

ABSCHLUSSPROJEKT WEBAPP-ENGINEERING II DOKUMENTATION



Entwicklung World Wild Wiki

Teilnehmer in dem Projekt sind

Kevin Wagner / Marcel Anders / Luis Eckert

Betreuender Dozent: Herr Bernhard Geisberger

Zusammenfassung

In der folgenden Dokumentation geht es um ein Web-Wiki (World Wild Wiki), auf dem verschiedene Communitys ihre eigenen Wikis erstellen und bearbeiten können. Jeder User kann dabei Eigentümer/Owner seines eigenen Wikis werden und auch andere User in das entsprechende Wiki einladen. Genauso können außen stehende User Anfragen an ein Wiki stellen, um diesem beizutreten.

Die Grundlegende Idee ist es, den verschiedensten Communitys einen Platz zum Austausch von Informationen, Tipps, Geschichten, etc. zu geben. Diese Plattform soll durch seine Funktionsweise ein Platz für kollaboratives Schreiben, Spiele / Film / Musik / etc. Wikis und ähnliches bieten.

Inhaltsverzeichnis

1	Einführung	2
2	Projektbeschreibung	2
3	Projektumgebung Windows Subsystem Linux (WSL)	2
4	Projektplanung	3
4.1	Business Model Canvas	3
4.1.1	Value Proposition Canvas	4
4.2	Verhaltensdiagramme	5
4.2.1	Strukturdiagramm Article Dashboard	5
4.2.2	Strukturdiagramm Login	6
4.3	Moqups	7
4.3.1	UI Moqup - Main Page	7
4.3.2	UI Moqup - Wiki	8
5	Projekt World Wild Wiki	9
5.1	Entwicklungsumgebung	9
5.2	Symfony PHP Framework	9
5.2.1	Templatestruktur	9
5.2.2	Benutzergruppen und Funktionalitäten	10
5.3	Demo-Wiki	10
5.3.1	Wiki-Arten	11
6	Projekt und Quellcode	11
7	Projekt Installation	12
7.1	Webserver WSL	12
7.1.1	Apache2	12
7.1.2	MYSQL	12
7.1.3	PHP	13
7.1.4	phpMyAdmin	13
7.1.5	Composer, Zip & Unzip	14
7.2	Symfony Projekt	14
7.2.1	Demo-Daten	15
8	Projekt-Fazit	15

1 Einführung

Die folgende Dokumentation wurde für das Abschlussprojekt des Faches Web-App-Engineering II erstellt. Es geht, um die Erstellung eines Web-Wikis, auf dem verschiedene Communitys ihre eigenen Wikis erstellen und bearbeiten können. Jeder User kann dabei Eigentümer/Owner seines eigenen Wikis werden und auch andere User in das entsprechende Wiki einladen. Genauso können außen stehende User Anfragen an ein Wiki stellen, um diesem beizutreten.

Es wird die grundlegende Struktur der Webanwendung und eine Einführung in deren Struktur beschrieben. Des Weiteren lässt sich eine Installationsanleitung im unteren Bereich eben dieser Dokumentation finden, damit das bestehende Projekt aufgesetzt werden kann. Die hierfür benötigten Dateien sind in einem Github-Repository (Main-Branch) zu finden. Den Link zu dem Repository ist:

https://github.com/kevin220901/Project_Web-App-Engineering_II.git

In der README.md ist auch noch einmal eine ausführliche Beschreibung der Funktionalitäten und eine grundlegende Einrichtung des Windows Subsystem für Linux Webservers zu finden.

2 Projektbeschreibung

Es soll eine öffentlich zugängliche Plattform entstehen, bei denen sich diverse Gruppierungen und Communitys kostenlos austauschen und Ihren Wissensdurst stillen können. Die Plattform WorldWildWiki soll eine Möglichkeit hierzu bieten. Durch das Erstellen von Themen spezifischen Wikis, kann der spätere Anwender Informationen zu den verschiedensten Bereichen finden, sich Austauschen oder Weiterbilden. Das Projekt wird in Symfony umgesetzt.

3 Projektumgebung Windows Subsystem Linux (WSL)

Das Windows Subsystem Linux (WSL) ist in dem Windows Betriebssystem eine Kompatibilitätsschicht zum Ausführen von Linux Systemen. Mithilfe des WSL können Entwickler eine GNU-/Linux-Umgebung (einschließlich der

meisten Befehlszeilentools, Hilfsprogramme und Anwendungen) direkt unter Windows unverändert ausführen, ohne den Mehraufwand eines traditionellen virtuellen Computers oder eines Dual-Boot-Setups betreiben zu müssen.

