|  |
| --- |
| Bildungs- und Kulturdepartement ‍**Fach- und Wirtschaftsmittelschulzentrum Luzern** |

**PA-Bericht**

Webapplikation für Fussballvereine

David Cvetkovic

28.02.2024

**PA-Steckbrief**

|  |  |
| --- | --- |
| **Merkmale** | **Beschreibung** |
| Projektname | Webapplikation für Fussballvereine |
| Projektstart | 19.02.2024 |
| Projektende | 01.03.2024 |
| Projektdauer | 10 Tage, pro Tag 8 Stunden |
| Betrieb | Fach- und Wirtschaftsmittelschulzentrum Luzern |
| Durchführung | getAbstract |
| Kandidat | David Cvetkovic |
| Verantwortliche Fachkraft | Adrian Imboden |
| Hauptexperte | Adrian Elsener |
| Zweitexperte | André Meyer |

Tabelle 1: PA-Steckbrief

**Inhaltsverzeichnis**

[1 Teil 1 6](#_Toc160013818)

[1.1 Aufgabenstellung 6](#_Toc160013819)

[1.1.1 Detaillierte Aufgabenstellung 6](#_Toc160013820)

[1.1.2 Benutzte Software 7](#_Toc160013821)

[1.1.3 Benutzte Hardware 7](#_Toc160013822)

[1.1.4 Methode 7](#_Toc160013823)

[1.1.5 Vorkenntnisse 7](#_Toc160013824)

[1.1.6 Vorarbeiten 7](#_Toc160013825)

[1.2 Projektorganisation 8](#_Toc160013826)

[1.2.1 Projektmanagementmethode 8](#_Toc160013827)

[1.2.2 Firmenstandards 8](#_Toc160013828)

[1.2.3 Organisation der Arbeitsergebnisse 8](#_Toc160013829)

[1.2.3.1 Versionierung 8](#_Toc160013830)

[1.2.3.2 Dokumentenablage 8](#_Toc160013831)

[1.3 Zeitplan 9](#_Toc160013832)

[1.3.1 Meilensteine 10](#_Toc160013833)

[1.4 Arbeitsprotokoll 11](#_Toc160013834)

[1.4.1 Tag 01: 19.02.2024 11](#_Toc160013835)

[1.4.2 Tag 02: 20.02.2024 11](#_Toc160013836)

[1.4.3 Tag 03: 21.02.2024 12](#_Toc160013837)

[1.4.4 Tag 04: 22.02.2024 12](#_Toc160013838)

[1.4.5 Tag 05: 23.02.2024 13](#_Toc160013839)

[1.4.6 Tag 06: 26.02.2024 13](#_Toc160013840)

[1.4.7 Tag 07: 27.02.2024 14](#_Toc160013841)

[1.4.8 Tag 08: 28.02.2024 14](#_Toc160013842)

[1.4.9 Tag 09: 29.02.2024 14](#_Toc160013843)

[1.4.10 Tag 10: 01.03.2024 14](#_Toc160013844)

[2 Teil 2 15](#_Toc160013845)

[2.1 Kurzfassung 15](#_Toc160013846)

[2.1.1 Kurze Ausgangssituation 15](#_Toc160013847)

[2.1.2 Umsetzung 15](#_Toc160013848)

[2.1.3 Ergebnis 15](#_Toc160013849)

[2.2 Wissensbeschaffung 16](#_Toc160013850)

[2.2.1 CI/CD 16](#_Toc160013851)

[2.2.1.1 GitHub Actions 16](#_Toc160013852)

[2.2.1.2 Linting 16](#_Toc160013853)

[2.3 Vorbereitung zur Umsetzung 17](#_Toc160013854)

[2.3.1 Kontextdiagramm 17](#_Toc160013855)

[2.3.1.1 Administrator 17](#_Toc160013856)

[2.3.1.2 Trainer 17](#_Toc160013857)

[2.3.1.3 Fussballspieler 17](#_Toc160013858)

[2.3.1.4 Helfer 17](#_Toc160013859)

[2.3.2 Mockup 18](#_Toc160013860)

[2.3.2.1 Login-Ansicht 18](#_Toc160013861)

[2.3.2.2 Registrierungsansicht 19](#_Toc160013862)

[2.3.2.3 Kalenderansicht 19](#_Toc160013863)

[2.3.2.4 Menüansicht 20](#_Toc160013864)

[2.3.2.5 Dashboard-Ansicht 20](#_Toc160013865)

[2.3.2.6 Analysenansicht 22](#_Toc160013866)

[2.3.3 User Stories 25](#_Toc160013867)

[2.3.4 Testkonzept 26](#_Toc160013868)

[2.3.4.1 Quadrant 1 (Q1) 26](#_Toc160013869)

[2.3.4.2 Quadrant 2 (Q2) 26](#_Toc160013870)

[2.3.4.3 Quadrant 3 (Q3) 26](#_Toc160013871)

[2.3.4.4 Quadrant 4 (Q4) 26](#_Toc160013872)

[2.3.4.5 Testfälle definieren 27](#_Toc160013873)

[2.3.4.5.1 Manuelle Tests 27](#_Toc160013874)

[2.3.4.5.2 Automatisierte Tests 30](#_Toc160013875)

[2.4 Datenbank 31](#_Toc160013876)

[2.4.1 ERM 31](#_Toc160013877)

[2.4.2 Backup 31](#_Toc160013878)

[2.5 Login 32](#_Toc160013879)

[2.6 Registrierung 35](#_Toc160013880)

[2.7 Kalender 40](#_Toc160013881)

[2.7.1 ICS-Datei 43](#_Toc160013882)

[2.8 Menü 44](#_Toc160013883)

[2.9 Dashboard 45](#_Toc160013884)

[2.10 Analysen 48](#_Toc160013885)

[2.11 Deployment 49](#_Toc160013886)

[2.12 Testdurchführung 50](#_Toc160013887)

[2.12.1 Testauswertung 50](#_Toc160013888)

[2.13 Ausführungsanleitung 51](#_Toc160013889)

[2.13.1 Webseite öffnen (XAMPP) 51](#_Toc160013890)

[2.13.2 Unit Tests ausführen 51](#_Toc160013891)

[2.13.3 Linters verwenden 51](#_Toc160013892)

[2.14 Schlusswort 52](#_Toc160013893)

[2.15 Abkürzungsverzeichnis 53](#_Toc160013894)

[2.16 Glossar 54](#_Toc160013895)

[2.17 Quellenverzeichnis 56](#_Toc160013896)

[2.18 Abbildungsverzeichnis 57](#_Toc160013897)

[2.19 Tabellenverzeichnis 58](#_Toc160013898)

[2.20 Anhang 59](#_Toc160013899)

# Teil 1

In diesem Kapitel geht es um die Umgebung der Facharbeit.

## Aufgabenstellung

Folglich wird aufgezeigt, was zur Aufgabenstellung gehört und verwendet wurde, um die Aufgabe zu lösen.

