

Konzeptbeschreibung

Team: TimeFlippers/NR. 2

Mitglied 1: Simon Draxl, 11830761

Mitglied 2: Calvin Gehrler, 11832080

Mitglied 3: Florian Gerhold, 11830663

Mitglied 4: Anna-Lena Hetzenauer, 11816161

Mitglied 5: Isabella Schmut, 11832184

Proseminargruppe: 2

Datum: 27.5.2020

1. Systemüberblick

Bei dieser IoT-basierten Software handelt es sich um eine webbasierte Anwendung für die Analyse der Produktivität von Mitarbeitern in der Softwareentwicklung. Sie eignet sich für Unternehmen, welche ihre Leistungen auf eine kreative Weise steigern wollen. Die Produktivität wird mit Hilfe von TimeFlip Würfeln gemessen, von denen jeder Mitarbeiter einen bekommt.

Die Würfel bieten einen unkomplizierten Weg, die Aktivitäten aufzuzeichnen. Für die Mitarbeiter entsteht kein zusätzlicher Aufwand, da sie nur den Würfel auf eine der zwölf darauf abgebildeten Tätigkeiten drehen müssen.

Bei einer Unachtsamkeit kann der Mitarbeiter die falsch erfassten Daten noch bis zu 2 Wochen mühelos selbst korrigieren. Spätere Änderungen müssen von Teamleiter bzw. Abteilungsleiter genehmigt werden. Wenn an anderer Stelle Veränderungen oder Korrekturen notwendig sind, ist man durch ein Audit-Log in der Lage, besondere Umstände zu erklären. Das Audit-Log ist aber unveränderlich.

Die aufgezeichneten Daten werden dann automatisch ins zentrale Backend geladen und von selbst zusammengefügt.

Die Daten können von Teamleitern oder Abteilungsleitern in einer übersichtlichen Webanwendung ausgelesen werden. Die Daten liegen sowohl tabellarisch, als auch in aufschlussreichen Diagrammen und Visualisierungen vor. Um Dispute innerhalb eines Teams zu verhindern und um Mitarbeiter zu schützen, sieht der Teamleiter die Daten des gesamten Teams und nicht die, der einzelnen Teammitglieder. Dasselbe gilt für Abteilungen.

In der Webanwendung haben Mitarbeiter die Möglichkeit, auf eine einfache Art und Weise ihre Urlaubstage anzugeben. Dadurch wirkt sich der Urlaub auch nicht negativ auf die Produktivität ihres Teams aus.

Wenn die Benutzer wollen, können sie sich für E-Mail-Benachrichtigungen anmelden. Diese liefern regelmäßig personalisierte Produktivitätsauswertungen, mit denen man sich selbst motivieren oder Verbesserungspotenzial finden kann.

Zudem gibt es noch ein Dashboard, welches jeder Benutzer einsehen kann. Besondere Leistungen oder eine hohe Produktivität werden dort mit diversen Badges belohnt, um die Motivation der Mitarbeiter zu steigern.

2. Use Cases

2.1 Akteure

Mitarbeiter:

Mitarbeiter sind Anwender des Systems und bekommen für die Produktivitätsanalyse jeweils einen TimeFlip Würfel zur Verfügung gestellt. Sie haben Zugriff auf persönliche Daten und Statistiken und können weitere beruflich-administrative Aktionen tätigen.

Teamleiter:

Teamleiter haben zusätzlich Einsicht in Daten und Statistiken des gesamten Teams.

Abteilungsleiter/Manager:

Abteilungsleiter und höheres Management sind zusätzlich berechtigt, Daten und Statistiken der gesamten Abteilung abzurufen.

Admin:

Admins sind zuständig für die Verwaltung aller Benutzer des Systems sowie für die Einrichtung der einzelnen TimeFlip Würfel und Minirechner.

