TH Köln Studiengang

Medieninformatik

EIS Projekt

Sommersemester 2017

**Dokumentation des Projekts**

**HarvestHand**

Studierende

Franziska Gonschor

Sergej Atamantschuk

Betreuer

Robert Gabriel

Prof. Dr. Gerhard Hartman

Prof. Dr.Kristian Fischer

Inhaltsverzeichnis

[1. Einleitung 14](#_Toc484788099)

[1.1. Nutzungsproblem 14](#_Toc484788100)

[1.2. Ziel des Projekts 14](#_Toc484788101)

[2. Proof of Concepts 15](#_Toc484788102)

[2.1. Benutzer anlegen 15](#_Toc484788103)

[2.2. Eintrag mit Ackerdaten erstellen 15](#_Toc484788104)

[2.3. Collaborators dem Eintrag hinzufügen 15](#_Toc484788105)

[2.4. Effiziente Datenübertragung 16](#_Toc484788106)

[2.5. Auswertung der Bodendaten 16](#_Toc484788107)

[2.6. Klimatische Daten der Externen Dienste 16](#_Toc484788108)

[2.7. Effiziente Erstellung interaktiver Tutorials 17](#_Toc484788109)

[2.8. Erstellung der Anleitungen für Mobiltelefone 17](#_Toc484788110)

[2.9. Didaktische Darstellung der Informationen 18](#_Toc484788111)

[3. Anforderungsanalyse 18](#_Toc484788112)

[3.1. User Profiles 18](#_Toc484788113)

[3.2. Personae 20](#_Toc484788114)

[3.3. Contextual Task Analysis 20](#_Toc484788115)

[3.3.1. Work Environment Analysis 22](#_Toc484788116)

[3.3.2. User Task Organization Model 22](#_Toc484788117)

[3.4. Systemanforderungen 23](#_Toc484788118)

[3.4.1. Funktionale Anforderungen 23](#_Toc484788119)

[3.4.2. Organisationale Anforderungen 24](#_Toc484788120)

[3.4.3. Qualitative Anforderungen 24](#_Toc484788121)

[3.5. Plattform Constraints 25](#_Toc484788122)

[3.6. General Design Principles 26](#_Toc484788123)

[3.7. Usability Goals 26](#_Toc484788124)

[3.8. Fazit 29](#_Toc484788125)

[4. Work Reengineering 30](#_Toc484788126)

[5. Conceptual Model Design 31](#_Toc484788127)

[5.1. Conceptual Mock – ups 33](#_Toc484788128)

[5.2. Evaluation 36](#_Toc484788129)

[6. Screen Design Standards 37](#_Toc484788130)

[6.1. Control Standarts 37](#_Toc484788131)

[6.2. Process window standarts 37](#_Toc484788132)

[6.3. Weitere Standarts 38](#_Toc484788133)

[6.4. SDS Prototyping 39](#_Toc484788134)

[7. Detaild UI Design 39](#_Toc484788135)

[7.1. Fazit 43](#_Toc484788136)

[8. Evaluation 43](#_Toc484788137)

[8.1. Evaluation anhand der Usability Goals und der Anforderungen 43](#_Toc484788138)

[9. Systemarchitektur 44](#_Toc484788139)

[9.1. Ressourcen 45](#_Toc484788140)

[9.2. Datenstrukturen 47](#_Toc484788141)

[9.3. Anwendungslogik des Systems 50](#_Toc484788142)

[9.4. Wichtige Anmerkungen 50](#_Toc484788143)

[9.5. Datenschutz 50](#_Toc484788144)

[10. Fazit 50](#_Toc484788145)

[10.1. Ausblick 50](#_Toc484788146)

[10.2. Persönliche Meinung 50](#_Toc484788147)

[11. Installationsdokumentation 51](#_Toc484788148)

[11.1. Systemanforderungen 51](#_Toc484788149)

[11.2. Server 51](#_Toc484788150)

[11.3. Client 51](#_Toc484788151)

[12. Anhang 52](#_Toc484788152)

[12.1. POCs (überarbeitet) 52](#_Toc484788153)

[12.1.1. Benutzer anlegen 52](#_Toc484788154)

[12.1.2. Collaborators dem Eintrag hinzufügen 52](#_Toc484788155)

[12.1.3. Effiziente Datenübertragung 52](#_Toc484788156)

[12.1.4. Auswertung der Bodendaten 53](#_Toc484788157)

[12.1.5. Effiziente Erstellung interaktiver Tutorials 53](#_Toc484788158)

[12.1.6. Klimatische Daten der Externen Dienste 53](#_Toc484788159)

[12.1.7. Erstellung der Anleitungen für Mobiltelefone 54](#_Toc484788160)

[12.1.8. Didaktische Darstellung der Informationen 54](#_Toc484788161)

[12.2. Personae 55](#_Toc484788162)

[12.3. Evaluation 57](#_Toc484788163)

[12.3.1. Tasks 57](#_Toc484788164)

[12.3.2. User 1 59](#_Toc484788165)

[12.3.3. User 2 60](#_Toc484788166)

[12.3.4. User 3 62](#_Toc484788167)

[12.3.5. User 4 62](#_Toc484788168)

Abbildungsverzeichnis

# Einleitung

## Nutzungsproblem

In vielen Entwicklungsländern ist die Bevölkerung sehr hoch und ein Großteil davon sind Kinder. Die Menschen leben in Armut und haben meist nur ungefähr einen US-Dollar pro Tag zum Leben zur Verfügung. Ca. 20% der dort lebenden Menschen sind stark unterernährt, was sich auf die dortige Armut zurückführen lässt. Oft wird dies durch mangelnde Aufklärung z.B. im Bereich Ackerbau verursacht, da das nötige Wissen für einen ertragreichen Anbau fehlt. Die Aufklärung der Bevölkerung wird jedoch dadurch erschwert, dass in diesen Entwicklungsländern ein Großteil der Menschen Analphabeten sind. So ist eine Aufklärung in Form von schriftlichen Informationen nicht möglich. Beispiel für diese Länder sind: Burkina Faso, Peru, Togo und Nepal. Die Bildungsbarriere zwischen den Entwicklungs- und den Industrieländern stellt ein erhebliches Problem dar.

## Ziel des Projekts

Ziel ist es ein System zu entwerfen, welches die Bevölkerung der Entwicklungsländer im Bereich Ackerbau aufklärt. Dabei soll es zum einen den Menschen direkt etwas vermitteln, beispielsweise durch Abbildungen und Erklärungen die als Ton abgespielt werden können. Zum anderen, soll es auch die Helfer vor Ort bei der Vermittlung wichtiger Strategien und Untersuchungen in diesem Bereich unterstützen. So können zum Beispiel ermittelte Werte des Bodens, gespeichert und verarbeitet werden, so dass eine Anbauempfehlung vorgeschlagen wird. Außerdem werden kurzfristige Informationen über bestimmte Ereignisse, z.B. das Wetter (Dürreperioden oder Niederschlag) an die Landwirte gesendet. Doch nicht nur der aktuelle Anbau soll gefördert werden, sondern das System soll die Landwirtschaft nachhaltig unterstützen, so dass auch langfristige Klimaveränderungen in Betracht gezogen werden. In Folge dessen, sollen die Ackerbauern darüber informiert werden, ob es beispielsweise noch Sinn macht Kakaopflanzen in den nächsten 5 Jahren anzubauen, oder ob sie auf eine andere Pflanze zurückgreifen sollen, da durch die Klimaveränderungen die Erträge einbrechen könnten.

Besonders wichtig ist es dabei, die Wissensbarriere zwischen den Industrie- und Entwicklungsländern zu schließen. Die Menschen in Afrika sollen von unserem Wissen profitieren können und daher werden die Informationen für sie gerecht aufbereitet.

# Proof of Concepts

Die Durchführung der POCs sollte wesentliche Merkmale des Systems in früheren Phasen der Entwicklung testen und die Realisierbarkeit des Systems gewährleisten. Im Folgenden wird die Durchführung der POCs dokumentiert. Die aus dem Konzept Überarbeiteten POCs sind in dem Anhang zu finden.

## Benutzer anlegen

Dieses POC ist relativ einfach realisierbar und aus dem Grund getestet, dass das Entwicklungsteam vorher keine Erfahrung mit der Userauthentifikation hatte. Das System erfordert wenige persönliche Datenfelder zu Registration. Falls der User kein Smartphone hat, wird er vom System über Telefonnummer identifiziert. Serverseitig werden Zur Authentifikation Module *Passport* und *Sessions* verwendet. Clientseitig erfolgt die Dateneingabe über die Textfelder der Android App. Wichtig ist zu beachten, dass das UI Concept für die Analphabeten akzeptable Informationspräsentationen enthält. Nach der Durchführung wurde festgestellt, dass die UI Elemente entsprechende visuelle Form haben oder verbal dargestellt werden sollen. Des Weiteren sollen die Helfer den Benutzern bei der Erstellung des Profils unterstützen.

Falls der Benutzer kein Smartphone besitzt und als Konsequenz kein Userprofil hat, sollte falls vorhanden seine Telefonnummer eingetragen werden. Falls der Benutzer gar kein Gerät besitz, ist die direkte Interaktion mit dem System nicht möglich. Daher werden alternative Authentifikationsmethoden betrachtet.

## Eintrag mit Ackerdaten erstellen

Erstellung des Eintrags ist eine weitere relativ einfache Input Funktion. Hier werden zur Verarbeitung relevante Ackerdaten eingegeben und in DB gespeichert. Scheitern könnte das POC bei der Erhebung der spezifischen Bodendaten vor Ort. Die Recherche während der Durchführung hat ergeben, dass es passende Werkzeuge gibt, mit den man die Werte vor Ort ermitteln kann. So gibt es z. B. einen kompakten PH – Wert Messer. Die Helfer müssen in dem Fall über solche Werkzeuge verfügen. Zudem gibt es mehrere APIs, die die Daten über Bodentemperatur - und Feuchtigkeit, Luftfeuchtigkeit usw. Regional zur Verfügung stellen. Im Fallback Fall kann man auf die APIs zugreifen. Die Datenvollständigkeit kann einfach auf dem Client vorm Speichern kontrolliert werden. Daher kann das System mit großer Wahrscheinlichkeit nicht an der Unvollständigkeit der Ackerdaten scheitern. Die Angabe der Daten erfolgt in der Regel vom Helfenden, so dass die eine gewöhnliche Darstellung der UI Elementen ausreicht.

Die Daten werden vom Client im JSON – Format an den Server übertragen, dort verarbeitet und in der Datenbank gespeichert.

## Collaborators dem Eintrag hinzufügen

Bei Speichern eines Eintrages kann der User, der den Eintrag erstellt, in der Datenbank nach anderen Usern suchen, um die dem Eintrag als sogenannte Collaborators hinzufügen. Die Collaborators bekommen somit die Zugriffsmöglichkeit auf den jeweiligen Eintrag. Der Kritische Punkt ist hier die Suche nach einem bestimmten User. Daher wird die Suche nach Email oder Passwort durchgeführt, sodass immer nur ein bestimmter User identifiziert wird. Die Durchführung hat ergeben, dass die Datenübertragung bei der Suche nach dem User sehr schnell passiert. Doch im echten Nutzungskontext des Systems herrschen andere Bedingungen, daher werden die Suchzeiten von 5-10 Sekunden akzeptiert.

## Effiziente Datenübertragung

Die Früheren Tests ergaben die Datenmengen bei einer Übertragung von ca. 10 – 30 kbps und die Übertragung erfolgte in wenigen Millisekunden. Die Daten wurden mit Android Studio Monitoringtools gemessen. Damit war die Durchführung des POCs erfolgreich. Um dies zu erreichen wurde die Repräsentation der Daten im JSON - Format so gestaltet, dass nur primitive Datentypen verwendet wurden. Das Modell der Tutorials wurde z.B. so entworfen, dass die Abweichung der Bodenwerte nach der Bodenanalyse in einer Art Flags gespeichert wurden. Auf diese Weise konnte z.B. der PH – Wert auf NORM, GREATER oder LESS gesetzt werden, wobei den Konstanten NORM, GREATER, LESS ein integer wert zugewiesen ist und dem System bekannt ist, wo es Abweichungen gibt. Dies wurde für die Testzwecke und aus Zeitgründen sehr vereinfacht dargestellt und wird in folgenden Entwicklungsschritten weiter ausgebaut.

## Auswertung der Bodendaten

Die Bodenanalyse umfasst den Vergleich der angegebenen Daten mit den normalen Standartwerten. Bei Abweichungen werden die Gegenmaßnahmen berechnet und in Form von Tutorials in der Datenbank gespeichert. Der Client kann im Folgenden auf die Tutorials zugreifen. So wird es z.B berechnet, wie viel und wie oft man eine Pflanze gießen soll in Abhängigkeit von den für die Pflanze bestimmten Standartwerten und der Fläche des Ackers, Art des Bodens, der Wetterbedingungen usw.

Bei der Durchführung wurde festgestellt, dass es keine passende Pfalzen Datenbank existiert, die direkt über ein API im System zur Bodenanalyse verwendet wird. Somit ist eine Datenbank zu vergleich der Werte notwendig. Daher wird Erstellung einer eigene Pflanzentopologie überlegt, um die für die Bodenanalyse relevanten Daten zu standarisieren und die Vollständigkeit zu gewährleisten.

## Klimatische Daten der Externen Dienste

Harvest Hand benötigt für eine vollständige Funktionalität Daten zur lokalen Wetterlage des Benutzers. Sie nehmen immerhin Einfluss auf den Anbauvorgang. Daher muss eine Wetter API implementiert werden, welche Daten zum Wetter zur Verfügung stellt. Als erstes musste viel Zeit in die Recherche nach einer geeigneten API investiert werden. Einige sind nicht frei verfügbar, andere gewährleisten nur eine bestimmte Anzahl von Abfragen in einem bestimmten Zeitraum.

Abschließend wurde sich für die Weather Underground API entschieden. Sie bietet mehr Daten als andere APIs an, es können 10 Abfragen pro Minute und 500 pro Tag getätigt werden. Dies ist als freie Version gegenüber anderen APIs eine hohe Anzahl von Abfragen pro Tag. Außerdem war ein ausschlaggebendes Kriterium, dass wir bereits das Wetter für die nächsten 10 Tage voraussagen können.



Abbildung 1 Leistungen Weather Underground API

Um einen Key für die Nutzung zu erhalten, wurde ein Konto angelegt bei welchem Daten zum Projekt und dem Nutzen der API angegeben werden mussten. Anschließend wurde der Key direkt mitgeteilt und die API konnte implementiert werden.

API Requests funktionieren ausschließlich über HTTP. Beispielhaft wurde ein Request für die Stadt San Francisco durchgeführt.



Abbildung 2 HTTP Request

Die Daten werden im JSON Format präsentiert. Man erhält zum einem ausführliche Daten zur Lokalität und zum anderen detaillierte Wetterdaten. Aus dieser großen Menge an Daten wurden die für das System relevanten Daten herausgefiltert. Der Anbieter bezieht die Daten von einigen Wetterstationen, sodass die Datenvalidität gewehrleistet werden soll. Des Weiteren wurden zahlreiche weitere Wetter – APIs betrachtet, damit alternative vorliegen.

## Effiziente Erstellung interaktiver Tutorials

Für die Ermittlung der Daten wurde ebenso Android Studio Monitoringtools verwendet. Das POC wurde durch Darstellung der Tutorials mit Bildern getestet. Die Materialien werden Dabei auf dem Lokalen Speicher des Geräts abgelegt und bei Verwendung in den RAM geladen. Daher ist besonders wichtig darauf zu achten, dass die Materialien in möglichst komprimierter Form vorliegen und nicht zu viele Ressourcen verbrauchen. Die Durchführung lieferte akzeptable werte, sodass das POC als erfolgreich gilt. Bei der RAM Nutzung wurden Werte im Bereich zwischen 5 bis 40 MB festgestellt von dem Start des Programms bis zur Darstellung der Tutorials. Die Prozessorbelastung lag im Schnitt Bei 10 Prozent.

## Erstellung der Anleitungen für Mobiltelefone

Für die Erstellung und das Verschicken der SMS – Nachrichten wurde ein API zu einem SMS Gateway getestet. Der Inhalt der SMS – Nachrichten wird im JSON – Format mit einem Request inklusive der Zielnummer an den Dienst geschickt und kann von dort weltweit weiter übermittelt werden. Zum Testen wurde API von websms (websms.de) verwendet. Der kritische Punkt war, dass dieses Unternehmen kostenlos nur eine 14 Tage Testversion von dem Dienst anbietet. Zur permanenten Nutzung dieser Funktion sollte ein anderer Anbieter recherchiert werden.

## Didaktische Darstellung der Informationen

Um dieses POC zu Testen wurden Information mit passenden Bildern dargestellt. Es wurden Icons erstellt um die Information möglichst deutlich für die Analphabeten präsentieren zu können. So wurden z.B. Buttons mit den Icons versehen. Des Weiteren wurde festgestellt, dass bestimmte Elemente sich nicht in visueller Form darstellen lassen. In dem Fall wurde als Lösung eine verbale Präsentation des Elements entwickelt. Dabei kann bei Interaktion mit dem Element sein Inhalt oder Funktionalität von der App vertont werden. Im Android System eignet sich ideal hierfür die Klasse **MediaPlayer**. Die Klasse ist dem Entwicklerteam von früheren Projekten bekannt und wurde schon zur Vertonung bestimmter Elemente verwendet. Daher wird dieses Konzept auch für dieses Projekt übernommen.

