



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN  
University of Applied Sciences



Beuth Hochschule für Technik Berlin  
Fachbereich VI - Informatik und Medien  
Luxemburger Str. 10, 13353 Berlin

# Bachelorarbeit

## Entwicklung einer Datenbank-Applikation für das "acaLoan-Raspi"-Projekt des PSE-Labors

Development of a database application for the "acaLoan-Raspi"  
project of the PSE laboratory

Oleksandra Baga

Beuth Hochschule für Technik Berlin  
Matrikelnummer 849852  
Bachelor of Engineering (B.Eng.)  
Computer Engineering - Embedded Systems  
E-Mail: oleksandra.baga@gmail.com

*Supervisor* Prof. Dr. Christian Forler  
Fachbereich VI - Informatik und Medien  
Beuth Hochschule für Technik Berlin

31.10.2020

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Aufgabestellung . . . . .	1
1.2	Technische Basis und Themengebiet . . . . .	2
<b>2</b>	<b>Der theoretische Hintergrund</b>	<b>5</b>
2.1	Über Raspberry Pi Board und OS . . . . .	6
2.2	Kontaktlose Chipkartentechnik MIFARE . . . . .	6
2.3	Sender-Empfänger-System mit RFID . . . . .	7
2.4	Datenbanken mit Python und SQLite . . . . .	7
2.5	HTTP für Design der verteilten Systeme . . . . .	7
2.6	Django Framework . . . . .	7
2.7	API . . . . .	7
2.8	Endliche Zustandsmaschine . . . . .	7
2.9	Clientseitiges JavaScript . . . . .	7
<b>3</b>	<b>Systemdesign</b>	<b>8</b>
3.1	Systemarchitektur . . . . .	8
3.2	User Stories . . . . .	8
3.3	UML . . . . .	8
3.4	Endliche Zustandsmaschine . . . . .	8
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Register-Client . . . . .	9
4.2	Server . . . . .	9
4.3	Display-Client . . . . .	9
<b>5</b>	<b>Application</b>	<b>10</b>
5.1	Usage in laboratory . . . . .	10
<b>6</b>	<b>Results and conclusion</b>	<b>11</b>
6.1	Results . . . . .	11
6.2	Conclusion . . . . .	11
<b>A</b>	<b>Appendix</b>	<b>12</b>
A.1	RFID . . . . .	12

<b>B</b>	<b>Abkürzungsverzeichnis</b>	<b>13</b>
<b>C</b>	<b>Abbildungsverzeichnis</b>	<b>14</b>

# Einleitung

## 1.1 Motivation und Aufgabestellung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung einer Datenbank-Applikation für das PSE-Labor<sup>1</sup> (Labor für Pervasive Systems Engineering), das sich an der Beuth Hochschule für Technik Berlin befindet und seit fast 5 Jahre ein wichtiger Teil des Studiums im Studiengang Technische Informatik und Medieninformatik ist. Die zwei Mitarbeiter, Andreas Döpfens und Brian Schüler, beitragen selbstmotiviert zur den Projekte, die zwar räumlich im PSE-Labor der Beuth-Hochschule sind und betrieben werden können<sup>2</sup>, wurden aber als "das acaLab" genannt, bei dem es sich sozusagen um ein virtuelles Labor in einem realen Labor handelt. Die acaLab-Projekte sollen als interessante Mitarbeiter-Beiträge neben den individuellen Tätigkeiten im Fachbereich 6 der Beuth-Hochschule bereichern und werden deshalb auch exemplarisch jedes Jahr auf der Langen Nacht der Wissenschaften<sup>3</sup> gezeigt. An dieser Stelle ist auch noch anzumerken, die acaLab-Projekte aus den Mitteln des PSE-Labors finanziert werden und damit möglichst viele Studierende für lehr- und erkenntnisreichen Abschlussarbeiten eingeladen sind, die der Vergabe die Aufgaben aus den acaLab-Projekten als Abschlussarbeiten spenden dem Fachbereich ein Budget. Die Erkenntnisse und Ergebnisse aus den acaProjekt-Tätigkeiten sind später in die Lehre einfließen zu lassen. Meine Abschlussarbeit widmet sich dem neuen Projekt des Labors namens "acaLoan-Raspi".

