

TDTool (To Do Tool)

Verwendete Technologien: HTML, CSS, PHP, SQL

Projektdauer: 05.06.2018 - dd.mmm yyyy

Projektteam: Marcus Heri (Projektleiter)

Inhaltsverzeichnis

[Arbeitsjournal 2](#_Toc517346486)

[Informieren 5](#_Toc517346487)

[Planen 6](#_Toc517346488)

[Entscheiden 7](#_Toc517346489)

[Realisieren 8](#_Toc517346490)

[Kontrollieren 9](#_Toc517346491)

[Auswerten 9](#_Toc517346492)

Arbeitsjournal

Mittwoch 05. Juni 2018

|  |  |
| --- | --- |
| Tagesziele | * Themenwahl * Brainstorm |
| Erreichung Ziele,  Erfolgserlebnisse | *Vormittag:*  Ich habe sehr viel nachgedacht, was ich in der Arbeit noch nicht habe und habe somit mich für ein To Do Tool entschieden. Auch habe ich mir schon ein bisschen Gedanken über die Programmstruktur gemacht. |
| Herausforderungen | Noch keine Herausforderungen. |
| Probleme | Bisher noch keine Probleme. |
| Lösungen | Keine Probleme keine Lösungen :D |
| Durchgeführte Tests | Ich habe noch keine Testfälle definiert. |
| Wissensbeschaffung | - |
| Beanspruchte Hilfeleistung | Keine Hilfeleistung benötigt. |
| Vergleich mit dem Zeitplan | Zeitplan noch nicht definiert. |
| Persönliche Tagesreflexion | Ich habe beide meine Tagesziele erfolgreich erreicht. |

Mittwoch 12. Juni 2018

|  |  |
| --- | --- |
| Tagesziele | * UML erstellen und umsetzen * Test schreiben |
| Erreichung Ziele,  Erfolgserlebnisse | *Vormittag:*  Wir hatten einen Test und ich habe das UML erstellt dies muss aber noch von Jerome abgesegnet werden. |
| Herausforderungen | Delegation und Aggregation nicht verwechseln. |
| Probleme | Keine Probleme. |
| Lösungen | Keine Lösungen |
| Durchgeführte Tests | Ist das UML richtig aufgebaut. |
| Wissensbeschaffung | Internet: Unterschied Delegation und Aggregation |
| Beanspruchte Hilfeleistung | Keine Hilfestellungen benötigt. |
| Vergleich mit dem Zeitplan | Noch keinen Zeitplan definiert. |
| Persönliche Tagesreflexion | Ich zweifle langsam das wir mit dem Tempo das Projekt abschliessen. Jerome hat immer noch nicht die Dokumentation oder den Projektauftrag geschrieben. |

Mittwoch 19. Juni 2018

|  |  |
| --- | --- |
| Tagesziele | * Startseite / Login programmiert und responsive |
| Erreichung Ziele,  Erfolgserlebnisse | *Vormittag:*  Ich habe die Startseite und das Login programmiert, da bei diesen Seiten MVC und einen Entwurf nicht von Nöten ist. |
| Herausforderungen | Mit Mobile-First Methode auszukommen (erstes Mal Verwendung dieser Methode). |
| Probleme | Registrierung eines Userers muss noch programmiert werden. |
| Lösungen | Verwendung des Grid Systems wird dies erleichtern. |
| Durchgeführte Tests | Ist die Seite responsive? |
| Wissensbeschaffung | Keine Wissensbeschaffung gebraucht. |
| Beanspruchte Hilfeleistung | - |
| Vergleich mit dem Zeitplan | Es drückt jetzt noch mehr am Schuh ich werde höchstwahrscheinlich noch zuhause arbeiten müssen |
| Persönliche Tagesreflexion | Da Jerome immer noch nicht mit der Dokumentation fetig ist habe ich entschieden das Projekt alleine fortzusetzen, aus diesem Grund:  Wenn ich schon alles mache soll er nicht die gleiche Note bekommen. |

