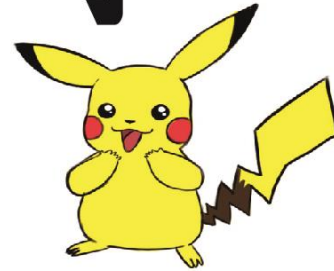


Project PI



Project PI Lastenheft

Version 4.0
20.02.2014

Autor:
Johannes Ucel, Yehezkel Sivan, Michael Stöger, Antonio Pavic

QS:
Johannes Ucel, Michael Stöger

Status:
Final

| <i>Version</i> | <i>Autor</i> | <i>QS</i> | <i>Datum</i> | <i>Status</i> | <i>Kommentar</i> |
|----------------|-------------------|-------------------|--------------|---------------|---|
| 1.0 | Sivan Yehezkel | Johannes Ucel | 17.02.2015 | Draft | Einführung, Zielbestimmung, Produkteinsatz |
| 2.0 | Michael Stöger | Johannes Ucel | 18.02.2015 | Draft | Produktfunktionen |
| 3.0 | Antonio Pavic | Johannes Ucel | 19.02.2015 | Draft | Produktleistungen, Produktdaten |
| 4.0 | Johannes Ucel | Michael Stöger | 20.02.2015 | Final | Qualitätsanforderungen, Zwingende Randbedingungen, Ergänzungen, Glossar, Korrekturen |

Inhaltsverzeichnis

| | |
|------------------------------------|-----------|
| 2 Zielbestimmung | 4 |
| 3 Produkteinsatz | 5 |
| 4 Produktfunktionen | 6 |
| 4.1 Systemsteuerung | 6 |
| 4.2 Sensitive Elemente | 7 |
| 4.3 Datenübertragung | 9 |
| 4.4 Steuerung über App | 9 |
| 5 Produktleistungen | 10 |
| 6 Produktdaten | 11 |
| 7 Qualitätsanforderungen | 12 |
| 8 Zwingende Randbedingungen | 13 |
| 8.1 Hardwareanforderungen | 13 |
| 8.2 Schnittstellen | 13 |
| 9 Ergänzungen | 14 |
| 9.1 Realisierung..... | 14 |
| 10 Glossar | 15 |

1 Einführung

In dem Projekt „Project PI“, geht es um ein **Plüschtier** (in unserem Fall ein „Pi-kachu“), welches mit verschiedenen **technischen Funktionen** ausgestattet wird.

Dieses Plüschtier soll speziell Kindern Spaß machen, da man zusätzlich zum Spielen auch aufgrund technischer Funktionen **Neugier zur Technik** entwickelt.

Unterstützt werden die technischen Funktionen des Plüschtiers von einer **Smartphone – App** (Android), wo man die Möglichkeit hat, mit dem Plüschtier andere Funktionen auszuprobieren.

Weiteres hat jeder die Möglichkeit, die Projektfortschritte in einem speziell für das Projekt kreierten **Web – Blog** mitzuverfolgen. Weiteres findet man dort auch **Tutorials** und **Hilfestellungen**, falls jemand ein ähnliches Projekt verfolgt.

2 Zielbestimmung

Ein Benutzer soll die Möglichkeit haben, interaktiv sein Plüschtier zu steuern. Dies ist mithilfe eines **Webportals** (auch als **mobile Ansicht** verfügbar) und einer **Smartphone – App** (Android) möglich. Dieses Webportal bzw. die App soll über ein anspruchsvolles Layout verfügen. Die Bedienung sollte möglichst simple sein, damit sich jeder Benutzer zurechtfindet. (Usability)

Besonders für die **Zielgruppe (Kinder)** soll das Plüschtier ein **modernes und alternatives** Spielzeug darstellen. Dies soll das **aktive Lernen** mit Spielzeugen und einen **sorgfältigen Umgang mit der Technik** fördern.

Speziell im technischen Bereich kann mithilfe dieses Projektes viel technisches Knowhow in den folgenden Bereichen entwickelt werden:

- Spracheingabe bzw. Ausgabe
- Foto-Auslöser mittels Taster und Sprachsteuerung
- HDMI – Übertragung zu einem Ausgabegerät
- Bewegungen durch Sensoren
- Augenfarbe der Stimmung anpassen (RGB – Ansteuerung)
- LED, welche abhängig von der Tageszeit ist (Erkennung mittels Helligkeitssensor)
- Notabschaltung über Innentemperatur (zusätzliche Warnausgabe)
- Annäherung erkennen mit Kommentaren vom Plüschtier

3 Produkteinsatz

Mit diesem Plüschtier sollen speziell Kinder die Möglichkeit haben, mit einem technisch **moderneren „Spielzeug“**, die Welt der Technik zu **erforschen**. Selbstverständlich dient das Plüschtier auch als **„herkömmliches Spielzeug“**.

