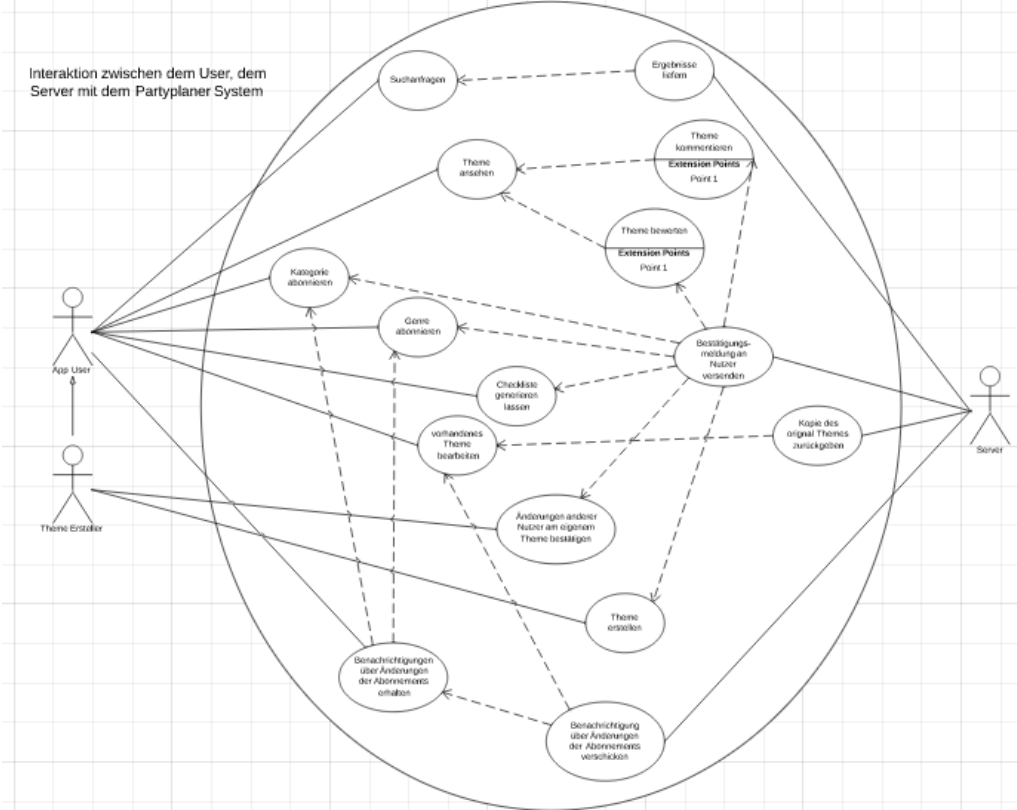


Logbuch

Datum	Gedanken
15.04.13	Idee: Partyplaner, weiterer möglicher Content: Einladungskarten, Menükarten, do-it-yourself Anleitungen, Ausstattung (Anlage, ...), Spiele/ Unterhaltung Irgendwo explizit Vorteile der Anwendung auflisten? Z.B. Zeitersparnis der Recherche, große Community, ... Ausblick: Timeline für die Planung integrieren, Kostencheckliste
17.04.13	Szanario Überarbeitung
22.04.13	Ist die Angabe des Autors beim Theme wichtig? -> eher nicht, da mehrere User daran arbeiten können und evtl. auch mehr daran arbeiten können als der Hersteller selbst. Desweiteren können so viele Versionen von einem Theme erstellt werden, dass die Angabe irrelevant ist. Wann sollen wir all und wann sequence für das Schema verwenden?
23.04.13	Feedback der Mentoren: Interaktion allgemeiner erklären -> könnten wir durch ein Use-Case Diagramm! Die Gliederung der Doku sollte parallel erstellt werden + Stichpunkten für den Inhalt Frage fürs nächste Meeting: sollen wir uns pro Woche 2x treffen? (1x Ausarbeitung Praxis, 1x Dokumentation schreiben)
24.04.13	Erster Use-Case Diagramm Entwurf „Interaktion zwischen dem User, dem Server mit dem Partyplaner System“ 
23.04.13	Dokumentation Gliederung + Stichpunkte: Konzept, Idee, Interaktion, Szanario + Feedbacks
28.04.13	Vorbereitung auf das Feedbackgespräch: <ul style="list-style-type: none"> • Umfang des XSD Schemas/ Genauigkeit? • Aufgabenabgrenzung zum nächsten Meilenstein?