Während im PSE-Labor stattfindenden Übungsveranstaltungen im Studiengang Technische Informatik werden vorhandene im PSE-Labor die Raspberry Pi Minicomputers (kurz: Raspi) an die Studierenden verliehen. Zu Beginn einer Laborübung werden die Raspi Boards den Studierenden vom Lehrkraft, der die Übung betreut, übergeben und am Ende der Laborübung zurückgezogen. Aus 16 vorhandenen im Labor Raspis, die markierte mit den Nummern 12-16 von Studierenden nach Hause (home-loan) genommen werden können.

Nachweislich ist das Vorgehen oft mit Reihe von Problemen verknüpft, die sich jedes Semester und fast jedes Mal wiederholen. Die folgenden Problemen wurden

<sup>1</sup><https://www.http://labor.beuth-hochschule.de/pse>

<sup>2</sup><https://labor.beuth-hochschule.de/pse/acalab-aktivitaeten-und-abschlussarbeiten/>

<sup>3</sup><https://www.langenachtderwissenschaften.de/>

von Mitarbeitern des Labors bereits festgestellt und regelmäßig verlangsamen den Prozess der Verleihung und Übungsführung:

- Studierende kennen ihre am Semesteranfang zugewiesene Gruppennummer auch nach mehreren Wochen nicht und geben den Lehrkraft einen Board mit einer falschen Registriernummer, der einer anderen Gruppe früher zugewiesen wurde und nur von der zugewiesenen Gruppe benutzt werden darf.
- Studierende versuchen einen Board nach Hause auszuleihen, der zu den Lab-Boards gehört und nur im Labor während der Übungszeit verliert werden darf. Außerplanmäßig von Studierenden darf Lab-Board nicht ausgeliehen und auch mit nach Hause (home-loan) nicht genommen werden.
- Es gibt ein Verwaltungsaufwand für die ausleihbaren Home-Boards, die von den Studierenden für jeweils eine Woche mit nach Hause genommen werden können. Die Mitarbeiter müssen handlich die Studentennamen, Matrikelnummer, Board und Zeit am Zettel registrieren und in einer Woche überprüfen, ob alle ausgeliehenen Boards pünktlich ins Labor zurückgekommen sind.
- Erfahrungsgemäß können Studierende nach Ablauf der Frist ein Ausleihgerät in einem sehr üblen Zustand der Verschmutzung oder Zerstörung zurückgeben, dass es besteht eine Notwendigkeit den Zustand des Gerätes stets zu kontrollieren, damit es immer bekannt wird, zum welchen Zeitraum Raspi Board zum letzten Mal funktionsfähig war und von wem ausgeliehen wurde.
- Falls gilt ein Raspi Board als verloren, es sollte eine Möglichkeit geben, alle vorherigen Ausleihen anzuschauen und festzustellen, von welchem Studierende es ausgeliehen und nicht zurückgegeben wurde. Mit den Zettelchen, auf denen einen Name von Studierende und eine Board Nummer gemerkt werden, ist es zu aufwändig nachvollziehen.

Somit ist schlusszufolgern, dass eine Notwendigkeit das Verleihprozedere für die Loan-Boards (Lab und Home) mit modernen Mitteln der Technischen Informatik zu lösen schon dringend besteht und eine lohnenswerte Aufgabe für zukünftige Abschlussarbeit ist.

## 1.2 Technische Basis und Themengebiet

Obwohl das Thema der Bachelorarbeit "Entwicklung eine Datenbank-Applikation" lautet, ist es keine einzige Aufgabe nur eine Datenbank zur Ausleihverwaltung zu

schrieben ist. Es lässt sich die folgenden Struktur definieren, indem drei verteilte Teilen zu entwickeln ist.

