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

**PA-Bericht**

Webapplikation für Fussballvereine

David Cvetkovic

20.02.2024

**PA-Steckbrief**

|  |  |
| --- | --- |
| **Merkmale** | **Beschreibung** |
| Projektname | Webapplikation für Fussballverein |
| 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](#_Toc159322510)

[1.1 Aufgabenstellung 6](#_Toc159322511)

[1.1.1 Detaillierte Aufgabenstellung 6](#_Toc159322512)

[1.1.2 Benutzte Software 7](#_Toc159322513)

[1.1.3 Benutzte Hardware 7](#_Toc159322514)

[1.1.4 Methode 7](#_Toc159322515)

[1.1.5 Vorkenntnisse 7](#_Toc159322516)

[1.1.6 Vorarbeiten 7](#_Toc159322517)

[1.2 Projektorganisation 8](#_Toc159322518)

[1.2.1 Projektmanagementmethode 8](#_Toc159322519)

[1.2.2 Firmenstandards 8](#_Toc159322520)

[1.2.3 Organisation der Arbeitsergebnisse 8](#_Toc159322521)

[1.2.3.1 Versionierung 8](#_Toc159322522)

[1.2.3.2 Dokumentenablage 8](#_Toc159322523)

[1.3 Zeitplan 9](#_Toc159322524)

[1.3.1 Meilensteine 10](#_Toc159322525)

[1.4 Arbeitsprotokoll 11](#_Toc159322526)

[1.4.1 Tag 01: 19.02.2024 11](#_Toc159322527)

[1.4.2 Tag 02: 20.02.2024 11](#_Toc159322528)

[1.4.3 Tag 03: 21.02.2024 11](#_Toc159322529)

[1.4.4 Tag 04: 22.02.2024 11](#_Toc159322530)

[1.4.5 Tag 05: 23.02.2024 11](#_Toc159322531)

[1.4.6 Tag 06: 26.02.2024 11](#_Toc159322532)

[1.4.7 Tag 07: 27.02.2024 11](#_Toc159322533)

[1.4.8 Tag 08: 28.02.2024 11](#_Toc159322534)

[1.4.9 Tag 09: 29.02.2024 11](#_Toc159322535)

[1.4.10 Tag 10: 01.03.2024 11](#_Toc159322536)

[2 Teil 2 12](#_Toc159322537)

[2.1 Kurzfassung 12](#_Toc159322538)

[2.1.1 Kurze Ausgangssituation 12](#_Toc159322539)

[2.1.2 Umsetzung 12](#_Toc159322540)

[2.1.3 Ergebnis 12](#_Toc159322541)

[2.2 Wissensbeschaffung 13](#_Toc159322542)

[2.2.1 CI/CD 13](#_Toc159322543)

[2.2.1.1 GitHub Actions 13](#_Toc159322544)

[2.2.1.2 Linting 13](#_Toc159322545)

[2.3 Vorbereitung zur Umsetzung 14](#_Toc159322546)

[2.3.1 Kontextdiagramm 14](#_Toc159322547)

[2.3.2 Mockup 14](#_Toc159322548)

[2.3.2.1 Login-Ansicht 14](#_Toc159322549)

[2.3.2.2 Registrierungsansicht 14](#_Toc159322550)

[2.3.2.3 Kalenderansicht 14](#_Toc159322551)

[2.3.2.4 Dashboard-Ansicht 14](#_Toc159322552)

[2.3.2.5 Analysenansicht 14](#_Toc159322553)

[2.3.3 User Stories 14](#_Toc159322554)

[2.3.4 Testkonzept 14](#_Toc159322555)

[2.3.4.1 Quadrant 1 (Q1) 14](#_Toc159322556)

[2.3.4.2 Quadrant 2 (Q2) 15](#_Toc159322557)

[2.3.4.3 Quadrant 3 (Q3) 15](#_Toc159322558)

[2.3.4.4 Quadrant 4 (Q4) 15](#_Toc159322559)

[2.3.4.5 Testfälle definieren 15](#_Toc159322560)

[2.3.4.5.1 Manuelle Tests 15](#_Toc159322561)

[2.3.4.5.2 Unit Tests 15](#_Toc159322562)