29.04.13	<ul style="list-style-type: none"> Überprüfung des Use-Case Diagramm <p>Im Beratungs-/ Praktikumstermin:</p> <ul style="list-style-type: none"> XSD-Schema zu Ende geführt inkl. Attributen der Elemente Schema auf sinnvolle Semantik überprüft Ausformulierung der Dokumentation (Idee, Szenario, Interaktion) Umsetzung des Feedbacks bzgl. des XSD <p>Privat</p> <ul style="list-style-type: none"> Use-case: vereinfacht und Aktionen zusammengefasst für die Übersichtlichkeit
30.04.13	Tabelle für Ressourcen und Methoden anhand Beispiel für 2. Teil-Aufgabe REST&http Buch S.21 erstellt
01.05.13	Buch gelesen und bemerkt: Theme.xsd muss geändert werden: es muss ein Container-Element um das theme-Element drum rum, damit wir eine Liste der Themes haben können theme maxOccurs=unbounded etc.
05.05.13	<ul style="list-style-type: none"> URI Design, Näheres über Methoden Recherche im REST-Buch und über die Notwendigkeit der HEAD & OPTIONS Methode in unserer App nachgedacht -> nicht notwendig, dies alles in Doku geschrieben
06.05.13	Meilensteintermin 1+2: nach Feedback Ressourcen und http-Methoden ausgebessert
07.05.13	<ul style="list-style-type: none"> Jersey/ Grizzley Codebeispiel implementiert Xml Dateien aus vorhandenen Schemata erstellt Neue xsd geschrieben, wo genre und kategorie in einem Schema sind und nicht getrennt in 2, denn: man müsste dann alle Kateogrien in die kategorien-xml datei, egal zu welchem Genre sie angehören Wieder verworfen, da die XML datei zu unübersichtlich wird und alte Idee wieder aufgegriffen: es wird 1 genre.xml geben, die alle genres mit deren id's enthält, dann wird es zu jedem genre eine eigene kategorie-{genre_id}.xml geben. Auf diese Weise sind sie doch geordnet und die xml/ xsd struturen bleiben einfach Xsd & xml aufwellformed und validität geprüft -> juhu geht Xsd mit xjc zu java klassen -> Fehler bei theme (Complex type and its child element share the same name "Beschreibung") -> korrigiert ObjectFactory für Genres und Kategorien nicht notwendig, da wir feste Werte vorgeben -> gelöscht
08.05.13	<ul style="list-style-type: none"> Server und Client implementiert URI Design nochmal geändert (Doku muss noch angepasst werden) URI der Kategorien sind jetzt doch hierarchisch von der Genre URI abhängig (leichter für identifikation der Kategorien id's, diese sind nämlich von den genre id's abhängig) Grundgerüst aller Dienste erstellt, alle Methoden-Köpfe
09.05.13	<ul style="list-style-type: none"> Hello world get erfolgreich Implementierung der GenresKategorienRessource (noch kein exception-handling, Response -> nur String Rückgabe) aber klappt alles! Juhu! Zeichenkodierung noch nicht ganz perfekt Aufgefallen: wir brauchen ja noch ein schema+xml Datei für users! Oder kann man die aus der themes.xsd datei exerpieren? -> ja erst mal so probieren -> geht irgendwie nicht... müssen mal am Montag nachfragen! Alle möglichen POST-Methoden probiert du schreiben -> gescheitert Die einzige DELETE Methode implementiert, Compiler muckt nicht, noch keine XML-Datei zum Testen vorhanden.
10.05.13	<ul style="list-style-type: none"> Javadoc in den code eingefügt Neuer Gedanke: kaskadierendes löschen muss noch implementiert werden:

	<p>wenn ein Genre/ eine Kategorie gelöscht wird, muss die Referenzierung darauf im Theme ebenfalls gelöscht werden.</p>
13.05.2013	<ul style="list-style-type: none"> • Alle Methoden implementiert(wenn auch noch nicht ganz funktional) • POST Methode mit Variante mit xml body Übergabe anstatt Queryparametern (klappt aber nicht ganz) • Exception-Handling • Abfragen, ob ID-Korrekt ist (im Wertebereich liegt) implementiert -> Ist dies bei uns überhaupt notwendig? Der Nutzer hat bei unserer app eh keinen Zugriff drauf/ ist nicht für ihn sichtbar. Sicherheitshalber trotzdem -> dazu habe ich die Überprüfung erst in jeder Methode lokal geprüft, nun aber eigene Hilfsfunktion Prüfe ID dazu • GET Themeliste hat keine ID Überprüfung da dies nach GET Theme weitergeleitet wird und da überprüft wird • Bei DELTE Theme ist dieses Problem bereits anders gelöst (Nachteil ist hier vllt Uneinheitlichkeit, aber nicht so wichtig, daher so lassen) • Änderung: gibGenres und ibKategorien müssen anstatt private public sein, damit Klasse Themes darauf zugreifen kann für die Überprüfung -> ok, nun sind die 4 geschachtelten for-Schleifen fertig, hab aber einen semantischen Fehler entdeckt: Genres können überprüft werden, ob diese existieren, aber Kategorien nicht, da die ID's der Kategorien genreübergreifend gleich sind... wir müssten die Schemen oder die XML-Dateien anpassen. Dieses Problem wird erst mal nach hinten verschoben.. • Prüfung ob Genre bzw. Kategorie Eingabe bei Theme erstellen, auch aus existierendem Pool nimmt. • PUT – implementiert: großes Problem ist aufgetreten: bisher haben wir die Elemente nie wirklich durch ihre ID aufgerufen, sondern die Eigenschaft ausgenutzt, dass der Index mit welcher gespeichert wurden gleich der ID war. Nun, wo wie ein altes Theme raus löschen müssen und die neue dafür einfügen taucht das Problem auf, dass der Index nun nicht mehr gleich der ID ist (bei add werden die Nodes ja immer hinten angefügt.). Anders beim Array haben wir ja nicht die Möglichkeit anzugeben an welcher Stelle der Knoten angefügt werden soll. • In der PUT wird die Genre/Kategorie Richtigkeit noch nicht überprüft • Wie soll man hier herausfinden, ob etwas bei der Änderung schief gelaufen ist? Es muss ja noch ein Response.status(4XX) zurückgegeben werden beim Fehlerfall, ... • Problem mit den setzten von Indizes und ID's lösen mit iteratoren?ne geht auch nicht -.-
19.05.13	<ul style="list-style-type: none"> • XMPP-Buch Kap.1,2,8 gelesen
20.05.13	<ul style="list-style-type: none"> • ThemesService GET funktioniert endlich!!! Irgendwie hat der xjc ein Object aus dem Theme-Titel gemacht, anstatt einen String..naja. • Reines post funktioniert – ohne Kontrolle ob Dateneingabe richtig ist. • Ebenso mit DELETE und PUT nun. -> alle Methoden funktionieren!
22.05.13	<ul style="list-style-type: none"> • Einrichtung Openfire + Smack
23.05.13	<ul style="list-style-type: none"> • ID-System korrigiert: ID nicht mehr über die Stellen geholt (da stellen und ID identisch waren -> inkonsistent), nun werden die elemente durch einen Vergleich der ID's geholt => alle Webservices angepasst. • Überprüfung auf Richtigkeit der Genres und Kategorien implementiert (ob die Eingaben der PUT/POST Methode der Themes korrekt ist): total komplizierte Schleifen, geht bestimmt effizienter. Kann bestimmt niemand außer mit

