

HWR meets Tierpark - Eine prototypische Implementierung eines automatisierten Eingangs zum Tierpark Berlin-Friedrichsfelde

Miklas Boskamp, Martin Böttcher, David Golla, Matthias Klassen, Fabian Klose
Hochschule für Wirtschaft und Recht Berlin

Zusammenfassung—Im Rahmen des Moduls Fachübergreifendes Labor wird den Studierenden die Möglichkeit geboten, Projekte zu realisieren, die eine Zusammenarbeit mit „echten“ Kunden erfordert. Zudem können sie Wissen aus vorherigen Semestern anwenden und ausbauen. Schließlich wird das agile Arbeiten in selbem Semester gelehrt, welches ebenso während des Projektzeitraums Anwendung findet.

Um die Campusattraktivität der HWR in Lichtenberg zu steigern, wird im Rahmen des Projektes ein prototypischer automatisierter Eingang zum Tierpark Berlin entwickelt. Dafür wird zum einen eine biometrische Lösung und zum anderen eine Zeitkarten-Lösung verfolgt. Durch verschiedene Aspekte liegt der Fokus auf der Zeitkarten-Lösung.

Innerhalb dieses Berichtes wird zu Beginn auf die Startphase des Projektes eingegangen. Anschließend wird erläutert, wie das Projekt gemanagt wurde und welche Techniken dabei zum Einsatz kamen. Zudem werden technische / informatische Aspekte aufgezeigt mit anschließender Ergebnisdiskussion und einem Ausblick. Am Ende des Berichtes werden verschiedene Lösungsansätze dargelegt.

I. EINLEITUNG

Die Hochschule für Wirtschaft und Recht (HWR) in Berlin hat, neben einem Campus in Schöneberg, einen Campus in Lichtenberg. Dieser Campus ist, auf Grund nicht vorhandener größerer Grünflächen und einem spärlichen gastronomischen Angebot, wenig attraktiv für den Aufenthalt von Studierenden und anderen Hochschulangehörigen.

Verantwortliche des Tierparks, der HWR und der Politik verständigten sich auf die Durchführung eines Studienprojektes im Rahmen dessen geprüft werden soll, inwiefern eine Umsetzung eines Zugangs zum Tierpark vom Campus der HWR in Lichtenberg möglich ist. Dieses Projekt wird am Fachbereich zwei von Studierenden der Studiengänge Technischen Facility-Management, Bauingenieurwesen und Informatik der HWR Berlin durchgeführt.

Nach bereits abgeschlossenen Projekten in den Modulen Software Engineering I und II, welche unter Berücksichtigung der verschiedenen Phasen, Techniken und Strategien des Software Engineering stattfanden, stehen im Rahmen des Moduls Fachübergreifendes Labor agile Projektmanagementtechniken im Vordergrund.

II. INITIALES PLANEN

Bevor die eigentliche Arbeit an einem Projekt beginnen kann, müssen Grundlagen gelegt werden. So ist es wichtig

festzuhalten, was das Ziel des Projektes ist, wie erste Anforderungen lauten und wie sich die Situation zu Beginn darstellt.

A. Start des Projektes

Erste Informationen, die dem Projektteam in Form eines Dokumentes zur Verfügung gestellt wurden, skizzieren die Kooperation zwischen dem Tierpark und der HWR. In diesem ist, wie im Kapitel I Einleitung beschrieben, die aktuelle Situation aufgezeigt. Ebenso ist dort ebenfalls eine Lösungsmöglichkeit beschrieben [1]:

Umsetzbar wäre dies durch die Schaffung eines automatisierbaren Zugangssystems für Inhaber von Jahreskarten des Tierparks (z.B. über automatische Kontrolle an einer Schleuse mittels biometrischen Merkmalen, Fingerabdrücken o.ä.).

Durch diese Information konnte das Informatik-Team bereits eine erste Idee gewinnen, wie eine Lösung aus Sicht eines Teils der Stakeholder aussehen könnte:

- automatisierter Zugang
- für Inhaber einer Jahreskarte des Tierparks
- Überprüfung mittels biometrischer Merkmale

B. Erste Beratung mit Kunden

Vor einem Treffen mit einem Tierparkvertreter wurden verschiedenen Lösungen vom Informatik-Team gesammelt und ausgearbeitet, damit diese dem Vertreter vorgestellt werden können (siehe VII Verschiedene Lösungsalternativen im Überblick abgesehen von VII-H Zeitkarte). Ebenso wurde innerhalb des Informatik-Teams beraten, welche Möglichkeiten es gibt, die biometrischen Daten zu speichern, und welche Vor- und Nachteile diese bieten (siehe VIII Ort der Speicherung von biometrischen Daten).

Während des Treffens wurde deutlich, dass der Tierpark eine Biometrie-Lösung bevorzugen würden. Weitere wichtige Informationen des Tierparks sind die Anforderungen, dass sie einer Produktion von neuen Karten nicht entgegenstehen, sollte dies notwendig sein, und dass es wichtig ist, dass die Lösung dieses Projektes mit dem Kassensystem, welches der Tierpark verwendet, kompatibel ist.

Im weiterführende Gespräch war der Tierpark nicht davon abgeneigt, eine Jahreskarten-Lösung zu vernachlässigen und stattdessen auf eine Lösung zu setzen, die speziell für HWR-Angehörige gefunden wird. So wurde ebenso die Idee einer Zeitkarten-Lösung (siehe VII-H Zeitkarte) besprochen.

C. Beendigung der Initialphase

Nach Abschluss der Gespräche traf sich das Informatik-Team, um das weitere Vorgehen zu besprechen. Es wurde sich darauf verständigt, die biometrischen Lösungen (Gesichtserkennung und Fingerabdruck) zu verfolgen. Außerdem soll ein Zeitkarten-Prototyp entwickelt. Wie die genaue informatische Umsetzung aussieht, wird in IV Technische / informatische Aspekte erläutert. Einhergehend wurde beraten, welche Hardware benötigt wird, damit diese schnell beschaffen werden kann.

