Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)
Fachbereich Informatik, Mathematik und Naturwissenschaften

Softwarepraktikum - Gruppe 3 - Wepromato

Nachreichung zum Pflichtenheft

Nichtfunktionale Anforderungen

Leipzig, April 2007

Vorgelegt von:
- Yves Adler (05IN)
- Marco Aust (05IN)
- Franz-Sebastian Lorenz (04IN)
- Katharina Schewzow (05IN)
- Manuel Unglaub (05IN)
- Nico Wagner (05IN)

Unterstützt von:
- Matthias Jauernig (06INM)

Lehrveranstaltung: Softwarepraktikum / Projektmanagement-Praktikum
Verantwortlicher Professor: Prof. Dr. rer.nat. Karsten Weicker

Inhaltsverzeichnis

1 Sicherheit

Die Sicherheit ist bei diesem Projekt ein wichtiges Thema, da sie zum einen die Stabilität des Systems erheblich erhöht und zum anderen dem Schutz der Projektdaten dient. Es ist wichtig sicher zu stellen, dass Unbefugte keinen Zugang zum System und den dort gehaltenen Daten haben.

1.1 HTTPS - Verbindung

Da der verwendete Apache Tomcat Server HTTPS - Verbindungen unterstützt, wollen wir diese auch nutzen. Insbesondere für die Authentifizierung der Nutzer.

1.2 Passwortsicherheit

Um sicher zu stellen, dass Unbefugte keinen Zutritt zum System erhalten ist es wichtig, dass alle Nutzer ein sicheres Passwort besitzen. Daher kontrolliert das System, ob die Passwörter folgende Bestimmungen erfüllen:

- Ein Passwort muss mindestens aus 8 Zeichen bestehen.
- In einem Passwort müssen Buchstaben und Zahlen enthalten sein.

Um das Ausspähen der Passwörter aus der Datenbank zu unterbinden, werden wir nur die SHA-1 - Hashes der Passwörter in der Datenbank ablegen. Die SHA-1 - Hashes werden wir mittels des Java Security - Packages berechnen.

1.3 SQL - Injektion

Als SQL - Injektion bezeichnet man das Ausnutzen einer Sicherheitslücke in SQL - Datenbanken. Angreifer versuchen über die Anwendung, die den Zugriff auf die Datenbank bereitstellt, eigene SQL - Befehle einzuschleusen um Zugriff auf die Daten zu bekommen, diese zu verändern oder gar die Kontrolle über den Server zu erhalten.

Um SQL - Injektions zu verhindern werden wir die Möglichkeiten von Java nuzten, insbesondere indem die PrepardStatement - Klasse verwendet wird. Die PreparedStatement - Klasse repräsentiert ein precompiliertes SQL Statement. Das SQL Statement wird in einem PreparedStatement Objekt gespeichert und kann so sicher ausgeführt werden. Mit den verschiedenen Set-Methoden werden die zu übergebenden Typen sicher festgelegt, so dass SQL - Injektions unterbunden werden.

1.4 Zugriffsschutz

Der Zugriffsschutz auf einzelne Funktionen und Daten wird mit Hilfe der Rollenverteilung realisiert. Es gibt folgende Rollen

- Administrator
- Betreuer
- Projektleiter
- Qualitätssicherer
- Teammitglied

2 Qualitätsanforderungen

2.1 Wiederverwendbarkeit

Wir haben uns entschieden, zur Enwicklung der Software die Methoden der Interface - basierenden Softwareentwicklung zu verwenden. Dadurch entstehen gut gekapselte Objekte, die einen hohen Grad der Wiederverwendung haben.

2.2 Portierbarkeit

Das System ist prinzipiell nicht an eine bestimmte Plattform gebunden und im Benutzerhandbuch sind Installationsanleitungen für verschiedene Plattformen enthalten.

