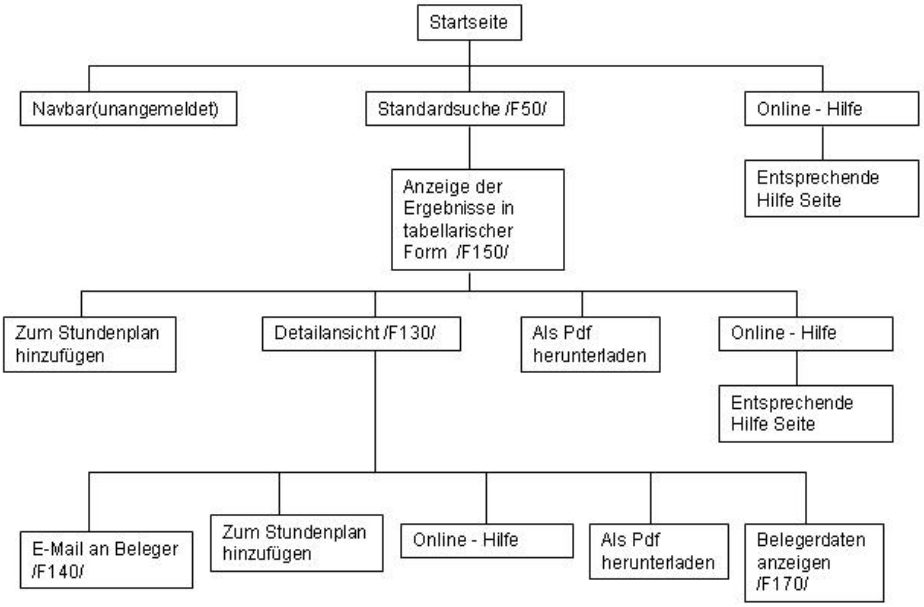
Im Folgenden wird unser Raumverwaltungssystem **RoomBA** analysiert. In den Diagrammen wird die Linkstruktur der Internetseiten dargestellt. Dabei ist zu beachten, dass es auf jeder Seite einen Link zur Online-Hilfe(/F185/) gibt, auch wenn dieser in den Diagrammen nicht immer explitzit angegeben ist.
Tabellarische Ansichten von Suchergebnissen, sowie Tages- und Wochenansicht können als PDF heruntergeladen werden.(/F180/)

2.1 Navigationsleisten



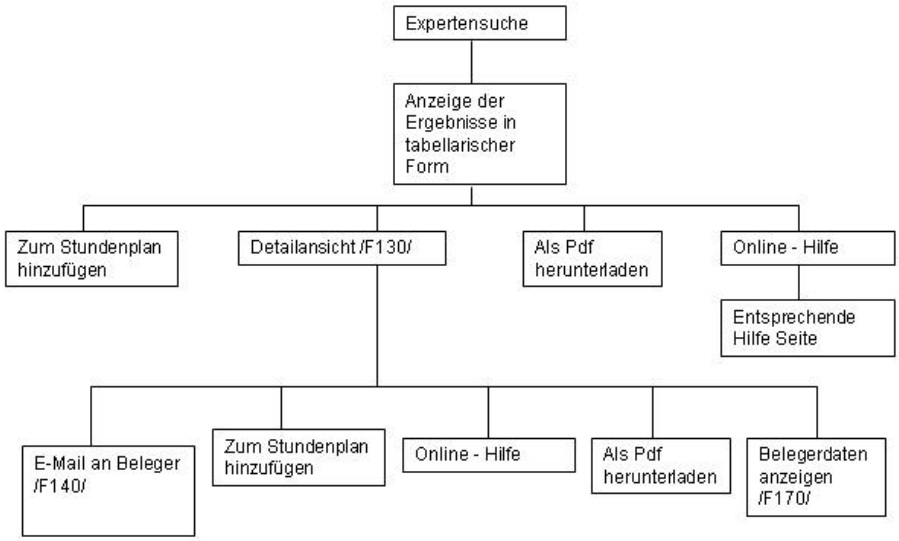
Die Navigationsleiste befindet sich immer auf der linken Seite. Darauf sind stets die aktuell möglichen Funktionen zu sehen. Wenn man am System angemeldet ist beinhaltet die Navigationsleiste zusätzliche Funktionen.