Außerdem verteilte das Team zwei Aufgabenbereiche. So wurde ein Team bestimmt, welches die hardwarenahe Entwicklung, und ein Team, welches die software-technische Entwicklung übernimmt. Innerhalb dieser Teams wurden ebenso Aufgabenbereiche verteilt. Schließlich wurde ein Mitglied bestimmt, welches organisatorische Aufgaben übernimmt.

III. TECHNOLOGIEN / PROJEKTMANAGEMENT

Nachdem in letzten Projekten in Software-Engineering I und II sequenzielle Vorgehensweise praktiziert wurden, wird in Fachübergreifendes Labor agiles Projektmanagement verwendet.

A. Sequenzielle Vorgehensweisen

Sequenzielle Vorgehensweisen, wie zum Beispiel das Wasserfallmodell, haben eine sehr starre Abfolge von Phasen. So wird mit der Implementierung erst begonnen, nachdem die Analyse und der Entwurf fertiggestellt wurden. Für kleine Projekte kann ein Modell dieser Art angebracht sein. Allerdings folgt aus dieser starren Abfolge ein hohe Trägheit des Projektteams. Sollte es während einer späteren Projektphase eine Änderung einer Anforderung geben, ist großer Aufwand notwendig, um diese Änderung einzupflegen [2].

Außerdem ist die Arbeitsweise in solchen Modellen relativ strikt. Es wird zu wenig zwischen den verschiedenen Mitgliedern gesprochen. Dies ist ebenso eine Fehlerquelle[2].

Schließlich bedeuten schlechte Anforderungen ein schlechtes Projekt. Sollte die anfänglich definierten Anforderungen nicht gut genug sein, wird nicht korrekt implementiert. Innerhalb dieser Modelle bedeutet Änderung einer Anforderung, wie oben bereits beschrieben, Aufwand[2].

B. Agiles Arbeiten

Wodurch können diese negativen Aspekte verhindert werden? Es können agile Werte und Prinzipien verfolgt werden. Es wurde ein Agiles Manifest[3] entwickelt, an dessen sich das Informatik-Team orientiert.

Folgende Werte beinhaltet das Agile Manifest:

- 1) Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- 2) Funktionierende Software mehr als umfassende Dokumentation
- 3) Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
- 4) Reagieren auf Veränderung mehr als das Befolgen eines Plans

Das Informatik-Team versucht während des Projektzeitraums diese Werte umzusetzen. So wurde auf die Bedürfnisse eines jeden Teammitglieds eingegangen. Jeder konnte jederzeit Meinungen über die aktuelle Arbeit äußern (bevorzugend während einer Retrospektive). Außerdem liegt das Augenmerk im Team auf die Software. Zwar ist jedem Mitglied bewusst, wie wichtig Dokumentation ist, dennoch werden nur wichtige Dinge konkret dokumentiert, damit eine schnelle und möglichst fehlerfreie Implementierung gewährleistet werden kann. Schon vor Beginn des eigentlichen Projektzeitraums war es dem Team wichtig in Kontakt mit anderen Stakeholder zu treten. Schließlich stellt das Team durch das Anwenden agiler Techniken sicher, dass eine Änderung einer Anforderung gut in das Gesamt-Projekt einpflegbar ist.

Auch die zwölf Prinzipien, die im Agilen Manifest definiert sind, werden vom Team berücksichtigt. Da im Rahmen dieses Berichtes nicht alle Prinzipien erläutert werden können, wird auf die für das Team wichtigsten Prinzipien eingegangen.

Ein wichtiges Prinzip ist das fünfte Prinzip¹. Zwischen der Mitglieder herrscht die Philosophie der Motivation, Unterstützung und des Vertrauens. Jedes Mitglied kann Probleme mitteilen. Es wird sich darum gekümmert. Ebenso wird jedem Mitglied die Freiheit gelassen, um seine Arbeit gut zu erledigen. Es wird sich gegenseitig darauf vertraut, dass Arbeiten erledigt werden.

Weiterhin wurde durchgehend versucht notwendige Treffen auf Daten und Zeiten zu legen, zu denen ein Treffen möglich ist. Die effektivste Möglichkeit Informationen auszutauschen ist ein Treffen zwischen den Mitgliedern².

Das komplette Projekt wird so einfach wie möglich gehalten³. So wird nur das umgesetzt, was zwingend notwendig ist, damit Ressourcen in nur notwendige Projektbestandteile investiert werden.

Schließlich ist das stetige reflektieren ein wichtiger Punkt⁴. Das Team trifft sich regelmäßig um über die vergangene Zeit zu sprechen und festzuhalten, was gut gelaufen ist und was verbessert werden kann (siehe ebenso III-C Iterativer Ansatz und III-D Retrospektive).

C. Iterativer Ansatz

Ferner wird im agile Projektmanagement ein Iterativer Ansatz gewählt. Das heißt, dass Abläufe während des Projektes wiederholt werden. Wie kann das konkret aussehen?⁵

Ein sehr bekanntes Vorgehen im Projektmanagement mit einem Iterativen Ansatz ist Scrum. Das Herz von Scrum ist

¹Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.

²Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.

³Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.

⁴In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

⁵Folgendes Beispiel ist lediglich eine Darstellung eines möglichen Rahmenwerkes, welches in Teams eingesetzt werden kann, welches ein iterativen Ansatz verfolgt.

der Sprint. Der Sprint ist eine Zeitbox, in der ein nutzbares Produkt-Inkrement entstehen soll. Sobald ein Sprint beendet wurde, beginnt ein neuer Sprint. Während eines jeden Sprints werden bestimmte sogenannte Sprint-Ereignisse durchgeführt. Ein Sprint startet mit einem Sprint Planning und endet mit einem Sprint Review und einer Sprint Retrospektive. Täglich wird ein Daily Scrum durchgeführt.[4]

Der Vorteil von iterativen Vorgehensweisen ist zum Einen das stetige Feedback, welches mit in die Projektplanung einfließen kann. So schaut das Team am Ende einer Iteration auf diese zurück und bewertet diese. Ebenso kann der Kunde mit einbezogen werden und ebenfalls Feedback äußern.

