

NFC ZEITERFASSUNGSSYSTEM

Bachelorarbeit der Hochschule Luzern

- Departement Informatik



Eingereicht von:
Philipp Bütler
Studiengang: BSc Informatik
Rotkreuz, 07.06.2019

Bachelorarbeit an der Hochschule Luzern – Informatik

Titel: NFC Zeiterfassungssystem

Studentin / Student 1: Philipp Bütler

Studiengang: BSc Informatik

Abschlussjahr: 2019

Betreuungsperson: Roger Diehl

Expertin / Experte: Marc Wiest

Codierung / Klassifizierung der Arbeit:

- ☒ **A: Einsicht (Normalfall)**
- ☐ **B: Rücksprache** (Dauer: Jahr / Jahre)
- ☐ **C: Sperre** (Dauer: Jahr / Jahre)

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich/wir die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt haben, alle verwendeten Quellen, Literatur und andere Hilfsmittel angegeben haben, wörtlich oder inhaltlich entnommene Stellen als solche kenntlich gemacht haben, das Vertraulichkeitsinteresse des Auftraggebers wahren und die Urheberrechtsbestimmungen der Fachhochschule Zentralschweiz (siehe Markblatt «Studentische Arbeiten» auf MyCampus) respektieren werden.

Ort / Datum, Unterschrift _____

Abgabe der Arbeit auf der Portfolio Datenbank:

Bestätigungsvisum Student/in

Ich bestätige, dass ich die Bachelorarbeit korrekt gemäss Merkblatt auf der Portfolio Datenbank abgelegt habe. Die Verantwortlichkeit sowie die Berechtigungen habe ich abgegeben, so dass ich keine Änderungen mehr vornehmen kann oder weitere Dateien hochladen kann.

Ort / Datum, Unterschrift _____

Verdankung

Ich möchte mich bei all jenen Bedanken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Bachelorarbeit beigetragen haben. Besonderen Dank geht an meinen Betreuer Roger Diehl, der mir während der ganzen Arbeit in konstruktiven Meetings und stets hilfreichen Inputs unterstützend zur Seite stand.

Eingangsvisum (durch das Sekretariat auszufüllen):

Rotkreuz, den _____ Visum: _____

Hinweis: Die Bachelorarbeit wurde von keinem Dozierenden nachbearbeitet. Veröffentlichungen (auch auszugsweise) sind ohne das Einverständnis der Studiengangleitung der Hochschule Luzern – Informatik nicht erlaubt.

Copyright © 2019 Hochschule Luzern – Informatik

Alle Rechte vorbehalten. Kein Teil dieser Arbeit darf ohne die schriftliche Genehmigung der Studiengangleitung der Hochschule Luzern – Informatik in irgendeiner Form reproduziert oder in eine von Maschinen verwendete Sprache übertragen werden.

Abstract

Das Arbeitsgesetz verpflichtet die Arbeitgeber zum Erfassen von Arbeitszeiten. Namentlich müssen Dauer, Beginn und Ende der geleisteten täglichen und wöchentlichen Arbeitszeit ersichtlich sein. Die vorliegende Bachelor Arbeit beschreibt ein auf NFC basierendes Zeiterfassungssystem um Arbeitszeiten zu erfassen. Dazu wurde fast der gesamte Produktentwicklungsprozess durchlaufen. Begonnen bei der Evaluation und Beschaffung der benötigten Hardware bis hin zu Tests in einem realen Umfeld. Für die Umsetzung der Arbeit wurde das iterativ-inkrementelle Projektmodell SoDa¹ eingesetzt. Die Hardwareevaluation zeigte, dass sich das Raspberry Pi 3 und das EXPLORE-NFC-WW Erweiterungsboard am besten für dieses Projekt eignen. Damit dem Benutzer eine Rückmeldung geliefert werden kann, wurde die Hardware mit dem Raspberry Pi 7'' Touchscreen Display ergänzt.

Das finale Produkt kann in drei Hauptkomponenten aufgeteilt werden. Dem Hardware System, welches für das Einlesen der NFC Tags und dem Feedback für den Benutzer verantwortlich ist. Dem Web UI für das Anzeigen der Erfassten Zeiten und dem Backend API, welches dem System die Funktionalität zur Verfügung stellt. Das Hardware System besteht aus der Hardware und der dazugehörigen Software. Die Software wurde mit der Programmiersprache Python geschrieben. Das Backend API sowie das Web UI wurden mit ASP.NET Core 2.0 realisiert.

Am Ende der Arbeit steht ein stabiles und getestetes Produkt, welches in einem ersten Schritt im Entwicklungsteam der Cure-IT AG eingesetzt wird. Mit dem Ziel das Zeiterfassungssystem zu einem späteren Zeitpunkt auch externen Kunden anzubieten.

¹ <https://www.hslu.ch/de-ch/informatik/studium/soda/>

Inhaltsverzeichnis

1	Projektübersicht	6
1.1	Ausgangslage.....	6
1.2	Ziel der Arbeit	7
2	Stand der Technik / Near Field Communication.....	8
3	Komponentenevaluation	9
3.1	NFC Lesegerät.....	9
3.2	NFC Lesegeräte	9
3.3	Fazit	10
4	Methoden	11
4.1	Projektplanung Soda.....	11
5	Realisierung	12
5.1	Hardware System.....	13
5.2	Backend API.....	14
5.3	Webfrontend	16
6	Evaluation und Validation	17
7	Ausblick	18
7.1	Weiteres Vorgehen	19
8	Abbildungsverzeichnis.....	20
9	Tabellenverzeichnis	20
10	Literaturverzeichnis	20
11	Anhang.....	21

1 Projektübersicht

1.1 Ausgangslage

Das Arbeitsgesetz verpflichtet die Arbeitgeber zum Erfassen von Arbeitszeiten. Namentlich müssen Dauer, Beginn und Ende der geleisteten täglichen und wöchentlichen Arbeitszeit ersichtlich sein. Typische Instrumente für das Erfassen sind beispielsweise Batch-Stationen oder manuell erfasste Timesheets. Eine sehr präzise Erfassung ist vor allem für Freelancer oder externe Angestellte wichtig, die auf Stundenbasis arbeiten. Dabei ziehen die bereits bestehenden Systeme einige Probleme mit sich:

Menschliche Fehler

Angestellte erfassen unvollständige oder falsche Daten. Dies führt für das HR zu zusätzlicher Arbeit, da diese Fehler wieder korrigiert werden müssen.

Mutmassliche Falscheingaben

Angestellte erfassen absichtlich falsche Zeiten oder stempeln für ihre Kollegen, um mehr Arbeitszeit auszuweisen.

Zeitintensiver Prozess

Das manuelle Erfassen von Arbeitszeit ist zeitaufwändig. Das kann dazu führen, dass das Erfassen der Zeiten zu einem späteren Zeitpunkt erfolgt, z.B. am Ende der Woche. Die Folgen davon sind ungenaue Zeiterfassungen und Fehler.

Aktuell existiert mit dem TimeTrackingTool ein Prototyp für das Erfassen und Verwalten von Arbeitszeiten. Der Prototyp ist eine verteilte Anwendung und besteht aus einem WebAPI, einer Cross Plattform Xamarin App und einer sehr rudimentär gestalteten Webpage. Die Benutzerverwaltung wird über ein Cloud Backend in Microsoft Azure realisiert. Das Erfassen der Zeiten ist im Prototyp sowohl über die mobile App sowie über die Webseite möglich.

