

# **HAW Hamburg**

## **2. Semester 17/18**

## **Media Systems (B.Sc.)**

Fett, Niklas 234 3476

August, Thore 233 2999

Vo, Michael 234 3599

Zander, Felix 192 5853

[thore.august@haw-hamburg.de](mailto:thore.august@haw-hamburg.de)

[illegible]

Interface
Planung
Präsentation
Datenbank



# Pflichtenheft: Time Tree

HAW Hamburg      Media Systems (B.Sc.) 2. Semester 17/18

Angewandtes Programmieren und Projektmanagement

Fett, Niklas	234 3476
August, Thore	233 2999
Vo, Michael	234 3599
Zander, Felix	192 5853

Winter Dezember 2017

# **Inhaltsverzeichnis**

<b>1 Zielbestimmung</b>	<b>S.1</b>
<b>1.1 Lastenheft</b>	<b>S.1</b>
1.1.1 <u>Musskriterien</u>	S.1
1.1.2 <u>Wunschkriterien</u>	S.1
1.1.3 <u>Abgrenzungskriterien</u>	S.1
<b>2 Produkteinsatz</b>	<b>S.2</b>
2.1 <u>Anwendungsbereich</u>	S.2
2.2 <u>Zielgruppe</u>	S.2
2.3 <u>Betriebsbedingung</u>	S.3
<b>3 Produktübersicht</b>	<b>S.3</b>
<b>4 Produktfunktion</b>	<b>S.4</b>
4.1 <u>Login-Logout</u>	S.4
4.2 <u>Anmelden</u>	S.5
4.2 <u>Nutzerfunktion</u>	S.6
<b>5 Produktdaten</b>	<b>S.8</b>
5.1 <u>Benutzerdaten</u>	S.8
5.2 <u>Projektdaten</u>	S.8
<b>6 Benutzeroberfläche</b>	<b>S.9</b>
<b>7 Nichtfunktionale Anforderungen</b>	<b>S.10</b>
<b>8 Technische Produktumgebung</b>	<b>S.10</b>
8.1 <u>Software</u>	S.10

# **1 Zielbestimmung**

## **1.1 Lastenheft**

### 1.1.1 Musskriterien

Für den Kurs Angewandtes Programmieren wurde die Aufgabe gegeben bis zum 1. Dezember 2017 eine funktionierende und praktikable Webapplikation zu entwerfen. Sie sollte auf einem leichten Framework basieren wie z.B. Node.js oder Express.js. Ebenfalls soll die Webapplikation Elemente von HTML und EJS enthalten. Sie sollte über eine Datenbank und Benutzerverwaltung verfügen. Für die Datenbanken standen TingoDB, MongoDB und MySQL

- Node.js oder Express.js als Framework
  - keine schwergewichtigen Frameworks (Wordpress, Ionic, usw.)
- mit Datenbank
  - MongoDB, TingoDB oder MySQL
- mit Benutzerverwaltung
  - Login etc.
- 120 Stunden Workload für jede Person inklusive Recherche

### 1.1.2 Wunschkriterien

TingoDB sollte verwendet werden, da die Webadministration der HAW MongoDB noch nicht freigeschaltet hatte zu Beginn des Projekts.

### 1.1.3 Abgrenzungskriterien

Für Fortgeschrittene Programmierer gibt es die Möglichkeit MySQL zu verwenden und die Webapplikation muss nicht online verfügbar sein.

## **2 Produkteinsatz**

Für unser Projekt in Angewandtes Programmieren wollten wir eine Applikation entwickeln die, passend zur derzeitigen Lage des Arbeitsmarktes, Zeiten erfasst. Mit steigenden Zahlen von Freelancer, Homeoffice-Angestellten und Zeitarbeitern im 20 Stundentarif hielten wir dies für angebracht.

Laut dem Arbeitszeitgesetz stehen jedem Mitarbeiter ab 6 Stunden Arbeit 30min Pause zu, bei 9 sind es 45min und hinzu kommen kleiner Pausen zu wie z.B. die laut der Bildschirmarbeitsverordnung (BildscharbV) vorgesehenen 10 min Augenpause bei Bildschirmarbeit. Oft sind diese Pausen jedoch für Firmen redundant, da Auftraggeber für Projekte jeweils nur die Gesamtarbeitszeit in Rücksicht nehmen.

In unserem Modell haben wir also eine Firma die Projekte und Aufträge von anderen Auftraggebern annimmt. Wir haben also reguläre gesetzlichen Arbeitszeit nach dem [Arbeitszeitgesetz - ArbZG](#) und tatsächliche Arbeitszeit, in den Pausen etc. nicht berücksichtigt werden. (Somit würde z.B. die tatsächliche Arbeitszeit einer 5 mal 4 Stunden Person anders aussehen als eine Person die 8 mal 8 Stunden und 1 mal 4 Stunden arbeitet)

Der Auftraggeber zahlt zwar für die gesetzlich geleistete Arbeitszeit, könnte aber Interesse haben an der tatsächlich geleisteten Arbeitszeit. Auch intern würde die Projektleitung sowie Firmenleitung oft gerne wissen wie die tatsächliche Arbeitszeit aussieht. So könnte sie z.B. besser regelmäßige Fachgespräche führen und diese anders einordnen oder Fachgruppentreffen organisieren um Homeoffice-Mitarbeiter auf neue Stände zu bringen.