Außerdem wird zum Anderen durch iteratives Vorgehen eine Grundlage für eine Prognose gelegt, indem die „Geschwindigkeit“ des Teams gemessen wird. Wenn verschiedene Aufgaben mit einem Umfang bewertet werden, sieht man zurückblickend, welchen Umfang alle Aufgaben, welche in der vergangenen Iteration bearbeitet wurden, haben. Im Laufe der Zeit lässt sich eine „Durchschnitts-Geschwindigkeit“ berechnen. Anhand dieser ist es möglich zu prognostizieren, wann zum Beispiel ein Meilenstein erreicht werden kann.

Das Informatik-Team verständigte sich auf den Einsatz eines iterativen Vorgehens. So wird zu Beginn einer Iteration geplant, was innerhalb der nächsten Iteration geschafft werden soll. Dies geschieht auch mit der Erfahrung, was in vergangenen Iterationen geschafft wurde. So kann das Team speziell am Ende der Projektphase planen, was bis zum Abschluss des Projektes geschafft werden kann.

Am Ende einer Iteration wird eine Retrospektive durchgeführt (siehe auch III-D Retrospektive). Durch die Retrospektiven konnte das Team zusammen feststellen, was in der letzten Iteration gut lief und was verbesserungswürdig ist.

D. Retrospektive

Retrospektiven dienen dem Team um die Arbeitsweise zu überprüfen. Dies findet in einem geschützten Rahmen statt, indem nur das Team daran teilnimmt. In Teams in denen der iterative Ansatz verfolgt wird, bezieht sich eine Retrospektive oft nur auf die vergangene Iteration[5].

Ein Ziel einer Retrospektive ist die Verbesserung der Zusammenarbeit im Team. Dadurch kann die Effektivität und die Effizienz des Teams gesteigert werden. Ebenso verfolgt in der Regel ein Team, welches Retrospektiven nutzt, das Ziel, dass ein Mitglied eventuell aufgestauten Frust abbauen kann, indem er seine Kritikpunkte äußern kann[5]. Ebenso können durch Retrospektiven Personen gewürdigt werden, indem Leistungen dieser Personen gesondert hervorgehoben werden.

Zu Beginn einer Retrospektive wird versucht eine lockere Stimmung aufzubauen, da Personen in einer gelassenen Atmosphäre eher dazu geneigt sind, ihre Meinung kundzugeben und mit anderen zu teilen. Anschließend werden Informationen und Erkenntnisse gesammelt mit denen eventuell Maßnahmen beschlossen werden können[5].

Das Team entschied sich das Plus-Delta-Vorgehen zu nutzen. Es wird eine Tabelle angelegt (siehe Tabelle I), mit zwei Spalten, in denen man zum einen festhält, was während der

letzten Iteration gut war, und zum anderen, was in zukünftigen Iterationen verbessert werden kann.

+	Δ
viele Treffen mit guten Resultaten	Anforderungen besser formulieren
gute Vorbereitung auf Treffen mit Kunden	Tisch- und Sitzanordnungen vor einem Treffen besser prüfen
Produktive Diskussionen innerhalb des Teams	

Tabelle I
BEISPIELHAFTE TABELLE EINER RETROSPEKTIVE IM PLUS-DELTA-VORGEHEN

IV. TECHNISCHE / INFORMATISCHE ASPEKTE

Wie in Abbildung 2 (Veranschaulichung der „Karten mit ein- und auschecken“-Lösung) gezeigt, besteht das Projekt aus drei Komponenten. Zum einen gibt es die hardwarenahe Seite, welche als Schnittstelle zu den verschiedenen Hardware-Komponenten (Kamera, Fingerabdruck-Scanner, Barcode-Scanner) dienen. Außerdem sind diese als Socket-Clients zu betrachten. Dort wurde sich für die Programmiersprache Python entschieden.

Die zweite Komponente ist der Socket-Server, welcher die zentrale Rolle innerhalb des Projektes einnimmt. Die Socket-Clients sind dauerhaft in Verbindung mit dem Server. Ebenfalls ist auf dem Server ein Modul implementiert, welches die Schnittstelle der Datenbank ist. Die Sprache, mit der der Server implementiert ist, ist JavaScript.

Die dritte Komponente ist schließlich die Datenbank. Dort wird MongoDB⁶ genutzt, welches eine NoSQL⁷-Datenbank ist.

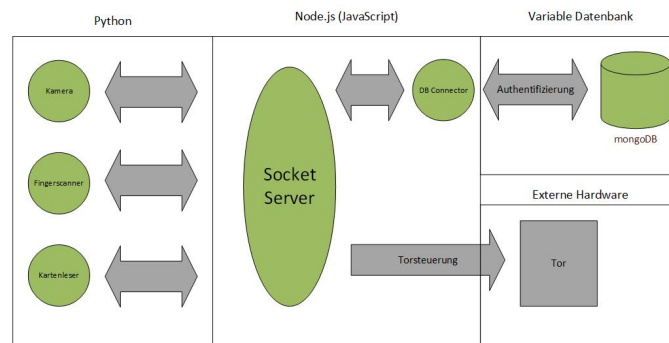


Abbildung 1. Darstellung der eingesetzten Architektur

A. Hardwarenahe Entwicklung

1) *Fingerabdruck-Scanner:* Für die Erkennung einer Person mit Hilfe eines biometrischen Merkmals, wurde ein Fingerabdruck-Scanner des Herstellers Seede Studio erworben. Dieser verwendet zum Scannen einen optischen Sensor, der zwar schnell und genau ist, aber schon mit einem hochauflösenden Ausdruck des Fingerabdrucks auf Papier,

⁶<https://www.mongodb.com/de>

⁷Not only SQL

überlistet werden kann. Da jedoch nur ein prototypischer Eingang entwickelt werden sollte, wurden die Kosten des Fingerabdruck-Scanners höher priorisiert, als die Sicherheit.