### Detaillierte Aufgabenstellung

Es soll eine Webapplikation für Fussballvereine erstellt werden. Die Webapplikation soll für die Spiel- und Trainingsplanung dienen. Dies kann durch eine Implementierung eines Kalenders gewährleistet werden. Zudem sollten die Fussballvereine in der Lage sein, ihre Analysen (Stärken und Schwächen des Gegners, an was muss die Mannschaft arbeiten, wo müssen sie sich verbessern usw.) einzutragen.

Nutzer der Anwendung sind folgende Personengruppen:

- Administrator (diejenigen Personen, welche die Applikation verwalten)

- Trainer

- Fussballspieler

- Helfer (z.B. Fahrer, welche die Mannschaft zum Spiel bringt)

Das Testing wird mit automatisierten sowie manuellen Tests durchgeführt.

Folgende Features sind erwartet:

Pflicht:

- Der Trainer kann in einem Kalender die Trainings- und Spieledaten eintragen (Datum, wer ist beteiligt, Standort)

- Alle Nutzer haben ein persönliches Login und können sich einloggen

- Es existiert eine funktionierende Anleitung, wie die Software auf einem LAMP-Stack kompatiblen Hoster (sind praktisch alle) deployed werden kann.

- Die Software ist auf GitHub mit einer sinnvollen Open-Source Lizenz öffentlich verfügbar

Priorität Hoch:

- Die Logins können über die Weboberfläche gemanaged werden (User erstellen, Passwort zurücksetzen, ...)

- Der persönliche Kalender kann als ICS heruntergeladen/abonniert werden (um es z.B. im Smartphone-Kalender zu importieren)

Priorität Mittel:

- Der Administrator/Trainer kann ein komplettes Backup als Datei herunterladen und auch wieder einspielen

- Die Spielresultate können bei den Spielen eingetragen werden (wer hat wann das Tor gemacht), Rote/Gelbe Karten, Notizen, ...

- Die Spielresultate werden auf einem Dashboard übersichtlich dargestellt

- Die Analysen der Gegner und an was die Mannschaft arbeiten kann, können eingetragen und angeschaut werden.

### Benutzte Software

* Microsoft-Office-Apps
* Visual Studio Code
* XAMPP
* Google Chrome
* Git
* Adobe XD

### Benutzte Hardware

* Persönlicher Laptop und Dockingstation von getAbstract

### Methode

Für die PA wird die Projektmethode «IPERKA» verwendet.

### Vorkenntnisse

* HTML, CSS, JS, PHP und SQL: Wissen durch einige Module in der Schule
* Visual Studio Code und XAMPP: In der Schule verwendet sowie auch privat

### Vorarbeiten

* Grundlegende Überlegungen, wie die Webapplikation aussehen könnte, aber noch kein Code vorhanden

## Projektorganisation

Nun wird aufgezeigt, was alles zur Projektorganisation gehört.

### Projektmanagementmethode

Für die Projektmanagementmethode wird «IPERKA» verwendet. Ich habe mich für diese Methode entschieden, weil es eine gute Struktur und Übersicht hat. Zudem kann man die einzelnen Tätigkeiten den IPERKA-Phasen zuweisen, wie auch im Zeitplan ersichtlich.

Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 1: IPERKA

### Firmenstandards

Für die Dokumentation wurde die Vorlage von dem Fach- und Wirtschaftsmittelschulzentrum Luzern verwendet.

### Organisation der Arbeitsergebnisse

Damit die Organisation der Arbeitsergebnisse verständlich ist, wird in diesem Kapitel erklärt, wie die Versionierung und Dokumentenablage sind.

#### Versionierung

Das Dokument wird täglich unter einer neuen Version lokal und auf GitHub gespeichert. Somit habe ich auch Zugriff auf jede vergangene Version. Der Code wird ebenfalls täglich lokal und auf GitHub gespeichert. So kann ich dann auch auf jede vergangene Version zugreifen.

#### Dokumentenablage

Die Dokumente und der Code, welche für die PA relevant sind, werden lokal (C:\xampp\htdocs\Fussball-Webapplikation) auf dem Laptop gespeichert. Zudem werden die Dokumente und auch der Code auf GitHub (<https://github.com/DC2305/Fussball-Webapplikation>) gespeichert.

## Zeitplan

Pro Tag gibt es vier Kästchen. Ein Kästchen steht für 2 Stunden.

Für die Sollzeit wurde grün verwendet und für die Istzeit rot. Zudem wurden die Meilensteine mit blau markiert.

Ein Bild, das Text, Reihe, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 2: Zeitplan

### Meilensteine

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Meilenstein** | **IPERKA-Phase** | **Beschreibung** |
| 22.02.2024 | Meilenstein 1 (M1) | Realisieren | Login und Registrierung implementieren |
| 23.02.2024 | Meilenstein 2 (M2) | Realisieren | Kalender implementieren |
| 28.02.2024 | Meilenstein 3 (M3) | Realisieren | Deployment-Anleitung schreiben |
| 01.03.2024 | Meilenstein 4 (M4) | Realisieren | Code deployen |
| 01.03.2024 | Meilenstein 5 (M5) | Kontrollieren | Kontrollieren und Abschliessen aller PA-Dokumente |

Tabelle 2: Meilensteine

## Arbeitsprotokoll

In diesem Kapitel geht es um das Arbeitsprotokoll.

### Tag 01: 19.02.2024

Ablauf:

Zuerst habe ich ein GitHub-Repository erstellt, damit ich auch dort meine Dokumente ablegen kann. Danach habe ich ein Excel erstellt, bei der ich meine Arbeitszeiten während der PA notieren kann. Nachdem ich dieses Excel erstellt habe, habe ich den Soll-Ist-Zeitplan erstellt. Dann habe ich mit der Dokumentationsstruktur angefangen und habe so auch gleich Kapiteln festgelegt und konnte bereits viele Kapiteln mit Inhalt befüllen. Auch das Testkonzept habe ich dann geschrieben und habe mich für das Quadrantenmodell entschieden, weil es das Testing sehr gut beschreibt. Während ich das gemacht habe, habe ich auch den Kriterienkatalog sowie auch den Leitfaden studiert. Zudem habe ich auch noch die Aufgabenstellung studiert.

Reflexion:

Ich bin sehr gut in die PA gestartet. Ich konnte die Tätigkeiten, welche für heute geplant waren, erledigen. Das Testkonzept habe ich schneller erledigt als geplant. Zudem habe ich mir einen guten Standpunkt für den nächsten Tag erarbeitet.

