

Prüfungsaufgaben

I148 - Internet-Anwendungsarchitekturen

Auf den folgenden Seiten werden die Themenvorschläge für die Hausarbeit in der Vorlesung „I148 - Internet-Anwendungsarchitekturen“ vorgestellt. Die Themen sind grundlegend aus einer funktionalen Sicht beschrieben, bedürfen aber vor der Entwicklung einer DV-technischen Konzeption.

Die Aufgaben werden in unterschiedlichen Schwierigkeitsgraden angeboten. Die Wahl des Schwierigkeitsgrads Aufgaben ist frei, jedoch erfolgt die Bewertung unter Berücksichtigung des ausgewählten Schwierigkeitsgrades.

Neben den in den einzelnen Themenvorschlägen beschriebenen Anforderungen sind die nachfolgend aufgezählten Punkte für alle Themen verpflichtend:

1. Vor und während der Entwicklung muss ein (kurz gehaltenes) DV-Konzept erstellt werden, das insbesondere die folgenden Punkte beinhaltet:
 - a. UML-Klassenmodell der zentralen Objekte (Domain Model) der Anwendung
 - b. Dialogmodell (Skizze der Maskenlayouts und Darstellung der Dialogflüsse)
 - c. Beschreibung der wesentlichen Testfälle (eine automatische Durchführung der Testfälle ist nicht erforderlich)
2. Annahmen/Abgrenzungen sind mit dem „Kunden“ abzusprechen und zu dokumentieren.
3. Die Entwicklung muss unter Nutzung der in der Vorlesung besprochenen Architektur (Struts 2 oder Angular, Spring, Hibernate) erfolgen. Jede Aufgabe ist mit einer verpflichtenden Frontendtechnologie ausgezeichnet. Der Einsatz von Zusatztechnologien ist erlaubt. Bei größeren Anpassungen sind diese mit dem „Kunden“ abzusprechen.
4. Es gelten die in den Vorlesungsfolien vorgestellten Musskriterien und funktionalen/nicht-funktionalen Anforderungen.
5. Bei Abgabe der Prüfungsleistung sind folgende Bestandteile verpflichtend:
 - a. Entwickelte Anwendung mit komplettem Quellcode (IntelliJ IDEA-Projektverzeichnis)
 - b. DV-Konzept
 - c. Installationsanleitung
 - d. Eidesstattliche Versicherung, dass die Arbeit selbständig verfasst wurde
 - e. Protokollierung der Projektverlaufs mit den geleisteten Stunden der Teammitglieder (Tag, Aufgabe, Stunden, Gesamtstunden pro Teammitglied)
6. Die Datenbank der Anwendung muss in einen initialen Stand versetzt werden können, der ein Testen der Anwendung ermöglicht.
7. Jedes Teammitglied führt sein eigenes Stunden- und Aufgabenprotokoll.
8. Genutzte Hilfsmittel und Quellen müssen angegeben werden.
9. Im Anschluss an der Abgabe der Hausarbeit wird eine Feedbackrunde mit jeder Gruppe durchgeführt und ggf. Detailfragen zur abgegebenen Prüfungsleistung gestellt. Die Teilnahme ist für alle Teammitglieder verpflichtend, der Terminabsprache erfolgt nach Abgabe der Hausarbeit.

Viel Erfolg!

Thema Studentendatenbank

(Schwierigkeitsgrad niedrig, Frontend Struts2)

Das System soll die Studenten der Nordakademie verwalten. Zurzeit liegen die Informationen in einer Excel-Tabelle in denormalisierter Form vor.

Folgende Informationen sind dort hinterlegt:

- Lfd. Nummer
- Name
- Vorname
- Geschlecht
- Anrede
- Matrikelnummer
- Straße
- PLZ
- Ort
- Geburtsdatum
- Geburtsort
- Telefon
- Studiengang
- Jahrgang
- Zenturie
- Userkennung
- Email
- Firma
- Firma2
- Firma kurz
- Adresszusatz
- Ansprechpartner
- Straße Firma
- PLZ Firma
- Ort Firma
- Telefon Firma
- Fax Firma
- Email Firma

Informationen über exmatrikulierte Studenten und Studenten ohne Abschluss werden in „Geschwister“-Tabellen gehalten, die eine ähnliche Struktur aufweisen.