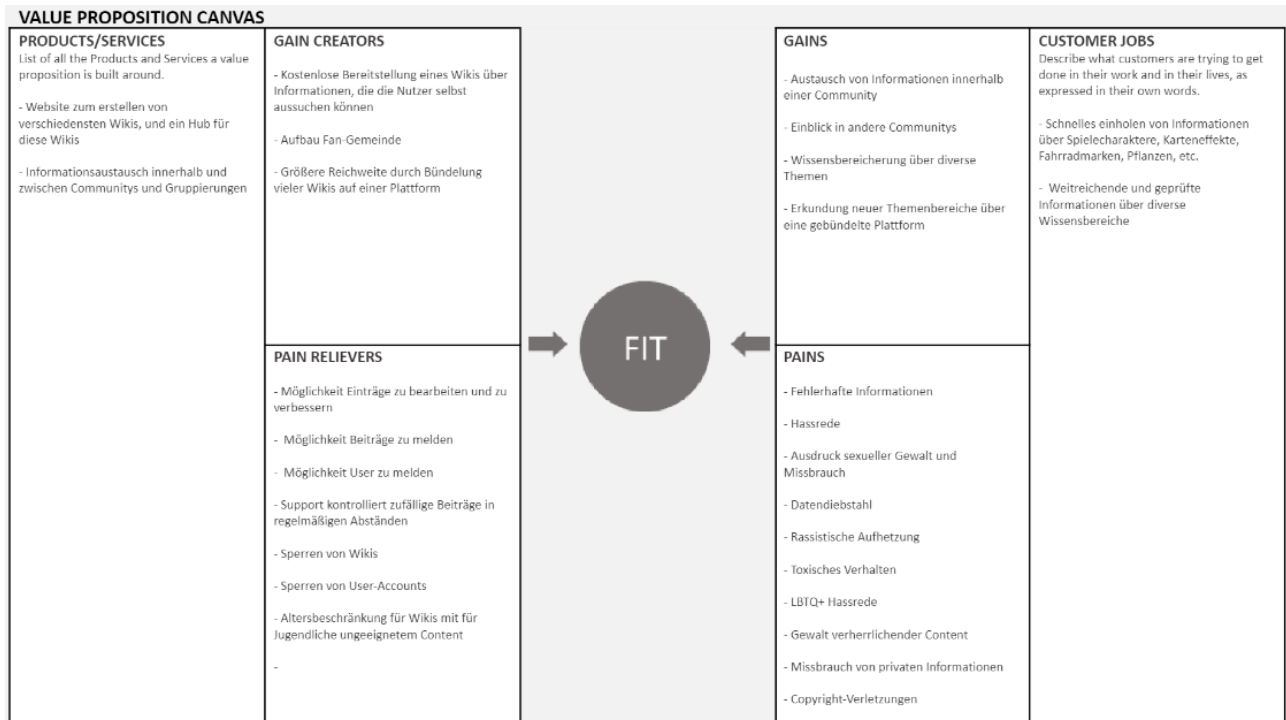
In dem Projekt wird das Subsystem für das Aufsetzen eines Webservers verwendet. Auf diesem Webserver soll später das PHP-Framework Symfony ausgeführt und das Web-Wiki gehostet werden.