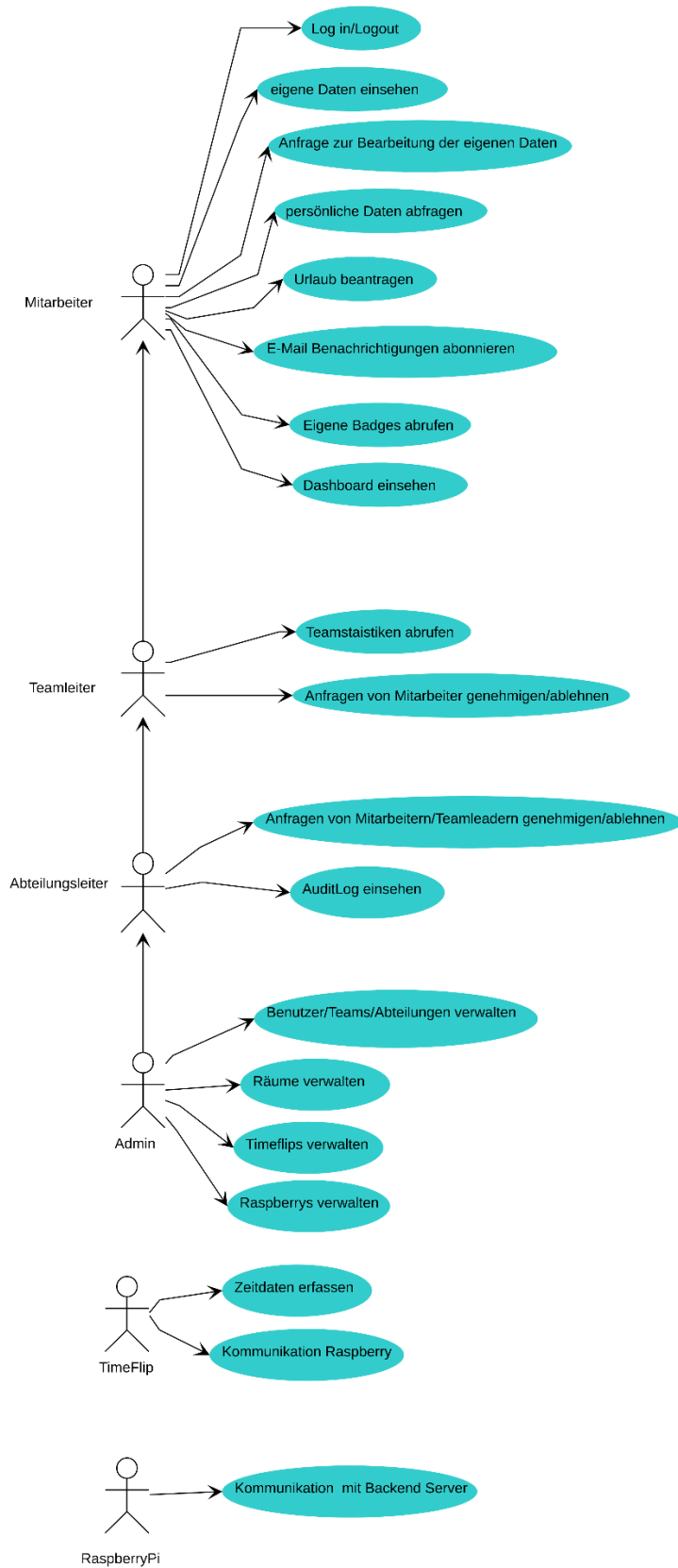
TimeFlip Würfel:

Für die Aufzeichnung der Zeitdaten werden TimeFlip Würfel verwendet. Diese müssen immer so gedreht und platziert werden, dass die aktuelle Tätigkeit nach oben zeigt. Die Aufzeichnungen werden gespeichert und an den zugehörigen Minirechner übertragen.

Minirechner (Raspberry Pi):

Zeitaufzeichnungen der Mitarbeiter werden von den Minirechnern angefragt und letztendlich an den zentralen Backend Server weitergeleitet.

2.2 Use Case Diagramm



2.3 Use Cases

2.3.1 Mitarbeiter

Anwendungsfall: **Login/Logout**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter verfügt über einen Benutzernamen und ein Passwort

Basisablauf:

- Mitarbeiter besucht die Website
- Mitarbeiter gibt seinen Benutzernamen und sein Passwort ein.
- Wenn der Mitarbeiter sich ausloggen will, klickt er auf den Menüpunkt Logout.

Involvierte Klassen: User, Employee, LoginListener, LogoutListener

Anwendungsfall: **Einsicht in persönliche Daten**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt

Basisablauf:

- Mitarbeiter klickt im Menü auf "Profil"
- Profil mit den persönlichen Daten des Mitarbeiters wird angezeigt

Involvierte Klassen: User, Employee

Anwendungsfall: **Einsicht in eigene Statistiken**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt

Basisablauf:

- Mitarbeiter klickt im Menü auf "My Statistics"
- Statistiken des eingeloggten Mitarbeiters werden angezeigt
- Man kann einen Tag wählen um sich Statistiken anzusehen.

Involvierte Klassen: User, Employee

Anwendungsfall: **Einsicht in Anfragen**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt

Basisablauf:

- Mitarbeiter klickt im Menü auf "Profile" -> „Profile“
- Unter den Button „Request Overview“ kommt der Mitarbeiter auf die Seite mit seinen Anfragen.
- Der Mitarbeiter sieht seine offenen, gestellten und akzeptierten Anfragen.

Involvierte Klassen: User, Employee

Anwendungsfall: **Erfasste Daten korrigieren**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt und Zeitaufzeichnungen sind verfügbar

Basisablauf:

- Mitarbeiter klickt im Menü auf “Meine Statistiken”
- Statistiken werden angezeigt
- Mitarbeiter klickt auf “Meine Zeitaufzeichnungen”
- Zeitaufzeichnungen werden in einer Tabelle angezeigt
- Wenn der Mitarbeiter etwas verändern will, muss er zuerst eine Anfrage an den Team und Abteilungsleiter stellen. Nachdem dieser die Anfrage genehmigt hat, darf der Mitarbeiter die Zeitaufzeichnungen von diesem Tag verändern.