Diese Tabellen spielt in wichtigen Geschäftsprozessen der Nordakademie wie Rechnungsstellung, Studentenausweise, Korrespondenz eine zentrale Rolle. Darüber hinaus ist diese Tabelle Vorlage für die Nutzerkennungen im Nordakademie-Informationssystem und in den Anwesenheitslisten und der Seminardatenbank.

Aufgabe der Semesterarbeit ist die Überführung der obigen Struktur in eine normalisierte Datenbank und die Bereitstellung von Pflege- und Anzeigefunktionen, die eine optimale Durchführung der Pflegeprozesse gewährleistet. Auf eine geeignete Validierungsprüfung bei der Erfassung der Daten ist dabei zu achten.

Anwendungsfälle

Folgende Anwendungsfälle sind zu realisieren:

1. Erfassen von Studienbewerbern

Das Prüfungsamt sammelt Daten über Studenten, die sich für einen Studienplatz bewerben. Änderungen und Löschungen von Bewerberdaten sind jederzeit möglich.

2. Immatrikulieren eines neuen Studenten

Wenn der Kooperationsvertrag mit einem Studenten und einem Unternehmen unterzeichnet ist, werden die Informationen der Studenten entweder neu erfasst oder es wird auf einen Datensatz eines Bewerbers zurückgegriffen. Es werden dem neuen Studenten ein Betrieb und ein Betreuer aus einer Auswahlliste zugeordnet. Dabei kann es passieren, dass der Betrieb und/oder der Betreuer noch nicht existieren. Ferner erfolgt die Einordnung in eine Manipel/Zenturie. Informationen zu Studenten können geändert werden. Allerdings können Studenten nicht mehr aus der Datenbank gelöscht werden, es kann lediglich ihr Status modifiziert werden.

3. Suchen von Studenten

Das System bietet eine komfortable Suchfunktion nach Studenten, die nach unterschiedlichen, teilweise optionalen Kriterien durchgeführt wird. Aus der angezeigten Liste kann die Bearbeitung eines Datensatzes begonnen werden.

4. Erstellen von Anwesenheitslisten/Ergebnislisten

Das System ist in der Lage am Anfang eines jeden Semesters eine Vorlage für eine Anwesenheitsliste für eine Zenturie beziehungsweise eine Ergebnisliste für eine Manipel zu erzeugen.

5. Verwaltungsbereich

Erforderliche Referenztabellen (z.B. Manipel, ...) sind in einem Verwaltungsbereich pflegbar zu machen.

Thema Bibliotheksverwaltung (Schwierigkeitsgrad hoch, Frontend Struts2)

Für die Bibliothek der Nordakademie soll ein neues Verwaltungssystem eingeführt werden, dass die Verwaltung des Bücherbestands und die Verfolgung der Leihvorgänge ermöglicht.

Aufgabe der Hausarbeit ist die DV-technische Konzeption und anschließende Realisierung der nachfolgend aufgezählten Anwendungsfälle im Rahmen einer neuen, webbasierten Anwendung.

Anwendungsfälle

Folgende Anwendungsfälle sind zu realisieren:

1. Verwaltung des Bibliotheksbestands

Eine der Hauptaufgaben ist die Erfassung und Verwaltung des aktuellen Bestands der Bibliothek. Die Verwaltungsfunktionen beinhalten das Anzeigen, Anlegen, Editieren und Löschen von Publikationen aus dem Bestand.

Der Benutzer muss die Möglichkeit haben, folgende Daten zu einer Publikation zu hinterlegen:

- Eindeutiger, NAK-interner Publikationsschlüssel (obligatorisch)
- Titel
- Autor(en)
- Datum der Veröffentlichung
- Verlag
- Art der Publikation (bspw. Buch, Zeitschrift, Hausarbeit, Bachelorarbeit etc., Auswahlliste)
- ISBN
- Schlagwörter (Auswahlliste, Mehrfachselektion möglich)
- Anzahl der Exemplare im Bestand

Dabei muss beachtet werden, dass nicht alle Daten immer angegeben werden können. Bspw. hat eine Hausarbeit keine ISBN.