2.2 Startseite



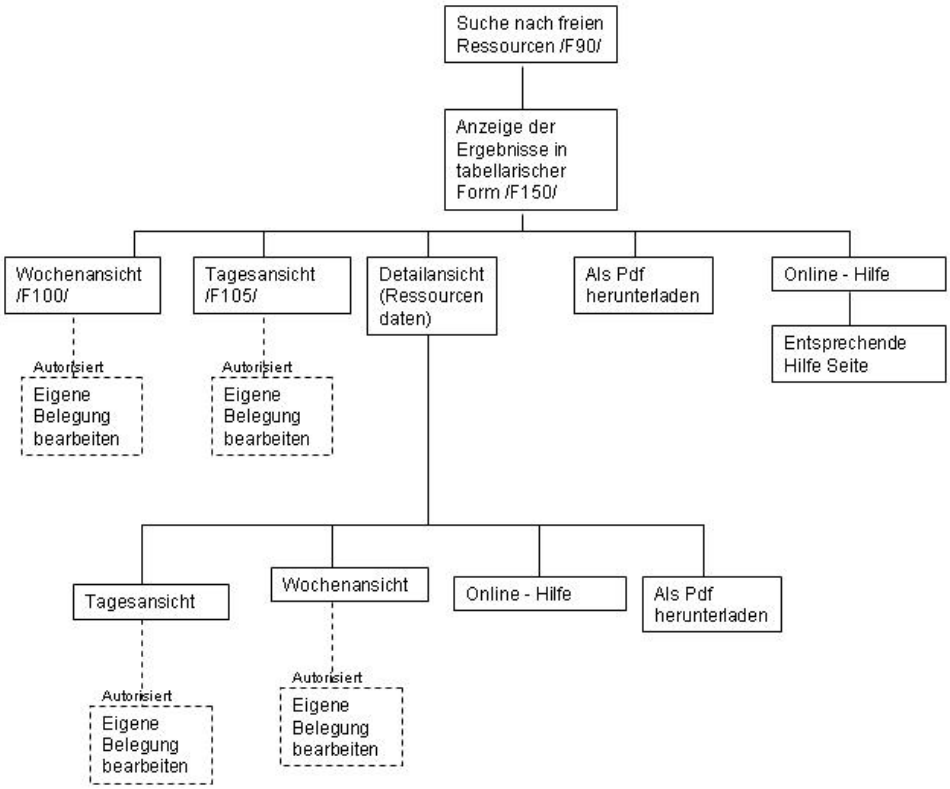
Der wichtigste Bestandteil auf der Startseite ist die Standardsuche, mit der man einfach nach Belegungen(Veranstaltungen) suchen kann.

2.3 Expertensuche



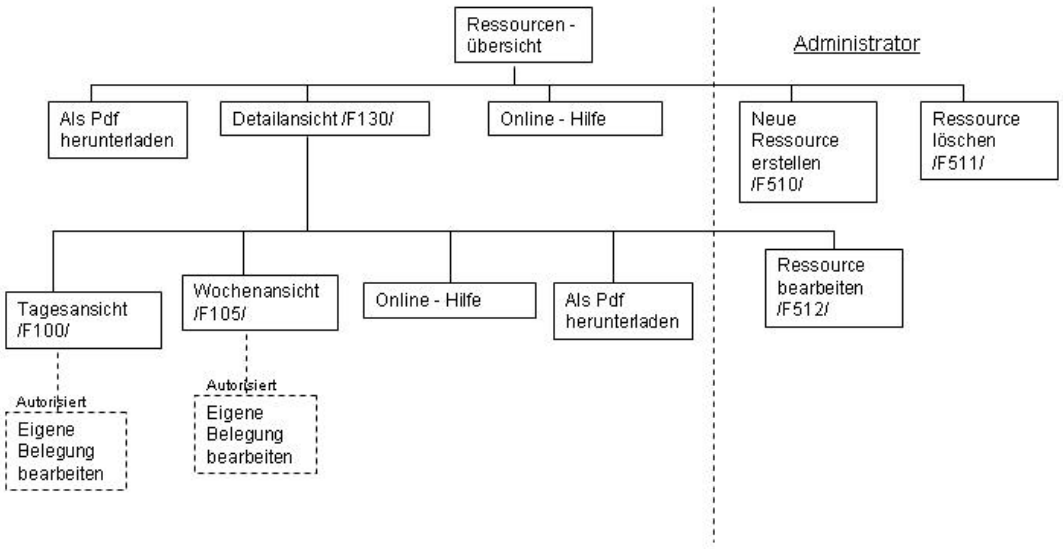
Die Expertensuche ist von der Linkstruktur genauso aufgebaut wie die Standardsuche.

2.4 Suche nach freien Ressourcen



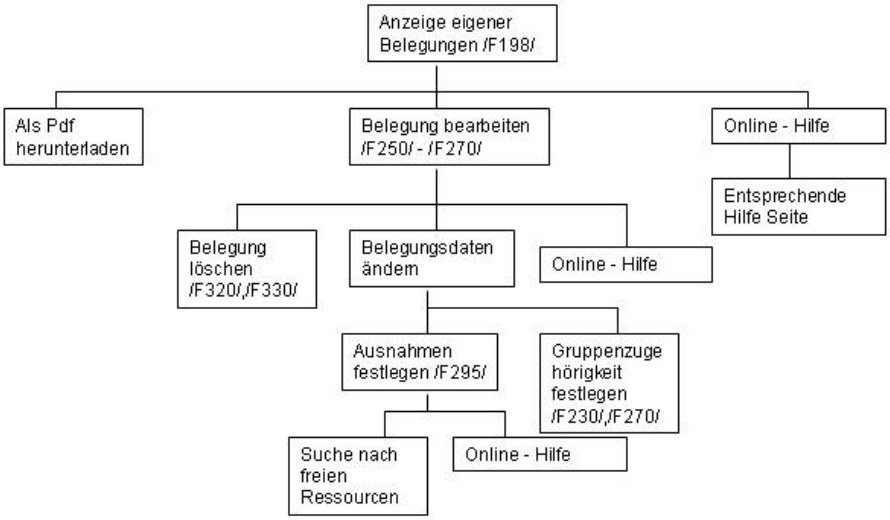
Bei der Suche nach freien Ressourcen kann so komfortabel gesucht werden wie in der Expertensuche für Veranstaltungen. Das heißt, man kann einschränken auf: Zeitraum zu dem die Ressource frei sein soll; Gruppen zu denen die Ressource gehört(z.B. Gebäude);Größe usw.
Die Ansicht erfolgt zuerst tabellarisch und ist dann einstellbar auf Tages- oder Wochenansicht. (Für Räume entspricht die Wochenansicht den Belegungsplänen, die in der Universität Passau vor den Hörsälen hängen) In dieser Ansicht kann man zur nächsten oder vorherigen Periode navigieren.
Als autorisierter Benutzer besitzt man die Möglichkeit in der Tages- oder Wochenansicht freie Zeiträume zu belegen.