Im Prozess der Durchführung des POCs wurde die **TextToSpeech** Klasse von Android entdeckt, die es ermöglicht dynamische Textinhalte auf dem Bildschirm vorzulesen. Die Implementierung der Klasse ist wesentlich einfache und die Klasse ist flexibler. Unten ist ein Beispiel vorgeführt, wie man den Inhalt eines **TextViews** beim Klicken vorlesen lässt. Der Text kann zu dem in verschiedenen Sprachen vorgelesen werden. Da ein **onClickListener** praktisch an jedes Element angebunden werden kann, können auch Inhalte von Bildern vertont werden, um die für die Analphabeten noch deutlicher zu präsentieren.

textView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String toSpeak = "Speak";  
 textToSpeech.speak(toSpeak, TextToSpeech.*QUEUE\_FLUSH*, null);  
 }  
});

# Anforderungsanalyse

## User Profiles

**Landwirt**

|  |  |  |
| --- | --- | --- |
| Merkmal | Ausprägung | |
|  | **Alphabet** | **Analphabet** |
| Demographische Charakteristiken | Jeden Alters und Geschlechts, sind meistens Familienbetriebe, Wohnhaft in einem Entwicklungsland | Jeden Alters und Geschlechts, sind meistens Familienbetriebe, Wohnhaft in einem Entwicklungsland |
| Qualifikationen | Haben meistens mittlere schulischen Qualifikationen vorzuweisen | Haben meistens keine schulischen Qualifikationen vorzuweisen |
| Fachwissen | vorhanden, bisheriges Wissen basiert auf Tradition und erlernten Qualifikationen, ist aber oft nicht auf dem neuesten Stand | Kaum vorhanden, bisheriges Wissen basiert auf Tradition, ist aber oft nicht auf dem neuesten Stand |
| Fähigkeiten bzw. Einschränkungen | Können Lesen und Schreiben, sind meist in der Lage körperliche Arbeit zu verrichten | Fehlende Lese- und Schreibkompetenz (Analphabetismus), sind meist in der Lage körperliche Arbeit zu verrichten |
| Verfügbare Technologien | Smartphones, Tablets, ältere Mobiltelefone | Smartphones, ältere Mobiltelefone |
| Computerkenntnisse bzw. -erfahrung | Teilweise vorhanden | Kaum vorhanden |
| Motivation | Ernteerträge steigern und somit ihre Lebensumstände verbessern, Qualifikationen steigern | Ernteerträge steigern und somit ihre Lebensumstände verbessern |
| Produkterfahrung | Keine, benötigt Einführung in das System | Keine, benötigt Einführung in das System |
| Aufgaben | Anbauempfehlungen und Tipps und relevante Daten zum Ackerbau erhalten | Anbauempfehlungen und Tipps zum Ackerbau erhalten |
| Auswirkung von Fehlern | Niedrige Ernteerträge oder ganze Ernteausfälle, sorgen für Verschlechterung der Lebensumstände, Wissenstand wird nicht verbessert | Niedrige Ernteerträge oder ganze Ernteausfälle, sorgen für Verschlechterung der Lebensumstände |
| Einstellungen und Werte | Technische Lösungen sind bekannt | Sind nicht den technischen Lösungen zugeneigt, die Sitten der Vorfahren haben größeren Wert |

**Helfer**

|  |  |
| --- | --- |
| Merkmal | Ausprägung |
| Demographische Charakteristiken | Über 18 Jahre, können männlich und weiblich sein, jeder Familienstand ist möglich, Wohnhaft in einem Industrieland |
| Qualifikationen | Abgeschlossene Berufsausbildung oder Studium im Bereich Landwirtschaft |
| Fachwissen | Allgemeines Wissen über die Funktionsweise in der Domäne |
| Fähigkeiten bzw. Einschränkungen | Können Fachwissen vermitteln, allerdings nicht in afrikanischer Sprache oder nicht in für Analphabeten akzeptabler Form |
| Verfügbare Technologien | Smartphones, Tablets, Computer |
| Computerkenntnisse bzw. -erfahrung | Sehr gut |
| Motivation | Berufliche Kompetenz, den Menschen in Entwicklungsländern Wissen vermitteln und ihr Leben somit zu verbessern. |
| Produkterfahrung | Vorhanden, wurde zuvor genau in das System eingeführt um das Lehrmaterial anwenden zu können |
| Aufgaben | Den Helfer bei der Vermittlung von Wissen durch Visualisierungen, Animationen und Audio unterstützen. |
| Auswirkung von Fehlern | Die Wissensvermittlung kann nicht vollständig stattfinden. |
| Einstellungen und Werte | Sind den technischen Lösungen zugeneigt, bevorzugen Verwendung von IT-Systemen |

## Personae

Da die Stakeholderanalyse und Userprofiles ein sehr breites Spektrum an unterschiedlichen Benutzern geliefert haben, wurden die Personae von erstellt, um ein klareres Bild von den potenziellen Benutzern des System zu machen. Die vorherigen Analysen ergaben zwei Schlüsselakteure – Helfer der Hilfsorganisationen und Landwirte in den Entwicklungsländern. Doch es gibt verschiedenste Organisationen und verschiedene Entwicklungsländer, in denen unterschiedliche Lebensbedingungen herrschen. Um die Benutzer besser zu verstehen und eine gewisse Abgrenzung zu schaffen, eignen sich optimal die Personae. Die Personae findet man im Anhang 9.2

## Contextual Task Analysis

Die Contextual Task Analysis soll Bedürfnisse und Wünsche der Benutzer an das System identifizieren. Dafür sollte das Umfeld der Benutzer betrachtet werden und es können Interviews mit zukünftigen Benutzern geführt werden, um Erfordernisse identifizieren zu können.

Im Bereich dieses Nutzungskontexts ist es dem Entwicklerteam nicht möglich, den Alltag der Benutzer genau zu beobachten und zu analysieren oder ein Interview mit zukünftigen Nutzern zu führen. Daher werden im Folgenden die Tasks anhand der angelegten User Profiles abgeleitet.

Aus der Stakeholderanalyse lassen sich zwei primären Stakeholdergruppen identifizieren – Helfer (von Hilfsorganisationen) und Landwirten der Entwicklungsländer. Im ersten Schritt wurde der Arbeits - und Kommunikationsumfeld dieser beiden Gruppen beobachtet und Informationen gesammelt. Da die direkte Konfrontation mit den primären Usern nicht möglich ist, müssen die Information aus den Recherchen bezogen werden. In der folgenden Tabelle werden die gesammelten Infos dargestellt.

Tabelle 1 Data Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actor | Trigger | Use Case(Task) | Task Scenario Sequence | Errors, Problems, Comments |
| Landwirt | Pflanzenwachstum steigern | Landwirte müssen Wasser aus den Brunnen holen, welches sie zu ihren Feldern transportieren. | 1. Bodentrockenheit visuell feststellen  2. Wasser aus dem Brunnen beziehen  3. Behälter zu den Feldern transportieren  4. Pflanzen gießen | Bodentrockenheit kann nur an der Oberfläche festgestellt werden, die benötigte Wassermenge kann nicht bestimmt werden |
|  |  | Pflanzen benötigen Dünger um den niedrigen Nährstoffgehalt im Boden ausgleichen zu können. | 1. geringes Pflanzenwachstum wird festgestellt  2. Düngemittel werden ggf. gekauft  3. Düngemittel werden auf den Äckern verteilt | Landwirt weiß nicht, wie und womit er düngen soll und in welcher Menge. |
|  | Qualitativ hochwertige Pflanzen ernten | Pflanzen werden nach der Reifezeit geerntet. | 1. Pflanzen werden auf Reife geprüft  2. Pflanzen werden zu Nahrung zubereitet oder auf dem Markt verkauft | Fehlendes Wissen zu Reifegraden von Pflanzen |
|  | Fachwissen erweitern | Neue Techniken und Methoden zum Ackerbau erlernen | 1. Informationen sichten  2. Informationen verinnerlichen und verstehen | Analphabeten können nur erschwert Informationen aufnehmen. |
|  | Äcker nachhaltig und effizient bearbeiten | Erlernen neuer Methoden um Felder zu bearbeiten, da Maschinen fehlen. | 1. neue Methoden kennenlernen  2. Methoden anwenden | Alle Äcker können nur per Hand bearbeitet werden, was oft sehr aufwendig und mühsam ist. Oft sind Böden steinig und halten kaum Wasser. |
|  | Wetterunabhängiger Anbau | Nach starken Regenfällen den Böden Nährstoffe zuführen. | 1. starke Regenfälle bemerken  2. Äcker düngen | Viele Böden sind unfruchtbar, da durch die starken Regenfälle die Nährstoffe ausgewaschen werden. |
|  | Unfruchtbare Böden wieder Fruchtbar machen | Bodenfruchtbarkeit wiederherstellen. | 1. Boden düngen  2. Boden wässern  3. Schädlingsbekämpfung | Viele Ackerflächen werden bis auf ihre letzten Nährstoffe ausgeschöpft. Anschließend liegen diese Flächen 30 Jahre brach bevor neue Pflanzen angebaut werden können. |
|  | Unabhängigkeit vom Staat | Genügend Pflanzen zum Ernähren der Familien und dem Verkauf auf dem Markt erzeugen. | 1. Pflanzen ernten  2. Familien ernähren  3. übrige Ernte verkaufen | Die Entwicklung einer gesunden Landwirtschaft wird verhindert, in dem Subventionen ohne Ziel und zeitlichen Rahmen verteilt werden. |
| Helfer | Gefährdete Existenz der  Landwirthaushalte | Unterstützen der Bauer bei Findung und Umsetzung einer Lösung | 1. Problem identifizieren  2. Informationen zum bestimmten Problem beziehen  2. Passende Lösung Entwickeln  3. Lösung umsetzen | Die Informationen zu den verschiedensten Pflanzenarten müssen gesammelt werden. Individuelle Maßnahmen müssen erarbeitet werden. Ständige Begleitung der Landwirte nötig |
|  | Wissensbarriere schließen | Fachwissen vermitteln in Form von Schulungen vor Ort | 1. Informationen vermitteln  2. Lehrmaterial zur Verfügung stellen | Helfer und Landwirte sprechen unterschiedliche Sprachen. Schriftliche Informationen müssen für Analphabeten in ein Format übertragen werden, welches sie verstehen. |
|  | Schulung | Übermittlung des Fachwissens in einer Lerngruppe | 1. Aufbereitung der Informationen im Vorfeld  2. Lerngruppen werden organisiert  3. Information wird in visueller und verbaler Form übermittelt | Lern Gruppen müssen organisiert werden. Die Informationen müssen in einer für Alphabeten und Analphabeten geeigneter Form dargestellt werden |

### Work Environment Analysis

kek

### User Task Organization Model

Anhand der gesammelten Daten wird schließlich das User Task Organization Model erstellt, um die Aufgaben der Primären Stakeholder zu veranschaulichen. Im Mittelpunk der Aufgabenorganisation steht die Wissensvermittlung zwischen Helfern und Landwirten. Die Helfer müssen hierfür die Informationen zur Pflanzen aufbereiten und durch die Schulungen sie in geeigneter Form den Landwirten übermitteln. Zur Aufgaben der Helfer gehört ebenso die Überzeugung der Bauer in dem, dass die Bauer in der Lage sind, mit der Unterstützung der Helfer ihre Lebenslage nachhaltig zu ändern und Ernteerträge zu steigern. Das geplante System sollte als Schnittstelle zur Wissensübermittlung zwischen den Helfern und Landwirten agieren und ebenso die Landwirte unabhängiger von der externen Unterstützung machen, indem sie Informationen von dem System beziehen. In folgender Abbildung wird das User Task Organization Model dargestellt.

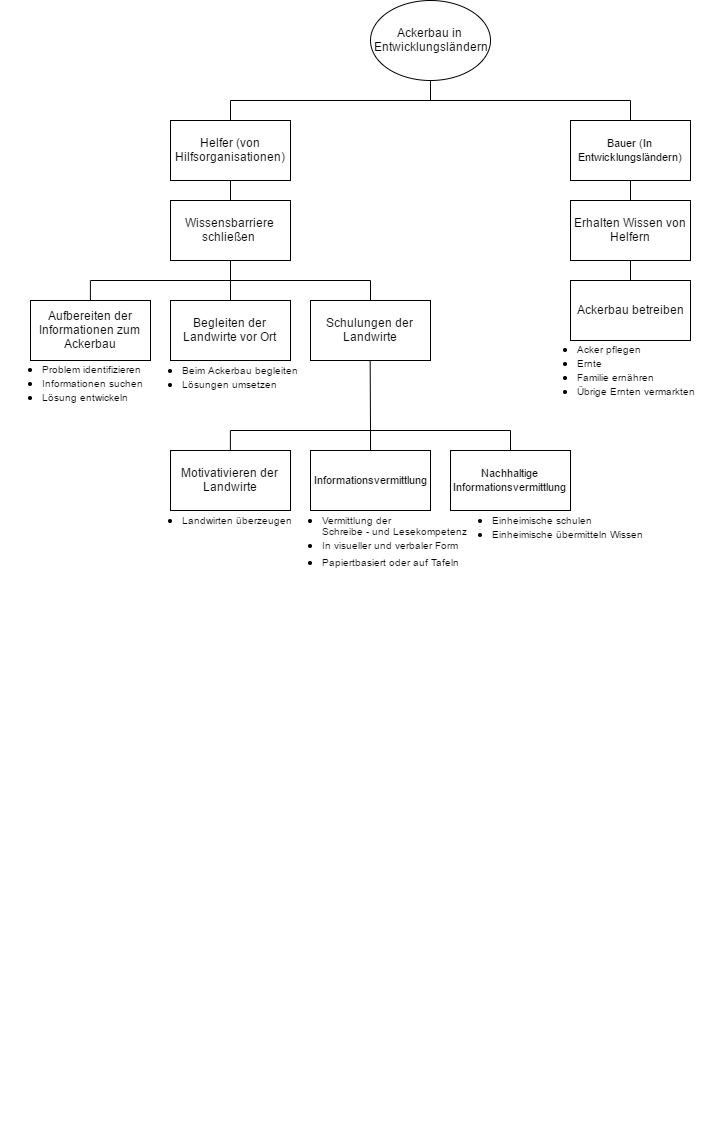


Abb. 1 User Task Organization Model

## Systemanforderungen

Im Folgenden werden Anforderungen anhand der in vorherigen durchgeführten Kontextanalysen an das System bzw. an das Projekt spezifiziert. Abweichend von dem Mayhew Modell werden nicht nur qualitative, sondern auch funktionale und Organisationale Anforderungen aufgelistet.

### Funktionale Anforderungen

F10 Das System muss den Usern verschiedene Möglichkeiten zur Authentifikation bieten, damit Benutzer mit verschiedenen Kompetenzen die Zugriffsmöglichkeiten bekommen

F20 Sobald der Helfer sich im System authentifiziert hat, muss das HarvestHand System Ihm die Möglichkeit bieten, Daten eines Ackers einzutragen

F30 Das System muss dem Helfer/Landwirt die Möglichkeit bieten jederzeit Daten von Ackerflächen zu protokollieren und zu verwalten

F40 Das System muss die Möglichkeit bieten, den Landwirt oder den Helfer einem Eintrag als Collaborator zu hinzufügen, damit er die Zugriffsmöglichkeiten auf einen Eintrag bekommt

F50 Das System muss Daten eines Ackers analysieren um eine passende Anbauempfehlung zu erstellen. Für die Analyse müssen relevante Bodendaten im System verfügbar sein

F60 Das System muss Helfer/Landwirte ermöglichen Daten zu den Äckern jederzeit zu aktualisieren, um passendes Wissen vermitteln zu können. Diese Funktionalität muss 24 Stunden am Tag zur Verfügung stehen und fehlerfrei ablaufen

F70 Das System muss Landwirten als Folge einer Bodenanalyse Anbauempfehlungen präsentieren. Dabei soll nach jeder Analyse mindestens eine Empfehlung präsentiert werden

F80 Die Helfer müssen Möglichkeit haben, auf eine Anbauempfehlung zuzugreifen

F90 Das System soll Landwirten Benachrichtigungen über vorhersehbare Wetterereignisse zur Verfügung stellen. Dabei sollen Warnungen vor extremen Wettereignissen mindestens 24 Stunden vor dem Eintreffen herausgegeben werden, um Vorsichtsmaßnahmen ergreifen zu können

F100 Das System muss den Helfern/Landwirten ermöglichen die Wissens- bzw. Sprachbarriere zu überwinden. Dabei soll auf textuelle Informationen möglichst verzichtet werden und Informationen über Visualisierungen, Animationen und Audio vermittelt werden

F110 Das System muss den Helfern ausführliche Informationen zum Ackerbau in verschieden Medienformaten zur Verfügung stellen. Dabei sollte zu jedem Thema mindestens eine Visualisierung oder eine Audiodatei zur Verfügung stehen, um damit das Wissen an die Landwirte weiterzugeben

F120 Das System muss den Benutzern aktuelle und valide Daten zum Ackerbau zur Verfügung stellen. Daher sollte in regelmäßigen Abständen von ca. 1 Monat die Daten auf Aktualität überprüft werden

F130 Das System muss Helfern und Landwirten eine Pflanzendatenbank mit relevanten Informationen zur Verfügung stellen. Dabei müssen zu jeder Pflanze mindestens Angaben zur Bewässerung, Bodenbeschaffenheit, Temperatur, Lichteinstrahlung, Wachstumszeit, Erntezeit und Nährstoffbedingungen zur Verfügung stehen

F140 Das System soll fähig sein, den Landwirten ohne Smartphones die Informationen zum Ackerbau zu Verfügung zu stellen.

### Organisationale Anforderungen

### Qualitative Anforderungen

* Alle funktionalen Anforderungen sollten am Ende des Projekts erfüllt sein, damit die Funktionalität des Systems nicht eingeschränkt wird und zu 100% gewährleistet werden kann.
* Das System sollte fehlerfrei und zuverlässig laufen. Bei 20 Testläufen sollte maximal einmal ein Problem auftauchen.
* Zur Erstellung eines Userprofils sollen nur minimal erforderliche Daten eines Users vom System erhoben werden. Maximal 5 Datenfelder sollen dem System übermittelt werden
* Das System soll 24/7 verfügbar sein, damit sich zu jeder Tageszeit zum Thema informiert werden kann.
* Das Nutzungsproblem muss mit Hilfe des Systems gelöst werden. Um dies überprüfen zu können, sollen User in Systemevaluation involviert werden. Dabei sollen mindestens 90% der Probanden den Inhalt des Systems erfassen können.
* Das System muss Erfordernisse/Erwartungen/Interesse der Stakeholder erfüllen. Dabei müssen vor allem alle Erfordernisse/Erwartungen/Interessen der Hauptakteure Helfer und Landwirte berücksichtigt werden. Alle Punkte dieser beiden Akteure müssen im System umgesetzt werden.
* Die Kommunikation zwischen den Komponenten muss effizient realisiert werden, die Datenübertragungszeiten sollen nicht 10 Sekunden überschreiten um eine akzeptable Reaktionszeit zu gewährleiten.
* Es müssen nur geringe Datenmengen verarbeitet werden, um eine effizienten Datenaustausch gewährleisten zu können. Nicht mehr als 100 KB pro Anfrage/Antwort sollten übertragen werden.
* Das System muss effizient die Ressourcen des Endgeräts verwenden um die Leistungsfähigkeit des Gärest nicht zu beeinträchtigen. Es soll nicht mehr als 50-100 Mb RAM verbrauchen und die Prozessorbelastung sollte 10% nicht überschreiten.
* Das System soll skalierbar sein, so dass weitere Erweiterungen effizient durchgeführt werden können.
* Das UI muss einen hohen Grad der Usability aufweisen. Diese muss durch die Evaluation des Systems gewährleistet werden. Dabei müssen mindestens 90% der User das System nachvollziehen und Inhalte verstehen können.
* Das System soll die in den Entwicklungsländern meistgesprochenen Sprachen unterstützen. Minimum Englisch, Französisch und Spanisch sollen verfügbar sein

## Plattform Constraints

Als Client agieren Im System Android Smartphones und mobile Telefone, was sehr unterschiedliche Plattformeinschränkungen ergibt. Android - Plattform und ihre Beschränkungen ist dem Entwicklerteam bereits bekannt. Im Folgenden werden wesentliche Einschränkungen der Zielplattformen aufgelistet.

**Android Client**

* Herstellerspezifische Ansätze. Unter Android laufen unterschiedlichsten Geräten mit verschiedene Spezifikation und Physischen Daten, so dass die Evaluation des Systems auf allen möglichen Geräten nicht realisierbar ist.
* Unterschiedliche Android Versionen. Je älter die Versionen, desto mehr Einschränkungen gibt es
* Die Größen des Bildschirms variieren sehr stark. Aus Nutzungskontextanalyse geht hervor, dass die Benutzer des Systems zum größten Teil billigere Smartphone mit kleineren Displays besitzen. Die Displaygröße von 3,7-4 Zoll wird als Standartgröße betrachtet
* Begrenzte Akkulaufzeit
* Begrenzte Leistungsfähigkeit des Endgeräts
* In bestimmten Fällen unstabile Datenverbindung oder gar keine Verbindung
* Begrenzter Datenvolumen
* Es muss in Java Programmiert werden

**Mobile Telefone**

Mobile Telefone sind in Ihrer Funktionalität stark eingeschränkt und werden. Diese Plattform wird als Schnittstelle des System für die User benutzt, die kein Smartphone Besitzen. Über Mobiltelefone ist die interaktive Kommunikation des Benutzers mit dem System nicht möglich. Im Folgenden werden wesentliche Einschränkungen genannt.