### Tag 02: 20.02.2024

Ablauf:

Für heute war geplant, dass ich das Kontextdiagramm erstelle sowie auch das Mockup. Bevor ich mit dem angefangen habe, habe ich noch Anpassungen in der Dokumentation vorgenommen. Danach kam der Hauptexperte, weil heute der erste Besuchstag war. Das Gespräch war sehr interessant. Nach dem Expertenbesuch habe ich weiterhin an der Dokumentation gearbeitet und die Dokumentationsstruktur angepasst. Dann habe ich das Kontextdiagramm erstellt und auch das Mockup. Das Mockup habe ich mit Adobe XD erstellt, weil ich schon mit dem gearbeitet habe und es ein sehr gutes Programm ist, um Mockups zu erstellen. Als ich mit dem Mockup fertig war, habe ich an der Dokumentation weitergearbeitet.

Reflexion:

Heute war es ein guter Tag. Ich konnte die geplanten Tätigkeiten auf heute erledigen und habe das Mockup schneller erledigt als geplant, was mir mehr Zeit für die Dokumentation gegeben hat. Für morgen ist geplant, dass ich die User Stories erstelle sowie die Testfälle definiere und mit der ersten Implementierung anfange, nämlich mit dem Login und der Registrierung.

### Tag 03: 21.02.2024

Ablauf:

Ich habe zuerst die User Stories erstellt. Da hatte ich mir überlegt, ob ich auch testen möchte, ob man die Webapplikation auch auf dem Smartphone benutzen kann. Dann habe ich mich dafür entschieden, diesen Test ebenfalls in den User Stories einzutragen. Nachdem ich das gemacht habe, habe ich die Testfälle definiert. Danach habe ich die Testing Frameworks Jest und PHPUnit installiert, weil ich JavaScript und PHPUnit verwende und ein wenig Erfahrung mit Jest und PHPUnit habe. Nachdem ich das installiert habe, habe ich auch den Linter ESLint für JS installiert sowie PHP\_CodeSniffer für PHP und Stylelint für CSS, damit ich auch lokal den Code linten kann. Auch wollte ich aber, dass der Code auf GitHub gelintet und getestet wird (GitHub Actions), weswegen ich dann auf GitHub ein Workflow-Ordner erstellt habe und dort in YAML-Dateien definiert habe, was gelintet und getestet werden sollte. Für das Linting auf GitHub habe ich mich für den Superlinter entschieden, weil dieser fast alle Programmiersprachen lintet, auch die, die ich verwende. Für das Testing habe ich dann auf GitHub wie auch lokal Jest und PHPUnit verwendet. Dann habe ich mit der Login-Implementierung und Registrierungsimplementierung angefangen. Dafür habe ich dann eine Datenbank und Tabelle erstellt. Das Ganze habe ich dann noch dokumentiert.

Reflexion:

Heute habe ich gut gearbeitet. Ich konnte alle geplanten Tätigkeiten erledigen. Allerdings habe ich mehr Zeit für die Login-Implementierung sowie die Registrierungsimplementierung gebraucht als erwartet.

### Tag 04: 22.02.2024

Ablauf:

Zuerst habe ich an der Dokumentation gearbeitet und ein paar Anpassungen vorgenommen. Dann habe ich bei den GitHub Actions die YAML-Dateien angepasst und die Linting-Errors behoben. Zudem erschien bei den GitHub Actions ein Fehler, wenn man die PHPUnit-Tests ausführen möchte. Ich habe dann versucht, diesen Fehler zu beheben, leider ohne Erfolg. Ich wollte nicht zu viel Zeit mit dem verschwenden, weswegen ich dann den Code von gestern noch auf der Webseite «Hostpoint» deployed habe, weil ich es gestern vergessen hatte. Ich habe mich für Hostpoint entschieden, weil es eine Webseite ist, mit der ich mich ein wenig auskenne und es einen ähnlichen Aufbau wie XAMPP hat. Nachdem ich den Code deployed habe, habe ich mit der Implementierung des Kalenders und Menüs angefangen. Mit dem Menü wurde ich fertig, allerdings werde ich noch am Kalender ein paar Anpassungen vornehmen müssen. Ich habe mich beim Kalender für die JS-Library «FullCalendar» entschieden, weil es viele verschiedene Funktionen hat, betreffend Kalender, und man kann Anpassungen vornehmen und den Kalender so gestalten wie man es möchte.

Reflexion:

Heute war es eher ein stressiger Tag, da der Kalender nicht so funktioniert hat, wie ich es mir erhofft hatte, weswegen ich unbedingt morgen Anpassungen vornehmen muss. Zudem muss ich auch noch herausfinden, wie ich den Fehler mit dem PHPUnit bei den GitHub Actions beheben kann. Auf der anderen Seite bin ich froh, dass ich mit dem Menü fertig geworden bin und die Linting-Errors behoben habe. Auch war alles in Ordnung, bezüglich dem Code-Deployment auf Hostpoint.

### Tag 05: 23.02.2024

Ablauf:

Ich habe am Anfang Anpassungen an der Kalenderimplementierung vorgenommen und den Code dann auf Hostpoint deployed. Allerdings musste ich auch dort Anpassungen vornehmen, weil die Ansicht anders aussah als lokal bei XAMPP. Nachdem ich die nötigen Anpassungen gemacht hatte, sah die Ansicht dann auch so aus wie bei mir lokal. Ich habe dann an der Dokumentation gearbeitet und habe nachher nochmals versucht, den Fehler bei den GitHub Actions zu beheben, bezüglich PHPUnit. Leider hatte ich auch da kein Erfolg und muss unbedingt schauen, dass ich das nächste Woche behebe.

Reflexion:

Der Tag war in Ordnung. Ich konnte den Kalender fertig implementieren sowie auch den Code auf Hostpoint deployen. Leider habe ich den Fehler bei den GitHub Actions, bezüglich PHPUnit, immer noch nicht behoben. Zudem konnte ich auch leider nicht mit der Implementierung des ICS-Downloads anfangen, weswegen ich auch hier schauen muss, dass ich das nachhole.

### Tag 06: 26.02.2024

Ablauf:

Zuerst habe ich den Fehler bei den GitHub Actions behoben, bezüglich PHPUnit. Und zwar musste ich die Version 10 von PHPUnit verwenden, weil es Probleme mit der Version 11 gab. Als ich das gemacht habe, habe ich an der Dokumentation gearbeitet. Dann habe ich auf GitHub eine Open-Source Lizenz hochgeladen, weil mein Projekt schlussendlich öffentlich auf GitHub sein wird. Nachdem ich die Lizenz hochgeladen habe, habe ich an der ICS-Implementierung gearbeitet, aber ich bin leider nicht weitergekommen. Weil ich bei der ICS-Implementierung nicht weiterkam, habe ich mir Überlegungen gemacht, wie ich das Dashboard implementieren könnte und habe dort mit den ersten Implementierungen angefangen.

Reflexion:

Ich habe heute gut gearbeitet, weil ich den Fehler bei den GitHub Actions behoben habe und zudem die Open-Source Lizenz auf GitHub hochladen konnte. Auch habe ich früher als geplant mit der Implementierung des Dashboards angefangen, weil ich bei der ICS-Implementierung nicht weitergekommen bin und somit nicht Zeit verlieren wollte. Ansonsten muss ich noch schauen, dass ich die Aufgabe mit dem ICS noch erledige.