Der Datenaustausch zwischen Raspberry Pi 3 und Fingerabdruck-Scanner findet über eine UART-Schnittstelle statt. Die Schnittstelle verfügt über zwei Datenleitungen und zwei Leitungen für die Spannungsversorgung, die eine serielle (bitweise) Datenübertragung ermöglichen. Um den Scanner zu verwenden, muss der Raspberry Pi 3 vordefinierte Befehle über die UART-Schnittstelle senden, die der Scanner mit einem Acknowledgement (Bestätigung) beantwortet.

Nachdem der Fingerabdruck-Scanner an den korrekten GPIO-Pins (General Purpose Input/Output) des Raspberry Pi 3 angeschlossen wurde, begann die Programmierung mit Python. Nach der Implementierung der Logik für die UART-Schnittstelle wurde getestet, ob der Scanner Befehle empfangen und ausführen konnte. Es wurden allerdings keine Acknowledgements empfangen, weshalb davon ausgegangen wurde, dass die Schnittstelle nicht richtig funktionierte. Um diese Vermutung zu überprüfen, wurde ein Arduino mit dem Raspberry Pi 3 verbunden und wiederum Daten über die UART-Schnittstelle verschickt. Dieser Test bewies, dass die UART-Schnittstelle des Raspberry Pi 3 ordnungsgemäß funktionierte und die Fehlerursache an einer anderen Stelle lag. Zu diesem Zeitpunkt wurde der Fokus der Arbeit jedoch auf die Zeitkarten-Lösung gerichtet, weshalb die Fehlersuche vorerst eingestellt wurde.

2) *Kamera*: Bedenken auf Seiten des Tierparks, dass der Fingerabdruck-Scanner im Winter nicht zuverlässig arbeiten könnte, waren ausschlaggebend, um als alternatives biometrisches Merkmal das Gesicht in Betracht zu ziehen. Für die Gesichtserkennung vor Ort wurde als erstes überlegt, eine CMUcam5 Pixy des Herstellers Charmed Labs zu nutzen. Gründe hierfür waren zum einen die Wahlmöglichkeit der Datenübertragung mit einem USB-Kabel oder auch eine direkte Verbindung mit 6 Datenleitungen zwischen dem Raspberry Pi 3 und der Kamera. Zum anderen ist die Kamera von Hause aus im Stande, gelernte Objekte zu erkennen. Wenn es also möglich ist, der Kamera beizubringen, jedes Gesicht als Objekt zu erkennen, wird das Gesicht einzeln als Objekt übermittelt und wäre so, ohne den Hintergrund, einfacher mit dem Datensatz zu vergleichen. Letztendlich stellte sich aber heraus, dass zum einen die Auflösung tatsächlich zu gering ist, um Gesichter vergleichen zu können und darüber hinaus die Kamera nur mit einem manuellen Fokus versehen ist. Die Gegebenheiten im Einsatzbereich machen eine geringe Anpassung erforderlich, um sowohl unterschiedlich nahe, als auch unterschiedlich große Personen gleich scharf erfassen zu können. Ein automatisierter Fokus direkt in der Kamera ist entsprechend unbedingt nötig.

Aktuelle Bilddaten verarbeitende Algorithmen zur Gesichtserkennung, wie sie zum Beispiel bei Microsoft oder Apple zum Einsatz kommen, erreichen heute noch keine hundert prozentige Zuverlässigkeit, weswegen der Einsatz einer solchen Kamera und entsprechender softwareseitiger Verarbeitung der Daten, heute nur als Ergänzung zum Fingerabdruck-

Scanner dienen könnte, um dort auszuschließen, dass nur ein Ausdruck vor den Scanner gehalten wird. Um einen Betrug durch Vorhalten eines Fotos auszuschließen, wäre der Einsatz einer zweiten Kamera mit einem leicht veränderten Winkel zu erwägen. Darüber hinaus wäre die Wahl einer Kamera hilfreich, die selbst über verschiedene Auslösemodi verfügt und so gleichzeitig verschiedene Bilder liefern kann. Zum Beispiel durch Anpassung der Blende, Belichtungszeit, aber auch eine Aufnahme in Graustufen. Diese verschiedenen Bilder können den Vergleich mit auf gleiche Weise gemachten Fotos in der Datenbank vereinfachen.

3) *Barcode-Scanner*: Die Verwendung eines Barcode-Scanners erlaubt es, den Universitätsausweis als Eintrittskarte für einen automatisierten Zugang zu verwenden. Dafür wird, bis auf den Barcode-Scanner an sich, keine weitere Hardware benötigt. Außerdem ist kein spezieller Barcode-Scanner nötig, es können alle Barcode-Scanner verwendet werden. Für die Verwendung muss der Barcode-Scanner lediglich über eine USB-Schnittstelle verbunden werden. Ein Barcode besteht aus einer Abfolge von schmalen und breiten Balken, die entweder weiß oder schwarz sind. Der Barcode-Scanner ermittelt mit Hilfe von Reflektion die Farbe und breite der verschiedenen Balken. Diese Balken stehen für binäre Werte. Diese Werte werden ins Dezimalsystem umgerechnet und ermöglichen so die Bestimmung des gescannten Universitätsausweises. Diese Aufgabe wird in wenigen Millisekunden erledigt und eignet sich damit ausgezeichnet als schnelle Einlasskontrolle.