Involvierte Klassen: User, Employee

Anwendungsfall: **Urlaub im System beantragen**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt und es stehen noch Urlaubstage zur Verfügung

Basisablauf:

- Mitarbeiter klickt im Menü auf “Profil”
- Profil wird angezeigt
- Mitarbeiter klickt bei Kalender auf “Urlaub eintragen”
- Ein Pop-Up öffnet sich und die Urlaubsdaten können eingetragen werden
- Falls die ungültige Daten eingetragen oder zu viele Urlaubstage beantragt werden, erscheint eine Fehlermeldung.
- Es wird eine Anfrage an den Abteilungsleiter und Teamleiter geschickt. Diese können die Anfrage entweder bestätigen oder ablehnen. Nachdem der bestätigt wurde, wird der Urlaub automatisch nachgetragen

Involvierte Klassen: User, Employee, Vacation, VacationService

Anwendungsfall: **E-Mail Benachrichtigungen abonnieren**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt und Benachrichtigungen sind nicht abonniert

Basisablauf:

- Mitarbeiter klickt im Menü auf “Profil”
- Profil wird angezeigt
- Im Profil erscheint eine Unterkategorie “E-Mail Benachrichtigungen”, in der sich eine Checkbox befindet
- Ist die Checkbox aktiviert, erscheint die Option, das Intervall der Benachrichtigungen einzurichten

Involvierte Klassen: User, Employee, Email, EmailService

Anwendungsfall: **eigene Badges abrufen**

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt

Basisablauf:

- Mitarbeiter klickt im Menü auf “Meine Statistiken”
- Statistiken werden angezeigt
- Mitarbeiter klickt auf “Meine Badges”
- Liste der verliehenen Badges erscheint
- Zusätzlich sieht der Mitarbeiter alle verliehenen Badges der letzten Woche auf dem Dashboard

Involvierte Klassen: User, Employee, Badge, BadgeEnum

Anwendungsfall: Produktivitäts-Dashboard abrufen (Überblick Firma)

Initiator: Mitarbeiter

Vorbedingung: Mitarbeiter ist eingeloggt

Basisablauf:

- Sobald der Mitarbeiter eingeloggt ist, erscheint auf der Startseite der Anwendung eine Ansicht mit von allen verliehenen Badges der letzten Woche, seine Statistiken und einen Kalender.

Involvierte Klassen: User, Employee

2.3.2 Teamleiter

Anwendungsfall: Team Statistiken abfragen

Initiator: Teamleiter

Vorbedingung: Teamleiter ist mit der Berechtigung "Teamleiter" oder höher eingeloggt

Basisablauf:

- Teamleiter klickt im Menü auf "Team Statistiken"
- Team Statistiken werden angezeigt

Involvierte Klassen: User, TeamLeader, Team

Anwendungsfall: Anfragen bestätigen

Initiator: Teamleiter

Vorbedingung: Teamleiter kann die Anfragen die ein Mitarbeiter geschickt hat bestätigen.

Basisablauf:

- Teamleiter klickt unter „Profile“ auf "Requests"
- Alle gestellten Anfragen werden aufgelistet
- Hier kann der Teamleiter nun Anfragen von Mitarbeitern akzeptieren oder ablehnen.

Involvierte Klassen: User, TeamLeader, Employee

2.3.3 Abteilungsleiter/Manager

Anwendungsfall: Abteilung Statistiken abfragen

Initiator: Abteilungsleiter

Vorbedingung: Abteilungsleiter/Manager ist mit der Berechtigung "Abteilungsleiter" oder "Manager" eingeloggt

Basisablauf:

- Abteilungsleiter klickt im Menü auf "Abteilungs Statistiken"
- Abteilungs Statistiken werden angezeigt

Involvierte Klassen: User, DepartmentLeader, Manager, Department

Anwendungsfall: Abteilung Badges abfragen

Initiator: Abteilungsleiter

Vorbedingung: Abteilungsleiter ist mit der Berechtigung "Abteilungsleiter" eingeloggt

Basisablauf:

- Abteilungsleiter klickt im Menü auf “Badges”
- Eine Auflistung aller verliehenen Badges mit Mitarbeitern wird angezeigt.
- Die Ansicht ist nicht zeitlich beschränkt.

Involvierte Klassen: User, DepartmentLeader, Manager, Department, Badges, BadgeEnum

Anwendungsfall: **Anfragen bestätigen**

Initiator: Teamleiter

Vorbedingung: Teamleiter kann die Anfragen, die ein Mitarbeiter oder Teamleiter geschickt hat, bestätigen.

Basisablauf:

- Abteilungsleiter klickt unter „Profile“ auf “Requests”
- Alle gestellten Anfragen werden aufgelistet
- Hier kann der Abteilungsleiter nun Anfragen von Mitarbeitern und Teamleitern akzeptieren oder ablehnen.

Involvierte Klassen: User, DepartmentLeader, TeamLeader, Employee

Anwendungsfall: **Audit-Log**

Initiator: Abteilungsleiter

Das Audit-Log kann nur von den Abteilungsleitern eingesehen werden, andere Mitarbeiter haben keinen Zugriff darauf.

Basisablauf:

- Es wird aufgezeichnet, wenn ein User auf der Website etwas verändert, wie zum Beispiel sich einloggt, einen Urlaub beantragt u.ä.
- Die Aufzeichnung kann dann eingesehen werden und bei Konfliktsituationen Klarheit verschaffen.