Wir haben uns ebenfalls gedacht um eine Übersicht zu behalten bei mehreren parallel laufenden Aufträgen leichter zu überschauen, sollte unsere Applikation dies berücksichtigen.

### **2.1 Anwendungsbereich**

Der Anwendungsbereich umfasst alle Berufsbereiche des Lebens. Die Applikation soll in der Lage sein von überall wo es Internet gibt, abrufbar zu sein um Zeiten einzutragen.

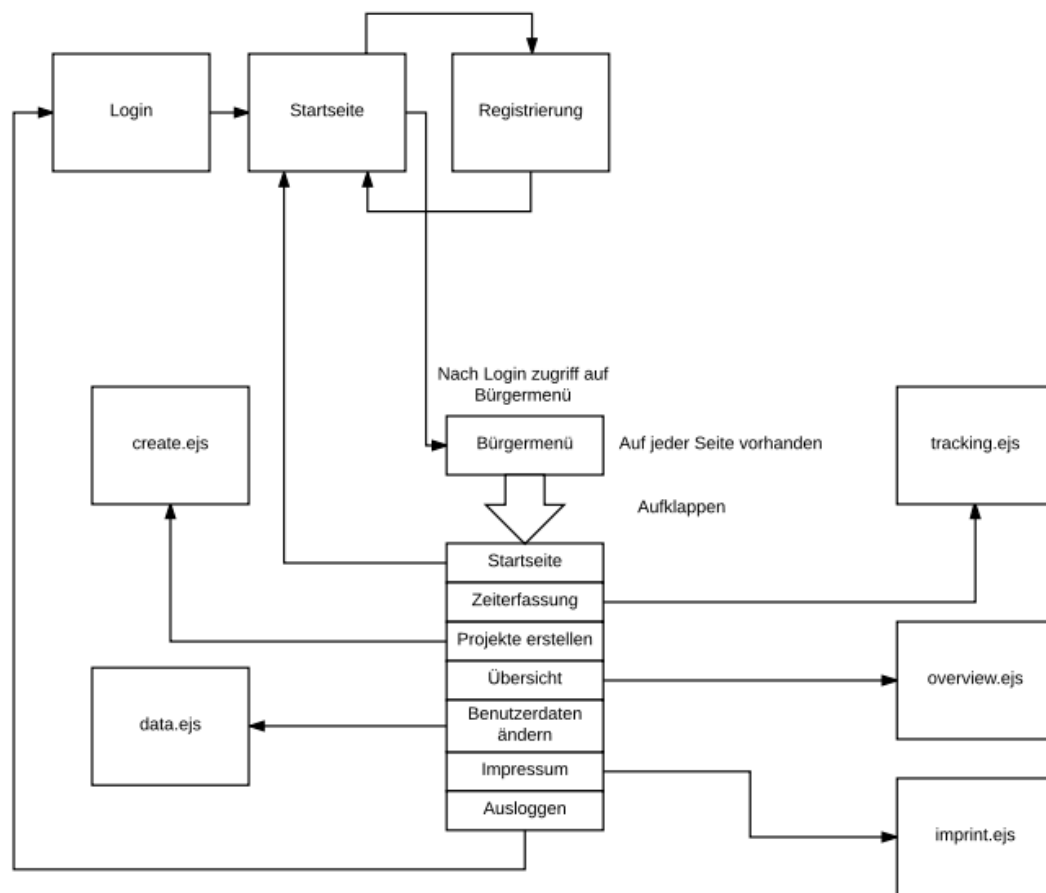
### **2.2 Zielgruppe**

Die Webapplikation soll vorwiegend kleinen Firmen oder Start-Ups helfen, kann jedoch für alle Arbeitsinstitutionen verwendet werden, wenn die nötigen Ressourcen gegeben sind. (Brandbreite, Serverauslastung)

### **2.3 Betriebsbedingung**

Solange ein Server und Internet zur Verfügung steht, kann die Webapplikation von überall aus verwendet werden, der Kunde muss sie lediglich richtig installieren.

### 3 Produktübersicht



## **4 Produktfunktion**

### 4.1 Login-Logout

Geschäftsprozess: Login

Kategorie: primär

Vorbedingungen: Keine

Nachbedingung Erfolg: Nach Login Zugriff zu allen Funktionen der Webapplikation

Nachbedingung Fehlgeschlagen: Fehlermeldung

Akteure: Alle

Auslösendes Ereignis: Jemand will die Applikation nutzen um Arbeitszeiten einzutragen

Beschreibung: Ohne sich einzuloggen sollten Nutzer lediglich in der Lage sein Impressum aufzurufen. Die Zeitlich erfassten Daten der Applikation sollten zur Sicherheit nur für Betriebsmitarbeiter zugänglich sein.

