

BENUTZER:

Wie viele User wird das System unterstützen ?

Werden Nutzer eigene Benutzerprofile haben ?

Werden Nutzer einen Pinnwand haben ?

Können User eigene Logos/Avatare haben ?

Werden User untereinander kommunizieren können ?

Können User ihre Ergebnisse auf Social-Networks wie Facebook, Twitter teilen ?

Können die Nutzer Dateien/Links anhängen bzw. teilen ?

Können mehrere Nutzer untereinander in einer Gruppen-Chat kommunizieren ?

Können die Nutzer Kommentare schreiben ?

Werden die Nutzer per E-Mail benachrichtigt, wenn sie eine PM erhalten ?

Können die Nutzer ihre Statistiken abrufen bzw. teilen ?

Müssen sich die Nutzer für bestimmte Kurse einschreiben oder können sie die Kurse willkürlich angehen ?

SYSTEM:

Wird das System verschiedene Kurse/Fächer anbieten ?

Wird das System privat sein ?

Wird das Interface customizable sein ?

Wird das System dynamische Tutorials haben ?

Können zur Interface Widgets hinzugefügt werden ?

Welche Softwarearchitektur soll verwendet werden ?

Ist eine Datenbank einzusetzen ?

Welche Software-Qualitätssicherungsmaßnahmen sind einzusetzen ?

Wird es einen Diskussionsforum geben ?

Können dritte auf das System zugreifen und ihre eigene Kurse anbieten ?

Wird Gamification unterstützt ?

Wird das System einen Dashboard haben ?

Können die User das System testen, bevor sie sich registrieren ?

Gibt es eine Suchfunktion für die Kurse ?

Gibt es eine Suchfunktion für die Benutzer ?

Gibt es eine Funktion für die Benutzer, wo sie ihre bisherigen Aktivitäten aufrufen können ?

Wie viele bzw. welche Sprachen werden unterstützt ?

Soll das System Geräteunabhängig sein ?

Wird das System einen Blog haben ?

Wer ist die Zielgruppe ?

Muss die Software mit einer anderen Software kommunizieren ?

Ist das System wartbar ?

Ist das System testbar ?

Ist das System erweiterbar ?

Enthält das System uservice ?

Enthält das System Counter ?

Wird das System einen Voting haben, wo die Nutzer über neue Features abstimmen können ?

Wie wird das Feedback verwaltet ?

CODE:

Wie sind die Klassen aufgeteilt ?

Wie sehen die Methoden aus ?

Wie sind die Abhängigkeiten zwischen verschiedenen Code-Teilen ?

Welche Pattern werden benötigt ?

GLOSSAR:

USERVOICE: Nutzer können Feedback zu dem System geben

AVATAR: Ist eine Grafikfigur, die dem Benutzer zugeordnet wird

COUNTER: Funktionalität, womit man feststellen kann, welche Features am meisten aufgerufen werden

PM: Private Message

PATTERN: Sind Schablonen für wiederkehrende Entwurfsprobleme

DASHBOARD: Visualisierung von Informationen in grafischer Form

DYNAMISCHE TUTORIALS: Nutzer werden beim klicken eines Buttons informiert, was diese denn auslöst

SOFTWARE-QUALITÄTSSICHERUNGSMABNAHMEN: Hierzu zählen; Funktionalität, Zuverlässigkeit, Knappheit, Benutzbarkeit usw.

WARTBAR: Wenn es möglich ist im Nachhinein die Software zu ändern bzw. verbessern.

TESTBAR: Wenn es möglich ist, die Software auf mögliche Fehler zu testen, bevor sie released wird

ERWEITERBAR: Wenn es möglich ist die Software im Nachhinein zu updaten

GAMIFICATION: Wenn spielerische Elemente wie Highscores, Ranglists, Experience Points in einem System enthalten sind

BLOG: Journal in dem Sachverhalte gepostet werden

FORUM: Dient zum Austausch von Meinungen

PINNWAND: Hier kann der User Informationen über sich bekanntgeben