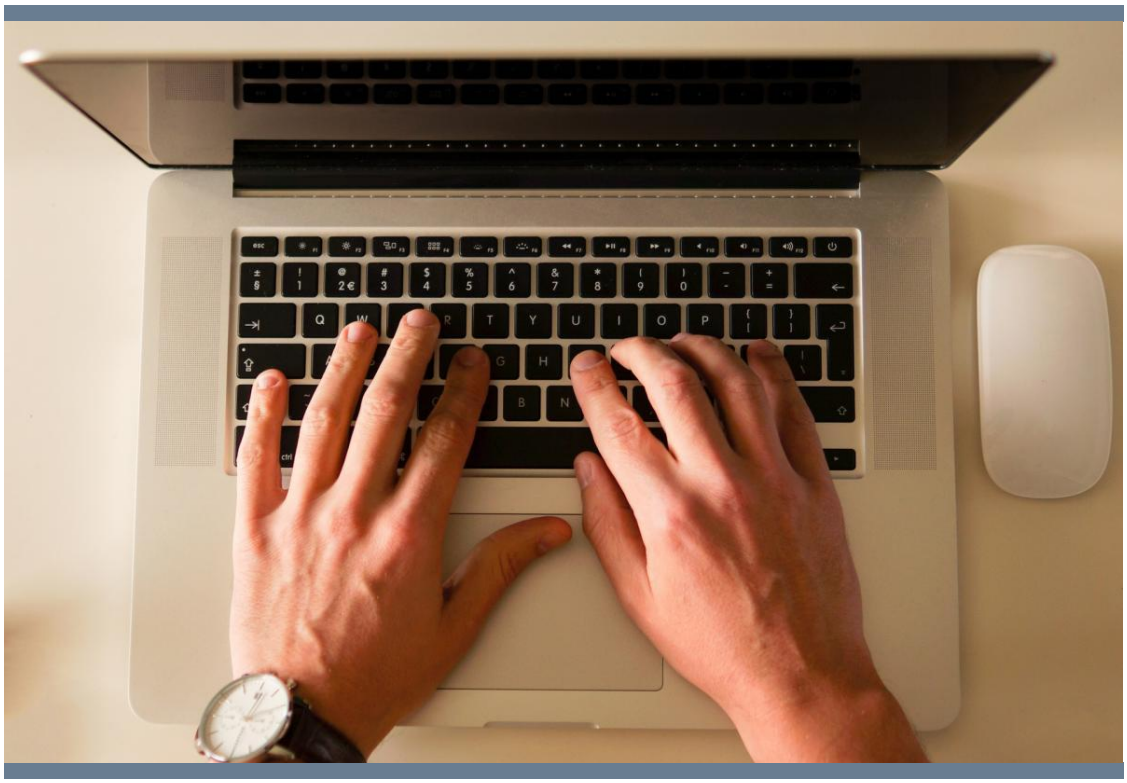


Technische Dokumentation Benutzer-Profiling

Projektteam Benutzer-Profiling

Martin Kieliger
Sandro Luder

Version 1.0, 28.12.2017

Berner Fachhochschule
Departement für Technik und Informatik

Inhaltsverzeichnis

1 Projektauftrag und Vision

Zur Überprüfung ob ein Benutzer bei der Anmeldung auf einer Webseite wirklich die erwartete Person ist, gibt es bereits einige vielversprechende Verfahren. Solche sind unter anderem, nach der Passworteingabe ein zusätzlicher SMS-Code oder das klassische TAN-Verfahren, wie es von vielen Banken beim online Banking eingesetzt wird. Zu den neueren Authentifizierungsmethoden gehört der Fingerabdruckscanner, welcher auch bei Smartphones eingesetzt wird, die Iriserkennung oder die Gesichtserkennung (Beispiel Windows Hello, welche letztere beiden Methoden vereint). Ein weiteres Verfahren für die Authentifizierung ist die Analyse des Tippverhaltens von Usern.
Es gibt viele verschiedene Ansätze zur Messung des Tastaturverhaltens einer Person, jeder Ansatz bietet Vor- und Nachteile. Der allgemeine Ablauf ist, ungeachtet des Ansatzes, meist ähnlich.
Vor der eigentlichen Authentifizierung ist eine Datenbeschaffung notwendig. Man spricht von der Aufnahmephase. Während dieser Zeit werden die Daten für die spätere Authentifizierung gesammelt. Beispielsweise kann während der Registrierung auf einer Webseite, ein Text abgetippt werden. Mit dieser Messung wird anschliessend ein Benutzerprofil mit einer einzigartigen Signatur erstellt. Jeder künftige Login Versuch kann dann mit dieser Signatur verglichen werden.
Die Algorithmen für derartige Softwarelösungen sind sehr komplex. Es muss auf extrem vieles geachtet werden. Psychologischer Zustand des Benutzers, Fehlererkennung, einhändiges tippen, unterschiedliche Druckpunkte der Tastaturen, virtuelle Tastaturen, etc.
In der uns zu Verfügung stehenden, relativ kurzen Zeit, haben wir ein Prototyp für eine solche Software entwickelt.
Der Nutzen dieser Software liegt beim Benutzer, sowie beim Webseitenbetreiber. Durch eine solche Zwei-Faktor-Authentifizierung wird der Zugang auf eine Webseite sicherer.
Allerdings mussten wir einige Einschränkungen definieren.
Für unser Projekt nehmen wir an, dass der User immer im Sitzen tippt, sich immer in demselben Geisteszustand befindet und beidhändig das Passwort eingibt.

2 Anforderungen

2.1 Funktionale Anforderungen

ID	Status	Prio	Beschreibung
F1.1	Freigegeben	M	Wenn der User sich registriert muss das System ein Benutzerprofil des Users anlegen. Es wird seine E-Mail Adresse, Vorname, Nachname und Passwort in der Datenbank gespeichert.
F1.2	Freigegeben	M	Wenn der User sich einloggt muss das System, anhand des erstellten Profils, sein Tippverhalten analysieren. Das Ergebnis wird in % angegeben.
F1.3	Freigegeben	M	Was wird gemessen und analysiert? • Duration: Dauer in ms wie lange eine Taste gedrückt wird. • Latency: Dauer in ms zwischen Drücken der ersten Taste und Loslassen der zweiten Taste. • Interval: Dauer in ms zwischen Loslassen der ersten Taste und Drücken der zweiten Taste.
F2.1	Entwurf	P1	Wenn der User sich erfolgreich eingeloggt hat, werden ihm die Daten der Analyse angezeigt (Genauigkeit, Zeit, etc…)
F3.1	Entwurf	P2	Nach 3 fehlgeschlagenen Authentifizierungen wird der Benutzer gesperrt.

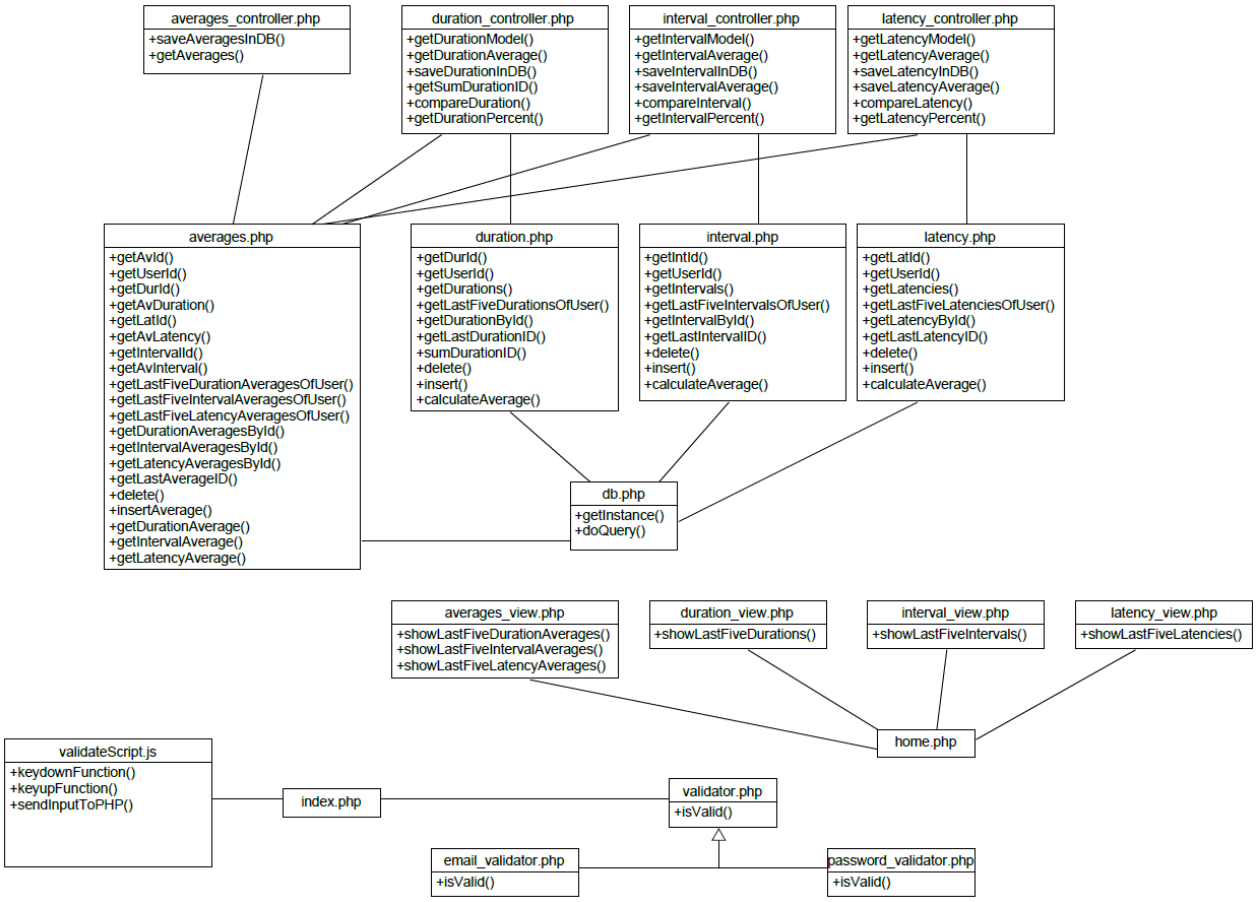
2.2 Technische Anforderungen

ID	Status	Prio	Beschreibung
T1.1	Freigegeben	M	HTML / CSS Website mit Benutzerlogin, sowie Benutzerregistration.
T1.2	Freigegeben	M	Die Website kann mit jedem Webbrowser geöffnet werden.
T2.1	Freigegeben	M	PHP wird als Backend verwendet und um die Datenbankanbindung herzustellen.
T2.2	Freigegeben	M	JavaScript wird benutzt um die User Inputs entgegenzunehmen.
T3.1	Freigegeben	M	Datenbank (MySQL) (für den Prototyp mit PHPMyAdmin erstellt)
T3.2	Freigegeben	M	Windows 10 Benutzer

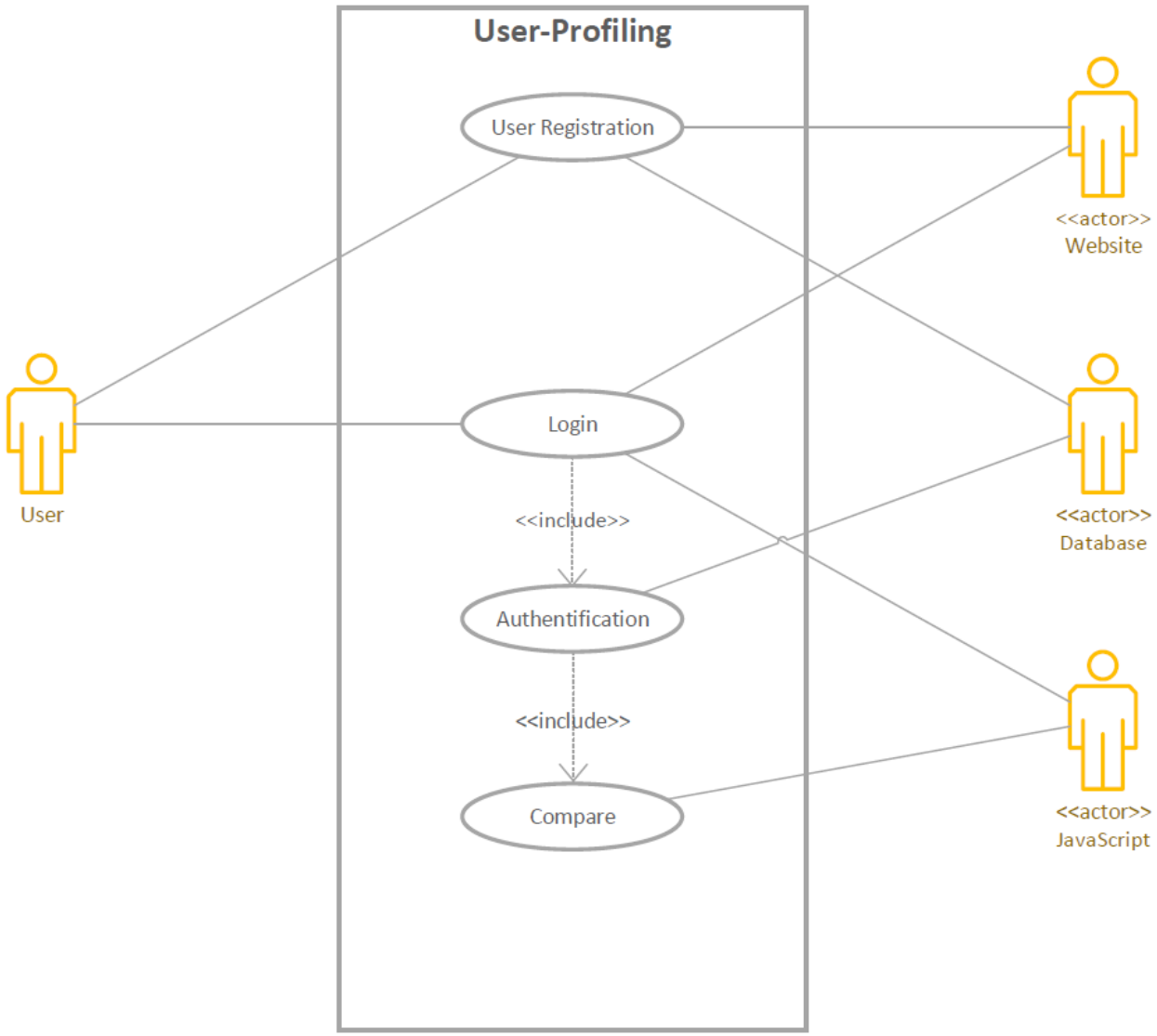
2.3 Qualitätsanforderungen

ID	Status	Prio	Beschreibung
Q1.1	Freigegeben	M	Das System muss den Benutzer bei 9 von 10 Mal erkennen.
Q2.1	Freigegeben	M	Das System muss eine Genauigkeit von 90% haben.
Q3.1	Freigegeben	M	Einfachheit für den Benutzer. (Muss nur tippen).
Q4.1	Freigegeben	M	Das System soll das Passwort ergänzen und **nicht** ersetzten.
Q5.1	Entwurf	P1	Um die Genauigkeit zu erhöhen, können noch weitere Überprüfungsmethoden implementiert werden. (“Frequency of Errors”, “Flight Time”, etc…)

3 Klassendiagramm

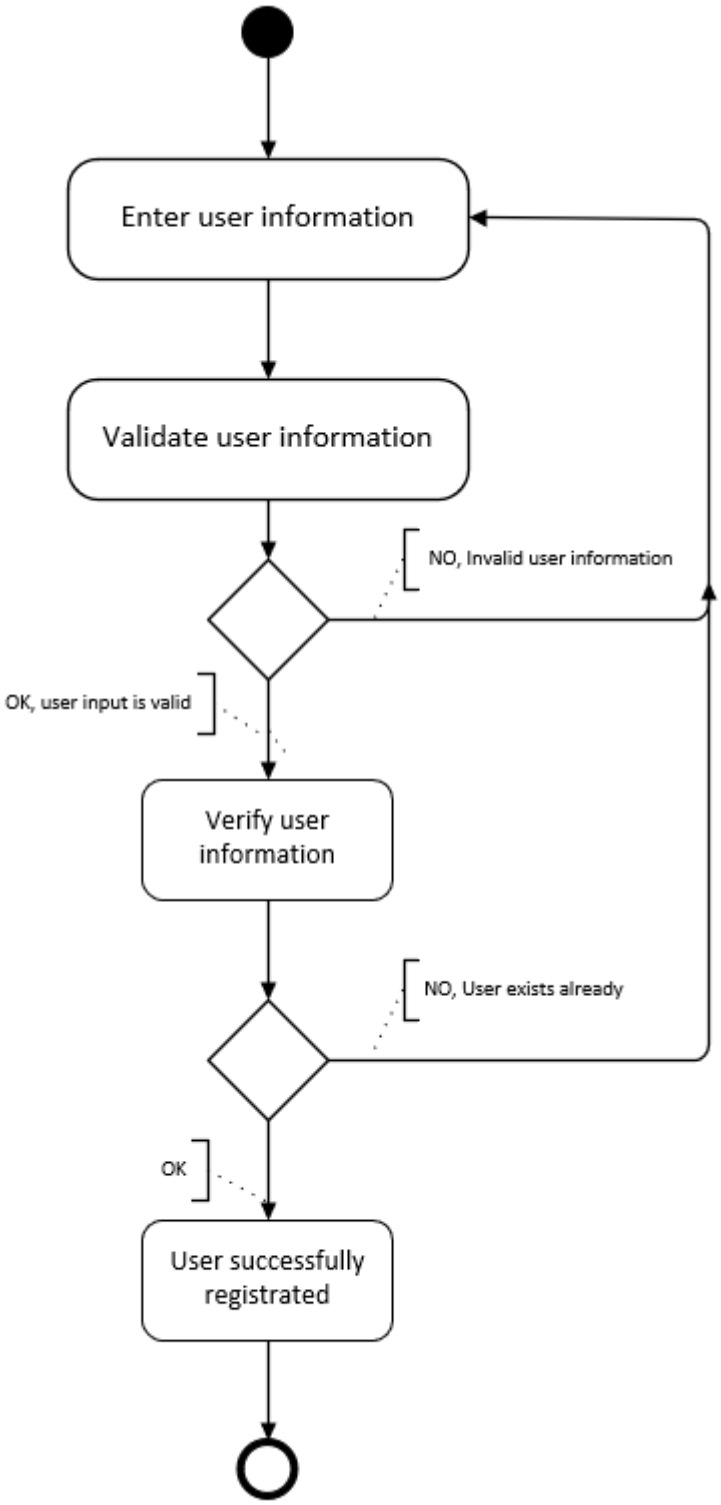


4 Use Case Diagramm



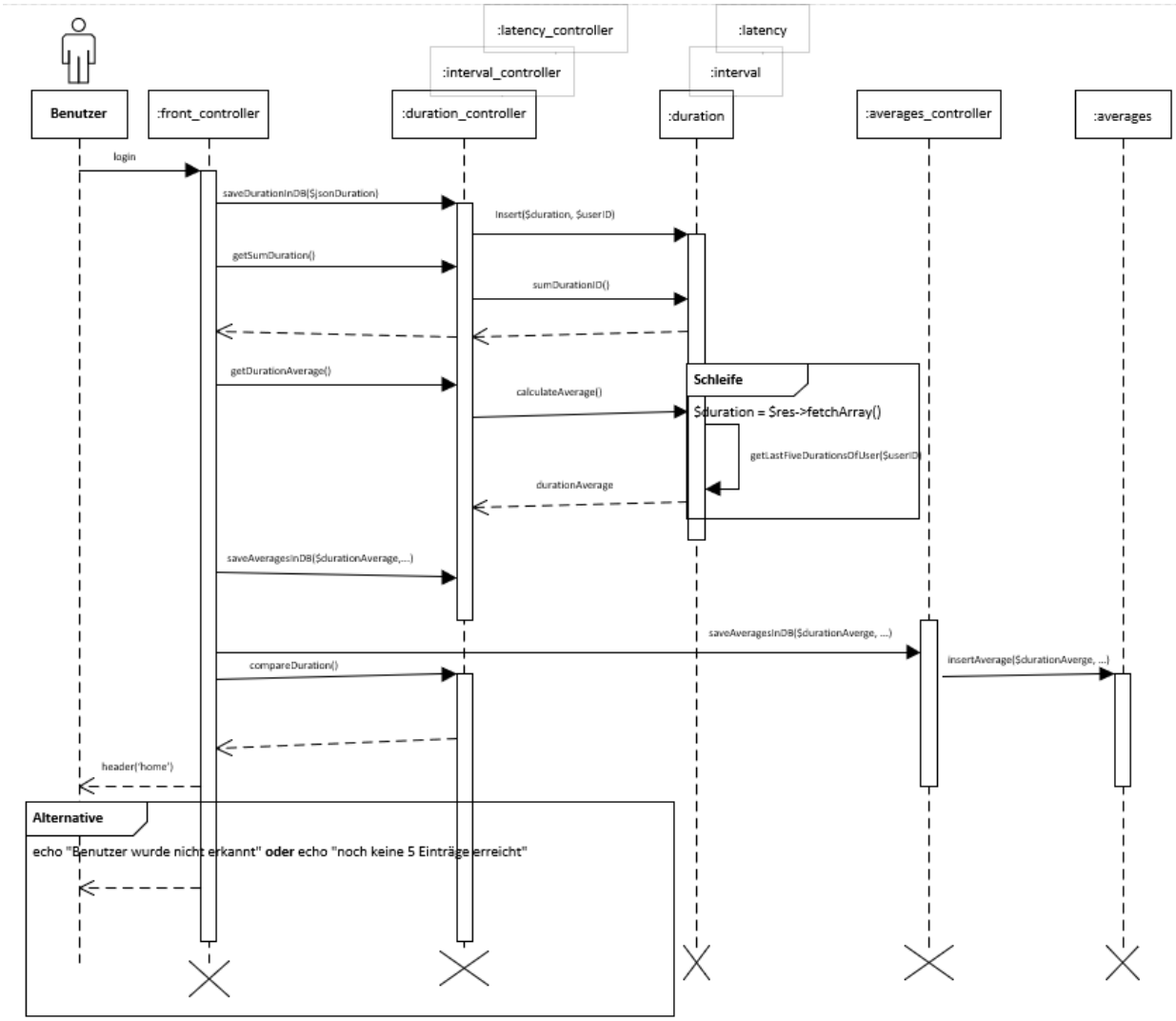
5 Activity Diagram

Für die Registrierung:



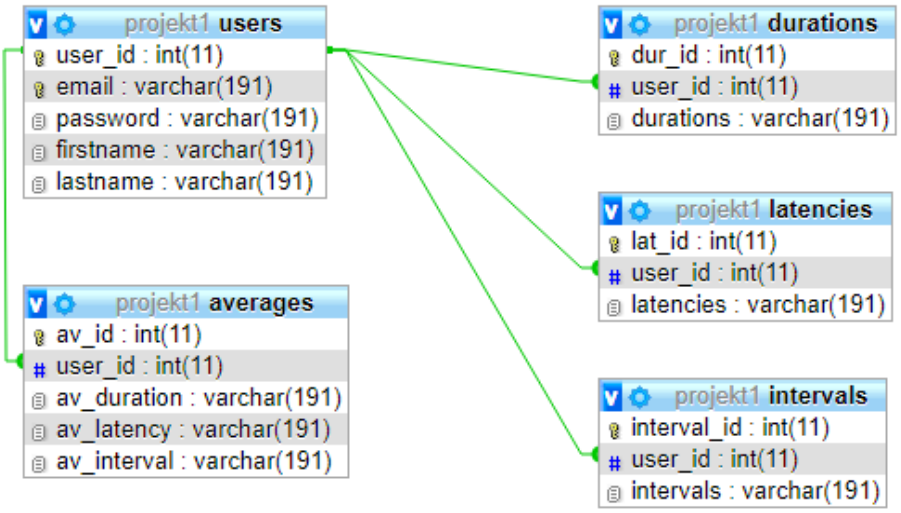
6 Sequenzdiagramm Login

Untenstehendes Diagramm zeigt den Ablauf nachdem der Benutzer auf den “Login-Button” geklickt hat. Aus Platzgründen wurde nur die Eigenschaft “Duration” dargestellt. Die anderen Objekte “Interval” und “Latency” verhalten sich gleich.



7 Datenbank

Aufgrund unserer Erfahrungen vom Modul “Software Engineering & Design” und unseren Kompetenzen im Bereich Datenbanken, haben wir uns für eine MySQL Datenbank entschieden. Das Schema ist relativ einfach. Es besteht aus den 5 Tabellen: *users, averages, durations, latencies* und *intervals*. Auf diese Weise gibt es für jede “messbare Eigenschaften” eine Tabelle und wir wissen dank der Foreign Keys auch, von welchem User, welcher Eintrag ist. Zusätzlich ist das Schema einfach erweiterbar. Beispiel: Will man das System erweitern indem man nachträglich noch das Attribut “Flight-Time” (Dauer zwischen Drücken der ersten und Drücken der zweiten Taste) analysieren möchte, so kann einfach eine neue Tabelle *flight-time* hinzugefügt werden, ohne dass das bestehende Schema stark verändert werden muss.

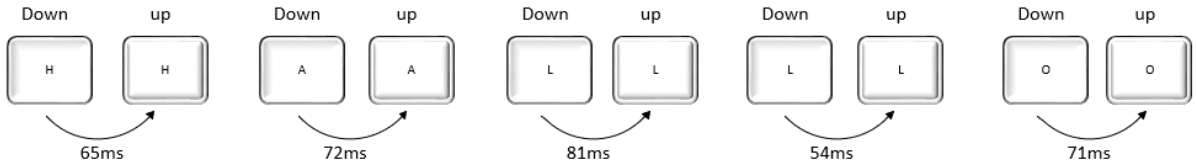


Beispiel der gespeicherten Daten in der Tabelle *durations*:

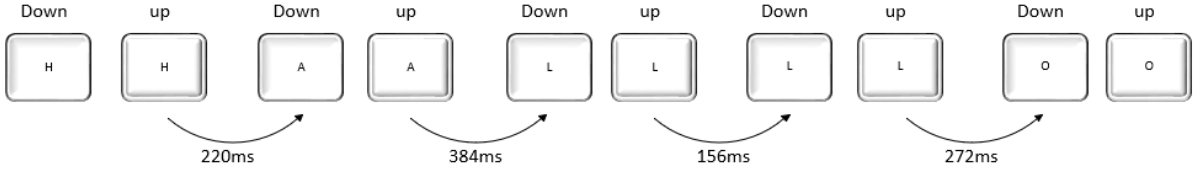
dur_id	user_id	durations
110	5	[76,89,76,57,69,56,70]

8 Messbare Eigenschaften

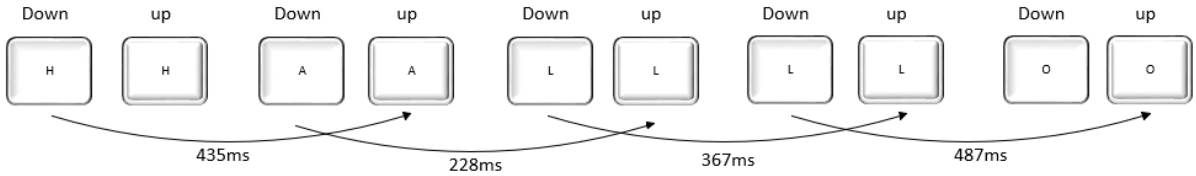
Duration: Wie lange ein Taste gedrückt wird



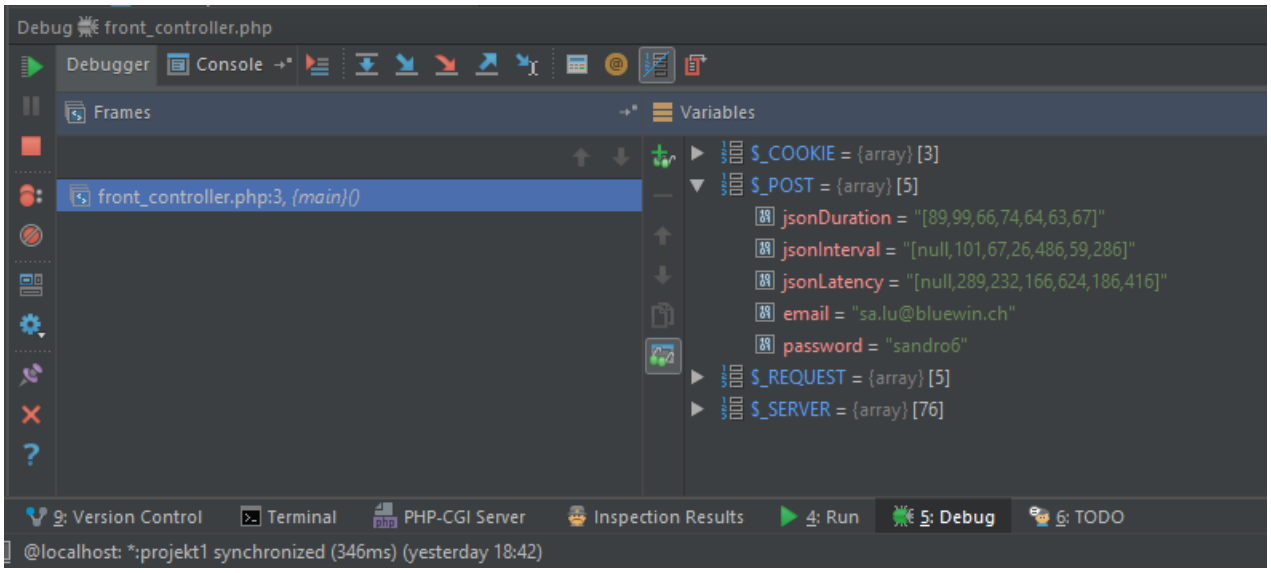
Interval: Dauer zwischen Loslassen der ersten und Drücken der zweiten Taste



Latency: Dauer zwischen Drücken der ersten und Loslassen der zweiten Taste



Untenstehendes Bild zeigt den Zustand nach erfolgreichem Login. In der Superglobalen Variable $_POST befinden sich die 5 Werte ‘jsonDuration’, jsonInterval’, ‘jsonLatency’, ‘email’ und ‘password’.



9 Probleme, Fragen & Lösungen

Problem	Lösung
Datenbank: Was für eine DB wollen wir? MySQL, noSql oder werden nur Cookies verwendet? Welche Daten werden wie, in welchem Format gespeichert?	Wie bereits im Abschnitt “Datenbank” haben wir uns für eine MySQL DB entschieden. Gespeicherte Daten sind “duration”, “latency” und “interval”.
Was wird als Backend verwendet?	Zur Diskussion stand für uns Java (da wir mit dieser Sprache vertraut sind) oder PHP. Wir haben uns für PHP entschieden. Die Arbeit für das Backend haben wir lange unterschätzt. Wir dachten zuerst, wir könnten vieles bereits mit JavaScript erledigen. Schlussendlich, geschieht nun das Meiste auf dem Server.
Wird ein Framework verwendet, wenn ja, welches?	Uns war anfangs kein Framework bekannt, welches sich gut für unsere Software eignen würde. Nach der ersten Besprechung mit Herrn Kaltz und intensiver Recherche im Internet, haben wir explizit auf ein Framework verzichtet. Der Aufwand, sich in ein neues Framework einzuarbeiten ist gross. Zudem ist der Lerneffekt für uns persönlich grösser wenn wir reinen PHP Code schreiben.
Wie wird der Code strukturiert? Welche Architektur wird verwendet?	Wir haben versucht das MVC Pattern zu implementieren. Es ist uns bereits aus den Modulen “SW Engineering & Design” und “Web Development” bekannt.
Zusammenarbeit im Team, Version Control: Was gibt es für Möglichkeiten und Tools für die Zusammenarbeit? Git war von Anfang an klar, aber gibt es noch bessere/zusätzliche Tools um die Zusammenarbeit und den Code zu managen?	Vagrant und/oder Docker waren noch Möglichkeiten. Wir haben uns aber dagegen entschieden, damit wir mehr Zeit in das eigentlichen programmieren investieren konnten. Da wir uns sonst auch in diese Tools wieder hätten “einarbeiten” müssen.
Aller Anfang ist schwer! “Benutzerprofiling” ist ein breiter Begriff. Von dem Thema her hatten wir keine grossen Einschränkungen. Das hatte Vor- und Nachteile. Unsere anfänglichen Ideen waren, dass wir den Benutzer einen Text abtippen lassen und daraus ein Benutzerprofil erstellen. Wir haben uns an Beispiele wie https://www.keytrac.net/de orientiert. Das würde bedeuten, dass der User unabhängig vom Text erkannt werden kann. Dies ist ein noch unerforschtes Gebiet und würde tief in die Richtung Biometrie gehen. Der Aufwand für ein 2er Team in dieser kurzen Zeit war nicht realistisch. Das wäre ein Thema für eine Bachelor- oder gar Masterarbeit.	

Psychologie des Benutzers und Fehler bei der Passworteingabe: Woran erkennt die Software ob der Benutzer das Passwort einhändig eintippt? Was ist, wenn er sich vertippt? Ein Benutzer unter Stress tippt völlig anders als ein ausgeruhter, entspannter User.	Für unseren Prototypen setzen wir also folgende Bedingungen voraus: • Der Benutzer muss absitzen und beidhändig tippen • Er ist immer in der gleichen Form (ausgeruht, nicht müde, nicht unter Stress) • Keine Fehler, er darf sich nicht vertippen
Es wurde lange Zeit nur die duration in der DB gespeichert. Interval und latency nicht.	In der DB waren die constraints nicht richtig für Interval und Latency. Die Referenz war auf sich selbst und nicht auf UserID, wie bei Duration.
$_POST['jsonDuration'] war immer NULL, obwohl die Daten in der DB gespeichert wurden. Wie konnten wir dann im PHP Script abfragen ob und was der Benutzer getippt hat, wenn es als NULL angegeben wird?	Beim Debuggen haben wir bemerkt, dass vom "ValidateScript.js" 3 POST requests und noch 1 POST request von der "form" im index.html an den "front_controller" geschickt wurden.
Wie funktioniert der Vergleich der Eingabe mit den Daten in der DB? Gibt es bereits Vergleichsalgorithmen die wir einbauen können? (z.B. Pattern Matching, KMP-Algorithm)	
Probleme mit dem GIT Repository, pushen, pullen und mergen.	