4 Projektplanung

4.1 Business Model Canvas

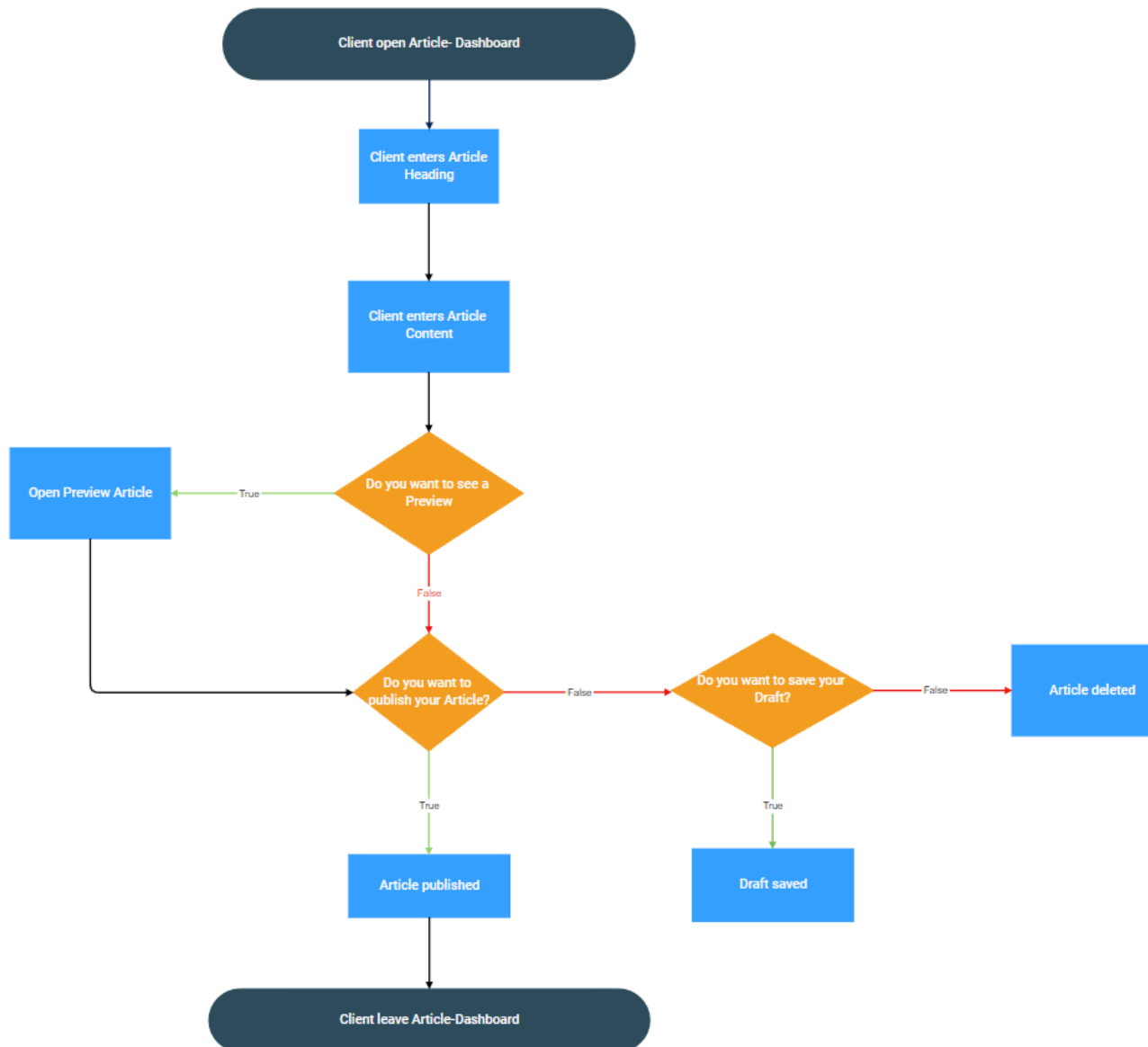
Business Model Canvas		Entwickelt für:	Entwickelt von:	Datum:	Version:
		Bernhard Geisberger	Luis Eckert, Kevin Wagner, Marc	14/04/2022	1.0
Schlüsselpartner	Schlüsselaktivitäten	Wertversprechen	Kundenbeziehungen	Kundensegmente	
<ul style="list-style-type: none">- Azure Cloud- Online-Donation-System (z.b. Paypal)	<ul style="list-style-type: none">- Moderation der Wikis & User Support- Plattform Entwicklung- Copyright Management	<p>Communitys:</p> <ul style="list-style-type: none">- Sammelstelle für Informationen und Geschichten- Positive Feedback (nur Upvotes möglich)- Wissensvermittlung- Vertrauenswürdig auf Richtigkeit der Inhalte- keine Weitergabe personenbezogener Daten- keine Werbemails	<ul style="list-style-type: none">- Kundenservice / Support (Mail, Telefon)- Datenschutz & - Sicherheit- Chatbot- Self-Service auf der Plattform (FAQ)	<p>Communitys:</p> <ul style="list-style-type: none">- Die Informationen und Tipps zu den verschiedensten Themen mit der Welt teilen möchten. <p>User:</p> <ul style="list-style-type: none">- Welche Fragen zu Charakteren, Spielzügen, Pflanzen, Diäten, etc. haben und diese auf der Website finden möchten.	
	Schlüsselressourcen		Marktkanäle		
	<ul style="list-style-type: none">- Plattform (Website)- Marke (World-Wild-Wiki)- Informationen		<ul style="list-style-type: none">- Website: world-wild-wiki.com- Social Media Marketing		
Kosten		Einnahmen			
<ul style="list-style-type: none">- Entwicklung & Wartung der Plattform- Webhosting (Domain, Webspace und Datenbank) / monatliche Zahlweise- Supportteam		<ul style="list-style-type: none">- Schalten von Werbung: 50%- Sammeln von Spenden (Wikipedia-Prinzip): 25%- Sondereinnahmen: 2%			

