Tally
digitale Strichliste am Raspberry Pi

Nicolai Tegtmeier
Dominik Scheffler
Philipp Frieling
Sebastian Reinke

23.06.2015

Inhaltsverzeichnis

1 Projektvorfeld **3**

1.1 Kaffeestrichliste bisher 3

2 Rahmenbedingungen **3**

2.1 Aufgabe - die neue Kaffeestrichliste 3

3 Ziele des Projekts **3**

3.1 Zielbestimmung 4

3.1.1 Der Benutzer Account 4

3.1.2 Das Programm 4

3.1.3 Der Administrator Account 4

3.2 Produkteinsatz 4

3.2.1 Anwendungsbereiche 4

3.2.2 Zielgruppen 4

3.2.3 Betriebsbedingungen 4

3.3 Produktumgebung 5

3.3.1 Software 5

3.3.2 Hardware 5

3.3.3 Orgware 5

3.4 Produktfunktion 5

3.4.1 Benutzerfunktion 5

3.5 Programmfunktion 6

3.6 Serverfunktion 6

3.6.1 Datenbank 6

3.7 Benutzerschnittstelle 6

4 Meilensteine des Projektes **6**

4.1 Raspberry Pi 2 - Die Hardware mit der passenden Software . 6

4.1.1 Betriebssystem und Verbindung zu anderen Rechnern 6

4.1.2 Der Webserver 7

4.1.3 Die passende Datenbank 7

4.1.4 Der Touchscreen 8

4.1.5 Produkte scannen 9

4.1.6 Wlan für den Raspberry 9

4.1.7 angepasster Bootscreen 10

4.1.8 Backups einrichten 11

4.2 Sonderzeichen 12

5 Auflistungen und Aufzhlungen **12**

6 Tabellen, Grafiken und Gleitobjekte **13**

6.1 Grafiken 13

6.2 Tabellen 14

6.3 Bewegliche Objekte 14

1 Projektvorfeld

1.1 Kaffeestrichliste bisher

In kleineren Unternehmen und Arbeitsgruppen gibt es oft eine zentralen Kaffee/ Getränke -automaten und Snacks, an der sich jeder Mitarbeiter bedienen kann. Um die Getränke und Snacks kaufen zu können wird eine Strichliste auf Papier, meist in der Nähe des Automaten oder der Snacks, angebracht damit sich jeder Mitarbeiter eintragen kann, was er gekauft hat. Diese Liste wird oft zur besseren Verwaltung in eine Datenbank oder Tabelle eingepflegt. Hier passiert es jedoch oft das die Strichliste nicht sehr leserlich ist, und es bei der Übertragung zu fehlern kommen kann, so dass ein Minus in der Kasse entsteht. Au/sserdem ist diese Methode für den Verantwortlichen sehr zeitaufwendig.

Um diesem Problem entgegen zu wirken, erarbeiten wird eine neue Methode um diese Daten 'digital' speichern zu können.

2 Rahmenbedingungen

2.1 Aufgabe - die neue Kaffeestrichliste

Um eine möglichst Energiesparende und handliche Lösung zu finden, sollte die neue Strichliste auf einem Raspberry Pi 2 realisiert werden. Zunächst soll eine passende Benutzeroberfläche für den Raspberry entwickelt werden, damit der Benutzer am Raspberry selber seine Einkäufe verbuchen kann. Dies soll mithilfe der integrierten Entwicklungsumgebung 'Qt Creator', die besonders zur Entwicklung von von plattformunabhängigen C++ Programmen gedacht ist, entwickelt werden.

Mithilfe eines Webservers, der ebenfalls auf dem Raspberry läuft, soll die Administration von jedem Computer im selben Netzwerk über eine Weboberfläche möglich sein. Hier sollen auch die Benutzer ihre Daten einsehen und ändern können. Dazu wird ein MySQL/ SQLite Datenbank, sowie PHP Unterstützung benötigt. Mittels eines Barcodescanners soll es möglich sein am Raspberry Produkte ein zu scannen.

3 Ziele des Projekts

Die neue 'digitale Kaffeestrichliste' wird auf Basis eines Raspberry Pi 2 entwickelt, um die alte Strichliste abzulösen. Dazu wurden folgende zu erreichende Ziele festgelegt:

3.1 Zielbestimmung

3.1.1 Der Benutzer Account

Über den Benutzer Account soll der Benutzer sich am Raspberry selbst und an der Weboberfläche anmelden. Am Raspberry selbst können Getränke und Snacks gekauft werden. Mithilfe der Weboberfläche kann der Benutzer Statistiken einsehen und sein Passwort ändern.