2. Suchen von Publikationen

Das System bietet eine komfortable Suchfunktion, die nach unterschiedlichen, teilweise optionalen Kriterien durchgeführt wird. Aus der angezeigten Liste kann die Bearbeitung eines Datensatzes begonnen werden.

3. Erfassung von Ausleihvorgängen

Neben der Verwaltung des eigentlichen Bestands sollen auch die Ausleihvorgänge über das System abgebildet werden. Die Anwendung muss dafür sowohl das Ausleihen von Publikationen als auch die Rückgabe erfassbar machen.

Folgende Daten sollen für einen Ausleihvorgang hinterlegt werden:

- Datum der Ausgabe
- Ausgeliehene Publikation
- Persönliche Daten des Ausleihers (z.B. Vorname, Name, Matrikelnummer)
- Rückgabedatum (automatisch vom System vergeben)

Das System muss sicherstellen, dass ein Ausleihvorgang nur bei ausreichendem Bestand möglich ist. Ist kein Exemplar mehr ausleihbar, muss zuerst eine Rückgabe eines Exemplars erfolgen, bevor der nächste Ausleihvorgang möglich ist.

4. Übersicht der offenen Ausleihvorgänge

Die Anwendung bietet eine Übersicht über die aktuell offenen Ausleihvorgänge. Diese erfolgt einerseits in Form einer tabellarischen Gesamtübersicht über alle ausgeliehenen Exemplare. Darüber hinaus werden auch in der Detailansicht einer einzelnen Publikation die zugehörigen, offenen Ausleihvorgänge angezeigt.

5. Mahnwesen

Die Bibliotheksverwaltung soll ein Mahnwesen bekommen, mit dem es möglich ist, säumige Ausleiher zu erkennen. Eine Ausleihe gilt als überzogen, sobald das Rückgabedatum um einen Tag überschritten ist. Für überzogene Ausleihen wird dann ein Mahnvorgang eröffnet. Innerhalb des Mahnvorgangs können bis zu drei Mahnungen mit dem jeweiligen Mahndatum erfasst werden. Zwischen dem Versenden zweier Mahnungen liegt in der Regel eine Woche.

Alle eröffneten Mahnvorgänge und die noch zu mahnenden Ausleihvorgänge werden in einer Tabelle auf der Hauptseite der Anwendung dargestellt, um dann durch den Anwender verfolgt werden zu können.

Es gibt zwei Möglichkeiten, einen Mahnvorgang zu beenden:

- 1.) Durch Erfassung der Rückgabe.
- 2.) Durch Markieren des Mahnvorgangs als „Verlust“. Der Bestand in der Bibliothek muss in diesem Fall entsprechend verringert werden. Ein Mahnvorgang darf erst als „Verlust“ markiert werden, wenn der säumige Ausleiher drei Mahnungen erhalten hat.

6. Verwaltung von Stammdaten

Für die Stammdaten "Art der Publikation", "Schlagwort" und "Ausleiher" muss es eigene Verwaltungsfunktionen geben, die eine Pflege (Anzeige, Anlegen, Bearbeiten, Löschen) dieser Daten erlauben.

7. Sortierbare Tabellenanzeigen

Alle tabellarischen Ansichten in der Anwendung (bspw. Suchergebnislisten) lassen sich nach den angezeigten Werten auf- und absteigend sortieren.

Weitere Randbedingungen:

Das System muss sämtliche Benutzereingaben auf gültige Werte überprüfen, um eine möglichst hohe Datenqualität im System zu gewährleisten. Die Datenhaltung soll in einer normalisierten Form erfolgen.

Das Rückgabedatum wird vom System automatisch auf Basis einer hinterlegten Verleihperiode gesetzt. Diese muss nicht über Anwendungsoberflächen pflegbar oder in der Datenbank hinterlegt sein, sollte sich jedoch zentral in der Anwendung ändern lassen (Konstante bzw. ausgelagerter Parameter).

Optionale Anwendungsfälle:

Die folgenden Anwendungsfälle können optional im Rahmen der Hausarbeit umgesetzt werden:

a) Verlängerung eines Verleihvorgangs

Ein Verleihvorgang kann durch den Anwender bis zu zwei Mal verlängert werden. In diesem Fall wird das Rückgabedatum jeweils um eine Verleihperiode nach hinten verlängert. Bitte beachten, dass auch die Verlängerung einer Ausleihe einen ggf. laufenden Mahnvorgang beenden muss.

