

Protokolleintrag vom: 27.10.16	Arbeitsaufwand: 4 Stunden
Erste Besprechung und Brainstorming	
Nachbereiten der W2 Vorlesung vom Montag den 26.09.2016. Erstes informieren über Serverseitige Programmiersprachen. Erste Auswahl besteht aus der Programmiersprachen besteht aus: Python, PHP, Ruby, Java. Diskussion über die einzelnen Programmiersprachen mit Kommilitonen. Einrichten von GitHub zum sharen von Code. Vor- und Nachteile wurden besprochen. Recherche zu einzelnen Webframeworks. Festlegen von wöchentlichen Treffen. Erste Projektideen werden besprochen	

Protokolleintrag vom: 4.11.16	Arbeitsaufwand: 3 Stunden
Weitere Recherche und Brainstorming	
Weiteres informieren über Programmiersprachen. Python und Java rücken in den Fokus. Diskussionen über Python und Java. Tim-Alexander Biba schlägt Python vor, da er mit Java schon Erfahrung hat und Java ihm kein Spaß gemacht hat. Als weiteres Argument führt er an, dass Python auch für Scriptprogrammierung benötigt wird. Weitere Recherchen bzgl. Webframeworks von Python und Java. Gegenüberstellung von Spring MVC, JSF und Vaadin von Java. Bei Python werden Django und Flask herausgesucht. Projektideen werden auch besprochen.	
<a href="https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html">https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html</a> <a href="http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html">http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html</a> <a href="https://vaadin.com/home">https://vaadin.com/home</a> <a href="https://www.djangoproject.com">https://www.djangoproject.com</a> <a href="http://flask.pocoo.org">http://flask.pocoo.org</a>	

Protokolleintrag vom: 14.11.16	Arbeitsaufwand: 3 Stunden
Entscheidung für Python	
Nach anfänglicher Diskussion mit anderen Kommilitonen in der Fachschaft über ihre Erfahrungen bei Webprogrammierung wurde sich für Python entschieden. Tim-Alexander Biba schlägt weitere Python Frameworks vor: Bottle und Pyramid. Dies werden auch gemeinsam Recherchiert und diskutiert. Beide sind aber nicht geeignet.	
<a href="https://bottlepy.org/docs/dev/">https://bottlepy.org/docs/dev/</a> <a href="http://docs.pylonsproject.org/projects/pyramid/en/latest/">http://docs.pylonsproject.org/projects/pyramid/en/latest/</a>	

Protokolleintrag vom: 24.11.16	Arbeitsaufwand: 4 Stunden
Informieren über weitere Tools	
Recherche über Texteditoren und IDEs. Die Wahl fällt auf PyCharm bei Moritz Müller und Tim-Alexander Biba. Einlesen in Github und Bitbucket sowie Diskussion über einheitliche Entwicklungsumgebung. Moritz Müller möchte Vagrant benutzen um einen gemeinsamen Unterbau für den Code zu haben. Projektidee wird auf einen Hotel-Club festgelegt.	

<https://help.github.com>  
<https://bitbucket.org>  
<https://www.vagrantup.com>  
<https://www.jetbrains.com/pycharm/>

Protokolleintrag vom: 29.11.16	Arbeitsaufwand: 4 Stunden
Entscheidung für Django	
Nach anfänglicher Diskussion über aufsetzen von Entwicklungsumgebung haben wir uns für Django als Framework entschieden. Aufsetzen der Entwicklungsumgebung mit PyCharm und Vagrant bei Moritz Müller. Tim-Alexander Biba benutzt PyCharm ohne Vagrant. Einlesen in Virtuelle Environments zum Abgrenzen der Python Pakete. Erste Programmiersuche mit Python.	
<a href="http://docs.python-guide.org/en/latest/dev/virtualenvs/">http://docs.python-guide.org/en/latest/dev/virtualenvs/</a>	

