

Projektauftrag ASE 06

Inhaltsverzeichnis:

Projektauftrag ASE 06.....	1
(a) Projektbezeichnung & Projektteam.....	2
Projektbezeichnung	2
Projektteam	2
(b) Ausgangssituation	2
(c) Projektbeschreibung	3
(d) Zielgruppen	3
(e) Domänenmodell	4
(f) Komponentendiagramm	4
(g) Abgrenzungen	5
(h) Funktionale Anforderungen	6
1. Alle Tasks anzeigen	7
1.1 Tasks nach Priorität anzeigen.....	7
1.2 Tasks nach GPS-Kontext anzeigen	7
1.3 Tasks nach Tag anzeigen.....	7
1.4 Tasks nach Folder anzeigen	7
1.5 Tasks nach Datum anzeigen	7
1.6 Tasks suchen.....	8
2. Detailansicht eines Tasks anzeigen	8
2.1 Stoppuhr für Task verwenden	8
2.2 Kontakt aus Adressbuch anzeigen.....	8
2.3 Ort des Tasks anzeigen	8
2.4 Bild anzeigen	9
2.5 Sprachnotiz abspielen	9
2.6 Alle Tags des Tasks anzeigen	9
3. Task löschen	9
4. Task bearbeiten	10
5. Task anlegen	10
5.1 Zusatzinformation hinzufügen	10
5.1.1 Sprachnotiz aufnehmen	10
5.1.2 Bild hinzufügen	10
5.1.3 Mit Ort assoziieren.....	10
5.1.4 Mit Kontakten aus Adressbuch assoziieren	11
5.1.5 Task taggen.....	11
5.1.6 Task einem Folder hinzufügen	11
6. Nerv-Erinnerung aktivieren.....	11
7. Erinnerung anzeigen.....	11
8. Synchronisation durchführen	12
(h 2) Iceberglist	13
(i) Nichtfunktionale Anforderungen	14
(j) Lieferumfang und Abnahme	15
(k) Arbeitsstruktur & Rollenverteilung	15
Auftraggeber.....	15
Projektteam	15
Mehrdimensionale Rollenverteilung	16
(l) Projektplan.....	16
(l 2) Projektstrukturplan	17
(m) Informationswesen	18
(n) Besonderheiten	19

(a) Projektbezeichnung & Projektteam

Projektbezeichnung

Less 2 Do

Projektteam

- Tretter Matthias, 0726390, Teamkoordinator
- Messlehner Philip, 0728061, Teamkoordinator Stellvertreter, Dokumentenbeauftragter Stv.
- Schraml Gerhard, 0728067, Technischer Architekt
- Kurz Johannes, 0727957, Technischer Architekt Stv.
- Strixner Heinrich, 0525726, Test-u. Dokumentenbeauftragter

Details siehe (k) Arbeitsstruktur & Rollenverteilung

(b) Ausgangssituation

In der heutigen Zeit spielt Organisation eine wichtigere Rolle als je zuvor. Mehr und mehr übernehmen computerunterstützte Organisationstools die Rolle, die einstmalig Stift & Papier inne hatten.

Erkennbar ist das an der Flut an verschiedenen Diensten, die alle dem User die Organisation seiner Termine und Aufgaben erleichtern sollen. Angefangen bei Webdiensten wie Remember The Milk (RTM), Toodledo, Todoist oder Google Tasks bis hin zu Desktopsystemen wie Things, The Hit List, OmniFocus oder die rudimentäre iCal-Unterstützung von Todo's am Mac, es gibt unzählige verschiedene sogenannte Todo-Applikationen.

Optimal für den Benutzer ist es, wenn er immer Zugriff auf seine Daten hat.

Desktopsysteme haben hier den Vorteil, dass sie keine Internetanbindung brauchen, um Zugriff auf ihre Daten zu gewähren, Webportale auf der anderen Seite können von jedem Computer mit Internetzugriff erreicht werden. Im Idealfall sollten derart wichtige Daten wie Todo's von jedem Computer mit Internetzugriff, von seinem eigenen Computer auch ohne Internetzugriff und auch von seinem Handy aus erreichbar sein. Das erfordert eine Synchronisierung der Daten über eine Webapp, eine Desktopapp und eine mobile App. Todo's gehören (fast) immer zu einem bestimmten Kontext, in dem sie erledigt werden können. Kontexte spielen im GTD-System (Getting Things Done) eine zentrale Rolle und werden auch von den meisten Todo-Apps unterstützt. Jedoch müssen die bestimmten Kontexte immer händisch vom Benutzer ausgewählt werden, um die darin enthaltenen Todo's anzuzeigen.

Fokussieren wir uns auf die Situation am iPhone: Eine Schlagwortsuche nach dem Begriff "Todo" im AppStore präsentiert einem kurz danach eine Liste mit Sage und Schreibe 98 Apps (Stand: 21.10.09). Aufgrund des Wesens des iPhones (Smartphone) und der Tatsache, dass Apple keine Todo-Unterstützung am iPhone anbietet ist diese Situation nicht verwunderlich, das iPhone ist geradezu prädestiniert dafür als zentraler Punkt seiner Organisation verwendet zu werden. Schränkt man die Suche allerdings auf Programme ein, die mit einem Onlinedienst synchronisieren wird man nur noch mit einer Hand voll Programme konfrontiert. Mit Toodledo syncen derzeit 9 Apps, die beliebtesten davon sind: Appigo Todo (7,99 €), Toodledo (2,39 €), Ultimate Todos (0,79 €) und Pocket Informant (10,49 €).

Einzig und allein die mobile iPhone-App von RTM umfasst ein GPS-basiertes Feature. Für die Verwendung dessen ist allerdings ein recht teurer RTM Pro-Account nötig, bei dem jährliche Kosten fällig werden. Es gibt keine Gratis- oder billig vorhandene Lösung, die dieses Feature bietet. Ebenso gibt es Todo-Apps die zu den einzelnen Todo's Audionotizen oder Verweise zu Adressbuchkontakten speichern können, allerdings fehlen diesen Programmen andere Features wie zB die Synchronisation.

Fazit: Es gibt keine Todo-App im Apple App-Store, die die von unserem Projektteam gewünschten Features alle bietet, und das zu einem günstigen Preis.

(c) Projektbeschreibung

Im Zuge der Lehrveranstaltung ASE entwickeln wir eine Applikation für das iPhone, welche es erlaubt, Tasks zu erfassen und zu verwalten. Tasks bestehen aus einer Bezeichnung, dem zeitlichen Rahmen (inkl. der Möglichkeit einer regelmäßigen Wiederkehr), der Priorität, dem Status, optionalen Kennzeichnungen (Flags) sowie dem Kontext. Das besondere an unserer Lösung ist, dass der Kontext mit einer geographischen Position verknüpft werden kann. Außerdem soll es die Möglichkeit geben, Bilder und Audiokommentare sowie in weiterer Folge die Verknüpfung des Tasks mit einem Kontakt aus dem Adressbuch zu speichern.