Involvierte Klassen: DepartmentLeader, LogEntry, LogEnum, AuditLogger, Logger

2.3.4 Admin

Anwendungsfall: **Benutzer verwalten**

Initiator: Admin

Vorbedingung: Admin ist mit der Berechtigung “Admin” eingeloggt

Basisablauf:

- Admin klickt im Menü auf “Benutzer”
- Liste aller registrierten Systembenutzer wird ausgegeben
- Admin kann neue Benutzer erstellen, bestehende Benutzer verwalten und bei Bedarf auch löschen

Involvierte Klassen: User, Admin

Anwendungsfall: **Teams verwalten**

Initiator: Admin

Vorbedingung: Admin ist mit der Berechtigung “Admin” eingeloggt

Basisablauf:

- Admin klickt im Menü auf “Teams”
- Liste aller Teams wird ausgegeben
- Admin kann neue Teams erstellen, bestehende Teams und dessen Mitglieder verwalten und bei Bedarf auch löschen

Involvierte Klassen: User, Admin, Team

Anwendungsfall: **Abteilungen verwalten**

Initiator: Admin

Vorbedingung: Admin ist mit der Berechtigung “Admin” eingeloggt

Basisablauf:

- Admin klickt im Menü auf “Departments”
- Liste Abteilungen wird ausgegeben
- Admin kann neue Abteilungen erstellen, bestehende Abteilungen mit dazugehörigen Teams verwalten und bei Bedarf auch löschen

Involvierte Klassen: User, Admin, Department

Anwendungsfall: **TimeFlip Würfel einrichten**

Initiator: Admin

Vorbedingung: Admin ist mit der Berechtigung Admin eingeloggt

Basisablauf:

- Admin klickt im Menü auf “TimeFlip Würfel”
- Liste aller aktiven TimeFlip Würfel wird ausgegeben
- Admin kann dem System nun neue TimeFlip Würfel hinzufügen/löschen
- Admin kann im gleichen Kontext Mitarbeitern TimeFlip Würfel zuweisen und die Kalibrierung der Würfel vornehmen
- Der TimeFlip Würfel speichert alle Daten in einem internen Speicher. Unter normalen Bedingungen liest der Raspberry die Daten sofort aus diesem Speicher aus. Bei einem Ausfall des Raspberrys oder anderen Unterbrechungen (TimeFlip außer Reichweite etc.) werden die Daten weiterhin in den Speicher geschrieben und vom Raspberry ausgelesen sobald der Würfel sich wieder mit dem Raspberry verbindet. Da der Speicher nur Daten für drei Tage hat, sollte er sich innerhalb von 3 Tagen mit einem Raspberry verbinden.

Involvierte Klassen: Admin, User, TimeFlip, Task TaskEnum

Anwendungsfall: **Minirechner einrichten**

Initiator: Admin

Vorbedingung: Admin ist mit der Berechtigung Admin eingeloggt

Basisablauf:

- Admin klickt im Menü auf “Minirechner”
- Liste aller aktiven Minirechner wird ausgegeben
- Admin kann dem System nun neue Minirechner hinzufügen/löschen, sowie deren Raumzuteilung und Justierung der Übertragungsparameter zum zentralen Backend einstellen

Involvierte Klassen: Admin, Server, RaspberryPi

Anwendungsfall: **Räume einrichten** Initiator:

Admin

Vorbedingung: Admin ist mit der Berechtigung Admin eingeloggt

Basisablauf:

- Admin klickt im Menü auf “Rooms”
- Liste aller bestehenden Räume wird ausgegeben
- Admin kann dem System nun neue Räume hinzufügen/löschen

Involvierte Klassen: Admin, Server, RaspberryPi

2.3.5 TimeFlip Würfel

Anwendungsfall: **Zeitdaten erfassen**

Initiator: Mitarbeiter, TimeFlip Würfel

Vorbedingung: TimeFlip Würfel ist aktiviert und einem Mitarbeiter zugeordnet

Basisablauf:

- Mitarbeiter dreht den Würfel, sodass die aktuell ausgeführte Tätigkeit nach oben zeigt
- Zeitdaten werden aufgenommen und im Onboard-Speicher gesichert

Involvierte Klassen: User, Mitarbeiter, TimeFlip, Task

Anwendungsfall: **Kommunikation mit Minirechner**

Initiator: TimeFlip Würfel

Vorbedingung: Zeitdaten wurden erfasst

Basisablauf:

- Zu einem festgelegten Zeitpunkt stellt der TimeFlip Würfel automatisch eine Verbindung zum Minirechner des gleichen Raumes her
- Die Kommunikation erfolgt mittels Bluetooth LE
- Wird eine Verbindung erfolgreich hergestellt, so können die Zeitaufzeichnungen der TimeFlip Würfel an den Minirechner übertragen werden

Involvierte Klassen: TimeFLip, RaspberryPi

2.3.6 Minirechner (RaspberryPi)

Anwendungsfall: **Kommunikation mit Backend Server**

Initiator: Minirechner (Raspberry Pi)