4.1.1 Value Proposition Canvas

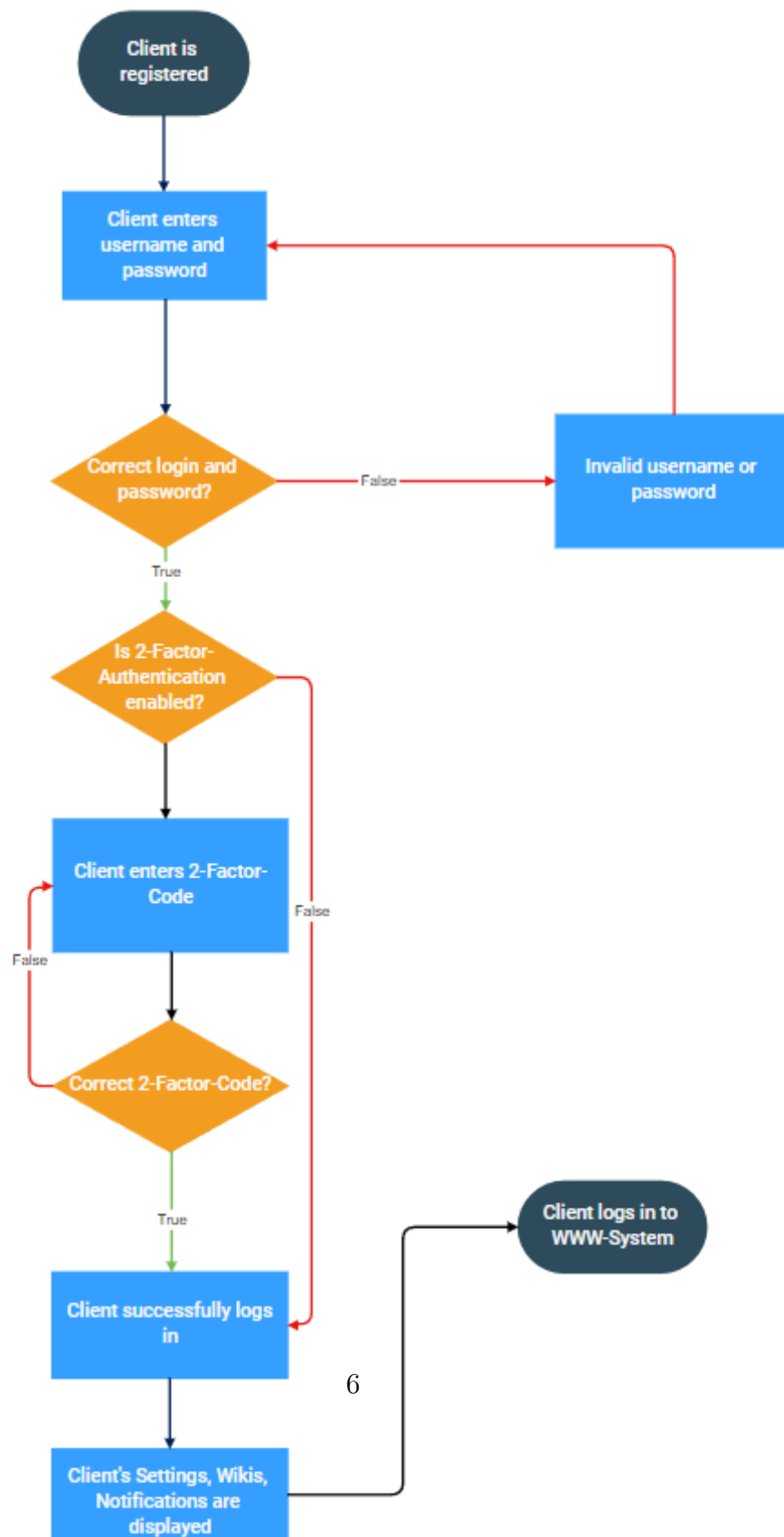


4.2 Verhaltensdiagramme

4.2.1 Strukturdiagramm Article Dashboard

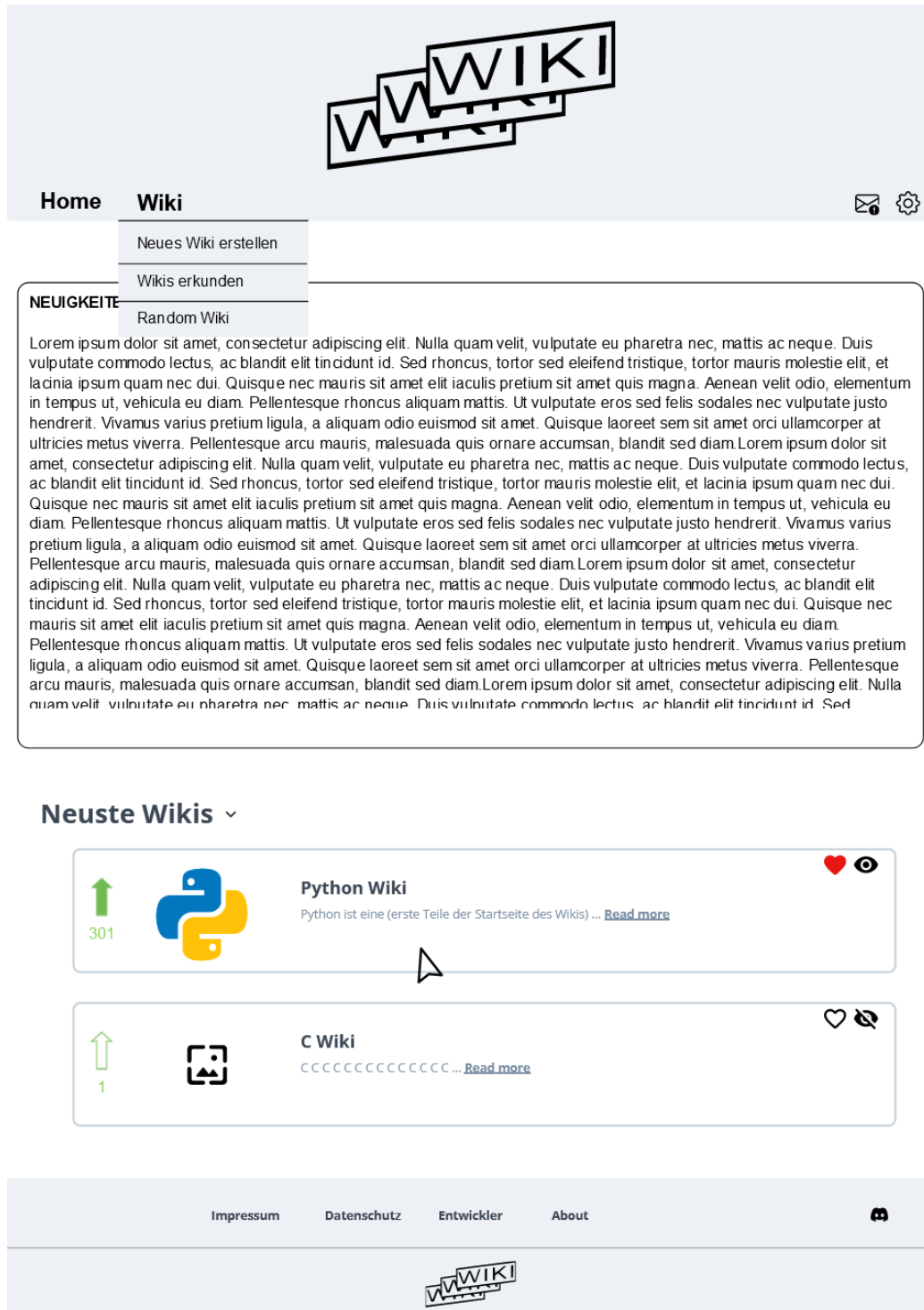


4.2.2 Strukturdiagramm Login

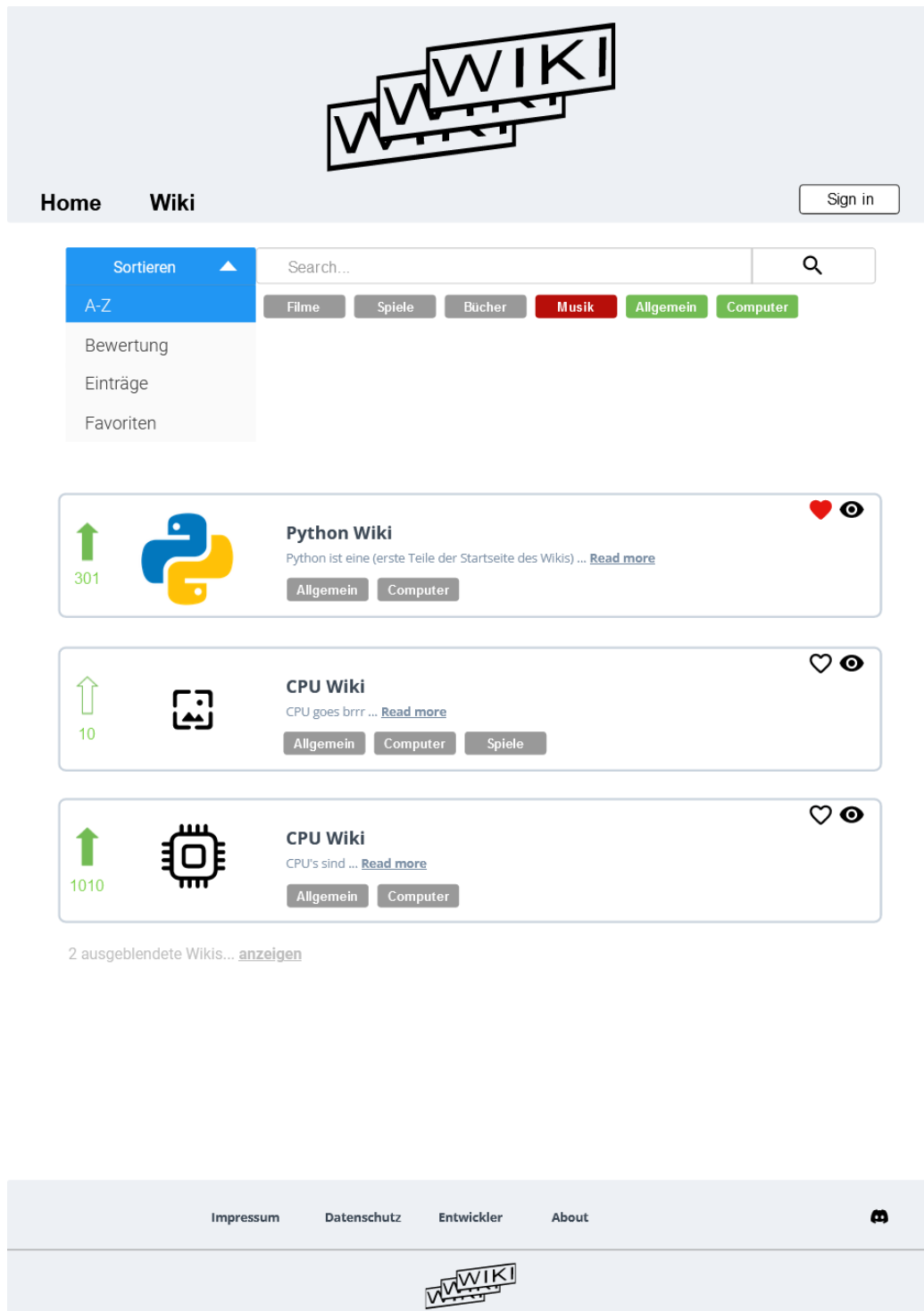


4.3 Moqups

4.3.1 UI Moqup - Main Page



4.3.2 UI Moqup - Wiki



5 Projekt World Wild Wiki

5.1 Entwicklungsumgebung

5.2 Symfony PHP Framework

Bei Symfony handelt es sich um eine Plattform zur Erstellung von Webanwendungen. Hierbei stellt Symfony Code-Bibliotheken für häufig verwendete Funktionen zur Verfügung, wodurch die Menge an zu schreibendem Originalcode deutlich reduziert wird. Hierdurch wird die benötigte Entwicklungszeit für ein Projekt geringer. Der Code wird außerdem in eine Reihe von Verzeichnissen, je nach Funktion aufgeteilt.

5.2.1 Templatestruktur

Das Projekt beinhaltet diverse Templates, welche das Projekt vervollständigen. Hierbei wurde Twig verwendet. Dies ist eine Template-Engine für die Programmiersprache PHP. Jede Komponente hat hier ein eigenes Teil-Template, welches sich an dem Basis Template orientiert. Für folgende Elemente besitzt unser Projekt ein eigenes Template:

1. Basis-Template (base)
2. Login (index)
3. Registrierung (register, confirmation_email)
4. Wiki (browse, createWiki, home, wikiPage, createEintrag, editEintrag, editWiki, eintragPage)
5. Custom Error (errors, noPermissions)
6. Error (error404)

Die Endung der Dateien ist .html.twig (Beispiel base.html.twig) An der Endung .twig erkennt man noch einmal die Verwendung der Twig-Template-Engine.