Ein sehr großer Vorteil ist, dass die **programmiertechnischen Funktionen** jederzeit **erweiterbar** sind, da man die Möglichkeit hat, verschiedene **Libraries hinzuzufügen**.

Aktuell ist das System auf **Deutsch** verfügbar, abhängig von dem Andrang aus **anderssprachigen Regionen**, ist das **Hinzufügen** von weiteren **Sprachen** jederzeit möglich. **(multilingual)**

4 Produktfunktionen

4.1 Systemsteuerung

/LF010/ Die Software zur Steuerung muss immer laufen, bis sie vom Benutzer beendet wird. Dies umfasst:

- Systemstart
- Neustart bei Programmabsturz

/LF20/ Die Software kann über Fernsteuerung beendet werden

- Beenden über Android App

/LF30/ Die Software kann über Fernsteuerung neu gestartet werden

- Neustart über Android App

/LF40/ Das System kann über Fernsteuerung heruntergefahren werden

- Herunterfahren über Android App

/LF41/ Alarm bei Überhitzung

Wenn der Raspberry Pi überhitzt, gibt die Software den spezifischen Sound in einer Endlosschleife wieder. Sinkt die Temperatur unter eine bestimmte Temperatur, stoppt die Wiedergabe des Sounds.

/LF42/ Herunterfahren bei Überhitzung

Befindet sich die Temperatur über einem kritischen Punkt, schaltet der Raspberry Pi selbstständig ab.

/LF43/ Steuerung über SSH

Am Raspberry läuft ein ssh-Server, über den der Raspberry Pi gewartet werden kann.

4.2 Sensitive Elemente

Das Verhalten der Software soll größtenteils durch Sensoren wie:

- Taster
- Annäherungssensor
- Lichtsensor

bestimmt werden.

Die Software soll mit Ausgaben reagieren, welche folgendermaßen möglich sind:

- Licht
- Ton

/LF50/ Taster auf der rechten Hand

- Der Taster auf der rechten Hand löst den Fotomechanismus aus, zuerst wird der spezifische Sound abgespielt und dann mit der Raspberry Pi Kamera ein Foto geschossen.
- Dieses Foto wird in „/Fotos“ abgelegt

/LF60/ Lichtsensor – Hell

- Ändert sich die Helligkeit im Raum von Dunkel auf Hell, so wird der dazugehörige Sound abgespielt
- Die Farbe der Augen wird weiß

/LF70/ Lichtsensor – Dunkel

- Ändert sich die Helligkeit im Raum von Hell nach Dunkel, so wird der dazugehörige Sound abgespielt
- Die Farbe der Augen wird grün

/LF80/ Taster am Bauch wird gedrückt

- Die Farbe der Augen wird für die Dauer des abgespielten Sounds rot.
- Spezifische Sounds werden abgespielt

/LF90/ Taster auf der rechten Hand wird gedrückt

- Farbe der Augen ändert sich für die Dauer der Aktion auf gelb.
- „Donnerblitz“ Sound wird abgespielt

/LF100/ Taster am Ohr wird gedrückt

- Farbe der Augen ändert sich für die Dauer der Aktion auf blau.
- Spezifischer Ton wird abgespielt

/LF110/ Annäherung

- Farbe der Augen ändert sich für die Dauer der Aktion auf Türkis
- Spezifische Geräusche werden abgespielt

4.3 Datenübertragung

Fotos, welche über die Kamera aufgenommen werden, sollen über das Netzwerk auf andere Geräte übertragen werden.

/LF120/ Übertragung via Samba

- Am Raspberry Pi läuft ein Samba-Server, der den Ordner „/Fotos“ für alle freigibt

/LF130/ Übertragung via App

- Die Android App kann die Fotos vom Raspberry Pi in den Speicher des Android-Geräts verschieben

/LF131/ Darstellung via HDMI

- Über den HDMI Ausgang soll das letzte Bild, welches mit der Kamera aufgenommen wurde angezeigt werden

4.4 Steuerung über App

Zusätzlich zur Systemsteuerung sollen noch weitere Funktionen der Software über die Android App gesteuert werden.