Der Benutzer soll an seine Tasks erinnert werden können, wobei es verschiedene Modi dafür geben soll. Neben normalen Erinnerungen, welche einmalig auftreten, soll es auch "nervige" Erinnerungen geben, die sich so lange in immer kürzer werdenden Abständen aktivieren, bis das Todo tatsächlich als erledigt gekennzeichnet wurde. Außerdem soll auf die Tasks aufgrund ihres Kontexts aufmerksam gemacht werden. Kommt der Benutzer nun in die Nähe eines Ortes, an dem ein Todo fällig ist, soll die Applikation darauf aufmerksam machen. Natürlich kann die Aktion auch vom Benutzer ausgehen, nach dem Schema "welche Tasks befinden sich in der Nähe meiner derzeitigen Position?". Weiters soll es die Möglichkeit der Synchronisierung der Tasks mit dem bestehenden Webdienst Toodledo geben, um die Applikationsdaten besser zugänglich zu machen, allerdings soll die Applikation auch im Offlinebetrieb lauffähig sein.

(d) Zielgruppen

Zeit wird weniger, Termine werden mehr. Neue Geräte ermöglichen den Zugriff auf das Internet, jeder ist immer und überall erreichbar.

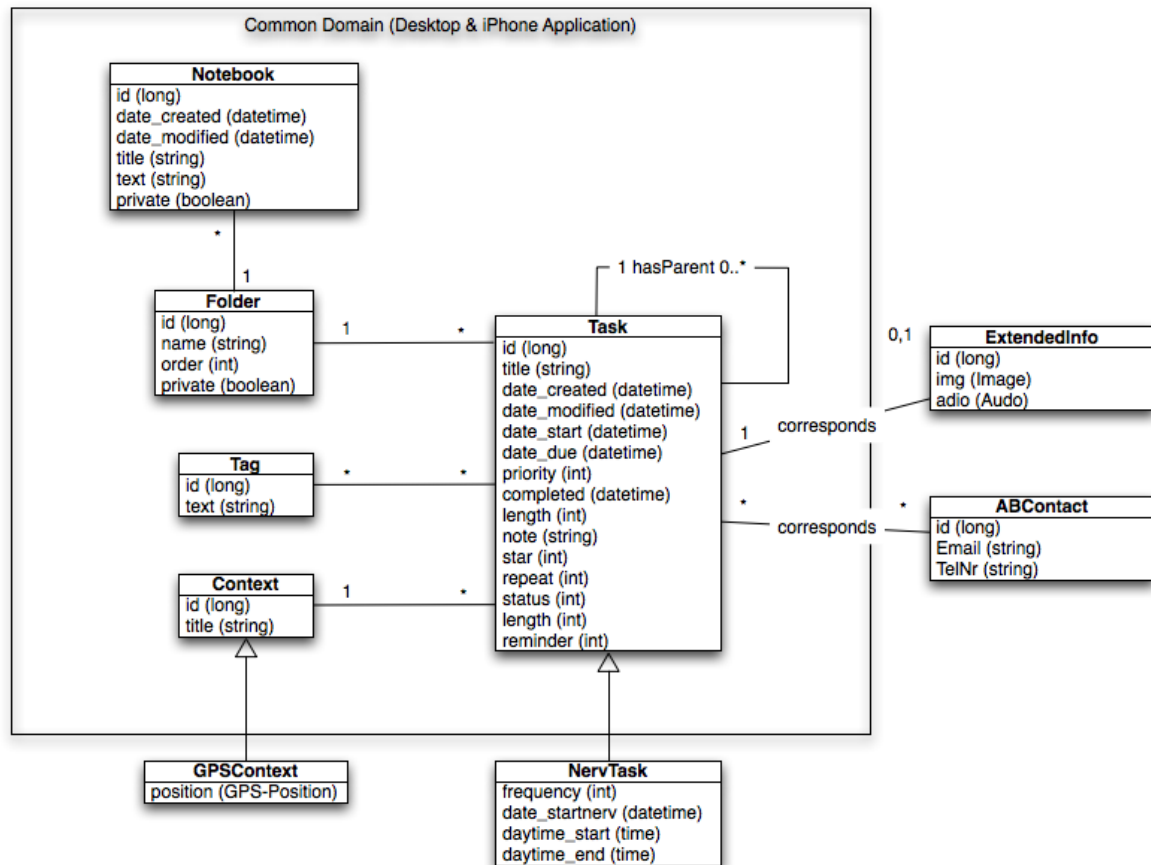
Um den Überblick zu bewahren reichen heute Stift & Papier für die Organisation nicht mehr aus. Unsere Anwendung richtet sich an alle Menschen, die ein iPhone besitzen und die totale Kontrolle über ihre noch zu erledigenden Tätigkeiten haben wollen. Wir wollen einen zentralen Punkt für den Tagesablauf eines modernen Menschen bieten, den er mit seinem Handy in der Hosentasche tragen kann. Vergesslichkeit ist dabei natürlich kein Problem, das iPhone soll als eine Art zweites Gehirn zur Unterstützung dienen.

Die Zielgruppe reicht von Schülern und Studenten, die unser Programm für die Organisation ihres Schul- bzw. Unialltags verwenden können, über Berufstätige die Unterstützung bei der Organisation ihrer Projekte benötigen bis hin zu einfachen Privatpersonen, die einfach gern an den nächsten Jahrestag mit ihrem Partner oder die aktuelle Einkaufsliste erinnert werden.