2.2.1 Server

Da die Anwenung mit Java geschrieben wird und auf dem Apache Tomcat läuft, kann das System unter allen Plattformen verwendet werden, für die folgende Komponenten verfügbar sind:

- Java SE 6 SDK
- Java EE 5 SDK
- Apache Tomcat 6.x
- MySQL - Server 5.x

2.2.2 Client

Auf der Seite der Benutzer des Systems wird nur vorausgesetzt, dass einer der folgenden Browser installiert ist:

- Browser mit der Gecko Engine (Netscape Navigator, SeaMonkey, Mozilla Firefox)
- Internet Explorer 5.0

2.3 Stabilität

Um die Stabilität der Anwendung sicher zu stellen, werden wir zum einen die oben genannten Sicherheitsaspekte beachten und umsetzen und zum anderen die Möglichkeiten von Java nutzen. Da Java besonders im Hinblick auf Sicherheit entwickelt wurde, gibt es effektive Methoden wie das Exception - Handling und den Verzicht auf Zeigerarithmetik, um die Stabilität der Anwendung zu gewährleisten.

2.3.1 Auftreten von Fehlfunktionen

Sollte tatsächlich eine Fehlfunktion auftreten, so ist nur ein erneutes Laden der im Browser angezeigten Seite notwendig. Wenn ein schwerwiegender Fehler auftritt, kann es notwendig sein, das sich der Nutzer neu am System anmelden muss. Bei Fehlern, die den Apache Tomcat oder die MySQL - Datenbank betreffen, ist unter Umständen ein Neustart dieser Dienste nötig. Die Zeit zum Neustarten von MySQL beziehungsweise Apache Tomcat liegt unter zwei Minuten.

2.3.2 Datenverlust

Wenn während der Eingabe von Daten eine Fehlfunktion auftritt, gehen nur die noch nicht in die Datenbank übertragenen Daten verloren. Dabei wird es sich lediglich um gerade eingegebene Daten handeln, da jedes Datum bei Abschluss einer Aktion sofort in die Datenbank geschrieben wird. Der Verlust von Daten, die in der Datenbank gespeichert sind, ist nur zu befürchten, wenn Unbefugte, unter Umgehung des Systems, Zugriff auf die MySQL - Datenbank haben.

2.4 Performance

Da es in der Anwendung keine aufwendigeren Berechnungen gibt, sollte das System performant genug sein, um dem Benutzer ohne nennenswerte Verzögerung ein Feedback auf seine ausgeführten Aktionen zu geben. Das System wird so konzipiert, dass ein Anstieg der Datenmenge zu einem nahezu linearen Verhalten bezüglich des Speicherverbrauchs und der Rechenzeit führt.

2.5 Benutzerfreundlichkeit

Obwohl die Benuzter der Software über gute bis sehr gute Computerkenntnisse verfügen werden, legen wir viel Wert auf ein intuitives und übersichtliches Design. Um dies zu erreichen, werden wir uns auf eine einheitliche Benutzerführung festlegen, wodurch es den Benutzern des System in wenigen Schritten möglich ist eine Aufgabe erfolgreich zu lösen. Das System wird über eine grafische Benutzeroberfläche mit übersichtlicher Menüführung verfügen, die mit Hilfe eines Webbrowsers dargestellt wird.

2.5.1 Benutzerhandbuch

Zum System gehört ein umfassendes Benutzerhandbuch, das unter anderem eine Installationsanleitung für verschiedene Plattformen enthält. Um den Benutzern des Systems bei Fragen zur Seite zu stehen, wird das Handbuch um eine im System integrierte Hilfe - Funktionalität ergänzt.

2.5.2 Schulungsdauer

Nach einer einstündigen Schulung wird ein Benutzer das System mit einer maximalen durchschnittlichen Fehlerrate von 5 Fehlern pro Tag bedienen können.

2.6 Barrierefreiheit

Da Barrierefreiheit ein Thema mit immer größerer Bedeutung wird haben wir uns entschlossen das System möglichst barrierefrei zu gestalten. Um das zu erreichen, werden wir folgende Maßnahmen ergreifen:

- Die Schrift ist im Browser skalierbar um Sehschwachen zu ermöglichen die Schriftgröße an ihre Sehleistung anzupassen.
- Die aktiven Elemente werden in einer sinnvollen Reihenfolge angesteuert um das Navigieren mit der Tastatur zu vereinfachen.
- Die logische Struktur der Anwendung wird mit den dafür vorgesehenen Elementen realisiert.