3.1.2 Das Programm

Das Programm das auf dem Raspberry läuft aktualisiert die Anzahl der Produkte im Lagerbestand automatisch, gibt Meldungen bei zu geringem Warebestand aus und gibt generelle Fehlermeldungen aus

3.1.3 Der Administrator Account

Die Administration mithilfe des Administrator Accounts findet ausschliesslich über die Weboberfläche statt. Hier kann der Administrator Accounts hinzufügen und verwalten, Waren hinzufügen und verwalten und den Lagerbestand verwalten.

3.2 Produkteinsatz

3.2.1 Anwendungsbereiche

Mitarbeiter, beziehungsweise die Administratoren, können eine Strichliste 'digital' anlegen. Diese fasst den zu bezahlenden Betrag für die Mitarbeiter zusammen.

3.2.2 Zielgruppen

Personengruppen und Unternehmen bei denen Getränke und Snacks privat für alle angeboten und privat bezahlt werden, jedoch Schwierigkeiten mit der Handhabung einer herkömmlichen Strichliste haben und den Bestand an Getränken und Snacks erfassen wollen.

3.2.3 Betriebsbedingungen

Die Strichliste soll mö glichst Wartungsarm und einen niedrigen Stromverbrauch haben. Desweiteren soll sie täglich vierundzwanzig Stunden laufen.

3.3 Produktumgebung

Das Produkt kann unabhängig an jedem Ort betrieben werden, solange eine Stromquelle in der nähe ist.

3.3.1 Software

Die einzige Software die vom Benutzer benötigt wird ist ein aktueller Webbrowser.

3.3.2 Hardware

Der Benutzer kann mit jedem Internetfähigen Gerät die Weboberfläche erreichen.

3.3.3 Orgware

Der Administrator kann die Betriebsparameter konfigurieren.

3.4 Produktfunktion

3.4.1 Benutzerfunktion

Ein im System registrierter Benutzer kann das System erst nutzen, wenn er angemeldet ist. Nur ein Administrator kann einen Benutzer anlegen und ihm einen Benutzernamen sowie Passwort zuwesien. Sobald ein Benutzer registriert ist kann er sich sowohl am Raspberry als auch an der Weboberfläche anmelden. Dafür benötigt er seinen Benutzernamen und sein Passwort. Abmelden ist bei beiden Oberflächen jederzeit möglich.

Der Administrator kann über die Weboberfläche Produkte hinzufügen,entfernen und deren Details ändern. Benutzer können über die Weboberfläche Favoriten einrichten und Statistiken einsehen.

Am Raspberry kann der registrierte und angemeldete Benutzer ein Produkt aus einer Liste wählen oder mithilfe eines Barcodescanners ein Prdoukt einscannen, welches im Warenkorb erscheint. Desweiteren kann er die Anzahl der Produkte erhöhen und seinen gesamten Warenkorb einsehen. Wenn er alle Waren im Warenkorb hat, kann er zur Kasse und die Waren durch einen Klick bezahlen, wodurch sein Konto belastet wird. Ein Abbruch oder das erreichen des Hauptmenüs ist jederzeit möglich.

3.5 Programmfunktion

Beim Kauf eines Produktes aktualisiert das Programm automatisch den Warenbestand. Bei geringem Warenbestand wird eine Warnmeldung ausgegeben und bei fehlen des Artikels wird dieser entfernt.

3.6 Serverfunktion

Auf dem Raspberry soll ein Apache 2 Server mit PHP Unterstützung laufen.

3.6.1 Datenbank

Als Datenbank soll die resourcenschonendere SQLite Datenbank verwendet werden

3.7 Benutzerschnittstelle

Die Bedienung des Raspberry erfolgt über einen eingebauten Touchscreen am Raspberry. Auß()erdem wird ein Barcodescanner installiert um Produkte einscannen zu können.

Zentrierter Text hingegen ist schon eher nützlich.

4 Meilensteine des Projektes

4.1 Raspberry Pi 2 - Die Hardware mit der passenden Software

Mit dem Raspberry Pi 2 war schon im Projektvorfeld eine passende Basis für das Projekt gelegt. Ausser dem Raspberry Pi 2 war zunächst ein 3,5 Zoll Touchscreen der Firma admatec mit dem Namen C-Berry vorhanden, das mittels eines Adapterboards an der GPIO Stiftleiste des Raspberry angesteckt wird. Das Display besitzt eine Auflösung von 320x240 Pixeln und wird komplett vom Raspberry mit Strom versorgt.