Erweiterung: Keine

Alternativen: Keine

Geschäftsprozess: Logout

Kategorie: primär

Vorbedingungen: Nutzer muss sich einloggen

Nachbedingung Erfolg: Weitergeleitet auf Login Seite ohne Zugriff auf Funktionen und Datenbank

Nachbedingung Fehlgeschlagen: Keine

Akteure: Alle

Auslösendes Ereignis: Keinen weiteren Zugriff auf die Webapplikation um interne Daten vor Nichtangestellten zu schützen

Beschreibung: Nutzer wird abgemeldet und verliert Zugriff zu Funktionen und Daten

Erweiterung: Keine

Alternativen: Keine



## 4.2 Anmelden

Geschäftsprozess: Registrieren

Kategorie: primär

Vorbedingungen: Anmeldeformular ausfüllen

Nachbedingung Erfolg: Man wird zurück auf die Login-Seite gesendet

Nachbedingung Fehlgeschlagen: Fehlermeldung, Formular korrigieren

Akteure: Alle

Auslösendes Ereignis: Zugriff auf den Login

Beschreibung: Ohne sich einzuloggen sollten Nutzer lediglich in der Lage sein Impressum aufzurufen. Die Zeitlich erfassten Daten der Applikation sollten zur Sicherheit nur für Betriebsmitarbeiter zugänglich sein.

Erweiterung: Keine

Alternativen: Keine

#### 4.3 Nutzerfunktion

Geschäftsprozess: Zeiterfassung Saldo

Kategorie: primär

Vorbedingungen: Eingeloggt sein

Nachbedingung Erfolg: Update der Datenbank

Nachbedingung Fehlgeschlagen: Keine

Akteure: Alle

Auslösendes Ereignis: Nutzer möchte Verwaltung mitteilen wann er zum Dienst antritt.

Beschreibung: Die Datenbank addiert die neu eingetragenen Zeit und zeigt daraufhin eine neue Saldo-Anzeige

Erweiterung: Keine

Alternativen: Keine

Geschäftsprozess: Zeiterfassung Projektzeit

Kategorie: primär

Vorbedingungen: Eingeloggt sein

Nachbedingung Erfolg: Update der Datenbank

Nachbedingung Fehlgeschlagen: Keine

Akteure: Alle

Auslösendes Ereignis: Nutzer möchte Verwaltung mitteilen wann er an einer bestimmten Aufgabe gearbeitet hat (Projekt)

Beschreibung: Die Datenbank addiert die neu eingetragenen Zeit

Erweiterung: Keine

Alternativen: Keine

Geschäftsprozess: Zeiterfassung Projektzeit

Kategorie: primär

Vorbedingungen: Eingeloggt sein

Nachbedingung Erfolg: Update der Datenbank

Nachbedingung Fehlgeschlagen: Keine

Akteure: Alle

Auslösendes Ereignis: Nutzer möchte Verwaltung mitteilen wann er an einer bestimmten Aufgabe gearbeitet hat (Projekt)

Beschreibung: Die Datenbank addiert die neu eingetragenen Zeit zu einem vom Nutzer ausgesuchten Projekt

Erweiterung: Keine

Alternativen: Keine

Geschäftsprozess: Projekterstellen

Kategorie: primär

Vorbedingungen: Eingeloggt sein

Nachbedingung Erfolg: Datenbank erhält zusätzlichen Eintrag für Projekt

Nachbedingung Fehlgeschlagen: Keine

Akteure: Alle

Auslösendes Ereignis: Nutzer möchte Projekt erstellen

Beschreibung: Die Datenbank erweitert sich um Projekt

Erweiterung: Keine

Alternativen: Keine

## **5 Produktdaten**

### **5.1 Benutzerdaten**

- Nutzer ID
- Nutzer Kennung
- Passwort
- Arbeitszeit Saldo

### **5.2 Projektdaten**

- Projekt ID
- Projekt Name
- Projektbeschreibung
- Projektteilnehmer
- Projektarbeitszeit

## 6 Benutzeroberfläche



**Time Tree - Aufwandslog**

**Wochenstatistik**

Soll: 11h  
Ist: 1h  
Diff.: -10h

**Projektzeiten**

Datum	Projekt	Zeit	Kommentar
9.9.17	Apvg	2.5h	Meilenstein
12.9.17	MGD2	4h	Konzeption
21.9.17	Apvg	12h	Backend
20.10.17	Apvg	1h	Vorstellung

Startseite  
Zeiterfassung  
Projekt erstellen  
Übersicht  
Benutzerdaten ändern  
Impressum  
Logout

- Bedienung über Web-Interface
- Benutzernavigation mit Maus
- Zentrale Navigation über Brüggermenü, rechts oben
- Fehlermeldungen erklären technische Probleme

## **7 Nichtfunktionale Anforderungen**

Keine

## **8 Technische Produktumgebung**

### 8.1 Software

Windows, MacOS oder Linux mit einem modernen Browser. Spezialisiert auf Windows OS mit Chrome

		Thore August	Niklas Fett	Michael Vo	Felix Zander		
Datum		Frontend, Interface designe, Head Designer	Backend, Head of Code	Projektmanager, Protokolant, Design	Quality Assurance Manager, Debugging, Formatierung		
KW 39	01.10.2017						
	02.10.2017						
	03.10.2017						
	04.10.2017	Erstes Group-Meeting/Kick-Off				2,5h	
	05.10.2017						
KW 40	06.10.2017	Interface Prototyp 2,5h		Nacharbeit Protokoll, Meetingergebnisse 1h			
	07.10.2017	Time Tree Logo 1h					
	08.10.2017						
	09.10.2017						
	10.10.2017						
	11.10.2017						
	12.10.2017	Meilensteinplan Pitch 3h					
	13.10.2017	Group Meeting				1h	
KW 41	14.10.2017	Nacharbeit Protokoll, Meetingergebnisse 1h					
	15.10.2017	MSP Pitch Kontrolle 0,5h					
	16.10.2017	Überarbeitung MSP 1,5h					
	17.10.2017					2h	Ausserplanmässiges Group Meeting
	18.10.2017					2h	
	19.10.2017	Präsentationsvorbereitung und Papiermodell				4h	
	20.10.2017	Präsentation und Abgabe des Konzepts				0,5h	
	21.10.2017			Nacharbeit Protokoll, Meetingergebnisse 1h			
KW 42	22.10.2017	interface 2h					
	23.10.2017	Anmeldesystem 1h	Erarbeitung Datenbank 4h				
	24.10.2017						
	25.10.2017				Kontrolle Anmeldesystem 1,5h		
	26.10.2017						
	27.10.2017	Group-Meeting				Verknüpfung Datenbank Anmeldesystem 2h	
	28.10.2017	Verknüpfung Datenbank Thore 1h Niklas 2h			Kontrolle Interface Code 1h		
	29.10.2017			Nacharbeit Protokoll, Meetingergebnisse 2h			
	30.10.2017						
	31.10.2017						
	01.11.2017						
	02.11.2017						
KW 44	03.11.2017	Group-Meeting				Planung und Besprechung Restablauf 2h	
	04.11.2017	Nacharbeit Protokoll, Meetingergebnisse 1h					
	05.11.2017					Debugging 1h	
	06.11.2017						
	07.11.2017						
	08.11.2017						
	09.11.2017	Verkopplung Datenbank Interface 3h					
	10.11.2017	Group-Meeting				Standgespräch und Resterarbeitung 1,5h	
KW 45	11.11.2017	Nacharbeit Protokoll, Meetingergebnisse 1h					
	12.11.2017					Verbesserung von Interface 1h	
	13.11.2017					Datenbank 1h	
	14.11.2017						
	15.11.2017	Nachbearbeitung HTML CSS 3h	Stopuhr 1,5h				
	16.11.2017						
	17.11.2017	Group-Meeting/Präsentationsvorbereitung				Gemeinsames Arbeiten 3h	
	18.11.2017						
KW 46	19.11.2017	Meetingnachbearbeitung 2h					
	20.11.2017	Datenbank Ausgabe 2h	Datenbank Ausgabe 4h				
	21.11.2017					Debugging 1h	
	22.11.2017						
	23.11.2017	Präsentationsvorbereitung				1h	
	24.11.2017	Group-Meeting/Präsentationsvorbereitung				Gemeinsames Arbeiten 3h	
	25.11.2017						
	26.11.2017	Zeitaufwand 1h				Korrekturlesung Zeitaufwand, Meilensteinplan 0,5h	
	27.11.2017					2h	
	28.11.2017					2h	
	29.11.2017					2h	Ausserplanmässiges Group Meeting
	30.11.2017	Präsentationsvorbereitung und Polishing				2h	
KW 48	01.12.2017	Präsentation und Abgabe/DEADLINE					
	02.12.2017						
	03.12.2017						
	04.12.2017						
	05.12.2017						
	06.12.2017						
	07.12.2017						
	08.12.2017						
KW 49	09.12.2017	Formatierung und Polishing Projektbericht 10h				Korrekturlesung Projektbericht	
	10.12.2017	Endabgabe Projektmanagement					

# Protokoll Gruppenmeeting

## Protokoll 05.10.17

Erarbeitung Hauptidee: Time Tracking Tool, Zeiterfassung mit Datenbank

### **Start 12:00**

Diskussion Projektname: Time Tree (4)

Vorschläge waren Track Stat Track Tree, Time Plan(e), SchedStat, SchedTrack(er)

12:45 Projektideenfindung beendet

12:50 Thore August verlässt Meeting für wichtigen Termin

Time Tracking in zwei Teilen

Part #1: Ankunfts- /Abkunftszeit tracken, +- Zeit an Wochenstatistik gekoppelt

Part #2: Task/Projektzeiterfassung, Stoppuhr -> Zeit an Projekten/Task tracken

~ Stop/Play-button

~ Formula

->Datum

->Zeit

->Task/Projektauswahl

->Kommentar/Sonstiges

~ Absende-Button

Stilistische Entscheidung: Hauptsprache und Navigation in Deutsch

Code in ENGLISCH!!!

