



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

<b>B</b>	<b>List of abbreviations</b>	<b>10</b>
<b>C</b>	<b>List of images</b>	<b>11</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 Studiengänge Technische Informatik<sup>2</sup> und Medieninformatik ist. 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 von Mitarbeitern des Labors bereits festgestellt und regelmäßig verlangsamten 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 verliehen 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 Studententname, Matrikelnum-

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

<sup>2</sup><https://www.beuth-hochschule.de/b-ti>

mer, 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 schreiben 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-Klient 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-Klient 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 Alternative für die technische Realisierung, da es die Kommunikation zwischen dem Mensch und dem 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 Webanwendung 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-Klient ablesen und nachdem sollte ein Name und die Anzahl schon ausgeliehenen Boards am Display-Klient angezeigt werden. Falls der Studierende zum Kurs zugelassen ist, darf dann ein gewünschtes Board am Register-Klient abgelesen werden. Es ist möglich, dass zu dem schon ausgeliehenen Lab-Board noch zusätzlich ein 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 Ausleihvorgang 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 Studierenden ein Board in einem inakzeptablen Zustand zurückgegeben war und die Ursachen mit dem Mitarbeiter des Labors nicht geklärt 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. Die entsprechenden Fehlerzustände sind für andere vorgesehen.

## Theoretische Hintergrund

- 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 mikroBUS

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

## List of abbreviations

<b>BOM</b>	Bill of Materials
<b>DEVKIT</b>	Development Kit
<b>Cloud</b>	Cloud Computing
<b>EKG</b>	Elektrokardiogramm
<b>GmbH</b>	Gesellschaft mit beschränkter Haftung
<b>I<sup>2</sup>S</b>	Inter-IC Sound
<b>IoT</b>	Internet of Things, Internet der Dinge
<b>IT</b>	Informationstechnik, Bereich der Informations- und Datenverarbeitung
<b>JSON</b>	JavaScript Object Notation
<b>LED</b>	Light-emitting diode, Leuchtdiode
<b>MQTT</b>	Message Queuing Telemetry Transport
<b>NVS</b>	Non-volatile storage
<b>PCB</b>	Printed circuit board
<b>PWM</b>	Pulse Width Modulation
<b>SPI</b>	Serial Peripheral Interface Bus
<b>UV</b>	Ultraviolettstrahlung

List of images

C