B. Socket-Verbindung

Für die zuvor beschriebenen Sensoren wurde ein Socket-Client geschrieben, der von allen Sensorprogrammen verwendet werden kann. Die Socket-Clients verbinden sich beim Start mit dem Socket-Server. Dieser dient in erster Linie dazu, die Kommunikation zwischen einem Eingangstor (Sensor) und der Datenbank zu generalisieren und zentralisieren. So ist es möglich mehr als ein Tor zu betreiben oder verschiedene Authentifizierungsmöglichkeiten anzubieten und weiterhin nur eine Datenbank zu verwenden. Die Clients kommunizieren mit dem Server indem sie JSON-Strings versenden und empfangen. Möchte ein Parkbesucher den Tierpark durch ein automatisiertes Tor betreten, muss er sich zunächst authentifizieren. Wie bereits beschrieben wurde hierfür zunächst ein Zeitkarten-Modell gewählt. Der Besucher scannt also sein Ticket. (beispielsweise seinen Studentenausweis) Daraufhin versendet der zugehörige Socket-Client eine JSON-Nachricht an den Socket-Server.

```
{
  "user": "9783238423",
  "entry": true,
  "time": 1512566524.54
}
```

Listing 1. Beispiel Request eines Socket-Clients

Versendet wird immer eine JSON Struktur. Für den key „user“ wird die eingescannte Nummer des Tickets übergeben. Der key „entry“ beschreibt, ob der Besucher den Park durch

das Tor betritt (entry = true) oder verlässt (entry = false). Der key „time“ gibt einen Timestamp für den Zeitpunkt des Eintritts/Austritts an. Dieser wird später zur Berechnung der Besuchsdauer verwendet.

Der Socket-Server nimmt den Request entgegen baut eine Verbindung mit der Datenbank auf. Es wird geprüft, ob der Besucher berechtigt ist den Park zu betreten und ob genügend Zeit-Guthaben verfügbar ist. Wenn alle nötigen Informationen aus der Datenbank gesammelt wurden, sendet der Server eine Antwort an den Client.

```
{
  "user": "9783238423",
  "success": true,
  "remaining_time": 52
}
```

Listing 2. Beispiel Response des Socket-Servers

In der Antwort wird erneut die Kartenummer übergeben. Zusätzlich zeigt der key „success“ an, ob der Besucher den Park betreten darf. Dieses Flag wird von der Torsoftware ausgewertet und das Tor wird gegebenenfalls geöffnet. Abschließend zeigt der key „remaining_time“ an wie viel Zeit der Besucher noch auf seiner Karte gespeichert hat.⁸

Für die Implementierung des Socket-Servers wurde Javascript und Node.JS verwendet. Der Socket-Client wurde in Python entwickelt. Die Kommunikation findet über das standardisierte WebSocket-Protokoll statt.

C. Datenbank

Die dritte Komponente der Architektur ist die Datenbank. Das Team verständigte sich auf die Nutzung einer NoSQL-Datenbank. Diese Art von Datenbanken verfolgen einen nicht relationalen Ansatz, wodurch das ACID⁹-Modell keine Anwendung findet, aber Komplexität vermieden werden kann und eine hohe Performance und ein hoher Durchsatz erzielt wird.[6]

Im Gegensatz zu SQL-Datenbanken werden bei NoSQL-Datenbanken die Daten nicht in Tabellen mit Spalten und Zeilen gespeichert. Grundsätzlich lassen sich NoSQL-Datenbanken in vier verschiedene Arten unterteilen, wie sie die Daten speichern[6]:

- dokumentenorientierte Datenbank
- Key-Value-Datenbank
- Graphendatenbank
- spaltenorientierte Datenbank

In einer dokumentenorientierten Datenbank werden die Daten in Dokumenten gespeichert. Solch ein Dokument ist eine schemalose Zusammenstellung von Daten. Diese können, aber müssen nicht, strukturiert sein. Zwischen den Dokumenten werden keine Relationen gespeichert[6]. Das Team entschied sich zur Anwendung einer solcher Datenbank, da

jeder Anwender in solch einem Dokument gespeichert werden. Außerdem kann das Team mit dieser Art der Datenbank flexibel arbeiten. So können für jeden Nutzer verschiedene Daten gespeichert werden, sollte dies notwendig sein. Auch die Entwicklung kann flexibler gestaltet werden, als mit einer herkömmlichen Datenbank, da unter Umständen keine Tabellen neu angelegt oder Spalten umbenannt werden müssen.

Entschieden wurde sich für die dokumentenorientierte Datenbank MongoDB. Das Dateiformat, welches MongoDB nutzt, ist BSON¹⁰. Dieses ist stark an JSON¹¹ angelehnt und kann leicht zu JSON konvertiert werden. Ebenso funktioniert dies auch in die andere Richtung: Zur Speicherung eines JavaScript-Objektes in einer MongoDB ist keine Konvertierung nötig.

Außerdem können mit einer MongoDB Umbaumaßnahmen innerhalb der Datenbank auch während des laufenden Betriebes vorgenommen werden. Das heißt, dass bei einer Produktivsetzung des Projektes durch Wartungen an der Datenbank kein Ausfall eben dieser zu befürchten ist und das System weiterhin erreichbar bleibt.

Die MongoDB ist ein Open-Source-Projekt. Dadurch fallen ebenso keine Kosten an, wie auch, dass eine große Community um das Projekt existiert. Eine Weiterentwicklung ist gewährleistet und bei Fragen wird einem von der Community geholfen. Zudem wird eine umfangreiche Dokumentation gepflegt, welche die Arbeit damit ebenso unterstützt.

V. ERGEBNISDISKUSSION

Nach Beendigung des Projektes ist ein Prototyp entstanden, der die Möglichkeit zeigt, wie die informationstechnische Seite des Projektes aussehen kann. Durch verschiedene Aspekte, wie beispielsweise Zeitknappheit, wurde auf die Lösung des Zeitkarteneinsatzes (siehe VII-H Zeitkarte) fokussiert. Dabei wurde ebenso auf die zu Beginn des Projektes geeinigte Architektur geachtet, wie auch auf die Implementierung unter Beachtung des Einsatzes eines Barcode-Scanners. So muss bei Nutzung des Prototypens kein virtueller Scanner simuliert werden, sondern es kann ein realer Scanner verwendet werden.