=>var keksKrümmel

=>const KeksKrümmel

13:30 Vorbereitung für Papiermodell und Struktur

### **Ende 14:30**

## Protokoll 13.10.17

Pitch Vorschlag für Arbeitsmodell

### **Start 12:30**

Arbeitsmodell Typ einfache Hierarchie:

Fest Rollenzuteilung,



Projektmanager

QA Manager

Head of Code Head Designer

Pro: Übersichtlich, leicht zu kontrollieren,

Contra: Wenig kreativer Freiraum,

Arbeitsmodell Typ Chaos:

Keine feste Rollenverteilung

Pro: Schnell, effizient in kleinen Gruppen, gut bei Leuten die alles können

Contra: Keine Übersicht, kann zu internen Probleme führen

Votum 4 zu 0 für einfache Hierarchie

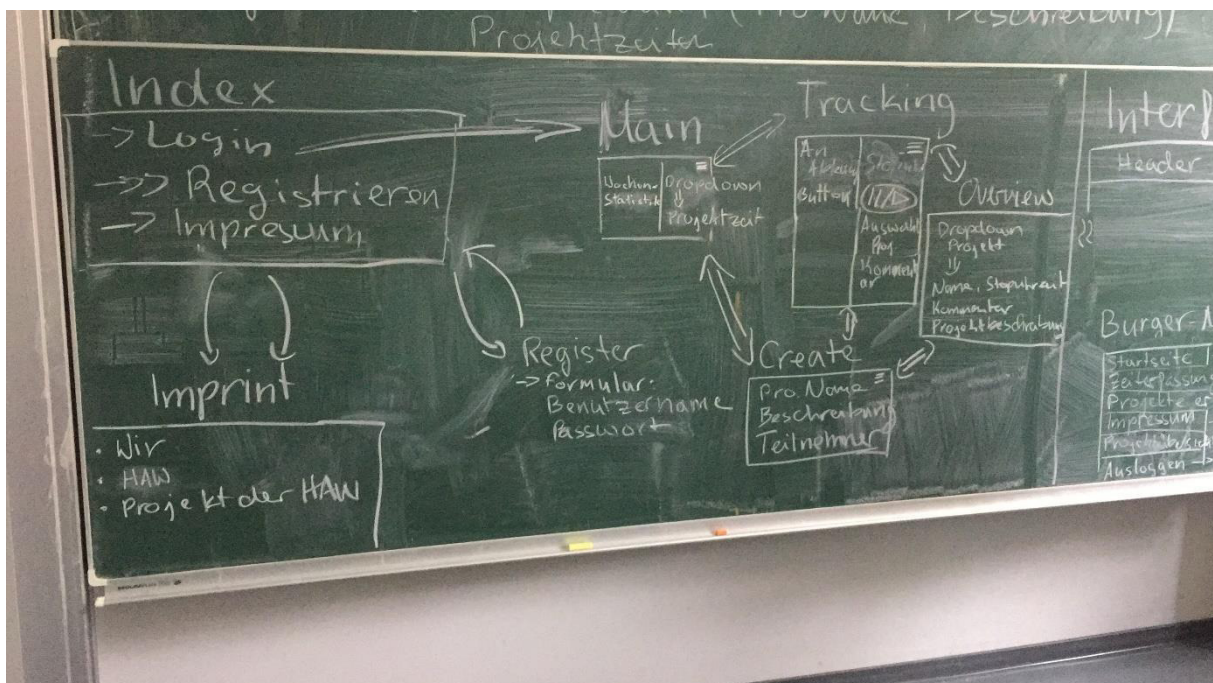
**Ende 13:30**

## Protokoll 17.10.17

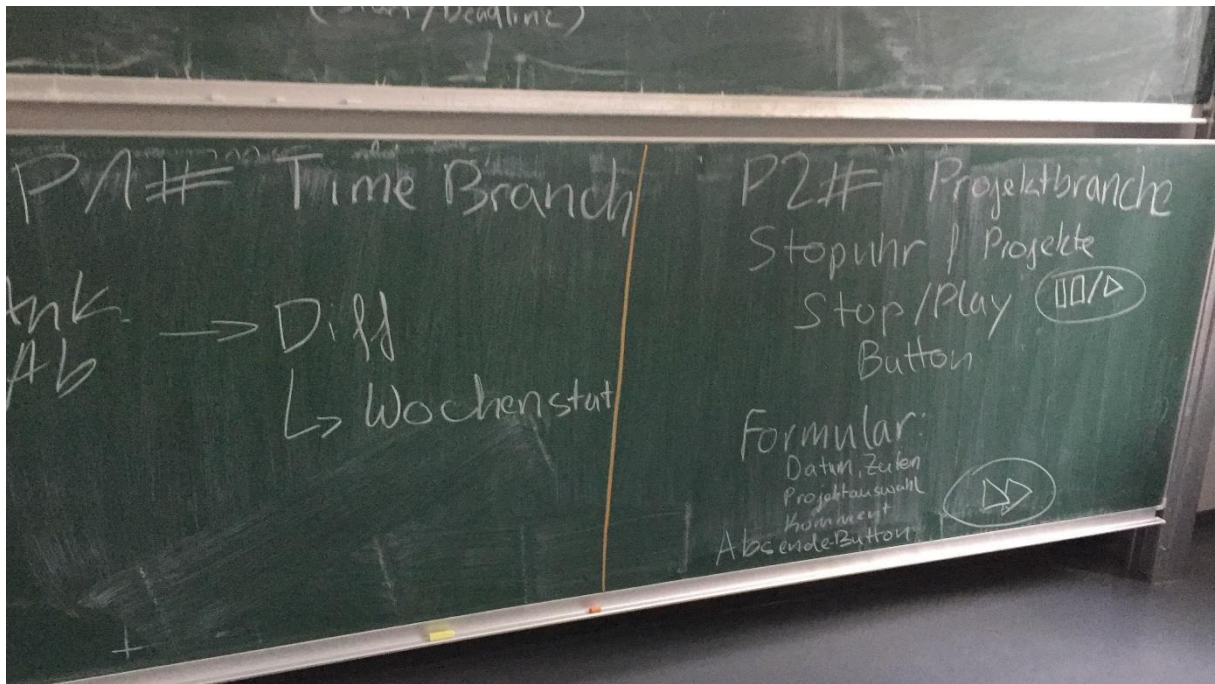
Erarbeitung Papiermodell Struktur

**Start 12:00**

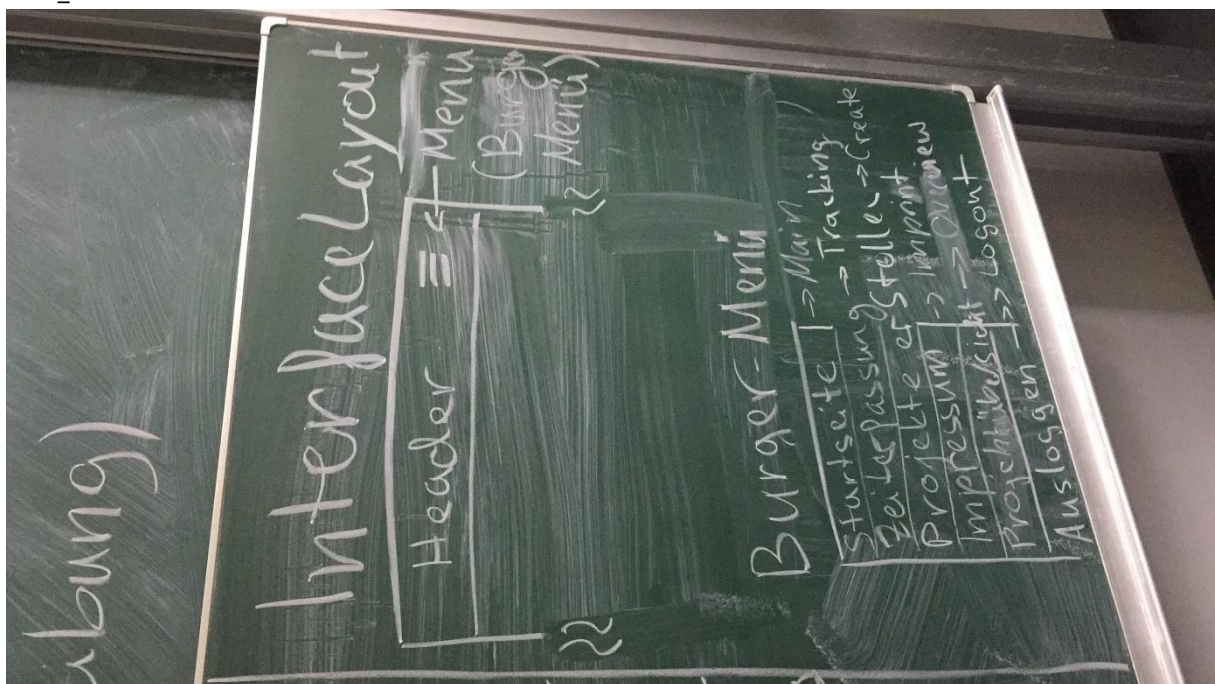
siehe Tafel Image IMG\_2784 bis IMG\_2788



