

In dieser Übung betrachten wir die Anforderungen der Software *Sidequest*. Das Ziel von *Sidequest* ist es Menschen zu helfen ihre Aufgaben zu organisieren.

1 Team Setup

Please group yourself in teams of 3-4 persons. Agree on a team name.

Team Name: The Hardcoders

Team Mitglieder:

1. Marco Bencic (3005314)
2. Benjamin Klett (3003345)
3. Fabian Strauß (3003662)

2 Probleme

3 Fragen

- Wie soll das System mit bestehender oder fehlender Onlineverbindung umgehen (Benji Klett)?
- Soll jeder *Plugins* zur Verfügung stellen können oder sollen die verfügbaren *Plugins* reguliert werden? (Marco Bencic)
- Soll die Anzahl der *Benutzer*, die die *Subquests* eines *Quests* erledigen auf ein Maximum beschränkt werden ?

4 Vision

Sidequest unterstützt Individuen und Gruppen wie z.B. Familien ihre Alltags- und Verwaltungsaufgaben im Griff zu behalten. Durch Künstliche Intelligenz lernt das System , welche Schritte zur Erreichung eines Ziels erforderlich sind und verfeinert Aufgaben (genannt *Quests*) automatisch zu Schritten (genannt *Subquests*). *Sidequest* kann erweitert werden, um nach und nach immer mehr Verwaltungsaufgaben automatisch zu übernehmen.

5 Ziele

Sidequest ist ein Mehrbenutzer System, dass es erlaubt, Aufgaben anzulegen, zu Unteraufgaben zu verfeinern und den Status der Aufgaben zu verfolgen. Es unterstützt, Teilaufgaben einer Aufgabe an andere Personen zu delegieren.

Sidequest unterstützt die Benutzer unterwegs und auf der Arbeit.

Sidequest erkennt, wenn Nutzer wiederkehrend semantisch ähnliche Aufgaben anlegen, und nutzt dies, um automatisch Subquests vorzuschlagen. Diese Erkennung funktioniert sowohl Nutzerbezogen, d.h. das System erkennt, dass ein Nutzer Aufgaben immer auf die gleiche Art erledigt, als auch über Nutzer hinweg, d.h. das System kann eine Aufgabe zerlegen, weil andere Nutzer sie bereits sinnvoll zerlegt haben.

Sidequest verfügt über ein Plugin-System, welches Entwicklern die Möglichkeit gibt, den Funktionsumfang zu erweitern. (Marco Bencic)

6 Functional Requirements

Systemqualität	sehr gut	gut	normal	nicht relevant
Funktionalität		x		
Zuverlässigkeit			x	
Benutzbarkeit	x			
Effizienz			x	
Wartbarkeit		x		
Portabilität	x			

6.1 Dictionary

Begriff: Benutzer

Bedeutung: Account eines Systembenutzers, der von mehreren Endgeräten aus benutzt werden kann.

Gültigkeit: Ein *Benutzer* existieren von der Registrierung (R X) eines Benutzers bis zu seiner Löschung (R Y).

Identifikation: EMail-Adresse, nur Sanity-Check (Eingabestring entspricht EMail-Format). Keine Überprüfung der Existenz eines passenden EMail-Kontos.

Begriff: Benutzereinstellungen

Bedeutung: Konfigurierbar ist: Zeitraum der automatischen Löschung, ...

Gültigkeit: Die Benutzereinstellungen existieren von der Anlage eines Benutzers bis zu seiner Löschung.

Identifikation: Ist 1:1 mit dem Benutzer verknüpft.

Querverweise: *Benutzer*

Begriff: Quest

Bedeutung: Eine *Quest* repräsentiert eine Aufgabe, die ein *Benutzer* lösen möchte. Es gibt für jede *Quest* einen Besitzer (Standart: Ersteller) und einen Bearbeiter (Standart: Besitzer).

Abgrenzung: Quest Template

Gültigkeit: Eine *Quest* existiert von ihrer Anlage bis zu ihrer Löschung.

Identifikation: Eine *Quest* wird über eine interne ID identifiziert.