* Installation der App nicht möglich
* Kommunikation zwischen Server und Client nur über SMS möglich
* Die Datenübertragungsmenge der SMS – Nachrichten begrenzt
* Inhalt der SMS – Nachrichten meistens auf Text begrenzt
* Verschicken der SMS ist kostenpflichtig
* Gewisse Modelle können Farben nicht unterstützen

Zusammenfassend sollte bei der Entwicklung des Android Clients im Wesentlichen darauf geachtet werden, dass rechenaufwendige Algorithmen auf den Server überlagert werden sollen und der Datenaustausch mit dem Server effizient gestaltet werden soll. Die Mobiltelefone fordern eine geeignete Präsentation der Information, die via SMS übertragen werden kann.

## General Design Principles

Bei der Entwicklung der General Design Principes empfiehlt Mayhew für das Produkt relevante allgemeine Literatur und Guidelines des Usanility Engineerings. Da es eine Android App entwickelt werden soll, werden die Design Prinzipien von der Android Developer Webseite verwendet. Android setzt auf Material Design und beschreibt sehr detailliert die Prinzipien der Gestaltung der UI – Komponenten wie Navigation, Icons, Textes, Fonts, Animationen usw. Android Design Principles sollen helfen gebrauchstaugliche Produkte zu entwickeln und werden im Projekt eingehalten.

## Usability Goals

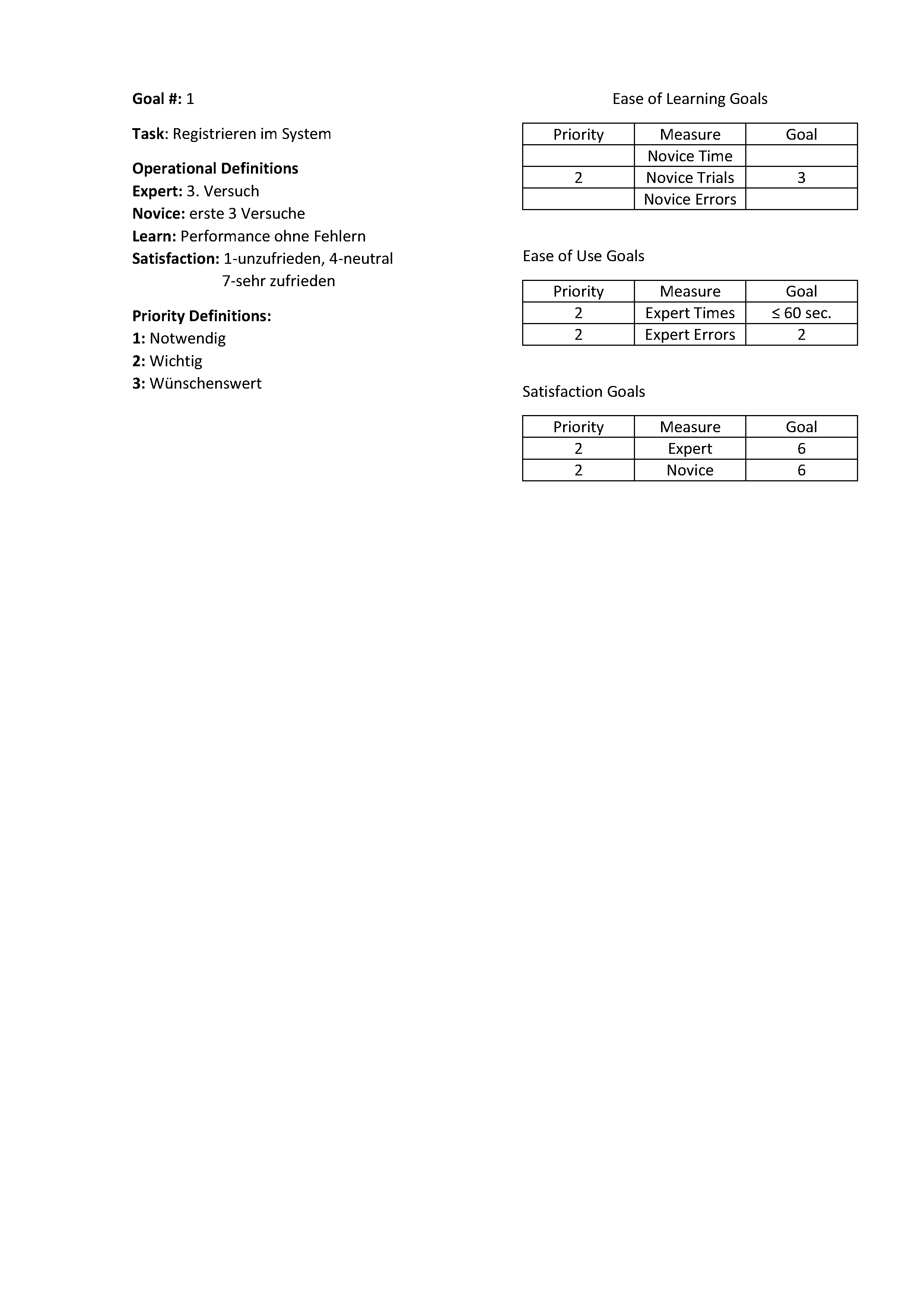
Schließlich werden die Usability Goals als Bewertungskriterium spezifiziert. Die Usability Goals werden aus der Task Analysis, den Userprofiles und weiteren Erkenntnissen der Recherchen abgeleitet.

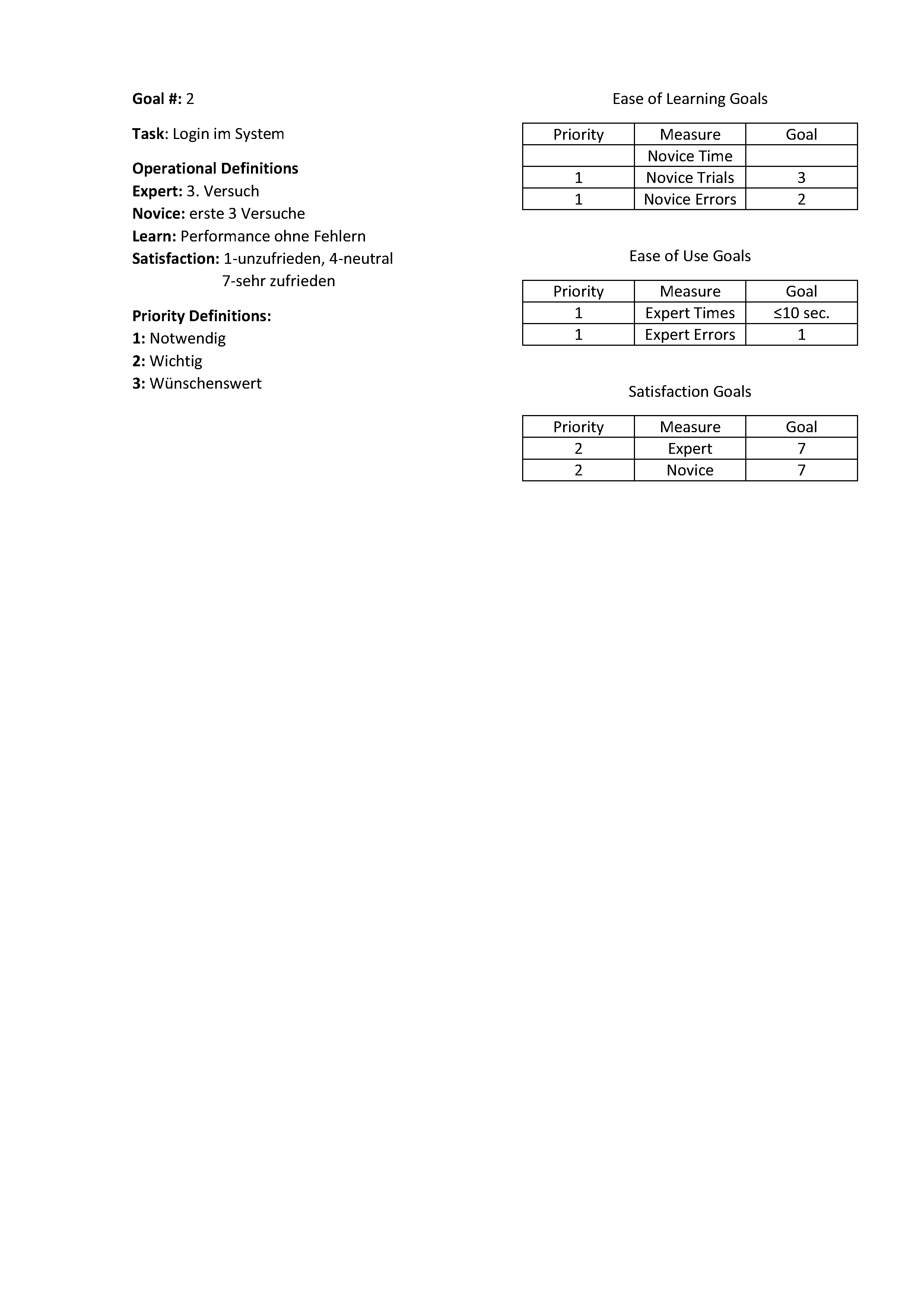
Qualitative Usablility Goals

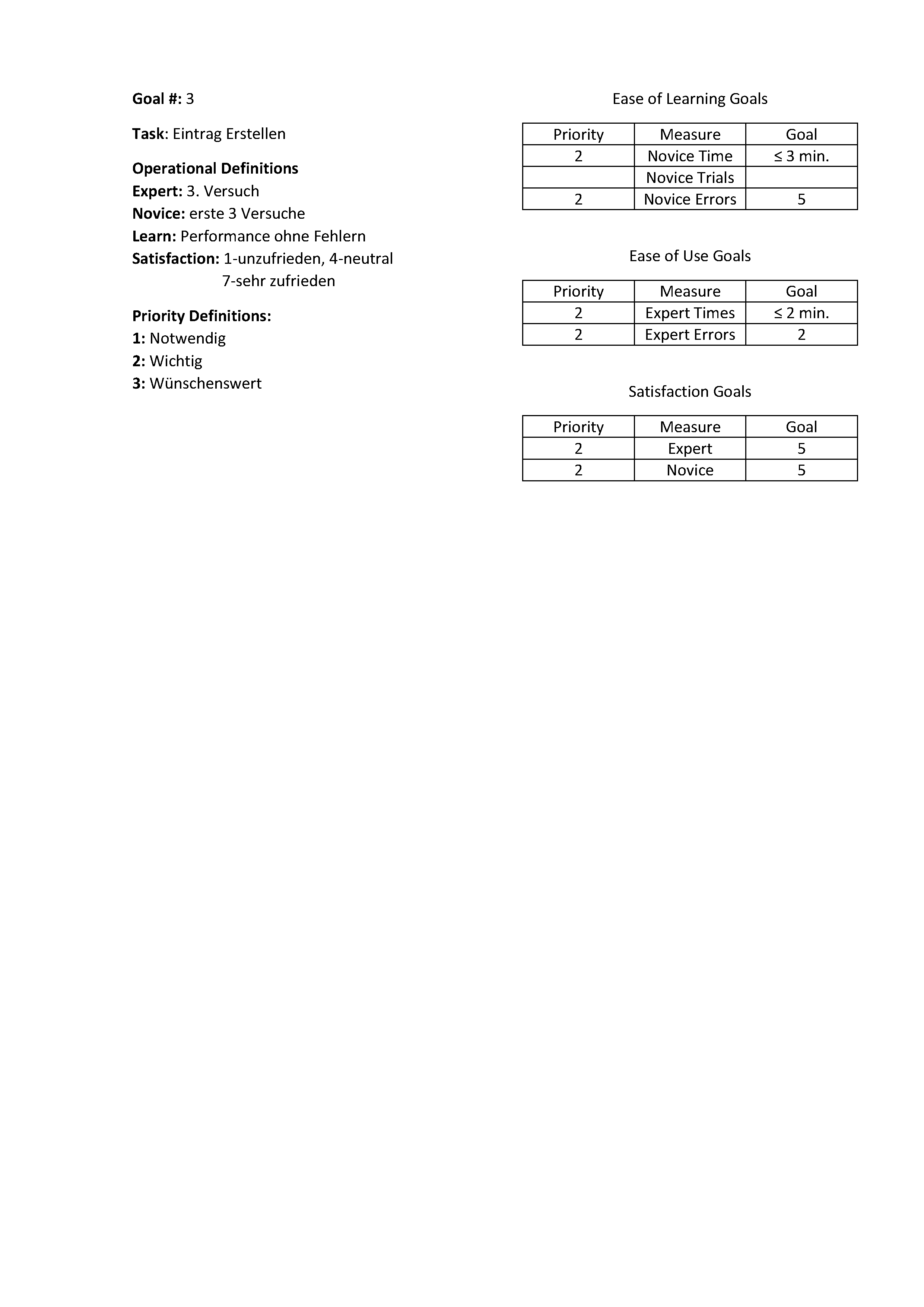
* Das System muss Lernprozesse des Ackerbaues fördern
* Die Landwirten sollen zum selbständigen Ackerbau motiviert werden
* Das System muss die täglichen Task der User unterstützen, ohne dass ihr Wissen überfordert wird
* Das UI muss nur für die User – Task relevante Information beinhalten

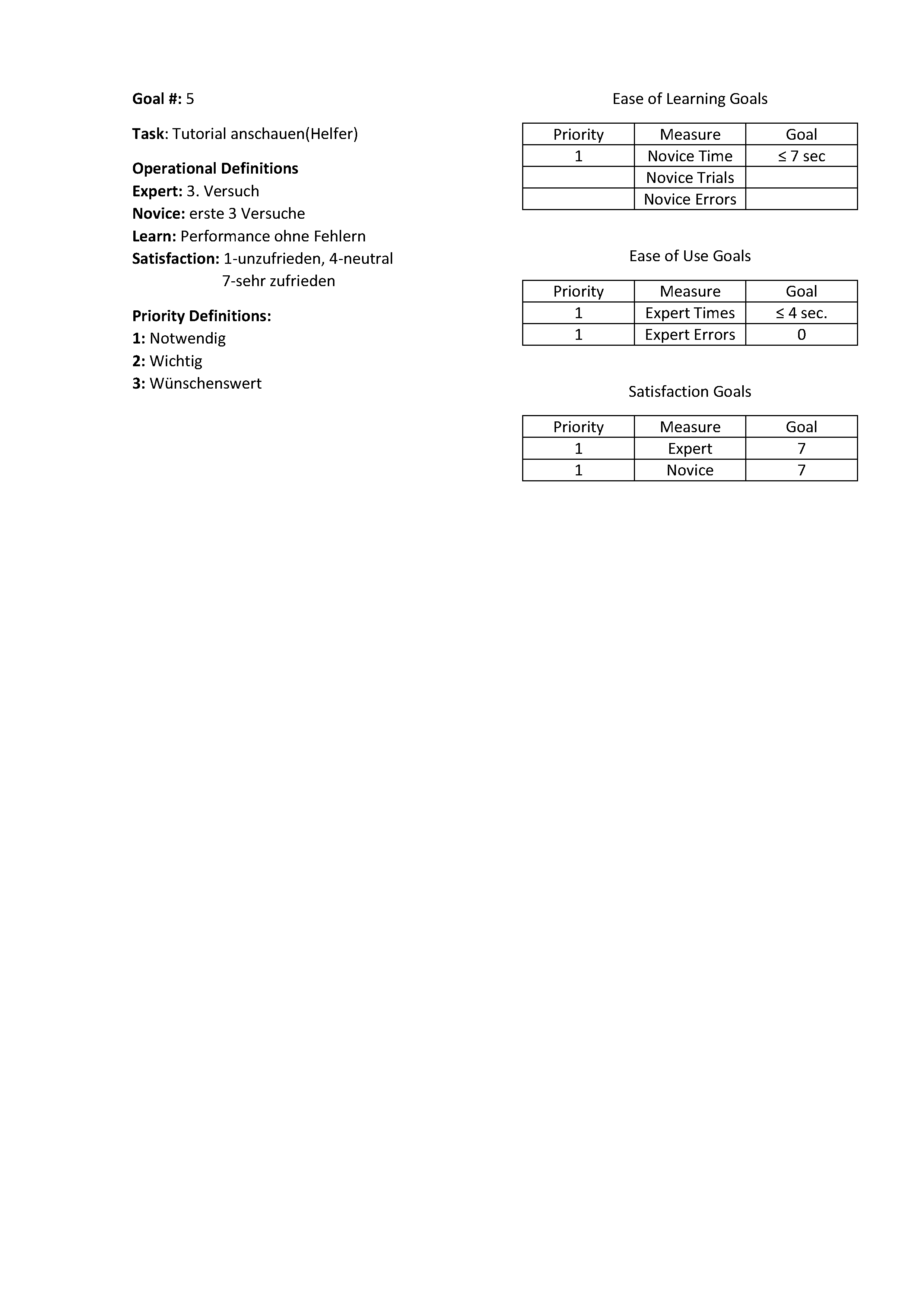
Quantitavie Usablility Goals

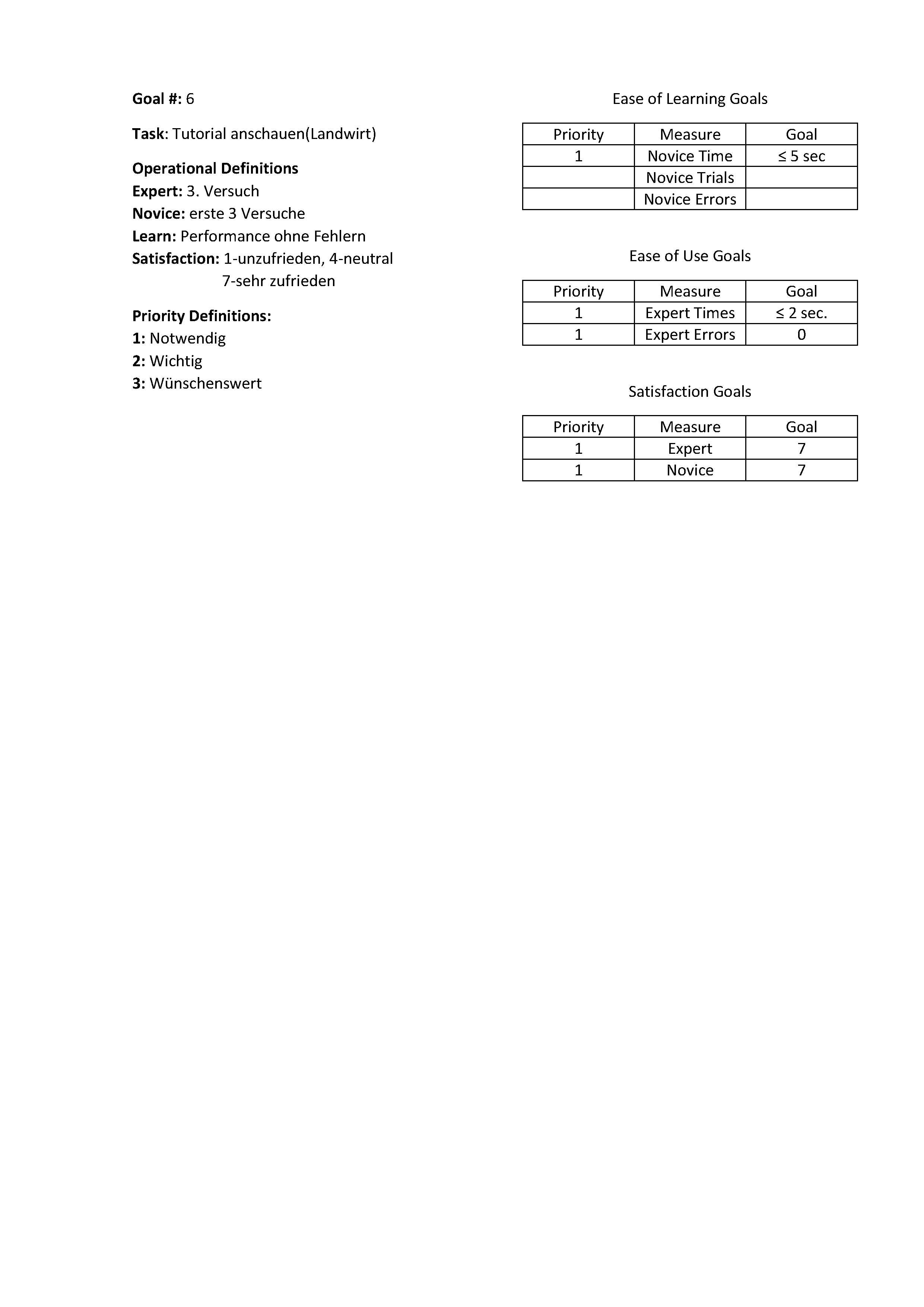
Alle Zeit Angaben (außer Login/Registration) gelten für den Fall, dass das System schon gestartet ist und der Benutzer bereits eingeloggt ist.