/LF140/ Farbe der Augen ändern

- Über die App soll die Farbe der Augen verändert werden können, dazu sollen Buttons mit den möglichen Farben zur Verfügung stehen

/LF150/ Töne wiedergeben

- Über Buttons in der App ist es möglich verschiedene auf den Typ des Stofftiers angepasste Töne wiederzugeben

5 Produktleistungen

- /LL100/** Der Verbindungsaufbau zur Android – App soll in unter 15 Sekunden erfolgen
- /LL110/** Bei einem Verbindungstimeout wird ein Reconnect alle 10 Sekunden für 5 Mal versucht
- /LL120/** Die Verbindung mit dem Produkt soll einfach, schnell und ohne Probleme mittels W-LAN hergestellt werden.
- /LL130/** Es können keine zwei oder mehr Funktionen gleichzeitig ausgeführt werden.
- /LL140/** Bei einer HDMI – Verbindung soll das Anzeigen der Bilder bzw. das Laden weniger als 4 Sekunden dauern.

6 Produktdaten

Es sollen folgende Daten persistent gespeichert werden.

/LD100/ Alle Informationen zu dem Produkt:

- Speicherung der Bilder
- Speicherung der Töne
- Speicherung der LED-Farbcodes

/LD110/ Speicherung temporärer Daten

- **Datum**
- **Uhrzeit**
- **Aktuelles Wetter**
- **Neu erlernte Wörter aus der App**

7 Qualitätsanforderungen

Es wird großen Wert auf die **Zuverlässigkeit** und der **Benutzerfreundlichkeit** gelegt, denn wenn sich ein Benutzer (in diesem Fall eher Kinder) bei einem Produkt nicht auskennt bzw. wohl fühlt, ist es unwahrscheinlich, dass er dieses Produkt weiterhin verwenden wird.

Selbiges gilt für die **Android App** bzw. für den **Web – Blog**. Hier wird besonders auf ein **simples User Interface** gesetzt, wo alle Funktionen leicht erreichbar sind und nichts versteckt im Menü ist.

Die Aktionen vom Benutzer sollten vom Server **schnell** und **fehlerfrei** übernommen werden. Des Weiteren soll das System problemlos jederzeit **erweiterbar** sein (Updates, Patches) und auch auf anderen Systemen **ohne Einschränkungen** funktionieren.

Folgende Qualitätsanforderungen sind geplant:

| Produktqualität | sehr gut | Gut | normal | Irrelevant |
|-----------------|----------|-----|--------|------------|
| Funktionalität | | x | | |
| Zuverlässigkeit | x | | | |
| Benutzbarkeit | x | | | |
| Effizienz | | x | | |
| Änderbarkeit | | x | | |
| Übertragbarkeit | X | | | |
| Sicherheit | x | | | |
| Erlernbarkeit | | x | | |
| Zeitverhalten | x | | | |
| Stabilität | | x | | |

8 Zwingende Randbedingungen

8.1 Hardwareanforderungen

Die Realisierung des Systems wird auf einem Raspberry A+ stattfinden.

Dieser verfügt über eine ARM11 – CPU, welche mit 700 MHz taktet. Weiteres verfügt er über 256 MB RAM und einem USB 2.0-/LAN-/HDMI – Anschluss.

Auf dem Raspberry A+ wird das Betriebssystem Raspbian laufen.

Für den Web – Blog wird ein Server mit PHP bzw. MySQL – Unterstützung benötigt.

Des Weiteren läuft die Android App für Android-Versionen ab 4.1 (Jelly Bean)

8.2 Schnittstellen

Abhängig von der Funktion werden verschiedene Schnittstellen verwendet.

Bei einer Übertragung der Fotos zu einem Fernseher/Monitor wird die HDMI – Schnittstelle verwendet.

Das Betriebssystem wird über eine SD – Karte geladen.

Die Dateiübertragung zu einem Rechner kann entweder über WLAN oder direkt über ein LAN – Kabel erfolgen.

9 Ergänzungen

9.1 Realisierung

Das vorliegende System wird mit Python realisiert, da die Programmierung mit Java aufwändiger wäre.

Der Web – Blog wird mit **PHP** und **MySQL** werden (Datenbanken). Des Weiteren wird für das Web – Portal das Erzeugen von HTML-, CSS und JavaScript – Dateien notwendig sein

Die Android – App wird mit Java programmiert (inkl. XML)

10 Glossar

PHP ist eine **serverseitige** Programmiersprache, das heißt, der Code wird vom Server direkt in z.B. einen HTML – Code umgewandelt und der Benutzer erhält dann diese Homepage. Der PHP – Code kann vom Benutzer nie eingesehen werden

MySQL ist ein Datenbankverwaltungssystem

Python ist eine schwachtypisierte Programmiersprache