

Projektarbeit
im Studiengang
Computer Networking Bachelor

Med-Eval

Medizinisches Evaluationssystem in Kooperation mit
Uniklinik Freiburg - Deutsche Telekom

Referent : Prof. Dr. Achim P. Karduck

Koreferent : -

Vorgelegt am : 17.02.2017

Vorgelegt von : Axel Butz, Eugen Jastremskoj, Marcin Krawczyk, Jan Ondruch, Jan Stodt
Matrikelnummer: 227712(Ax) 247725(Eu) 246286(Jan S) 248667(Mar)
255358 (Jan O) (Brno University of Technology(VUT)),

Abstract

In the **Med-Eval** project, a cloud-based web application was created to support clinical trial surveys. Med-Eval is based on an open-source survey tool which has been adapted for the use in clinical environments. The survey tool allows a modular set-up system to be configured for the creation of the questionnaires. The orchestration of the survey system is handled by the Open Source Tools Ansible.

Im Projekt **Med-Eval** wurde eine Cloud gestützte Web-Applikation für die Unterstützung der Umfragen von klinischen Studien erstellt. Med-Eval basiert auf einem Open-Source Umfragetool welches für die Verwendung in klinischem Umfeld angepasst wurde. Durch das Umfragetool ist ein modulares Baukasten-System für die Erstellung der Fragebögen konfigurierbar. Die Orchestrierung des Umfragesystems erfolgt mittels dem Open Source Tools Ansible.

Inhaltsverzeichnis

Abstract	i
Inhaltsverzeichnis	iv
Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Abkürzungsverzeichnis	ix
Listings	x
1 Einleitung	1
2 Grundlagen	3
2.1 Kenntnisse und Fähigkeiten	3
2.2 Projektunterstützende Maßnahmen	3
2.3 Entwicklungsprozess	4
3 Realisierung	5
3.1 limesurvey	5
3.2 limesurvey Erweiterung	5
3.2.1 Requirements Server	5
3.2.2 Requirements Limesurvey	5
3.2.3 Statements zur Sicherheit	6
3.2.4 Ansible Grundlagen	7
3.3 Installation mit Ansible	7
3.3.1 Ubuntu PPA	7

3.4	Playbook	8
3.4.1	Installation	8
3.4.2	Hardening	9
3.4.3	Backup und Restore	9
3.5	Hinweise zur Wartung	9
3.5.1	Systemsoftware	10
3.5.2	Limesurvey und Konfigurationsdateien	10
3.5.3	Entwicklung von Erweiterungen	10
3.5.4	Körper	12
3.6	Codeanhänge	14
	Literaturverzeichnis	27
	Eidesstattliche Erklärung	29

Abbildungsverzeichnis

Abbildung 1: Beispiel Desktop and Mobile Umfrage	1
--	---

Tabellenverzeichnis

Abkürzungsverzeichnis

Listings

3.1	body.css	14
3.2	map.js	14
3.3	wetter.js	15
3.4	URL Parsing	22
3.5	PDF Creation	23
3.6	Requirements	25
3.7	Bash Deploy	25

1 Einleitung

Im Projekt Med-Eval ist eine Web-Applikation für die Unterstützung von klinischen Studien entstehen. Studienteilnehmer/innen sollen per QR-Code den Link zum Studien-Fragebogen öffnen. Die Navigation durch den Fragebogen soll auf dem Endgerät (Laptop/Smartphone) schrittweise erfolgen, und lokal überprüft sein. Die Daten kann jederzeit vom Server abgerufen werden, für eine später Auswertung. Wir nutzen hierfür das Open Source Tool Limesurvey, welches wir für die medizinische Umgebung angepasst haben. Die Verbindung bei einer Umfrage und Abruf der Daten werden über SSL gesichert werden. Auch der Server ist gegen die gängigsten Angriffsarten abgesichert. Die Software kann durch das Ansible Deployment System auf jeden beliebigen Server, welcher die Voraussetzungen erfüllt, innerhalb von kurzer Zeit installiert werden. Limesurvey ist ein modulares Baukasten-System für die Erstellung der Fragebögen konfigurierbar. Wir haben es um einen weitere Fragetyp, welche es einem Teilnehmer einer Umfrage ermöglicht ein oder mehrere Körperteile anzugeben. Des weiteren haben wir eine Möglichkeit erarbeitet einer Umfrage Umweltdaten wie Wetter und Mondstand hinzufügen basierend auf dem Standort des Umfrageteilnehmers. Dies Daten können im Zusammenhang mit der Wirksamkeit eines Medikaments eine Rolle spielen. Die Umfrage sollte auf Desktop sowie gängigen Smartphone nutzbar sein.

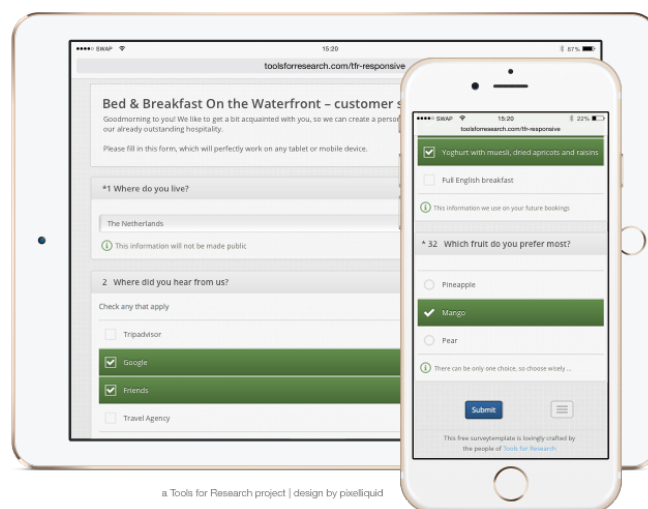


Abbildung 1: Beispiel Desktop and Mobile Umfrage

<https://www.toolsforresearch.com/sites/toolsforresearch.com/files/preview.png>

2 Grundlagen

Soll das Projekt Med-Eval in zukünftigen Semesterprojekten oder Thesis Arbeiten erweitert werden, sollte zunächst die derzeit verwendeten Technologien betrachtet werden. Diese sind eine essenzielle Anforderung an die Studenten/innen.

2.1 Kenntnisse und Fähigkeiten

Eine zentrale Fähigkeit, welche für die Weiterführung benötigt wird, ist die Fähigkeit Server Systeme (ohne Grafische Oberfläche) installieren, bedienen sowie warten zu können. Diese Fähigkeit wird bereits im ersten Semester vermittelt. Zudem ist es sehr Ratsam sich mit der Orchestrierungssoftware Ansible vertraut zu machen. Diese Software ermöglicht es den Anwender schnell und mit wenig Aufwand vorkonfigurierte Abläufe neu deployen zu können. Die Plattform Med-Eval wurde vollständig mittels Ansible automatisiert. Ratsam wäre zudem die Fähigkeit Risiken einer Software zu ermitteln und geeignete Gegenmaßnahmen ergreifen zu können. Hierbei kann jedoch in dem meisten Fällen auf Informationen im Internet zugegriffen werden. Weiterhin wäre es sehr ratsam sich mit Script Sprachen der Webentwicklung wie JavaScript oder Node.js vertraut zu machen, bzw. dort bereits Vorkenntnisse vorweisen zu können. Des weitem 2 Erweiterungen auf Systemebene mit Python geschrieben, weshalb es ratsam ist sich grundlegende Kenntnisse von Python anzueignen.

2.2 Projektunterstützende Maßnahmen

Soll das Projekt erfolgreich abgeschlossen werden, ist es Ratsam Projektunterstützende Maßnahmen zu ergreifen. Dies umfasst die Verwendung einer Versionsverwaltung zu nützen <https://github.com>. So können parallel Code Änderungen vorgenommen werden, eine Vollständige Historie des Codes erhalten so wie gezielt neue Funktionen in das existierende System aufgenommen werden. Neben der Code Verwaltung ist ein weiterer wichtiger Punkt die Team Kommunikation. Diese sollte einfach und schnell umsetzbar sein. Um dies zu erreichen, kann auf die Teamkommunikationssoftware Slack <https://slack.com> genutzt werden. Diese Software wurde erfolgreich in Firmen und Organisationen wie der NASA sowie IBM erprobt.

2.3 Entwicklungsprozess

Um einen möglichst reibungslosen Entwicklungsprozess zu erreichen sollte auf die Entwicklungskonzepte Test getriebene Entwicklung oder Agile zurückgegriffen werden. Besonders beim gewünschten Verwendungszweck der Plattform Med-Eval ist eine Entwicklung mit vielen Test sehr ratsam. So können bereits früh im Projekt wiederkehrende Fehler behoben oder gar vermieden werden. So viel in Regression Tests von Med-Eval auf, das die Download URL für jeden Releasewechsel von LimeSurvey verändert wird.

3 Realisierung

3.1 limesurvey

LimeSurvey (früher PHPSurveyor) ist eine Online-Umfrage-Software, welche unter GPL Open Source Lizenz. Sie ermöglicht es, ohne Programmierkenntnisse Online-Umfragen zu erstellen, diese zu veröffentlichen, sowie deren Ergebnisse zu speichern und für weiter Auswertung in bestimmte Formate zu exportieren.

Sie ist in PHP geschrieben und nutzt eine MySQL(auch MariaDB)-, PostgreSQL- oder MSSQL-Datenbank. Die Software bietet eine Vielzahl von Sprachen und Dialekten für die Oberfläche und Umfragen an.

3.2 limesurvey Erweiterung

3.2.1 Requirements Server

- Server mit Ubuntu Version 14.04
- Internet-Domain
- Laufender ssh Zugriff mit root Rechten per sudo
- Freischaltung von ssh, http, dns, https ins Internet, falls hostseite Firewall vorhanden ist
- ein gültiges SSL/TLS Zertifikat, jedoch ist dies ein optionale Schritt da wir per Let's encrypt, dies automatisch beziehen.

3.2.2 Requirements Limesurvey

- Apache Web mit PHP Modul
Benötigte PHP Erweiterungen:
 - mbstring (Multibyte String Funktionen)
 - PDO Datenbanktreiber für MySQL (pdo_mysql oder pdo_mysql) oder Postgres (pdo_pgsql) oder MSSQL (pdo_sqlsrv für Windows and pdo_dblib für Linux)
 - PHP-Standard-Bibliotheken (wie hash, session, etc.)