2.5 Ressourcenübersicht



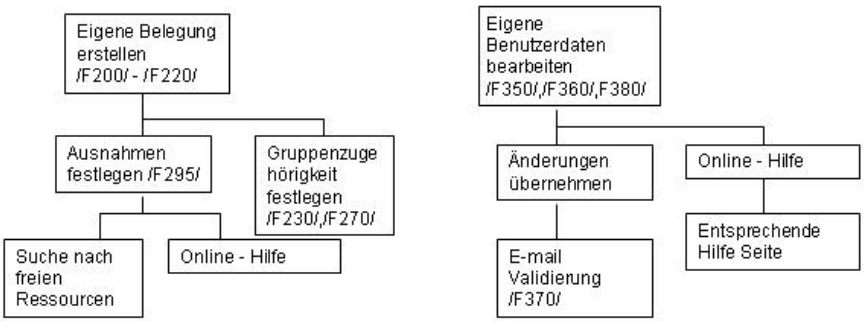
Die Ressourcenübersicht wird tabellarisch und sortiert bzw. sortierbar nach Gruppen angezeigt. Für die einzelnen Ressourcen kann sich der Benutzer eine Detailansicht anzeigen lassen, dann die Belegungen in Tages- oder Wochenansicht einsehen und hier zur nächsten oder vorherigen Periode navigieren.(/F107/)
Auch hier hat der autorisierte Benutzer das Recht, Ressourcen zu belegen. Ist der Administrator angemeldet, kann er in der Übersicht auch neue Ressourcen anlegen oder Ressourcen löschen.

2.6 Anzeige der eigenen Belegungen



Als autorisierter Benuter kann man sich seine eigenen Belegungen in tabellarischer Form anzeigen lassen. Von dort aus ist es möglich die Belegungen zu bearbeiten und zu löschen.

