Projektaufgabe – Projektaufgabe

Version 1.0
Praktikum XML-Technologie
Ausgabe am 18. Dezember 2019

Vorbemerkung

Damit wir von der Praktikumsleitung her verschiedene Lösungen parallel installieren können, fügen Sie bitte zu den folgenden Namen Ihren Gruppennamen als Präfix hinzu:

- Zu dem Ordner mit ihren Projektdaten im Verzeichnis webapp von BaseX.
- Zu dem Namen des maximal einen Ordners, den Sie im Verzeichnis static von BaseX unterbringen wollen.
- Zu allen Pfaden in RestXQ-Annotationen.
- Zu allen Namen von Datenbanken in BaseX.

Projektaufgabe

Das Praktikumsprojekt ist Blackjack. Beschreiben Sie in Ihrer Dokumentation eine Variante, die Sie realisieren. Sie dürfen Abstriche bei der vollen Funktionalität, z.B. bei Wetteinsätzen, machen. Mehrere Personen sollen jeweils in ihrem eigenen Browser-Fenster spielen können (Multi-Client-Lösung). Ihre Blackjack-Anwendung soll eine Lounge anbieten, in der offene Spiele ausgewählt werden können oder ein neues Spiel gestartet werden kann. In der Lounge sollen sich Spieler*innen registrieren können (es genügt, einen Namen anzugeben; eine Passwort-Verwaltung ist nicht notwendig). Die Lounge soll eine Highscore-Liste für abgeschlossene Spiele anbieten.
Im Unterschied zur Beschreibung auf Blatt 2 verlangen wir nicht mehr dass Spiele gespeichert und neu geladen werden können, da das im Multi-Client-Context konzeptionell nicht so einfach zu handhaben ist. Beachten Sie jedoch die sonstigen Überlegungen zur Lounge, die in Blatt 2 aufgeführt sind.
Entwerfen, **implementieren** und **dokumentieren** Sie das Spiel Blackjack als Web-Anwendung, rein mit XML-Technologien (kein JavaScript!). Serverseitig verwenden Sie BaseX in einem WebSocket/STOMP-fähigen Web-Server (Jetty) zur Speicherung von XML-Daten und zur Ausführung von XQuery-Funktionen. Für das Mapping von Requests auf XQuery-Funktionen verwenden Sie Annotationen in XQuery-Modulen nach RestXQ. Client-seitig verwenden Sie HTML mit eingegebetteten Formularen und SVG mit dem WebSocket Element. Ein Blackjack-Server muss parallele Spiele von verschiedenen Clients unterstützen. Ein einzelnes Blackjack-Spiel wird auf verschiedenen Clients gespielt (Multi-Client-Lösung).
Zum Entwurf gehören:

- Eine Beschreibung der Benutzungsoberfläche für die verschiedenen Stadien des Spiels mit den jeweiligen Interaktionsmöglichkeiten in Form von Skizzen (Blatt 2).
- Ein UML-Klassendiagramm mit Daten und Methoden (Blätter 3 und 4).
- Gegebenenfalls ein Activity Diagram oder State Diagram zur Beschreibung des Systemverhaltens, inklusive Begründung für den Diagramm-Typ (Blätter 4 und 5.)
- Eine Beschreibung der Requests und Responses zwischen Client und Server nach dem REST-Prinzip und ihre Abbildung auf Queries mit RestXQ (Blatt 5). Insbesondere darf

der Client zu jedem Zeitpunkt des Spiels nur die Interaktionsmöglichkeiten anbieten, die gerade zulässig sind.

Im Sinne einer modell-getriebenen Entwicklung soll Ihre Implementierung systematisch aus ihrem Design entwickelt werden. Hinweise dazu finden Sie auf den Aufgabenblättern.
Dokumentieren Sie Ihren Entwurf und zentrale Aspekte (Designentscheidungen) Ihrer Implementierung. Diskutieren Sie insbesondere:

- Was ist die Architektur Ihrer Lösung? Welche Komponenten gibt es? Wie verhält sich Ihre Lösung zum MVC-Architekturstil?
- Wie parametrisieren Sie den SVG-Code Ihrer Spielkarten. Wie stellen Sie Konsistenz und Redundanzfreiheit her?
- Wie gehen Sie mit dem Update Constraint von XQuery um?