Die Implementierung einer Jahreskarten-Lösung mit Hilfe von Biometrie wurde weniger Beachtung geschenkt, da im Laufe des Projektes die Komplexität der Entwicklung solch eines Prototypens deutlich wurde und entschieden wurde, dass dies im Rahmen dieses Projektzeitraums nicht möglich sein würde. Dennoch wurde erste Versuche mit einer Kamera und einem Fingerabdruck-Scanner durchgeführt, die für spätere Entwicklungen nutzbar sind.

VI. AUSBLICK

Durch die erste Version eines Prototypens ist eine Grundlage geschaffen worden, die späteres Arbeiten an diesem Projekt gut ermöglicht. So wird es notwendig sein, zu prüfen, wie die Anforderungen an die Zeitkarten-Lösung verfeinert werden können. Beispielsweise ist zu klären, wie den Nutzer die

⁸Vollständige Dokumentation der Schnittstelle:
<https://github.com/HWRmeetsTierpark/HWRTP-core/wiki/Socket-Schnittstelle>

⁹Atomicity, Consistency, Isolation, Durability

¹⁰Binary JSON

¹¹JavaScript Object Notation

Möglichkeit gegeben wird, Zeit, die sie im Tierpark verbringen möchten, kaufen können.

Der weit größere Anteil wird die Jahreskarten-Lösung in Anspruch nehmen. Es wird zum Beispiel zu prüfen sein, wie sich zum einen biometrische Daten am besten vergleichen lassen, und zum anderen, wie sich Daten aus den Hardware-Komponenten am besten lesen lassen und für den Vergleich vorbereitet werden können.

VII. VERSCHIEDENE LÖSUNGsalTERNATIVEN IM ÜBERBLICK

In diesem Abschnitt sind die verschiedenen Lösungsmöglichkeiten übersichtlich dargestellt. Zunächst wird die Lösung kurz beschrieben. Teilweise werden Lösungen mit Grafiken veranschaulicht¹². Anschließend ist eine Gegenüberstellung von Vor- und Nachteilen dargestellt.

A. Karten mit ein- und auschecken

Um in den Park zu kommen muss der Nutzer mit einer Karte in den Park einchecken. Nach dem Einchecken ist kein weiteres Eintreten in den Park möglich. Sobald der Nutzer diese Karte zum Auschecken aus dem Park nutzt, wird die Sperrung zum Einchecken entfernt, wodurch ein erneutes Eintreten möglich ist. Sollte der Nutzer einen anderen Ausgang wählen, wodurch kein Auschecken möglich ist, wird die Karte nach einer Zeit x oder zu einem Zeitpunkt x entsperrt, wodurch anschließend ein Einchecken in den Park wieder möglich ist.

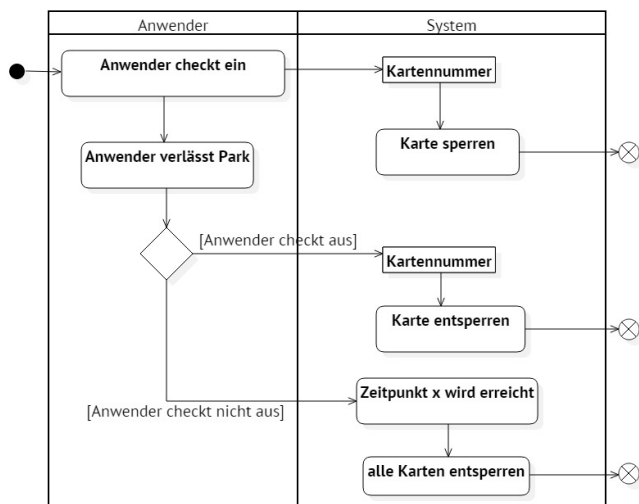


Abbildung 2. Veranschaulichung der „Karten mit ein- und auschecken“-Lösung

¹²Die Grafiken sind nicht komplett UML-konform, da diese zur Veranschaulichung für Kunden (nicht UML-vertrauten Personen) genutzt werden.

Vorteile	Nachteile
unkomplizierte Entwicklung	Karten gehen leicht verloren, d. h. Mitarbeiter für Sperrung nötig
einfache Anwendung für den Nutzer	Karten müssen produziert werden
nicht so datenschutzkritisch	Karten können weitergegeben werden

Tabelle II
VOR- UND NACHTEILE DER „KARTEN MIT EIN- UND AUSCHECKEN“-LÖSUNG

B. Gesichtserkennung

Um den Park zu betreten wird das Gesicht analysiert und bewertet, ob diese Person eine Jahreskarte besitzt. Nachdem der Nutzer in den Park eingetreten ist, wird diese Person für einen weiteren Eintritt innerhalb der Zeit x gesperrt.

Vorteile	Nachteile
biometrische Lösung, d. h. personenbezogen	datenschutzkritisch
aufwändigere Manipulation	noch nicht ausgereift
	Eintrittsvorgang zeitaufwendig

Tabelle III
VOR- UND NACHTEILE DER GESICHTSERKENNUNGS-LÖSUNG

C. Handerkennung

Um den Park zu betreten wird geprüft, ob der Nutzer denselben Handvenenabdruck hat, wie der hinterlegte. Zudem wird geprüft, ob dieser Nutzer eine Jahreskarte hat. Nachdem der Nutzer in den Park eingetreten ist, wird diese Person für einen weiteren Eintritt innerhalb der Zeit x gesperrt.

Vorteile	Nachteile
biometrische Lösung, d. h. personenbezogen	Hygiene
	datenschutzkritisch

Tabelle IV
VOR- UND NACHTEILE DER HANDERKENNUNGS-LÖSUNG

D. Fingerabdruck

Um den Park zu betreten wird der Fingerabdruck analysiert und bewertet, ob diese Person eine Jahreskarte besitzt. Nachdem der Nutzer in den Park eingetreten ist, wird diese Person für einen weiteren Eintritt innerhalb der Zeit x gesperrt.