4.1.1 Betriebssystem und Verbindung zu anderen Rechnern

Als erster Schritt wurde in Bezug auf die Hardware und die Software die auf dem Raspberry laufen soll entschieden das es von Vorteil ist das speziell

für den Raspberry zugeschnittene Raspbian Betriebssystem zu verwenden. Als Alternative stand eine Ubuntu Version ‚speziell für ARM-Prozessoren entwickelt, zu Verfügung. Hier stellte sich aber schnell nach einigen Recherchen heraus das Raspbian durch die Optimierung auf den Raspberry und den guten Support am besten für das Projekt geeignet ist. Um die weitere Einrichtung des Raspberrys zu vereinfachen wurde, neben der vorkonfigurierten und installierten SSH Verbindung zur Verwaltung über ein Konsolenprogramm (Windows: Putty), eine freie Implementierung des Remote Desktop Protocols für Linux Names 'xrdp' installiert. .

```
1 sudo apt-get install xrdp
```

Damit ist eine einfache Remote Desktopverbindung zum Raspberry möglich.

4.1.2 Der Webserver

Als nächster Schritt wurde ein vollständiger Webserver auf dem Raspberry eingerichtet. Hierzu wurde zunächst ein Apache Server der Version 2 installiert .

```
1 sudo apt-get install apache2
```

der über eine hohe Stabilität und Geschwindigkeit verfügt und serverseitig die Skriptsprache PHP unterstützt. Im Anschluss wurde das PHP5 Paket installiert um eine volle Unterstützung für die kommende Website zu gewährleisten.

4.1.3 Die passende Datenbank

Nun stand die Auswahl des passenden Datenbankverwaltungssystems an. Zur Auswahl standen die Systeme MySQL und SQLite. Bei Recherchen zu den beiden Datenbanken stellte sich schnell heraus das die MySQL Datenbank zwar auf dem Raspberry lauffähig ist aber keine hohe Stabilität und sehr resourcenlastig auf dem Raspberry läuft. Daher wurde das SQLite Datenbanksystem ausgewählt. .

```
1 apt-get install sqlite3
2 apt-get install php5-sqlite
```

SQLite ist eine relationale Datenbank und benötigt im Gegensatz zu MySQL keine ständig laufende Software da die Datenbank aus einer einzigen, wenigen Kilobyte grossen, Datei besteht.

4.1.4 Der Touchscreen

Als erstes wurde der Treiber für den bereitgestellte C-Berry Touchscreen installiert .

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.36.tar.gz
tar zxvf bcm2835-1.36.tar.gz
cd bcm2835-1.36
./configure
make
sudo make check
sudo make install
```

und im anschluss die Beispielsoftware zum anzeigen von Bildern installiert .

```
wget http://admatec.de/sites/default/files/downloads/C-Berry.tar
    .gz
tar zxvf C-Berry.tar.gz
cd C-Berry/SW/tft_test
make
sudo ./tft_test
```

Nach ersten Tests mit dem C-Berry Touchscreen fiel auf das es weniger gut als Primäres Display geeignet ist da vom Desktop ein Screenshot erstellt wird und dieser auf dem Display ausgegeben wird. Starke Verzögerungen im Bildaufbau und eine schlechte Bedienung mithilfe des Touchscreens waren die Folge.

Um dem Problem entegegen zu wirken musste ein neuer Touchscreen mit besserer Anbindung an den Rapsberry und einer höheren Bildwiederholungsrate gesucht werden. Dabei kam die Idee auf ein 7 Zoll resistiven Touchscreen von Pollin zu verwenden das durch seine Grösse viel Platz für das Tally Programm bietet und eine sehr gute Touch Bedienung verspricht. Von Vorteil war auch die eigene externe Grafikeinheit die eine hohe Bildwiederholungsrateermöglicht und ein externer USB- Touchcontroller.

Jedoch stellte die externe Stromversorgung ein Problem dar, da man jeweils ein Stromkabel für den Raspberry und eins für das Display benötigt, damit äre eine Energiesparende Lösung ebenfalls nicht möglich.

Deshalb fiel die Entscheidung zugunsten des 3,5 Zoll Touchscreen 4DPi-35 von 4D System aus. Der Touchscreen besitzt eine Auflösung von 480x320 Pixeln und durch die High Speed 48Mhz SPI Verbindung werden hohe und konstante Bildwiederholungsraten ermöglicht. Die Vorteile bei diesem Display waren die kleine Bauform und keine zusätzlich benötigte Stromquelle. Ausserdem wird vom Hersteller direkt ein passender Kernel für Raspbian zur Verfügung gestellt. .