### Tag 07: 27.02.2024

Ablauf:

Am Anfang habe ich noch das Dashboard fertig implementiert und den Code dann auf Hostpoint deployed. Allerdings gab es ein Fehler, und zwar war der Name der Datenbank falsch. Den Fehler konnte ich mithilfe der Error-Logs von Hostpoint finden und beheben. Dann habe ich noch an der Dokumentation gearbeitet. Ich habe dann an der ICS-Implementierung gearbeitet und bin dort so weit gekommen, dass die Kalendereinträge als ICS-Datei heruntergeladen werden, aber nur der Titel und ein Teil der Beschreibung sind dann ersichtlich und nicht alle Informationen des Kalenders. Da muss ich noch schauen, dass ich es hinbekomme, diesen Fehler zu beheben. Dann habe ich wieder an der Dokumentation gearbeitet und weitere Kapitel mit Inhalt befüllt. Als ich dann die Anpassungen in der Dokumentation gemacht hatte, habe ich eine README-Datei in meinem GitHub-Repository eingefügt. In diesem README ist eine Installations- und Ausführungsanleitung von XAMPP, den verwendeten Testing Frameworks und Linters vorhanden. So ist dann sichergestellt, dass die Person, welche mein Repository besucht, weiss, wie man was installiert und ausführt. Nachdem ich das README erstellt hatte, habe ich an den Unit Tests gearbeitet. Schliesslich habe ich mit der Implementierung der Analysenseite begonnen.

Reflexion:

Der heutige Tag war in Ordnung. Ich konnte wieder den Code erfolgreich auf Hostpoint deployen, nachdem ich den Fehler behoben hatte. Laut Zeitplan bin ich aber im Rückstand, bezüglich der ICS-Implementierung. Da muss ich schauen, dass ich noch mit dem fertig werde. Ich werde weiteres im Code austesten müssen, bezüglich dem ICS. Ansonsten bin ich froh, dass ich noch ein wenig Zeit hatte, mit der Analysenseite anzufangen. So habe ich mir einen guten Standpunkt für morgen erarbeitet.

### Tag 08: 28.02.2024

### Tag 09: 29.02.2024

### Tag 10: 01.03.2024

# Teil 2

In diesem Kapitel geht es um die Projektdokumentation.

## Kurzfassung

Nun wird die ganze Arbeit zusammengefasst mithilfe der Punkte «Kurze Ausgangssituation», «Umsetzung» und «Ergebnis».

### Kurze Ausgangssituation

Fussballvereine haben ihre Spiel- und Trainingsplanung entweder physisch auf Papier, in einer Excel-Tabelle oder auf einem Computer im Kalender. Das ist nicht übersichtlich und die Kommunikation mit den betroffenen Personen ist eher schwierig.

Es soll nun eine Webapplikation entwickelt werden, welche die Planung und Kommunikation mithilfe eines Kalenders, Dashboards und einer Analysenseite vereinfacht. Dazu soll es auch ein Login und eine Registrierung geben, weil verschiedene Personen aus dem Fussballverein die Webapplikation benutzen werden.

Die Webapplikation soll auf GitHub als Open-Source veröffentlicht werden und das Deployment muss auch funktionieren.

### Umsetzung

Für die Umsetzung wurden XAMPP, HTML, CSS, JS, PHP und MySQL verwendet. Mit MySQL musste eine Datenbank mit den Tabellen für das Login, die Registrierung, den Kalender, das Dashboard und die Analysenseite erstellt werden, damit die Daten gespeichert werden können. Für das Deployment wurde der Webhosting-Anbieter «Hostpoint» benutzt.

### Ergebnis

Wenn man die Webapplikation öffnet, muss man sich zuerst einloggen oder registrieren. Die registrierten Benutzer sind in der Datenbank ersichtlich. Nachdem man sich eingeloggt hat, sieht man den Kalender. Im Kalender kann man Einträge machen, die dann in der Datenbank gespeichert werden. Mithilfe eines Menüs kann man zur Dashboard-Seite weitergeleitet werden. Auf der Dashboard-Seite kann man die Spielresultate mit weiteren Notizen eintragen, die dann ebenfalls in der Datenbank gespeichert werden. Wenn man den Eintrag bearbeitet, wird das in der Datenbank aktualisiert und wenn man den Eintrag löscht, dann wird dieser Eintrag aus der Datenbank gelöscht. Auf der Analysenseite können die Stärken und Schwächen der eigenen Mannschaft sowie der gegnerischen Mannschaften eingetragen werden. Diese können auch aktualisiert und gelöscht werden. Schlussendlich kann man sich auch noch von der Webapplikation ausloggen.

Die Webapplikation ist auf GitHub als Open-Source veröffentlicht worden und zudem wurde der Code auf Hostpoint deployed.

## Wissensbeschaffung

Unter dem Kapitel «Quellenverzeichnis» sind alle Quellen aufgelistet, welche für die Wissensbeschaffung benutzt wurden.

### CI/CD

CI ist eine Praxis, bei der Entwickler kleine Änderungen und Überprüfungen an ihrem Code vornehmen. CD ist die automatisierte Bereitstellung von fertigem Code an Umgebungen wie Test und Entwicklung.



Abbildung 3: CI/CD

#### GitHub Actions

Für CI/CD kann GitHub Actions verwendet werden. Wenn man etwas auf GitHub hochlädt, dann kann man mit den GitHub Actions sicherstellen, dass der Code korrekt durchläuft und bei Fehlern dem Benutzer das meldet.

#### Linting

Linting ist die automatisierte Überprüfung des Quellcodes auf programmatische und stilistische Fehler. Dies geschieht mit einem Lint-Tool (auch Linter genannt).

## Vorbereitung zur Umsetzung

Nun wird aufgezeigt, was für die Vorbereitung zur Umsetzung verwendet wurde.

### Kontextdiagramm

Das Kontextdiagramm stellt das umzusetzende System dar.

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 4: Kontextdiagramm

#### Administrator

Der Administrator verwaltet die ganze Webapplikation.

#### Trainer

Der Trainer soll die Webseite benutzen können.

#### Fussballspieler

Dem Fussballspieler sollte es auch möglich sein, die Webseite benutzen zu können.

#### Helfer

Der Helfer ist die Person, welche die Mannschaft zu den Spielen fährt. Weil die Webapplikation für kleinere Vereine gedacht ist, ist der Helfer der Trainer meistens selbst.

### Mockup

Das Mockup zeigt, wie die Webapplikation etwa am Schluss aussehen sollte.

#### Login-Ansicht

Wenn man die Webapplikation öffnet, muss man sich zuerst einloggen. Auch kann man sich registrieren, indem man unten «Hier registrieren» klickt.