2.7 Eigene Benutzerdaten bearbeiten; Eigene Belegung erstellen

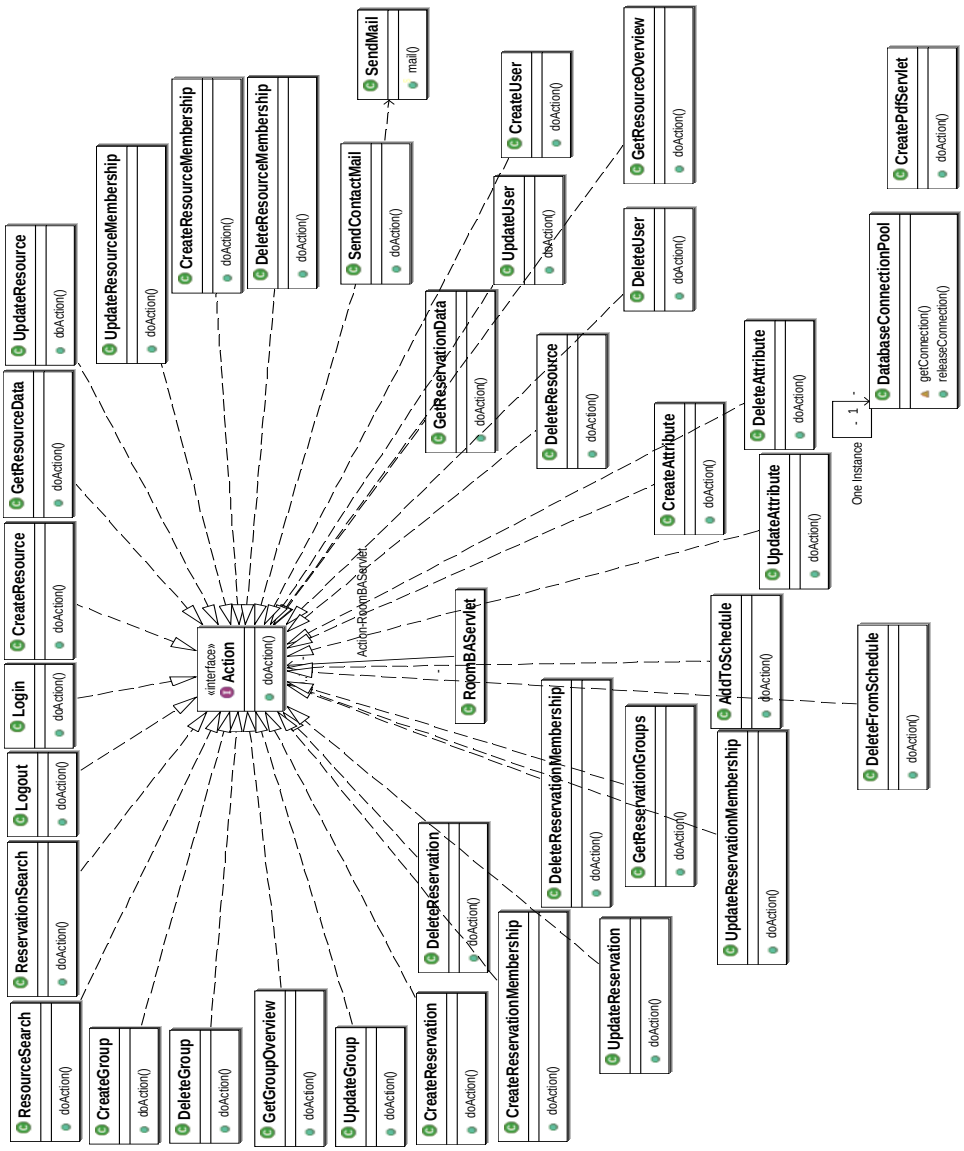


Wenn ein Benutzer seine E-Mail - Adresse ändert, muss die neue Adresse validiert werden. Hierzu wird dem Benutzer eine E-Mail mit einem Link geschickt. Klickt er auf den Link, kommt er auf die *VerifyEMail.jsp* und die neue Addresse wird gespeichert. Werden andere Daten aus /F360/ geändert, werden diese nach Bestätigung sofort übernommen.
Legt ein Benutzer eine neue Belegung mit Ausnahme an, kann er mit Hilfe der *Suche nach freien Ressourcen* eine Ressource für den abweichenden Termin belegen.

3 Klassenbeschreibung

Die Unterteilung der Beschreibungen entspricht dem MVC-Model, das unserem Entwurf zugrunde liegt.
Um das Programm so allgemein wie möglich zu halten, beziehen sich die Klassennamen auf Reservations und Resources. Angewendet auf die Universität wären Reservations Veranstaltungen und Resources Räume.

3.1 Model



Als zentrales Entwurfsmuster wird das Command Pattern verwendet. Jeder Anfragebearbeiter implementiert das Action - Interface und damit die Methode doAction(), die beim Seitenaufruf mit den Anfrageparametern ausgeführt wird.

3.1.1 Java - Klassen

Action ist die Schnittstelle zwischen dem RoomBAServlet und den Java-Klassen. Es besitzt die Funktion *doAction()*, die von jeder Klasse implementiert wird, die eine Aktion ausführt.

ResourceSearch realisiert die Suche nach freien Ressourcen unter Angabe eines Zeitraums (/F90/).

ReservationSearch erstellt die Datenbankanfrage für die Standard- und Expertensuche nach Belegungen und gibt diese an den DatabaseConnectionPool weiter.

Sie ist ein Datencontainer der alle Informationen zu einer Veranstaltung enthält (/F50/ bis /F80/).

Logout meldet einen Benutzer vom System ab (/F410/).

Login führt den Login eines Users aus, also Überprüfung der Logindaten und (/F400/) Registrierung am System.

UpdateReservation ermöglicht das Ändern einer Belegung (/F250/,/F260/,/F290/,/F295/).

CreateReservation enthält die Funktionalitäten zum Anlegen einer neuen Belegung durch einen angemeldeten Benutzer (/F200/,/F210/).

DeleteReservation erlaubt es einem angemeldetem Benutzer eine bereits angelegte Belegung wieder zu löschen. Dabei darf eine Belegung nur von dem Benutzer gelöscht werden, der sie auch angelegt hat (/F320/,/F330/).

GetReservationData liest die Daten einer Belegung aus der Datenbank aus und gibt sie zurück.

CreateReservationMembership implementiert das Hinzufügen einer Belegung zu einer Gruppe (/F230/).

DeleteReservationMembership erlaubt das Löschen der Zugehörigkeit einer Belegung zu einer Gruppe (/F270/).

UpdateReservationMembership ist für das Ändern der Gruppenzugehörigkeiten der einzelnen Belegungen zuständig (/F270/).

CreateUser erstellt aus den übergebenen Daten einen neuen Benutzer und speichert diesen in der Datenbank (/F500/).

UpdateUser aktualisiert die Daten eines Benutzers. Bei Änderung der E-Mail Adresse erfolgt eine Verifikation derselben. (/F360/,/F370/,/F380/).

DeleteUser löscht einen Benutzer aus der Datenbank (/F501/,/F550/).