Optionale PHP Erweiterungen:

- GD-Bibliothek mit FreeType Unterstützung installiert. (Voraussetzung für CAPTCHAs oder Statistik-Graphen)
- IMAP wird für das Email bounce tracking system benötigt
- LDAP-Bibliothek (wird benötigt, um Umfrageteilnehmer über LDAP importieren zu können)
- Zip und Zlib für das ComfortUpdate

3.2.3 Statements zur Sicherheit

- Codeausführung welche nicht für Limesurvey benötigt wird, sollte nach Möglichkeit abgeschaltet werden
- Brute Force Angriffe auf HTTP, HTTPS und SSH sollen nach einer bestimmten Anzahl von Fehlversuchen unterbunden werden
- Verschlüsselung der Daten wird stark empfohlen, da Gesundheitsbereich und Datenschutzgesetz dies empfehlen, da aber keine vertretbare Lösung gefunden wurde, welche Personen mit Behinderung ausschließen würden. Sollte eine Lösung gefunden werden, die diesen Gesichtspunkt löst, kann dies in der Zukunft implementiert werden. In der Zwischenzeit wird ein Hoster empfohlen, welcher Bundesdatenschutz unterliegt.
- Da SQL nicht im gewünschten Rahmen Patches anbietet, wurde auf MariaDB umgestellt, welche sich eigenständig patchen lässt.
- Sollten keine Sicherheitsupdates oder nur sporadisch Updates für Ubuntu, Apache sowie PHP5 in der aktuell verwendeten Ubuntu Version (14.04 LTS) angeboten werden, wird eine Migration auf eine aktuelle LTS Version empfohlen. Sollte dies nicht möglich sein, sollte eine ausschließlich interner Betrieb mit keinen direkten Zugang zum Internet in Erwägung gezogen werden.
- Tests ergaben zudem, das das Setup in der aktuellen Konfiguration nicht von der OpenSSL Lücke Heartbleed <https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-0160> nicht betroffen ist.

3.2.4 Ansible Grundlagen

Ansible <https://www.ansible.com> bietet eine Open-Source Plattform, welche zur Orchestrierung und allgemeinen Konfiguration und Administration von Computern verwendet wird. Ansible kombiniert Softwareverteilung, Ad-hoc-Kommando-Ausführung sowie Konfigurationsmanagement innerhalb eines Programms. Netzwerkcomputer werden per SSH angesprochen und erfordern deshalb neben Python keine weitere Abhängigkeit. Tasks, welche sich zu Playbooks kombinieren lassen, sind in der Markup Language YAML geschrieben. Die Firma RedHat ist derzeit Sponsor des Hauptentwicklers Michael DeHaan, welcher bereits andere Server-Provisioning Software geschrieben hat. Große Firmen wie IBM oder NSA nutzen Ansible um Server zu deployen oder Updates der Umgebung aufzuspielen.

3.3 Installation mit Ansible

Soll Ansible genutzt werden, muss dies zunächst auf dem für die Orchestrierung Computer installiert werden. Je nach Betriebssystem kann dieses per Paket-Manager des OS oder mittels Python Paket-Manager pip installiert werden. Ubuntu bedarf aufgrund der meist veralteten Paketquellen einer aufwendigeren Installation.

- <https://www.digitalocean.com/community/tags/ansible?type=tutorials>
- <https://serversforhackers.com/an-ansible-tutorial>
- http://docs.ansible.com/ansible/intro_getting_started.html

3.3.1 Ubuntu PPA

```
1 $ sudo apt-get install software-properties-common
   $ sudo apt-add-repository ppa:ansible/ansible
3 $ sudo apt-get update
   $ sudo apt-get install ansible
```

3.4 Playbook

Nachdem Ansible installiert wurde, kann nun ein Playbook erstellt werden. Playbooks an sich stellen eine Ordnerstruktur dar, welche sich an einen von Ansible vorgegebenen Standard halten sollte. [Ans17]

Playbooks beinhalten verschiedene Unterordner für sogenannte Rollen, welche einen Workflow beschreiben. Diese Rollen beinhalten wiederum verschiedene Ordner. Diese sind meist tasks, templates, files sowie vars. Tasks beinhaltet ein in YAML geschriebenen Workflow, welche die auszuführenden Aktionen beschreibt. Templates beinhalten üblicherweise Dateien, welche mittels Ansible angepasst werden können, während diese an das Ziel System übertragen werden. Der Ordner files beinhaltet meist statische Dateien wie Konfigurationen. Der Ordner vars beinhaltet eine in YAML geschriebene Datei, welche Host spezifische Variablen beinhaltet, dies kann eine URL, ein Benutzername oder Namen von zu installierende Paketen sein. Ist ein Playbook erfolgreich beschrieben, kann dies mittels Ansible Konsolen Kommando ausgeführt werden. Da dies aber meist statisch aufgebaut ist, ist es zu empfehlen ein Bash Deploy Script zu schreiben, so das nur dieses ausgeführt werden muss (siehe Anhang 3.7).

Im Rahmen des Aufbaus des Environments wurden verschiedene Playbooks erstellt.

3.4.1 Installation

Die Installations Role umfasst alle Schritte um Limesurvey auf einem Ubuntu 14.04 LTS Host zu installieren. So werden Abhängigkeiten von Limesurvey installiert und die Limesurvey Software heruntergeladen. In Test viel auf, das Limesurvey keine statische Download URL anbietet. Dies führte zu Mehraufwand, da jeweils die korrekte URL von Hand gesucht und in die entsprechenden Vars eingetragen werden mussten. Um dies zu beheben, wurde ein Python Script entwickelt, welches dies übernimmt und die Software in einen vorbestimmten Ordner lädt (siehe Anhang 3.4). Ist dies geschehen, wird Limesurvey in den Apache Ordner verschoben und verschiedene Konfigurationsdateien in die entsprechenden Ordner verschoben. Es wird eine leere Datenbank sowie ein User für Limesurvey erstellt.