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 5: Mockup Login

#### Registrierungsansicht

Wenn es den Benutzern nicht gibt, sollte sich diese Person registrieren können.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 6: Mockup Registrierung

#### Kalenderansicht

Nachdem sich der Benutzer angemeldet hat, sollte diese Person den Kalender sehen.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 7: Mockup Kalender

#### Menüansicht

Das Menü soll oben links sein und der Benutzer soll es öffnen können.

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 8: Mockup Menü

#### Dashboard-Ansicht

Beim Dashboard soll der Benutzer die Spielresultate, mithilfe des Buttons «Eintrag hinzufügen», eintragen können sowie auch weitere Notizen (Wer hat wann das Tor geschossen, Rote/Gelbe Karten usw.).

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 9: Mockup Dashboard

So müsste es dann etwa aussehen, wenn man einen neuen Eintrag hinzufügen möchte:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 10: Mockup Dashboard - Neuer Eintrag

Nachdem man den neuen Eintrag gespeichert hat, sollte es dann so aussehen:

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 11: Mockup Dashboard - Eintrag hinzugefügt

Zusätzlich sollte es möglich sein, dass man die Notizen vom erstellten Eintrag öffnen kann.

Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 12: Mockup Dashboard - Notizen

#### Analysenansicht

Bei den Analysen sollten die Mannschaften ihre Stärken und Schwächen eintragen können sowie auch der gegnerischen Mannschaften.

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 13: Mockup Analysen

Wenn man «Unsere Mannschaft» auswählt, kann man die Stärken und Schwächen der eigenen Mannschaft eintragen.

Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 14: Mockup Analysen - eigene Mannschaft

Wenn man «Gegnerische Mannschaften» auswählt, kann man die Stärken und Schwächen der gegnerischen Mannschaften eintragen. Mit einem Dropdown-Menü soll es möglich sein, die jeweilige Mannschaft auszuwählen.

Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 15: Mockup Analysen - gegnerische Mannschaften

### User Stories

In diesem Kapitel sind die User Stories ersichtlich.

|  |  |
| --- | --- |
| **ID** | **User Story** |
| US-001 | Als Benutzer möchte ich mich registrieren können, damit ich mich einloggen kann. |
| US-002 | Als Benutzer möchte ich mich einloggen können, damit ich die Webseite benutzen kann. |
| US-003 | Als Benutzer möchte ich beim Login und bei der Registrierung darauf hingewiesen werden, wenn ich etwas Falsches eingegeben habe, damit ich die eingegebenen Anmeldedaten korrigieren kann. |
| US-004 | Als Benutzer möchte ich Kalendereinträge machen können, damit ich eine Planung habe. |
| US-005 | Als Benutzer möchte ich den Kalender herunterladen/abonnieren können, damit ich es im persönlichen Kalender importieren kann. |
| US-006 | Als Benutzer möchte ich zur Kalenderseite, Dashboard-Seite und Analysenseite weitergeleitet werden, mithilfe eines Menüs, damit ich auch die anderen Ansichten anschauen kann. |
| US-007 | Als Benutzer möchte ich die Spielresultate eintragen können, damit ich sehen kann, wie die letzten Spiele ausgegangen sind. |
| US-008 | Als Benutzer möchte ich Analysen eintragen können, damit ich weiss, woran unsere Mannschaft arbeiten muss und wie die gegnerischen Mannschaften sind. |
| US-009 | Als Benutzer möchte ich mich ausloggen können, damit ich nicht weiterhin angemeldet bin. |
| US-010 | Als Benutzer möchte ich auf dem Smartphone die Webseite so benutzen können wie auf dem Laptop, damit ich auch die Webseite auf dem Smartphone benutzen kann. |

Tabelle 3: User Stories

### Testkonzept

Für das Testing wurden Google Chrome (Version 122.0.6261.58 (64-Bit)) und Windows 10 Home (Version 22H2) benutzt. Zudem wurde für das Testkonzept das Quadrantenmodell verwendet.

Ein Bild, das Text, Handschrift, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 16: Quadrantenmodell

#### Quadrant 1 (Q1)

Für das Testing werden Unit Tests geschrieben, welche zum 1. Quadrant (Q1) gehören.

#### Quadrant 2 (Q2)

Auch werden für das Testing die User Stories getestet. Das gehört zum 2. Quadrant (Q2).

#### Quadrant 3 (Q3)

Es werden auch manuelle Tests geschrieben, die zum 3. Quadrant (Q3) gehören.

#### Quadrant 4 (Q4)

Zudem werden auch Linters verwendet, die zum 4. Quadrant (Q4) gehören.

#### Testfälle definieren

In diesem Kapitel werden die Testfälle definiert.

##### Manuelle Tests

Jeder manuelle Test enthält eine Referenz zur dazugehörigen User Story.

|  |  |
| --- | --- |
| **ID** | MT-001 |
| **User Story** | US-001 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll eine Registrierungsseite bestehen. |
| **Schritte** | 1. Webseite öffnen 2. Die Option zum Registrieren auswählen |
| **Erwartetes Resultat** | Es besteht eine Registrierungsseite und man kann sich registrieren. |
| **Tatsächliches Resultat** | - |

Tabelle 4: MT-001

|  |  |
| --- | --- |
| **ID** | MT-002 |
| **User Story** | US-002 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll eine Login-Seite bestehen. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen |
| **Erwartetes Resultat** | Es besteht eine Login-Seite und man kann sich einloggen. |
| **Tatsächliches Resultat** | - |

Tabelle 5: MT-002

|  |  |
| --- | --- |
| **ID** | MT-003 |
| **User Story** | US-003 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll darauf hingewiesen werden, wenn die Anmeldedaten falsch sind. |
| **Schritte** | 1. Webseite öffnen 2. Bei der Login-Seite oder Registrierungsseite falsche Anmeldedaten eingeben |
| **Erwartetes Resultat** | Es wird darauf hingewiesen, dass man falsche Anmeldedaten eingegeben hat. |
| **Tatsächliches Resultat** | - |

Tabelle 6: MT-003

|  |  |
| --- | --- |
| **ID** | MT-004 |
| **User Story** | US-004 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es sollen Kalendereinträge gemacht werden können. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Kalender sehen |
| **Erwartetes Resultat** | Es können Kalendereinträge gemacht werden. |
| **Tatsächliches Resultat** | - |

Tabelle 7: MT-004

|  |  |
| --- | --- |
| **ID** | MT-005 |
| **User Story** | US-005 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll der Kalender heruntergeladen/abonniert werden können. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Kalender sehen 4. Kalender abonnieren/herunterladen |
| **Erwartetes Resultat** | Es kann der Kalender heruntergeladen/abonniert werden. |
| **Tatsächliches Resultat** | - |