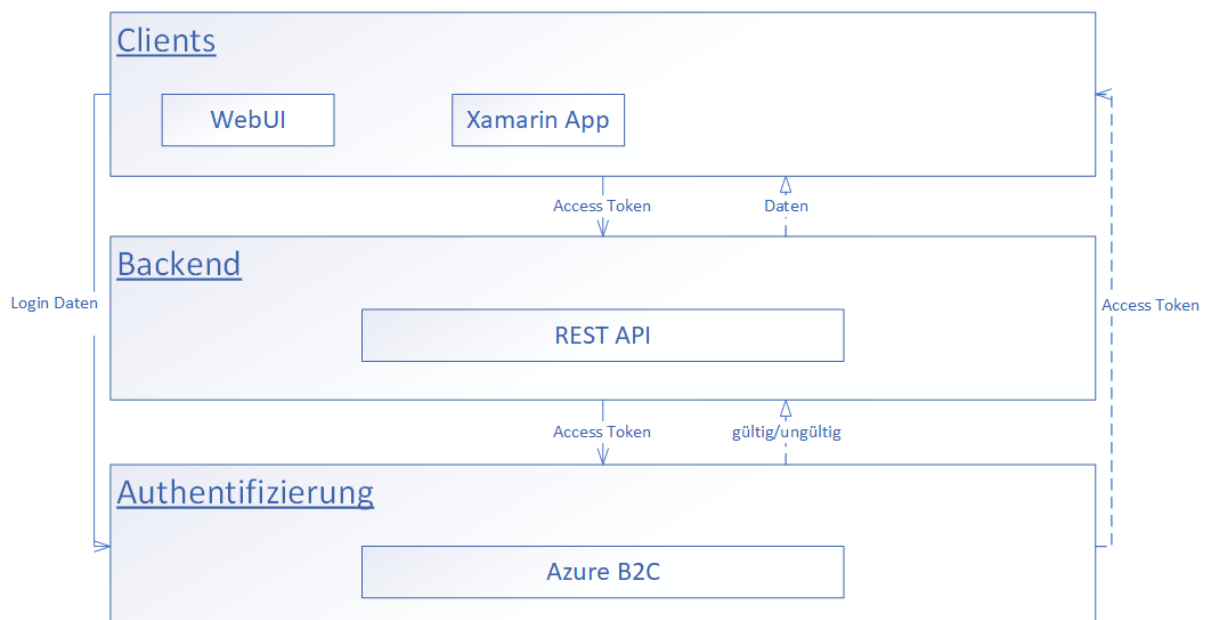


Abbildung 1: Aufbau des vorhandenen Prototyps

Clients

Endbenutzer nutzen das System über unterschiedliche Clients. Die Applikation ist so konzipiert, dass das Backend vom Frontend getrennt ist um eine beliebige Anzahl unterschiedlicher Clients für unterschiedliche Anwendungsfälle zu unterstützen. Aktuell existiert eine dedizierte App für das reine Erfassen der Zeiten und

ein zusätzliches Web UI für das Verwalten der Daten oder die Kontrolle der monatlichen Zeiterfassungen. Jede Logik im Zusammenhang mit der Generierung von Zeitaufzeichnungen (einschliesslich der Mitarbeiterauthentifizierung), wird Client-Seitig durchgeführt, bevor die erfasste Zeit an das Backend geliefert wird. Das Web UI ist mit ASP.NET Core 2.0² umgesetzt.

Backend

Das Backend der Applikation stellt den Clients über eine gemeinsame API die Funktionalität des Systems zur Verfügung. Die Menge der Funktionen, die der Client einem Benutzer zu Verfügung stellt, hängt von der Client-Implementierung ab. Der Zugriff auf die Funktionen wird jedoch durch rollenbasierte Authentifizierung über die User Session eingeschränkt. Das Backend API ist mit ASP.NET Core 2.0 als RESTful implementiert und folgt dem Model-View-Controller (MVC)- Pattern.

Authentifizierung

Um Modularität zu erreichen, ist die Applikation auf die Verwendung eines externen Authentifizierungsdienstes ausgelegt, auf den Clients und Backend unabhängig voneinander zugreifen.

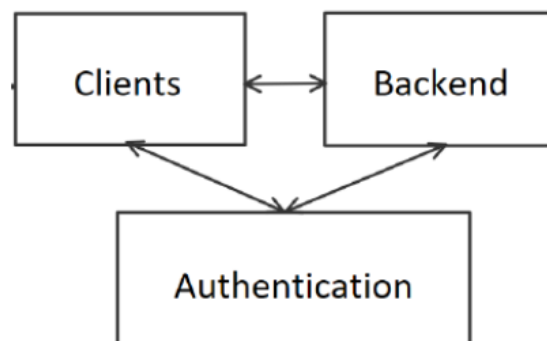


Abbildung 2: Hauptkomponenten der Applikation mit deren Beziehungen

1.2 Ziel der Arbeit

Ziel des Projektes ist es, den bestehenden Prototypen durch einen weiteren Client zu erweitern. Der neue Client soll eine auf NFC basierende Batch-Möglichkeit anbieten, um einen weiteren Weg für die Erfassung der Zeiten zu ermöglichen. Im Projekt werden folgende Punkte erwartet:

- Auswahl und Beschaffung von geeigneten NFC Komponenten sowohl für den Empfänger, sowie für den Sender.
- Zusammenbau und Konfiguration der Hardware.
- Entwickeln der Software, um Zeiten per NFC zu erfassen.
- Erweiterung des Web APIs für die Registrierung der NFC Erfassungen.
- Erweiterung der Webpage für die Darstellungen der NFC Erfassungen.

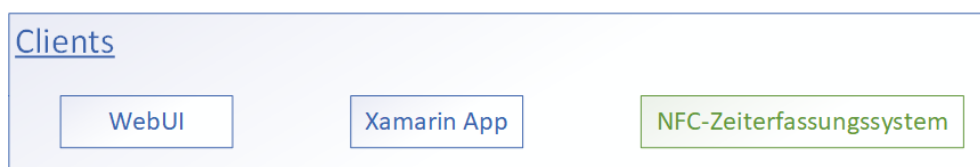


Abbildung 3: Clients inklusive dem NFC Zeiterfassungssystem

² <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>

2 Stand der Technik / Near Field Communication

Wie bereits in den vorherigen Kapiteln erläutert, ist das Ziel des Endsystems Zeiten mittels NFC zu Erfassen. Die Abkürzung NFC steht für Near Field Communication und bezeichnet einen auf der RFID-Technik basierenden internationalen Übertragungsstandard. Wie der Begriff «Nahfeldkommunikation» bereits andeutet, wird dieser Standard für das Übertragen von Daten über kurze Strecken verwendet. Die Besonderheit von NFC liegt darin, dass beide Geräte in einen Abstand von wenigen Zentimetern gehalten werden müssen, damit eine Übertragung stattfinden kann. Das hat den Vorteil, dass es sicher gegen das Abhören von aussen ist, denn mögliche Täter müssen sich bis auf engsten Raum nähern, um Daten auszuspionieren.

NFC eignet sich für einige Anwendungsbereiche. Am naheliegendsten ist die Nutzung für den Datenaustausch beispielsweise zwischen zwei Smartphones. Durch aneinanderhalten von NFC-fähigen Geräten können mit einem Klick Daten, wie Kontakte oder Links, übertragen werden. Ein stark wachsender Anwendungsbereich ist das bargeldlose Bezahlen. Besitzt man eine NFC-fähige Kreditkarte reicht ein kurzes Annähern oder Berühren des Terminals um den Geldbetrag zu begleichen. Weitere Anwendungen sind Papierlose Eintrittskarten (E-Tickets), Smart Posters in der Aussenwerbung oder für die Zugangskontrolle. (Near Field Communication, 2019)



Abbildung 4: NFC-fähige Kreditkarte (Brinkmann, 2019)

RFID (Radio-Frequency Identification) oder auf Deutsch «Identifizierung mit Hilfe elektromagnetischer Wellen» bezeichnet eine Technologie für Sender-Empfänger-Systeme zum automatischen und berührungslosen Identifizieren und Lokalisieren von Objekten mit Radiowellen. Ein RFID-System besteht aus einem Transponder mit einem kennzeichnenden Code, der sich am oder im Gegenstand befindet sowie einem Lesegerät zum Auslesen dieser Kennung. (RFID, 2019)

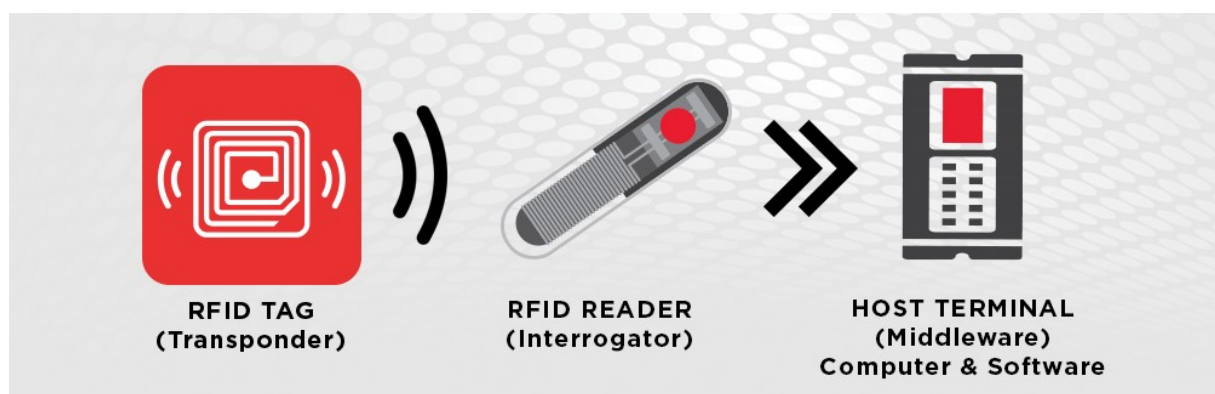


Abbildung 5: RFID Sender-Empfänger-System (Access event solutions, 2019)

Für das Zeiterfassungssystem wird also ein RFID Tag (Transponder) und ein Lesegerät benötigt. Der Transponder muss mit einer Mitarbeiter ID ausgestattet werden, damit der Benutzer vom System eindeutig identifiziert werden kann, nachdem die ID vom Lesegerät ausgelesen und an das System weitergeleitet wurde.

3 Komponentenevaluation

3.1 NFC Lesegerät

Auf dem Markt gibt es eine grosse Diversität von NFC Lesegeräten. Erhältlich ist vieles, von einfachen RFID-Modulen für Single-Board Computer oder Mikrocontroller bis hin zu kompletten Plug-and-Play-USB-NFC-Geräten. Allerdings ist das reine Erkennen der NFC-Tags für die Applikation nicht ausreichend. Nach dem Erkennen sollen die gewonnenen Informationen an das Backend API weitergeleitet werden, um Zeiten zu erfassen. Dies ist nur mit einem Computerprogramm möglich. Dieses Programm kann auf einem Single-Board Computer ausgeführt werden. Ein SBC ist ein Computersystem, bei dem sämtliche zum Betrieb nötigen elektronischen Komponenten auf einer einzigen Leiterplatte zusammengefasst sind (Einplatinencomputer, 2019). Das hat den Vorteil, dass der Computer sehr klein ist und sich gut in einem Gehäuse verbauen lässt. Zusätzlich kostet er durch seine Simplifizierung viel weniger als ein herkömmlicher Computer. Im Rahmen dieser Arbeit wurden verschiedene Systeme untersucht, um abzuwägen, welches sich am besten eignet. Die nachfolgende Tabelle zeigt eine Zusammenfassung dieser Untersuchung. Der vollständige Evaluationsbericht ist im Anhang A: Evaluationsdokument zu finden.

Tabelle 1: Übersicht der untersuchten SingleBoard Computer

Hardware Komponente	Vorteile	Nachteile
Single-Board Computer		
Raspberry Pi 3 Model B+	- Grosse, aktive Community - Allrounder	
ASUS Tinker Board S	- Schnelle CPU - Viel RAM - Viel Speicher	Preis
Raspberry Pi Zero W	- Grösse - Preis	Adapter notwendig um Hardware anzuschliessen
Banana Pi M64	Viel Speicher	Preis-Leistungs-Verhältnis

3.2 NFC Lesegeräte

Alle im vorherigen Kapitel beschriebenen SBC's besitzen eine GPIO Schnittstelle. GPIO heisst ausgeschrieben General-Purpose Input/Output und beschreibt eine Schnittstelle, welche für Ein- und Ausgaben genutzt werden kann. Über diese GPIO Schnittstelle können beispielsweise RFID Module mit dem Board zusammengefügt werden. Die nachfolgende Tabelle zeigt einen Auszug aus dem Evaluationsdokument (Anhang: A) und zeigt eine Zusammenstellung der untersuchten NFC Lesegeräte.

Tabelle 2: Übersicht der untersuchten NFC Lesegeräte

Hardware Komponente	Vorteile	Nachteile
NFC Lesegeräte		
Mifare RC522 RFID Modul	• Preis • Grösse	• Pin Leiste muss angelötet werden • Die GPIO Schnittstelle muss selbst richtig verbunden werden
EXPLORE-NFC-WW	• Plug and Play (kein Löten, keine Pins verbinden) • Grösse	• Preis • Benötigt ganze GPIO Schnittstelle
CHERRY TC 1100		• Grösse

3.3 Fazit

Die Untersuchung hat ergeben, dass die geeignetsten Komponenten für dieses Projekt das **Raspberry Pi 3 Model B+** in Kombination mit dem **EXPLORE-NFC-WW** Erweiterungsboard sind. Das Raspberry Pi zeichnet sich hauptsächlich durch sein gutes Preis-Leistung-Verhältnis und durch die grosse Community aus. Die Applikation auf dem SBC ist mit dem Lesen des NFC Chips und dem Aufrufen des APIs nicht sonderlich komplex. Somit ist die Hardware des Raspberry Pis ausreichend und es muss nicht auf das Leistungsstärkere Tinker Board ausgewichen werden. Die Leistung des Pi Zero W wäre auch ausreichend. Allerdings wird die Arbeit mit dem Zero erschwert, da für das Anschliessen von Monitoren und Tastaturen Adapter benötigt werden. Ausserdem ist es nicht notwendig, dass der SBC möglichst klein ausfällt.

Das EXPLORE-NFC-WW Erweiterungsboard für das Raspberry Pi zeichnet sich hauptsächlich durch die gute Synergie mit dem Raspberry Pi aus. Es kann ohne Vorarbeit bequem auf die GPIO Schnittstelle gesteckt werden. Das Erweiterungsboard ist zwar mehr als doppelt so teuer wie das Mifare RFID Modul, ist aber mit 24 Franken trotzdem gut erschwinglich. Die Grösse ist optimal an das Raspberry Pi angepasst und um einiges eleganter als der Smartcard Reader CHERRY TC 1100, welcher für den Anwendungsfall viel zu gross ist.

Damit die NFC Technologie funktioniert wird nicht nur ein NFC Lesegerät benötigt, sondern auch ein NFC-Tag (Transponder). Beim Kauf des NFC Erweiterungsboards wird standardmässig ein NFC-Tag mitgeliefert. Die NFC Tags sind nicht für ein Lesegerät spezifisch, sondern entsprechen den NFC Standards und sind mit allen Lesegeräten kompatibel. Aus diesen Gründen wird auf eine Empfehlung für die Tags verzichtet.

Damit das Zeiterfassungssystem in der Praxis eingesetzt werden kann, ist es zwingend notwendig, dass das System dem Benutzer ein Feedback geben kann. Dies beinhaltet unter anderem eine Begrüssung des Benutzers nach einer erfolgreichen Zeiterfassung oder das Anzeigen einer Fehlermeldung, falls die Erfassung nicht erfolgreich war. Dies ist mit den bereits genannten Komponenten nicht möglich. Aus diesem Grund wird die bisherige Hardware mit dem Raspberry Pi 7'' Touchscreen Display ergänzt. Bei dem Display handelt es sich um den offiziellen LCD-Touchscreen der Raspberry Pi Foundation, der somit perfekt für die Verwendung mit dem Raspberry Pi ausgelegt ist.

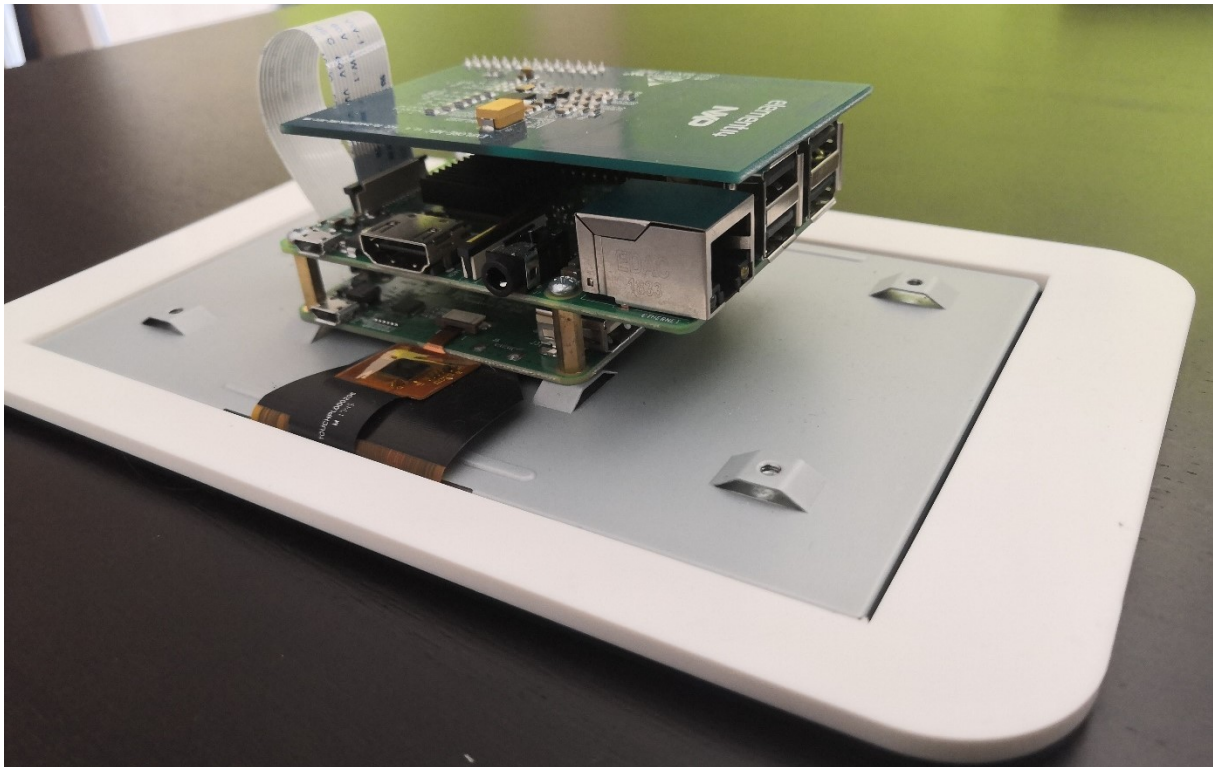


Abbildung 6: Hardware des Zeiterfassungssystems

4 Methoden

4.1 Projektplanung Soda

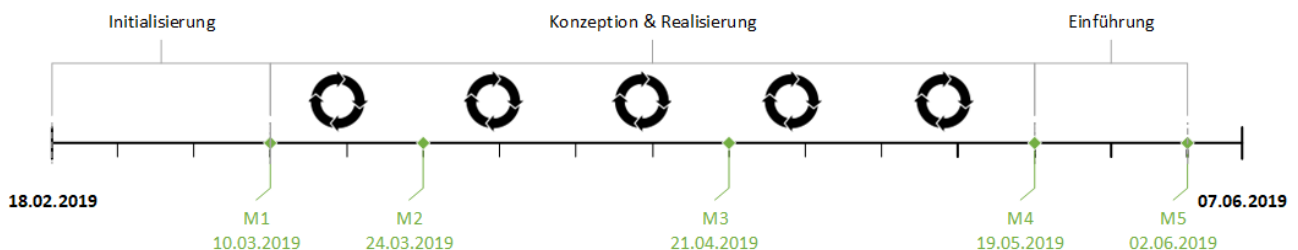


Abbildung 7: Projektplan für das NFC Zeiterfassungssystem

Das Projekt wurde nach dem Projektmodell SoDa³ (Software Development Agile) umgesetzt. Wie für dieses Projektmodell typisch finden in der Konzeption -und Realisierungsphase Sprints in der Länge von zwei Wochen statt. Das hat den Vorteil, dass nicht bereits in der Initialisierungsphase alle Anforderungen vollständig erfasst werden müssen. Sowohl die Anforderungen sowie das Entwickeln der Software finden in einem iterativ-inkrementellen Prozess statt. Damit der Projektfortschritt messbar ist, sah die Projektplanung 5 Meilensteine vor. Die nachfolgende Tabelle listet alle Meilensteine auf. Die ausführlichen Meilensteinbeschreibungen sind im Projektmanagementplan im Anhang B: Projektmanagementplan ersichtliche.

³ <https://www.hslu.ch/de-ch/informatik/studium/soda/>

Tabelle 3: Übersicht der Meilensteine

Meilenstein	Datum	Titel
MS 1	10.03.2019	Abschluss der Initialisierungsphase
MS 2	24.03.2019	Auswahl der NFC Komponenten vollzogen und bestellt
MS 3	21.04.2019	Zusammenbau und Konfiguration der Hardware
MS 4	19.05.2019	Erweiterung des bestehenden Prototyps (API & Webproject)
MS 5	02.06.2019	Version 1.0

5 Realisierung

Das Zeiterfassungssystem besteht aus drei Hauptkomponenten: Das hardwarenahe Programm für das Einlesen der NFC Tags, dem Web UI für das Anzeigen der erfassten Zeiten und dem Backend API, welches dem System die Funktionalität zur Verfügung stellt. Abbildung 8 zeigt ein Sequenzdiagramm, welches das Zusammenspiel und die Beziehungen der einzelnen Komponenten beschreibt. Für eine bessere Lesbarkeit wurde das Diagramm vereinfacht. Auch auf die Darstellung der Zugriffe auf das Azure B2C Authentifizierungssystem wird verzichtet. Das Sequenzdiagramm illustriert mit der Zeiterfassung und dem anschliessenden Anzeigen der Zeiten die zwei Hauptworkflows des Systems. Der Workflow für das Erfassen der Zeiten wird durch den Benutzer ausgelöst, indem er einen NFC Tag mit seiner ID an das Lesegerät hält. Damit er die erfassten Zeiten ansehen kann, muss er sich über ein Login im Web UI anmelden.

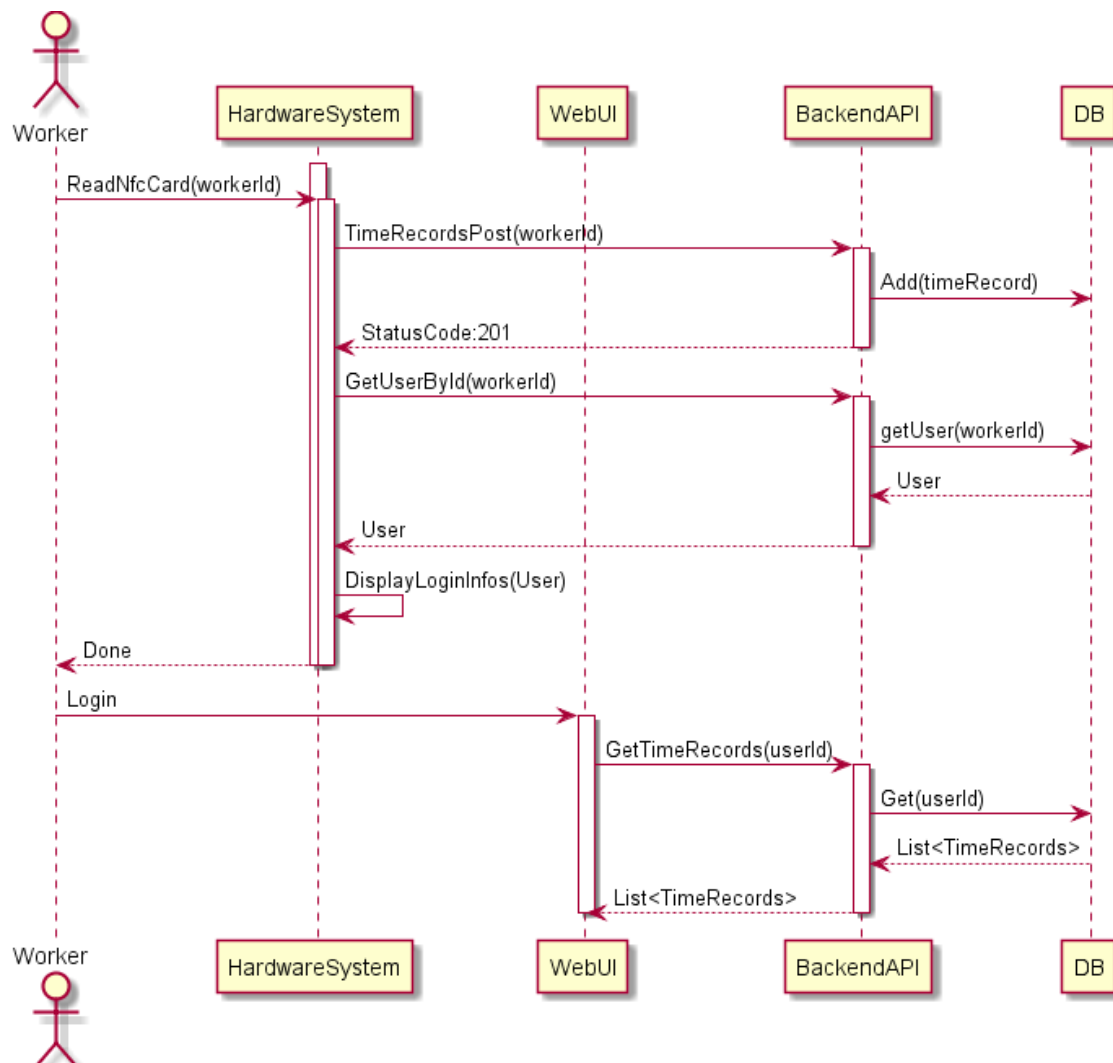


Abbildung 8: Sequenzdiagramm: Zusammenspiel der Systemkomponenten

Zeiterfassungen werden nicht direkt auf einen Benutzer, sondern auf ein Projekt des Benutzers gespeichert. Das ergibt Sinn, da es nicht unüblich ist, dass eine Person gleichzeitig an mehreren Projekten arbeitet. Damit der Benutzer über das Zeiterfassungssystem auf ein Projekt buchen kann, muss dem Zeiterfassungsgerät ein Projekt zugeordnet werden. Dies wird erreicht, indem jedem Gerät einen eigenen Benutzer zugeordnet wird. Die Zuweisung über einen Benutzer anstelle einer direkten Zuweisung bietet entscheidende Vorteile. Änderungen eines Projektes können ohne Zugriff auf das Gerät gemacht werden. Wird dem spezifischen Gerätebenutzer ein neues Projekt zugeteilt, ändert sich automatisch auch das Projekt des Geräts. Ein weiterer Vorteil ist, dass die Authentifizierung auf dem Backend API über den Station User gemacht werden kann. Diese Authentifizierung ist notwendig, da aus Sicherheitsgründen alle Methoden auf dem API durch rollenbasierte Zugriffskontrolle geschützt sind. Unter rollenbasierter Zugriffskontrolle versteht man das Zuweisen von Benutzerrollen um die Zugriffssteuerung zu gewährleisten. Hierbei besitzt jede Benutzerrolle unterschiedliche Rechte und somit unterschiedliche Zugriffe auf die Backend API Methoden. Die alternative Möglichkeit, den Benutzern direkt Rechte und Zugriffe auf die Methoden zu geben, hat den Nachteil, dass bei zunehmender Benutzerzahl die Rechteverwaltung unübersichtlich und somit fehleranfällig wird. Für das Zeiterfassungssystem wurde eine separate Rolle «NFC Station» erstellt.

Um ein neues Zeiterfassungsgerät in Betrieb nehmen müssen folgende Schritte ausgeführt werden:

1. Falls das Projekt noch nicht besteht, neues Projekt erstellen.
2. Neuen Station Benutzer mit Rolle «NFC Station» erstellen und dem Projekt zuweisen.
3. Mit dem Station Benutzer auf dem Zeiterfassungsgerät einloggen.

5.1 Hardware System

Das Hardware System besteht aus der in Kapitel 3 beschriebenen Hardware und der dazugehörigen Software. Die Software wurde mit der Programmiersprache Python geschrieben. Die Scriptsprache zeichnet sich durch die gute Integration auf dem Raspberry Pi aus. Die Sprache selbst wie auch Python IDEs sind standardmässig auf dem Raspberry Pi installiert. Ausserdem steht mit der «nxppy» Library⁴ ein Python Wrapper für die Interaktion mit dem EXPLORE-NFC Board zur Verfügung, was das Lesen der NFC Tags erheblich vereinfacht. Das Python Programm besteht aus drei Komponenten. Diese sind im nachfolgenden Diagramm ersichtlich.

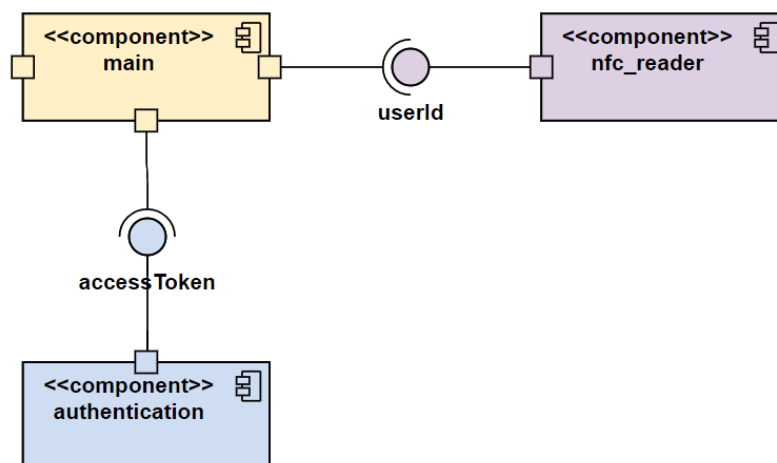


Abbildung 9: Komponentendiagramm der Architektur des Hardware Systems

nfc_reader

Der NFC-Reader ist für das Lesen der NFC Tags verantwortlich. Nachdem ein Tag erkannt wurde wird

⁴ <https://github.com/svvitale/nxppy>

dessen Inhalt ausgelesen und geprüft, ob sich auf dem Tag eine valide UUID befindet. Ist das zutreffend wird die UUID der Main-Komponente zur Verfügung gestellt.

authentication

Die Authentication-Komponente ist für die Authentifizierung auf dem Backend API verantwortlich. Dies ist notwendig, da alle Methoden auf dem API zugriffsgeschützt sind. Beim ersten Benutzen des Hardware Systems muss sich der Station User auf dem Gerät anmelden. Das kann sowohl automatisch durch hinterlegen von Benutzernamen und Passwort oder manuell durch einloggen auf Azure B2C stattfinden. Nach dem einloggen wird in der Software ein Zugriffstoken hinterlegt, welches bei jedem Aufruf auf das API mitgeschickt wird. Dieses Token hat eine Gültigkeit von einer Stunde. Damit man sich nicht stündlich neu anmelden muss, erhält ein Client zusätzlich zum Zugriffstoken ein Aktualisierungstoken. Mit dem Aktualisierungstoken kann nach Ablauf des Zugriffstokens ein Neues erworben werden, ohne dass man sich neu anmelden muss. Die Gültigkeit des Aktualisierungstokens gilt bis auf Widerruf. (Configurable token lifetimes in Azure Active Directory, 2019)

main

Die Main-Komponente ist für den Output auf der Konsole, sowie für das Aufrufen der API Methoden verantwortlich. Auf der Konsole wird eine Willkommensmeldung angezeigt mit der Aufforderung, den NFC Tag an das Lesegerät zu halten. Im Anschluss wird der Benutzer mit Namen und Vornamen begrüsst und der Zeitstempel wird angegeben. Schlägt die Zeiterfassung fehl, beispielsweise durch fehlen der UUID auf dem NFC Tag, wird der Benutzer entsprechend benachrichtigt. Damit dieses Verhalten möglich ist werden von dieser Komponente jeweils zwei API Aufrufe gemacht. Zum einen für das Erfassen der Zeit und zum andern für das Anzeigen des Benutzernamens.

```

888      888      888
888  o  888      888
888 d8b 888      888
888 d888b 888 .d88b. 888 .d8888b .d88b. 88888b.d88b. .d88b.
888d88888b888 d8P Y8b 888 d88P" d88""88b 888 "888 "88b d8P Y8b
88888P Y88888 888888888 888 888 888 888 888 888 88888888
8888P Y8888 Y8b. 888 Y88b. Y88..88P 888 888 888 Y8b.
888P Y888 "Y8888 888 "Y8888P "Y88P" 888 888 888 "Y8888

-----
Please enter your card:
Welcome Bachelor Test User!

You logged in at 05.06.2019 15:07:52

Have a nice day! :)

```

Abbildung 10: Konsolenausgabe nach erfolgreichem Einloggen

5.2 Backend API

Das Backend API soll Client und Server entkoppeln und den Clients eine einheitliche Schnittstelle bereitstellen, über welche sie Zugang zu Datenzugriffs- und Manipulationsfunktionen haben. Deshalb wurde das Backend API als RESTful implementiert. Die Implementation des Backends wurde mit ASP.NET Core 2.0 realisiert. Der Vorteil von .NET Core liegt in der vertrauten Codebasis, der grossen Entwickler-Community, der Erweiterbarkeit durch Plugins von Drittanbietern und dem einfachen Deployment.

Software Pattern

Die Implementierung der Backend API folgt dem Model-View-Controller (MVC)- Pattern. Hierbei werden die Models durch Datenbank Entities dargestellt. Die Views sind JSON- Darstellungen von Data Transfer Objects (DTO). Controller implementieren die Logik und sind das Bindeglied zwischen den Models und den Views. ASP.NET Core bietet standardmässig eine sehr gute MVC-Integration unter der Beachtung von Namenskonventionen, was die Entwicklung stark vereinfacht. Für die Datenbank wird Microsoft SQL Server mit dem Entity Framework (EF) als Objekt-Relationales Mapping (ORM) Framework verwendet.

NFC Zeiten erfassen

Damit über den POST API Endpoint «/api/TimeRecord/station» Zeiten erfasst werden können, müssen einige Vorgaben erfüllt werden:

Das Aufrufen der Methode ist nur mit der Rolle «NFC Station» erlaubt. Diese Einschränkung ist wichtig, da über diese Methode das Erfassen von Zeiten für einen anderen Benutzer gemacht werden kann. Für den Station User ist das notwendig, aber im Normalfall soll das natürlich verhindert werden.

Das Zeiterfassungsgerät respektive der Station User muss einem Projekt zugewiesen sein. Andernfalls können die Zeiten nicht auf ein Projekt gebucht werden.

Als Parameter wird nur die ID für den einloggenden Benutzer benötigt. Mittels dieser ID wird der Benutzer aus der Datenbank ermittelt und es wird überprüft, ob der Benutzer im System existiert, die Rolle «Worker» besitzt und ob der Benutzer dem Projekt der Station zugewiesen ist. Durch diese Vorgaben wird vermieden, dass für beliebige ID's Zeiten erfasst werden können und erreicht, dass der Benutzer nur auf Projekte buchen kann, für welche er arbeitet.

Über das Web UI hat der Arbeiter die Möglichkeit auch vor Monatsende seine Zeiterfassungen für den aktuellen Monat abzuschliessen. Nach dem Monatsabschluss sind keine weiteren Zeiterfassungen mehr möglich. Die letzte Vorgabe prüft, ob noch Zeiten für den laufenden Monat erfasst werden können.

Sind alle Vorgaben erfüllt wird der Status Code 201 zurückgeliefert. Ist dies nicht der Fall, wird ein entsprechender Error Code zurückgeliefert. Abbildung 11 zeigt einen Auszug der möglichen Antwortcodes mit entsprechender Beschreibung. Der Auszug stammt von der Dokumentation des API, welche mit Swagger⁵ realisiert ist.

Responses

Response content type

application/json

Code	Description
201	<i>The TimeRecord was successfully added.</i>
400	<i>Station user ID or worker ID was formatted invalid or null.</i>
401	<i>The Worker user is not assigned to the specified project or the date is Locked by a sign-off request.</i>
404	<i>Station user not found or no project assigned to the station user.</i>

Abbildung 11: Auszug aus dem Swagger UI der Antwortcodes

⁵ <https://swagger.io/tools/swagger-ui/>

5.3 Webfrontend

Das Web UI ist eine Implementation eines Clients für das Backend API. Umgesetzt wurde das Web UI wiederum mit ASP.NET Core 2.0. ASP.NET Core verwendet die Razor-Engine⁶ zur Erstellung dynamischer Webseiten. Mit der Razor Syntax ist es möglich beliebige .NET-Core-Ausschnitte auszuführen. Beispielsweise kann eine Benutzerrolle geprüft und entsprechend Buttons ein- oder ausgeblendet werden. Dieser Mechanismus bietet eine grosse Flexibilität und Wiederverwendung von Code in den Views.

Im bestehenden Prototyp war das Anzeigen von erfassten Zeiten bereits möglich, allerdings erst in einem Prototyp Status. Für eine klare Abgrenzung der Arbeit wurden im Rahmen dieser Arbeit zwei neue Pages mit neuem Design für das Anzeigen der Zeiterfassungen erstellt. Die API Methoden, um die erfassten Zeiten zu laden, konnte jedoch wiederverwendet werden.

Nach dem Login wird der Benutzer auf die Startseite weitergeleitet. Auf dieser ist die tägliche Arbeitszeit angezeigt. In einer Tabelle wird für jede Zeile das Projekt, die Kommen -und Gehenszeit sowie der Ersteller des Eintrages festgehalten. Bei Erfassungen über das NFC Zeiterfassungsgerät entspricht der Ersteller dem jeweilige Station User. Auf der zweiten Seite sind alle erfassten Zeiten nach Monat gegliedert einsehbar.

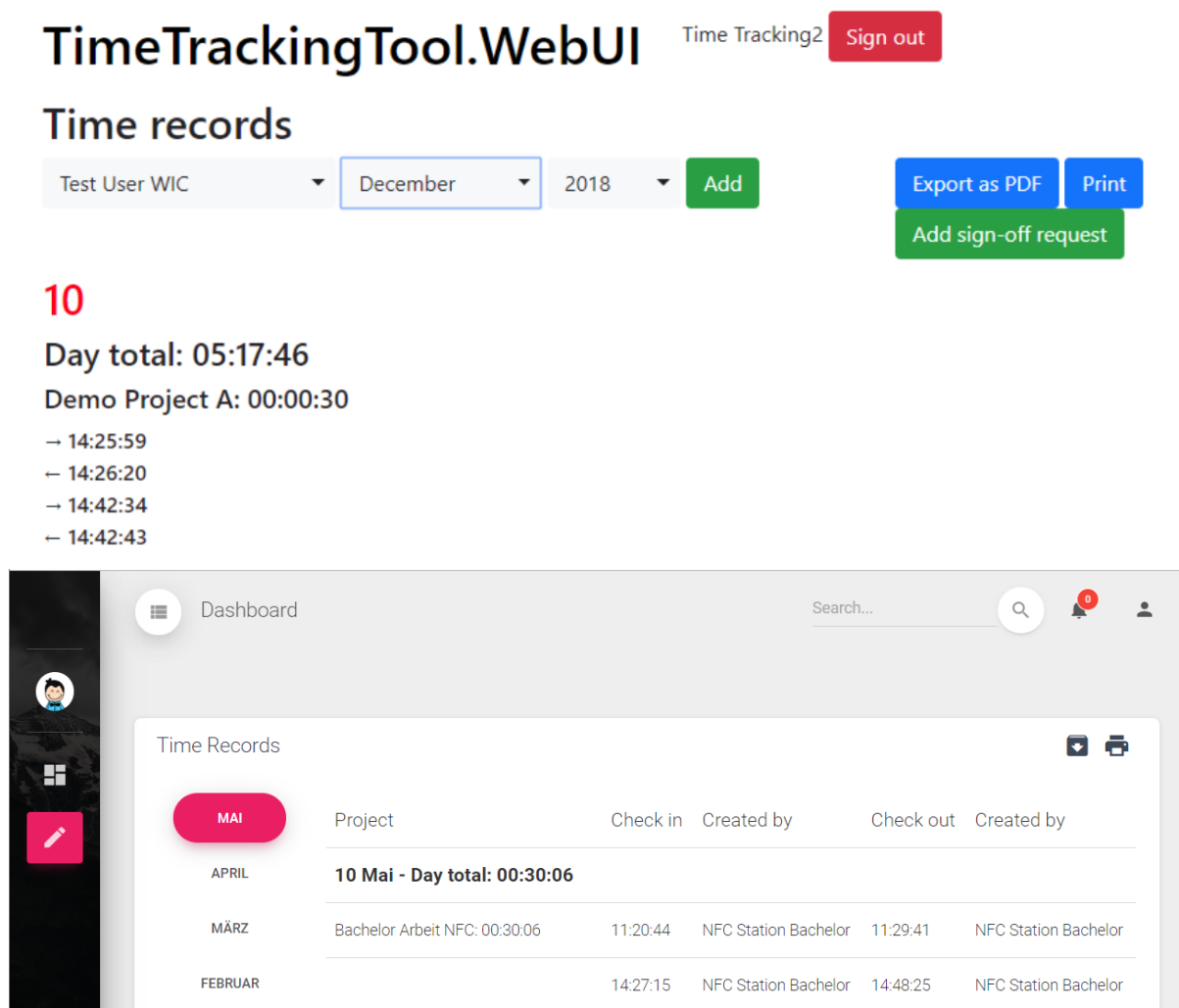


Abbildung 12: Anzeigen der Zeiterfassungen im Prototyp (oben) und dem neuen Zeiterfassungssystem (unten)

⁶ <https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-2.2&tabs=visual-studio>

6 Evaluation und Validation



Abbildung 13: Gesamtsystem

In Tabelle 4 sind die Ziele der Arbeit mit ihrem Status ersichtlich. Damit die Ziele im Laufe des Projektes nachverfolgt und verifiziert werden konnten, wurden die Ziele in Features aufgegliedert, die wiederum in Stories und Tasks unterteilt wurden. Eine detaillierte Liste mit allen Projektitems ist im Anhang C: Auszug Azure DevOps (Sprintplanung) zu finden.

Tabelle 4: Ziele der Arbeit mit ihrem Erfüllungsgrad

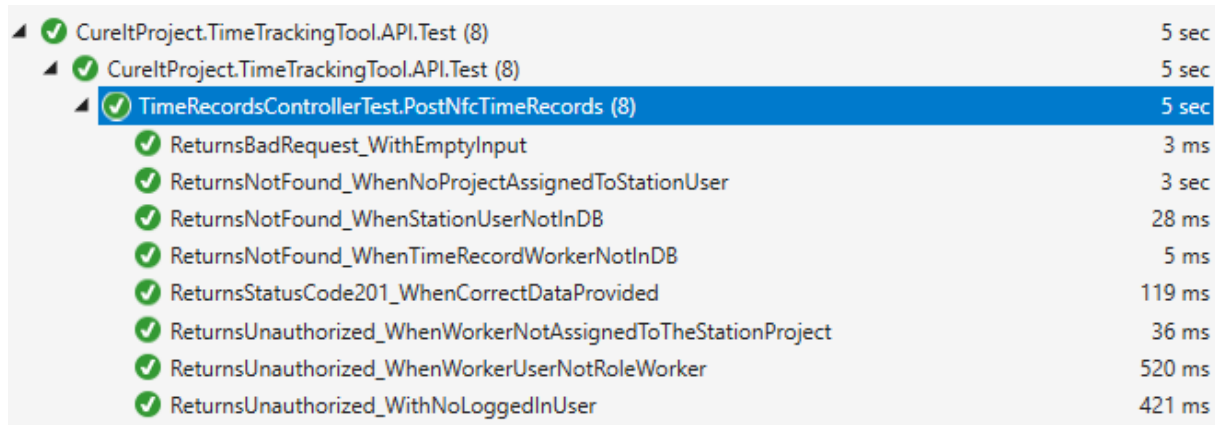
Ziele der Arbeit	Erfüllt	Nicht erfüllt
Auswahl und Beschaffung von geeigneten NFC Komponenten sowohl für den Empfänger, sowie für den Sender.	✓	
Zusammenbau und Konfiguration der Hardware.	✓	
Entwickeln der Software um Zeiten per NFC zu erfassen.	✓	
Erweiterung des Web APIs für die Registrierung der NFC Erfassungen.	✓	
Erweiterung der Webpage für die Darstellungen der NFC Erfassungen.	✓	

Die Sicherstellung, dass die Projektitems korrekt und vollständig implementiert sind, wurde durch das Ausführen von Tests erreicht. Abhängig von der Komponente können das sowohl automatisierte Unit Test oder manuelle Tests sein. Die drei Komponenten des Hardware Systems wurden mit manuellen Tests getestet. Dies hauptsächlich, da sie grosse Abhängigkeiten von Umsystemen haben und nahe an der Hardware sind. Die Hauptaufgabe der Main-Komponente ist das Aufrufen von Methoden auf dem Backend API. Die Reader-Komponente liest NFC Tags und die Authentication-Komponente fordert Tokens von Azure an. Die kompletten Testfälle sind im Hardware Testdokument im Anhang F: Hardware Testdokument ersichtlich.

Obwohl das Backend API für das Schreiben und Lesen der Daten in die Datenbank zuständig ist, wurde das API mit automatischen Unit Tests geschrieben. Um dies zu ermöglichen wurde die In-Memory Datenbank von EF Core verwendet. Das hat den Vorteil, dass für die Tests keine physische Datenbank verwendet werden muss, sondern das gesamte Verhalten der Datenbank im Memory simuliert wird. Somit laufen die Unittests

unabhängig von anderen Umsystemen. Abbildung 14 zeigt die Unittests für NFC Zeiterfassungen im Backend API.

Auf Unit Tests für das Web UI wurde verzichtet, da es sich dabei lediglich um das Anzeigen von Daten ohne Interaktionsmöglichkeiten eines Benutzers handelt. Das Sicherstellen der Funktionalität wird durch die Systemtests abgedeckt.



▲ ✓ CurelProject.TimeTrackingTool.API.Test (8)	5 sec
▲ ✓ CurelProject.TimeTrackingTool.API.Test (8)	5 sec
▲ ✓ TimeRecordsControllerTest.PostNfcTimeRecords (8)	5 sec
✓ ReturnsBadRequest_WithEmptyInput	3 ms
✓ ReturnsNotFound_WhenNoProjectAssignedToStationUser	3 sec
✓ ReturnsNotFound_WhenStationUserNotInDB	28 ms
✓ ReturnsNotFound_WhenTimeRecordWorkerNotInDB	5 ms
✓ ReturnsStatusCode201_WhenCorrectDataProvided	119 ms
✓ ReturnsUnauthorized_WhenWorkerNotAssignedToTheStationProject	36 ms
✓ ReturnsUnauthorized_WhenWorkerUserNotRoleWorker	520 ms
✓ ReturnsUnauthorized_WithNoLoggedInUser	421 ms

Abbildung 14: Unittests für NFC Zeiterfassungen im Backend API

Mit Systemtests wurde die gesamte Funktionalität des NFC Zeiterfassungstools getestet. Dies wurde mit manuellen Tests in einem realen Umfeld der Firma Cure-IT AG verwirklicht. Involviert waren drei Personen, die drei Tage lang ihre Arbeitszeiten über das NFC Zeiterfassungstool getestet haben. Jeder dieser drei Personen hatte einen persönlichen NFC Tag mit einer darauf gespeicherten Benutzer ID. Durch die Systemtests wurde der Hauptworkflow des Zeiterfassungssystems abgedeckt.

- Lesen und Erkennen verschiedener NFC Tags
- Erfassen der Zeiten zu den richtigen Personen
- Anzeigen der Zeiterfassungen auf dem Web UI
- Geräteaufzeit von mehreren Tagen

Die Auswertung der Systemtests ist im Anhang E: Systemtests Testdokument zu finden.

7 Ausblick

Am Ende der Bachelorarbeit steht ein stabiles und getestetes Produkt. Man kann von einem erfolgreichen Projekt sprechen. Es war eine spannende Bachelorarbeit, welche sich durch ihre Vielschichtigkeit auszeichnet. Denn sie befasst sich nicht mit dem reinen Entwickeln, sondern deckt fast den gesamten Produktentwicklungsprozess ab, begonnen mit der Auswahl der geeigneten Hardwarekomponenten und dem Abwägen ihrer Vor- und Nachteile bis hin zu Feldtests in einem realen Umfeld. Das hatte zur Folge, dass ich mich im Rahmen dieser Arbeit mit Situationen beschäftigen musste, die ich im normalen Arbeitsalltag als Softwareentwickler nicht antreffe. Ein Beispiel dafür sind die Risiken mit der Hardware. Was passiert mit dem Projekt, wenn die Lieferzeit mehrere Wochen dauert anstelle den versprochenen drei Arbeitstagen? Oder wie reagiere ich, falls die Hardware unvollständig oder falsch geliefert wird? Solchen Risiken wurde mit einem gezielten Augenmerk auf die Risikoanalyse entgegengewirkt.

Ein weiterer Punkt, der mir während diesem Projekt gut vor Augen geführt wurde ist, dass Design Entscheidungen meistens sowohl Vorteile sowie auch Nachteile nach sich ziehen. Solche Entscheidungen sollten bewusst und im Wissen der Konsequenzen getroffen werden. Erlebt habe ich das in der Wahl der Programmiersprache für das Hardware System. Durch Python ist das Auslesen der NFC Tags gut unterstützt und wird durch eine Library vereinfacht. Auf der Negativseite ist die Authentifizierung auf Azure schwerer im Vergleich zu .NET.

Diesen Trade-Off musste ich eingehen und habe mich bewusst für Python entschieden – prompt bin ich bei der Authentifizierung auf die grössten Probleme im Laufe dieses Projektes gestossen. Auf der anderen Seite hat das Auslesen der NFC Tags reibungslos funktioniert.

Da ich meine Bachelorarbeit alleine gemacht habe, war ich Software Architekt, Software Entwickler und Tester in einem. Das verlangte einiges an Selbstdisziplin um den ordentlichen Projektablauf einzuhalten. Ab und zu habe ich mich ertappt wie die Grenzen fliegend verlaufen sind. Beim Sprintplanen musste der Entwickler das eine oder andere Mal ermahnt werden, dass zuerst vollständig geplant und erst im Anschluss entwickelt wird. Auch die Sprintreviews fielen rückblickend ein bisschen knapp aus. Sie fanden mehr im Rahmen von Statusupdates mit dem Auftraggeber statt.

7.1 Weiteres Vorgehen

Nach dem Abschluss dieser Arbeit wird das NFC Zeiterfassungssystem in einem ersten Schritt im Entwicklungsteam der Cure-IT AG eingesetzt. Durch diesen Einsatz können die Praxistauglichkeit und Stabilität des Produkts über einen längeren Zeitraum geprüft werden. Ziel ist es, das Zeiterfassungsgerät als möglichen Client mit dem Release des gesamten Zeiterfassungstools unseren Kunden anzubieten.

In der aktuellen Version werden die Benutzer ID's auf den NFC Tags unverschlüsselt gespeichert. Die Sicherheitsvorkehrungen setzen ganz auf den Vorteil von NFC, dass der Datenaustausch nur über sehr kurze Distanzen funktioniert. Der Benutzer selber kann jedoch seine eigene ID ohne Probleme auslesen oder verändern. Da das Backend API allerdings durch Rollenbasierte Zugriffskontrolle gesichert ist, ist es schwierig mit der ID Schaden anzurichten. Um die Sicherheit zu steigern, sollten in Zukunft die Tag Informationen verschlüsselt abgespeichert werden.

Das Zeiterfassungsgerät ist in der jetzigen Version genau einem Projekt zugeordnet. Im Normalfall ist das ausreichend. Arbeitet aber ein Mitarbeiter am selben Arbeitsplatz für unterschiedliche Projekte müssten zwei Geräte vorhanden sein. In einer zweiten Version wäre es sicher denkbar, dass einem einloggenden Benutzer alle seine Projekte angezeigt werden. Zum Einloggen wählt der Benutzer über das Touch Display sein Projekt und die Zeiterfassung wird auf das ausgewählte Projekt gespeichert.

Abschliessend kann ich sagen, dass ich sehr zufrieden bin mit dem Resultat meiner Bachelorarbeit. Ich blicke auf 16 arbeitsintensive, aber sehr interessante Wochen mit zahlreichen Erfolgserlebnissen, einigen nervenaufreibenden Knackpunkten und sicher auch der einen oder anderen spannigen Arbeitsstunde zurück.

8 Abbildungsverzeichnis

Abbildung 1: Aufbau des vorhandenen Prototyps.....	6
Abbildung 2: Hauptkomponenten der Applikation mit deren Beziehungen	7
Abbildung 3: Clients inklusive dem NFC Zeiterfassungssystem.....	7
Abbildung 4: NFC-fähige Kreditkarte (Brinkmann, 2019).....	8
Abbildung 5: RFID Sender-Empfänger-System (Access event solutions, 2019).....	8
Abbildung 6: Hardware des Zeiterfassungssystems	11
Abbildung 7: Projektplan für das NFC Zeiterfassungssystem	11
Abbildung 8: Sequenzdiagramm: Zusammenspiel der Systemkomponenten	12
Abbildung 9: Komponentendiagramm der Architektur des Hardware Systems	13
Abbildung 10: Konsolenausgabe nach erfolgreichem Einloggen	14
Abbildung 11: Auszug aus dem Swagger UI der Antwortcodes	15
Abbildung 12: Anzeigen der Zeiterfassungen im Prototyp (oben) und dem neuen Zeiterfassungssystem (unten)	16
Abbildung 13: Gesamtsystem	17
Abbildung 14: Unittests für NFC Zeiterfassungen im Backend API	18

9 Tabellenverzeichnis

Tabelle 1: Übersicht der untersuchten SingleBoard Computer	9
Tabelle 2: Übersicht der untersuchten NFC Lesegeräte.....	10
Tabelle 3: Übersicht der Meilensteine.....	12
Tabelle 4: Ziele der Arbeit mit ihrem Erfüllungsgrad.....	17

10 Literaturverzeichnis

- Access event solutions. (25. 05 2019). *RFID Systems [Grafik]*. Von <https://www.accesseventsolutions.com/rfid-systems> abgerufen
- Brinkmann, M. (25. 05 2019). *ghacks.net [Grafik]*. Von <https://www.ghacks.net/2012/08/21/how-to-protect-your-credit-card-with-rfid-chip-from-unauthorized-scans/> abgerufen
- Configurable token lifetimes in Azure Active Directory*. (23. 05 2019). Von docs.microsoft.com: <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-configurable-token-lifetimes> abgerufen
- Einplatinencomputer*. (25. 03 2019). Von wikipedia: <https://de.wikipedia.org/wiki/Einplatinencomputer> abgerufen
- Near Field Communication*. (14. 03 2019). Von wikipedia: https://de.wikipedia.org/wiki/Near_Field_Communication abgerufen
- RFID*. (14. 03 2019). Von wikipedia: <https://de.wikipedia.org/wiki/RFID> abgerufen

11 Anhang

Anhang A: Evaluationsbericht

Anhang B: Projektmanagementplan

Anhang C: Auszug Azure DevOps (Sprintplanung)

Anhang D: Meilensteinberichte

Anhang E: Systemtests Testdokument

Anhang F: Hardware Testdokument