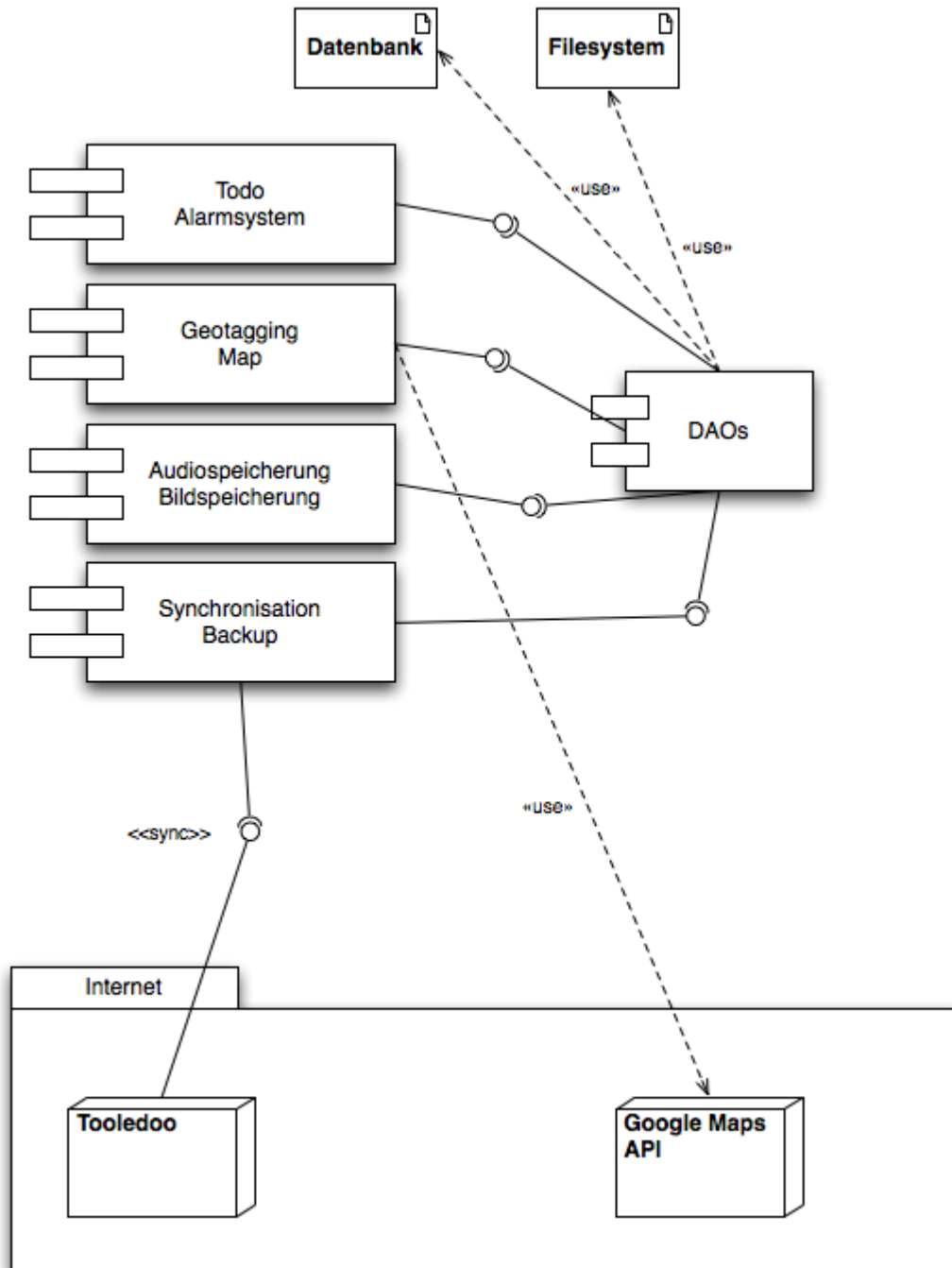
Bestehenden Benutzern von klassischen Todo-Anwendungen bieten wir einiges an Mehrwert: Durch die Synchronisation mit Toodledo ist der Zugriff auf die Tasks von allen möglichen internetfähigen Endgeräten gesichert. Das kann sich als sehr wichtig herausstellen, hat man mal sein iPhone vergessen oder ist der Akku gerade leer. Aber auch wenn kein Internetzugang vorhanden ist, kann man immer einfach und bequem von seinem iPhone aus seine Tasks managen. Benutzer von Desktopanwendungen genießen den Vorteil, die Organisation ihrer Tasks auch von unterwegs in der U-Bahn erledigen zu können und nicht immer ihren Computer dafür zu benötigen. Darüber hinaus bieten die zusätzlichen Features wie die Geo-Codierung der Tasks und

Audionotizen (genauere Beschreibung im nächsten Abschnitt: Featureliste), Audionotizen, Nerv-Reminders etc. einen weiteren großen Mehrwert, vor allem für die kontextbasierte Organisation von Tasks.

(e) Domänenmodell



(f) Komponentendiagramm



(g) Abgrenzungen

Die Applikation "Less 2 Do" soll sich vor allem an die Bedürfnisse mobiler Benutzer richten. Dabei stehen vor allem ein durchdachtes User Interface und eine gute Usability im Vordergrund. Die zur Verfügung stehenden Features sollen mit möglichst wenigen Bearbeitungsschritten durchgeführt werden können und die Benutzbarkeit der Software ist wichtiger als eine möglichst große Featurevielfalt ("eierlegende Wollmichsau").

Des weiteren entwickeln wir einen Todo-Client für das iPhone der als zusätzliches Feature eine Synchronisation mit dem Online-Anbieter Toodeedo anbietet, und keinen Toodeedo-Client. Deswegen nehmen wir uns heraus manche von Toodeedo angebotenen

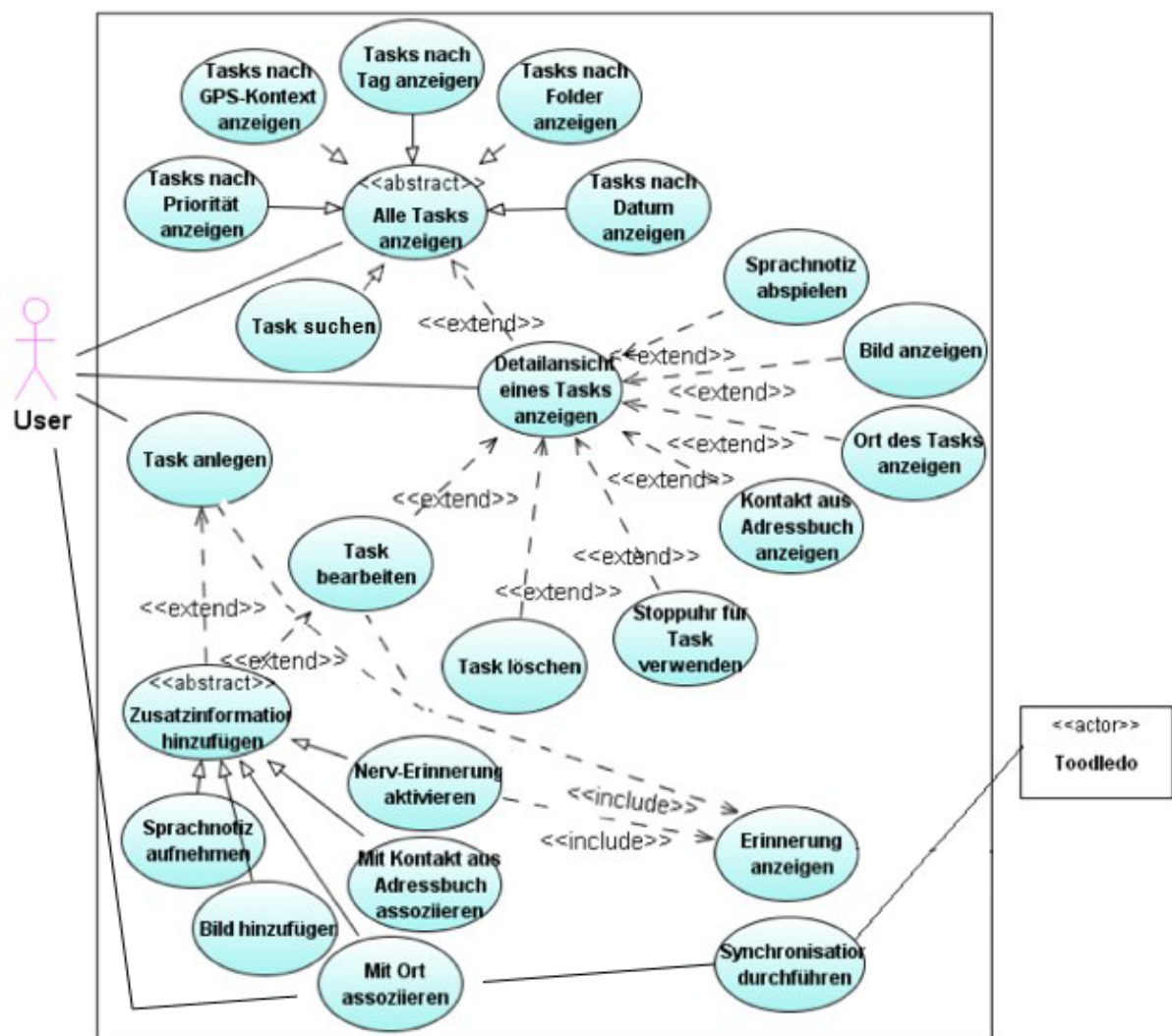
Features nicht zu unterstützen, und im Gegensatz die Funktionalität von Toodledo um andere Punkte zu erweitern, die dann natürlich nicht gesynct werden können. Auch ist keine Synchronisation mit anderen Online-Anbietern wie zum Beispiel Remember The Milk oder Google Tasks geplant.