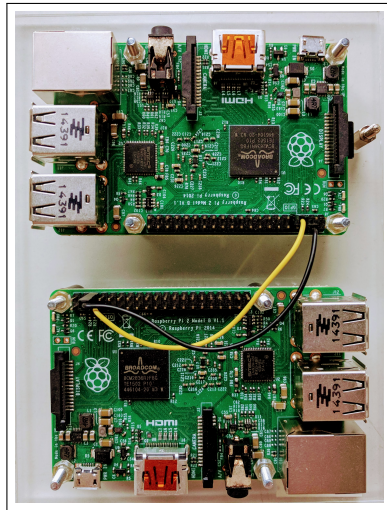
Erstens wird an einem uComputer ein sogenannten Register-Client realisiert. Dafür ein RFID-Leser an uComputer angeschlossen wird. Register-Client ist neben dem Eingangstür eines kleinen Lagerraums des PSE-Labors zu platzieren ist, wo Raspi-Boards aufbewahrt werden. Es ist geplant, dass Studierende einen Board selbst aus dem Fach nehmen könnte und dann mit Hilfe des Register-Clients den genommenen Board auf sich oder seine Gruppe registrieren lassen. Der Register-Client hat selbst keinen Zugriff zur Datenbank und sollte nur die abgelesene Daten von der Smartcard der Studierende zur Server schicken.

Zweitens ist ein Display-Client zu entwickeln, der den Studierenden es zulässt, die Begrüßung des System und eine Beschreibung die für Ausleihe notwendigen Schritten zu sehen. Es sollte in einem Browser-Fenster die aktuelle Server-Kommunikation und Auskunft angezeigt wird, ob die Ausleihe gelang oder ein Fehler aufgetreten war. Ein Android/iOS-Tablett ist eine gute Wahl für die technische Realisierung, da es die Kommunikation zwischen den Mensch und das System leicht und ohne erweiterte Hinweise zulässt.

Drittens ist ein Web-Server für die Datenbank-Applikation schließlich zu implementieren. Er umfasst alle Datensätze über die vorhandenen im Labor Raspi-Boards, registrierten zum Kurs Studierenden und die abgewickelten Leihvorgänge. Web-Server wird mit einem Web-Framework Django erstellt. Django verfügt nun über die Funktionalität und Datenbasis, um die dafür erforderlichen Aktionen durchzuführen. Als Web-Framework bietet Django eine Reihe von Komponenten und Funktionen (Benutzerauthentifizierung, Hochladen von Dateien, Umgang mit Daten usw.), die bei jeder Webanwendungen benötigt werden. Mit einem Web-Framework muss ein Entwickler keine Zeit damit verschwenden, denselben Code von Grund auf neu jedes Mal zu schreiben.

Wie erfolgt nun die Abwicklung des eigentlichen Leihvorgangs von der Ausleihe bis zur Rückgabe eines Boards? Zuerst wird eine Studentenkarte am Register-Client ablesen und nachdem sollte einen Name und die Anzahl schon ausgeliehenen Boards am Display-Client angezeigt werden. Falls der Studierende zum Kurs zugelassen ist, darf dann ein gewünschten Board am Register-Client abgelesen werden. Es ist möglich, dass zu den schon ausgeliehenen Lab-Board noch zusätzlich einen Home-Board nach Hause mitgenommen wird. Das abgelesene Board ist entweder auszuleihen oder zurückzugeben. Sämtliche im Verleih befindlichen Geräte werden von den Mitarbeitern des Labors regelmäßig nach jeder Übung vor dem nächsten Ausleihevorgang auf Funktionsfähigkeit geprüft. Es kann sein, dass einem Studierenden die Home-Loan-Absicht eines Boards (12-16) verweigert wird, da in der Vergangenheit schon

einmal vom Studierende ein Board in einem inakzeptabel Zustand zurückgeben war und die Ursachen mit den Mitarbeiter des Labor nicht geklären hat. Falls der Studierende, dem Home-Loan verboten wurde, ein Board auszuleihen versucht, wird eine Fehlermeldung auf dem Bildschirm gezeigt und der Leihvorgang mit dem Fehlerzustand terminiert.