Die Automation über Ansible benötigt je nach Geschwindigkeit des Servers so wie dessen Internet Anbindung zwischen 2 und 5 Minuten. Eine händische Installation würde, angenommen sämtliche Schritte sind bekannt ca. 10 bis 20 Minuten benötigen.

3.4.2 Hardening

Nachdem die Installation abgelaufen ist, wird Hardening vorgenommen. Hierbei werden, zB. verschiedenste Parameter in den PHP 5 und Apache Konfigurationen geändert, um die Sicherheit zu stärken. Zudem wird die UFW Firewall installiert und Einstellungen festgelegt. Zusätzlich wird die Software fail2ban, welche verdächtige IPs automatisch sperrt, installiert. Auch wird die Software ModSecurity installiert, welche verschiedenste Attacken aufspüren und blockieren kann.

Die Automation über Ansible benötigt je nach Geschwindigkeit des Servers so wie dessen Internet Anbindung zwischen 2 und 5 Minuten. Ein händisches Hardening würde, angenommen sämtliche Schritte sind bekannt ca. 15 bis 25 Minuten benötigen.

3.4.3 Backup und Restore

Soll ein Server neu aufgebaut werden, muss zunächst ein Backup der Datenbank angefertigt und anschließend wieder Wiederhergestellt werden. Hierzu wurden 2 Roles geschrieben, welche dies automatisieren dies übernehmen.

Die Automation über Ansible benötigt je nach Geschwindigkeit des Servers so wie dessen Internet Anbindung zwischen 2 und 5 Minuten. Ein händisches Hardening würde, angenommen sämtliche Schritte sind bekannt ca. 15 bis 25 Minuten benötigen.

3.5 Hinweise zur Wartung

Wir empfehlen eine Trennung von Entwicklungssystem, Testumgebung und Produktionssystem.

- Entwicklungssystem für Entwicklung und testen neuer Technologien - spiegelt nicht Produktionssystem wieder
- Testumgebung spiegelt den Softwarestand des Produktionssystems wieder, sowie die notwendigen Änderungen aus der Entwicklung
- Produktionssystem enthält den aktuell freigegebenen Softwarestand. Änderungen dürfen nur nach Freigabe durch Test eingespielt werden. Änderungen sollen zu einem günstigen Zeitpunkt eingespielt werden.

3.5.1 Systemsoftware

Änderungen lassen sich in der Regel meist direkt übernehmen, jedoch ist ein Mindestmaß an Test durchgeführt werden. Sollte von Ubuntu 14.04 auf 16.04 oder neuer umgestellt werden, muss zunächst überprüft werden, ob sich die Paketnamen geändert haben.

3.5.2 Limesurvey und Konfigurationsdateien

Änderungen an Limesurvey sowie dazugehörige Konfigurationsdateien (Hardening) sollten mittels git versioniert werden (Tags pro Releasewechsel). Jede Änderung oder Upgrade von Limesurvey so wie den Konfigurationsdateien sollten durch alle verfügbaren Test überprüft werden. Diese Änderungen sollten nur im Entwicklungssystem vorgenommen werden. Entsprechende Releasenotes der entsprechenden Software sollten konsultiert werden. Bevor eine Änderung in das Produktionssystem übernommen wird, sollten die Benutzerdaten in geeigneter Form gesichert werden. Wird ein Major Release eingespielt werden, sollte ein mit Sample Daten in der Testumgebung erfolgen.

3.5.3 Entwicklung von Erweiterungen

Limesurvey bietet die Möglichkeit Zusatz Funktionen ins Framework mit einzubinden. Einerseits ist es möglich über Extensions, die in PHP geschrieben sind, in Limesurvey einzubinden. Zum anderen bietet sich die Möglichkeit eine Umfrage Seite direkt über JavaScript, HTML und CSS zu manipulieren. In unseren 2 Implementierungen wurde ausschließlich mit der zweiten Variante gearbeitet. Dabei wurde der Code direkt in die Seite eingefügt und nicht als externe Datei geladen.

3.5.3.1 Allgemeine Hinweise

- Für die Nutzung von JavaScript muss in Limesurvey. In den Sicherheitseinstellungen muss der XSS Schutz ausgestellt werden, da sonst die Ausführung von eingebundenen Code durch Limesurvey selbst unterbunden wird.
- Sofern möglich den Javascript Code immer in den Hilfstext Block einbinden, um Kollisionen mit dem Fragetext zu vermeiden.
- Sofern eine neue Funktion auch CSS Inhalte enthalten, sollten diese Elemente im Template Editor angefügt werden. Dabei immer auf einer Kopie arbeiten, um ein Roll-back zu vereinfachen.
- Der JavaScript Parser in Limesurvey hat leider Probleme mit `\"` und `\"`. Dadurch kommt es immer wieder zu Komplikationen beim Ausführen von JavaScript

Code. Um den Parser zufriedenstellen reicht es aus, wenn VOR und HINTER einer geschweiften Klammer jeweils ein Leerzeichen steht.

3.5.3.2 Geolocation, Wetterdaten und Mondphasen

Für die Erfassung von Geolocation und Wetter werden jQuery Konstrukte sowie die Weather Underground API genutzt. Für letztere ist ein API Key nötig, diesen kann man sich bei <https://www.wunderground.com/weather/api/>. Dieser API Code muss an betreffender Stelle eingefügt werden um Wetterdaten erfassen zu können. Die Abfrage für jeweiligen Daten läuft rein programmatisch linear, wie folgt ab: Erfassen der Geolocation, auf Basis der Geolocation wird über die Weather Underground API das Wetter am jeweiligen Ort erfasst. Die Berechnung der Mondphasen geschieht unabhängig von der Lokation. Die Formeln sowie der Source Code sind dabei aus <http://www.ben-daglish.net/moon.shtml> entnommen. Die Daten werden, sofern möglich, automatisch in die Formularfelder eingetragen. Dazu müssen die jeweiligen Formularfelder mit bestimmten IDs versehen sein. Diese sollten bei Erstellung der Frage wie folgt vergeben werden:

- ORT1 (Ort)
- WTR1 (Wetter)
- TMP1 (Temperatur)
- LFT1 (Luftfeuchtigkeit)
- MND1 (Mondphase Simple)
- MND2 (Mondphase Conway)
- MND3 (Mondphase Trig1)
- MND4 (Mondphase Trig2)

Dabei sei zu beachten: die Berechnung der Mondphasen erfolgt auf unterschiedliche Art und Weise, jedoch liefern die Berechnungen alle ein ähnliches Ergebnis. Welches man davon nutzt, ist eigentlich egal. Die Werte die dabei errechnet bilden die Mondphase wie folgt ab:

- 15 – Vollmond
- 8 – erstes Viertel
- 0 – Neumond
- 24 – letztes Viertel

Wichtig: nie den Durchschnitt von allen Formeln Berechnen und als Mondphase auswerten. Immer nur eine Berechnungsvariante wählen.

3.5.4 Körper

Der Klickbare Körper besteht aus drei einzelnen Teilen diese müssen an unterschiedlichen Stellen eingebaut werden. Das Grundgerüst selbst wird durch eine Vektorgrafik gebildet. Das funktionierende Beispiel läuft basiert auf dem Beispiel von folgender Seite. <https://www.sitepoint.com/dynamic-geo-maps-svg-jquery/>. Dabei sei zu beachten, dass die SVG Dateien für den menschlichen Körper schon vorhanden sind, jedoch nur die Außenlinien eine Funktion bringen. Ein einfacher Klick auf ein Körperteil liefert keine Reaktion des Codes. Dies sollte noch nachgebessert werden. Außerdem sollte noch Knopf implementiert werden mit dem man zwischen männlich und weiblich wechseln kann. Aus Zeitgründen ist uns die Implementierung nicht mehr gelungen. Der Code des SVG Files wird in die Frage box mit eingebunden. Und wird bei Aufruf der Frage aufgerufen und anhand der Bildschirmgröße ohne Qualitätsverlust skaliert. Der JavaScript Code wird analog zum Wetter in der Hilfe Box untergebracht. Dieser Code ist dafür verantwortlich, dass das gewählte Körperteil hervorgehoben wird und das jeweilige Körperteil in die Antwortbox geschrieben wird bzw. bei erneutem Klicken entfernt wird und die Hervorhebung rückgängig gemacht wird. Um die Hervorhebung zu erzielen, muss im Template Editor noch zusätzlicher Code in die CSS Datei *template.css* geändert werden. Hierbei nochmals der Verweis auf die allgemeinen Hinweise: Arbeiten Sie auf einer Kopie von einem Template! Die gewählten Körperteile werden in eine Antwortbox geschrieben. Nehmen Sie hierzu eine mittlere Text Box als Frage Typ.

3.5.4.1 PDFtool

Diese Funktion soll es erlauben auf Basis der, in der Datenbank gespeicherten Umfragen, PDF Dateien zur Verfügung zu stellen. Als Basis dient dabei eine Node.js Applikation. Die grundlegendes Idee ist es über das Node.js Script die Umfrage ID und den Umfragen Namen aus der Datenbank zu ziehen, diese Daten an ein Python Script zu übergeben. Das Python Script erstellt dann auch Basis das Übergeben Daten (<PARAM LISTE>) dann PDF Dateien, die den jeweiligen QR Code(der auf die Umfrage verweist) enthält. Am Schluss werden alle Dateien wie auf einem FTP Server dem Nutzer zur Verfügung gestellt. Aus Zeitlichen Engpässen ist die Implementierung dieser Funktion nur teilweise erfüllt. Funktionen wie das Python Script und die Node.js Basis sind bereits Implementiert. Was fehlt, ist die Zusammenführung von Node.js und Python Script sowie die Funktion im Node.js Code an die Daten aus der Datenbank zu kommen.

3.5.4.2 Danksagungen

Besonderer Dank gilt dem Support von Limesurvey, besonders Herrn Flür, welcher uns in der Implementierung an einigen Baustellen die passenden Hinweise zu Lösung gegeben hat. Außerdem den Limesurvey Foren welche an vielen Stellen wichtige Hinweise und Vorschläge für die Lösungsansätze geboten haben.

3.5.4.3 Ausblick

An noch vielen Punkten gibt es noch Baustellen die noch nicht ganz gelöst sind. Der menschliche Körper benötigt noch einen Umschaltmechanismus zwischen männlichem und weiblichem Körper. Außerdem muss das SVG File so überarbeitet werden, dass man nicht nur an die Ränder eines Körperteils Klicken muss, um eine Reaktion zu erhalten, sondern, wie im Karten Beispiel, mitten rein Klicken kann. Die PDF Dateien Darstellung bedarf noch der Implementierung, der Datenbankzugriff sowie Kombination von Node.js und Python Script.

3.6 Codeanhänge

Listing 3.1: body.css

```

1  .on {
2      fill: pink;
3      stroke: red;
4      stroke-width: 2;
5  }
6
7  body {
8      padding: 20px;
9  }

```

Listing 3.2: map.js

```

1  <script>
2  var allStates = $("svg.us");
3
4  allStates.on("click", "*", function() {
5
6      if($(this).hasClass("on")) {
7          $(this).removeClass("on");
8          removeFromForm($(this).attr('id'));
9      } else {
10         $(this).addClass("on");
11         writeToForm($(this).attr('id'));
12     }
13
14 } );
15
16 function writeToForm(name) { document.getElementById(
17     document.querySelector('[id^="answer"]').id).value += name
18     +";";
19 }
20
21 function removeFromForm(name) {
22
23     var content = document.getElementById(document.
24         querySelector('[id^="answer"]').id).value;
25     var outPutContent = content.replace(name+";", "");
26     document.getElementById(document.querySelector('[id^="
27         answer"]').id).value = outPutContent;

```

```
24
25 }
26
27 </script>
```

Listing 3.3: wetter.js

```
1 < p id = "location" > < /p>
2
3 <
4   p id = "city" > < /p>
5
6 <
7   p id = "weather" > < /p>
8
9 <
10  p id = "temperature" > < /p>
11
12 <
13  p id = "humidity" > < /p>
14
15 <
16  p id = "mondphasen" > < /p>
17
18 <
19  p id = "simpleMoon" > < /p>
20
21 <
22  p id = "conwayMoon" > < /p>
23
24 <
25  p id = "trig1Moon" > < /p>
26
27 <
28  p id = "trig2Moon" > < /p> <
29    script type = "text/javascript" >
30      // loading functions
31      $(document).ready(function() {
32        getLocation();
33      });
34
35 // var x = document.getElementById("location");
36 // var longitude;
```

```
37 // var latitude;
38
39 function getLocation() {
40     if (navigator.geolocation) {
41         navigator.geolocation.getCurrentPosition(showPosition,
42             );
43     } else {
44         x.innerHTML = "Geolocation is not supported by this
45             browser.";
46     }
47 }
48 // writing coordinates into html file// only for debug
49 reasons...
50
51 function showPosition(position) {
52     // x.innerHTML = "Latitude: " + position.coords.latitude
53     +
54     // "<br>Longitude: " + position.coords.longitude;
55     longitude = position.coords.longitude;
56     latitude = position.coords.latitude;
57     weatherFunction(longitude, latitude);
58     getMoonPhases();
59 }
60
61
62 function weatherFunction(longitude, latitude) {
63     var APIKEY = "YOUR API KEY";
64     var weatherURL;
65     var cityVar;
66     jQuery(document).ready(function($) {
67         $.ajax({
68             // Insert API key here, replace <APIKEY> with
69             your API key
70             url: "https://api.wunderground.com/api/" + APIKEY
71                 + "/geolookup/q/" + latitude + "," +
72                 longitude + ".json",
73             dataType: "jsonp",
74             success: function(parsed_json) {
75                 var location = parsed_json['location']['l'];
76                 city = parsed_json['location']['city'];
77                 // var xC = document.getElementById("city");
78                 // xC.innerHTML = "Das Wetter fuer " + city;
```

```
71         document.getElementById(document.
           querySelector(' [id^="answer"][id$="ORT1"] '
           ).id).value = city;
72         weatherURL = "https://api.wunderground.com/
           api/" + APIKEY + "/conditions/lang:DL/q/"
           + location + ".json";
73         getWetter(weatherURL)
74     }
75
76     });
77 });
78 }
79
80 function getWetter(link) {
81
82     var wearherVar;
83     var temperatureVar;
84     var humidityVar;
85     jQuery(document).ready(function($) {
86         $.ajax({
87             url: link ,
88             dataType: "jsonp",
89             success: function(parsed_json) {
90
91                 weatherVar = parsed_json['current_observation
                    ']['weather'];
92                 temperatureVar = parsed_json['
                    current_observation']['temp_c'];
93                 humidityVar = parsed_json['
                    current_observation']['relative_humidity'
                    ];
94
95
96                 document.getElementById(document.
                   querySelector(' [id^="answer"][id$="WTR1"] '
                   ).id).value = weatherVar;
97                 document.getElementById(document.
                   querySelector(' [id^="answer"][id$="TMP1"] '
                   ).id).value = temperatureVar;
98                 document.getElementById(document.
                   querySelector(' [id^="answer"][id$="LFT1"] '
                   ).id).value = humidityVar;
99             }
100         });
101     });
102 }
```

```

        ).id).value = humidityVar;
99      // var xW = document.getElementById("weather▷
        ");
100     // var xT = document.getElementById("▷
        temperature");
101     // var xH = document.getElementById("humidity▷
        ");
102
103     // xW.innerHTML = "Das wetter ist : "+▷
        weatherVar;
104     // xT.innerHTML = "Die Temperatur in Celsius ▷
        : "+temperatureVar;
105     // xH.innerHTML = "Die luftfeuchtigkeit ▷
        betraegt : "+humidityVar;
106     }
107     });
108
109     });
110 }
111
112 function getMoonPhases() {
113     //var date = new Date();
114     var date = new Date(2016, 01, 24, 20, 35, 0);
115     var year = date.getFullYear(); // getYear —> y2k bug
116     var month = date.getMonth();
117     var day = date.getDate();
118
119
120
121     // var xM = document.getElementById("mondphasen");
122     // var xSM = document.getElementById("simpleMoon");
123     // var xCM = document.getElementById("conwayMoon");
124     // var xT1M = document.getElementById("trig1Moon");
125     // var xT2M = document.getElementById("trig2Moon");
126
127
128     // xM.innerHTML = "Mondphasen:"
129     document.getElementById(document.querySelector(' [id^="▷
        answer" ] [id$="MND1" ] ').id).value = Simple(year, month, ▷
        day);
130     document.getElementById(document.querySelector(' [id^="▷

```



```

        answer"[id$="MND2"]').id).value = Conway(year, month, day);
131 document.getElementById(document.querySelector('[id$="MND3"]').id).value = Trig1(year, month, day);
        answer"[id$="MND3"]').id).value = Trig1(year, month, day);
132 document.getElementById(document.querySelector('[id$="MND4"]').id).value = Trig2(year, month, day);
        answer"[id$="MND4"]').id).value = Trig2(year, month, day);
133 // xSM.innerHTML = "Simple "+Simple(year, month, day);
134 // xCM.innerHTML = "Conway "+Conway(year, month, day);
135 // xT1M.innerHTML = "Trig1 "+Trig1(year, month, day);
136 // xT2M.innerHTML = "Trig2 "+Trig2(year, month, day);
137
138 // 15 —> Vollmond
139 // 8 —> erstes Viertel
140 // 0 —> Neumond
141 // 24 —> letztes Viertel
142
143 }
144
145 function Simple(year, month, day) {
146     var lp = 2551443;
147     var now = new Date(year, month - 1, day, 20, 35, 0);
148     var new_moon = new Date(1970, 0, 7, 20, 35, 0);
149     var phase = ((now.getTime() - new_moon.getTime()) / 1000)
        % lp;
150     return Math.floor(phase / (24 * 3600)) + 1;
151 }
152
153 function Conway(year, month, day) {
154     var r = year % 100;
155     r %= 19;
156     if (r > 9) {
157         r -= 19;
158     }
159     r = ((r * 11) % 30) + parseInt(month) + parseInt(day);
160     if (month < 3) {
161         r += 2;
162     }
163     r -= ((year < 2000) ? 4 : 8.3);
164     r = Math.floor(r + 0.5) % 30;

```

```

165     return (r < 0) ? r + 30 : r;
166 }
167
168 function Trig1(year, month, day) {
169     var thisJD = julday(year, month, day);
170     var degToRad = 3.14159265 / 180;
171     var K0, T, T2, T3, J0, F0, M0, M1, B1, oldJ;
172     K0 = Math.floor((year - 1900) * 12.3685);
173     T = (year - 1899.5) / 100;
174     T2 = T * T;
175     T3 = T * T * T;
176     J0 = 2415020 + 29 * K0;
177     F0 = 0.0001178 * T2 - 0.000000155 * T3 + (0.75933 +
        0.53058868 * K0) - (0.000837 * T + 0.000335 * T2);
178     M0 = 360 * (GetFrac(K0 * 0.08084821133)) + 359.2242 -
        0.0000333 * T2 - 0.00000347 * T3;
179     M1 = 360 * (GetFrac(K0 * 0.07171366128)) + 306.0253 +
        0.0107306 * T2 + 0.00001236 * T3;
180     B1 = 360 * (GetFrac(K0 * 0.08519585128)) + 21.2964 -
        (0.0016528 * T2) - (0.00000239 * T3);
181     var phase = 0;
182     var jday = 0;
183     while (jday < thisJD) {
184         var F = F0 + 1.530588 * phase;
185         var M5 = (M0 + phase * 29.10535608) * degToRad;
186         var M6 = (M1 + phase * 385.81691806) * degToRad;
187         var B6 = (B1 + phase * 390.67050646) * degToRad;
188         F -= 0.4068 * Math.sin(M6) + (0.1734 - 0.000393 * T)
            * Math.sin(M5);
189         F += 0.0161 * Math.sin(2 * M6) + 0.0104 * Math.sin(2
            * B6);
190         F -= 0.0074 * Math.sin(M5 - M6) - 0.0051 * Math.sin(
            M5 + M6);
191         F += 0.0021 * Math.sin(2 * M5) + 0.0010 * Math.sin(2
            * B6 - M6);
192         F += 0.5 / 1440;
193         oldJ = jday;
194         jday = J0 + 28 * phase + Math.floor(F);
195         phase++;
196     }
197     return (thisJD - oldJ) % 30;

```

```

198 }
199
200 function GetFrac(fr) {
201     return (fr - Math.floor(fr));
202 }
203
204 function Trig2(year, month, day) {
205     n = Math.floor(12.37 * (year - 1900 + ((1.0 * month - 0.5) / 12.0)));
206     RAD = 3.14159265 / 180.0;
207     t = n / 1236.85;
208     t2 = t * t;
209     as = 359.2242 + 29.105356 * n;
210     am = 306.0253 + 385.816918 * n + 0.010730 * t2;
211     xtra = 0.75933 + 1.53058868 * n + ((1.178e-4) - (1.55e-7) * t) * t2;
212     xtra += (0.1734 - 3.93e-4 * t) * Math.sin(RAD * as) - 0.4068 * Math.sin(RAD * am);
213     i = (xtra > 0.0 ? Math.floor(xtra) : Math.ceil(xtra - 1.0));
214     j1 = julday(year, month, day);
215     jd = (2415020 + 28 * n) + i;
216     return (j1 - jd + 30) % 30;
217 }
218
219 function julday(year, month, day) {
220     if (year < 0) {
221         year++;
222     }
223     var jy = parseInt(year);
224     var jm = parseInt(month) + 1;
225     if (month <= 2) {
226         jy--;
227         jm += 12;
228     }
229     var jul = Math.floor(365.25 * jy) + Math.floor(30.6001 * jm) + parseInt(day) + 1720995;
230     if (day + 31 * (month + 12 * year) >= (15 + 31 * (10 + 12 * 1582))) {
231         ja = Math.floor(0.01 * jy);
232         jul = jul + 2 - ja + Math.floor(0.25 * ja);

