SmartPlant

Anforderungsanalyse

|  |  |
| --- | --- |
| **Autor(en):** | Fynn Bucher |
| **Version:** | 1.0 |
| **Erstellt am:** | 6. Dezember 2017 |
| **Letzte Änderung:** | 14. Dezember 2017 |
| **Speicherort:** | ProjektICT-BZ |

Inhalt

[1 Revisionen 3](#_Toc467835242)

[2 Einleitung 4](#_Toc467835243)

[2.1 Systemidee 4](#_Toc467835244)

[2.2 Management Summary 4](#_Toc467835245)

[2.3 Projektteam 4](#_Toc467835246)

[2.4 Glossar 4](#_Toc467835247)

[3 Ausgangslage (IST), Problembereiche 6](#_Toc467835248)

[3.1 Beschreibung der Ausgangslage 6](#_Toc467835249)

[3.2 Problembereiche und Schwachstellen 6](#_Toc467835250)

[4 Ziele (SOLL) 7](#_Toc467835251)

[4.1 Beschreibung der Ziele 7](#_Toc467835252)

[4.2 Produktperspektive, Nutzen 7](#_Toc467835253)

[4.3 Zielkonflikte 7](#_Toc467835254)

[4.4 Abgrenzung 7](#_Toc467835255)

[5 Anforderungsanalyse 8](#_Toc467835256)

[5.1 Identifizierung der Akteure 8](#_Toc467835257)

[5.1.1 Kontextdiagramm 8](#_Toc467835258)

[5.2 Anforderungskatalog 8](#_Toc467835259)

[5.2.1 F.REQ: Funktionale Anforderungen 9](#_Toc467835260)

[5.2.2 NF.REQ: Nichtfunktionale Anforderungen 9](#_Toc467835261)

[6 Anhang 16](#_Toc467835275)

[6.1 Termine 16](#_Toc467835276)

[6.2 Referenzen 16](#_Toc467835277)

# Revisionen

|  |  |  |
| --- | --- | --- |
| **Datum** | **Name** | **Kommentar** |
| 06.12.2017 | Fynn Bucher | Dokument erstellt. |
| 13.12.2017 | Fynn Bucher | Zeitplanung + Meilensteine definiert. |
| 21.12.2017 | Fynn Bucher | Überarbeitung/Anpassung für neues Projekt. |
|  |  |  |
|  |  |  |
|  |  |  |

# Einleitung

## Systemidee

Lieben Sie pflanzen über alles? Fühlen Sie sich manchmal einsam? Dann haben wir genau das richtige für Sie. Mit Pflanzi der liebensvollen Pflanze fühlen Sie sich bestimmt nichtmehr alleine.

## Management Summary

Die Idee ist das ich mit Raspberry PI und mit Sensoren, die Luft-Bodenfeuchtigkeit und Temperatur messen Daten der Pflanze auslese. Diese Daten werden dann mit einem Programm verarbeitet. Hauptziel ist es herauszufinden ob die Pflanze gegossen werden muss oder nicht.

Die Idee ist das ich mit einer „App-Website“ eine Tamagotchi Pflanze giessen kann das wird dann per php auf eine Website übertragen. Das Resultat wird per Twitter, Mail oder auf der Pflanzi Website(welche ich natürlich selbst mache.) ausgeben. Die Nachricht sollte in etwa so aussehen: Ich verdurste!!!! (zu wenig Wasser) oder Ich vertrinke!!!!(zu viel Wasser) oder Mir geht’s supeeeeerrr!!!!(genug Wasser).

Das Produkt Pflanzi ermöglicht es dem Benutzer eine Übersicht über das Giessen seines Tamagotchis zu bekommen. Es wird dem Benutzer eine Nachricht der Pflanze per Twitter und oder E-Mail und oder SMS zugesendet oder man kann die Nachricht auch auf der Pflanzi Website einsehen.