Tabelle 8: MT-005

|  |  |
| --- | --- |
| **ID** | MT-006 |
| **User Story** | US-006 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll zur gewünschten Seite weitergeleitet werden können. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Menü auswählen |
| **Erwartetes Resultat** | Man soll zur gewünschten Seite weitergeleitet werden können. |
| **Tatsächliches Resultat** | - |

Tabelle 9: MT-006

|  |  |
| --- | --- |
| **ID** | MT-007 |
| **User Story** | US-007 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es sollen Spielresultate eingetragen werden können. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Menü auswählen 4. «Dashboard» im Menü auswählen |
| **Erwartetes Resultat** | Die Spielresultate können eingetragen werden. |
| **Tatsächliches Resultat** | - |

Tabelle 10: MT-007

|  |  |
| --- | --- |
| **ID** | MT-008 |
| **User Story** | US-008 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es sollen Analysen eingetragen werden können. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Menü auswählen 4. «Analysen» im Menü auswählen |
| **Erwartetes Resultat** | Die Analysen können eingetragen werden. |
| **Tatsächliches Resultat** | - |

Tabelle 11: MT-008

|  |  |
| --- | --- |
| **ID** | MT-009 |
| **User Story** | US-009 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll möglich sein, sich auszuloggen. |
| **Schritte** | 1. Webseite öffnen 2. Einloggen 3. Ausloggen |
| **Erwartetes Resultat** | Man kann sich ausloggen. |
| **Tatsächliches Resultat** | - |

Tabelle 12: MT-009

|  |  |
| --- | --- |
| **ID** | MT-010 |
| **User Story** | US-010 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll möglich sein, die Webseite auch auf dem Smartphone zu benutzen. |
| **Schritte** | 1. Webseite auf Smartphone öffnen 2. Webseite benutzen wie auf dem Laptop |
| **Erwartetes Resultat** | Man kann die Webseite auf dem Smartphone benutzen. |
| **Tatsächliches Resultat** | - |

Tabelle 13: MT-010

##### Automatisierte Tests

Jetzt werden die Unit Tests sowie die Linting Tests mit einer Referenz zur dazugehörigen User Story aufgezeigt.

|  |  |
| --- | --- |
| **ID** | UT-001 |
| **User Story** | US-004 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es soll getestet werden, dass das Startdatum nicht grösser sein darf als das Enddatum. |
| **Erwartetes Resultat** | Es wird zurückgegeben, dass das Startdatum nicht grösser sein darf als das Enddatum. |
| **Tatsächliches Resultat** | - |

Tabelle 14: UT-001

|  |  |
| --- | --- |
| **ID** | LT-001 |
| **User Story** | US-003 |
| **Tester** | D. Cvetkovic |
| **Testdatum** | - |
| **Beschreibung** | Es sollen die programmatischen und stilistischen Fehler im Code angezeigt werden. |
| **Erwartetes Resultat** | Es werden programmatische und stilistische Fehler im Code angezeigt. |
| **Tatsächliches Resultat** | - |

Tabelle 15: LT-001

## Datenbank

Für die Datenbank wurde MySQL verwendet, weil es ein Teil von XAMPP ist.

### ERM

### Backup

## Login

Mit einer Login-Seite können sich die Benutzer anmelden, um dann die Webseite zu benutzen. Somit musste zuerst ein Formular erstellt werden, damit sich der Benutzer anmelden kann. Zusätzlich muss das Formular auf die Registrierungsseite verweisen, sollte man kein Account haben.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 17: Login-Implementierung

Wenn der Benutzer keine Anmeldedaten eingegeben hat, wird die Person darauf hingewiesen.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 18: Login-Implementierung – keine Eingaben

Auch erhält man eine Fehlermeldung, wenn man die falschen Anmeldedaten eingegeben hat.Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 19: Login-Implementierung - falsche Anmeldedaten

Der ganze Code zur Login-Implementierung ist im Anhang zu finden.

## Registrierung

Auch besteht die Möglichkeit, sich zu registrieren. Dafür muss man dann auf der Login-Seite «Hier registrieren» auswählen und wird dann dementsprechend zur Registrierungsseite weitergeleitet.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 20: Registrierungsimplementierung

Wenn der Benutzer nichts eingibt, sieht die Person die Meldung.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 21: Registrierungsimplementierung - keine Eingabe

Wenn der Benutzer ungültige Werte eingegeben hat, wird die Person auch darauf hingewiesen.

Ein Bild, das Text, Screenshot, Schrift, Webseite enthält.

Automatisch generierte Beschreibung

Abbildung 22: Registrierungsimplementierung - falsche Werte

Zudem wird der Benutzer darauf aufmerksam gemacht, wenn Werte bei «Passwort» und «Passwort bestätigen» nicht übereinstimmen.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 23: Registrierungsimplementierung - Passwort stimmt nicht überein

Eine weitere Funktion, die implementiert wurde, ist, dass der Benutzer auch darauf hingewiesen wird, wenn der Benutzername schon existiert.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 24: Registrierungsimplementierung - Benutzername existiert bereits

Die registrierten Benutzer werden in der Datenbank angezeigt. Das Passwort ist gehashed.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 25: Registrierter Benutzer in der Datenbank

Wie der ganze Code zur Registrierungsimplementierung aussieht, ist im Anhang zu finden.

## Kalender

Nachdem sich der Benutzer angemeldet hat, soll die Person den Kalender sehen.

Ein Bild, das Text, Screenshot, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 26: Kalenderimplementierung

Das Planungsformular neben dem Kalender ist dafür da, dass man Einträge im Kalender machen kann.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 27: Planungsformular

Nachdem man die Werte, die man im Planungsformular eingegeben hat, gespeichert hat, wird der Eintrag im Kalender hinzugefügt sowie in der Datenbank.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 28: Eintrag im Kalender

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 29: Eintrag vom Kalender in der Datenbank

Wenn man die eingegebenen Werte nochmals anschauen möchte, kann man den Eintrag auswählen und es erscheinen dann die Informationen.

Ein Bild, das Text, Screenshot, Display, Software enthält.

Automatisch generierte Beschreibung

Abbildung 30: Informationen zum Eintrag im Kalender

Der ganze Code zur Kalenderimplementierung ist im Anhang zu finden.

### ICS-Datei

## Menü

Mithilfe des Menüs kann man jeweils zum Kalender, Dashboard oder zur Analysenseite weitergeleitet werden.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 31: Menüimplementierung

Dabei wurden Funktionen implementiert, die das Menü öffnen und schliessen.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 32: Menü öffnen und schliessen - Code

Im Anhang befindet sich der ganze Code zum Menü.

## Dashboard

Auch kann man auf der Webapplikation die Spielresultate und weitere Notizen eintragen, die dann auf einem Dashboard angezeigt werden. Dabei muss zuerst ein neuer Eintrag erstellt werden.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 33: Dashboard-Implementierung

Nachdem «Eintrag hinzufügen» gedrückt wurde, kann der neue Eintrag erstellt werden.

Ein Bild, das Text, Screenshot, Zahl, Software enthält.