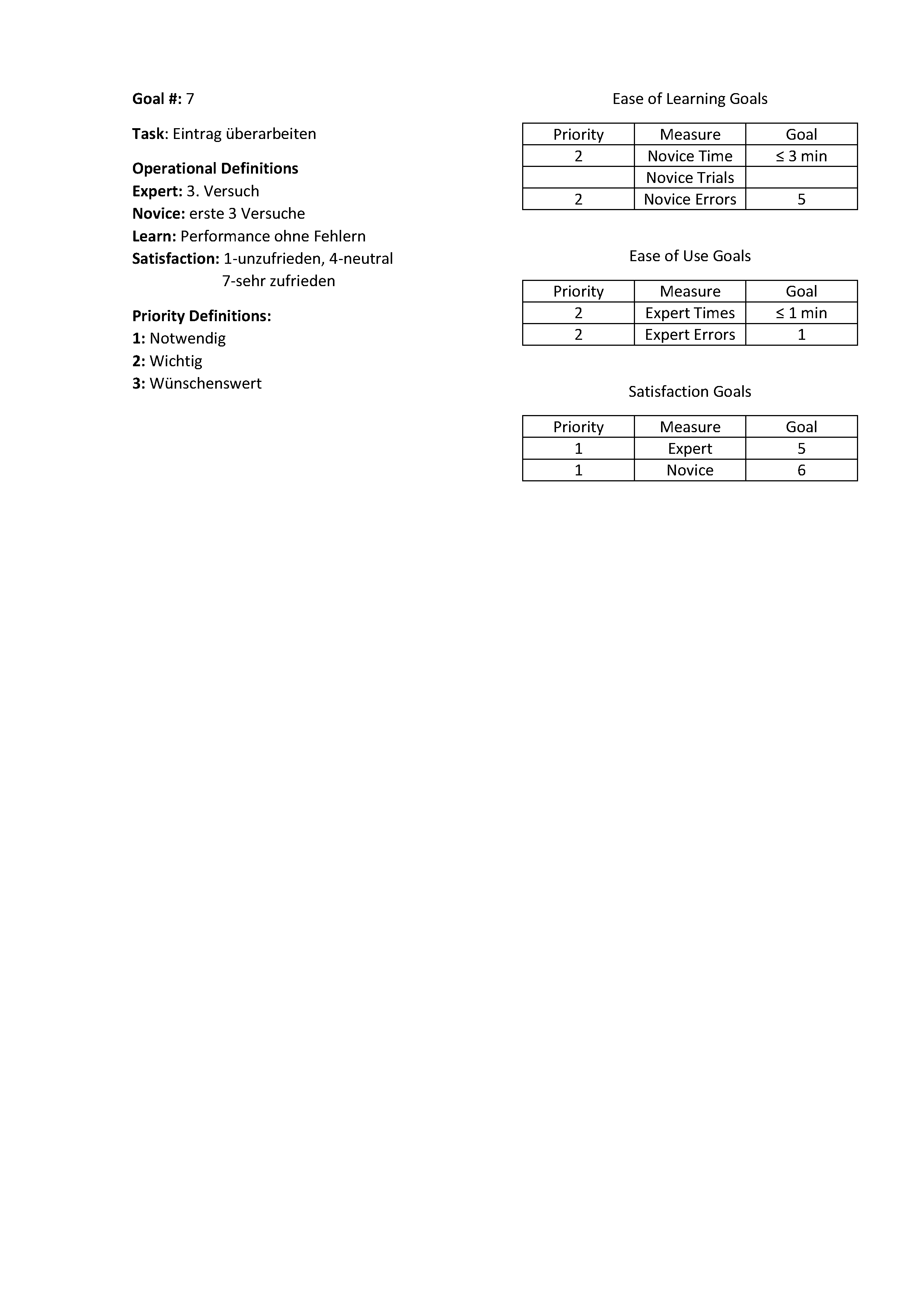












## Fazit

Die Requirements analysis ergab wichtige Erkenntnisse für das System. Die wichtigsten Stakeholder des Systems sind Helfer und Landwirten. Dennoch besonders die sehr vielfältigen Merkmale der Landwirte machen die Arbeitsreorganisation sehr kompliziert. Hier müssen variierende Schreibe – und Lesekompetenzen der Landwirte beachtet werden. Entsprechend den Kompetenzen der Stakeholder muss auch die Präsentation der Systeminformationen angepasst werden. Die Analphabeten Landwirte haben andere Anforderungen und brauchen dementsprechend andere Präsentation als die Alphabeten Landwirte und die Helfer. Des Weiteren variieren die Kenntnisse im Bereich der IT – Systeme von „Vorhanden“ bis „gar keine“. Als Konsequenz können gegeben falls die sich in der IT – Welt schon längst etablierten Interaktionsparadigmen für Landwirte nicht verständlich sein. Es ist also notwendig, für die gleichen Informationen mindestens zwei Präsentationen zu erstellen, damit die verschiedenen Stakeholder das System entsprechend eigenen Erwartungen benutzen können. Diese Erkenntnisse sind ausschlaggebend für die weitere Modellierung des Systems.

# Work Reengineering

Als erster Schritt vor der eigentlichen Entwicklung und Evaluation wird die Reorganisation der deskriptiven Aufgabenmodellierung durchgeführt. In diesem Schritt wird das Reengineered Task Organisation Model erstellt.

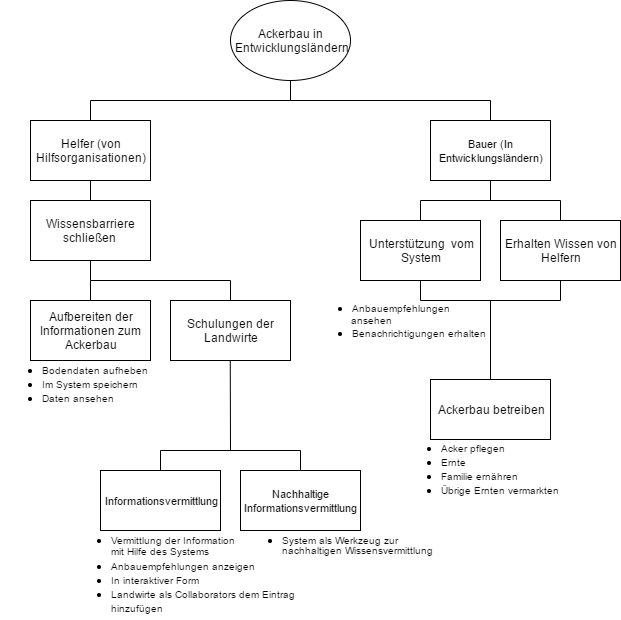


Abbildung 3 Reengineered Task Organization Model

Reorganisation der Useraufgaben sollte in der ersten Linie höherer Performance und Automatisierung bestimmter Arbeitsprozesse führen. In diesem Sinne wird das System die Informationssammlung und die Informationsvermittlung Prozesse vereinfachen und automatisieren. Zum einen soll das System die Helfer bei der Identifikation des Problems und einer Lösungsentwicklung durch die automatische Bodenanalyse unterstützen und als Medium zur Wissensübermittlung dienen. Zum anderen bekommen die Landwirte die Möglichkeit, die Information direkt vom System zu beziehen und werden dadurch autonomer.

# Conceptual Model Design

Conceptual Model Designs ist erster Schritt der Design und Testing Prozesse. Hier werden die grundlegenden Regeln für high– Level UI Entwickelt und später als Grundlage für weitere Designentscheidungen verwendet. Conceptual Model Design muss in vorherigen Kapiteln durchgeführte Analysen referenzieren.

**Processes – oriented**

In dem System werden von den Usern keine Produkte erstellt oder verwaltet, sondern bestimmte Prozesse zur Erledigung der Aufgaben durchgeführt. Die Prozesse sind von dem Task Organization Model abgeleitet

**Prozesse**

Zu grundlegenden Prozessenim System gehören die Authentifikation der User, Erstellen der Entries, Darstellung der Entry – Daten und Darstellung der Tutorials. Hier werden die Prozesse hierarchisch Dargestellt.

System zur Unterstützung der Ackerbauer in Entwicklungsländern

Authentifikation

Registration

Userdaten eingeben

Login

Userdaten verwalten

Aktualisieren

Löschen

Entry erstellen

Daten eingeben

Collaborators hinzufügen

Speichern

Entry Daten verwalten

Aktualisieren

Löschen

Entry – Daten ansehen

Liste mit Entries

Details ansehen

Tutorial ansehen

Erhalten der Systembenachrichtigungen

Aus dem oberen hierarchischen Modell wird ersichtlich, dass die User nach der Authentifikation im System ein Eintrag mit den Ackerdaten erstellen können und danach zu dem die Infos und Tutorials ansehen können. Des Weiteren ist die Möglichkeit gegeben, die Userdaten und Entrydaten zu verwalten

**Design presentation rules for processes**

Da es im Projekt Material Design Prinzipien von Android angehalten werden, wird zu Gestaltung der einzelnen Prozesse Android Styleguide verwendet. Die High – Level und Sub – level Prozesse werden in einzelnen Windows (Activity) dargestellt, Navigation zwischen denen durch Buttons und Pfeilen erfolgen soll. Nach der erfoglreichen Registration und dem Login wird Activity mit der Liste von Entries angezeigt, auf die der bestimmte Benutzer Zugriff hat. Falls der User im System eingeloggt ist, wird bei Start des Systems direkt die Liste mit allen Entries angezeigt, von wo man zu den anderen Activities navigieren kann.

**Design rules for windows**

Android Material Design Dokumentation umfasst sehr detaillierte Spezifikationen und best practices der UI Gestaltung, welche auch eingehalten werden. Die Dokumentation beschreibt vor allem die Gestaltungsprinzipien der Activities und vielen weiteren UI – Elementen wir Buttons, Dialogfenstern, Inputfeldern usw. Auf formale Auflistung wird hier verzichtet, Details können in der Android [Dokumentation](https://material.io/guidelines/) angesehen werden.

**Navigation**

Wie schon erwähnt, Erfolg die Navigation durch das System mit Hilfe von Android Buttons. Nach der Registration/Login gelangt man in die Root – Activity. Von dem Root – Activity, wo die Liste von Entries angezeigt wird, navigiert man durch den Click auf einen Listeneintrag zur der Detail – View eines Entrys. Da werden ausführliche Infos zu dem Entry präsentiert. Von dort navigiert der Button **tutorial** zu der Activity, die ein interaktives Tutorial präsentiert.

Root – Activity verfügt über ein Floating Action Button zum Öffnen eines Formulars, wo die Daten eines Entrys eingetragen und gespeichert werden können. Nach dem Speichern navigiert das System automatisch in die Root – Activity. Zum Bearbeiten des Userprofiles klick man auf den Imagebutton in der Actionbar der Root – Activity. Die Zurück – Navigation erfolgt mit Hilfe von Zurück – Pfeil in der Actionbar, der immer zu einer für jede Activity definierten Parent – Activity führt.

Im Folgenden Kapitel werden die Conceptual Model Mock – ups erstellt, um die oben beschriebenen Designkonzepte zu verdeutlichen.

## Conceptual Mock – ups

Conceptual Mock – ups stellen wichtige Prozesse des System Grafisch dar. In diesem Projekt wird auf grafische Tools zur Erstellung der Mockups aus zeitlichen Gründen verzichtet. Stattdessen werden die Mockups auf einem Stück Papier gezeichnet.

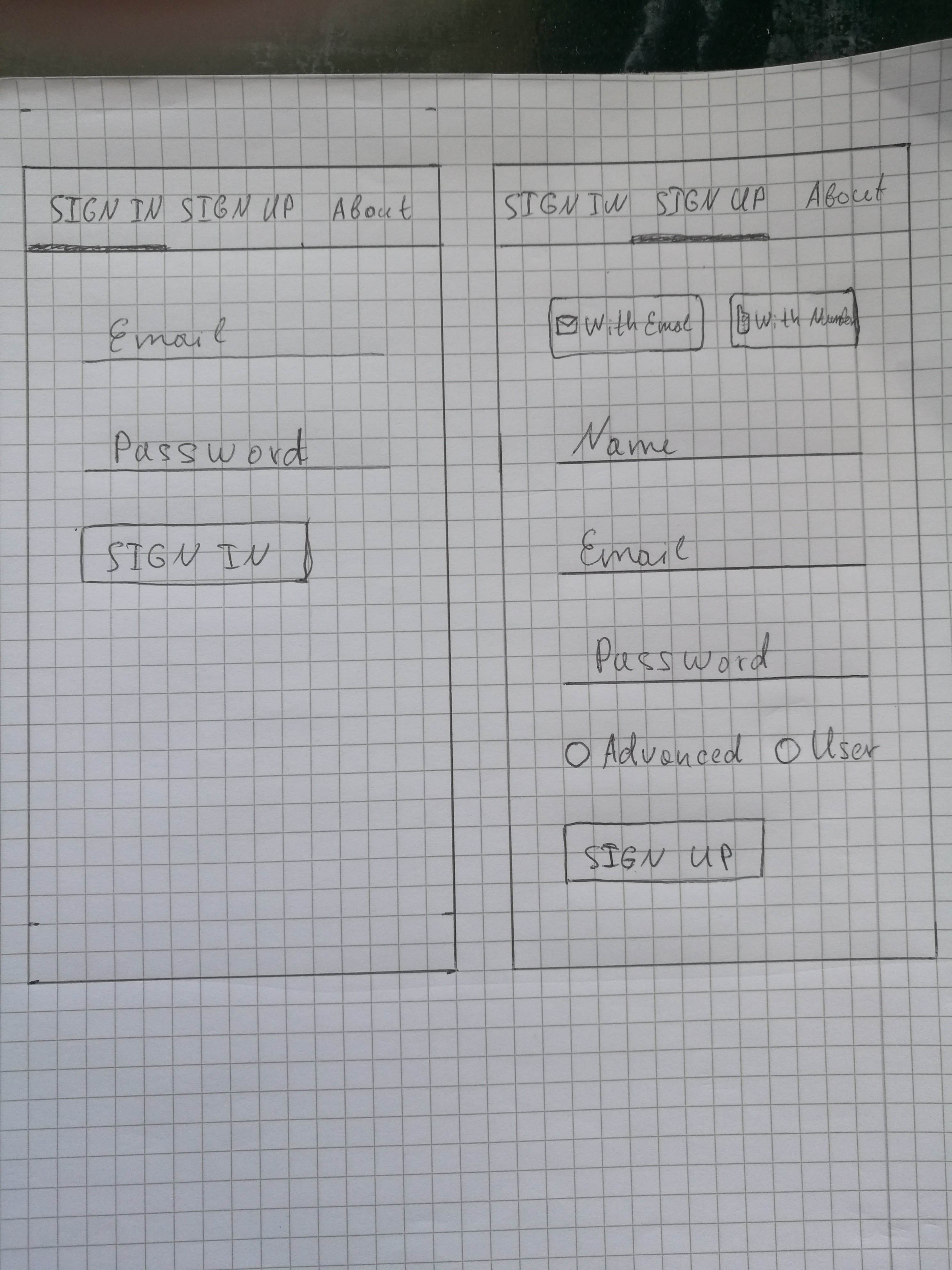


Abbildung 4 Sign up und Sign in

Regestration/Login werden mit **Tabs** präsentiert, was die Navigation viel effizienter macht. **Sign Up** Tab ermöglicht verschiedene Registrationsmethoden – die klassische mit Email für erfahrene User und die Registration nur mit Mobiltelefonnummer für diejenigen, die kein Smartphone besitzen oder keine Emailadresse haben. Die Formulare können mit jeweiligen Buttons umgeschaltet werden.

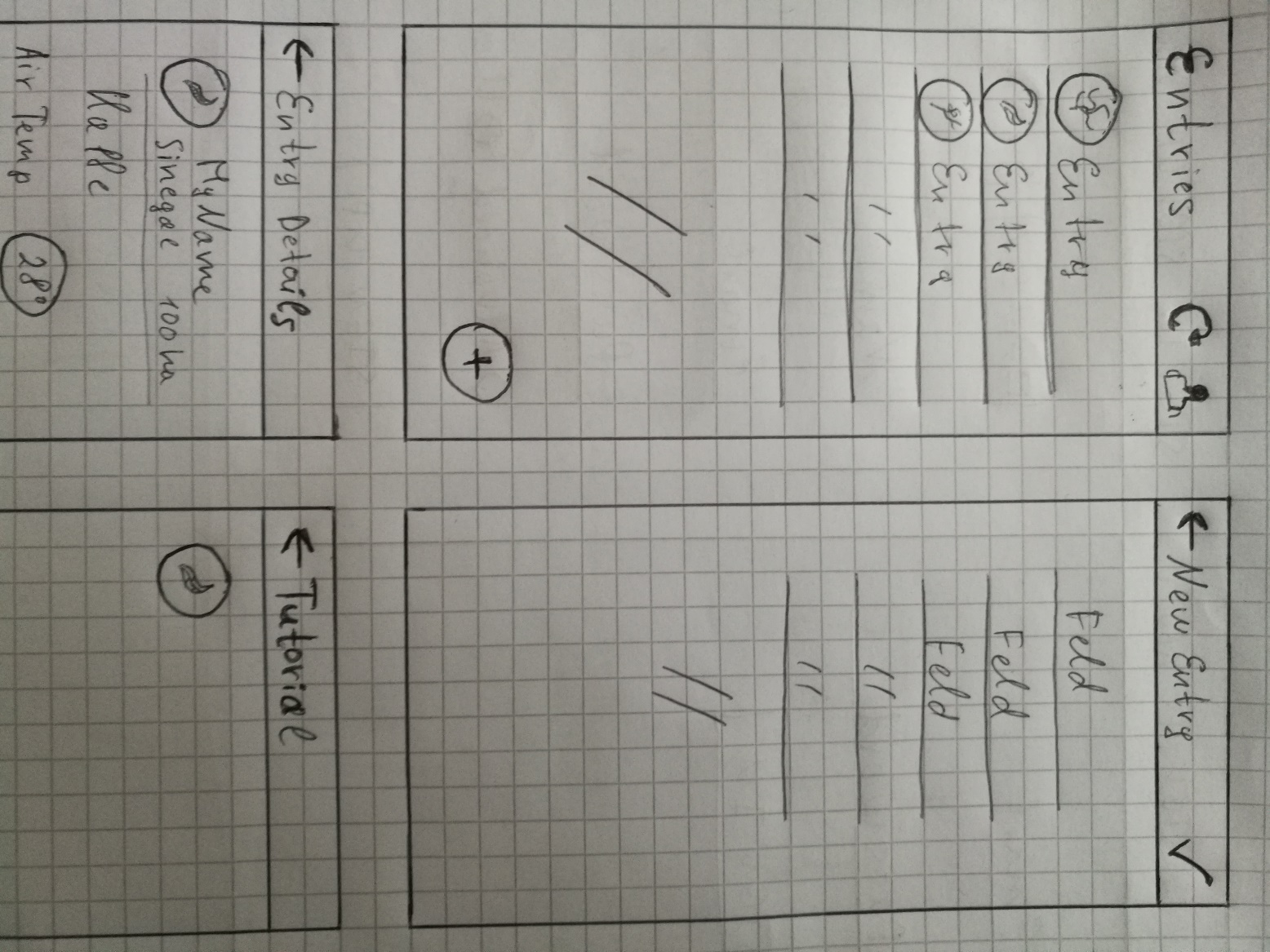
Alle Entries, auf die der bestimmte Benutzer Zugriff hat, werden in einer Liste dargestellt. Mit dem Cklick auf Floating Action Buttons öffnet sich andere Activity mit dem Formular (Siehe Abbildung 6) zum Erstellen eines neuen Eintrys. Dieser Button ist nur im Modus für erfahrene Benutzer sichtbar. Die Listen Einträge sollen nur kurze kern Informationen eines Entrys präsentieren.