Protokolleintrag vom: 9.12.16	Arbeitsaufwand: 3 Stunden
Programmieren in Python und Diskussion über Datenbanken	
Weiteres Lernen von Python. Diskussionen im Team über möglich Datenbanken die benutzt werden sollen sowie Recherche zu den einzelnen. Mögliche Kandidaten sind PostgreSQL, MySQL, MariaDB, SQLITE3. Heraussuchen von Software zur Verwaltung der Datenbanken. Die Wahl fällt hier auf PostgreSQL-Studio, da dies bereits aus dem Modul Datenbanken bekannt ist.	
<a href="https://www.postgresql.org/docs/">https://www.postgresql.org/docs/</a> <a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a> <a href="https://www.sqlite.org/docs.html">https://www.sqlite.org/docs.html</a>	

Protokolleintrag vom: 15.12.16	Arbeitsaufwand: 5 Stunden
Erstes Programmieren mit Django	
Erstes Programmieren mit Django mithilfe eines YouTube-Tutorials. Schauen der Videos gemeinsam und anschließend erste Versuche mit der Django Admin Seite und Django Login. Als Datenbank wird nun PostgreSQL festgelegt. Anbindung der Datenbank wird realisiert. Entwerfen der Datenbank in der dritten Normalform auf Papier. Einpflegen der Datenbank in PostgreSQL.	
<a href="https://www.youtube.com/watch?v=FNQxxpM1yOs&amp;list=PLQVvva0QuDeA05ZouE4OzDYLHY-XH-Nd">https://www.youtube.com/watch?v=FNQxxpM1yOs&amp;list=PLQVvva0QuDeA05ZouE4OzDYLHY-XH-Nd</a>	

Protokolleintrag vom: 20.12.16	Arbeitsaufwand: 3 Stunden
Programmieren mit Django; Erste Zweifel.	
Erstellen erster Views und Models für die Webanwendung. Einlesen in HTML zum gestalten erster HTML-Templates. Diskussionen im Team und mit Kommilitonen ob Django nicht doch zu einfach, da kaum eigener Code	

von Nöten ist zum Erstellen der Webanwendung und da einzelne Codezeilen in Framework-Code kopiert werden müssen.

<https://selfhtml.org>

Protokolleintrag vom: 5.1.17

Arbeitsaufwand: 4 Stunden

Umschwenken auf Flask Framework und ändern der Datenbank

Neustart des Projektes mit Flask, da Django zu viel zusätzliche Funktionen mitbringt die nicht in der Prüfungsdiskussion erklärbar wären. Einlesen in Flask und ändern der Datenbank in SQLITE, auf Anraten von Kommilitonen. Erstellen der Projektstruktur. Erstes Programmieren in Flask mit Hilfe des Mega Tutorials von Miguel Grinberg. Probleme mit Git, wegen Änderungen am Remote Repository.

<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>

Protokolleintrag vom: 9.1.17

Arbeitsaufwand: 3 Stunden

Erstellen der Datenbank und suche nach Flask-Erweiterungen

Erstellen der Datenbank mit Hilfe von SQLite Studio. Erste Versuche die Datenbank mit SQLAlchemy anzusprechen. Suche nach weiteren Flask-Erweiterungen wie Flask-WTF und Flask-Login. Einlesen in die Erweiterungen. Weiteres Tutorial wird zu Hilfe gezogen. Umstellen der Projekt Struktur um die Entwicklung zu vereinfachen.

<https://sqlitestudio.pl/index.rvt>

<https://flask-wtf.readthedocs.io/en/stable/>

<https://flask-login.readthedocs.io/en/latest/>

[https://www.youtube.com/watch?v=gDSLrpxR3G4&list=PLei96ZX\\_m9sWQco3fwfSMqyGL-JDQo28l](https://www.youtube.com/watch?v=gDSLrpxR3G4&list=PLei96ZX_m9sWQco3fwfSMqyGL-JDQo28l)

Protokolleintrag vom: 9.1.17

Arbeitsaufwand: 3 Stunden

Herunterladen von Bootstrapp und Einlesen von Jinja2 Templating

Herunterladen von Bootstrap da dies schneller geht um Templates zu erstellen. Aufteilen der Templates in base.html in login.html und register.html. Versuche mit Flask-WTForms Daten in die Datenbank bei der Registrierung zu schreiben. Klappt aktuell noch nicht. Tim-Alexander Biba versucht alternativ eine Lösung mit Hilfe von JQuery zu erarbeiten.