5.2.2 Benutzergruppen und Funktionalitäten

Funktion / User						
Funktion (Sichtbarkeit) / User	Plattform Admin	Wiki Owner	Wiki Admin	Collaborator	Normaler User	Unregistriert
User Einstellungen	✓	✓	✓	✓	✓	✗
Plattform Einstellungen	✓	✗	✗	✗	✗	✗
Neues Wiki	✓	✓	✓	✓	✓	✗
Wiki Einstellungen	✓	✓	✓	✗	✗	✗
Wikis Favorisieren / Verstecken / Bewerten	✓	✓	✓	✓	~	✗
Beiträge einsehen	✓	✓	✓	✓	~	~
Beiträge erstellen	✓	✓	✓	✓	~	✗
Beiträge bearbeiten	✓	✓	✓	✓	~	✗
Beiträge löschen	✓	✓	✓	✶	✶	✗

In der Webanwendung World Wild Wiki gibt es verschiedene Funktionalitäten, die ein User besitzen kann. Unregistrierte User besitzen keine Nutzerrechte und können lediglich die Wikis der anderen Nutzer einsehen. Erst durch eine Registrierung ist es möglich in der Rechtehierarchy aufzusteigen. Auch für das Umschalten in Dark und Lightmode besitzt die Anwendung verschiedene Funktionen. Durch das Setzen eines Cookies wird zum Beispiel festgestellt, welcher Mode gerade aktiv ist. Es gibt verschiedene Formulare, welche genutzt werden können, um Markdown Dateien in Wiki Beiträge zu parsen, welche dann später für den Benutzer zu sehen sind. Dies wird durch einen integrierten Markdown Parser ermöglicht. Verwendet wird hierzu Marked (ersichtlich in Github unter [markedjs/marked](https://github.com/markedjs/marked)).

5.3 Demo-Wiki

Innerhalb der Webanwendung befindet sich ein bereits vorgefertigtes Demo-Wiki über Drag Racing bzw. Dragster Beschleunigungsrennen. Hier werden die Vorzüge von Markdown ersichtlich. Es ist möglich Bilder einzufügen, verschiedene Teile hervorzuheben oder die Schriftfarbe anzupassen, was dem

Nutzer eine Reihe an Möglichkeiten zur Gestaltung des Wikis gibt. Der Markdown-Code des Beispiel-Wikis ist frei einsehbar für die Nutzer im Quellcode.

5.3.1 Wiki-Arten

Hier wird in private Wikis, öffentliche Wikis und View-Only Wikis unterschieden. Hier gelten folgende Dinge:

Private Wikis:

1. Das Wiki wird nicht im allgemeinen Wiki Browser angezeigt.
2. Normale User können keine Beitrittsanfragen an das Wiki senden, um Collaborator zu werden.
3. Der Wiki Owner und die Wiki Admins können nur selbst Anfragen an Nutzer schicken und diese in das Wiki einladen.

View-Only Wikis:

1. Jeder Besucher des Wikis (oder nur jeder Nutzer der eingeloggt ist) kann Beiträge einsehen und das Wiki bewerten.
2. Nur User \geq Collaborator können Beiträge erstellen und bearbeiten.
3. Jeder kann Anfragen an das Wiki senden, um diesem beizutreten oder es zu bearbeiten.

Öffentliche Wikis:

1. Jeder eingeloggte User kann Beiträge erstellen, bearbeiten und einsehen.
2. Wie Wikipedia nur dass man eingeloggt sein muss.

6 Projekt und Quellcode

Innerhalb des Quellcodes findet man viele Kommentare, anhand derer man sich in das Projekt einarbeiten und gegeben falls für seine eigenen Zwecke verändern oder anpassen kann. Alle Funktionen und Templates sind durchkommentiert.

