

## Prüfungsleistung

In diesem Semester geht es darum, Anwendungen zu entwickeln, die verschiedene einfache Prozesse in Vereinen unterstützen. Team n bearbeitet dabei Thema n, die Themen finden Sie unten.

### Abgabe

- Dokumentation des fachlichen und des technischen Entwurfs ihrer Anwendung als UML-Diagramme (Sie können, müssen dabei aber kein UML-Modellierungstool verwenden; abzugeben ist jeweils ein PDF-Datei mit den Diagrammen).
- Programmcode sowie Skripte zum Datenbankaufbau.  
(Am Beispiel NetBeans: Exportieren Sie das Projekt für den Client und das Projekt für den Server jeweils als Zip-File (File → Export → To Zip ...))
- Präsentationsunterlagen für die Abschlusspräsentation.
- Eine 2-3 seitige Reflektion zu Ihrem Lernprozess (Aspekte können unter anderem sein: Was ist uns weniger gut gelungen und woran könnte das gelegen haben? Was könnten wir das nächste Mal anders machen? Was ist uns besonders gut gelungen? Bei welchen Themen ist uns ein wirkliches Verstehen schwergefallen? Woran könnte das gelegen haben? Bei welchen schwierigen Themen konnte wird Verständnishürden überwinden? Wie ist uns das gelungen? Welche Themen waren besonders bedeutsam für uns und warum? Welche Kompetenzen wurden durch das Programmierprojekt ausgebaut und verfestigt? Welche Querverbindungen können wir ziehen zu Inhalten anderer Vorlesungen?)

### Anforderungen

- Die Software muss dabei das in der Lehrveranstaltung vorgestellte Framework „Jakarta“ verwenden, so wie es in der Veranstaltung besprochen wurde (JSF für den Client, Kommunikation über REST-Services mit dem Server, Entwicklung des Servers unter Nutzung von EJB, Zugriff auf die relationale Datenbank per JPA). Die Nutzung anderer Frameworks (z.B. SpringBoot) ist nicht zulässig, weitere Bibliotheken (z.B. Jackson) dürfen nur nach vorheriger Absprache verwendet werden.
- Für die Realisierung des Servers gilt außerdem: Die Implementierung muss den in der Veranstaltung vorgestellten Architektur- und Entwurfsmuster genügen, genauer: Der Server muss in die Packages „facade“, „core“ (mit den Packages „entities“ und „use cases“) und „dataaccess“ strukturiert sein. Business Objekte dürfen nicht direkt in der Datenbank persistiert werden, dafür sind spezielle Entity-Objekte erforderlich. Der Datenzugriff erfolgt in DAO-Klassen. Die Klassen im Package „core“ sind bis auf die Annotationen für die Session Beans frei von Annotationen
- Für die Realisierung des Clients gilt: Realisiert werden soll eine Multipage-Anwendung. Jeder Dialog hat seine eigene Managed Bean; ein Dialog besteht dabei üblicherweise aus 2 Pages.
- Insgesamt gilt: Fachliche Prüfungen gehören in den Server, nicht in den Client. Die Funktionalität des Servers muss vollständig über REST-Services verfügbar sein und getestet werden können.

### Weiterer Ablauf

- Kurze Zwischenpräsentation (ca. 10 Minuten pro Gruppe) am 14.06.
- Abschlusspräsentation (ca. 20 Minuten) pro Gruppe am 05.07.2024
- Die übrigen Termine stehen für die gemeinsame Arbeit an der Prüfungsleistung zur Verfügung
  - Ausnahme: Keine Veranstaltung am 31.05., stattdessen am 29.05. von 17.00 Uhr bis 18.30 Uhr ein Online-Termin mit der Möglichkeit, Fragen zu stellen

## Aufgabenstellungen

### Hinweise:

- **Zur Zuordnung:** Team  $n$  bearbeitet Thema  $n$
- **Zum fachlichen Umfang:** Die Funktionalität der Anwendung wird mit User Stories beschrieben, werden User Stories verwendet. User Stories sind bekanntlich keine detailliert ausformulierten funktionalen Anforderungen, sondern sind „negotiable“; wenden Sie sich daher bei Rückfragen zum fachlichen Umfang an Prof. Schekelmann.
- **Zur Beschreibung:** Aus Gründen der besseren Lesbarkeit sind die User Stories in der grammatikalisch männlichen Form beschrieben.