Abbildung 5 Liste mit Entries

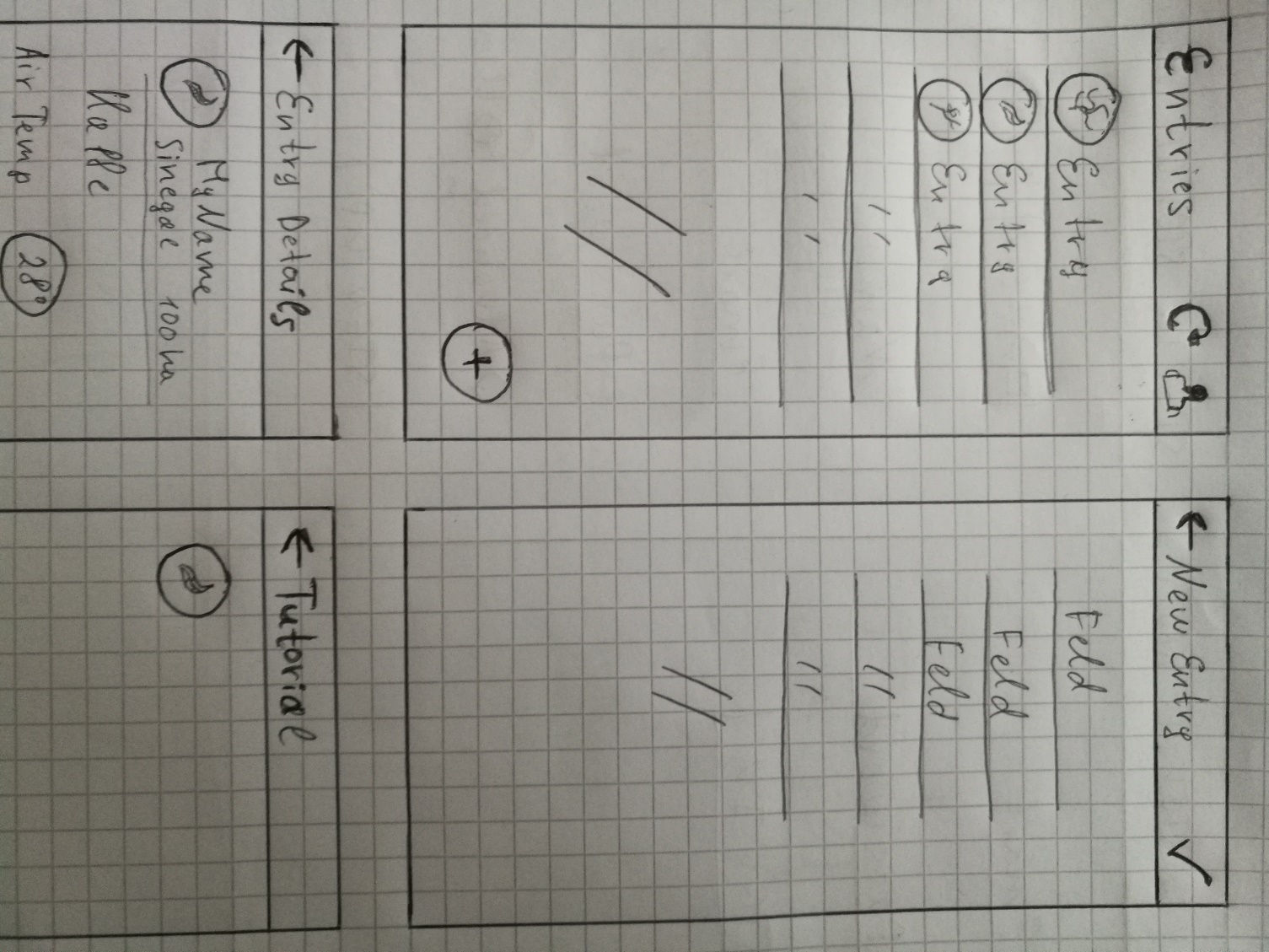
In dieser Activity werden Daten eines Ackers Eingegeben und im System gespeichert. Es werden mehrere Textfelder und Dropdown – Menus im Fenster zur Eingabe benutzt. Prozess der Dateneingaben kann nur von den Alphabeten erfolgen. Dieses Fenster ist daher nur für die Alphabeten User erreichbar.

Abbildung 6 Neuen Eintrag erstellen

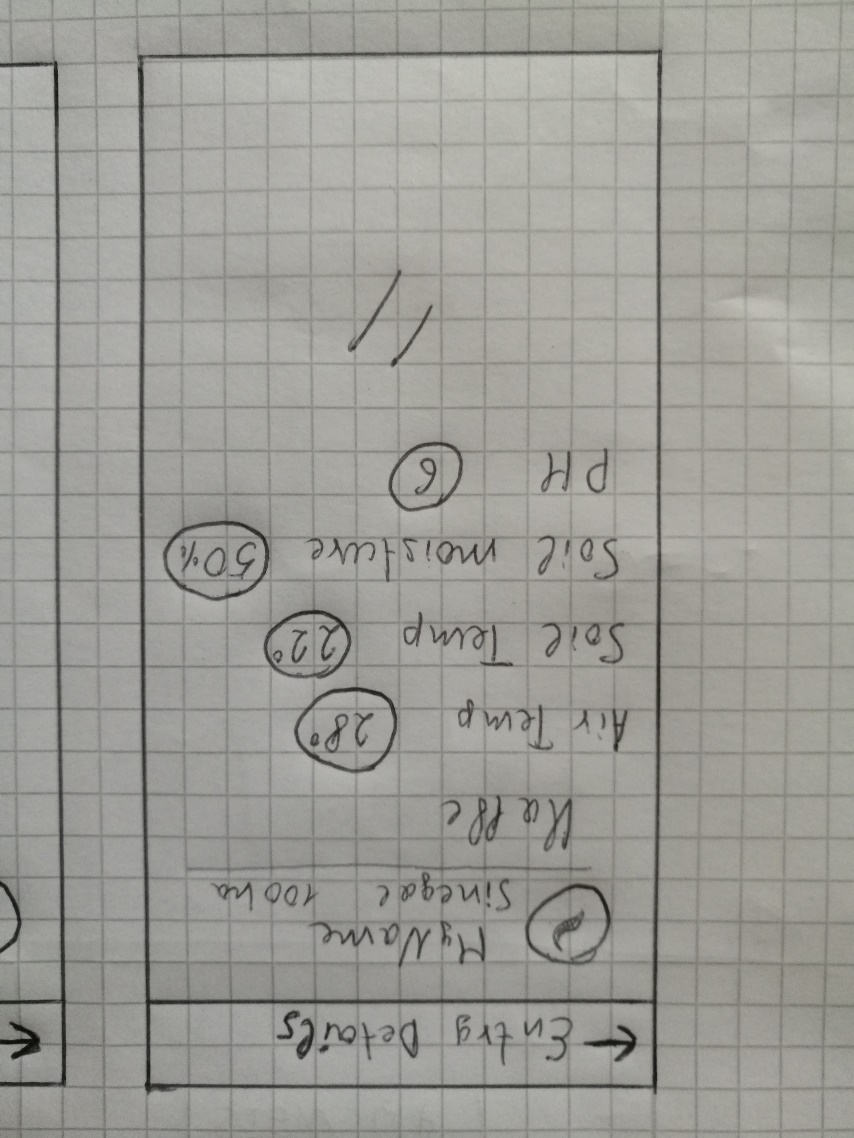
In diesem Fenster werden Details zu einem Entry dargestellt. Hier sollen alle einzelnen Eigenschaften eines Eintrags präsentiert werden. Die passende Visualisierung ist an dieser Stelle besonders wichtig. Die Details müssen sowohl für Alphabeten als auch für Analphabeten verfügbar sein. Das heißt, es müssen zwei verschieden Präsentationen der Information erstellt werden. Abbildung 7 ist die Präsentation für Alphabeten und wird ebenfalls visuelles Feedback zu dem Zustand des Ackers enthalten, abhängig von den ermittelten Daten.

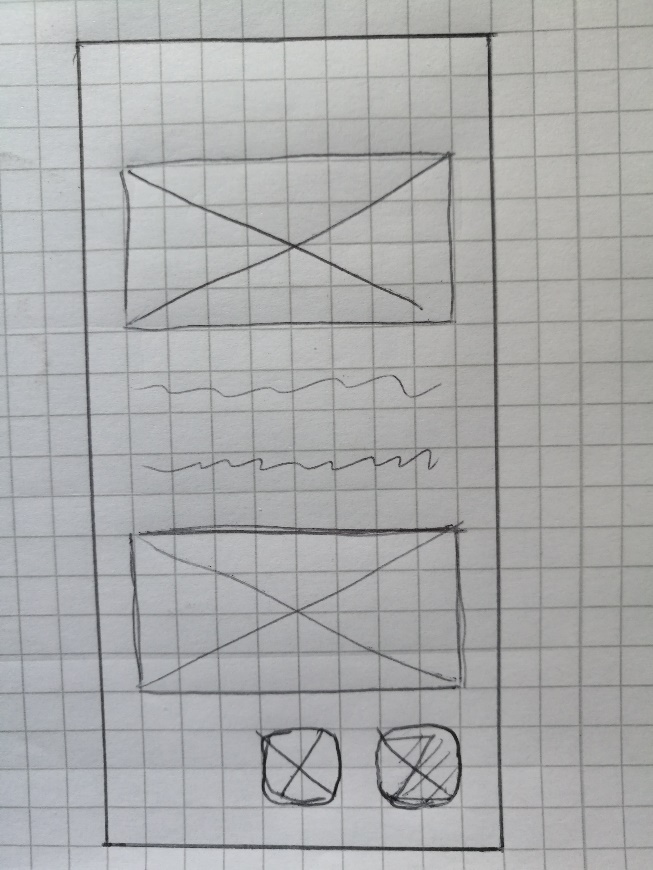
Abbildung 7 Entrie Details

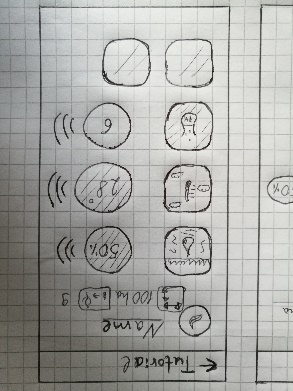


Die Präsentation der Ackerinformationen für die Analphabeten wird direkt mit den Anbauempfehlungen kombiniert und in einem Fenster dargestellt, damit der Benutzer weniger Zwischenschritte machen muss, um die Tutorials anschauen zu können. Hier ist besonders darauf zu achten, dass wo möglich die Information nicht in textueller Form, sondern in visueller oder Verbaler Form dargestellt wird. Um ein Objekt der realen Welt zu präsentieren, müssen möglichst genau beschreibende Abbildungen gewählt werden. Da wo es auf textuelle Informationen nicht verzichtet werden kann, müssen die Texte vertont werden. Die Mischung aus verbalen und visuellen Informationspräsentationen soll ein für Analphabeten gebrauchstaugliches UI schaffen. Diese Ansicht ist für Beide Profiltypen verfügbar. Auf diese Weise können sowohl Helfer auf die Tutorials zugreifen um die Schulungen durchzuführen, als auch die Landwirte selbst.

Abbildung 8 Tutorial zu Einem Eintrag

Abbildung 9 Ausführliche Anbauerklärung



Um die ausführlichen Informationen zum Ackerbau anzuzeigen öffnet sich bei Klick auf ein bestimmtes Icon über dem Tutorial – Activity ein neues Fenster. Z. B bei Klick auf Icon werden ausführliche Anbauempfehlungen zu Verbesserung der Bodenfeuchtigkeit in weiteren Bildern, Animationen, und vertonten Texten dargestellt.

## Evaluation

Für kleinere Projekte empfiehlt Mayhew nur die abschließende Evaluation des Projekts nach dem Level 3. Für dieses Projekt wird ebenfalls auf die Iterative Evaluation in jedem Level verzichtet. In der ersten Linie wird dies aus zeitlichen Gründen gemacht. Des Weiteren sind für die Evaluation detaillierte Bilder und Animationen notwendig, um die Gebrauchstauglichkeit des Systems besonders für Analphabeten bewerten zu können. Diese liegen im aktuellen Stadium des Projekts nicht vor. Im Rahmen des Projekts wird nur eine abschließende Evaluation nach dem Level 3 durchgeführt.

# Screen Design Standards

In diesem Kapitel werden die Screen Design Standards festgelegt, um ein konsistentes UI zu Schafen und ease of learning und ease of use zu unterstützen. Zu Erstellung der Standards werden als Grundlage die Style Guides von Android verwendet.

## Control Standarts

Die Steuerungselemente werden in der folgenden Tabelle aufgeführt. Diese Elemente ermöglichen die Kontrolle über alle Interaktionsprozesse im System.

|  |  |
| --- | --- |
| Aktion | Control |
| Navigation | Buttons, Tabs, anklickbare Listenelemente, Image – Buttons |
| Eins auswählen | Checkbox |
| Eingabe | Textfeld, Dropdown-Menu(spinner) |
| Liste | Dropdown-Menu |

## Process window standarts

Die Einzelnen Activitys, in denen Prozesse ausgeführt werden, sollen ebenso einheitlich gestalten werden um die Konsistenz des UI gewährleisten zu können. Dazu schlägt Mayhew die Process window standarts festzulegen, nach denen die einzelnen Fenster designt werden sollen. Dazu werden ebenso in Android etablierten Prinzipien verwendet Im Folgenden werden wesentliche Punkte genannt.

**Actionbar**

* Die Actionbar sollte in jedem Prozess – Fenster implementiert werden und dem User darüber Auskunft geben, wo er sich grade befindet
* Die Actionbar soll Navigations – Icons haben oder Buttons, die die Funktionalität eines Fensters erweitern
* In jedem Fenster außer dem Root – Fenster soll ein Back – Pfeil enthalten sein, um die Navigation zum Parent – Fenster zu ermöglichen

**Activity**

* Activities (Fenster) außer dem Login – Activity müssen Light – Gray – Background haben
* Activities mit vielen Inputelementen oder Informationen müssen scrollbar sein
* Aktionen wie Save/Delete müssen in der Actionbar mit entsprechenden Icons dargestellt werden
* Activities, die Usereingeben fordern, müssen benötigte Inputfields, Checkboxes und Dropdownmenus haben
* Listenelemente sollen in Card – Form präsentiert werden, Jedes Element enthält ein Bild, Namen, Location und Fläche
* Details – Activity sollen alle verfügbare Daten Visualisieren
* Tutorials – Activity werden überwigen mit Interaktiven Bildern und Audio dargestellt
* Activities, die in bestimmten Fällen keinen Inhalt haben, sollen den Empty – State deutlich visualisieren.
* Möglichst alle UI – Elemente des Screens sollen anklickbar sein, derer Inhalt soll vertont werden.

## Weitere Standarts

**Dialog box standards**

Android Dialogs werden verwenden, um bestimmte Aktion des Users zu bestätigen. Z. B. Delete/Update/Discard Aktionen erfordern eine bestätigen der User, um zufällige Ausführung dieser Aktionen zu vermeiden.

**Message box standards**

Andoid Toast – und Snackbarmessages geben dem User zusätzliche Informationen oder informieren ihn über Background Prozesse.

**Input device interaction standards**

Das System wird mit Touch – Gesten und falls vorhanden den Hardware – Buttons gesteuert.

**Feedback Standarts**

Die Ladeprozesse des Systems werden mit Android Progress – Indikatoren visualisiert. Der Status der ausgeführten Prozesse wird Toast – und Snackbarmessages dem User angezeigt. Die Snackbarmessages sollen zudem den Status farblich präsentieren. Erfolgreich ausgeführte Prozesse solle in grüner Farbe dargestellt werden, die Fehler in roter Farbe.

Nicht ausgefüllte Inputfelder müssen rot markiert werden. Zudem sollte es auf den Fehler genau hingewiesen werden. Z. B wenn der eingegebene Passwort zu kurz, soll dies explizit in der Fehlermeldung angegeben werden. Zu Kontrolle der Inputfelder werden ebenso Andorid – Tools verwendet.

Weitere Usereingaben wie Checkboxen und Dropdown – Menus müssen falls nötig ebenfalls kontrolliert werden und im Fehlerfall müssen die Benutzer explizit über Toast – und Snackbarmessages informiert werden.

In der folgenden Tabelle werden wichtige Designeigenschaften des Systems aufgeführt, die angehalten werden müssen.

Tabelle 2 Style Guide

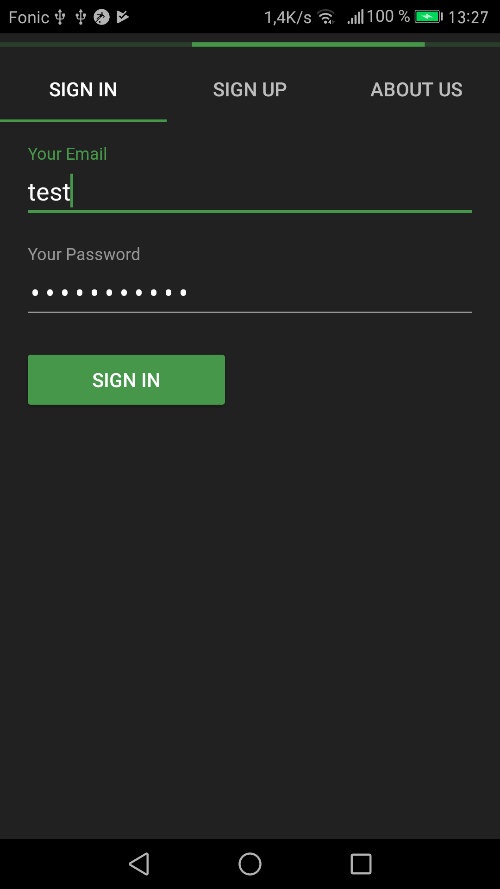
|  |  |  |
| --- | --- | --- |
| Element | Eigenschaft | Anmerkung |
| Colors | | |
| Primary | #212121 | Farbe der Actionbar |
| Primary Dark | #000000 |  |
| Accent Color | #47974b |  |
| Hint Color | #969696 | Wird in InputFields benutzt |
| Error Color | #fd5656 | Farbe des Textes der Errormassages |
| Successful Color |  | Farbe des Textes der Errormassages zur erfolgreichen Aktionen |
| Main Background Color | #c4c4c4 | Backgroundcolor der Activities |
| Push Button | | |
| Color | Accent Color | Farbe der normale Push Buttons |
| Border | none |  |
| Radius | none |  |
| Shadow | noen |  |
| Cards | | |
| Fill | white |  |
| Margin | 6dp |  |
| Radius | 2dp |  |
| Elevation | 2dp |  |
| Tutorial Icons | | |
| Color | multicolored | Farbige Bilder |
| Radius | 4dp |  |
| Border | 2dp |  |

## SDS Prototyping

Da die Screen Design Standards von Android exakt übernommen werden, wäre das Prototyping überflüssig. Die Darstellung einzelner Elemente wird von den [Android Design Guidelines](https://material.io/guidelines/) nicht abweichen. Dieser Schritt wird daher mit Detailed User Interface Design kombiniert.

# Detaild UI Design

In diesem Kapitel werde die einzelnen Activities und UI – Elemente dargestellt und erläutert. Vorwiegen werden bei der detaillierten Erstellung des UI Android Style Guides angehalten. Solche Eigenschaften wie Iconsgröße, Textgröße, Abstände zwischen Elementen werden exakt übernommen, das diese schon längt die best practices sind. Auch die Icons, die Routinen Aufgaben wie löschen, aktualisieren, überarbeiten oder speichern, werden von [Material Design Icons](https://material.io/icons/) verwendet. Im Grunde genommen müssen nur die Activities individuell gestaltet werden, die für Analphabeten verfügbar sind. Dies bedeutet, dass die Informationen nicht nur mit Bildern präsentiert werden müssen, sondern es spezielle Icnos erstellt werden müssen, die sowohl für Analphabeten als auch für die User mit eingeschränkten Kenntnissen im Umgang mit interaktiven IT – Systemen verständlich sind. Die Material Icons sind dagegen sehr schlicht und monochrom gestaltet, so dass sie gegebenenfalls nur für erfahrene Benutzer verständlich sind. Im Folgenden werden die Lösungen präsentiert.

Abbildung 10 Präsentiert das Login – Fenster. Das Fenster ist in den im oberen Kapitel definierten Farben gestaltet. Zwischen **SIGN IN** und **SIGN UP** Fenstern erfolgt mit Tabs, was die Navigation sehr effizient macht. Über den Tabs ist eine Progressbar sichtbar, die die Ladeprozesse des Systems visualisiert und signalisiert dem User, dass das System nicht hängt, sondern eine interne Funktion ausführt. Die Progressbar muss in allen Aktivities implementiert werden, in denen zeitintensive Funktionen ausgeführt werden. **SIGN UP** ist dem entsprechend ähnlich aufgebaut. Unterschied ist nur in dem, dass im **SIGN UP** Fragment zusätzlich noch die Checkboxes zu Eingabe benutzt werden, um den Profiltype zu ermitteln.

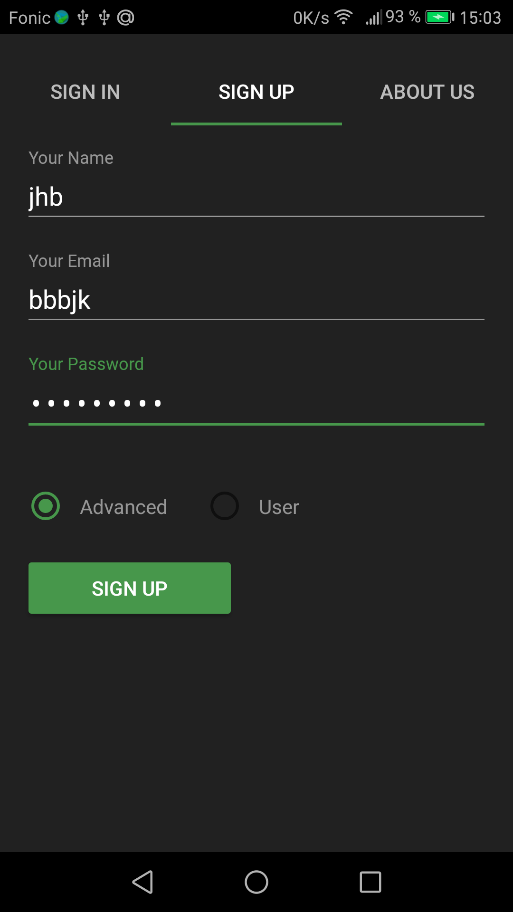


Abbildung 10 Sign In Fragment

Abbildung 11 Sign up Fragment

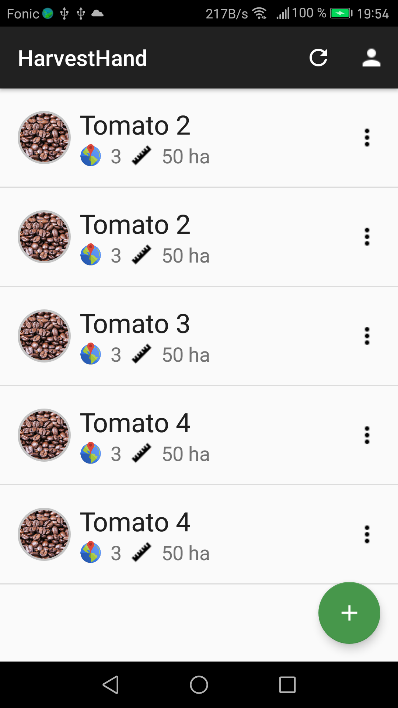
Die Entrys werden in einer Liste präsentiert. Ein Listeneintrag enthält kurze Übersicht über ein Entry und ein Bild, das die Bestimmte Pflanze darstellt. Im Beispiel Links sind Kaffee – Acker mit einer entsprechenden Abbildung dargestellt. Der Plus Floating Action Button navigiert zum Formular zur Erstellung eines neuen Eintrags. Wichtig ist hier, dass die Informationen entsprechende Visualisierungen haben. Die Übersicht Informationen müssen auch für Analphabeten verständlich sein. Daher wird es auf die Material Icons verzichtet. Stattdessen müssen möglichst aussagekräftigere Abbildungen verwendet werden. Aus Zeitgründen konnten die Abbildungen nicht erstellt werden. In Abbildung 12 sind mögliche Lösungen vorgestellt. Jeder Eintrag hat ein weiteres Popup – Menü (3 vertikale Punkte rechts) mit dem man den Jeweiligen Eintrag löschen oder bearbeiten kann. Dieses Menü sowohl das Plus – Button sind nur für Alphabeten verfügbar, das es da viele textuelle Inputs benötigt werden.

Abbildung 12 Liste mit Entries

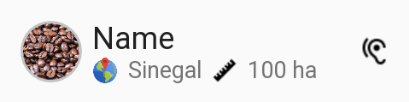
Für die Analphabeten muss ein Eintrag etwas anders aussehen. Hier wird ein Icon zum Anhören des ganzen Inhaltes eines Eintrags angebracht. Da es auf die textuellen Informationen vollständig nicht verzichtet werden kann, muss diese Funktion überall da verfügbar sein, wo Textelemente vorhanden sind.

Abbildung 13 Item - Präsentation für Analphabeten

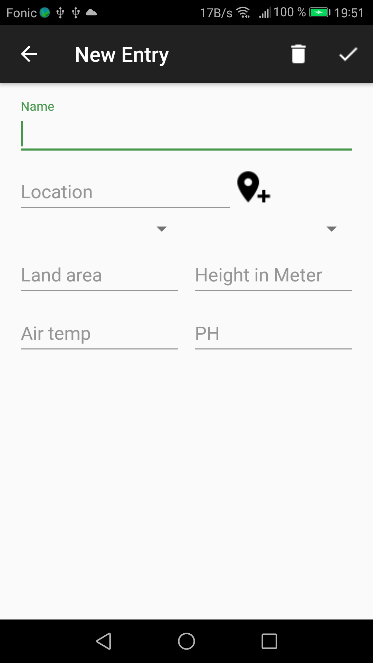
Hier ist ein Formular abgebildet, das zum Erstellen eines Neuen Eintrags verwendet wird. Diese Funktionen ist nur für User mit vorhandenen Lese – und Schreibekompetenzen verfügbar. Hier werden Inputfelder mit Dropdownmenüs und Icons Kombiniert. Alle Elemente werden nach Android Style Guides gestaltet, da es hier keine besondere Präsentation nötig ist. Die Actionbar hat ein Navigationspfeil, mit dem man immer zu Parent – Activity navigieren. Des Weiteren sollen sich immer die Aktionsbuttons wie Save/Delete in diesem Fall in Actionbar befinden, damit diese Aktionen schnell erreicht werden können, ohne zu scrollen. Wenn das Fenster verlassen wird, bevor gemachte Änderungen gespeichert wurden, muss eine Dialogbox angezeigt werden, um die Aktion zu bestätigen. Löschvorgänge müssen ebenfalls bestätigt werden. Dies gilt für alle Fenster, wo ähnliche Vorgänge ausgeführt werden können.

Abbildung 14 Erstellung eines Entrys

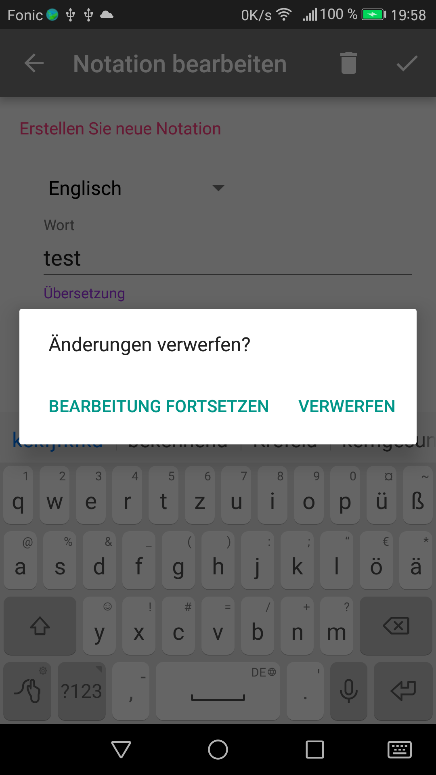
Dialogboxen werden nach Andorid Style Guides gestaltet.

Abbildung 15 Diologbox

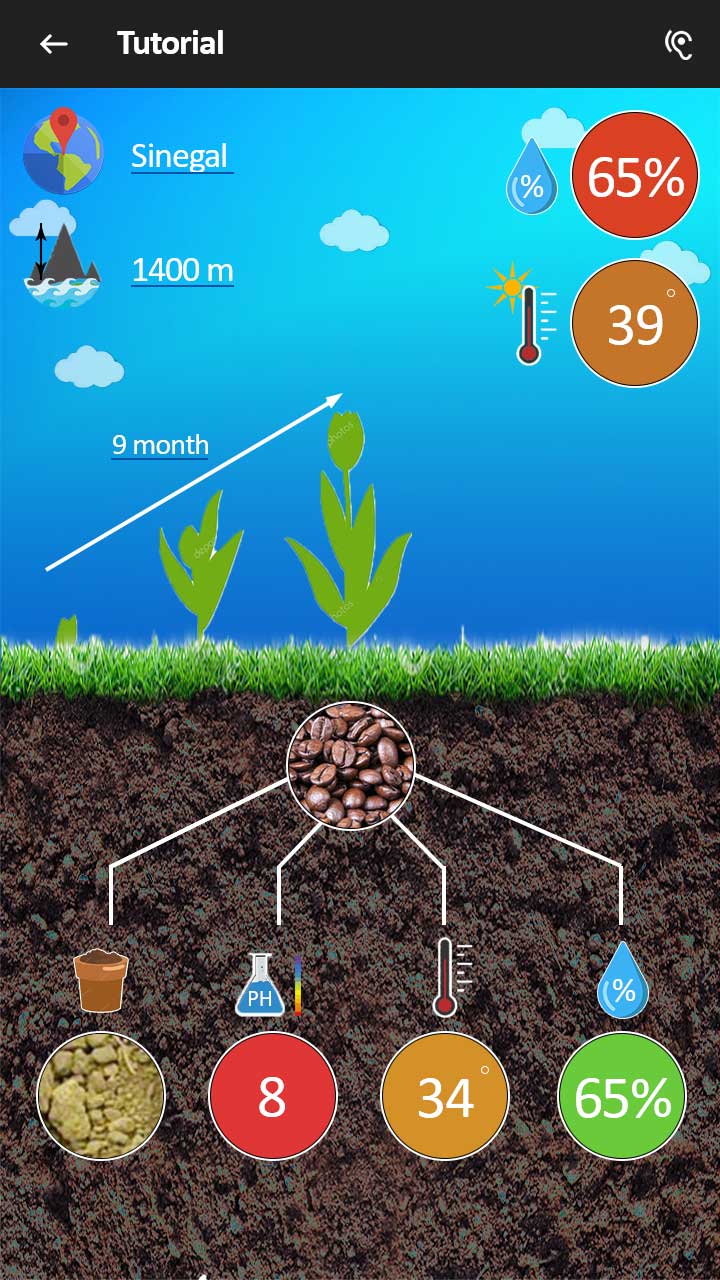
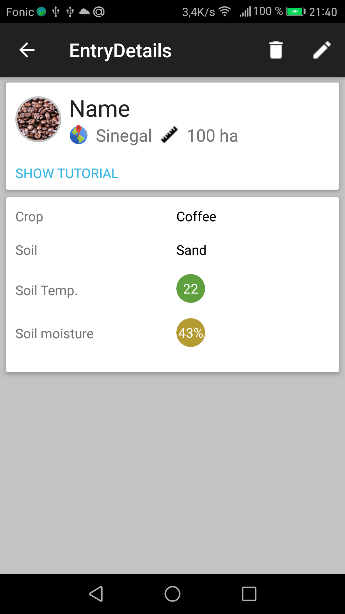
****Wie schon in vorherigen Kapitel erwähnt, können die Prinzipien von Android Material Design für die Analphabeten nicht funktionieren. Daher wurde ein eigens Konzept der Visualisierung für die Analphabeten erarbeitet. Die in Kapitel 5.1 dargestellten Tutorial Mock – ups wurden komplett überarbeitet. Bei diesem Konzept wird auf die Abbildung der realen Objekte der Welt gesetzt. Die Informationen und Objekte werden möglichst mit beschreibenden Bildern präsentiert. Die untergestrichenen Texte werden beim Anklicken vertont. Es sind in Prinzip alle Elemente interaktive. Die einzelnen Werte wie PH – Wert oder Lufttemperatur werden in Farbigen Kreisen dargestellt. Die Hintergrundfarbe eines Kreises gibt Feedback über den Zustand bestimmter Eigenschaften. Grün ist dabei „gut“, gelb „befriedigend“ und rot „schlecht“. Beim Anklicken einer der Eigenschaft öffnet sich ein Popup – Fenster mit einer ausführlichen Erklärung und Anbauempfehlung. Dies wurde von dem Tutorial Mock – up in Kapitel 5.1 übernommen.

Abbildung 16 Tutorial Activity



Die in der Abbildung 17 dargestellte Ansicht ist nur für die Alphabeten User verfügbar, hier werden die Informationen klassisch dargestellt. Zu der Ansicht in der Oberen Abbildung kann man aus diesem Activity ebenfalls navigieren. Dies ermöglicht z. B den Helfern den Zugriff auf die Tutorials, damit die diese bei schulengen vor Ort präsentieren können. Die Liste der Elemente ist vollständig. Da der Inhalt dieser Activity dynamisch ist, wurden die Elemente nur beispielhaft aufgeführt.

Abbildung 17 Entry Details

In der Abbildung 17 ist die Fehlerbehandlung dargestellt. Hierfür werden Android Standardtools verwendet

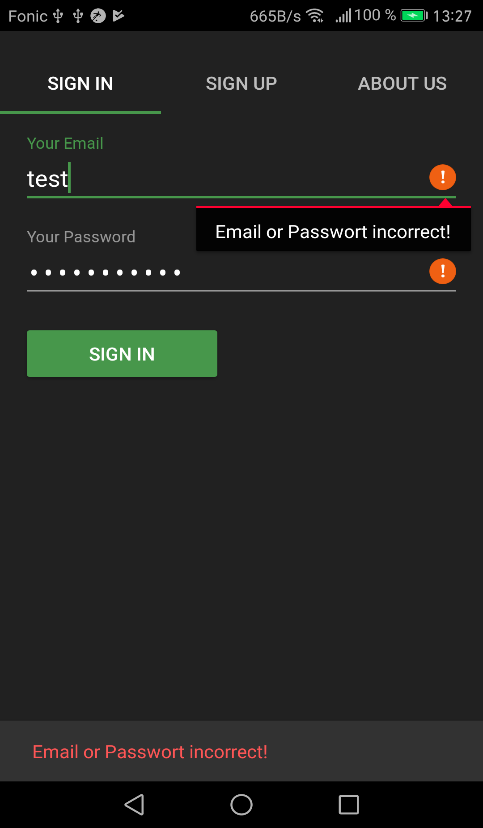
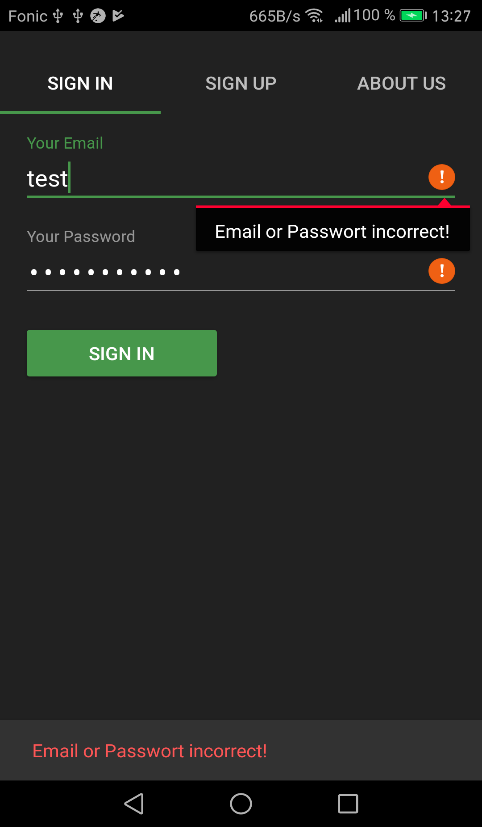


Abbildung 18 Fehlerbehandlung

## Fazit

Die größte Herausforderung bei der detaillierten Konzipierung des UIs stellt die Anpassung für die Analphabeten. Hier muss der Kompromiss zwischen der textuellen, visuellen und verbalen Präsentatione gefunden werden. Zudem müssen sehr viele Abbildungen und Icons erstellt werden, um die Information angemessen darzustellen.

Um das UI für Alphabeten und Analphabeten zu optimieren, werden schließlich zwei Modi der Darstellung implementiert, die sich grundsätzlich in dem unterscheiden, dass der Analphabeten – Modus kein Zugriff auf die Verwaltung der Entries hat. Delete/Update/Edit/Add Aktionen sind in dem Modus nicht verfügbar. Des Weiteren fehlt in dem Modus die klassische Darstellung der Entry – Details Informationen. Stattdessen gelangt man direkt zu der Interaktiven Ansicht aus Abbildung 16. Der Alphabeten – Modus hat Zugriffmöglichkeiten sowohl auf alle Verwaltungsfunktionen, als auch auf die Interaktive Ansicht.

# Evaluation

Im Folgenden wird die Durchführung der Evaluation des Detailed User Interfaces Design dokumentiert. Die Evaluation wird nach Mayhew erfolgen. Im Rahmen der Evaluatione sollen sowohl ease – of – learning, als auch ease – of – use Aspekte des Systems getestet werden.

Für die Helfer ist die Effizienz der Arbeiterledigung im Fokus. Da die Interaktionsparadigmen schon bekannt sind, ist für die Helfer ease – of – use Aspekt wichtiger. Für die Analphabeten und die User mit eingeschränkten Kenntnissen im Umgang mit interaktiven Systemen steht ease – of – learning im Fokus, da diese Stakeholdergruppe sich erstmals mit Interaktionen des Systems vertraut machen muss. Des Weiteren muss Inhalt der Abbildungen verinnerlicht werden. Aus diesem Grund werden beide Aspekte getestet.

Da es für das Entwicklung Team nicht möglich ist, die direkten Stakeholder des Systems in die Evaluation zu involvieren, werden Personen für die Tests rekrutiert, die die ähnlichen Eigenschaften aus den definierten Userprofiles haben. Die Evaluation erfolgt in der gewöhnlichen Umgebung der Benutzer, in diesem Fall in einem Wohnzimmer.

## Evaluation anhand der Usability Goals und der Anforderungen

Zu der quantitativen Einschätzung der Durchgeführten Tests wurden die in Kapitel 3.7 definierten quantitativen Usability Goals verwendet. Die Zeiten, die die Bestimmten User für jeweilige Tasks gebraucht haben, wurden aufgezeichnet und dokumentiert. Näher dazu im Anhang.

Grundsätzlich lieferten die Evaluation akzeptable Ergebnisse. Die Zahlen in den Usability Goals konnten in den Tests erreicht werden. Dafür war das UI so organisiert, dass die Kernfunktionalitäten des Systems in wenigen Schritten erreichbar sind.

Auch die Visualisierungen des Systems wurden von den involvierten Benutzern im Schnitt gut erkannt. Bei Thinking-Aloud-Test konnten die User die meisten Icons in Tutorial – Activity problemlos verstehen und erklären.

Das aller erste Problem wurde bei der Registration im System ermittelt. Hier wurde die Tab Navigation nicht direkt erkannt, die User versuchten sich direkt im **SIGN IN** Fenster einzuloggen und bekamen dementsprechend eine Fehlermeldung „*Incorrect Email or Password*“, was sie irritierte. Einer der Gründe dafür war, das die Tab Überschriften sich stark ähneln und die Unterschiede nicht wahrgenommen werden. Als Lösung hierfür wird **SIGN IN** in **LOG IN** umbenannt. Des Weiteren soll im Login – Fenster genaue darauf hingewiesen werden, dass der Registrationsformular sich in anderem Fenster befindet.

Für die Analphabeten ist die Registration im System ohne weiteren Hilfe klar nicht möglich. Um aber diesen Vorgang für die Analphabeten zu optimieren, fehlen die Visuellen Informationen zu den Aktionen wie z.B Icon in den Eingabefeldern, die etwas darüber sagen würden, welche Eingabe benötigt werden. Es wird ebenfalls überlegt, die Authentifikation nur mit Hilfe der Telefonnummer durchzuführen und somit diesen Vorgang noch effizienter zu gestalten.

Einige Icons sollen überarbeitet werden, da die nicht eindeutig interpretiert werden konnten. So ein Problem wurde bei dem **Bodentemperatur – Icon** und **Fläche – Icon** festgestellt.

Weiterer kritischer Punkt war die Sprach – Funktion der App. Es wurde nicht erkannt, dass die Texte beim Anklicken vorgelesen werden könnten. Dies muss dringen korrigiert werden. Auch das **Ear – Icon** war nicht ganz eindeutig und konnte nur von den erfahrenen Benutzern als Vorlesefunktion erkannt werden.

# Systemarchitektur

Im Folgenden wird die Systemarchitektur genauer erläutert. Das im Konzeptplan vorgestellte Modell wurde überarbeitet um die Unterschiede zwischen der möglichen Clients deutlicher zu machen.

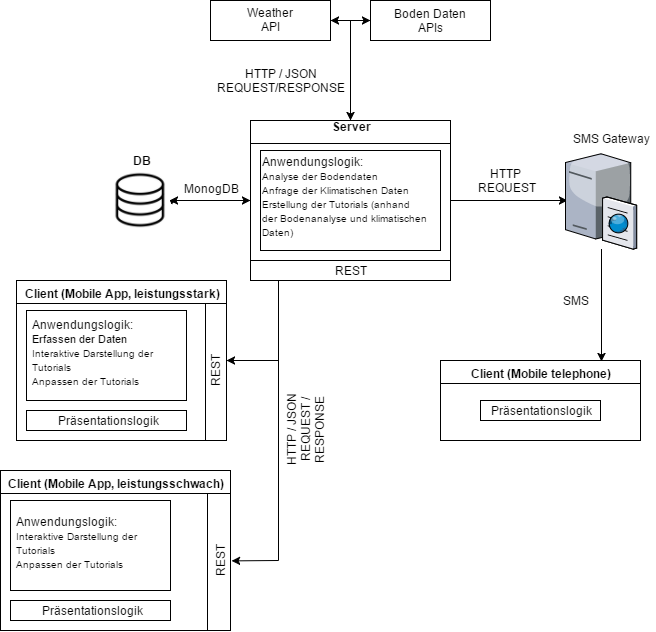


Abb. 2 Systemarchitekturmodell (Überarbeitet)

Bei der Entwicklung des Systems muss besonders beachtet werden, dass die Benutzer die Smartphones mit sehr unterschiedlichen Leistungsfähigkeiten besitzen können. Die Helfer benutzen eher moderne und leistungsstärkere Smartphones oder Tabletts. Die Bauer dagegen haben schwächere und kleinere Geräte mit älteren Android Versionen. Diese Erkenntnis aus dem Nutzungskontext ist besonders wichtig und die Verteilung des Systems muss entsprechend realisiert werden.

## Ressourcen

Hier wird API zum Zugriff auf Systemressourcen definiert. Im Folgeneden Kapitel werden die Ressourcen genauer beschrieben.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| URI | Verb | Semantik | Contenttype(req) | Contenttype(res) | Statuscodes |
| /signup | POST | Registration der  Benutzer | JSON | JSON | 200 – Registration erfolgreich |
|  |  |  |  |  | 500 – interner DB Fehler |
|  |  |  |  |  | 409 – der User mit der eingegebenen Email existiert bereits |
| /signin | POST | Login des Benutzers | JSON | JSON | 200 – Registration erfolgreich |
|  |  |  |  |  | 500 – interner DB Fehler |
|  |  |  |  |  | 401 – Login Daten falsch eingegeben |
| /users/:id | GET | Bestimmte Userdaten werden abgerufen | Parameter | JSON | 200 – Daten werden abgerufen |
|  |  |  |  |  | 204 – Kein User mit der id gefunden |
|  |  |  |  |  | 500 – interner DB Fehler |
| /users/:id | PUT | Bestimmte Userdaten aktualisieren | Parameter, JSON | JSON | 200 – Daten aktualisiert |
|  |  |  |  |  | 204 – Kein User mit der id gefunden |
|  |  |  |  |  | 500 – interner DB Fehler |
| /users/:id /entries | GET | Vom User Erstellte Entries werden abgerufen | Queryparameter | JSON | 200 – Daten abgerufen |
|  |  |  |  |  | 204 – Kein Entry gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /entries | POST | Entry wird erstellt | JSON | JSON | 200 – Entry erfolgreich erstellt |
|  |  |  |  |  | 500 – interner Fehler |
| /entries/:id | PUT | Entry wird aktualisiert | Parameter, JSON | JSON | 200 – Entry aktualisiert |
|  |  |  |  |  | 204 – Kein Entry gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /entries/:id | GET | Daten eines Entries werden abgerufen | Parameter | JSON | 200 – Daten abgerufen |
|  |  |  |  |  | 204 – Kein Entry gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /entries/:id | DELETE | Entry wird gelöscht | Parameter | JSON | 200 – Entry gelöscht |
|  |  |  |  |  | 204 – Kein Entry gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /entries/:id/tutorial/:id | GET | Tutorial zu einem Entry wird abgerufen | Parameter | JSON | 200 – Tutorial abgerufen |
|  |  |  |  |  | 204 – Kein Tutorial gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /norms | POST | Eine Norm wird erstellt | JSON | JSON | 200 – Norm erstellt |
|  |  |  |  |  | 500 – interner Fehler |
| /norms/:id | PUT | Eine Norm aktualisieren | JSON | JSON | 200 – Norm aktualisiert |
|  |  |  |  |  | 204 – Keine Norm gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /norms/:id | DELETE | Bestimmte Norm löschen | Parameter | JSON | 200 – Norm gelöscht |
|  |  |  |  |  | 204 – Keine Norm gefunden |
|  |  |  |  |  | 500 – interner Fehler |
| /norms | GET | Norms nach Queryparametern abrufen | Queryparametern | JSON | 200 – Daten abgerufen |
|  |  |  |  |  | 204 – Keine Norms gefunden |
|  |  |  |  |  | 500 – interner Fehler |

## Datenstrukturen

Alle Entitäten des Systems sowie Daten der externen Dienste werden im JSON – Format repräsentiert. Zu den internen Daten des Systems gehören Benutzerdaten (user), Daten eines Ackers(entry) und die dynamisch erstellten Anbauempfehlungen (tutorial). Des Weiteren werden standarisierte Daten zur den verschiedenen Pflanzenarten (norm) in der Datenbank des Systems gespeichert. Zur externen Daten gehören die Wetterdaten und bestimmte Bodendaten.

**Repräsentation eines Users**

Zur Informationen eines User gehören nur für die Authentifikation benötigte Angaben. user\_type wird verwendet um die Präsentation der Informationen an Analphabeten anzupassen. TYPE\_USER identifiziert einen Analphabeten und Präsentation in der App wird entsprechend.

**var** userSchema = mongoose.Schema({  
 name: {type: String, required: **true**},  
 gender: String,  
 email: {type: String, unique: **true**},  
 pass: {type: String},  
 user\_type: {type: Number, required: **true**} /\*TYPE\_PROFI = 1, TYPE\_USER = 0\*/

phone\_number:{type: Number, unique: **true** }

});

**Repräsentation eines Entry**

Entry – Struktur wird zur Speicherung bestimmter Ackerdaten verwendet.

**var** entrySchema = mongoose.Schema({  
 /\*Ein beschreibender Name\*/  
 entry\_name: {type: String, required: **true**},  
 /\*Art der Pflanze, durch id identifiziert\*/  
 art\_id: {type: Number, required: **true**},  
 /\*Verknüpfter Tutorial\*/  
 tutorial\_id: String,  
 /\*ID des Erstellers\*/  
 owner\_id: {type: String, required: **true**},  
 /\*Colaborators, die den Zugriff auf den Eintrag haben\*/  
 collaborators:[String],  
   
 /\*  
 \* Properties  
 \*/

location: String,  
 /\*Fläche in ha\*/  
 surface: Number,  
 /\*Art des Bodens\*/  
 soil: Number,  
 air\_temp: Number,  
 soil\_temp: Number,  
 soil\_moisture: Number,  
 ph\_value: Number,  
 /\*Anbauhöhe über dem Meerspiegel\*/  
 height\_meter: Number  
  
});

**Repräsentation einer Norm**

Eine Norm speichert standartwerte einer bestimmten Pflanze und wird zur Bodenanalyse verwendet. Und die Abweichungen festzustellen werden die Daten eines existierenden Eintrags mit den Standartwerten abgeglichen. Norm wird von Entwicklungsteam mit Hilfe von Recherchen erstellt, da es kein passendes API mit Pflanzendaten gefunden war. Diese Entität beinhaltet für den Anbau einer Pflanze relevanten Daten und kann im Prozess der Entwicklung verändert sein.

**var** normsSchema = mongoose.Schema({  
 crop\_id: Number,  
 name: [String],  
 family: String,  
 air\_temp: {  
 max: Number,  
 min:Number  
 },  
 soil: String,  
 soil\_temp: {  
 max: Number,  
 min:Number  
 },  
 soil\_moisture: {  
 max: Number,  
 min: Number  
 },  
 ph\_value: Number,  
 height\_meter: {  
 max: Number,  
 min: Number  
 },  
 mature\_after\_month: Number  
 /\*.  
 .  
 \*/  
});

**Repräsentation eines Tutorials**

Eine Norm speichert standartwerte

**var** normsSchema = mongoose.Schema({  
 crop\_id: Number,  
 name: [String],  
 air\_temp: {  
 max: Number,  
 min:Number,  
  
 },  
 soil\_temp: {  
 max: Number,  
 min:Number,  
 },  
 soil\_moisture: Number,  
 ph\_value: Number  
 /\*.  
 .  
 .  
 .  
 \*/  
});

**Externe Wetterdaten**

Als Wetterdienst wird Weather Underground verwendet (wunderground.com). Unten ist eine beispielhafte JSON – Response. Die benötigte Daten wie Temperatur, Höhe müssen aus dem Response herausgefiltert werden. Die Wetterdaten werden vom Server nach Location abgefragt und falls nötig gefiltert an den Client gepusht. Der Client sollte nur dann benachrichtig werden, wenn ein Wetterereignis den Pflanzenanbau beeinflussen kann. Dadurch werden die Anbauempfehlungen flexibler gestaltet und können an die Aktuellen Wetterbedingungen angepasst werden.

{…

"local\_tz\_short": "PDT",

"local\_tz\_long": "America/Los\_Angeles",

"local\_tz\_offset": "-0700",

"weather": "Partly Cloudy",

"temperature\_string": "66.3 F (19.1 C)",

"temp\_f": 66.3,

"temp\_c": 19.1,

"relative\_humidity": "65%",

"wind\_string": "From the NNW at 22.0 MPH Gusting to 28.0 MPH",

"wind\_dir": "NNW",

"wind\_degrees": 346,

"wind\_mph": 22.0,

…}

**SMS Getaway**

Zum Verschicken von SMS wird ein POST – Request an websms REST API geschickt und von dort an die Telefonnummer. Dies ist eine optionale Funktion des Systems und nur dann implementierte, wenn Ressourcen dazu vorhanden sein werden. Besonders wichtig ist hierbei, dass die Nachriten in einer binären Form verschickt werden müssen, damit die Mobiltelefone sie verarbeiten können. Für die Anbauempfehlungen bedeutet es, dass sie möglichst kompakt und nur in Form von Texten und einfachen gegeben falls schwarzweißen Bildern gestaltet werden müssen. Die SMS sind zudem weniger interaktiv und sind für das Projekt nicht relevant. Weiter wird Request Body Beispiel vorgestellt (Quelle: <https://websms.de/entwickler/apis/rest-sms-api>). Details können in der API – Doku nachgelesen werden.

**{**

"userDataHeaderPresent" **:** **false,**

"messageContent" **:** **[** "..."**,** **...** **],**

"test" **:** **false,**

"recipientAddressList" **:** **[** **...,** **...** **],**

"senderAddress" **:** "..."**,**

"senderAddressType" **:** "national"**,**

"sendAsFlashSms" **:** **false,**

"notificationCallbackUrl" **:** "..."**,**

"clientMessageId" **:** "..."**,**

"priority" **:** **...,**

**}**

**Externe Bodendaten**

Zur Abfrage der Bodenfeuchtigkeit und Bodentemperatur in bestimmter Location können APIs von ClearAg (https://www.clearag.com/) und Department of Agriculture and Food ([link zur seite](https://www.agric.wa.gov.au/daily-summary-data#!/weatherstations/dailySummary)) verwendet werden. ClearAg liefert ausführliche Daten zur Bodenfeuchtigkeit. Department of Agriculture and Food stellt Daten zur Bodentemperatur zu Verfügung. Diese Information können vom System abgefragt werden, falls die Daten von dem Benutzer vor Ort nicht erhoben werden können.

## Anwendungslogik des Systems

## Wichtige Anmerkungen

## Datenschutz

Das System verarbeitet nur wenige private Daten der Benutzer wie Email und Telefonnummer. Diese können in der Datenbank verschlüsselt gespeichert werden. Weiterer Austausch der privaten Daten findet im System nicht statt, so dass keine besonderen Maßnahmen der Datensicherheit nötig sind.

# Fazit

## Ausblick

## Persönliche Meinung

# Installationsdokumentation

## Systemanforderungen

Zum ausführen des Systems werden Nodejs und MongoDB unter Windows 7-10 und Internetverbindung gebraucht. Für den Client ist mindestens Android 4.0 notwendig. Server und Client müssen sich in gleichem Netz befinden. Im Folgenden werden benötigte Schritte detailliert beschrieben

## Server

* Nodejs herunterladen und installieren
* Benötigte Packeges mit dem npm – Befehl über console installieren: *npm –i*
* MongoDB herunterladen und installieren
* MongoDB zur umgebungsvariablen des Systems hinzufügen
* Datenbank mit dem Konsolenbefehl starten: *mongod –dbpath db*
* Server in der Console starten: *node server*

## Client

* HarvestHand.apk ausführen und installieren
* IP Des Servers ermitteln und in der App eintragen
* App benutzen

# Anhang

## POCs (überarbeitet)

### Benutzer anlegen

**Beschreibung**: Helfer und Bauer müssen jeweils Profile erstellen können, um Zugriffsrechte zu erhalten. Dabei werden verschieden Daten der Benutzer als JSON – Objekt an Server geschickt und gespeichert. Es soll einfache Authentifikation mit Email und Passwort oder nur mit Handynummer erfolgen. Zwei Profilmodi mit verschiedenen Zugriffsrechten für Alphabeten und Analphabeten sollen verfügbar sein.

**Exit**: Jeder User kann abhängig von eigene Kompetenzen Profiltyp auswählen und sich registrieren. Nach Eingabe erforderlicher Daten wird der User in Datenbank angelegt.

**Fail**: Der Benutzer hat keine Email und keine Handynummer und kann sich somit nicht registrieren. Dem Benutzer fehlen Kenntnisse im Umgang mit interaktiven Systemen.

**Fallback**: Der Helfer soll die Bauer bei Erstellung eines Profils unterstützen. Außerdem kann beim Start der App eine Anleitung zum Erstellen von Profilen vorgeführt werden. Falls der Benutzer keine Möglichkeit zur Registration im System hat, sollen alternative Authentifikationsmethoden eingesetzt werden.

### Collaborators dem Eintrag hinzufügen

**Beschreibung**: Der User muss einem Eintrag als Collaborator hinzugefügt werden, damit er die Ackerdaten ansehen kann und dazu spezialisierte Empfehlungen erhält. Der Benutzer wird nach der Email oder Handynummer gesucht, sodass immer nur ein genauer Benutzer gefunden wird.

**Exit**: Profil des gesuchten Benutzers wird in der Datenbank gefunden und dem Eintrag hinzugefügt. Die Datenbanksuche und die Datenübertragungszeit dürfen nicht länger als 5-10 Sekunden dauern.

**Fail**: Das Benutzerprofil wird nicht in der Datenbankgefunden oder die Suche überschreitet die zugelassene Suchzeit.

**Fallback**: Das Profil des Benutzers muss neu angelegt werden

### Effiziente Datenübertragung

**Beschreibung**: Die Daten Übertragung zwischen Systemkomponenten muss effizient realisiert werden, so dass im JSON – Format nur für Verarbeitung relevante Daten übertragen werden, und keine Bilder, Audio oder Video Dateien. Die Struktur der JSON – Daten sollte möglichst einfach gehalten werden und primitive Datentypen enthalten.