Thema Mitgliederverwaltung (Schwierigkeitsgrad niedrig, Frontend Angular)

Der Reitverein Elmshorn verwaltet seit jeher seine Mitglieder in Form einer Excel Tabelle.

Alle Mitglieder haben eine eindeutige Mitgliedsnummer. Neue Mitglieder werden zurzeit als Zeilen angefügt und erhalten die nächste freie Mitgliedsnummer. Einmal vergebene Nummern werden nicht neu vergeben. Ausgetretene Mitglieder werden aus datenschutzrechtlichen Gründen gelöscht. Austritte erfolgen immer zum Jahresende, sie müssen mindestens 3 Monate vor Ende des Kalenderjahres eingereicht werden. Familienmitglieder eines Mitglieds erhalten einmal 3 Euro Rabatt pro Jahr.

Es soll ein webbasiertes Programm erstellt werden, um die Verwaltung der Mitglieder zu vereinfachen und die Datenqualität zu erhöhen.

Anwendungsfälle

Folgende Anwendungsfälle sind zu realisieren:

1. Anlegen neuer Mitglieder

Das Programm erlaubt es, neue Mitglieder mit allen notwendigen Daten anzulegen.

Bisher werden in der Excel Tabelle folgende Daten erfasst:

- Mitgliedsnummer
- Name
- Adresse
- Geburtstag
- Datum Eintritt
- Datum Kündigung
- Datum Austritt
- Mitgliedsart
- aktueller Jahresbeitrag
- pro Jahr eine Spalte Beahlt/Betrag
- Kontoverbindung
- Verweis zum zuerst eingetretenen Familienmitglied (wenn vorhanden)

Folgend finden sich die Jahresbeiträge für die unterschiedlichen Mitgliedsarten:

- Vollmitglied: 25€
- Ermäßigt: 23€
- Jugendliche (bis 18 Jahre): 15€
- Fördermitglied: 10€
- Rabatt für Familienmitglieder: 3€

2. Stammdatenpflege

Das System bietet eine einfache Stammdatenpflege mit Hilfe derer benötigte Datensätze verwaltet werden können.

3. Suchen von Mitgliedern

Das System bietet eine komfortable Suchfunktion, die nach unterschiedlichen, teilweise optionalen Kriterien durchgeführt wird. Aus der angezeigten Liste kann die Bearbeitung eines Datensatzes begonnen werden.

4. Bearbeitung bestehender Mitglieder

Der Wechsel der Mitgliedsart kann nur zum Jahreswechsel erfolgen.

5. Löschen von ausgetretenen Mitgliedern

Beachten Sie die Anforderung 7.

6. Übersicht aller Mitglieder

Tabellarische Übersicht aller aktuellen Mitglieder, zusätzliche Druckansicht als Nachweis für Jahresabschluss.

7. Übersicht der zu entrichtenden/entrichteten Jahresbeiträge

Tabellarische Übersicht Bankverbindung, Beitrag, Zahlstatus für ein ausgewähltes Jahr, zusätzlich Druckansicht als Nachweis für den Jahresabschluss).

Weitere Randbedingungen:

Das System muss sämtliche Benutzereingaben auf gültige Werte überprüfen, um eine möglichst hohe Datenqualität im System zu gewährleisten. Die Datenhaltung soll in einer normalisierten Form erfolgen.

Fehlerverfolgungssystem (Schwierigkeitsgrad niedrig, Frontend Angular)

Es soll ein System implementiert und bereitgestellt werden, das eine einfache Erfassung und Verfolgung von Fehlern in der Entwicklung eines Softwaresystem unterstützt. Fehler können von Benutzern eingestellt und beschrieben werden und durchlaufen anschließend einen Prozess, der letztendlich entweder zur Beseitigung des Fehlers oder zu einer Zurückweisung des erfassten Fehlers führt.