### Team 1: Ausleihliste von Sportgeräte in einem Sportverein

- Als Benutzer will ich mich in der Anwendung als Mitglied oder Gerätewart registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Bankverbindung, Benutzergruppe (Mitglied oder Gerätewart)
- Als Benutzer will ich mich in der Anwendung als Mitglied oder Gerätewart anmelden
- Als Gerätewart will ich Sportgeräte anlegen, ändern oder löschen
  - Das System soll folgende Informationen zu einem Sportgerät verwalten: Name zuständiger Gerätewart
- Als Mitglied will ich die Ausleihe eines Sportgeräts beantragen.
  - Gegeben sein ein Mitglied, dass bereits 5 Sportgeräte ausgeliehen hat. Wenn ein Mitglied ein weiteres Sportgerät ausleiht, so soll das System en Antrag verweigern
- Als Organisator will ich die Ausleihe eines Sportgeräts erlauben oder verweigern.
  - Das System soll folgende Informationen zur Ausleihe verwalten: Sportgerät, Mitglied, Ausleihdauer
- Als Organisator will ich die Rückgabe eines Sportgeräts durch ein Mitglied verbuchen
- Als Mitglied will ich eine Liste aller meiner Beantragungen (inkl. ob Antrag gewährt oder nicht) und meiner offenen Ausleihen sehen (inkl. spätestens Rückgabedatum)
- Als Organisator will ich eine Liste aller überfälligen Ausleihen sehen.

### Team 2: Kursanmeldung in einem Sportverein

- Als Benutzer will ich mich in der Anwendung als Mitglied oder Kursleiter registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Bankverbindung, Benutzergruppe (Mitglied oder Kursleiter)
- Als Benutzer will ich mich in der Anwendung als Mitglied oder Kursleiter anmelden
- Als Kursleiter will ich Kurse anlegen, ändern oder löschen
  - Das System soll folgende Informationen zu einem Kurs verwalten: Name, Startdatum, Anzahl Termine, Rhythmus (wöchentlich, zweiwöchentlich, monatlich), Kursleiter
- Als Mitglied will ich mich zu einem Kurs an- oder abmelden.
  - Gegeben sei ein Kurs, der schon begonnen hat. Wenn ein Mitglied versucht sich zu diesem Kurs abzumelden, soll das System das Mitglied darauf hinweisen, dass eine Abmeldung nicht mehr möglich ist.
  - Gegeben sein ein Kurs, der bereits abgeschlossen ist. Wenn ein Mitglied versucht sich zu diesem Kurs anzumelden, soll das System das Mitglied darauf hinweisen, dass eine Anmeldung nicht mehr möglich ist.
- Als Kursleiter will ich eine Liste mit allen Anmeldungen zu einem bestimmten Kurs sehen

**Team 3: Verwalten einer Helferliste für Vereinsveranstaltungen**

- Als Benutzer will ich mich in der Anwendung als Mitglied oder Organisator registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Mitglied oder Organisator)
- Als Benutzer will ich mich in der Anwendung als Mitglied oder Organisator anmelden
- Als Organisator will ich Events anlegen, ändern oder löschen
  - Das System soll folgende Informationen zu einem Event verwalten: Name, Termin, Organisator
- Als Mitglied will ich mich für ein Event als Helfer ein- bzw. austragen
  - Gegeben sei ein Event, dass in weniger als 7 Tagen beginnt. Wenn ein Mitglied sich für dieses Event als Helfer einträgt, so soll das System das Mitglied darauf hinweisen, dass es sehr ungünstig ist, wenn das Mitglied sich jetzt noch austrägt, die Austragung aber durchführen.
- Als Organisator will ich eine Liste mit allen eingetragenen Helfern zu einem bestimmten Event sehen

**Team 4: Organisation von Konzerten in einem Musikverein**

- Als Benutzer will ich mich in der Anwendung als Dirigent oder Musiker registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Dirigent oder Musiker), beim Musiker außerdem das gespielte Instrument
- Als Dirigent will ich Stücke und Komponisten anlegen, ändern oder löschen
  - Zu Stücken sollen Name, Zeitpunkt der Komposition und Aufführungsdauer verwaltet werden, zum Komponisten Name und Vorname
- Als Dirigent will ich Konzerttermine anlegen, ändern oder löschen können
  - Zu einem Konzert sollen Ort, Datum und Uhrzeit und Dauer verwaltet werden
- Als Dirigent will ich die Zuordnung von Stücken zu Konzertterminen verändern können
  - Gegeben sei eine Zuordnung von Stücken zu einem Konzerttermin. Wenn der Dirigent dem Konzert ein weiteres Stück zuordnet und die gesamte Aufführungsdauer aller dem Konzerttermin zugeordneten Stücke die Dauer des Konzerttermins überschreitet, so soll das System die Zuordnung zurückweisen.
- Als Dirigent will ich die Reihenfolge der Zuordnung von Stücken einem Konzerttermin verändern können
- Als Benutzer will ich zu einem Konzerttermin alle Stücke in der richtigen Reihenfolge sehen