Vorteile	Nachteile
biometrische Lösung, d. h. personenbezogen	datenschutzkritisch
aufwändigere Manipulation	

Tabelle V
VOR- UND NACHTEILE DER FINGERABDRUCK-LÖSUNG

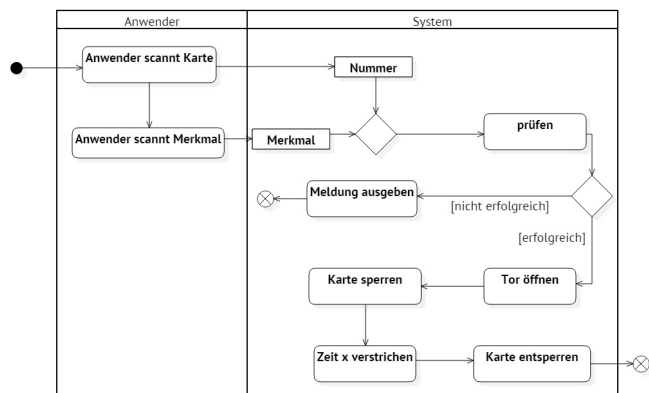


Abbildung 3. Veranschaulichung der „Biometrie“-Lösungen (VII-B Gesichtserkennung, VII-C Handerkennung und VII-D Fingerabdruck)

E. Personalausweis mit Nummer

Um den Park zu betreten wird geprüft, ob der Nutzer mit dieser Personalausweisnummer eine Jahreskarte besitzt. Nachdem der Nutzer in den Park eingetreten ist, wird diese Person für einen weiteren Eintritt innerhalb der Zeit x gesperrt.

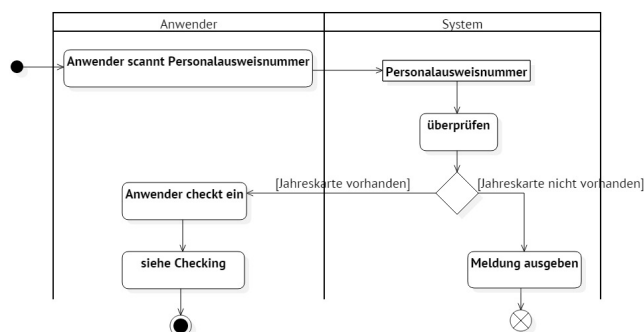


Abbildung 4. Veranschaulichung der „Personalausweis mit Nummer“-Lösung

Vorteile	Nachteile
personenbezogen (eindeutig), d. h. personenbezogen	Personalausweis kann weitergegeben werden
nicht so Datenschutzkritisch (mit passendem Lesegerät)	
Implementierung wenig aufwendig	
für alle Arten von Personalausweis innerhalb Deutschlands nutzbar (alt und neu)	

Tabelle VI

VOR- UND NACHTEILE DER „PERSONALAUSWEIS MIT NUMMER“-LÖSUNG

F. Personalausweis mit Fingerabdruck

Um den Park zu betreten wird geprüft, ob der Nutzer denselben Fingerabdruck hat, wie dieser, der auf dem Personalausweis gespeichert ist. Zudem wird geprüft, ob dieser Nutzer eine Jahreskarte hat. Nachdem der Nutzer in den Park

eingetreten ist, wird diese Person für einen weiteren Eintritt innerhalb der Zeit x gesperrt.

Vorteile	Nachteile
biometrische Lösung, d. h. personenbezogen	nur neue Ausweisart nutzbar
nicht so datenschutzkritisch (da Speicherung auf Personalausweis, d. h. Projektteam für Sicherheit nicht verantwortlich)	nur für Personen nutzbar, die den Fingerabdruck auf dem Ausweis speichern lassen haben

Tabelle VII

VOR- UND NACHTEILE DER „PERSONALAUSWEIS MIT FINGERABDRUCK“-LÖSUNG

G. App

Damit ein Nutzer in den Park eintreten kann, muss sich dieser mittels des Fingerabdrucks oder ähnlichem an seinem Handy verifizieren. Anschließend wird ein QR-Code generiert, mit dem Eingetreten werden kann. Anschließend wird der Code ungültig und kann nicht mehr genutzt werden.

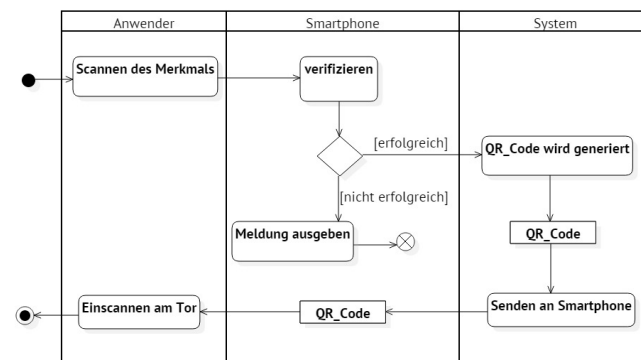


Abbildung 5. Veranschaulichung der „App“-Lösung

Vorteile	Nachteile
einfache Anwendung für Nutzer	QR-Code kann weitergeschickt werden
Implementierung wenig aufwendig	Mehrere Fingerabdrücke können im Handy gespeichert sein

Tabelle VIII

VOR- UND NACHTEILE DER APP-LÖSUNG

H. Zeitkarte

Es gibt eine Karte mit einem Barcode (zum Beispiel Studenenausweis). Mit diesem Barcode kann Zeit gekauft werden, die man im Park verbringen darf. Das heißt, beim Ein- und Austritt muss diese Karte gescannt werden.

Diese Lösung sollte allerdings nur für HWR-Angehörige nutzbar sein, da diese speziell aus der Kooperation zwischen der HWR und dem Tierpark entstanden ist. Außerdem wäre diese Karte nur von Montag bis Freitag und nicht an Feiertagen nutzbar.