Protokolleintrag vom: 13.1.17

Arbeitsaufwand: 2 Stunden

Jinja2 Templating und Datenbank Probleme

Es lassen sich weiterhin keine Daten in die Datenbank einlesen. Es wird an den HTML-Templates weitergearbeitet und parallel mit JQuery und WTForms versucht eine ein Formular für die Registrierung von Benutzern zum Laufen zu bekommen.

Protokolleintrag vom: 19.1.17	Arbeitsaufwand: 4 Stunden
Datenbankprobleme und wechseln zu MariaDB	
Vermutung es liegt an der falschen Datenbank Formatierung, da alle in Tutorials *.db Dateien benutzen und wir eine *.sqlite Datei. Installation von XAMPP zum Prüfen ob es mit einer anderen Datenbank wie MariaDB besser klappt. Anfangen mit der Erstellung der Dokumentation.	
<a href="https://www.apachefriends.org/de/index.html">https://www.apachefriends.org/de/index.html</a>	

Protokolleintrag vom: 23.1.17	Arbeitsaufwand: 6 Stunden
Neustart des Projektes mit PHP	
Neustart des Projektes mit PHP, da kein Bestehen der Prüfung mit der Flask-Webanwendung möglich wäre. PHP lernen mit Hilfe eines Tutorials. Übertragen der HTML-Templates und gemeinsames Programmieren der Index.php. Lernen des PHP Aufbaus und Syntax. Lernen der Anbindung der Datenbank(PDO und Prepared Statements). Alte Flask MariaDB wird weiterbenutzt.	
<a href="http://php-einfach.de">http://php-einfach.de</a>	

Protokolleintrag vom: 23.1.17	Arbeitsaufwand: 5 Stunden
Entwicklung von Login, Logout	
Erste Datenbankverbindung steht. Gemeinsame Entwicklung der register.php. Anfangen der functions.inc.php. Aufteilen der Arbeiten in Coden und Dokumentation. Erstes Entwickeln von Registrieren und Login und Logout.	
<a href="http://php-einfach.de">http://php-einfach.de</a>	

Protokolleintrag vom: 26.1.17	Arbeitsaufwand: 6 Stunden
Arbeit an Sessions und Login/Logout	
Versuche erste Sessions zu realisieren. Gemeinsames Bugfixing und gemeinsame Änderungen am HTML-Layout. Weiterarbeit an Login und Logout.	
<a href="http://php-einfach.de">http://php-einfach.de</a>	

Protokolleintrag vom: 28.1.17	Arbeitsaufwand: 8 Stunden
Weiterarbeit an Sessions und Hashen des Passwortes.	

Weiterarbeit mit Hilfe des Tutorials. Erarbeiten von Securitytokens und einfügen von Sessions. Hashen des Passwortes und Weiterarbeit an functions.inc.php. Nach Login sollen den Benutzern Hotels zum Vergleichen angezeigt werden. Auch Funktionen zum Reset von Passwort, Email und Benutzername hinzugefügt. Erste arbeiten an internal.php. Erste Vorbereitungen zum Erarbeiten der tipp.php.

<http://php-einfach.de>

Protokolleintrag vom: 29.1.17

Arbeitsaufwand: 6 Stunden

HTML-Feintuning und hinzufügen von Funktionen

Gemeinsames Erweitern der HTML-Templates und Änderungen am Design der Webanwendung. Weiterarbeiten an internal.php und tipp.php. Gemeinsame Änderungen an der Datenbank, hinzufügen von der Tipp Tabellen und Änderungen von Beziehungen.

<http://php-einfach.de>

Protokolleintrag vom: 30.1.17

Arbeitsaufwand: 4 Stunden

Erstellen der Präsentation und Benutzerhandbuch

Letze Änderungen an der Webanwendung. Letzte Änderungen an der Tippfunktion. Gegenlesen der Rohfassung der Dokumentation.