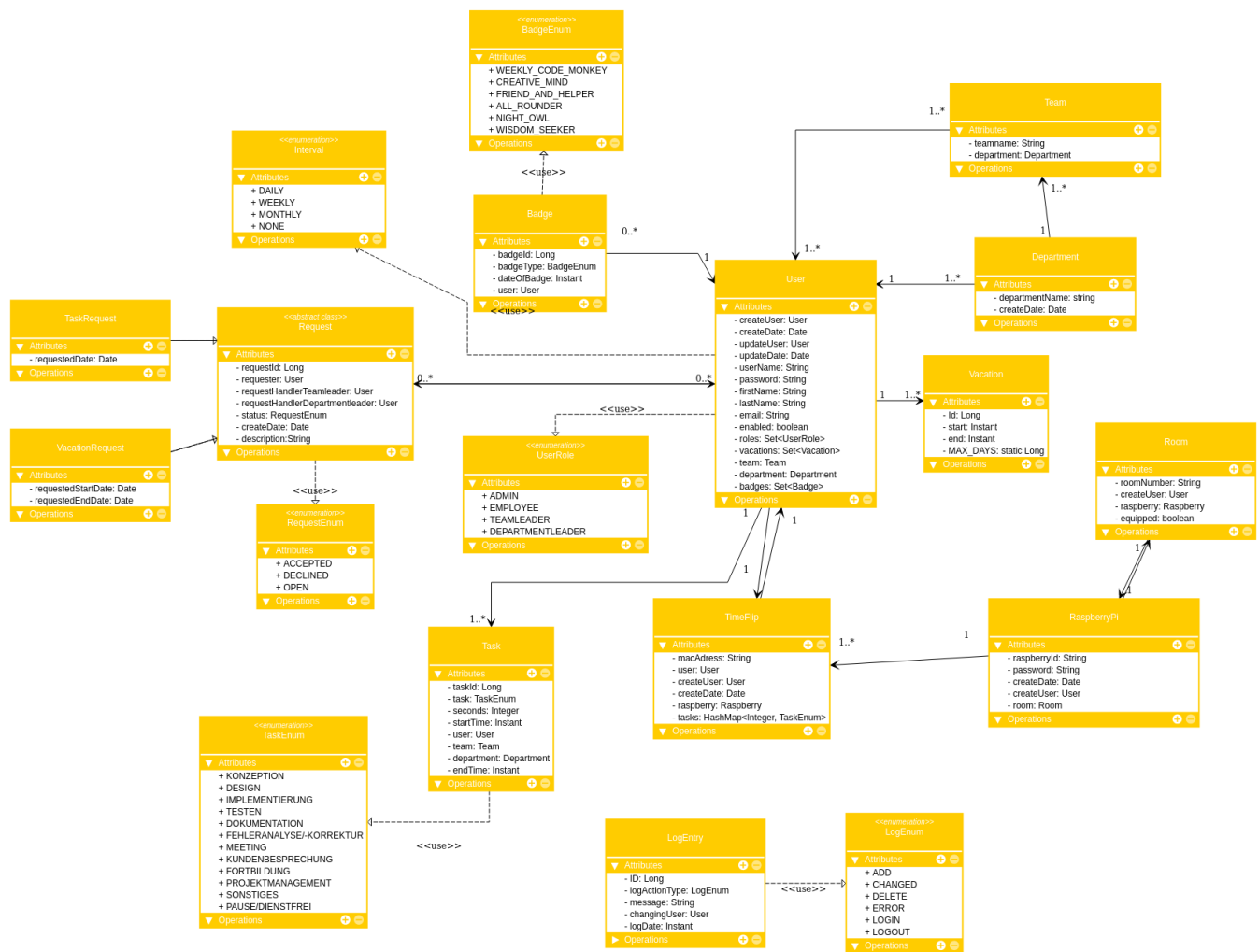
Vorbedingung: Zeitdaten wurden erfolgreich vom TimeFlip Würfel auf den Minirechner übertragen

Basisablauf:

- Raspberry Pi nimmt Verbindung zum zentralen Backend Server auf
- Verbindung erfolgt über eine Webschnittstelle
- Wird die Verbindung erfolgreich aufgebaut, so können die Daten aller TimeFlip Würfel gesammelt an das zentrale Backend übermittelt werden

Involvierte Klassen: RaspberryPi, Server

3. Klassendiagramm



User + Roles: Der User kann eine oder mehrere Rollen haben. Jeder User besitzt einen TimeFlip und kann seine eigenen Daten und Statistiken, die daraus berechnet werden, einsehen. Je nach Rolle hat der User verschiedene Sichten auf die Daten und Statistiken anderer Mitarbeiter. Employees sind einem Team zugeteilt, von dem jeder einen Teamleiter hat. Jeder User ist ebenfalls einem Department zugeteilt, von dem jeder Department einen Head of Department besitzt.

E-Mail: Jeder User besitzt eine E-Mail-Adresse. Wenn sich ein User für den E-Mailservice einschreibt, kann er festlegen in welchem Intervall er E-Mails erhält.

Vacation: Weiters kann sich jeder User einen Urlaub beantragen, wobei er eine fixe Anzahl von Urlaubstagen nicht überschreiten darf.

TimeFlip: Der Time Flip jedes Users, besitzt eine Liste von Aufgaben (Tasks) des Users, die über ein fest Enum definiert wird. Diese Aufgaben werden in einer Liste im Time Flip gespeichert und die gesammelten Daten werden an den Raspberry Pi gesendet.