```
wget http://www.4dsystems.com.au /downloads/4DPi/kernel4dpi_1
    .3-3_pi2.deb

sudo dpkg -i kernel4dpi_1.3-3_pi2.deb
```

Nachdem das erforderliche Paket installiert wurde, musste der Raspberry herunter gefahren werden, das Display an den GPIO-Ports angebracht werden und der Raspberry wieder hoch gefahren werden. Danach war ein sofortiger Betrieb möglich. Dadurch das das Display direkt als Primäres Display erkannt wird ist kein HDMI Bildschirm mehr notwendig und die Bedienung kann ausschliesslich über den Touchscreen erfolgen.

4.1.5 Produkte scannen

Zur vereinfachten Eingabe von Produkten soll ausserdem ein Barcodescanner angebracht werden. Hier sollte entweder ein handeslüblicher Barcodescanner per USB an den Raspberry angeschlossen werden, der Barcode mittels USB Webcam und passender Software oder aber ein Barcodescannermodul an den GPIO's des Raspberry als Lösung dienen.

Da jedoch das vorhandene Barcodescannermodul nur mit 3V betrieben werden kann, des Raspberry aber 5V an den GPIO's zur Verfügung stellt, wurde überlegt eine Webcam an den Raspberry anzuschliessen und mittels einer Software das Bild auf Barcodes zu untersuchen. Die Wahl fiel allerdings auf das Scannen mittels USB Barcodescanner, das dies eine einfacherere Handhabung mit sich bringt und der Barcodescanner direkt als USB Tastatur erkannt wird.

4.1.6 Wlan für den Raspberry

Damit am Raspberry nicht immer ein Patchkabel angeschlossen werden muss, wurde der Edimax WLAN USB Stick am Raspberry angeschlossen. Der Edimax WLAN Stick wird automatisch von Raspbian erkannt da schon alle erforderlichen Pakete und Treiber installiert sind, weshalb dieser WLAN Stick empfohlen wird falls man WLAN am Raspberry benutzen möchte. Zunächst wurde in der automatische 'Power Saving' Modus abgeschaltet. Hierzu wurde eine Konfigurationsdatei für den Treiber des WLAN Sticks erstellt. .

```
1| sudo nano /etc/modprobe.d/8192cu.conf
```

mit folgendem Inhalt .

```
1| options 8192cu rtw_power_mgnt=0 rtw_enusbss=0
```

Um nun eine Verbindung mit dem Wlan herzustellen wurde die folgende Datei .

```
1| sudo nano /etc/network/interfaces
```

um folgenden Inhalt ergänzt .

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ap-scan 1
wpa-scan-ssid 1
wpa-ssid "WLAN-NAME"
wpa-psk "WLAN-PASSWORT"
```

damit eine Verbindung mit dem WLAN Netz hergestellt werden kann. Anschliessend muss nur noch per .

```
sudo service networking restart
```

der Netzwerkdienst neu gestartet werden und eine Verbindung mit dem angegebenen WLAN Netz wird hergestellt.

4.1.7 angepasster Bootscreen

Während eines Meetings kam die Idee auf den Bootscreen zu verändern und ein Bild an zu zeigen. Dazu wird eine weitere Software benötigt, der 'Frame Buffer Imageviewer'. .

```
sudo apt-get install fbi
```

Danach muss ein passendes Skript geschrieben werden welches das Bild beim booten anzeigt. .

```
sudo nano asplashscreen
```

.

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          asplashscreen
# Required-Start:
# Required-Stop:
# Should-Start:
# Default-Start:     S
# Default-Stop:
# Short-Description: Show custom splashscreen
# Description:       Show custom splashscreen
### END INIT INFO

do_start () {

    /usr/bin/fbi -T 1 -noverbose -a /etc/splash.png
    exit 0
}

case "$1" in
  start|"")
    do_start
```

```
22      ;;
23    restart|reload|force-reload)
24      echo "Error: argument '$1' not supported" >&2
25      exit 3
26      ;;
27    stop)
28      # No-op
29      ;;
30    status)
31      exit 0
32      ;;
33    *)
34      echo "Usage: asplashscreen [start|stop]" >&2
35      exit 3
36      ;;
37  esac
38
39  :
```

Dieses Skript wird nun in den Ordner verschoben damit es beim Start ausgeführt werden kann .

```
1  sudo mv asplashscreen /etc/init.d/asplashscreen
```

Als letztes wird das Skript für den Raspberry ausführbar gemacht und als Bootscript registriert .