Wir wollen auch den GTD-Ansatz nicht zu 100 % unterstützen, wir möchten einen einfach bedienbaren Todo-Client entwickeln dessen Bedienung keine Fragen aufwirft und der dadurch auch von technisch weniger versierten Benutzern gerne verwendet wird. Manche Todo-Applikationen im Apple AppStore unterstützen auch zusätzliche Funktionen wie die Bereitstellung einer Shoppinglist. Derartige Features sind in unserem Programm nicht geplant, eine Auflistung der geplanten Features findet sich unter (h) Funktionale Anforderungen.

Die Anwendung soll direkt mit Toodledo, ohne Einsatz einer weiteren Middleware, synchronisieren. Dies erspart uns die Kosten für die Bereitstellung und Instandhaltung eines eigenen Servers.

In der Anfangsphase soll das Programm nur in der Sprache Englisch verfügbar sein. Eine i18n Erweiterung nach Abschluss der LVA wird jedoch nicht ausgeschlossen!

(h) Funktionale Anforderungen



1. Alle Tasks anzeigen

Abstrakter Anwendungsfall - Konkrete Implementierung siehe folgende Punkte

1.1 Tasks nach Priorität anzeigen

Kurzbeschreibung: Alle gespeicherten Tasks werden geordnet nach ihrer Priorität angezeigt.

Vorbedingung: Es sind Tasks vorhanden.

Nachbedingung: Auf dem Bildschirm wird eine nach Priorität sortierte Liste von Tasks angezeigt.

Fehlersituation: Es sind noch keine Tasks vorhanden.

Systemzustand im Fehlerfall: Ausgabe "keine Tasks vorhanden"

Trigger: Der Benutzer drückt den Button "Tasks nach Priorität ordnen".

1.2 Tasks nach GPS-Kontext anzeigen

Kurzbeschreibung: Liste aller Tasks in einem bestimmten GPS-Kontext anzeigen.

Vorbedingung: Das iPhone muss mittels GPS lokalisiert werden können.

Nachbedingung: Tasks in der Nähe dieses Punktes werden am Bildschirm angezeigt (in Listenform oder auf einer Karte)

Fehlersituation: Im Umkreis des angegebenen Punktes sind keine Tasks vorhanden.

Systemzustand im Fehlerfall: Listenform: Ausgabe "keine Tasks im Umkreis vorhanden";

Karte: leere Karte wird angezeigt

Trigger: Der Benutzer wählt die GPS-basierte Anzeige von Kontexten.

1.3 Tasks nach Tag anzeigen

Kurzbeschreibung: Liste aller Tasks anzeigen, die mit einem bestimmten Tag assoziiert sind.

Vorbedingung: Es wird ein Tag eingegeben.

Nachbedingung: Auf dem Bildschirm wird eine Liste aller Tasks angezeigt, die mit dem eingegebenen Tag assoziiert sind.

Fehlersituation: Der eingegebene Tag ist mit keinem Task assoziiert.

Systemzustand im Fehlerfall: Ausgabe am Bildschirm "keine Tasks zu Tag 'soundso' gefunden"

Trigger: Der Benutzer drückt den Button "Tasks nach Tag suchen"

1.4 Tasks nach Folder anzeigen

Kurzbeschreibung: Liste aller Tasks anzeigen, die in einem bestimmten Folder liegen.

Vorbedingung: Es wird die Folder-Struktur angezeigt und der Benutzer wählt den entsprechenden Folder aus.

Nachbedingung: Am Bildschirm werden alle Tasks, die im ausgewählten Folder liegen, angezeigt.

Fehlersituation: Der ausgewählte Folder ist leer.

Systemzustand im Fehlerfall: Ausgabe am Bildschirm "ausgewählter Folder ist leer".

Trigger: Der Benutzer drückt den Button "Tasks nach Folder anzeigen"

1.5 Tasks nach Datum anzeigen

Kurzbeschreibung: Liste aller Tasks anzeigen, nach Datum sortiert (Defaultansicht)

Vorbedingung: Es sind Tasks vorhanden.

Nachbedingung: Am Bildschirm werden alle Tasks ausgegeben, nach Datum sortiert

Fehlersituation: Keine Tasks vorhanden

Systemzustand im Fehlerfall: Ausgabe am Bildschirm "keine Tasks vorhanden".
Trigger: Programm wird gestartet, oder Ansichtsoption "nach Datum sortiert" wird ausgewählt)

1.6 Tasks suchen

Kurzbeschreibung: Ein Task wird vom Benutzer durch Eingabe spezifischer Suchdaten gesucht.

Vorbedingung: Es sind Tasks vorhanden

Nachbedingung: Am Bildschirm wird der entsprechende Task, bzw. eine Liste der Tasks, welche mit den Suchkriterien übereinstimmen angezeigt.

Fehlersituation: Task wird nicht gefunden

Systemzustand im Fehlerfall: Ausgabe am Bildschirm "keine Tasks gefunden".

Trigger: Der Benutzer drückt den Button "Task suchen"

2. Detailansicht eines Tasks anzeigen

Kurzbeschreibung: Es wird der Task mit allen seinen Informationen dargestellt

Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde

Nachbedingung: Der Task wird am Bildschirm mit allen Informationen angezeigt

Fehlersituation: keine Einträge vorhanden

Systemzustand im Fehlerfall: es kann kein Task in der Detailansicht angezeigt werden

Trigger: der Benutzer wählt aus der Liste der Tasks einen Task an

2.1 Stoppuhr für Task verwenden

Kurzbeschreibung: Es soll die Zeit für das Abarbeiten des Tasks aufgezeichnet werden

Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.

Nachbedingung: Nach dem Stoppen der Uhr wird die verbrauchte Zeit zum Task gespeichert.

Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt

Systemzustand im Fehlerfall: Es kann keine Zeitaufzeichnung gestartet werden

Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für die Zeitaufzeichnung

2.2 Kontakt aus Adressbuch anzeigen

Kurzbeschreibung: Es sollen zu einem speziellen Task die verlinkten Kontakte aus dem Adressbuch angezeigt werden

Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.

Nachbedingung: Ausgabe einer Liste der verlinkten Kontakte aus dem Adressbuch des jeweiligen Tasks

Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt oder keine Verlinkung zum Adressbuch vorhanden

Systemzustand im Fehlerfall: Es können keine Kontakte angezeigt werden

Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für die Anzeige der Kontakte

2.3 Ort des Tasks anzeigen

Kurzbeschreibung: Es soll zu einem speziellen Task der verlinkte Ort angezeigt werden

Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.