RaspberryPi: Pro Raum sammelt ein Raspberry Pi alle Daten von allen Time Flips, die sich in diesem Raum befinden. Der Raspberry Pi sendet diese Daten dann weiter zum zentralen Server

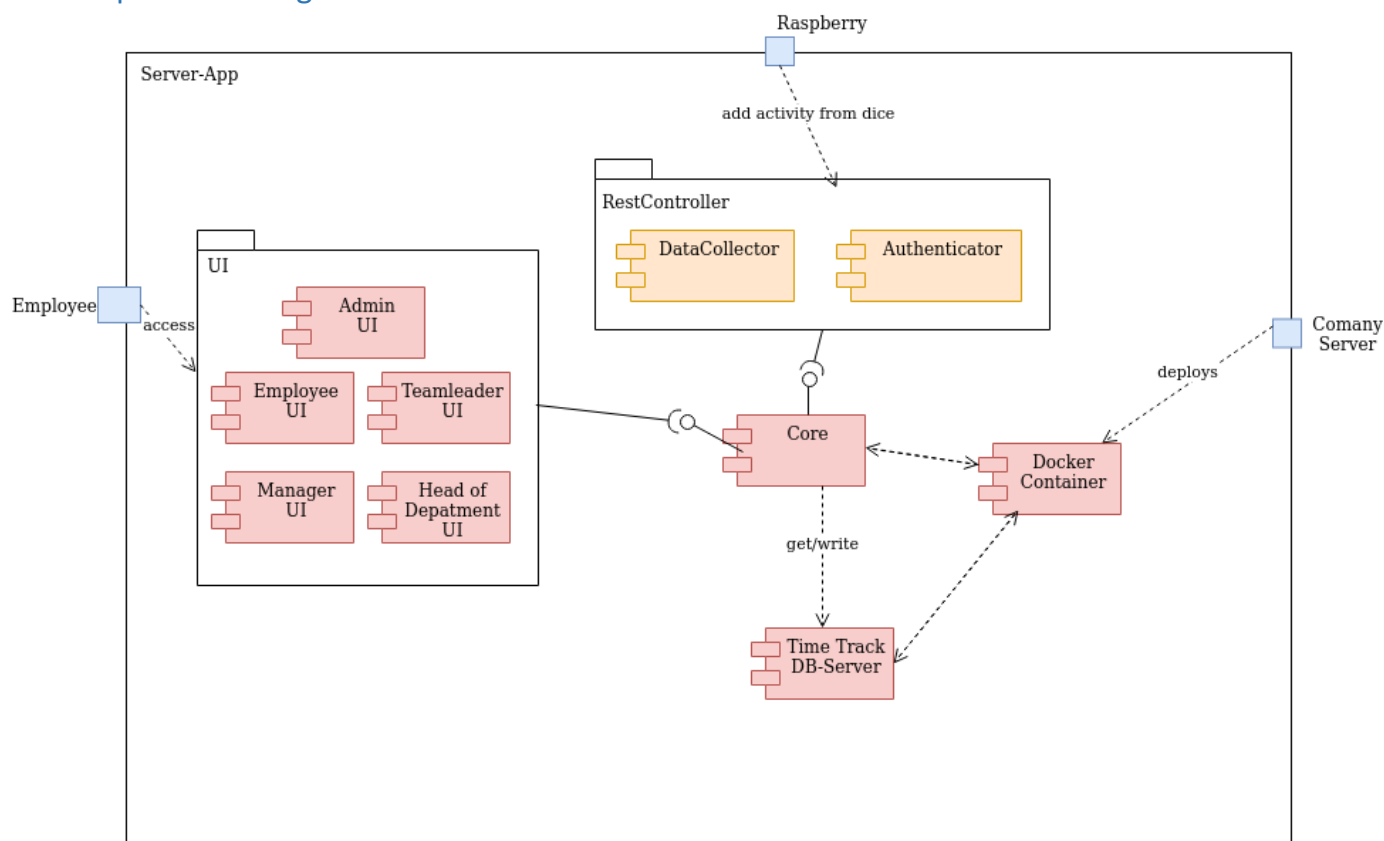
auf dem Auch die restlichen Daten der Angestellten des Unternehmens gespeichert sind.

Badges: Wenn die Auswertung eines Time Flip Würfels ergibt, dass ein User sich in einem Gebiet besonders hervorgetan hat, erhält er dafür Auszeichnungen, sogenannte Badges.

Log-Entry: das Audit-Log zeichnet alle Tätigkeiten des User auf und speichert diese in die Datenbank.

4. SW-Architektur

4.1 Komponentendiagramm



4.2 Technologien

4.2.1 Java

Java ist eine objektorientierte, systemunabhängige Programmiersprache und Laufzeitumgebung. Da sie sehr schnell, sicher und zuverlässig ist, findet man sie von Spielkonsolen bis zu Internet in den verschiedensten Anwendungen.

Wir verwenden Java für die Implementierung der Klassen.

4.2.2 Spring

Spring ist ein quelloffenes Framework für die Java Plattform und wird oft für Web-Anwendungen verwendet. Spring soll vor allem die Entwicklung mit Java vereinfachen und gute Programmierpraktiken fördern.

Wir haben Spring verwendet um unsere Webapplikation aufzubauen.

4.2.3 Java Server Faces

JSF ist ein Framework-Standard zur Entwicklung von grafischen Benutzeroberflächen für Webanwendungen mit Java. Mit JSF kann man einfach Komponenten für Benutzerschnittstellen in Webseiten einbinden und die Navigation definieren.

JSF findet man in unserer Webapplikation für die Gestaltung der GUI um diese für den Mitarbeiter benutzerfreundlicher zu machen.