CreateResource legt eine neue Ressource an (/F510/).

UpdateResource setzt die Daten einer Ressource neu (/F510/ bis /F512/,/F525/).

DeleteResource löscht eine Ressource aus der Datenbank.

GetResourceData liefert die Ressourcendaten (/D140/) zurück.

GetResourceOverview liefert eine Übersicht über die angeforderte Ressource zurück.

CreateResourceMembership dient zum Hinzufügen einer Ressource zu einer Gruppe (/F535/).

UpdateResourceMembership ist die Klasse, deren doAction Methode das Ändern der Gruppenzugehörigkeit einer Ressource zuläßt (/F535/).

DeleteResourceMembership entfernt eine Ressource aus einer Gruppe (/F535/).

CreateGroup stellt einem Administrator die Funktionen zum Anlegen einer neuen Gruppe zur Verfügung (/F530/,/F540/).

DeleteGroup ermöglicht einem Administrator das Löschen einer Gruppe.

GetGroupOverview holt alle existierenden Gruppen aus der Datenbank und gibt sie aus (/F531/,/F541/).

UpdateGroup gibt einem Administrator die Möglichkeit, eventuelle Änderungen an einer Gruppe vorzunehmen (/F532/,/F542/).

GetResourcesByGroup liefert einem Benutzer alle Ressourcen sortiert nach Gruppen.

CreateAttribute ist die Action, mit der ein Administrator neue Attribute, die Ressourcen zugeordnet werden können, anlegen kann (/F520/).

DeleteAttribute ist die Action, mit der ein Administrator existierende Attribute löschen kann. Falls das zu löschende Attribut (/F521/) Ressourcen zugeordnet ist, werden diese Zuordnungen ebenfalls gelöscht.

UpdateAttribute erlaubt einem Administrator die Umbenennung von Attributen (/F522/).

AddToSchedule ist die Action, mit der man eine Belegung zu seinem persönlichen Stundenplan hinzufügen kann. Diese werden dann in der Session in einem ScheduleBean gespeichert (/F190W/).

DeleteSchedule erlaubt das Löschen einer Belegung aus seinem Stundenplan (/F190W/).

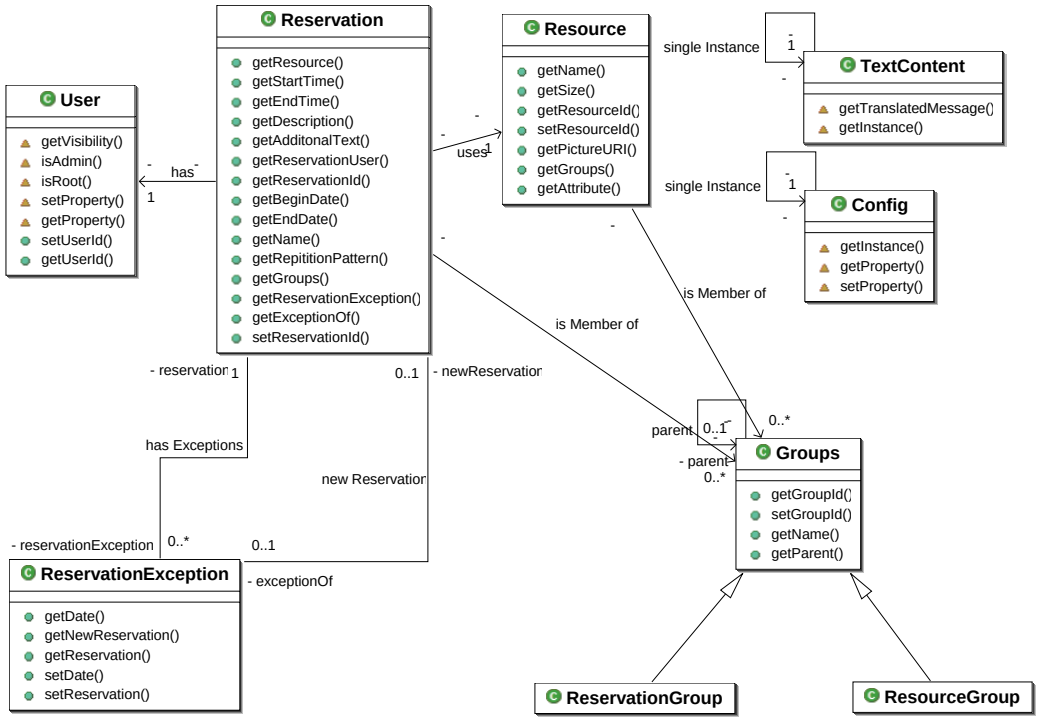
DataBaseConnectionPool verwaltet eine bestimmte Anzahl von Datenbankverbindungen, die von einem Administrator in der Konfigurationsdatei festgelegt werden kann. Falls keine Verbindungen verfügbar sind, schläft der Thread so lange, bis wieder eine frei ist oder ein Timeout erreicht wird. Zur Implementierung wird das Singleton Pattern verwendet, damit die Existenz genau eines Objektes sichergestellt ist und Zugriffe von verschiedenen Threads synchronisiert werden können.