Nachbedingung: Anzeige des verlinkten Ortes des jeweiligen Tasks in einer Karte
Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt oder keine Verlinkung zu einem Ort vorhanden
Systemzustand im Fehlerfall: Es kann kein Ort angezeigt werden
Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für die Anzeige des Ortes

2.4 Bild anzeigen

Kurzbeschreibung: Es soll zu einem speziellen Task das verlinkte Bild angezeigt werden
Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.
Nachbedingung: Anzeige des verlinkten Bildes des jeweiligen Tasks
Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt oder keine Verlinkung zu einem Bild vorhanden
Systemzustand im Fehlerfall: Es kann kein Bild angezeigt werden
Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für die Anzeige des Bildes

2.5 Sprachnotiz abspielen

Kurzbeschreibung: Es soll zu einem speziellen Task die verlinkte Sprachnotiz abgespielt werden
Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.
Nachbedingung: Abspielen der verlinkten Sprachnotiz des jeweiligen Tasks
Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt oder keine Verlinkung zu einer Sprachnotiz vorhanden
Systemzustand im Fehlerfall: Es kann keine Sprachnotiz abgespielt werden
Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für das Abspielen der Sprachnotiz

2.6 Alle Tags des Tasks anzeigen

Kurzbeschreibung: Es werden alle Tags zum zugehörigen Task angezeigt
Vorbedingung: Es wurde ein Task ausgewählt, der in der Detailansicht dargestellt wird.
Nachbedingung: Liste der Tags zum zugehörigen Task wird angezeigt.
Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt oder keine Tags zum Task gespeichert.
Systemzustand im Fehlerfall: Es kann keine Liste der Tags dargestellt werden.
Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Button für das Anzeigen der Tags

3. Task löschen

Kurzbeschreibung: Ein Task soll gelöscht werden können
Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.
Nachbedingung: Der User bestätigt die Löschung des Tasks, ansonst bleibt der Task im System.
Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt
Systemzustand im Fehlerfall: Es kann kein Task gelöscht werden
Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Löschen-Button

4. Task bearbeiten

Kurzb Beschreibung: Ein Task soll bearbeitet werden können

Vorbedingung: Es sind Tasks vorhanden, wobei ein Task ausgewählt wurde und in der Detailansicht dargestellt wird.

Nachbedingung: Der User bestätigt Eingabe der Änderung, ansonst werden die Änderungen nicht gespeichert

Fehlersituation: Keine Einträge vorhanden bzw. kein Task ausgewählt

Systemzustand im Fehlerfall: Es kann kein Task bearbeitet werden

Trigger: Der Benutzer wählt in der Detailansicht des Tasks den Edit-Button

5. Task anlegen

Kurzbeschreibung: Es soll ein neuer Task angelegt werden

Vorbedingung: Programm gestartet

Nachbedingung: Wenn der Benutzer die Eingaben bestätigt wird der Task gespeichert, ansonst wird die Änderung verworfen

Fehlersituation: Anlegen wird abgebrochen

Systemzustand im Fehlerfall: Die Änderungen werden verworfen und die Übersicht der Tasks wird wieder angezeigt

Trigger: Der Benutzer wählt den „+“-Button in der Übersicht der Tasks

5.1 Zusatzinformation hinzufügen

Abstrakter Anwendungsfall - Konkrete Implementierung siehe folgende Punkte

5.1.1 Sprachnotiz aufnehmen

Kurzbeschreibung: Es soll zu einem Task eine Sprachnotiz gespeichert werden

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angelegt

Nachbedingung: Die Sprachnotiz wird zum Task hinzu gespeichert

Fehlersituation: Das Speichern der Sprachnotiz wird abgebrochen

Systemzustand im Fehlerfall: Die Änderungen werden verworfen und die Sprachnotiz nicht gespeichert. Der Bearbeitungsmodus des Tasks wird weiter angezeigt

Trigger: Der Benutzer wählt den Button für das hinzufügen der Sprachnotiz

5.1.2 Bild hinzufügen

Kurzbeschreibung: Es soll zu einem Task ein Bild gespeichert werden

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angelegt

Nachbedingung: Das Bild wird zum Task hinzu gespeichert

Fehlersituation: Das Speichern des Bildes wird abgebrochen

Systemzustand im Fehlerfall: Die Änderungen werden verworfen und das Bild nicht gespeichert. Der Bearbeitungsmodus des Tasks wird weiter angezeigt

Trigger: Der Benutzer wählt den Button für das hinzufügen eines Bildes

5.1.3 Mit Ort assoziieren

Kurzbeschreibung: Es soll zu einem Task eine Ort gespeichert werden (mittels GPS-Koordinaten)

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angelegt

Nachbedingung: Ein Ort wird zum Task hinzu gespeichert

Fehlersituation: Das Speichern des Ortes wird abgebrochen

Systemzustand im Fehlerfall: Die Änderungen werden verworfen und der Ort nicht gespeichert. Der Bearbeitungsmodus des Tasks wird weiter angezeigt

Trigger: Der Benutzer wählt den Button für das hinzufügen eines Ortes

5.1.4 Mit Kontakten aus Adressbuch assoziieren

Kurzbeschreibung: Es soll zu einem Task Kontakte aus dem Adressbuch gespeichert werden

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angelegt

Nachbedingung: Kontakte werden zum Task hinzu gespeichert

Fehlersituation: Das Speichern der Kontakte wird abgebrochen

Systemzustand im Fehlerfall: Die Änderungen werden verworfen und die Kontakte nicht gespeichert. Der Bearbeitungsmodus des Tasks wird weiter angezeigt

Trigger: Der Benutzer wählt den Button für das hinzufügen eines Kontakts

5.1.5 Task taggen

Kurzbeschreibung: Ein Task wird mit einem Tag gekennzeichnet

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angezeigt.

Nachbedingung: Der Tag wurde zum Task gespeichert.

Fehlersituation: Das Speichern des Tags wurde abgebrochen.

Systemzustand im Fehlerfall: Die Änderung wird verworfen und das normale Bearbeiten des Tasks fortgesetzt.

Trigger: Der Benutzer wählt den Button für das hinzufügen eines Tags

5.1.6 Task einem Folder hinzufügen

Kurzbeschreibung: Ein Task wird einem Folder hinzugefügt.

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angezeigt.

Nachbedingung: Die Folderzuordnung wird zum Task gespeichert.

Fehlersituation: Das Speichern der Folderzuordnung wurde abgebrochen.

Systemzustand im Fehlerfall: Die Änderungen werden verworfen.

Trigger: Der Benutzer wählt den Button für die Zuordnung eines Tasks zu einem Folder

6. Nerv-Erinnerung aktivieren

Kurzbeschreibung: Es soll zu einem Task die Nerv-Erinnerung eingeschaltet werden, d.h. man möchte wiederholt an die Erledigung der Aufgabe erinnert werden.

Vorbedingung: Der Task befindet sich im Bearbeitungsmodus bzw. wird gerade angelegt.

Nachbedingung: Die Nerv-Erinnerung wird zum Task hinzugespeichert.

Fehlersituation: Das aktivieren der Nerv-Erinnerung wird abgebrochen

Systemzustand im Fehlerfall: Die Änderung wird verworfen

Trigger: Der Benutzer aktiviert die Checkbox "Nerv-Erinnerung" bzw. markiert den Task mittels Sternbewertung.

7. Erinnerung anzeigen

Kurzbeschreibung: Eine Erinnerung an einen fälligen Task wird angezeigt.

Vorbedingung: Das Programm ist gestartet und das Fälligkeitsdatum und die Fälligkeitszeit des Tasks ist erreicht

Nachbedingung: Am Bildschirm wird der Erinnerungstext ausgegeben.