29.05.13	verstehen:D
	<ul style="list-style-type: none"> • Ok, ist doch noch irgendwie ein Fehler drin, bei Testdatensätzen wird noch ein falscher Rückgabewert zurückgegeben.
	<ul style="list-style-type: none"> • Feedbackgespräch • -> keine Collection-Nodes verwenden, brauchen wir nicht • -> Datenstruktur für Items überlegen
02.06.13 +letzten Tage	<ul style="list-style-type: none"> • XMPP Client gebaut • + Menü für die Konsole, zum Testen der Funktionen • Abonnieren funktioniert, Abonnement kündigen auch. • Publishen noch nicht (Listener und so sind aber schon halbwegs drin) • Exceptionhandling + Abfangen von Fehleingaben des Benutzers
	<ul style="list-style-type: none"> • Publishing reine Funktionalität, noch keine Fehler abfangen • Papierprototyp GUI (Theme Anzeigen/ Bearbeiten erst mal als Editor + XML Code) • Swing tutorials reingezogen
06.06.13	<ul style="list-style-type: none"> • Nonfunktionale GUI erstellt.
08.06.13	
09.06.13	
10.06.13	
11.06.13	<ul style="list-style-type: none"> • „GUI-interne“ Funktionen: <ul style="list-style-type: none"> ○ Popups ○ Fenster leeren ○ Sichtbarkeiten von Elementen (Etwas ist erst sichtbar, wenn etwas anderes ausgewählt ist)
12.06.13	<ul style="list-style-type: none"> • Verbindung zwischen XMPP Server und GUI hergestellt <ul style="list-style-type: none"> ○ Erste Funktionalitäten in die GUI eingebaut ○ Daten Holen für anzeige ○ Einloggen/ ausloggen • Sämtliche PopUp's integriert
13.06.13	<ul style="list-style-type: none"> • ID's der Themes/ Kategorien/ Genres geändert -> nun alle global eindeutig ➔ Einfacher für die abfragen • Webservices geändert: <ul style="list-style-type: none"> ○ GenreKategorien umgeschrieben : anstatt Text/Plain jetzt application/XML rückgabewerte ○ Überprüfungen der eingaben serverseitig ausgenommen, soll alles clientseitig passieren • UsersService „deleted“ : haben wir erst mal aus unserem Konzept herausgenommen, da wir jetzt nur noch auf Hauptfunktionalitäten fokussieren, anstatt auf Nebenfunktionalitäten • Derzeitig greift der Client noch nicht über URIS hinzu, sondern objekte...
14.06.13	<ul style="list-style-type: none"> • Anzeige der Themes / Kategorien / Genres abhängig voneinander funktioniert nun! • Theme Bearbeiten Pop-Up + vieles mehr nun so angepasst, dass man publish subscribe mit dem GUI-Client testen kann • Konsolen-Client ebenso auf die Änderungen der Webservices und dem Smackzeug angepasst, dass das pubsub getestet und ausgearbeitet werden kann • Publishen funktioniert nicht! Egal mit send/publish mit oder ohne Payload
15.06.13 + 16.06.13	<ul style="list-style-type: none"> • Webservices-Client erstellt • Alle GET's sind fertig und sollten auch funktionieren (müssen noch getestet werden)

17.06.13

- PUTS, POSTS, DELETES Signaturen erstellt.
- TestKlasse erstellt zum testen des webservice client, klappt teilweise, teilweise kommen noch exceptions
- Put, post, delete implementiert, aber noch nicht getestet
- Mentoren Feedback:
 - Publish Problem: konnte nicht gelöst werden
alle möglichen sends (versch./keine parameter) wurden nochmal ausprobiert, sämtliche configuration getestet, mit Hilfe eines anderen xmppserver getestet, neue Nodes erstellt, debug logs auf dem server und bei eclipse durchsucht -> nichts gefunden
Mentoren Feedback: lads bei git hoch, wir melden uns per email, leider können wir kein Feedback geben
 - GET Problem: konnte nicht gelöst werden
sobald path params in der uri auftauchen, funktioniert der zugriff nicht mehr
Mentoren Feedback: können leider nicht helfen.
 - Andere Probleme: konnte nicht gelöst werden
Mentoren Feedback: keine Zeit mehr dazu das anzugucken
- Beim gui client zu testzwecken ein publish-tab erstellt, damit ich nicht zugleich im konsolencient, als auch im guiclient auf haben muss.
- Anzeige bei stöbern nun komplett ohne nullpointer exception: vorher gabs Probleme bei clearselection, da dann der selectionchanged listener darauf angesprungen ist
- Publish funktioniert, aber wird noch nicht angezeigt: immer derselbe listener wird hinzugefügt: TODO: jeder user bekommt einen eigenen listener
- Delete listener erst einmal rausgenommen (Benachrichtigung, sobald ein node gelöscht wird)
- Button + funktion : alle abos kündigen eingerichtet – denn wir haben ein neues problem -> abos können nicht mehr gekündigt werden.
- XML SCHEMEN NEU geschrieben!!!! Damit bei http-Methoden JAXB objekte zurückgegeben werden können, da bei schachtelung kein root element... = heiliger Kuhmist, das funktioniert!
- So, ab dieser Stelle nehmen wir die Kommentare auch aus der Funktionalität heraus, da wir sonst wieder das Schema neu schreiben müssten und kompilieren. Das kennen und können wir ja schon
- Restclient „RestTest“ wurde erfolgreich ausgeführt ♥
- Endclient angefangen an den REST-Client anzupassen

18.06.13

- Dokumentation stichwortartig gesamten Inhalt durchgesprochen
- Grizzley-Server wird nun auch immer beendet, sobald user sich ausloggt.
- Publishen, subscriben, unsubscribe funktioniert manchmal, manchmal nur teilweise -> kp warum!
- Man kann nur auf das node publishen/ items empfangen, wenn man es in dieser sitzung neu abonniert hat.
- Versucht anstatt id's die titel der genres/ Kategorien/ Themes anzuzeigen -> verworfen, da unr fehler aufgetreten sind
- Paketstrukturänderung für Übersichtlichkeit, paar Klassen ausgelagert
- globalen Client erstellt, der auf xmpp und auf rc zugreift.
- Loginframe GUI inkl. Funktionalität