SendMail ist eine Klasse, die nicht das Action-Interface implementiert. Sie dient zum Versenden von Emails unter Angabe von Empfänger, Betreff und Text (/F140/).

SendContactMail schickt über SendMail im Falle einer unerwünschten Belegung eine Benachrichtigung an den Belegenden.

3.1.2 Beans

Die Beans holen sich bei Bedarf die nötigen Daten aus der Datenbank. Das Ändern der Daten erfolgt in den jeweiligen „Update“ - Klassen.



User ist der Datencontainer für einen Satz persönlicher Daten.Die einzelnen Properties werden dynamisch über eine Map verwaltet. (/F550/)

Groups ist der Datencontainer für eine Auflistung von Gruppen in Baumstruktur.

ReservationGroup holt die Daten einer Belegungsgruppe aus der Datenbank.

ResourceGroup holt die Daten einer Ressourcengruppe aus der Datenbank.

Config liest die Konfigurationsdatei ein und stellt dem System die Einstellungen zur Verfügung. (/F197W/)

TextContent liest eine Sprachdatei ein und stellt dem System die Übersetzung zur Verfügung. (/F195W/).

Resource ist der Datencontainer für die Daten einer Ressource.

Reservation ist der Datencontainer für die Informationen zu einer Belegung.

ReservationException speichert die Daten einer Ausnahme zu einer wiederholenden Belegung.

3.2 Controller

RoomBAServlet Das RoomBAServlet leitet Actions mit den benötigten Parametern von den JSP-Seiten an die entsprechende Klasse weiter und ruft je nach Rückgabe der Funktion doAction() die entsprechende JSP-Seite auf. Für das Mapping zwischen Anfrageparametern und Klassennamen wird das direkt von Java in der Class - Klasse bereitgestellte abstract Factory Pattern verwendet.

3.3 View

Mehrere JSP-Seiten bilden die View - Schicht. Hierbei besitzt jede unterstützte Sprache (/F195W/) ein Unterverzeichnis, in dem jeweils eine sprachangepasste Version der folgenden Seiten vorhanden ist:

3.3.1 Rahmenseiten

Header.jsp beinhaltet den Kopf jeder HTML-Seite und zeigt das RoomBA-Logo sowie vom Administrator festlegbare Informationen wie z.B. Seitentitel an.

Footer.jsp beinhaltet das Ende jeder HTML-Seite sowie konfigurierbaren Text.

NavBar.jsp bildet die Navigationsleiste und zeigt die Grundfunktionen an. Zusätzlich wird die Seite *Login.jsp* eingebunden bzw. nach erfolgreichem Login der Logout-Button (/F410/) und die zusätzlichen Funktionen von autorisierten Benutzern bzw. Administratoren zur Verfügung gestellt. Des weiteren werden dort die Auswahlmöglichkeiten für mehrsprachige Webseiten eingebunden (/F195W/).

Login.jsp stellt das Login-Formular zur Verfügung (/F400/).

HtmlView.jsp Rahmenseite für die anderen JSP - Seiten, in die Header.jsp, Footer.jsp, NavBar.jsp usw. eingebunden werden.

PdfView.jsp zeigt eine Seite in druckerfreundlicher Form an.

3.3.2 Für anonyme Benutzer sichtbare Seiten

Auf jeder dieser Seiten wird in der rechten oberen Ecke ein Hilfe-Symbol mit der Verknüpfung zu den Hilfeseiten angezeigt (/F185/). Seiten, die Formulare beinhalten, bieten die Möglichkeit, diese vorzubelegen (/L60/).

Search.jsp beinhaltet das Formular für die Standardsuche (/F50/) und bildet die Startseite von RoomBA. Ein vorläufiger Entwurf hiervon ist im Pflichtenheft in Kapitel 7 abgebildet.

ExtSearch.jsp beinhaltet ein erweitertes Formular, das die Expertensuche ermöglicht (/F60/ - /F80/).

ResourceSearch.jsp Auf dieser Seite wird die Suchfunktion nach freien Ressourcen für Belegungen mit den im Pflichtenheft beschriebenen Einschränkungen angeboten (/F90/).

ResourceInfo.jsp zeigt alle verfügbaren Ressourceninformationen an (siehe /F100/, /F105/, /F110/, /F115/) und bietet Administratoren die Möglichkeit, diese (/F512/) sowie die Ressourcenattribute (/F525/) zu ändern .

WeeklyView.jsp zeigt die übergebenen Belegungen in kalendarischer Form für eine Woche an (/F105/, /F115/, /F190W/) und verlinkt sie mit ihren Detailansicht (/F120/, /F130/). Desweiteren existieren Links zur vorangegangenen und nachfolgenden Woche (/F107/, /F117/).

DailyView.jsp zeigt die übergebenen Belegungen in kalendarischer Form für einen Tag an (/F100/, /F110/, /F190W/) und verlinkt sie mit ihren Detailansicht (/F120/, /F130/). Desweiteren existieren Links zum vorangegangenen und nachfolgenden Tag (/F107/, /F117/).