Automatisch generierte Beschreibung

Abbildung 34: Eintrag auf Dashboard hinzufügen

Wenn der neue Eintrag gespeichert wurde, dann ist es auf dem Dashboard ersichtlich sowie in der Datenbank.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 35: Eintrag auf Dashboard ersichtlich



Abbildung 36: Eintrag vom Dashboard in der Datenbank

Die Notizen kann man sich anschauen, wenn man «Bearbeiten» auswählt.

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 37: Notizen auf Dashboard anschauen

Zudem besteht die Möglichkeit, den Eintrag zu bearbeiten. Wenn dieser bearbeitet wird, wird es auf dem Dashboard sowie in der Datenbank aktualisiert. Ebenso kann man den Eintrag löschen. Dabei wird der Eintrag aus dem Dashboard gelöscht sowie aus der Datenbank.

Der ganze Code zur Dashboard-Implementierung ist im Anhang zu finden.

## Analysen

## Deployment

Der ganze Code wurde auf der Webseite «Hostpoint» deployed. Die Domäne heisst «zuzocidi.myhostpoint.ch». Damit ich aber schlussendlich meine Webapplikation auf Hostpoint benutzen konnte, musste ich folgende URL eingeben:

<https://zuzocidi.myhostpoint.ch/src/>

Wie man den Code auf Hostpoint deployed und was sonst noch zu beachten ist, findet man in der Anleitung, welche sich im Anhang befindet.

## Testdurchführung

### Testauswertung

## Ausführungsanleitung

In diesem Kapitel wird aufgezeigt, wie man die Webseite benutzt und wie man den Code testet. Im [GitHub-Repository](https://github.com/DC2305/Fussball-Webapplikation) gibt es ebenfalls eine Anleitung dazu als README. Dort wird zusätzlich noch erklärt, was man alles installieren muss.

### Webseite öffnen (XAMPP)

Damit man die Webseite lokal mit XAMPP öffnen kann, muss man in der URL folgendes eingeben:

localhost/Fussball-Webapplikation/src/

### Unit Tests ausführen

Um die Unit Tests von Jest auszuführen, muss man im Terminal im Ordner der Webapplikation gehen und folgendes eingeben:



Abbildung 38: Jest-Tests ausführen

Die Unit Tests von PHPUnit führt man so aus:



Abbildung 39: PHPUnit-Tests ausführen

### Linters verwenden

Um den Linter «ESLint» für die Datei «script.js» zu verwenden, muss man folgendes im Terminal eingeben:



Abbildung 40: ESLint verwenden

Den Linter «PHP\_CodeSniffer» verwendet man so, wenn man zum Beispiel die Datei «config.php» linten möchte:



Abbildung 41: PHP\_CodeSniffer verwenden

Damit man den Linter «Stylelint» korrekt verwendet, muss man folgendes im Terminal eingeben, wenn man beispielsweise die Datei «style.css» linten möchte:



Abbildung 42: Stylelint verwenden

## Schlusswort

## Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **Abkürzung** | **Bedeutung** |
| CD | Continuous Delivery/Deployment |
| CI | Continuous Inegration |
| CSS | Cascading Style Sheets |
| ERM | Entity Relationship Model |
| HTML | Hypertext Markup Language |
| ICS | Internet Calendar Scheduling |
| IPERKA | Informieren, Planen, Entscheiden, Realisieren, Kontrollieren, Auswerten |
| JS | JavaScript |
| LAMP | Linux, Apache, MySQL, PHP |
| LT | Linting Test |
| MT | Manueller Test |
| PA | Praktische Arbeit |
| PHP | PHP: Hypertext Preprocessor |
| SQL | Structured Query Language |
| URL | Uniform Resource Locator |
| US | User Story |
| UT | Unit Test |
| WAMP | Windows, Apache, MySQL, PHP |
| XAMPP | Cross-Platform, Apache, MySQL, PHP, Perl |
| YAML | YAML Ain’t Markup Language |

Tabelle 16: Abkürzungsverzeichnis

## Glossar

|  |  |
| --- | --- |
| **Begriff** | **Bedeutung** |
| Deployment | Es bedeutet die Bereitstellung von Software. |
| Domäne | Bei einer Domäne handelt es sich um eine Internetadresse. |
| Dropdown-Menü | Mit einem Dropdown-Menü kann man einen Wert aus einer Liste auswählen. |
| Error-Log | Ein Error-Log ist ein Fehlerprotokoll. |
| ESLint | Das ist ein Linter für JS. |
| FullCalendar | FullCalendar ist eine JS-Library, mit der man Kalender erstellen kann. |
| Gehashed | Wenn das Passwort «gehashed» ist, bedeutet das, dass das Passwort verschlüsselt wurde. |
| Git | Ein Programm, welches ermöglicht, Code über den Terminal auf GitHub hochzuladen. |
| GitHub | Eine Entwicklerplattform, die es Entwicklern ermöglicht, ihren Code zu erstellen, zu speichern, zu verwalten und zu teilen. |
| GitHub Actions | Es ist eine Plattform für CI/CD, mit der man die Workflows automatisieren kann. |
| Hostpoint | Das ist ein Webhosting-Anbieter. |
| Jest | Das ist ein Testing Framework für JS. |
| Library | Eine Sammlung von Ressourcen, die während der Softwareentwicklung zur Implementierung eines Computerprogramms genutzt werden. Die Ressourcen können unter anderem Funktionen enthalten. |
| Linter | Ein Tool, das Quellcode auf Fehler, Schwachstellen und Stilprobleme analysiert, um die Codequalität zu verbessern. |
| Linting | Das Verfahren, wenn man den Quellcode auf Fehler, Schwachstellen und Stilprobleme analysiert. |
| Linting Test | Das sind die Tests zum Linting. |
| Manuelle Tests | Die Tests werden von dem Menschen selbst durchgeführt und nicht automatisch. |
| Open-Source | Das ist der Quellcode, der zur möglichen Änderung und Weiterverbreitung frei verfügbar gemacht wird. |
| PHP\_CodeSniffer | Das ist ein Linter für PHP. |
| PHPUnit | Das ist ein Testing Framework für PHP. |
| README | Ein README ist eine Datei, die Informationen über das Projekt auf GitHub enthält. |
| Repository | Ein Repository enthält den ganzen Code, die Dateien und den Revisionsverlauf jeder Datei. Es ist wie ein Projektordner auf GitHub. |
| Stylelint | Das ist ein Linter für CSS. |
| Terminal | In einem Terminal kann man Befehle eingeben, Dateien bearbeiten, Programme ausführen und Dokumente öffnen. |
| Testfall | In einem Testfall wird festgehalten, was getestet werden soll. |
| Testing Framework | Eine Reihe von Richtlinien oder Regeln, die zum Erstellen und Entwerfen von Testfällen verwendet werden. |
| Testkonzept | Es beschreibt, wie die Testfälle getestet werden und was dabei verwendet wird. |
| Unit Tests | Es werden einzelne Teile vom Code getestet, ohne Abhängigkeiten. Die Tests werden automatisch ausgeführt. |
| User Stories | Eine kurze Beschreibung, bei der ersichtlich ist, was der Benutzer will. |
| Webhosting | Ein Onlinedienst, der es ermöglicht, Webseiten oder Webanwendungen im Internet zu veröffentlichen. |
| Workflow | Bei den GitHub Actions ist ein Workflow ein konfigurierbarer automatisierter Prozess, der ein Schritt oder mehrere Schritte ausführt. |

Tabelle 17: Glossar

## Quellenverzeichnis

Elsener, A. (19. Februar 2024). Quadrantenmodell. BBZW Sursee - Modul 450.