Beachten Sie bei der Implementierung die oben angegebenen Regeln zur Namensgebung und dokumentieren Sie sie.
Die Dokumentation soll eine Installationsanleitung enthalten und u.a. angeben, welche Datenbanken in BaseX angelegt werden sollen und über welche URL Ihr Spiel gestartet werden kann. Sie können auch eine Setup-Methode definieren, mit der Ihr System vom Browser aus installiert werden kann.
Die Dokumentation soll beschreiben, welche Abstriche Sie eventuell am Funktionsumfang vorgenommen haben. Solche Abstriche führen natürlich zu einem Punktabzug in der Abschlussnote.
Die Dokumentation soll darüber hinaus eine kurze Reflektion zum Praktikum beinhalten, zur Organisation Ihrer Arbeit im Team sowie zur Thematik, Organisation und Betreuung im Praktikum.
Geben Sie an, wer in Ihrem Team für welche Artefakte (Modelle, Code-Bestandteile, Abschnitte in der Dokumentation) hauptverantwortlich und mitverantwortlich war.
Die Dokumentation soll mit DocBook erfolgen. Es soll eine HTML- oder eine PDF-Fassung der Dokumentation mit abgegeben werden.

Testatabnahme

Jede Gruppe bekommt einen eigenen Testat-Termin von 60 Minuten. Sie können sich schon vor Weihnachten in eine Liste von möglichen Testatterminen eintragen. Die Einträgungen können geändert werden. Im Zweifelsfall fragen Sie die Dozentin per E-Mail nach weiteren Optionen. Geben Sie dabei zwei bis drei Halbtage an, an denen Sie als Gruppe für das Testat zur Verfügung stehen.
Jede Gruppe hat 40 Minuten Zeit, um ihren Entwurf zu Blackjack zu präsentieren und ihre Lösung zu demonstrieren. Die Präsentation soll sich an der Dokumentation orientieren (konzeptuellen Ebene: Modelle, Architektur, Designentscheidungen) und kann einzelne Punkte mit Code-Ausschnitten demonstrieren. Ein kleiner Teil der Präsentation ist eine Demo Ihres Blackjack-Spiels.
An der Präsentation müssen sich alle Gruppenmitglieder in angemessenem Umfang beteiligen.
Anschließend an die Präsentation findet eine ca. zehnminütige Befragung zu Ihrem Projekt und zu den verwendeten Technologien statt.
Die Benotung der Praktikumsleistung basiert auf der Abschlusspräsentation und den in der Erarbeitungsphase erworbenen Bonuspunkten.
Zwei Tage vor der Testat-Abnahme geben Sie das vollständige System inklusive der Dokumentation im DocBook- oder im PDF-Format sowie die Folien Ihrer Präsentation auf einem geeigneten Datenträger oder per E-Mail ab.

Leistungsstand

Die folgenden Lernziele sollten Sie im Praktikum erreicht haben:

- Verständnis für den Hintergrund von XML.
- Beherrschen der Sprache XML: Instanzen mit Elementen, Attributen, Referenzen und Text; Definition von Entitäten in DTD.
- Parsen von XML-Dokumenten
- Visualisieren von XML-Dokumenten in Web-Browsern mit CSS.
- Einsatz von Kodierungsformaten in XML.
- Umgang mit DocBook
- Praktische Verwendung von XML Namespaces, speziell für standardisierte Anwendungen wie DocBook und SVG.
- Entwurf, Strukturierung und Implementierung von SVG-Graphiken.
- Praktische Verwendung von SVG.
- Praktische Verwendung von XLink im Kontext von SVG.
- Praktische Verwendung von SVG im Kontext von XHTML.
- Weitere Vertrautheit mit XML DTD; Modellierung von XML-Anwendungen; Deklaration von Elementen und Attributen.
- Verständnis für die Bedeutung von XPath, XSLT und anderen Standards.
- Praktische Verwendung von XPath zur Lokalisierung von Knoten in XML-Dokumenten und zur Berechnung von Daten.
- Praktische Verwendung der Pull- und Push-Prinzipien von XSLT.
- Fähigkeit, XSLT-Programme zu parametrisieren.
- Entwicklung modularer und konfigurierbarer XSLT-Programme.
- Formulierung von XQuery-Abfragen und XQuery-Updates.
- Definition und Aufruf von XQuery-Funktionen in Modulen.
- Benutzen von einer XML-Datenbank (BaseX) als XQuery-Prozessor.
- Umsetzung von Objektmethoden mit XQuery und XQuery Update Facility auf XML-kodierten Daten.
- Annotierung von XQuery-Methoden mit RestXQ.
- Umgang mit dem XQuery Update Constraint.
- Abbilden von HTTP-Requests und -Responses auf XQuery-Funktionen mit RestXQ.
- Testaufrufe von XQuery-Methoden mit RestXQ-Annotationen.
- Architektur von Web-Anwendungen im XStack
- Verständnis für die Motivation für Multi-Client-Web-Anwendungen und Server Push.
- Annotieren von WebSocket Funktionen.
- Umgang, Anwendung und Eigenschaften des WebSocket Elements.