```

```

233     }
234     return jul;
235 } <
236 /script>

```

Listing 3.4: URL Parsing

```

1  import os
2  import urllib
3
4  from BeautifulSoup import BeautifulSoup
5
6  limesurvey_download_page = 'https://www.limesurvey.org/de/↵
    stable-release'
7  limesurvey_download_base_url = 'https://www.limesurvey.org'
8  archive_file_name = 'limesurvey.tar'
9
10 path = os.path.join("/tmp", archive_file_name)
11
12 page = urllib.urlopen(limesurvey_download_page).read()
13 soup = BeautifulSoup(page)
14
15 url = []
16 for div_class in soup.findAll('div', {"class": "uk-width ↵
    -1-1"}):
17     for a_class in div_class.findAll('a', {"class": "btn btn-↵
        success"}):
18         url.append(a_class['href'])
19
20 for element in url:
21     if 'targz' in element:
22         limesurvey_download_base_url += element
23         break
24
25 urllib.urlretrieve(limesurvey_download_base_url, path)

```

Listing 3.5: PDF Creation

```
1  """
2  This Python script creates a pdf based on the given title and
    url
3
4  """
5  import os
6  import sys
7
8  from qrcode import *
9  from reportlab.lib.pagesizes import A4
10 from reportlab.lib.styles import getSampleStyleSheet
11 from reportlab.lib.units import cm
12 from reportlab.platypus import SimpleDocTemplate, Paragraph,
    Spacer, Image
13
14
15 def create_qrcode():
16     """
17     create_qrcode creates a qr code based on the given title
18     :return:
19     """
20     try:
21         survey_title = sys.argv[1]
22         survey_url = sys.argv[2]
23     except IndexError:
24         print("Please check your input of the title")
25         sys.exit(1)
26
27     qr = QRCode(version=20, error_correction=ERROR_CORRECT_H)
28
29     try:
30         qr.add_data(survey_url)
31     except UnboundLocalError:
32         print("Please check your input of the title")
33         sys.exit(1)
34
35     qr.make() # Generate the QRCode itself
36
37     # im contains a PIL.Image.Image object
38     im = qr.make_image()
```

```

39
40     # To save it
41     im.save(survey_title + ".png")
42
43     return survey_title + ".png"
44
45
46 def create_pdf(logo_name):
47     """
48     create_pdf creates a pdf based on the before created ↵
49     qrcode as png image.
50     :return:
51     """
52
53     try:
54         pdf_name_with_suffix = sys.argv[1] + ".pdf"
55         pdf_name = sys.argv[1]
56
57     except IndexError:
58         print("Please check your input of the title")
59         sys.exit(1)
60
61     try:
62         survey_url = sys.argv[2]
63     except IndexError:
64         print("Please check your input of the url")
65         os.remove(logo_name)
66         sys.exit(1)
67
68     created_pdf = SimpleDocTemplate(pdf_name_with_suffix, ↵
69                                     pagesize=A4,
70                                     rightMargin=72, ↵
71                                     leftMargin=72,
72                                     topMargin=72, ↵
73                                     bottomMargin=18)
74
75     story = []
76     logo = logo_name
77
78     style = getSampleStyleSheet()['Normal']
79     style.wordWrap = 'LTR'
80     style.leading = 24

```

```

76     style.fontSize = 24
77
78     title = '<para alignment="center"><font size=24>' + >
           pdf_name + '</font>'
79     story.append(Paragraph(title, style))
80     story.append(Spacer(1, 150))
81     im = Image(logo, 10 * cm, 10 * cm)
82     story.append(im)
83     story.append(Spacer(1, 150))
84
85     story.append(Paragraph(survey_url, style))
86     created_pdf.build(story)
87
88     os.remove(logo_name)
89
90
91 def main():
92     """
93     main method
94     :return: none
95     """
96     logo_name = create_qrcode()
97     create_pdf(logo_name)
98
99 if __name__ == "__main__":
100     main()

```

Listing 3.6: Requirements

```

1  qrcode
2  reportlab==3.3.0
3  BeautifulSoup==3.2.1

```

Listing 3.7: Bash Deploy

```

1  ansible-playbook -i hosts deploy.yml -k -u root

```


Literaturverzeichnis

[Ans17] ANSIBLE: *Documentation*. http://docs.ansible.com/ansible/playbooks_best_practices.html#directory-layout. Version: 1 2017

Eidesstattliche Erklärung

Wir versichern, dass wir die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet haben. Alle Stellen der Arbeit die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Uns ist bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

, den 17.02.2017 Axel Butz, Eugen Jastremskoj, Marcin Krawczyk, Jan Ondruch, Jan Stodt