```
1  sudo chmod a+x /etc/init.d/asplashscreen
2  sudo insserv /etc/init.d/asplashscreen
```

4.1.8 Backups einrichten

Wie wir im vorangegangen Text gesehen haben, sind die Zeilenumbche automatisch ganz gut. Neue Abschnitte werden durch eine Leerzeile erzeugt. Wenn man trotzdem eine Zeile in einem Absatz abbrechen will, so geht das natrlich auch.
Aber eigentlich sehen diese Umbche seltsam aus und man sollte sie vermeiden. Da ist es doch in der Regel besser, gleich einen neuen Absatz zu beginnen. In Abschnitt 4.2 sehen wir, wie eine gute Absatzstruktur aussieht. brigens knnen wir auch einen neuen Seitenbeginn erzwingen.

Aber auch das sollte man in der Regel vermeiden. LaTeX bricht Seiten in der Regel selbst vernnftig um, und wenn man spter noch Text einfgt, stehen manuelle Seitenumbrche fast immer an der falschen Stelle.[1]

4.2 Sonderzeichen

Viele Sonderzeichen haben in LaTeX eine spezielle Bedeutung und drfen nicht einfach so verwendet werden, sondern werden durch spezielle Befehle erzeugt. Das ist ganz hnlich wie HTML. Zum Beispiel ist & ein Trennzeichen in Tabellen, $ signalisiert den Beginn und das Ende von mathematischem Text, % macht eine Zeile zu einem Kommentar, der von LaTeX natrlich ignoriert wird, { und } sind fr die Parametebergabe bei Befehlen reserviert, _ und ^ haben ihre Bedeutung bei dem Setzen von mathematischen Formeln und # ist erforderlich bei selbst definierten Befehlen.

5 Auflistungen und Aufzhlungen

Wir fassen zusammen, was wir bisher knnen:

- Die Schrift anpassen
- Den Text ausrichten. Da gab es folgende Mglichkeiten:
  - Blocksatz
  - linksndig
  - rechtsbndig
  - zentriert. Zentrierte Text ist vor allem gut um
    - Text hervorzuheben
    - Objekte, wie Tabellen und Grafiken zu zentrieren
- Umbrche erzwingen
- Sonderzeichen setzen

Bisher knnen wir noch nicht:

1. Tabellen erzeugen

[1] Manuelle Seitenumbrche sollten deswegen hchstens dann eingefgt werden, wenn das Dokument vollstndig fertig ist und die Seitenumbrche von LaTeX nicht zufriedenstellend waren.

2. Grafiken einbinden
3. mathematischen Text setzen. Dazu gehrt
   (a) weitere mathematische Sonderzeichen, wie z.B.
      i. griechische Buchstaben, etwa α, ζ, usw.
      ii. spezielle Akzente, wie $\vec{a}$ oder $\ddot{x}$
      iii. echte Sonderzeichen, wie $\oplus$ oder $\perp$
      iv. alle Mglichen Klammern, wie $\lfloor x \rfloor$
      v. und noch vieles mehr...
   (b) spezielle mathematische Objekte, wie Matrizen
   (c) automatische Nummerierung von Formeln
4. Referenzen und Bibliographie erzeugen
5. Prentationsfolien erstellen

Offensichtlich kann man hier ziemlich tief schachteln. Ob das allerdings immer sinnvoll ist?

6 Tabellen, Grafiken und Gleitobjekte

6.1 Grafiken

Hier hatten wir Gtig kommt?

6.2 Tabellen

Name	gemessen von Dr. Tex		
	Alter (Jahre)	Gre (in cm)	Gewicht (in kg)
Andreas	7	120	25
	10	141	34
	14	163	50
Beate	6	110	22
	9	138	32
	13	156	46
Tina	8	132	30
	11	151	43
	15	174	51

6.3 Bewegliche Objekte

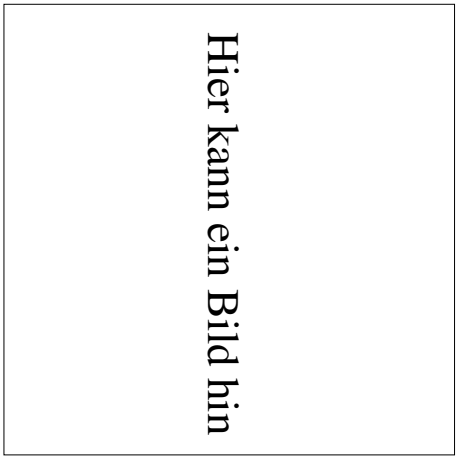

Abbildung 1: Dasselbe nochmal als Gleitobjekt.