TableView.jsp zeigt die übergebenen Belegungen (z.B. /F50/ - /F90/, /F190W/) in tabellarischer Form an (/F150/, /F198/, /F199/) und verlinkt sie mit ihren Detailansichten (/F120/, /F130/).

ResourceView.jsp kombiniert die Anzeige der Raumdaten (*ResourceInfo.jsp*) und der Raumbelegungen (*WeelkyView.jsp*, *DailyView.jsp*, *TableView.jsp*) und verlinkt sie mit ihren Detailansichten (/F120/, /F130/).

Bietet ausserdem für den Administrator die Möglichkeit, diese Daten zu verändern (/F512/).

FreeTimeTableView.jsp zeigt die Suchergebnisse der Zeitraumsuche (/F90/) in tabellarischer Form an (/F150/) und verlinkt diese mit dem Formular zum Erstellen von Belegungen (/F200/).

FreeTimeCalenderView.jsp zeigt die Suchergebnisse der Zeitraumsuche (/F90/) als temporären Kalender an (/F160/) und verlinkt diese mit dem Formular zum Erstellen von Belegungen (/F200/).

ResourcesByGroup.jsp Anzeige der Räume sortiert nach ihren Gruppenzugehörigkeiten mit Verlinkung auf deren Detailansicht (*ResourceView.jsp*). Der Administrator erhält ausserdem die Möglichkeit, Räume zu erstellen und zu löschen (/F510/, /F511/).

ReservationDetails.jsp zeigt die Details einer Belegung an (/F130/) und bietet die Möglichkeit zur Kontaktaufnahme mit dem Veranstalter (/F140/).

Contact.jsp Formular zur Kontaktaufnahme mit dem Veranstalter einer Belegung (/F140/).

PersonView.jsp zeigt die sichtbaren Daten (siehe /F380/) eines autorisierten Benutzers an (/F170/).

Help.jsp zeigt eine Hilfeseite für die per Parameter übergebene Seite an (/F185/).

3.3.3 Für autorisierte Benutzer zusätzlich sichtbare Seiten

EditUserData.jsp zeigt sämtliche verfügbare eigenen (/F340/) Daten des Benutzers an (/F350/) und bietet die Möglichkeit, diese teilweise zu verändern (/F360/), sowie deren Sichtbarkeit festzulegen (/F380/). Ändert dieser die Email-Adresse, wird zur Verifikation eine Email versandt, in der ein Validierungscode enthalten ist (/F370/). Einem Administrator ist es gestattet, sämtliche Daten zu verändern (/F502/).

EditReservation.jsp bietet die Möglichkeit, neue, auch wiederholende, Belegungen zu erstellen (/F200/, /F210/), zu löschen (/F320/, /F330/) bzw. vorhandene eigene (/F240/) Belegungen zu ändern (/F250/, /F260/, /F290/). Hierbei wird bei einer Neuanlegung die Möglichkeit geboten, Belegungsdaten von anderen eigenen Belegungen zu importieren. Des weiteren werden Gruppenzugehörigkeiten und Ausnahmen angezeigt. Der Benutzer kann diese löschen, ändern oder neu anlegen (/F230/, /F270/). Hierzu wird auf die Seiten *EditGroupDependence.jsp* bzw. *EditException.jsp* verlinkt.

EditGroupDependence.jsp Die ausgewählte Gruppenzugehörigkeit wird angezeigt und kann geändert werden (/F270/). Ausserdem können hier neue Zugehörigkeiten erstellt werden (/F230/). Für Administratoren kann diese Seite alternativ dazu genutzt werden, Gruppenzugehörigkeiten von Räumen anzuzeigen und zu ändern (/F535/).

EditException.jsp Die Daten der Belegung sowie der Ausnahme werden angezeigt. Es kann festgelegt werden, ob, zu welchem Termin und wohin die Veranstaltung in dieser Ausnahme verschoben werden soll (/F295/, /F330/). Hierzu steht wiederum die Zeitraumsuche zur Verfügung.

VerifyEMail.jsp Zu dieser Seite wird nach erfolgreicher Validierung mit Hilfe der Bestätigungsemail (/F370/) verlinkt.

3.3.4 Für Administratoren zusätzlich sichtbare Seiten

GroupOverview.jsp zeigt die Belegungs- bzw. Raumgruppen gemäß ihrer Hierarchie an und bietet die Möglichkeit, diese zu löschen (/F531/, /F542/), sowie Links zu *EditGroup.jsp* um sie zu ändern (/F532/, /F541/) bzw. neue Gruppen anzulegen (/F530/, /F540/).

EditGroup.jsp zeigt eine (neue) Raum- bzw. Belegungsgruppe an und bietet die Möglichkeit, deren Namen und Obergruppe zu ändern (/F532/, /F541/).

UserOverview.jsp bietet eine Übersicht über alle autorisierten Benutzer und Administratoren an, verlinkt auf deren Benutzerdetails, wo deren Daten geändert werden können (/F502/). Hier können diese Benutzer auch gelöscht werden (/F501/), sowie neue autorisierte Benutzer und Administratoren angelegt werden (/F500/).