## Projektteam

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Kürzel** | **Funktion** | **Kontakt (Telefon, Mail...)** |
| Urs Nussbaumer | nuu | Projektleiter | 041 371 24 28 urs.nussbaumer@ict-bz.ch |
| Fynn Bucher | buf | Entwickler | 079 703 22 36  fynnbucher@yahoo.com |
|  |  |  |  |
|  |  |  |  |

## Glossar

Erläutern Sie alle im Projekt verwendeten Fachbegriffe und Abkürzungen in alphabetischer Reihenfolge.

|  |  |
| --- | --- |
| **Begriff** | **Beschreibung** |
| PHP | Ist eine Open Source skript Sprache welche in HTML eingebettet werden kann. |
| HTML | HTML bedeutet HyperText Markup Language und ist eine Auszeichnungssprache für Websites. Das Grundgerüst einer Website wird mit HTML beschrieben und dann von einem Webbrowser wie Firefox interpretiert und angezeigt. |

# Ausgangslage (IST), Problembereiche

In diesem Kapitel wird die Ausgangslage beschrieben und es werden die Problembereiche identifiziert, die sich in der heutigen Situation zeigen.

## Beschreibung der Ausgangslage

Wer kennt es nicht man vergiesst sein Tamagotchi zu giessen/füttern und das nächste Mal wenn man drauf schaut ist es schon von uns gegangen.

## Problembereiche und Schwachstellen

Tamagotchi kann zu wenig oder zu viel gegossen werden und sterben.

# Ziele (SOLL)

In diesem Kapitel werden die übergeordneten Ziele beschrieben, die mit dem zu entwickelnden System erreicht werden sollen.

## Beschreibung der Ziele

Ziel ist es, das Pflanzen giessen zu optimieren um Schaden an den Pflanzen zu vermeiden.

## Produktperspektive, Nutzen

Bei erfolgreichem Abschluss wird der Benutzer/Auftraggeber eine Nachricht per Twitter oder E-Mail oder SMS oder auf der Pflanzi Website bekommen wenn sein Pflanzi Tamagotchi zu wenig oder zu viel Wasser hat. Somit kann er sie dem entsprechend giessen.

# Anforderungsanalyse

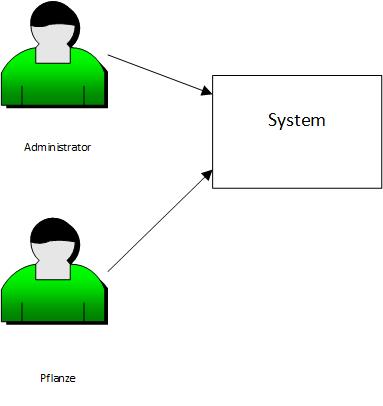
Die Anforderungen an das zu entwickelnde System definieren alle zu erfüllenden Eigenschaften oder die zu erbringende Leistung, sowie allfällige technische Vorgaben und weitere Forderungen des Kunden im Zusammenhang mit dem (den) zu erstellenden Produkt(en).

## Identifizierung der Akteure

Ein Akteur bezeichnet eine Rolle, die mit dem System interagiert. Es werden 4 Akteure unterschieden:

* **Administrator:** Administrator; Der Administrator hat Rechte auf das Verwalten (löschen von Beiträgen) des Twitter-Accounts der Pflanze und der Website der Pflanze.
* **Pflanze:** Ich will mitteilen (per E-Mail, Twitter, SMS oder auf meiner eigenen Website) ob ich genug zu viel oder zu wenig Wasser habe / bekomme.

### Kontextdiagramm



## Anforderungskatalog

Der Anforderungskatalog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein soll. Es wird unterschieden zwischen funktionalen und nicht-funktionalen Anforderungen.

Jede Anforderung wird in Alltagssprache, in der Form einer "User-Story", formuliert, mit einer ID zur späteren Identifizierung versehen sowie mittels 3 verschiedener Kategorien priorisiert: 1 = hohe Prorität, 2 = mittlere Priorität, 3 = keine Priorität. Diese drei Prioritäten repräsentieren die Verpflichtungen "must", "should" und "nice to have".