**Exit**: Es wird geringer Datenvolumen zwischen Komponenten transferiert. Die Datenmengen überschreiten nicht 100 KB pro eine Übertragung, Übertragungszeiten überschreiten nicht 10 Sekunden. Die Daten zur Visualisierung der Informationen werden lokal auf dem Client – Gerät gespeichert und abgerufen.

**Fail**: Die Kommunikation zwischen Systemkomponenten verbraucht mehr als 100 KB Datenvolumen pro Anfrage und dauert länger als 10 Sekunden. Datenübertragung wird stakt beeinträchtigt oder abgebrochen.

**Fallback**: Schlechte Kommunikationsinfrastruktur in den betroffenen Ländern muss bei der Entwicklung der Systemarchitektur berücksichtig werden. Längere Datenübertragungszeiten sollen zugelassen werden, im Fail – Fall soll die Anfrage erneut gesendet werden

### Auswertung der Bodendaten

**Beschreibung**: Die aufgehobenen Daten werden im JSON an den Server geschickt, da analysiert und darauf aufbauend werden Tutorials zum Ackerbau für Benutzer erstellt. Dazu sollen auch Informationen zur aktuellen Wetterbedingungen bezogen werden. Für den Analyse – Algorithmus müssen bestimmte Bodendaten wie Bodenfeuchtigkeit, Lufttemperatur, PH – Wert etc. zur Verfügung stehen. Die rechenaufwändigen Algorithmen werden dadurch nicht auf dem Client realisiert und beeinträchtigen nicht die Performance des Endgeräts.

**Exit**: Bei der Eingabe der Daten wird auf die Vollständigkeit zugewiesen. Die Daten sind für die Verarbeitung vollständig, werden na Server geschickt, ausgewertet und zur weiteren Verarbeitung an den Client geschickt.

**Fail**: Der Algorithmus arbeitet fehlerhaft oder die Daten zur Analyse sind nicht vollständig, so

dass nicht valide Empfehlungen für Benutzer erstellt werden.

**Fallback**: Dem Benutzer wird eine standarisierte nicht individuell angepasste Anleitung angezeigt.

### Effiziente Erstellung interaktiver Tutorials

**Beschreibung**: Es Werden dynamisch personalisierte Anbau - Tutorials auf dem Client in interaktiver form dargestellt. Dazu verwendete Bild – Audio oder – Videodateien sollen lokal in möglichst komprimierter Form vorliegen, so dass die Performance und Akkuleistung des Endgeräts nicht beeinträchtigt wird.

**Exit**: Die Anleitungen werden auf Basis der vom Server Erhaltenen Daten erstellt und interaktiv dargestellt. Dabei sollte nicht mehr als 50-100 MB RAM und nicht mehr als 10% Prozessorleistung pro Session verwendet werden.

**Fail**: Der Algorithmus arbeitet fehlerhaft oder die lokalen Daten werden zu lange aus dem Speicher geladen oder können nicht gelesen werden. Oder es ist nicht ausreichend RAM verfügbar und die Applikation wird geschlossen. Dadurch wird der Client stakt belastet und die Anleitungen werden nicht angezeigt.

**Fallback**: Eine in textueller Form oder nur in Bildern gestaltete Anleitung anzeigen. Auf die interaktive Visualisierung, Videos und Animationen wird verzichtet.

### Klimatische Daten der Externen Dienste

**Beschreibung**: Zur Erstellung der genaueren Empfehlungen werden klimatische Daten von externen Diensten gebraucht. Abhängig von aktuellen Wetterbedingungen werden die Empfehlungen ständig aktualisiert. Dazu wird ein Wetter – API ausgesucht, das nach REST realisiert ist und die Daten im JSON – Format abgerufen werden können.

**Exit**: Die Kommunikation mit Externen Diensten erfolgt mittels JSON – Format. Die Wetterdaten werden erfolgreich von externen Diensten abgefragt und zur Analyse der Bodendaten und Definition der Anleitungen herangezogen.

**Fail**: Der externe Dienst ist nicht erreichbar oder die gelieferten Daten sind nicht valide. Die Anleitungen werden fehlerhaft.

**Fallback**: Es müssen alternative zuverlässige Dienste verwendet werden oder es wird auf externe Daten verzichtet. Dadurch werden die Anleitungen nicht Wetterabhängig dargestellt.

### Erstellung der Anleitungen für Mobiltelefone

**Beschreibung**: Falls ein Benutzer kein Smartphone besitz, soll er die Empfehlungen in vereinfachter Form zeitnah auf seinem Mobiltelefon per SMS erhalten. Wegen der Plattformbeschränkungen können die Informationen nur in Form von Texten oder Bildern dargestellt werden. Dazu soll ein API für SMS Gateway verwendet.

**Exit**: Aktuelle Informationen werden im JSON – Format an einen externen Dienst geschickt und von dort per SMS an die Telefonnummer des Benutzers verschickt.

**Fail**: SMS wird nicht verschickt, da das SMS Gateway nicht erreichbar ist. Oder die Funktion wird gar nicht realisiert, da das Verschicken von SMS kostenpflichtig ist.

**Fallback**: Auf die Funktion kann wird verzichtet. Die Helfer informieren die Ackerbauer vor Ort.

### Didaktische Darstellung der Informationen

**Beschreibung**: Das UI des Systems und die Informationen sollen didaktisch sinnvoll Visualisiert oder in anderen Formen dargestellt werden, damit auch die Analphabeten sie nachvollziehen können. Dabei ist wichtig, dass die UI Elementen Visualisierungen oder verbale Darstellung enthalten, damit die Funktionalität auch für Analphabeten verständlich wird

**Exit**: Die Informationen und UI Elemente sind für alle Stakeholdergruppen nachvollziehbar. Die Interaktionselemente sind mit sprechenden Bildern oder Audio versehen. Die User können dem Interaktionsfluss folgen.

**Fail**: Die Form der Informationsdarstellung ist ungeeignet, so dass das System nicht für alle Stakeholdergruppen gebrauchstauglich wird. Die User können das System nicht benutzen

**Fallback**: Auf die Funktion darf nicht verzichtet werden. Es sollen intensive Recherchen im Bereich der didaktischen Informationsdarstellung und Wissensvermittlung durgeführt werden.

## Personae

**Landwirt Amadou**

Amadou ist 30 Jahre alt, ein afrikanischer Ackerbauer und lebt in Burkina Faso. Er hat eine Frau und 3 Kinder, welche er durch die Erträge seiner Hirseernte ernähren muss. Doch oft fallen die Erträge niedrig aus. Durch Dürre vertrocknen die Pflanzen, durch hohe Niederschläge werden die Pflanzen aus dem Boden gespült. Amadou besitzt ein Smartphone, allerdings kann er es nicht nutzen wie Menschen aus einem Industrieland. Lese- und Schreiben stellt für ihn ein erhebliches Problem dar, da er keine schulische Ausbildung genießen konnte und auch seine technischen Erfahrungen eher gering sind. Er weiß, dass er etwas an seinem Anbauverfahren ändern muss, denn die Existent seiner Familie ist bedroht.

**Landwirtin Alika**

Alika ist 37 Jahre alt, eine afrikanische Bäuerin und lebt in Nigeria. Sie und ihr Mann bauen Vanille an, um ihre Familie mit 4 Kindern zu ernähren. Vor einigen Jahren konnten sie immer gute Erträge erzielen und sorglos davon leben, doch ihre Ernten fallen Jahr für Jahr schlechter aus. Dies treibt die Familie immer weiter in die Armut. Woran das liegt, können Sie sich nicht erklären. Grund dafür ist, dass ihnen Fachwissen fehlt. Sie konnten keine Schule besuchen und haben den Anbau aus Tradition weitergeführt. Außerdem sind Analphabeten und können nicht Lesen und Schreiben. Ein Smartphone konnte sich die Familie vor einiger Zeit leisten, jedoch fehlt ihnen das Wissen es richtig zu benutzen. Alika möchte die Ursache für die niedrigen Erträge herausfinden und ist auch bereit ihre Tradition des Vanilleanbaus aufzugeben, wenn sie eine Alternative finden würde mit welcher sie den Lebensunterhalt ihrer Familie sichern kann.

**Landwirt Bouba**



Bouba ist ein Landwirt einer Dorfgemeinschaft in Senegal. Er ist 43, hat drei Töchter und einen behinderten Sohn. Seine Familie lebt hauptsächlich von dem Reisanbau. Die Reisanbbaugebiete sind aber in dem Land recht klein und machen nur geringen Anteil an der gesamten Landwirtschaft aus. Bouba tut alles, was ihm möglich ist, um seien Familie zu ernähren. Doch trotz aller Mühen reicht es für höchstens zwei Mahlzeiten pro Tag, die hauptsächlich aus Reis bestehen, Fleisch gibt es immer seltener. Die Familie verhungert und ihre Existenz ist unter Gefahr. Den Umzug in die Stadt kann er sich nicht leisten. Außerdem muss er sich verschulden, um Saatgut und Düngermittel kaufen zu können. Selbstverständlich hat der Bouba keine Spezielle Technik und Verzuge, die ihm die Arbeit auf dem Acker erleichtern würden. Alle seine Kenntnisse im Gebiet basieren auf den Traditionen seiner Vorfahren. Er gehört zu den 58% der Analphabeten in Senegal, so dass Erwerb neues Wissen für Ihn eine erhebliche Barriere ist. Zudem hat er kein Smartphone, sondern nur ein altes Handy. Der Bouba ist trotzdem für jede Lösung offen, die ihm dabei helfen würde, Existenz seiner Familie sichern zu können.

**Helfer Andreas Baumann**

Andreas Baumann ist 32 Jahre alt und betreibt einen landwirtschaftlichen Betrieb mit Ackerbau in Niedersachsen (Deutschland). Er hat sich als Aufgabe gemacht sein Wissen an Menschen in Entwicklungsländern weiterzugeben. Sein Studium in Agrarwissenschaften hat er erfolgreich abgeschlossen und übernahm den Betrieb seines Vaters. Dementsprechend ist sein Fachwissen in diesem Bereich besonders hoch und weiß dieses zu vermitteln. Als moderner Mann besitzt er Erfahrung im Umgang mit Smartphone, Tablet und Computer. Er ist offen und bereit sich für die Aufgabe als Vermittler weiter zu bilden und freut sich auf seine Aufgabe.

**Helferin Katharina Müller**



Katharina Müller ist 24 Jahre alt und betreibt gemeinsam mit ihrer Familie einen Ackerbaubetrieb in Schleswig-Holstein (Deutschland). Sie hat eine Ausbildung als Landwirtin und ihre Meisterprüfung absolviert. Katharina möchte den Menschen vor Ort Tipps und Tricks im Bereich Ackerbau zeigen, um die Erträge zu steigern. Dabei möchte sie auch Ansprechpartnerin für Frauen und Kinder der Familien sein. Sie besitzt ein gutes Fachwissen und ist besonders in der Praxis erfahren. Außerdem verfügt sie über ein gutes technisches Verständnis und weiß mit einem Smartphone, Tablet und einem Computer umzugehen.

## Evaluation

### Tasks

**Test Task 1 (Helfer)**

**Ackerdaten eintragen**

Erstellen eines neuen Eintrags im System, nach dem die Bodendaten vor Ort aufgehoben wurden.

Sie Sind in Senegal bei dem Landwirt Bouba. Im Rahmen Ihrer Mission in diesem Land, ist Ihre Aufgabe die Landwirte vor Ort zu unterstützen und das Wissen im Bereich Ackerbau langfristig zu vermitteln. Dennoch ist oft die fehlende Schreibe – und – Lesekompetenz der Landwirte eine Barriere, so dass ein passendes Werkzeug zur Wissensübermittlung fehlt. Ihnen wird das System zur Verfügung gestellt, welches dieses Problem lösen soll. Erstellen Sie dazu einen neuen Eintrag mit Ackerdaten:

1. Registrieren Sie sich im System
2. Navigieren Sie zum Formular
3. Geben Sie die benötigten Daten Ein
4. Speichern sie die Daten
5. Fertig!

**Test Task 2 (Helfer)**

**Anbauempfehlung anschauen**

Nach dem Sie ein Eintrag im System erstellt haben, müssen Sie nun die Anbauempfehlung dem Landwirt anzeigen. Erstmals müssen Sie die detaillierten Ackerdaten anschauen. Schauen Sie sich die Darstellung an und Identifizieren Sie die Probleme, die das Systems ermittelt hat. Navigieren sie dann zu der Anbauempfehlung des jeweiligen Eintrags.

1. Navigieren Sie zur detaillierten Informationen
2. Identifizieren Sie die Probleme der Pflanze
3. Navigieren Sie zum Anbauempfehlungen
4. Anbauempfehlungen dem Landwirt präsentieren
5. Fertig!

**Test Task 3 (Landwirt – Analphabet)**

**Registrieren im System**

Der Analphabet soll sich (mit der Unterstützung des Helfers) im System registrieren. Dazu soll er zum jeweiligen Fenster navigieren und sich dann entweder mit Email oder der Telefonnummer registrieren. (Auf detaillierte Beschreibung wird hier verzichtet)

**Test Task 4 (Landwirt – Analphabet)**

**Anbauempfehlung anschauen**

Der Analphabet soll die Anbauempfehlungen anschauen (Auf detaillierte Beschreibung wird hier verzichtet).

### User 1

Location: *Room*

Username: *Dolly* User NO: *1*

Data Collector: *Sergej Atamantschuk*  Iterations: *1*

1. **Are you:**

Male

x Female

1. **How old are you**

x 18-25

26-40

41-55

Over 55

1. **Are you color blind in any way?**

x No

Yes (please describe)

1. **Describe your education background**

None

Elementary school

Middle schools

High school (attended but did not graduate)

High school degree

College (no degree)

College degree

Graduate school (no degree)

x Graduate school degree

Other (please explain)

1. **How would you describe your experience level in agriculture?**

**x**  Novice

Advanced

Competence

Expertise

1. **Do you have any experience using Android system?**

No

x Yes

Yes, but only older versions

Yes, but only newer versions

1. **How would you describe your general level of expertise in using Android applications**

None

Low

x High

Other (please describe)

1. **How would you describe your general level of smartphone experience?**

None

Low

x High

Other (please describe)

**Data Collection Sheet – Task 1(User 1)**

**Correct steps:**

Mit dem Tab zur Registrierung navigieren. Benötigte Daten eingeben. Danach landet man in dem Rootactivity. Plus Floating Action Button anklicken und zum Formular wechseln. Daten eingeben. In Actionbar Save – Button anklicken. Done!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Correct steps | Correct ? | | Time | Errors/Comments |
|  | Y | N | 1.Iteration  2 min  2.Iteration  <=1 min |  | |
| **Touch** oder **Wischen zu dem** **SIGN UP** Tab, um zum Registrationsformular zu navigieren |  | N | Versucht sich direkt im Sign in Fenster zu registrieren,  bekommt Fehlermeldung. Comment vom  User: „*SIgn In“* in „*Log In*“ umbenennen, Tab  Navigation ist Praktisch | |
| Datenfelder ausfüllen | Y |  | Check boxen sollen sprechenden Namen haben,  die Auswahl des User-Profils war nicht eindeutig | |
| Auf **SIGN UP** Button klicken | Y |  |  | |
| Plus Floating Action Button anklicken | Y |  |  | |
| Datenfelder ausfüllen:  Falss bestimmte Daten nicht vorliegen, auf ein jeweiliges **Icon,** klicken, um diese Daten von System automatisch ermitteln zu lassen | Y |  | Sollte deutlicher drauf hingewiesen werden,  dass Felder wie „Location“ oder „Airtemp“ vom  System automatisch ausgefüllt werden müssen. | |
| Auf **Save – Icon** in der Action Bar klicken | Y |  |  | |
| Feedback Meldung erhalten | Y |  | Toast Message erkannt | |

**Data Collection Sheet – Task 2 (User 1)**

**Correct steps:**

Auf Listeneintrag klicken und zur Detailansicht wechseln. Probleme einer Pflanze anhand der Visualisierung interpretieren. Auf **SHOW TUTORIAL** klicken. Zur Anzeige der detaillierten Empfehlungen auf jeweilige Icons klicken.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Correct steps | Correct ? | | Time  (2.Iteration) | Errors/Comments |
|  | Y | N | <= 3 sec |  |
| Auf einen Listeneintrag in der Liste **klicken** | Y |  |  |
| Angezeigte Information interpretieren | Y |  | Informationen verständlich dargestellt, Die Bedeutung der Farben sollte jedoch erläutert werden |
| Button **SHOW TUTORIAL** anklicken | Y |  | Der Inhalt der meisten Icons wurde erkannt, die Icons sollten jedoch noch präzisiert werden, um die Realen Objekte der Welt eindeutiger darzustellen |

### User 2

Location: *Room*

Username: *Maman* User NO: *2*

Data Collector: *Sergej Atamantschuk*  Iterations: 2

1. **Are you:**

Male

x Female

1. **How old are you**

18-25

26-40

x 41-55

Over 55

1. **Are you color blind in any way?**

x No

Yes (please describe)

1. **Describe your education background**

None

Elementary school

Middle schools

High school (attended but did not graduate)

High school degree

College (no degree)

x College degree

Graduate school (no degree)

Graduate school degree

Other (please explain)

1. **How would you describe your experience level in agriculture?**

Novice

Advanced

Competence

**x**  Expertise

1. **Do you have any experience using Android system?**

x No

Yes

Yes, but only older versions

Yes, but only newer versions

1. **How would you describe your general level of expertise in using Android applications**

x None

Low

High

Other (please describe)

1. **How would you describe your general level of smartphone experience?**

None

x Low

High

Other (please describe)

**Data Collection Sheet – Task 3 (User 2)**

**Correct steps:**

Auf Tab **SIGN UP** klicken. Auf Button mit **Email** oder **Telefonnummer** registrieren klicken. Daten eingeben. Auf **SIGN UP** Button klicken.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Correct steps | Correct? | | Time(2. Iteration) | Errors/Comments |
|  | Y | N | <=30 sec |  |
| **Touch** oder **Wischen zu dem** **SIGN UP** Tab, um zum Registrationsformular zu navigieren |  | N | Versucht sich direkt im *Sign in* Fenster zu registrieren, bekommt Fehlermeldung. |
| Button **Mit** **Email** oder **Mit** **Telefonnummer** anklicken | Y | Y |  |
| Jeweilige Datenfelder ausfüllen |  | Y | Checknoxen waren unklar |

**Data Collection Sheet – Task 4 (User 2)**

**Correct steps:**

Auf Listeneintrag klicken und direkt zur interaktiven Ansicht wechseln. Zur Anzeige der detaillierten Empfehlungen auf jeweilige Icons klicken.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Correct steps | Correct? | | Time(2. Iteration) | Errors/Comments |
|  | Y | N | <=2 sec |  | |
| Auf einen Listeneintrag in der Liste **klicken** | Y |  |  | |
| Daten interpretieren können | Y |  |  | Der Inhalt konnte sehr gut und schnell erkannt werden. Alle Icons wurden verstanden | |
| Falls nötig, **Ear-Icon** in der Actionbar anklicken, um die Benutzungsanleitung anzuhören | Y |  |  | Icon zum Anhören wurde erkannt | |
| Falls nötig, auf Text klicken, um den Inhalt anzuhören |  | N |  | Der Text, der vertont werden kann, sollte deutlicher dargestellt werden | |
| Zur ausführlichen Info auf beiliege Icons klicken | Y |  |  |  | |

### User 3

### User 4