Funktionale Anforderungen

Folgende funktionale Anforderungen muss das System erfüllen:

1. Es gibt einen Registrierungsdialog, über den sich neue Entwickler im System registrieren können. Jeder Entwickler hat dabei einen vollständigen Namen, eine E-Mail-Adresse und ein Passwort. Er wird durch eine eindeutige Benutzerkennung identifiziert.
2. Jeder Entwickler muss sich vor der Benutzung des Systems mit seiner Benutzerkennung und dem Passwort anmelden.
3. Ein Fehler kann einen der folgenden Zustände haben:
 - a. Anlegt
 - b. In Bearbeitung
 - c. Behoben
 - d. Abgelehnt
 - e. Wiedereröffnet
 - f. Geschlossen
4. Jeder im System angemeldete Entwickler kann einen neuen Fehlerbericht im System anlegen. Dabei wird neben einer kurzen Überschrift auch eine textbasierte Beschreibung der Fehlersituation vom Benutzer angegeben. Das System weist dem Fehler eine eindeutige Fehlernummer zu. Der Fehler bekommt initial den Status „Angelegt“. Der Ersteller des Fehlerberichts wird als Autor hinterlegt, das aktuelle Datum und die Uhrzeit werden ebenfalls erfasst.
5. Ein Entwickler kann einen Fehler mit Status „Angelegt“ oder „Wiedereröffnet“ auswählen und die Bearbeitung des Fehlers starten. Der Zustand wird auf „In Bearbeitung“ gesetzt und der Entwickler als Bearbeiter hinterlegt. Ein Fehler kann immer nur von einem Entwickler bearbeitet werden.
6. Der Bearbeiter eines Fehlers kann diesen in den Zustand „Behoben“ oder „Abgelehnt“ setzen. Die Bearbeitung ist damit abgeschlossen.
7. Der Autor des Fehlerberichts kann einen Fehlerbericht mit Status „Behoben“ oder „Abgelehnt“ in die Folgezustände „Wiedereröffnet“ oder „Geschlossen“ setzen.
8. Fehlerberichte mit Status „Geschlossen“ können nicht mehr bearbeitet werden.
9. Bei allen Zustandsänderungen können die jeweiligen Entwickler Kommentare hinterlegen, die chronologisch aufsteigend im Fehlerbericht angezeigt werden. Der Name des Entwicklers wird mit dem Kommentar angezeigt. Außerdem soll der Zeitpunkt der letzten Änderung des Fehlerberichts festgehalten werden.
10. Das System soll eine tabellarische Übersicht mit den im System erfassten Fehlerberichten anbieten. Aus dieser Sicht heraus sind die einzelnen Bearbeitungen möglich.

11. Optionale Anforderung: Die tabellarische Übersicht soll eine Sortierung nach verschiedenen Kriterien (Fehlernummer, Datum der Erstellung, ..) ermöglichen.
12. Optionale Anforderung: Die tabellarische Übersicht soll eine Filtermöglichkeit anbieten, die alle geschlossenen Fehlerberichte ausblendet.

Terminumfrage

(Schwierigkeitsgrad hoch, Frontend Angular)

Thema dieser Aufgabe ist ein webbasiertes Terminumfragesystem (analog bspw. Doodle). Das implementierte System soll die Festlegung von Besprechungs- und Gruppenterminen erleichtern, indem Terminumfragen von einem Organisator eingestellt werden können. Die betroffenen Umfrageteilnehmer können dann ebenfalls über die Anwendung Zu- oder Absagen zu den einzelnen vorgeschlagenen Terminen und Uhrzeiten melden, so dass nach Rückmeldung aller Teilnehmer der Organisator den günstigsten Termin auswählen kann.

Im Gegensatz zu verfügbaren Systemen wie Doodle soll das implementierte System die Funktionen nicht offen und anonymisiert anbieten, sondern nur von autorisierten Benutzern verwendet werden können.

Funktionale Anforderungen

Folgende funktionale Anforderungen muss das System erfüllen:

1. Es gibt einen Registrierungsdialog, über den sich neue Benutzer im System registrieren können. Jeder Benutzer hat dabei einen vollständigen Namen, eine Email-Adresse und ein Passwort. Er wird durch eine eindeutige Benutzerkennung identifiziert.
2. Jeder Benutzer muss sich vor der Benutzung des Systems mit seiner Benutzerkennung und dem Passwort anmelden.
3. Jeder Benutzer kann eine eigene neue Terminumfrage einstellen. Neben einer Überschrift und einer textbasierten Beschreibung gibt er dabei einen oder mehrere mögliche Termine an (Datum und Uhrzeit). Nach Abschluss der Aktion wird die Terminumfrage im System angelegt und wird anschließend allen angemeldeten Benutzern zur Teilnahme angeboten. Der Benutzer selbst wird als Ersteller der Terminumfrage in der Datenbank hinterlegt und mit angezeigt.
4. Der Ersteller einer Terminumfrage kann selbst keine Rückmeldung zu dieser Terminumfrage erfassen. Als Ersteller wird davon ausgegangen, dass er an allen vorgeschlagenen Terminen anwesend ist.
5. Jeder Benutzer kann an bereits erstellten Terminumfragen teilnehmen. Dazu werden die vom Ersteller angegebenen Daten sowie die bereits hinterlegten Rückmeldungen von anderen Benutzern angezeigt. Der Benutzer kann nun zu jedem einzelnen Termin der Umfrage den Status „Okay“ oder „Nicht okay“ hinterlegen, um so seine An- oder Abwesenheit mitzuteilen. Der Benutzer kann die Terminumfrage selbst oder bereits hinterlegte Rückmeldungen von anderen Benutzern nicht verändern. Beim Speichern wird die Rückmeldung mit Bezug auf den aktuellen Benutzer hinterlegt. Dieser wird bei späteren Ansichten der Umfrage als Ersteller der Rückmeldung mit ausgegeben. Auf diese Weise entsteht die Übersicht der einzelnen Terminteilnehmer.
6. Ein Benutzer kann seine Rückmeldung zu einem Termin nachträglich editieren und den Status verändern, er kann die Rückmeldung jedoch nicht mehr löschen.
7. Der Ersteller einer Terminumfrage kann diese nachträglich verändern. Dabei können sowohl die Textinformationen als auch die vorgeschlagenen Termine verändert werden. Eine Änderung der Terminumfrage hat zur Folge, dass alle bereits hinterlegten Rückmeldungen von anderen Benutzern gelöscht werden und die Abstimmung von vorne beginnt.

8. Der Ersteller einer Terminumfrage kann diese löschen. Nach einer Sicherheitsabfrage werden dann sowohl die Terminumfrage als auch alle hinterlegten Rückmeldungen gelöscht.
9. Terminumfragen können von allen Benutzern jederzeit zur Anzeige aufgerufen werden. Dabei werden die hinterlegten Textdetails als auch die einzelnen Rückmeldungen der Benutzer in sinnvoller Form angezeigt. Die Übersicht der Rückmeldungen zu den einzelnen Terminvorschlägen der Umfrage soll in tabellarischer Form erfolgen. Am unteren Ende der Tabelle soll das System eine Summe anzeigen, die die „Okay“-Meldungen zum jeweiligen Terminvorschlag aufsummiert. So ist schnell erkennbar, welcher Terminvorschlag aktuell die meisten positiven Rückmeldungen hat.
10. Das System soll jedem Benutzer eine Übersichtsseite anbieten. Diese listet möglichst sinnvoll und platzsparend eine Grobübersicht aller Terminumfragen auf. Dabei soll unterschieden werden in Terminumfragen, die der Benutzer selbst erstellt hat, Terminumfragen, für die der Benutzer bereits eine Rückmeldung erfasst hat, sowie Terminumfragen, die noch nicht beantwortet wurden.
11. Wenn ein Ersteller seine Terminumfrage ändert und damit die Rückmeldungen anderer Benutzer löscht, so soll das System anschließend bei allen betroffenen Benutzern nach der nächsten Anmeldung im System eine entsprechende Meldung anzeigen. Auf diese Weise erfahren die Benutzer, an welchen Umfragen sie erneut teilnehmen müssen.
12. Ein Ersteller kann seine Terminumfrage für beendet erklären. Weitere Rückmeldungen oder ein Editieren der Terminumfrage sind dann nicht mehr möglich. Der Ersteller wählt dabei einen seiner Terminvorschläge aus, um diesen als den letztendlichen Termin zu deklarieren. Das muss dabei nicht unbedingt der Termin mit den meisten positiven Rückmeldungen sein. Alle Benutzer, die für den ausgewählten Termin mit „Okay“ gestimmt haben, bekommen diesen anschließend in der Übersichtsseite als festen Termin angezeigt.