Fehlersituation: Programm ist nicht gestartet

Systemzustand im Fehlerfall: Die Erinnerung wird nicht angezeigt.

Trigger: Aktuelles Datum und aktuelle Zeit sind ident mit dem Fälligkeitsdatum und der Fälligkeitszeit

8. Synchronisation durchführen

Kurzbeschreibung: Die lokal gespeicherten Daten (XML) werden mit Toodledo synchronisiert.

Vorbedingung: Es sind Daten zum Synchronisieren vorhanden

Nachbedingung: Die Daten am Toodledo-Server und iPhone sind ident.

Fehlersituation:

- a) Keine Konnektivität zu Toodledo vorhanden.
- b) Benutzer mit anderen Account eingeloggt, bzw. unterscheiden sich mehr als 5% der Tasks am Server von den Tasks am iPhone
- c) Ein Task wird an am iPhone geändert, jedoch wurde der gleiche Task bereits über eine andere Applikation verändert

Systemzustand im Fehlerfall:

- a) Synchronisation wurde nicht durchgeführt und muss zu einem späteren Zeitpunkt noch einmal versucht werden.
- b) Für jeden Benutzer wird ein eigenes File zum Speichern der Daten offline angelegt. Zusätzlich wird nachgefragt, ob der Sync wirklich durchgeführt werden soll, falls sich mehr als 5% der Daten unterscheiden.
- c) Der User wird befragt, ob die neuere Version, die am Server liegt überschrieben, downgeloadet oder die veränderte Version am iPhone als Kopie an den Server gesendet werden soll.

Trigger: Implizit - jedes Mal wenn sich lokale Daten ändern (beim Anlegen/Bearbeiten/Löschen)

Explizit - der Benutzer betätigt den Sync-Button

(h 2) Iceberglist

Feature	Usecase	Priorität	Release	Aufwand	Verantwortlichkeit
Tasks nach Datum anzeigen	1,5	1	R1		TA + TA2
Detailansicht des Tasks anzeigen	2	1	R1		TDB
Tasks nach Priorität anzeigen	1,1	1	R2		TA + TA2
Tasks nach GPS-Kontext anzeigen	1,2	1	R2		TK + TK2
Tasks nach Tag anzeigen	1,3	1	R2		TA + TA2
Tasks nach Folder anzeigen	1,4	1	R2		TA + TA2
Task löschen	3	1	R3		TDB
Task anlegen	5	1	R3		TDB
Tasks suchen	1,6	1	R4		TK + TK2
Ort des Tasks anzeigen	2,3	1	R4		TK + TK2
Alle Tags des Tasks anzeigen	2,6	1	R4		TA + TA2
Gesamten Folder des Tasks anzeigen	2,7	1	R4		TK + TK2
Task bearbeiten	4	1	R4		TDB
Erinnerungen/Nerv-Erinnerungen	6	1	R4		TA + TA2
Erinnerung/Nerv-Erinnerung anzeigen	7	1	R4		TA + TA2
Synchronisation durchführen	8	1	R5		TA + TA2
Mit Ort assoziieren	5.1.3	1	R5		TK + TK2
Task taggen	5.1.5	1	R5		TA + TA2
Task einem Folder hinzufügen	5.1.6	1	R5		TK + TK2
Stoppuhr für Task verwenden	2,1	2	R5		TA + TA2
Bild anzeigen	2,4	2	R5		TDB + TK/TK2
Bild hinzufügen	5.1.2	2	R5		TDB + TK/TK2
Kontakt aus Adressbuch anzeigen	2,2	3	offen		
Sprachnotiz abspielen	2,5	3	offen		
Sprachnotiz aufnehmen	5.1.1	3	offen		
Mit Kontakt aus Adressbuch assoziieren	5.1.4	3	offen		
Push-Benachrichtigungen		3	offen		

Prioritäten:

- 1 ... Must-have
- 2 ... Nice-to-have
- 3 ... Erweiterungsmöglichkeit

Releases:

- R1 ...intern (21.11.)
- R2 ...intern(5.12.)
- R3 ...mit MR2
- R4 ...intern (31.12.)
- R5 ...mit MR3

(i) Nichtfunktionale Anforderungen

- Qualitätsanforderungen
 - Usability: Die Applikation muss wie bereits erwähnt auch für technisch nicht versierte Anwender sehr einfach und intuitiv bedienbar sein. Die Notwendigkeit eines Benutzerhandbuchs soll vermieden werden. Es muss möglich sein, neue Tasks mit Defaultwerten nur durch einen Klick für die Erstellung des Tasks, die Eingabe der Taskbezeichnung und durch wiederum einen Klick zum Speichern hinzuzufügen. Auch das ändern einer Option darf maximal 3 zusätzliche Klicks erfordern.
 - Ästhetik: Das User-Interface muss durch Wahl eines passenden Logos, passenden Farben und Kontrasten sowie einem passenden Layout ansprechend gestaltet sein, und einen freundlichen, aufgeräumten Eindruck erwecken. Es darf nicht vom eigentlichen Kern der Anwendung ablenken und muss sich dezent im Hintergrund halten. Die Apple User Interface Guidelines fürs iPhone müssen dabei berücksichtigt werden, die Bedienung soll sich an der gewohnten Bedienung von iPhone-Apps orientieren und soll keine Überraschungen mit sich bringen.
 - Sicherheit/Verfügbarkeit: Sowohl die Integrität, Konsistenz als auch die Verfügbarkeit der Daten muss gewährleistet sein. Das wird zum einen durch einen Offline-Modus gesichert, zum anderen muss die Synchronisation fehlerlos verlaufen bzw. müssen Auswege (zB Definition der Serverdaten als Masterdaten) bei Sync-Konflikten ermöglicht werden. Ebenso muss eine Möglichkeit vorgesehen werden, die Daten wieder vom Device zu löschen, um Missbrauch derer zu verhindern
 - I18N: Das Programm wird in Englisch zur Verfügung stehen um ein breites Spektrum an Benutzern zu erreichen.
 - Erweiterbarkeit: Das Datenmodell muss eine einfache Erweiterung um andere Sync-Knoten (eigener Syncdienst, RTM, Synchronisation mit der Desktop-Todo-App der ASE-Gruppe 03) ermöglichen
 - Korrektheit: Die aufgezeichneten Daten müssen zu jedem Zeitpunkt korrekt sein
 - Authentizität: Es muss nachvollziehbar sein, wann die letzten Änderungen an einem Task vorgenommen wurden
 - Fehlertoleranz: Unvorhergesehene Eingaben des Benutzers dürfen nicht zu Systemfehlern führen sondern müssen erkannt und verarbeitet werden können.
 - Testbarkeit: Die Komponenten des Programmes müssen eine Black Box darstellen und für sich getestet werden können
- Leistungsanforderungen
 - Antwortzeit: Das Ansprechverhalten des Systems muss schnell sein und innerhalb der Aufrechterhaltung der Konzentrationsspanne des Benutzers erfolgen, um die User nicht durch eine langwierige Aufzeichnung von Tasks von deren Aufzeichnung abzuhalten
 - Synchronisation: Die Synchronisierung muss möglichst transparent (Änderungen müssen sichtbar sein) und schnell durchgeführt werden können. Zusätzlich muss ein Timeout für die Requests gesetzt werden, um unnötige Blockierung anderer Aktionen zu vermeiden.
 - Startup-Time: Die Zeitspanne vom Starten des Programms bis zur möglichen Benutzbarkeit durch den User darf nicht zu groß sein und muss sich an der Konkurrenzprodukte orientieren.
- Umgebungsanforderungen
 - Das Programm ist darauf ausgelegt auf allen iPhone- und iPod Touch-Versionen lauffähig zu sein. Voraussetzung ist Version 3.0 des iPhone OS. GPS-basierte Kontexte werden auf Geräten ohne eingebautes GPS

- (iPhone Classic, iPod Touch) nicht vollständig unterstützt (nur Lokalisierung über WLAN/Handymasten).
- Der Umfang der Software soll die 10 MB - Grenze nicht übersteigen, um somit auch per mobilem Datennetz aus dem AppStore installiert werden zu können

(j) Lieferumfang und Abnahme

Die Abnahme findet in mehreren Schritten statt.