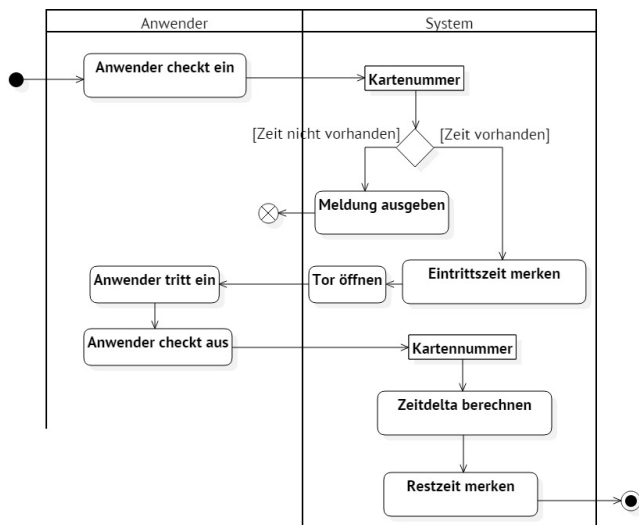


Abbildung 6. Veranschaulichung der „Zeitkarten“-Lösung

Vorteile	Nachteile
einfache Anwendung für Nutzer	Karte kann weitergegeben werden (hat aber weniger Nutzen)
Implementierung wenig aufwendig	Bevorzugung von HWR-Angehörigen vor sonstigen Personen
autark vom Kassensystem des Tierparks	

Tabelle IX
VOR- UND NACHTEILE DER ZEITKARTEN-LÖSUNG

VIII. ORT DER SPEICHERUNG VON BIOMETRISCHEN DATEN

Folgend werden die verschiedenen Orte für die Speicherung von biometrischen Daten aufgeführt, beschrieben und bewertet

A. Biometrische Datenspeicherung auf einer Karte

Die biometrischen Vergleichsdaten werden auf einer Karte gespeichert, welche dem Nutzer ausgehändigt wird. Beim Eintrittsvorgang des Nutzers muss dieser die Karte einlesen lassen und entsprechendes biometrisches Merkmal von ihm prüfen lassen.

Vorteile	Nachteile
keine sichere Kommunikation zwischen Client und Server notwendig	Karten gehen verloren; Daten gehen leicht verloren
Datenschutz leichter einhaltbar	beide Merkmale (Jahreskartenbesitz und biometrisches Merkmal) in einer Hand, d. h. Manipulation eher möglich
	aktuelle Jahreskarten nicht nutzbar

Tabelle X
VOR- UND NACHTEILE DER SPEICHERUNG DER BIOMETRISCHEN DATEN AUF EINER KARTE

B. Biometrische Datenspeicherung in einer Datenbank

Die biometrischen Vergleichsdaten werden in einer Datenbank gespeichert. Der Nutzer muss beim Eintrittsvorgang das entsprechende biometrische Merkmal von ihm „einlesen“ lassen, welches anschließend mit der Datenbank verglichen wird.

Vorteile	Nachteile
keine Karte mit sensiblen Daten kann verloren gehen	zwingend sichere Verbindung zwischen Client und Server notwendig
Manipulation kann fast vollständig ausgeschlossen werden	sehr datenschutzkritisch
aktuelle Jahreskarte kann genutzt werden	Hochsicherheitsserver notwendig

Tabelle XI
VOR- UND NACHTEILE DER SPEICHERUNG DER BIOMETRISCHEN DATEN IN EINER DATENBANK

LITERATUR

- [1] „Skizze einer Kooperation HWR Berlin und Tierpark,“ nicht öffentlich zugängliches Dokument, April 2017.
- [2] S. Zimmermann, „Projektmanagement und Qualitätssicherung,“ Präsentation, 2017.
- [3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, „Manifest für Agile Softwareentwicklung,“ Webseite, 2001, <http://agilemanifesto.org/iso/de/manifesto.html>.
- [4] M. J. Sutherland and M. K. Schwaber, *Der Scrum Guide*, Scrum.Org and ScrumInc., Juli 2016.
- [5] M. Standfest and M. Prohl, „Retrospektiven & Lessons Learned,“ Präsentation, 2017.
- [6] N. Litzel, „Was ist NoSQL?,“ Webseite, Juni 2017, <https://www.bigdata-insider.de/was-ist-nosql-a-615718/>.

ABBILDUNGSVERZEICHNIS

1	Darstellung der eingesetzten Architektur	3
2	Veranschaulichung der „Karten mit ein- und auschecken“-Lösung	6
3	Veranschaulichung der „Biometrie“-Lösungen (VII-B Gesichtserkennung, VII-C Handerkennung und VII-D Fingerabdruck) . . .	7
4	Veranschaulichung der „Personalausweis mit Nummer“-Lösung	7
5	Veranschaulichung der „App“-Lösung	7
6	Veranschaulichung der „Zeitkarten“-Lösung . . .	8

TABELLENVERZEICHNIS

I	Beispielhafte Tabelle einer Retrospektive im Plus-Delta-Vorgehen	3
II	Vor- und Nachteile der „Karten mit ein- und auschecken“-Lösung	6
III	Vor- und Nachteile der Gesichtserkennungs-Lösung	6
IV	Vor- und Nachteile der Handerkennungs-Lösung	6

V	Vor- und Nachteile der Fingerabdruck-Lösung . .	6
VI	Vor- und Nachteile der „Personalausweis mit Nummer“-Lösung	7
VII	Vor- und Nachteile der „Personalausweis mit Fingerabdruck“-Lösung	7
VIII	Vor- und Nachteile der App-Lösung	7
IX	Vor- und Nachteile der Zeitkarten-Lösung	8
X	Vor- und Nachteile der Speicherung der biome- trischen Daten auf einer Karte	8
XI	Vor- und Nachteile der Speicherung der biome- trischen Daten in einer Datenbank	8