Bewertungsschema Praktikum XML Technologie

Anne Brüggemann-Klein, Philipp Ulrich

1. Präsentation: 10 LP (Leistungspunkte)
   a. Inhalt: Die Präsentation gibt einen guten Einblick in das Projekt
   b. Struktur: Die Präsentation folgt einem roten Faden (kurzer Abriss der Dokumentation, s.u.)
   c. Beteiligung: Alle Gruppenmitglieder beteiligen sich in angemessenen Umfang an der Präsentation
2. Projekt
   a. Design, UI und Architektur: 30 LP
      i. Klare Darstellung der Funktionalität von Lobby und Spiel (s. zur Lounge auch Blatt 2)
      ii. Umfassende Modellierung von Daten, Screens, Interaktionsmöglichkeiten, Funktionalitäten, Abläufen mit Skizzen, UML-Klassendiagrammen, anderen UML-Diagrammtypen, Tabellen u.ä.
      iii. Angemessene Parametrisierung, z.B. Anzahl der Spieler*innen, Grafiken
      iv. Klare Benutzerführungen und Grafiken, ggf. Animationen
      v. MVC-Architektur wurde erläutert und sinnvoll eingesetzt
      vi. Client-Server Kommunikation nach REST Prinzip mit RESTXQ Annotationen
   b. Funktionalität: 40 LP
      i. Allgemeines
         1. Alle geforderten Funktionalitäten wurden umgesetzt
         2. Spiel ist multi-client-fähig
         3. Zusatzfunktionen und Besonderheiten des Spiels / der Lobby
      ii. Lounge
         1. Anmeldung von Spieler*innen ✓
         2. Umsetzung der High-Score-Liste ✓
      iii. Spiel
         1. Parametrisierung des Spiels umgesetzt: z.B. Anzahl der Spieler*innen, Grafiken
         2. Mehrere Spiele gleichzeitig spielbar (nur in Single-Client-Variante gefordert, bei Multi-Client-Variante Bonus)
   c. Implementierung und Code Qualität: 20 LP
      i. Allgemeines: Daten und Code aus Design systematisch abgeleitet (Idee des Model-Driven Design)
      ii. XQuery
         1. Sinnvolle funktionale Zerlegung
         2. Ggf. Aufteilung in Module
         3. Umgang mit XQuery Update Constraints: Problemstellung und Darstellung der Lösung
      iii. XSLT und SVG
         1. SVG-Graphiken selbst definiert, aus Skizze systematisch abgeleitet
         2. Konsistenz und Redundanzfreiheit im SVG-Code
         3. Parametrisierung der Karten, des Spielbretts und der Kartenanzahl
      iv. Code ist dokumentiert

v. Sinnvoller Einsatz der im Praktikum vorgestellten XML-Technologien, kein Einsatz anderer Technologien wie JavaScript
vi. Gruppen-Präfixe verwendet, um Projekt parallel zu anderen Lösungen installierbar zu machen (siehe Projektaufgabe)

3. Dokumentation: 10 LP
   a. Klarer, gut strukturierter Überblick über das Projekt (NUR über das Endprodukt, nicht über die Geschichte der Übungsblätter)
   b. In der Projektbeschreibung geforderte Punkte sind dokumentiert
   c. Reflektion ist vorhanden und nachvollziehbar
   d. Muss im DocBook-Format verfasst werden (Zielformate HTML oder PDF müssen zusätzlich zum Quellformat abgegeben werden)
   e. Angemessener Umfang
   f. Installationsanleitung (Readme) ist vorhanden

ToDo:

- Buttons mit Funktionen verlinken (Hit + Stand) ✓
(- DTD)
- Chips als Buttons implementieren
  (+ Reset-Button, Confirm) ✓
- HighScore: Endspielstand ✓
- Spiel beenden: alle pleite/verlassen
(- Nach jeder Runde HighScore speichern?) ✓
- Lobby: Spiel beitreten/anzeigen
- HighScoreboard: Position (XQuery-Funktion) ✓
- Doku + Präsentation
- Gewinn nach Runde auszahlen!
- Dealer-Phase implementieren!
- Name + Punktezahl anzeigen (Karten + Konto)!
- Multi-Client: Umstellung auf WebSockets!
- Spieler ausgrauen/ausblenden falls ausgeschieden
- Mehrere Spiele gleichzeitig ✓
- Code auskommentieren + README