[2.4 Datenbank 16](#_Toc159322563)

[2.4.1 ERM 16](#_Toc159322564)

[2.4.2 Backup 16](#_Toc159322565)

[2.5 Login und Registrierung 17](#_Toc159322566)

[2.6 Kalender 18](#_Toc159322567)

[2.6.1 ICS-Datei 18](#_Toc159322568)

[2.7 Dashboard 19](#_Toc159322569)

[2.8 Analysen 20](#_Toc159322570)

[2.9 Deployment 21](#_Toc159322571)

[2.10 Unit Tests 22](#_Toc159322572)

[2.11 Testdurchführung 23](#_Toc159322573)

[2.11.1 Testauswertung 23](#_Toc159322574)

[2.12 Ausführungsanleitung 24](#_Toc159322575)

[2.13 Schlusswort 25](#_Toc159322576)

[2.14 Abkürzungsverzeichnis 26](#_Toc159322577)

[2.15 Glossar 27](#_Toc159322578)

[2.16 Quellenverzeichnis 28](#_Toc159322579)

[2.17 Abbildungsverzeichnis 29](#_Toc159322580)

[2.18 Tabellenverzeichnis 30](#_Toc159322581)

[2.19 Anhang 31](#_Toc159322582)

# 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

### 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, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 2: Zeitplan

### Meilensteine

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Meilenstein** | **IPERKA-Phase** | **Beschreibung** |
| 22.02.2024 | Meilenstein 1 (M1) | Realisieren | Login und Registrierung implementiert |
| 23.02.2024 | Meilenstein 2 (M2) | Realisieren | Kalender implementiert |
| 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

### Tag 03: 21.02.2024

### Tag 04: 22.02.2024

### Tag 05: 23.02.2024

### Tag 06: 26.02.2024

### Tag 07: 27.02.2024

### 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

### Umsetzung

### Ergebnis

## 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.

### Mockup

#### Login-Ansicht

#### Registrierungsansicht

#### Kalenderansicht

#### Dashboard-Ansicht

#### Analysenansicht

### User Stories

### Testkonzept

Für das Testing wurden Google Chrome (Version 121.0.6167.185 (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 4: 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

##### Manuelle Tests

##### Unit Tests

## Datenbank

### ERM

### Backup

## Login und Registrierung

## Kalender

### ICS-Datei

## Dashboard

## Analysen

## Deployment

## Unit Tests

## Testdurchführung

### Testauswertung

## Ausführungsanleitung

## 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 |
| PA | Praktische Arbeit |
| PHP | PHP: Hypertext Preprocessor |
| SQL | Structured Query Language |
| WAMP | Windows, Apache, MySQL, PHP |
| XAMPP | Cross-Platform, Apache, MySQL, PHP, Perl |

Tabelle 3: Abkürzungsverzeichnis

## Glossar

|  |  |
| --- | --- |
| **Begriff** | **Bedeutung** |
| 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. |
| Linter | Ein Tool, das Quellcode auf Fehler, Schwachstellen und Stilprobleme analysiert, um die Codequalität zu verbessern. |
| Manuelle Tests | Die Tests werden von dem Menschen selbst durchgeführt und nicht automatisch. |
| Repository | Ein Repository enthält den ganzen Code, die Dateien und den Revisionsverlauf jeder Datei. Es ist so wie ein Projektordner auf GitHub. |
| Testfall | In einem Testfall wird festgehalten, was getestet werden soll. |
| 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. |

Tabelle 4: Glossar

## Quellenverzeichnis

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

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

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

## Abbildungsverzeichnis

[Abbildung 1: IPERKA 8](#_Toc159322506)

[Abbildung 2: Zeitplan 9](#_Toc159322507)

[Abbildung 3: CI/CD 13](#_Toc159322508)

[Abbildung 4: Quadrantenmodell 14](#_Toc159322509)

## Tabellenverzeichnis

[Tabelle 1: PA-Steckbrief 2](#_Toc159322502)

[Tabelle 2: Meilensteine 10](#_Toc159322503)

[Tabelle 3: Abkürzungsverzeichnis 26](#_Toc159322504)

[Tabelle 4: Glossar 27](#_Toc159322505)

## Anhang

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