Es finden jede Woche Treffen mit dem Tutor statt. Bei diesen Treffen werden alle Themen besprochen die dem Projektteam unklar sind und für die LVA benötigt werden.

Im MR2 wird ein Prototyp vorgeführt, welcher den Funktionsumfang und die Funktionsweise veranschaulichen soll. Zusätzlich werden Probleme behandelt und verschiedene informative Dokumente präsentiert. Des weiteren kann ein Ausblick gegeben werden ob "Nice-to-Have"-Funktionalitäten im Rahmen dieser LVA realisiert werden können.

Das MR3 ist die letzte Abnahme und gleichzeitig die Übergabe.

Im Rahmen der Übergabe wird eine Präsentation über das Projekt gehalten.

Jeder der beteiligten Studenten wird über seine Erfahrungen und sein Teilgebiet vortragen.

Anschließend wird die Applikation direkt auf den Endgeräten erklärt und vorgeführt.

Die Vorführung dient dazu, den Anwesenden ein Gefühl der Benutzeroberfläche und des Handlings zu geben. Zusätzlich kann sich jeder, nicht nur anhand von Bildern, selbst von der Geschwindigkeit und Qualität unserer Anwendung überzeugen.

Eine Benutzerdokumentation und eine Version Release History wird übergeben. Das Programm soll in weiterer Folge im Apple App-Store zur Verfügung stehen und wird sich gegenüber der Konkurrenz zum ersten mal bewähren. Der Lieferumfang für den Endbenutzer setzt sich aus dem Programm und einer Beschreibung für den Anwender zusammen.

(k) Arbeitsstruktur & Rollenverteilung

Auftraggeber

Nicht vorhanden, Eigeninitiative

Projektteam

Name	Rolle	Aufgaben
Tretter Matthias	Teamkoordinator (TK)	<ul style="list-style-type: none"> • Projektorganisation, -administration • Aufgabenverteilung • Führung und Motivation der Übungsgruppe • Sprachrohr • Interaktionsdesign
Messlehner Philip	Teamkoordinator Stellvertreter, Dokumentenbeauftragter Stellvertreter (TKSTV, DBSTV)	<ul style="list-style-type: none"> • übernimmt im Abwesenheitsfall sämtliche Aufgaben des TK • Kontrolle, Dokumentation der Aktivitäten des TK • Unterstützung des TK • Analyse

		<ul style="list-style-type: none"> • Unterstützung des Dokumentenbeauftragten in seinen Funktionen
Schraml Gerhard	Technischer Architekt (TA)	<ul style="list-style-type: none"> • Systemarchitektur • Code-Reviewer • technische Qualitätssicherung • Infrastrukturplanung
Kurz Johannes	Technischer Architekt Stellvertreter (TASTV)	<ul style="list-style-type: none"> • übernimmt im Abwesenheitsfall sämtliche Aufgaben des TA • Kontrolle, Dokumentation und Unterstützung des TA
Strixner Heinrich	Test- und Dokumentenbeauftragter (DB)	<ul style="list-style-type: none"> • Testleiter • Dokumentenrichtlinien • Dokumentenüberwachung

Mehrdimensionale Rollenverteilung

		Tretter Matthias (TK)	Messlehner Philip (TKSTV,DBSTV)	Schraml Gerhard (TA)	Kurz Johannes (TASTV)	Strixner Heinrich (DB,TEST)
Rolle	Hauptaufgabe	PM	Unterst. TK,DB	SW Arch.	Unterst. TA	DB QoS
TK	PM & Tracking	Verantwortlich	Berichte	Berichte	Berichte	Berichte
TA	Entwurf & Design	Interaktions- design	Interaktions- design	Allgemein Verantw., Systemdesign	Technischer Aspekt	Interaktions- design
TA	Implementierung	UI / Usecases	UI / Usecases	API Sync	API Sync	UI / Usecases
TEST	Testing	Real-Life Test	RL Test	Blackbox- Tests	Blackbox- Tests	Verantwortlich RL Test
DB	Dokumentation	Mitarbeit	Mitarbeit	Mitarbeit	Mitarbeit	Verantwortlich

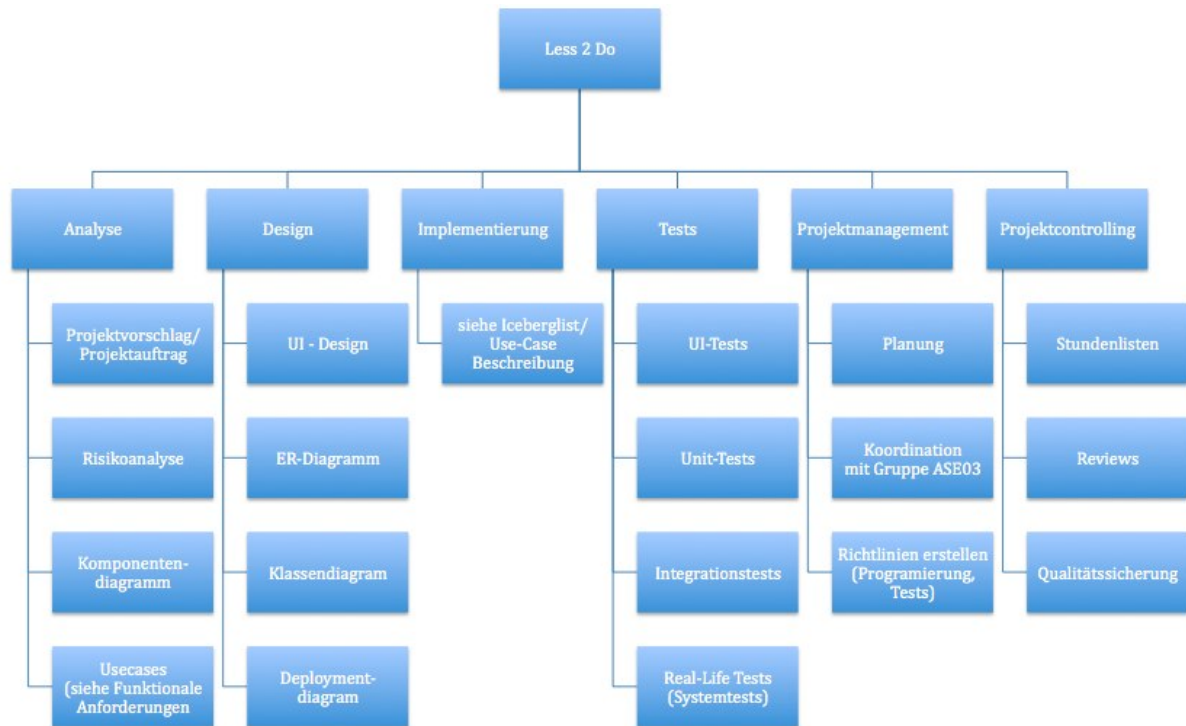
(I) Projektplan

Der Startschuss des Projektes erfolgte mit dem Kickoff-Meeting am 16.10.09 und endet voraussichtlich mit der Abnahme des Projektes beim MR3 Ende des WS09. Das Projekt teilt sich weiters in 5 interne Releases (R1- R5) auf.