7 Projekt Installation

7.1 Webserver WSL

7.1.1 Apache2

1. `sudo apt-get update`
2. `sudo apt-get upgrade`
3. `sudo apt-get install apache2`
4. `sudo apache2ctl configtest`
5. `sudo a2enmod rewrite`
6. `sudo a2enmod headers`
7. `sudo service apache2 start` / `sudo service apache2 restart`

7.1.2 MYSQL

1. `sudo apt-get install mysql-server`
2. `sudo service mysql start`
3. `sudo mysql_secure_installation`
4. `sudo mysql`
5. `SELECT user, authentication_string, plugin, host FROM mysql.user;`
6. `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'YourMYSQLPassword';`
7. `FLUSH PRIVILEGES && exit`
8. `mysql -u root -p + Dein Password`
9. `CREATE USER 'WikiDB'@'localhost' IDENTIFIED BY 'B4db)%aHm84';`
10. `GRANT ALL PRIVILEGES ON *.* TO 'WikiDB'@'localhost' WITH GRANT OPTION;`

7.1.3 PHP

1. `sudo apt-get install php7.4 php7.4-cli php7.4-common php7.4-curl php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml libapache2-mod-php`
2. `php -v` (Ist das Ergebnis 7.4.3 ?)
3. `sudo service apache2 restart`
4. `sudo chown YOUR_WSL_USER /var/www/html`
5. `sudo chmod 755 -R /var/www/html`

7.1.4 phpMyAdmin

1. `sudo apt-get install phpmyadmin (apache2, yes, Neues_Password, abort)`
2. `mysql -u root -p + Dein Password`
3. `UNINSTALL COMPONENT "file://component_validate_password";`
4. `exit && sudo apt-get install phpmyadmin`
5. `mysql -u root -p + Dein Password`
6. `INSTALL COMPONENT "file://component_validate_password"; && exit`
7. `sudo phpenmod mbstring` **Merk dir** `/etc/apache2/conf-enabled/`
8. `sudo service apache2 restart`
9. `sudo -H nano /etc/apache2/apache2.conf &&` **In der geöffneten Datei jetzt** `Include /etc/phpmyadmin/apache.conf` **einfügen und die Datei speichern!**
10. `sudo service apache2 restart`

Userdaten PHP-Admin:

Username: **WikiDB**

Password: **B4db)%aHm84**

7.1.5 Composer, Zip & Unzip

1. `sudo php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"`
2. `sudo php composer-setup.php`
3. `sudo php -r unlink('composer-setup.php');"`
4. `sudo mv composer.phar /usr/local/bin/composer`
5. `sudo apt-get install -y zip`

7.2 Symfony Projekt

1. `cd /var/www/html`
2. `composer create-project symfony/website-skeleton WebAppProject, y`
3. `cd /var/www/html/WebAppProject`
4. `php bin/console about`
5. `sudo nano /etc/apache2/sites-available/000-default.conf`

Folgenden Code in die Datei einfügen:

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/WebAppProject/public

DirectoryIndex index.php
<Directory "/var/www/html/WebAppProject/public">
    AllowOverride All
    Allow from All

    <IfModule mod_rewrite.c>
        Options -MultiViews
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </IfModule>
</Directory>
```

1. `sudo a2dissite 000-default`
2. `sudo a2ensite 000-default`
3. `sudo service apache2 restart`

7.2.1 Demo-Daten

Zur Verwendung der API (hcaptcha) muss die ENV-Datei angepasst werden. Wenn die Symfony Seite auf localhost läuft wirft die hcaptcha API einen Error. Es funktioniert wenn man über ein anderes Gerät auf den anderen Server über die IP-Adresse zugreift. Dann kann man sich ohne einen Error von hcaptcha registrieren.

Anpassung ENV-Datei meteo-concept:

1. Füge bei `HCAPTCHA_SITE_KEY` ein: `ac885c54-5dce-44b4-9158-5e47a93c3cea`
2. Füge bei `HCAPTCHA_SECRET` ein: `0xBaa8Af099Bf8938725700742b2CBAf71ef604203`

8 Projekt-Fazit

Die anfänglichen Schwierigkeiten konnten gut gelöst und umgesetzt werden. Es wurde schnell aber ersichtlich, dass der geplante Projektinhalt den Rahmen gesprengt hat und wir uns zu viel für den gegebenen Zeitplan vorgenommen hatten. Dennoch wurden die meisten Vorhaben tadellos umgesetzt. Schwierigkeiten gab es besonders bei der initialen Einrichtung von Symfony. Hier gab es betriebssystemabhängige Probleme, die eine problemlose Ausführung verhinderten. Das Projekt ist frei auf Github verfügbar und kann genutzt werden. Hier ist auch der Code für die DEMO-WIKI Markdown Dateien ersichtlich.