**Team 6: Organisation von Proben in einem Musikverein**

- Als Benutzer will ich mich in der Anwendung als Dirigent oder Musiker registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Dirigent oder Musiker), beim Musiker außerdem das gespielte Instrument
- Als Dirigent will ich Stücke und Komponisten anlegen, ändern oder löschen
  - Zu Stücken sollen Name, Zeitpunkt der Komposition und Aufführungsdauer verwaltet werden, zum Komponisten Name und Vorname
- Als Dirigent will ich Probentermine anlegen, ändern oder löschen können
  - Zu einer Probe sollen Ort, Datum und Uhrzeit verwaltet werden
- Als Dirigent will ich die Zuordnung von Stücken zu Proben verändern können
- Als Benutzer will ich zu einer Probe alle Stücke sehen

**Team 7: Meinungsbildung in einem Ortsverein einer Partei**

- Als Benutzer will ich mich in der Anwendung als Ortsvereinsvorsitzender oder Parteimitglied registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Ortsvereinsvorsitzender oder Parteimitglied)
- Als Ortsvereinsvorsitzender will ich Themen anlegen, ändern und löschen
  - Das System soll zu einem Thema einen Titel, eine Beschreibung und bis zu 3 Schlagwörtern verwalten.
- Als Ortsvereinsvorsitzender will ich die Diskussion eines Themas starten und beenden
- Als Parteimitglied will ich die Diskussion zu einem Thema verfolgen können.
- Als Benutzer will ich mich mit eigenen Diskussionsbeiträgen an einer Diskussion beteiligen können
  - Das System soll zu einem Diskussionsbeitrag folgende Informationen verwalten: Zugehöriges Thema, Beitrag, Datum des Beitrags, Parteimitglied
  - Gegeben sein eine Diskussion zu einem Thema. Wenn ein Benutzer sich an einer Diskussion beteiligen möchte und die Diskussion noch nicht gestartet oder bereits beendet ist, dann soll das System den Diskussionsbeitrag zurückweisen.
- Als Benutzer will ich alle Diskussionen zu Themen sehen, die ein bestimmtes Schlagwort enthalten

**Team 8: Vorschlagswesen in einem beliebigen Verein:**

- Als Benutzer will ich mich in der Anwendung als Vorsitzender oder Mitglied registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Vorsitzender oder Mitglied)
- Als Mitglied will ich Verbesserungsvorschläge machen können
  - Das System soll zu einem Verbesserungsvorschlag einen Titel, eine Beschreibung und den Vorschlagenden verwalten.
- Als Vorsitzender will ich einen Vorschlag zur Abstimmung stellen
- Als Mitglied will ich für oder gegen einen Vorschlag stimmen
- Als Vorsitzender will zu einem Vorschlag das anonymisierte Abstimmungsergebnis sehen
- Als Mitglied will ich sehen, wie ich zu einem Vorschlag abgestimmt habe

**Team 9: Vogelzählung in einem Naturschutzverein**

- Als Benutzer will ich mich in der Anwendung als Vorsitzender oder Unterstützer registrieren
  - Das System soll folgende Informationen zu einem Benutzer verwalten: Name, Vorname, Adresse, Benutzergruppe (Vorsitzender oder Unterstützer)
- Als Vorsitzender will ich Vogelarten verwalten
  - Das System soll folgende Informationen zu Vogelarten verwalten: Name, typische Erkennungsmerkmale
- Als Vorsitzender will ich für einen bestimmten Tag zu einer Vogelzählung aufrufen
  - Das System soll folgende Informationen zum Aufruf verwalten: Datum, Ort
- Als Vorsitzender will ich für eine Vogelzählung festlegen, welche Vögel gezählt werden sollen.
- Als Mitglied will ich Vogelsichtungen zu einer Vogelzählung melden
  - Das System soll folgende Informationen zu jeder Vogelsichtung verwalten: Ort, Datum, Dauer der Zählung, gesichtete Vögel zu Vogelart.
- Als Vorsitzender will ich zu einer Vogelzählung die Gesamtsummen gesichteter Vögel nach Vogelarten sehen.