**Pic. 1.:** Das Raspi Loan-Board

Nachdem nun grundlegenden Funktionsweisen der Bestandteilen des Verteilten Systems geklärt sind, geht es zur Auswahl der Hardware. Der wichtigste Aspekt beim Hardwarekauf ist einerseits das vorhandene Budget des PSE-Labors und zum Anderen die Aufgaben, die zusammen spielenden Hardware erfüllen sollen. Zunächst müssen die 16 für die Ausleihe zur Verfügung stehenden Raspi Loan-Boards mit einem geeigneten RFID-Tag ausgestattet werden. Es ist selbstklebende NFC Tags zu verwenden, die klein und dünn sind und lassen sich auf der Rückseite des Schutzschirms zu befestigen und ein Aussicht und Funktionen des Geräts nicht zu beeinflussen.

Es ist nicht unerwähnt zu lassen, dass Tags nicht

ausgelesen werden können, wenn diese auf einen metallischen Gegenstand/Oberfläche geklebt wurden, da die Kommunikation aus physischen Gründen zwischen Tag und Lesegerät gestört werden kann. Die Raspi Loan-Boards sind mit einem Schutzschirm aus dem ein transparenter thermoplastischer Kunststoff geschützt und es wurde ins Labor getestet, dass die geklebte auf der Rückseite RFID-Tag lassen sich ablesen und den Board auf RFID Chip-Kartenleser zu platzieren. Erfolgreich getestet wurde RFID Leser/Schreiber ACR122U<sup>4</sup>, der wurde auf der Basis der 13,56 MHz kontaktlosen (RFID) Technologie entwickelt. Der ACR122U USB Kartenleser unterstützt nicht nur Mifare<sup>®</sup> Technologien, sondern auch alle vier Typen von NFC -Tags. Diese Anforderung ist für vorliegende Abschlussarbeit wichtig, da Studierenden-Ausweise der Beuth Hochschule mit dieser Technologie gelesen werden können. Desweiteren soll an dem oben genannten Register-Client der dort angeschlossene RFID Chip-Kartenleser über geeignete Software in Betrieb genommen und die eingelesenen Daten an den Server übermittelt werden. Dafür wird ein Raspberry Pi<sup>5</sup> verwendet, der zählt zu den beliebtesten Single Board Computern zählt und lässt die andere Schnittstellen anzubinden. Der kompakter Einplatinencomputer wenig verbraucht Strom aber dennoch genügend Power für ein Linuxbetriebssystem liefert. Detaillierte Beschreibungen zu jeder getroffenen Auswahl finden Sie im entsprechenden Abschnitt des Kapitels "Theoretischer Hintergrund

<sup>4</sup><https://www.conrad.de/de/p/rfid-reader-acr122u-nfc-usb-802554010.html>