Querverweise: *Hauptquest, Sidequest*

Begriff: Hauptquest

Bedeutung: Eine *Hauptquest* ist eine *Quest* ohne übergeordnete *Quest*.

Begriff: Besitzer

Bedeutung: Der *Besitzer* einer *Quest* ist ein *Benutzer*, der für eine *Quest* verantwortlich ist und Rechte an dieser in vollem Umfang hat. Es gibt für jede (*Sub*-)*Quest* nur einen Besitzer. Dieser kann den *Bearbeiter* abändern bzw. neu zuteilen durch Anfragen.

Abgrenzung: Bearbeiter

Gültigkeit: Ein *Besitzer* existiert für jede (*Sub*-)*Quest* von Anfang bis zur Löschung dieser.

Identifikation: Der *Besitzer* ist einer *Quest* fest zugeordnet.

Querverweise: *Bearbeiter*

Begriff: Bearbeiter

Bedeutung: Der *Bearbeiter* einer *Quest* ist ein *Benutzer*, der berechtigt ist, eine (*Sub*-)*Quest* zu bearbeiten. Es kann mehrere Bearbeiter für eine (*Sub*-)*Quest* geben. Diese haben keine Übertragungsrechte, eine Anfrage zum Erlangen des Bearbeiterstatus kann nur vom *Besitzer* angenommen oder abgelehnt werden.

Abgrenzung: Besitzer

Gültigkeit: Ein *Bearbeiter* existiert für jede (*Sub-*)*Quest* von Anfang bis zur Löschung dieser .

Identifikation: Der *Bearbeiter* ist einer *Quest* fest zugeordnet.

Querverweise: *Besitzer*

Begriff: Gruppe

Bedeutung: Eine *Gruppe* ist ein spezieller *Benutzer*, der stellvertretend für eine Menge konkreter *Benutzer* sowohl *Besitzer* als auch *Bearbeiter* sein kann. Anstelle dieses Gruppenbenutzers kann jedes Gruppenmitglied gleichermaßen *Besitzer*- u. *Bearbeiter* verändern und (*Sub-*)*Quests*, die speziell für diese *Gruppe* erstellt wurden, an Nichtmitglieder delegieren.

Abgrenzung: Benutzer

Gültigkeit: Eine *Gruppe* existiert von der Erstellung bis zur Löschung dieser.

Identifikation: Die *Gruppe* ist ein spezieller *Benutzer* und wird daher auch durch eine interne ID identifiziert.

Querverweise: *Benutzer*

Begriff: Hierarchisches Quest

Bedeutung: Eine *hierarchisches Quest* ist ein *Quest*, das *Subquests* besitzt, die in einer bestimmten Reihenfolge abgearbeitet werden müssen.

Abgrenzung: Quest

Identifikation: Eine *hierarchisches Quest* wird über eine interne ID identifiziert.

Querverweise: *Hauptquest*, *Sidequest*

Begriff: Plugin

Bedeutung: Ein *Plugin* ist eine nachträglich installierte Software, welche den Funktionsumfang von *Sidequest* erweitert.

Abgrenzung: Standard Optionen

Gültigkeit: Damit ein *Plugin* funktioniert, muss dieses installiert und aktiviert werden

Identifikation: Eine *Plugin* wird über seinen Herausgeber identifiziert

Begriff: Subquest

Bedeutung: Eine *Subquest* ist eine *Quest*, die einer anderen *Quest* untergeordnet ist. Um eine *Quest* zu erfüllen, müssen zunächst alle *Subquests* erfüllt werden.

Begriff: Quest Template

Bedeutung: Ein *Quest Template* repräsentiert ein wiederkehrendes Ziel, dass *Benutzer* lösen möchten. Durch *Quest Templates* können *Quests* automatisch mit *Subquests* befüllt werden.

Abgrenzung: Quest

Gültigkeit: Ein *Quest Template* existiert als Programmbestandteil ständig.

Identifikation: Eine *Quest Template* wird über eine interne ID identifiziert.