Phase	Termin/ Zeitraum	Ziel/Termin
Anforderungsanalyse	16.10.09	Kickoff-Meeting, Startschuss des Projekts
	16. - 19.10.09	Sammeln von Projektideen, Entscheidung für Projektvorschlag
	22.10.09	MR1, Präsentation des Projektumfangs Abgabe der in MR1 geforderten Dokumente und Protokolle
	25.10. - 10.11.	Einlernphase in Objective-C und Cocoa Touch
Entwurf	25.10. - 15.11.	Grundlegendes Design und Schnittstellendefinitionen (Maps, Voice, Todos,

		Syncs, Toodledo-API), Programmierrichtlinien festlegen.
	25.10. - 15.11.	Entwickeln eines Testplans und mögliche Testfälle. Konzept für Akzeptanztests erstellen.
	25.10. - 15.11.	Erstellen einer Datenbankstruktur inkl. ER Diagramm.
	25.10. - 15.11.	Präzisierung der funktionalen Anforderungen. (Detaillierte Beschreibung der Anwendungsfälle, Erstellen von Anwendungsfalldiagrammen und AkteurInnenliste und Hierarchie, erstellen einer Übersicht)
Implementierung	25.10. - 21.11.	Erstellung des User Interfaces und Abbildung der wichtigsten Anwendungsfälle.
	25.10. - 15.11.	Implementierung der Toodledo-API
	21.11.	Release1 (R1) "Erste lauffähige Applikation"
	21.11. - 5.12.	Grundsystem mit vordefinierten Librarys und Layoutdesign
	5.12.	R2 "Erstes lauffähiges Grundsystem"
	1.12. - 31.12.	Implementierung der Funktionalitäten in den Modulen
	Mitte Dezember	R3 = MR-2, Präsentation des lauffähigen Prototypen mit GUI, Abgabe des Letztstandes der in MR-2 geforderten Dokumente und Protokolle
	1.12. - 31.12.	Datenspeicherung (Audio, Text, GEO-Info)
	31.12.	R4 "Synchronisation"
	31.12.09 - 15.1.2010	Implementierung der weiteren Features
Test	15.1.2010 - 31.1.2010	Real-Life Test der Applikation
	Mitte Jänner	R5 = MR-3, Präsentation des Endproduktes, Abgabe der finalen Dokumente

(I 2) Projektstrukturplan



(m) Informationswesen

Das Informationswesen gliedert sich in folgende Teile:

Für den obligatorischen E-Mail-Verkehr wurde mit Hilfe von Google Groups ein Mailverteiler angelegt.

Diese Google Group dient auch gegebenenfalls dafür, einfache Websites oder Blogs online zu stellen.

Darüber hinaus befindet sich die Dokumentation des Projektes in einem kollaborierten Google Docs-Ordner. Dadurch ist die einfache Verteilung der Dokumentation, sowie sogar gemeinsames, gleichzeitiges Arbeiten an Dokumenten möglich. Alle anderen, von Google Docs nicht unterstützten, projektspezifischen Dateien werden per Dropbox automatisch mit allen Beteiligten synchronisiert (www.dropbox.com). Weitere verwendete Tools für Dokumentation, Diagramme und Implementierung sind OmniGraffle, GanttProject und XCode sowie Interface Builder. Zusätzlich werden wichtige projektspezifische Daten auf das Repository (github.com) geladen.

Für Besprechungen wird entweder ein Termin vereinbart oder das VOIP Tool Skype verwendet. Dieses Tool hat sich bereits in vorangegangenen Projekten bestens bewährt. Regelmäßige Absprachen und ein gemeinsamer Projektkalender ermöglichen eine reibungslose Koordination und schaffen die Möglichkeit für alle Teammitglieder, den Überblick zu bewahren.

Geplant sind wöchentliche Treffen in der Gruppe, sowie vereinzelte "Coding Sessions", in der große Milestones gemeinsam erreicht werden können. Gemeinsames Arbeiten ist sehr förderlich für die Motivation und es können Probleme zwischen den Teammitglieder sofort ausgetauscht und darauf reagiert werden. Es wurde auch ein regelmäßiges

Reporting mit dem Tutor vereinbart.

(n) Besonderheiten

Less 2 Do ist ein Projekt welches im Zuge der Vorlesung Advanced Software Engineering von fünf motivierten Studenten durchgeführt wird.

In diesem Semester gibt es zum ersten mal die Möglichkeit, mit Unterstützung von zwei Tutoren, Programme für Apple-Plattformen zu erstellen. Für die Entwicklung eines Programmes für die iPhone-Plattform wird ein Intel-Mac benötigt. Im Team wird bis auf eine Ausnahme das Betriebssystem Snow Leopard (Mac OS 10.6) und die XCode-Version 3.2 zum Einsatz kommen, als Programmiersprache wird Objective C 2.0 und als Framework Cocoa Touch eingesetzt.

Das Testen wird grob in Real-Life Tests und Blackbox-Tests unterteilt. Die Blackbox-Tests testen eine Funktionalität für sich und können somit (fast) immer auf dem iPhone-Simulator am Mac ausgeführt werden. Real-Life Tests testen das Gesamtsystem und müssen somit auf einem iPhone oder iPod Touch durchgeführt werden. Mit derzeitigem Stand besitzen 3 der 5 Teammitglieder ein iPhone und diese sind somit für die Real-Life Tests verantwortlich.

Less 2 Do wird eine iPhone Applikation, die eine Schnittstelle für die Kooperation mit einer anderen Projektgruppe offen lässt. Diese Projektgruppe verfolgt ein ähnliches Ziel, nur als Desktop-Version für Mac OSX und keine mobile Version für das iPhone. Die Projektgruppen werden in einem parallel laufenden Gemeinschaftsbetrieb eine Toodledo-API für Objective-C erstellen und als OpenSource-Projekt veröffentlichen. Auf diese Weise werden Serversynchronisation sowie Datenstrukturen zwischen den zwei Gruppen vereinheitlicht, was für beide Gruppen zum einen eine Arbeitserleichterung darstellt, zum anderen bleibt so die Möglichkeit für eine eventuell spätere Zusammenarbeit mit Aussicht auf einen eigenen Online-Sync-Dienst unabhängig von Toodledo offen. "Less 2 Do" soll nach erfolgreicher Abgabe über den Apple AppStore, für einen Kleinstbetrag kommerziell vertrieben werden. Prinzipiell ist "Less 2 Do" aber Open Source.