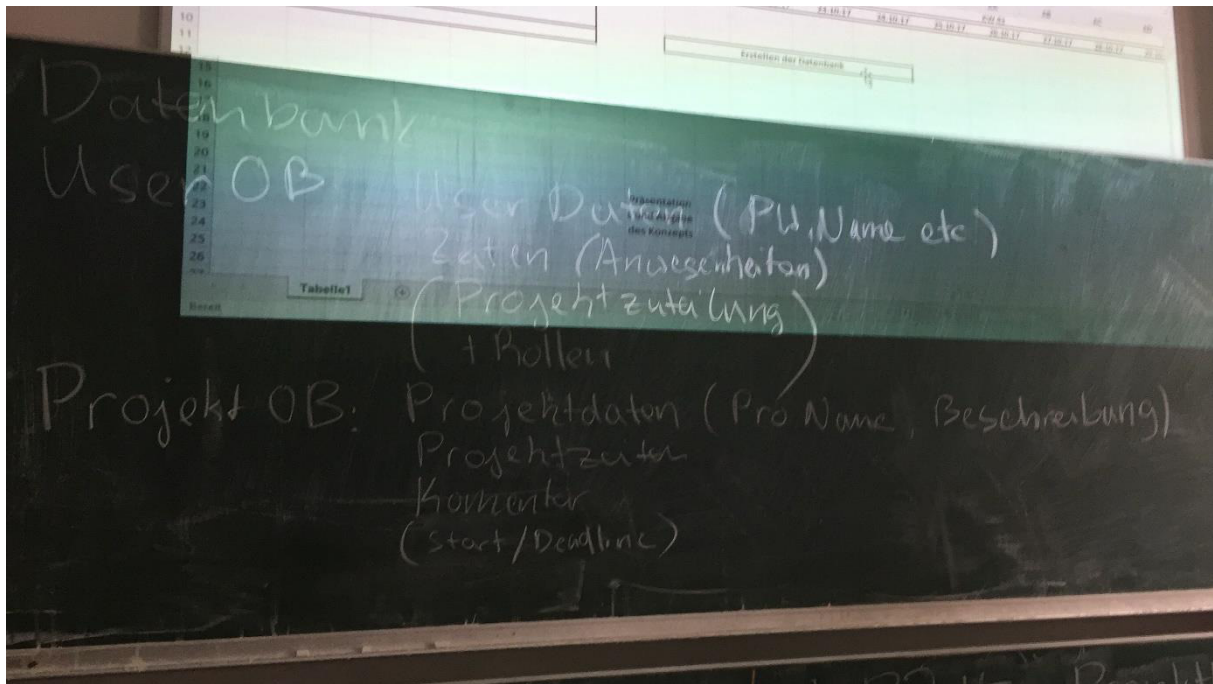
IMG\_2784



IMG\_2785



IMG\_278



IMG\_2788

**Ende 14:00**

## Protokoll 27.10.17

**Start 12:05**

Verknüpfung Datenbank und Anmeldesystem

**Ende 14:00**

## Protokoll 03.11.17

### Group Meeting Erstellung der DB und Initialisierung der Server.js

#### **Start 12:15**

Thore erklärt HTML/CSS Code

Niklas und Thore besprechen sich

Seiten erstellen

->Server.js erstellen/Authentifizierung

Initialisierung der Datenbank

estimated time 2h

Freitag

12:20

Package.json initialisieren (nodemon, express etc.)

jquery installiert (jquery Dropdown-Menü)

Session initialisiert

Erstellung der PW Funktion

12:55

Hinzufügen einer Passwort-Verifikation durch 2-mal eingeben

13:46

Login Implementierung erfolgreich

#### **Ende 13:55**

## Protokoll 10.11.17

#### **Start 13:15**

Besprechung zum Stand.

Liegen gut in der Zeit

Nähere Besprechungen zum Code

#### **Ende 14:45**

## Protokoll 17.11.17

**Start 12:00**

Datenbank Ausgabe erarbeiten

**Ende 15:00**

## Protokoll 24.11.17

Rückgabe von DB in HTML

**Start 9:45**

Zeiterfassung

Erstellung

Übergabe von DB auf die HTML Seiten

**Ende 12:45**

## Protokoll 29.11.17

**Start 12:20**

Ergebnisse darstellen auf der Startseite (Main)

Wochenzeiten Ausgabe

Tracking optimiert. Änderung auf Berechnung und dann Speicherung

**Ende 14:20**



# Lessons Learned

## **1 Teamfindung**

Bei der Teamfindung hatten wir zwei Modelle zur Auswahl, der gleich Stellung aller Mitglieder um so viel Input zu haben. Allerdings war unser komplettes Team darauf ausgelegt eine Rolle zu übernehmen die einem Neu war um maximalen Lernerfolg zu erzielen.

Auf Wunsch von Michael wurde in der Abstimmung für eine kleine Hierarchie gestimmt. Dies wurde getan da die kleine Hierarchie Übersichtlicher ist. Zwar gibt sie weniger kreativen Freiraum ist aber nicht so einschränkend wie die große Hierarchie. Sie kommt außerdem mit einer gewissen Eigeninitiative, diese kann jedoch Vorteil oder Nachteil sein.