Querverweise: *Quest*, *Hauptquest*, *Sidequest*

6.2 Requirements

- R1 Ändern von Zuständigkeiten für *Quests* (Fabian Strauß)
 - R1.1 Jede *Quest* bzw. *Subquest* hat sowohl *Besitzer* als auch *Bearbeiter*, beide können unabhängig von einander an andere Nutzer übertragen werden.
 - R1.2 Die Übertragung eine (*Sub-*)*Quest* kann sowohl beim aktuellen *Besitzer* als auch beim *Bearbeiter* angefragt werden, diese können die Anfrage annehmen oder ablehnen (User werden

bekannt durch peröhnliches Austauschen der Profildaten oder durch Mitgliedschaft in einer gemeinsamen Gruppe).

- R1.3 Der aktuelle Besitzer/Bearbeiter kann auch einen anderen User um Übernahme einer (*Sub-*)*Quest* bitten, welche diese annehmen können.
- R2 Löschen von *Quests* (Datenschutzbeauftragter)
 - R 2.1 *Benutzer* können *Quests* manuell löschen, und zwar einzeln (unabhängig vom Status) oder in ganzen Zeiträumen (nur abgeschlossene *Quests*). (Tom Gallo, Josa Kirscher)
 - R 2.2 Abgeschlossene *Quests* werden automatisch nach einem bestimmten Zeitraum gelöscht, wenn dies in den *Benutzereinstellungen* so konfiguriert ist. (Tom Gallo, Josa Kirscher)
- *Benutzer* können einen vollständigen Verlauf ihrer *Quests* anschauen (Tim Dahmen)
- *Benutzer* können *Quests* manuell anlegen. (Tim Dahmen)
- Die manuelle Anlage von *Quests* und *Subquests* soll so einfach und schnell wie möglich erfolgen. (Tim Dahmen)
- *Quests* können verschiedene Status haben. (Tim Dahmen)
- R3 Benutzerverwaltung (Datenschutzbeauftragter) (Marco Bencic)
 - R 3.1 *Benutzer* müssen mit Email Adresse und Namen angelegt werden können.
 - R 3.2 *Benutzer* müssen gelöscht werden können
 - R 3.3 Gelöschten *Benutzern* zugewiesene *Quests* müssen an einen anderen noch existierenden *Benutzer* übertragen werden können. Dieser muss solche *Quests* annehmen können, andernfalls werden sie gelöscht.
- R4 Installation/Deinstallation von Plugins (IT-Abteilung) (Marco Bencic)
 - R 4.1 *Plugins* müssen direkt in *Sidequest* installiert und deinstalliert werden können
 - R 4.2 Es muss eine Option geben, in der man aktiv zustimmen muss, dass man *Plugins* verwenden will
 - R 4.3 Bei Neuinstallation und beim Erscheinen neuer Plugins soll eine Popup-Nachricht beim Öffnen der App den *Benutzer* darauf hinweisen, dass diese existieren und zur Installation bereit stehen. (Fabian Strauß)
- R5 Subquests
 - R 5.1 *Benutzer* können *Quests* manuell in Subquests zerlegen. (Tim Dahmen)
 - R 5.2 *Subquests* können wiederum in weitere *Subquests* zerlegt werden (nested *Subquests*) (Marco Bencic)
- R6 Questvergabe (Marco Bencic)
 - R 6.1 *Quests* müssen verschiedenen *Benutzern* zugewiesen werden können
 - R 6.2 *Subquests* können auf mehrere Benutzer aufgeteilt werden sofern diese nicht Teil eines *hierarchischen Quests* sind
- R7 Benutzereinstellungen (Datenschutzbeauftragte) (Benji Klett)
 - R 7.1 Der Name und die E-Mailadresse eines *Benutzers* müssen sich ändern lassen
 - R 7.2 Ein *Benutzer* kann die Einwilligung der Datenverarbeitung durch KI zurückziehen
- R8 Anlage von *Gruppen* zur gemeinsamen Bearbeitung von *Quests* (Fabian Strauß)
 - * R1.1 Jeder *Benutzer* soll *Gruppen* anlegen können. (*Sub-*)*Quests* können dann speziell für diese *Gruppe* zur Bearbeitung erstellt werden.
 - * R1.2 Im Gegensatz zu normalen (*Sub-*)*Quests* sollen alle Gruppenmitglieder Besitzerrechte ausüben und delegieren können, *Besitzer* bleibt jedoch bei Gruppenquests immer die *Gruppe*. Die Bearbeitung kann auch an Nichtmitglieder delegiert werden.