19.06.13

- Popup klasse geschrieben zum auslagern der popups, sieht auch schöner aus
- Versucht das Terminierproblem zu lösen (Programm läuft noch im Hintergrund, auch wenn der Benutzer es geschlossen hat) -> kA woran es liegt, kanns nicht lösen. Dachte erst es liegt daran, dass der grizzley server noch weiter läuft, aber anscheinend ist das nicht das Problem.
- ThemeInfo Klasse geschrieben + implementiert, zur Zeit zeigt der aber noch nicht alle Daten vom Theme an, sondern die kryptischen Objektnamen
- Anzeigebugs bei Benachrichtigungen behoben
- Theme erstellen Menü zu Theme erstellen + bearbeiten Menü geändert!!! (viel Arbeit), dazu dann auch objektnamen Inhalte anzeigen lassen
- Verbesserungswürdig: ausgeblendete Components sollen nicht wenn sie eingeblendet werden das ganz layout verrutschen.
- isSubscribed()-Methode für Strings geschrieben
- themes erstellen/ bearbeiten funktioniert so nicht, da nicht alle element root elemente sein können
- Versucht publish/ JID problem zu lösen: server gibt mit eine andere JID, als ich selbst.. name@localhost und name@localhost/smack -> deswegen läuft das wahrscheinlich nicht (über den smack debugger herausgefunden) -> zur Bereinigung eine UnSubscrib-Everybody Funktion geschrieben. Die funktioniert auch. Aber publishen funktioniert nicht obwohl dieser stanza geschickt wurde:

```
<message id="g0__user1@localhost__3w1qE"
  to="user1@localhost/smack"
  from="pubsub.localhost">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="g0">
      <item id="6ca85bc3-d689-40e6-8347-7208ab42511836"/>
    </items>
  </event>
  <headers xmlns="http://jabber.org/protocol/shim">
    <header name="pubsub#subid">
      srUCm9KMFJ8kRPYT88HmTIBG002W1IzIi3f2183k</header>
    </headers>
  </message>
```

20.06.13

- Also alle JID-Konflikte wurden gelöst. Daher keine Ahnung woran das liegen könnte, dass es nicht klappt! Heute Mittag ging es noch
- Weiteres werkeln am publishen
 - Da laut debugger stanzas versendet werden, liegt es vllt am Isitener, der nicht richtig eingestellt ist -> experiment: handler manuell aufrufen => LOL das funktioniert.
- Publish methode Rückgabewert geändert: jetzt boolean, für die Meldung ob publishing erfolgreich war
- Bei mehreren user wenn etwas gepublisht wird, werden nicht alle user benachrichtigt, auch wenn beide „online“ sind. -> wahrscheinlich weil die handle methode beim publisher aufgerufen wird, global funktioniert es nicht.
- Versuchen anstatt nur die id's, nun auch die titel der G, K und T's anzuzeigen
- Ok, abgesehen davon, dass die Theme Titel, wie wir gestern schon herausgefunden haben, nicht gemarshallt werden können, funktioniert.
- Titel werden nun auch bei publishmenü und bearbeitenmenü und abomenü angezeigt
- Theme Titel konnten damals, als der Rest client nur strings verlangt hat mit „.getAllgemeines().getThemeTitel()“ geklappt. Jetzt nicht.

	<ul style="list-style-type: none">• Versucht den baum auch mit titel darzustellen, aber klappt nicht – erst einmal theme Createn Funktion•
21.06.13+	<ul style="list-style-type: none">• Dokumentation
22.06.13	<ul style="list-style-type: none">• Kleine Verbesserungen am Grizzley Server
23.06.13	<ul style="list-style-type: none">• PUT und POST in der GUI funktional implementiert, aber aufgrund aufwendigem Schema+nicht passenden Aufbau des Schemas sehr verkrüppelt• Konsolenclient bisschen verbessert• Javadoc• Alle Klassen nochmal durchgeschaut und Code verschönert• Versucht alles Warnings zu eliminieren• LYX Tools gegooglet für Listings und Querverweise• Dokumentation: Änderung beim publishen, Schluss, überall mal was verbessert.