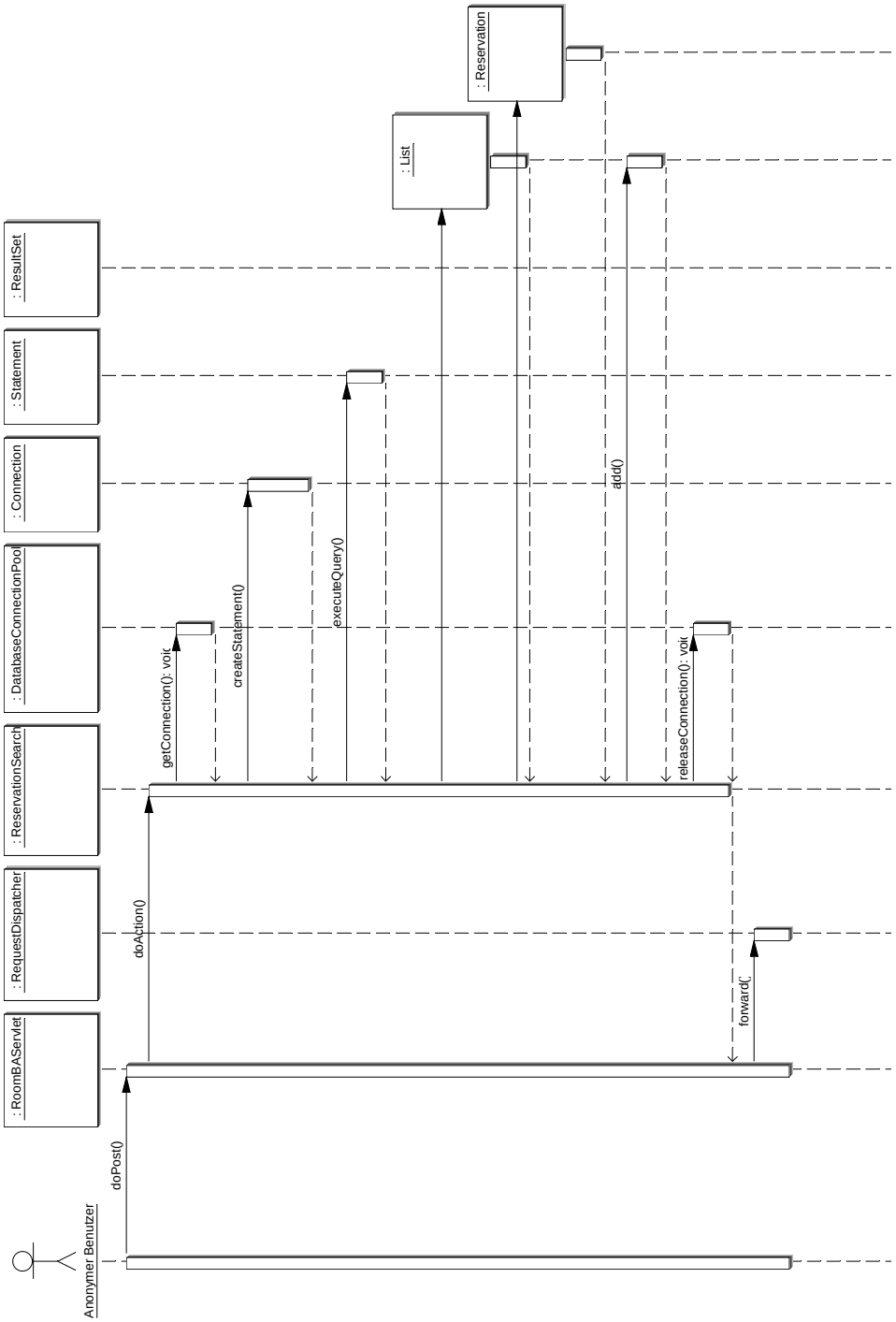
EditAttributeGroup.jsp zeigt eine Übersicht über alle vorhandenen Attribute an und ermöglicht, deren Namen zu ändern, sie zu löschen, sowie neue Attribute anzulegen (/F520/ - /F522/).

3.3.5 Pdf - Anzeige mittels Servlet

CreatePdfServlet dient der Erstellung einer PDF-Datei aus einer vorgegebenen HTML Seite./F180/

4 Datenfluss

Als Beispiel für den Datenfluss in unserem System steht das folgende Sequenzdiagramm der Expertensuche.



Der anonyme Benutzer startet eine Suchabfrage auf der ExtSearch.jsp. Das RoomBAServlet behandelt die Action und gibt die Anfrage mittels Class.forName() an die ReservationSearch - Klasse weiter. Diese holt sich von der DataBase-ConnectionPool - Klasse ein Connection-Objekt. Damit wird eine Anfrage an die Datenbank gestellt. Das Ergebnis der Anfrage wird in einer Liste von Reservation - Objekten gespeichert. Das Connection - Objekt wird wieder an den DataBaseConnectionPool zurückgegeben. Danach wird die ensprechende JSP - Seite aufgerufen, die die Daten aus den Reservation - Objekten ausliest.

5 Datenbankschema