<sup>5</sup><https://www.raspberrypi.org/>

## Der theoretische Hintergrund

Als schon im Abschnitt "Einleitung" erwähnt wurde, bei der Aufgabestellung es um eine Entwicklung einer Webanwendung geht. Die zu realisierende Software basiert sich, wie die meisten Webanwendungen, auf einer Client-Server-Architektur, wobei der Client Informationen eingibt, während der Server die eingegebene Daten empfängt, bearbeitet und speichert. Eine Webanwendung ist ein Computerprogramm, das eine bestimmte Funktion unter Verwendung eines Webbrowsers als Client ausführt. Die Webanwendung kann so einfach wie ein Kontaktformular auf einer Website oder so komplex wie eine Textverarbeitungs- oder Bildbearbeitungsprogramm sein, die Sie auf Ihr Computer im Browser ausführen können. Um die Webanwendung zu starten, muss der Benutzer keine zusätzlichen Programme installieren. Sie wird auf jedem Gerät mit Browser und Internetzugang ausgeführt. Der Client ist nicht vom Betriebssystem auf dem Computer des Benutzers abhängig. Bei der Entwicklung von Webanwendungen müssen daher keine separaten Versionen für Windows, Linux, Mac OS und andere Betriebssysteme geschrieben werden. Zur Implementierung der Clientseite werden HTML, CSS, JavaScript und Ajax verwendet. Zum Erstellen der Serverseite von Webanwendungen werden Programmiersprachen wie PHP, ASP, ASP.NET, Perl, C / C ++, Java, Python, Ruby und NodeJS verwendet. In dem vorherigen Schritt wurde es die Entscheidung getroffen, die Serverseite mit Web-Framework Django zu erstellen. Django wird mit einem leichten Webserver geliefert, mit dem eine Website schnell zum Laufen gebracht werden kann, ohne Zeit mit der Einrichtung eines Servers verschwenden zu müssen. Wenn der Entwicklungsserver von Django gestartet wird, überwacht er die Codeänderungen in Echtzeit. Es wird nach dem Ändern des Codes automatisch neu geladen.

Da In den letzten Jahren sich Webanwendungen rasant weiterentwickelt und die Desktop-Lösungen schrittweise ersetzt haben und sind zu einem wesentlichen Bestandteil des Geschäfts in der modernen Welt geworden haben, es sollte auch nicht unerwähnt bleiben, welche Vorteile die Webanwendungen haben.

- **Zugriff von jedem Gerät** Die Webanwendung kann überall auf der Welt von einem Computer, Tablet oder Smartphone aus verwendet werden, das mit dem Internet verbunden ist.

Sparen



Web-Apps können auf allen Plattformen ausgeführt werden und müssen nicht mehr separat für Android und iOS entwickelt werden.

#### Anpassungsfähigkeit

Wenn native Anwendungen bestimmte Betriebssysteme erfordern, eignen sich jedes Betriebssystem (Windows, MAC, Linux usw.) und jeder Browser (Internet Explorer, Opera, FireFox, Google Chrome usw.) für die Arbeit mit einer Webanwendung. ).

#### Mangel an Client-Software

Günstiger und einfacher zu installieren, zu warten und zu aktualisieren. Das Aktualisieren auf die neueste Version erfolgt beim nächsten Laden der Seite.

#### Netzwerksicherheit

Das Websystem verfügt über einen einzigen Einstiegspunkt, der zentral geschützt und konfiguriert werden kann.

#### Skalierbarkeit

Mit zunehmender Belastung des Systems ist es nicht erforderlich, die Kapazität der Client-Standorte zu erhöhen. Mit einer Webanwendung können Sie in der Regel nur mithilfe von Hardwareressourcen eine größere Datenmenge verarbeiten, ohne den Code neu schreiben und die Architektur ändern zu müssen.

#### Verhinderung von Datenverlust

Benutzerdaten werden in der "Cloud" gespeichert, für deren Integrität die Hosting-Anbieter verantwortlich sind, und sind vor Verlust geschützt, wenn die Festplatte des Computers beschädigt wird.

## 2.1 Über Raspberry Pi Board und OS

## 2.2 Kontaktlose Chipkartentechnik MIFARE

- 2.3 Sender-Empfänger-System mit RFID
- 2.4 Datenbanken mit Python und SQLite
- 2.5 HTTP für Design der verteilten Systeme
- 2.6 Django Framework
- 2.7 API
- 2.8 Endliche Zustandsmaschine
- 2.9 Clientseitiges JavaScript

# Systemdesign

## 3.1 Systemarchitektur

## 3.2 User Stories

## 3.3 UML

## 3.4 Endliche Zustandsmaschine

# Implementation

## 4.1 Register-Client

Der Standard legt das physikalische Layout der mikroBus-Pinout-Verbindung, die verwendeten Kommunikations- und Stromversorgungspins auf dem Mainboard fest.