Im Verlauf der Zeit können neue Anforderungen hinzukommen und/oder bestehende Anforderungen können wegfallen. Anforderungen, die wegfallen, sind im Dokument zu belassen und als gestrichen zu markieren.

Hinter einer Anforderung kann ihn eckigen Klammern in der Form [Z#] eine Zusatzinformation hinterlegt werden und mit [F#] bzw. [F#, F#, …] können eine oder mehrere offene Fragen referenziert werden.

### F.REQ: Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| F.REQ.001 | Als Pflanze möchte ich meine Daten meinem Besitzer per Twitter mitteilen. | 1 |
| F.REQ.002 | Als Pflanze möchte ich meine Daten meinem per E-Mail mitteilen. | 1 |
| F.REQ.003 | Als Pflanze möchte ich meine Daten meinem per SMS mitteilen. | 1 |
| F.REQ.004 | Als Pflanze möchte ich meine eigene Website | 1 |
| F.REQ.005 | Als Pflanze möchte ich meine Daten im Blog-Format auf meiner Website ausgeben. | 2 |
| F.REQ.006 | Als Pflanze möchte ich eine Über mich Seite | 2 |
| F.REQ.007 | Als Pflanze möchte ich ein Stimmungsbarometer auf der Hauptseite haben. [Z1] | 2 |
| F.REQ.008 | Als Pflanze will ich zusätzlich zum Stimmungsbarometer ein Bild von mir haben welches meine Stimmung wiederspiegelt. | 2 |
| F.REQ.009 | Als Admin möchte ich mich auf der Website einloggen können. | 3 |
| F.REQ.010 | Als Admin möchte ich Blog Beiträge auf der Website löschen können. | 3 |
| F.REQ.011 | Als Admin möchte ich mich in den Twitter-Account der Pflanze einloggen können. | 3 |
| F.REQ.012 | Als Admin möchte ich Tweets der Pflanze löschen können. | 3 |

#### Zusatzinformationen

[Z1] Das Stimmungsbarometer wird hochgehen wenn die Pflanze genug Wasser hat und runtergehen wenn sie zu wenig hat.

### NF.REQ: Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| NF.REQ.001 | Jeder User-Input muss auf Schädlichkeit geprüft werden (sodass ein Einschleusen von JavaScript-Code, SQL-Injection, u.ä. nicht möglich ist) | 1 |
| NF.REQ.002 | Der Quellcode muss gut dokumentiert sein. | 1 |
| NF.REQ.003 | Das Design der Website soll schön und farblich pflanzenähnlich sein | 1 |
| NF.REQ.004 | Dass ich mich an meine Programmierrichtlinien halte | 1 |
| NF.REQ.005 | Pflanzen Daten sollen per Datenbank verwaltet werden. | 1 |

# Anhang

## Termine

|  |  |  |
| --- | --- | --- |
| **Datum** | **Ziel, Beschreibung, Meilenstein** | **Verantwortlich** |
| 13.12.2016 | Fertigstellung Zeitplanung, Überarbeitung dieses Dokumentes |  |
| 20.12.17 | Besprechung: Meilenstein 1 (Raspberry Pi zusammen gebaut und Daten können per Twitter gelesen werden.) |  |
| 22.12.17 | Besprechung: Meilenstein 2 (Pflanzen-Daten werden per SMS verschickt.) |  |
| 12.01.17 | Besprechung: Meilenstein 3 (Pflanzen-Website) |  |
| 17.0.17 | Besprechung: Meilenstein 4 (Admin-Login, Admin kann Beiträge löschen, Admin kann Tweets löschen.) |  |

## Referenzen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** | **Titel/Beschreibung** | **Autor** | **Dokument oder URL** |
| 1 | Zeitplanung & Meilensteine | Fynn Bucher | <https://github.com/MIGHTYFROGGOD/Projekt-SmartPlant/blob/master/Documents/Zeitplanung.xlsx> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |