



Fabian Schäfer

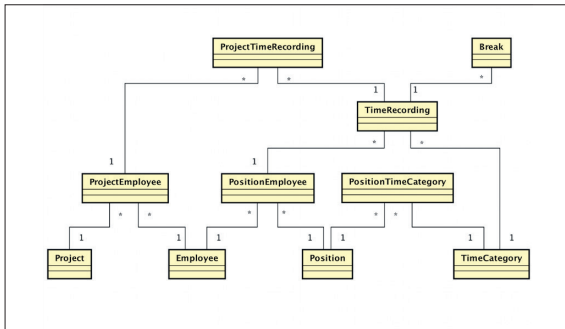


Andreas Stefanits

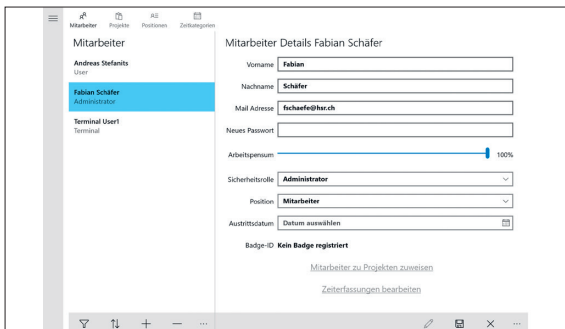
Diplomanden	Fabian Schäfer, Andreas Stefanits
Examinatoren	Prof. Hansjörg Huser, Jürg Jucker
Experte	Stefan Zettel, Ascentiv AG, Zürich, ZH
Themengebiet	Software
Projektpartner	S3CC, Pfäffikon, SZ

## Arbeits- und Projektzeiterfassung

Basierend auf Windows 10 IoT, Raspberry Pi und Microsoft Azure Cloud



Domänenmodell des Backend Servers



Screenshot des Admin-Clients



Testaufbau des Terminal-Clients mit Raspberry Pi inklusive Touch-Display und NFC Reader

**Ausgangslage:** Das Ziel der vorliegenden Bachelorarbeit war es, einen Prototyp für ein Zeiterfassungssystem zu realisieren. Die grundlegende Architektur sollte dabei mit den folgenden Komponenten und Technologien umgesetzt werden:

- Server: Server- sowie Datenbanklogik basierend auf Microsoft Azure Cloud
- Terminal-Client: Zeiterfassung mittels Raspberry Pi Terminal mit Touch-Display, NFC Reader und Windows 10 IoT
- Admin-Client: Systemverwaltung mittels Admin-Client in Form einer Windows 10 Universal App

**Vorgehen/Technologien:** Die Arbeit wurde in eine Architektur-, eine Entwicklungs- und eine Testphase aufgeteilt. Die Architekturphase begann mit der Erfassung und Analyse der Anforderungen. Im nächsten Schritt erfolgte die Evaluation der Technologien, wobei der Fokus auf der Ermittlung einer geeigneten Client-Server-Schnittstelle lag. Diese sollte herstellerunabhängig, sinnvoll erweiterbar sowie einfach aufgebaut sein. Die Entscheidung fiel auf eine HTTP-basierte REST-Schnittstelle, die alle diese Kriterien erfüllt. Im letzten Schritt wurden, basierend auf den gesammelten Erkenntnissen, Domänenmodelle und Klassendiagramme für Client und Server entwickelt, sowie GUI-Konzepte für die Clients erarbeitet. Zu Beginn der Umsetzungsphase teilten wir uns auf, um die Entwicklung des Servers und des Terminal-Clients parallel voranzutreiben. Dank der zuvor detailliert ausgearbeiteten Definition der REST-API, hat das Zusammenspiel zwischen Client und Server auf Anhieb funktioniert. Der im zweiten Teil der Umsetzungsphase entwickelte Admin-Client liess sich ebenfalls nahtlos in die REST-Schnittstelle integrieren. In der letzten Phase, der Testphase, war die Ausarbeitung des Testplans die zentrale Aufgabe. Dieser diente dann als Grundlage für die Systemtests auf dem Terminal- und Admin-Client, um darüber die Funktionalität und Zuverlässigkeit des gesamten Systems zu testen. Ein weiteres wichtiges Testelement waren die Integrationstests, die clientseitig mit automatisierten Tests über die REST-API die korrekte Funktionsweise des Servers überprüften.

**Ergebnis:** Das Ergebnis dieser Arbeit ist ein cloudbasiertes Zeiterfassungssystem mit einer Dokumentation der Architektur und der API sowie einem Benutzerhandbuch. Einige «Nice-to-have»-Anforderungen, die den Rahmen dieser Arbeit gesprengt hätten, können dank der einfach erweiterbaren und gut dokumentierten REST-API mit wenig Aufwand auch nachträglich noch implementiert werden.