## 4.2 Server

Die *Cloud* oder *Cloud Computing* Begriff kommt offensichtlich aus dem Englischen und heißt auf Deutsch "Wolke". Der Begriff beschreibt einen oder mehrere entfernte Server, auf die man seine Daten von einem Gerät über das Internet hochladen kann. Dann übernimmt die Cloud die Aufgaben wie die Datenverarbeitung oder komplizierte Programmabläufe. Während der Datenverarbeitung weißt der Nutzer nicht, wie viele Server hinter der Cloud stecken und welche komplizierte Hardware für die Berechnungen benötigt werden.

## 4.3 Display-Client

Die *Cloud* oder *Cloud Computing* Begriff kommt offensichtlich aus dem Englischen und heißt auf Deutsch "Wolke". Der Begriff beschreibt einen oder mehrere entfernte Server, auf die man seine Daten von einem Gerät über das Internet hochladen kann. Dann übernimmt die Cloud die Aufgaben wie die Datenverarbeitung oder komplizierte Programmabläufe. Während der Datenverarbeitung weißt der Nutzer nicht, wie viele Server hinter der Cloud stecken und welche komplizierte Hardware für die Berechnungen benötigt werden.

# Application

## 5.1 Usage in laboratory

Der Standard legt das physikalische Layout der mikroBus-Pinout-Verbindung, die verwendeten Kommunikations- und Stromversorgungspins auf dem Mainboard fest. Der Zweck von mikroBUS ist es, eine einfache Erweiterbarkeit der Hardware mit einer großen Anzahl von standardisierten kompakten Zusatzboards zu ermöglichen, von denen jede einen einzelnen Sensor, Display, Encoder oder Motortreiber, eine integrierte Schaltung hat. Der von MikroElektronika entwickelte mikroBUS ist ein offener Standard - jeder kann mikroBUS in seinem Hardwaredesign implementieren. Die Abbildung<sup>1</sup> ?? zeigt die Pinout Spezifikation des Herstellers, die man entsprechend ändern kann und die neue Verbindungen für den eigenen Projekt feststellen. Wenn ein Modul eine Schnittstelle verwendet, die bereits auf mikroBUS vorhanden ist, benutzt man diese exakten Pins und markiert diese entsprechend. Wenn ein Pin nicht verwendet wird, sollte er als NC (für "Not Connected") markiert sein.

---

<sup>1</sup><https://download.mikroe.com/documents/standards/mikrobus/mikrobus-standard-specification-v200.pdf>

## Results and conclusion

### 6.1 Results

Der Standard legt das physikalische Layout der mikroBus-Pinout-Verbindung, die verwendeten Kommunikations- und Stromversorgungspins auf dem Mainboard fest. Der Zweck von mikroBUS ist es, eine einfache Erweiterbarkeit der Hardware mit einer großen Anzahl von standardisierten kompakten Zusatzboards zu ermöglichen, von denen jede einen einzelnen Sensor, Display, Encoder oder Motortreiber, eine integrierte Schaltung hat.

### 6.2 Conclusion

Der Standard legt das physikalische Layout der mikroBus-Pinout-Verbindung, die verwendeten Kommunikations- und Stromversorgungspins auf dem Mainboard fest. Der Zweck von mikroBUS ist es, eine einfache Erweiterbarkeit der Hardware mit einer großen Anzahl von standardisierten kompakten Zusatzboards zu ermöglichen, von denen jede einen einzelnen Sensor, Display, Encoder oder Motortreiber, eine integrierte Schaltung hat.

# Appendix

## A.1 RFID

RFID (englisch radio-frequency identification oder „Identifizierung mit Hilfe elektromagnetischer Wellen“) bezeichnet eine Technologie für Sender-Empfänger-Systeme und wird bei der Abschlussarbeit verwendet, um die vorhandenen zur Ausleihe Raspberry Pi Boards zu markieren und identifizieren.

## Abkürzungsverzeichnis

**Raspi**    Raspberry Pi Board und Minicomputer

**RFID**    Radio-frequency identification



# Abbildungsverzeichnis

C

Abb. 1	Das Raspi Loan-Board . . . . .	4
--------	--------------------------------	---