4.2.4 Prime Faces

Prime Faces ist eine Open-Source-Komponentenbibliothek für Benutzeroberflächen für JSF-basierte Anwendungen.

PrimeFaces findet man in unserer Webapplikation für die Gestaltung der GUI um diese für den Mitarbeiter benutzerfreundlicher zu machen.

4.2.5 Maven

Maven ist ein Build-Management-Tool, das auf Java basiert. Mit ihm werden vor allem Java-Programme standardisiert erstellt und verwaltet um die eigene Konfiguration zu minimieren.

4.2.6 MySQL

MySQL ist eine quelloffene relationale Datenbank. Das Programm wird häufig für die Datenspeicherung für Webservices verwendet. Der grundsätzliche Aufbau des Speichersystems sieht einen Server vor, dem die externen Clients wiederum Anfragen schicken. Nahezu alle Betriebssysteme sind dafür kompatibel.

Wir verwenden diese Datenbank um die Daten unserer Applikation zu speichern und zu verwenden.

4.2.7 Tiny B

Eine moderne und Bluetooth LE API, die für das Vernetzen mit Hilfe von verschiedensten Programmiersprachen wie C oder Java verwendet wird.

Findet bei der Kommunikation zwischen Würfel und Webapplikation Verwendung.

4.2.8 JUnit

JUnit ist ein Framework, das zum Testen von Java Programmen eingesetzt wird. Meist werden nur einzelne Klassen oder Methoden (daher das Unit im Namen) getestet. Ein solcher Test kennt nur zwei Ausgänge, ist er grün, gelingt er, ansonsten ist er rot.

JUnit haben wir dafür verwendet um unsere Webapplikation zu testen.

4.2.9 Raspberry Pi

Ein Einplatinencomputer, der aus einem SoC (System of a Chip) und einer ARM GPU besteht. Meist wird er mit einem Linux Betriebssystem, dem Raspian betrieben.

Wir verwenden den Raspberry als Kommunikationsschnittstelle zwischen der WebApplikation und dem Timeflip Würfel.

4.2.10 Docker & Docker Compose







Docker ist eine Freie Software zur Isolierung von Anwendungen mit Containervirtualisierung. Docker vereinfacht die Bereitstellung von Anwendungen, weil sich Container, die alle nötigen Pakete enthalten, leicht als Dateien transportieren und installieren lassen.

Docker Compose ist ein Tool um mehrere Docker Applikationen zu definieren und lauffähig zu machen.

Wir verwenden Docker um unsere Webapplikation für andere lauffähig zu machen.

5. GUI

Congratulations to our winners of last week!

Miguel Herran	Jen Barber	Laurie Bream	Erlich Bachman	Monica Hall	Nelson Bighetti
					


Profile

Username:

First Name:

Last Name:

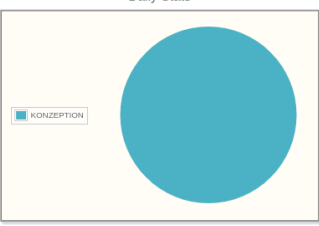
E-Mail:

Interval for mails: 

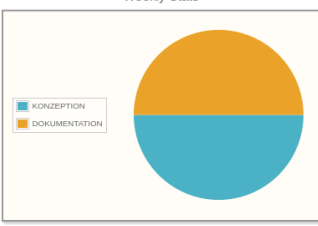
Password:

My Statistics

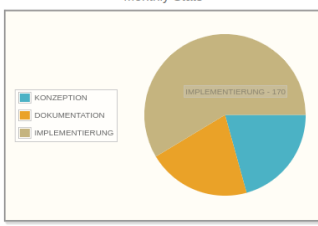
Daily Stats



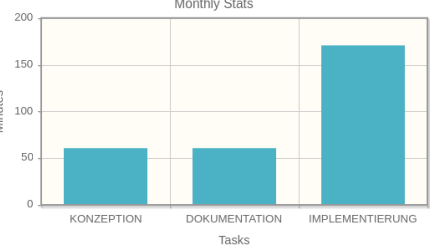
Weekly Stats



Monthly Stats

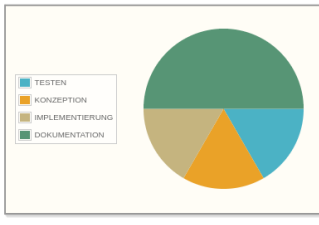


Monthly Stats

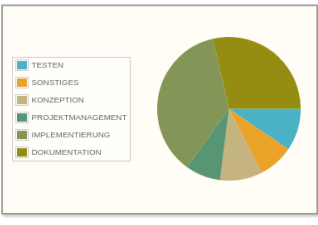


Statistics of Team Network & Security

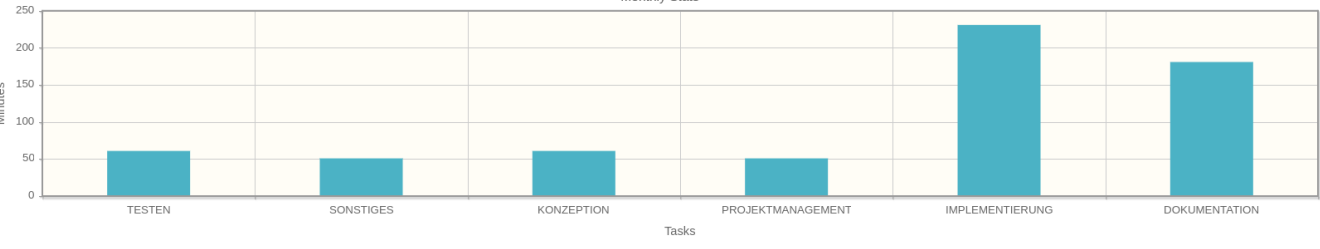
Weekly Stats

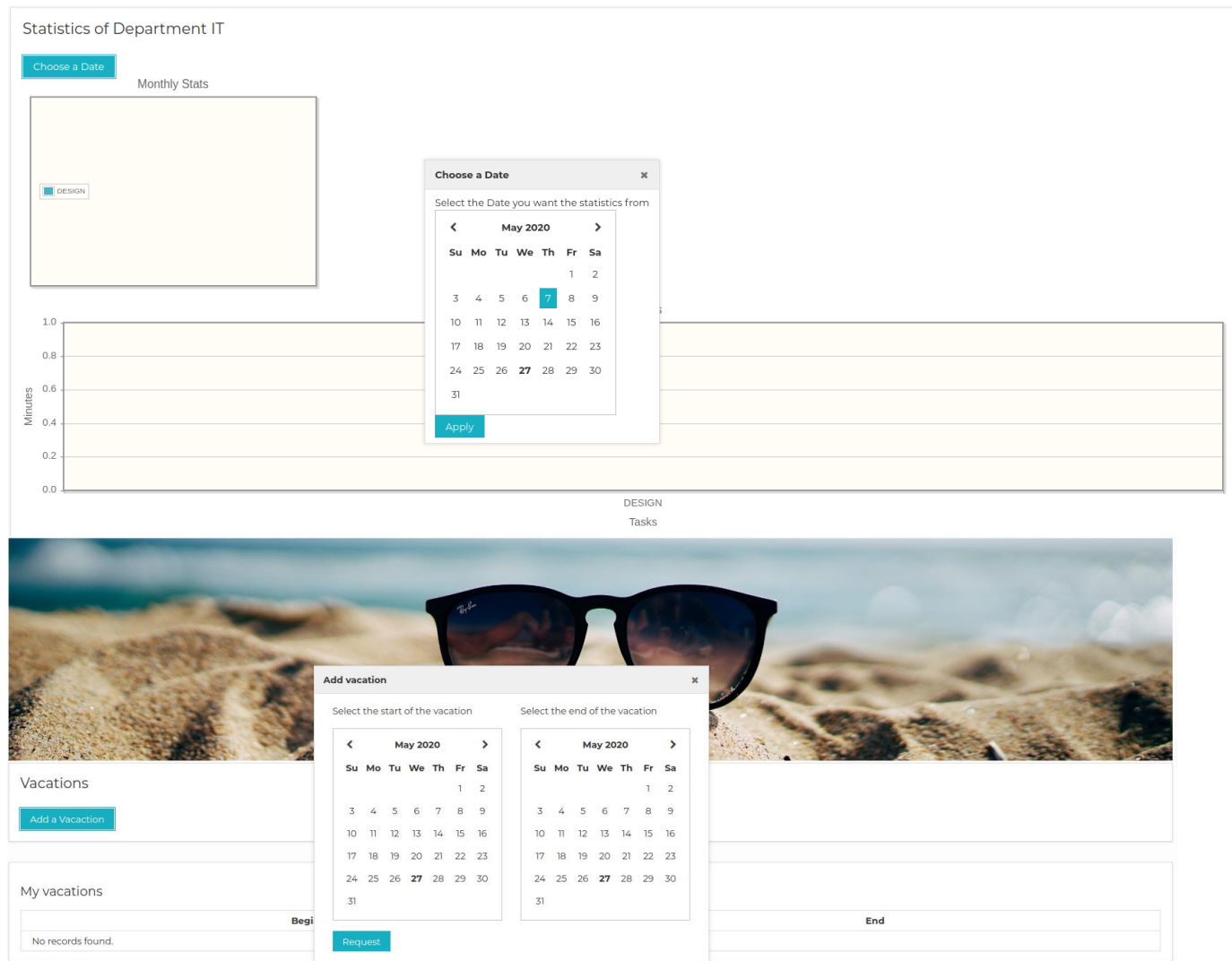


Monthly Stats



Monthly Stats





6. Projektplan

6.1. Milestones:

- 1) **Backend**; Start: 19.März – Ende: 21. Mai
- 2) **Timeflip and Raspberry Pi**; Start: 6. April – Ende: 21. Mai
- 3) **Web Applikation**; Start: 7.Mai – Ende: 28. Mai
- 4) **Testing**; Start: 24.Mai – Ende: 4.Juni
- 5) **Finalisation**; Start: 4.Juni – Ende: 19.Juni

6.2. Issues:

Die Aufgaben wurden mittels Git Issues den Personen zugewiesen und wie unten im Zeitplan eingetragen. Simon Draxl hat die Implementierung des Timeflips übernommen. Die restlichen Aufgaben wurden zu gleichen Teilen auf die anderen Teammitglieder aufgeteilt.