Quality Assurance Manager wurde Felix Zander. Seine Hauptaufgabe bestand darin alles zu überprüfen auf Kundenzufriedenheit. So wurde die Formatierung des Codes von ihm bestimmt und ein Großteil der Debugging Arbeit.

Head of Code wurde Niklas, die Hauptaufgabe bestand darin sich schnell in JavaScript einzulesen, eine bis dato neue Sprache für uns, und zu entscheiden was im Rahmen der Möglichkeit liegt. Ebenfalls war er Anlaufstelle für jegliche Fragen zur Datenbank und dem Backend.

Head Designer wurde Thore August. Er wollte aus Eigeninteresse sich um CSS und Interface kümmern und übernahm auch das Design von Logo und will in Zukunft als Webdeveloper tätig werden.

Somit wurde Michael Vo der Projektmanager, dessen Hauptaufgabe die Strukturierung des Projektes ist. Hinzu kamen Designvorschläge und Protokollierung sowie Ideenfindung.

## **2 Sammelphase und Themendarstellung**

Die Sammelphase half uns relative schnell zu gestalten wie das künftige Projekt zu gestalten war. Wir überlegten uns unsere eigenen Szenarien wie unser Lebensalltag war als Zeitarbeitskraft und beschlossen auf dieser Tatsache unser Projekt daran zu orientieren.

Entschieden uns hierfür für ein zwei System Modell. Eine Webapplikation die im Alltag sowohl rechtlich als auch wirtschaftlich zur Übersicht helfen kann und Flexibilität offenhält.

Unsere Webapplikation soll eine um denn Globus verfügbare Zeiterfassung sein, sie soll nützliche Daten und Übersichten geben für Projektmanager, Investoren aber auch denn Mitarbeitern selber helfen zu sehen wann sie wo gearbeitet haben.

### **3 Arbeitsstruktur-Vorbereitung**

Unser Workshop oder intern genannt Meetings, soll zwei Funktionen erfüllen. Zum einen die Themenfindung und Spezifikation der Strukturen und zum anderen im Verlaufe der Projektarbeit näher an die Zusammenfindung zwischen einzelnen Mitarbeitern dienen.

In der Zweiten Phase war es uns wichtig, dass wir alle an einem Strang ziehen und wissen was der andere beabsichtigte mit einzelnen Schritten. Also wurde beschlossen jede Woche am Freitag mit gegebenenfalls weiteren Meetings vor besonderen Terminen wie Abgabe und Präsentation zu haben.

### **4 Ablauf Konzeptabgabe**

1. Begrüßung
2. Projektvorstellung
  - a. Anhand eines Papiermodells
3. Einleitung
  - a. Warum, Wieso und Wie wir unser Projekt gestalten wollen?
4. Beweggründe
5. Themenanalyse und Empfehlungen
6. Endziel
7. Feedback

Wir begrüßten uns, stellten uns und unsere Rollen im Team jeweils vor. Stellten unser Projekt vor.

Prof. Pläß fand unser Thema sehr gut erarbeitet und hatte keine Einwendung gegen unser Projekt und somit verlief das Workshop relativ reibungslos.

### **5 Zusammenfassung**

Insgesamt wurde unsere Projektstruktur relativ klar und wir konnten gut ausarbeiten welche Rahmenbedingungen wir brauchen für einen reibungslosen Ablauf. Unser Projekt kam gut an beim Klienten und auch die Zusammenarbeit klappte. Zwar waren die geplanten Treffen etwas lang, halfen aber bei der Zielorientierung.

## **6 Lessons Learned**

Insgesamt verlief die Arbeit relativ Reibungslos, die Meetings waren jedoch manchmal zu lang von der Dauer her. Die Erarbeitung von Code war relativ schwer, jedoch hatten wir gute Bezugspunkte aus dem Tutorium entnehmen können. Ebenfalls war der Austausch gut zwischen Tutoren und uns.

An manchen Stellen wurde jedoch die Schwäche der Flachen Hierarchie klar, ein härteres Eingreifen des Projektmanagers hätte Arbeitsoptimierung bewirken können bei der Qualitätskontrolle jedoch war es aufgrund der Eigenständigkeit jedem selber überlassen wie tief er sich in die Materie einliest.

Auf Seiten des Codes wurde uns klar wie wenig Anhaltspunkte es für eine Datenbank wie TingoDB gab. So musste vieles mit lange Gruppen-Meetings gelöst werden, wobei uns Zeit verging für das Verbessern von anderen Features. Aber dennoch wurde das Projekt zu einer soliden Basis für zukünftige Verbesserungen.

Das Front-End hatte kaum bis keine Probleme aufgrund der großen Hilfe die es auf Seite der Tutoren und öffentlichen Quellen gab.

Gelernt haben wir eine Menge, vor allem auf Struktureller Basis.