Funda of Web IT. (26. Februar 2024). *Funda of Web IT*. Von Funda of Web IT: https://www.fundaofwebit.com/php-solutions/php-crud-using-bootstrap-model-in-php abgerufen

GitHub Actions. (21. Februar 2024). *GitHub Actions*. Von GitHub Actions Jest: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-nodejs abgerufen

GitHub Actions. (21. Februar 2024). *GitHub Actions*. Von GitHub Actions PHPUnit: https://github.com/php-actions/phpunit abgerufen

GitHub, Inc. (19. Februar 2024). *GitHub*. Von GitHub: https://github.com/features/actions abgerufen

Saladino, G. (19. Februar 2024). *Perforce Software*. Von Perforce Software: https://www.perforce.com/blog/qac/what-is-linting abgerufen

Super-Linter. (21. Februar 2024). *GitHub Super-Linter*. Von Super-Linter: https://github.com/super-linter/super-linter abgerufen

Synopsys, Inc. (19. Februar 2024). *Synopsys*. Von Synopsys: https://www.synopsys.com/glossary/what-is-cicd.html abgerufen

## Abbildungsverzeichnis

[Abbildung 1: IPERKA 8](#_Toc160013725)

[Abbildung 2: Zeitplan 9](#_Toc160013726)

[Abbildung 3: CI/CD 16](#_Toc160013727)

[Abbildung 4: Kontextdiagramm 17](#_Toc160013728)

[Abbildung 5: Mockup Login 18](#_Toc160013729)

[Abbildung 6: Mockup Registrierung 19](#_Toc160013730)

[Abbildung 7: Mockup Kalender 19](#_Toc160013731)

[Abbildung 8: Mockup Menü 20](#_Toc160013732)

[Abbildung 9: Mockup Dashboard 20](#_Toc160013733)

[Abbildung 10: Mockup Dashboard - Neuer Eintrag 21](#_Toc160013734)

[Abbildung 11: Mockup Dashboard - Eintrag hinzugefügt 21](#_Toc160013735)

[Abbildung 12: Mockup Dashboard - Notizen 22](#_Toc160013736)

[Abbildung 13: Mockup Analysen 22](#_Toc160013737)

[Abbildung 14: Mockup Analysen - eigene Mannschaft 23](#_Toc160013738)

[Abbildung 15: Mockup Analysen - gegnerische Mannschaften 24](#_Toc160013739)

[Abbildung 16: Quadrantenmodell 26](#_Toc160013740)

[Abbildung 17: Login-Implementierung 32](#_Toc160013741)

[Abbildung 18: Login-Implementierung – keine Eingaben 33](#_Toc160013742)

[Abbildung 19: Login-Implementierung - falsche Anmeldedaten 34](#_Toc160013743)

[Abbildung 20: Registrierungsimplementierung 35](#_Toc160013744)

[Abbildung 21: Registrierungsimplementierung - keine Eingabe 36](#_Toc160013745)

[Abbildung 22: Registrierungsimplementierung - falsche Werte 37](#_Toc160013746)

[Abbildung 23: Registrierungsimplementierung - Passwort stimmt nicht überein 38](#_Toc160013747)

[Abbildung 24: Registrierungsimplementierung - Benutzername existiert bereits 39](#_Toc160013748)

[Abbildung 25: Registrierter Benutzer in der Datenbank 39](#_Toc160013749)

[Abbildung 26: Kalenderimplementierung 40](#_Toc160013750)

[Abbildung 27: Planungsformular 40](#_Toc160013751)

[Abbildung 28: Eintrag im Kalender 41](#_Toc160013752)

[Abbildung 29: Eintrag vom Kalender in der Datenbank 41](#_Toc160013753)

[Abbildung 30: Informationen zum Eintrag im Kalender 42](#_Toc160013754)

[Abbildung 31: Menüimplementierung 44](#_Toc160013755)

[Abbildung 32: Menü öffnen und schliessen - Code 44](#_Toc160013756)

[Abbildung 33: Dashboard-Implementierung 45](#_Toc160013757)

[Abbildung 34: Eintrag auf Dashboard hinzufügen 45](#_Toc160013758)

[Abbildung 35: Eintrag auf Dashboard ersichtlich 46](#_Toc160013759)

[Abbildung 36: Eintrag vom Dashboard in der Datenbank 46](#_Toc160013760)

[Abbildung 37: Notizen auf Dashboard anschauen 46](#_Toc160013761)

[Abbildung 38: Jest-Tests ausführen 51](#_Toc160013762)

[Abbildung 39: PHPUnit-Tests ausführen 51](#_Toc160013763)

[Abbildung 40: ESLint verwenden 51](#_Toc160013764)

[Abbildung 41: PHP\_CodeSniffer verwenden 51](#_Toc160013765)

[Abbildung 42: Stylelint verwenden 51](#_Toc160013766)

## Tabellenverzeichnis

[Tabelle 1: PA-Steckbrief 2](#_Toc160013767)

[Tabelle 2: Meilensteine 10](#_Toc160013768)

[Tabelle 3: User Stories 25](#_Toc160013769)

[Tabelle 4: MT-001 27](#_Toc160013770)

[Tabelle 5: MT-002 27](#_Toc160013771)

[Tabelle 6: MT-003 27](#_Toc160013772)

[Tabelle 7: MT-004 28](#_Toc160013773)

[Tabelle 8: MT-005 28](#_Toc160013774)

[Tabelle 9: MT-006 28](#_Toc160013775)

[Tabelle 10: MT-007 29](#_Toc160013776)

[Tabelle 11: MT-008 29](#_Toc160013777)

[Tabelle 12: MT-009 29](#_Toc160013778)

[Tabelle 13: MT-010 30](#_Toc160013779)

[Tabelle 14: UT-001 30](#_Toc160013780)

[Tabelle 15: LT-001 30](#_Toc160013781)

[Tabelle 16: Abkürzungsverzeichnis 53](#_Toc160013782)

[Tabelle 17: Glossar 55](#_Toc160013783)

## Anhang

Die Anhänge sind in der Datei «PA-Bericht\_mit\_Anhängen\_Cvetkovic\_David.zip».