- R9 Belohnungssystem (Fabian Strauß)
 - * R 9.1 *Quests* bzw. *Subquests* müssen Belohnungspunkte zugewiesen werden können, auf dessen Basis die App solche Punkte für andere *Benutzer* bei Erledigung vergibt. Die KI-Unterstützung soll eine Klassifizierung der (Unter-)Aufgaben vornehmen, um die automatische Punktevergabe zu optimieren.
 - * R 9.2 *Benutzer* sollen zusätzlich progressiv steigende Boni auf erledigte Aufgaben bekommen können, wenn diese zb. täglich erledigt werden mit Beginn von vorne, falls ein Benutzer die Kette unterbricht.
 - * R 9.3 Ein für jeden *Benutzer* sichtbares Ranking soll existieren, dass die *Benutzer* mit den höchsten Punktzahlen dort auflistet.

6.3 Use-Cases

Numbers: SQ-3.1

Name: Quest abschließen

Actors: Besitzer der Quest, Bearbeiter der letzten Subquest

Trigger: Die letzte unbeendete Subquest wird beendet

Preconditions: Der Status von mindestens einer Subquest ist nicht abgeschlossen

Postconditions / Goal: Der Status aller Subquests und damit der Status der Quest ist abgeschlossen, die Quest steht nicht mehr zur Bearbeitung bereit (Der Status kann nicht mehr geändert werden).

Postconditions Special Cases: Besitzer akzeptiert die erfolgreiche Bearbeitung eines Bearbeiters einer Subquest nicht, Status der Quest und der Subquest stehen dann immer noch auf offen.

Steps Default Case:

1. Bearbeiter letzter offener Subquest setzt deren Status auf abgeschlossen
2. Besitzer bekommt eine Nachricht über die Statusänderung
3. Besitzer bestätigt die vollständige Bearbeitung der Subquest
4. Die Subquest und die gesamte Quest hat jetzt den Status abgeschlossen
5. Beteiligte Benutzer bekommen eine Nachricht, die auf die erfolgreiche Bearbeitung der Quest hinweist.

Special Cases:

3a Besitzer lehnt die erfolgreiche Bearbeitung der letzten offenen

3a1 Die Anfrage wird auf abgelehnt gesetzt.

3a2 Der antragstellende Bearbeiter bekommt eine Mitteilung über die Ablehnung seiner Anfrage. Subquest ab.

Numbers: SQ-3.2

Name: Benutzer registrieren

Actors: Anwender

Trigger: Anwender installiert die App, öffnet diese und fordert eine Registrierung an.

Preconditions: Das gewünschte Benutzerkonto existiert nicht im System.

Postconditions / Goal: Ein Benutzerkonto existiert für den Anwender im System.

Postconditions Special Cases: Es wurde ein neues Konto erstellt

Steps Default Case:

1. Anwender fordert eine Registrierung im System an
2. System zeigt Eingabemaske für die erforderlichen Profilingaben
3. Anwender gibt Benutzernamen, E-mail-Adresse und optional einen Vor- bzw. einen Nachnamen ein und bestätigt die Eingabe.
4. Ein Hinweis über die erfolgreiche Erstellung des Kontos wird angezeigt und das Hauptmenü wird angezeigt, welches sich im in das Konto eingeloggt Zustand befindet.

Special Cases:

3a Benutzername oder E-Mail existieren bereits, E-Mail besteht den Sanity-Check nicht.

3a1 Anwender bekommt entsprechende Fehlermeldung.

3a2 Bestätigung bleibt aus, Anwender verbleibt in der Registrierungsansicht. Subquest ab.

7 Non-Functional Requirements