Donnerstag 21. Juni 2018

|  |  |
| --- | --- |
| Tagesziele | * Doku abschliessen * Serverseitig validieren und register aufbauen |
| Erreichung Ziele,  Erfolgserlebnisse | *Vormittag:*  Doku abgeschlossen (bis auf Zeitplan)  *Nachmittag:*  Ich bin nicht ganz fertig geworden die Einfügung in die Datenbank bugt noch rum. |
| Herausforderungen | Daten in die Datenbank einfügen |
| Probleme | Seite refreshed mit Ajax |
| Lösungen | Onclick verwenden und onsubmit rausnehmen |
| Durchgeführte Tests | Fügt es die Daten in die Datenbank ein? |
| Wissensbeschaffung | <https://www.youtube.com/watch?v=xb8aad4MRx8>  Datenbank Connection |
| Beanspruchte Hilfeleistung | Andreas (Lehrmeister, konnte nicht weiterhelfen) |
| Vergleich mit dem Zeitplan | Bis jetzt habe ich noch genug Zeit |
| Persönliche Tagesreflexion | Ich konnte die Registrierung einfach responsive machen was toll ist. |

Freitag 22. Juni 2018

|  |  |
| --- | --- |
| Tagesziele | * Zeitplan * Use-Case Diagramm * Register optimieren * Login Seite Fertigstellen |
| Erreichung Ziele,  Erfolgserlebnisse | *Vormittag:*  Ich habe den Zeitplan gemacht und das Register optimiert.  *Nachmittag:*  Use-Case Diagramm und Login Seite Fertigstellen |
| Herausforderungen | Passwort encripten |
| Probleme | keine |
| Lösungen | keine |
| Durchgeführte Tests | Wird das Login abgespeichert? |
| Wissensbeschaffung | <https://www.youtube.com/watch?v=xb8aad4MRx8> |
| Beanspruchte Hilfeleistung | Nils beim leiten auf die chat.php |
| Vergleich mit dem Zeitplan | Ich bin wieder in Time |
| Persönliche Tagesreflexion | Ich bin super stolz was ich alles heute erreicht habe. |

Informieren

Ausgangslage

**Ausgangslage**

Eine To Do Liste, um Projekte zu managen welche Benutzer erstellen oder Löschen können. Diese Projekte beinhalten, welche auch von dem Nutzer erstellt, bearbeitet oder gelöscht werden können.

Probleme bei diesem Projekt wird sein Zeitaktuelle Messager Funktion für den Kunden einzufügen.

Projektauftrag

**Ist-Situation**

Jetzt habe ich entschieden dieses Projekt mit dem Pattern MVC umzusetzen, aus dem Grund das es Wartbarer ist. Dieses Projekt ist dazu da, um ein zu testen, ob wir ein richtiges Login programmieren können und auch die Felder überprüfen (User und Serverseitig).

**Ziellösung**

Mann soll bei meinem Projekt sich anmelden können und dann ein Projekt starten in dem man

Kartei-Karten rumschieben kann. Es wird auch einen allgemeinen Chat geben, bei welchen man Fragen stellen kann.

Verfeinerung des Auftrages

**Überprüfung der Eingabe**

Die Eingaben beim Login und bei der Registrierung werden zuerst überprüft bevor sie einen Account erstellen können.

**Projekte**

Mann kann Projekte erstellen oder löschen.

**Chat**

Mann hat einen Chat indem Fragen gestellt werden können.

Planen

--Zeitplan--

Entscheiden

**Variante** A (MVC)

Vorteile bei diesem Pattern sind, dass das System sehr einfach wartbar ist. Bei kleinen Anwendungen ist es meist sehr viel Aufwand.

**Variante B (MVVM)**

Mann kann Daten aus der Datenbank einfach anpassen. Verbraucht erheblichen Speicherbedarf.

**Variantenvergleich / Entscheidungstabelle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kriterium | Gewichtung  In % | MVC | | MVVM | |
| Wert | Gewichteter Wert | Wert | Gewichteter Wert |
| Komplexität | 50 | 3 | 150 | 3 | 150 |
| Wartbarkeit | 30 | 3 | 90 | 2 | 60 |
| Durchführungsdauer | 20 | 2 | 40 | 2 | 40 |
| Total | 100 % |  |  |  |  |
| **Variantentotal** |  |  | **280** |  | **250** |

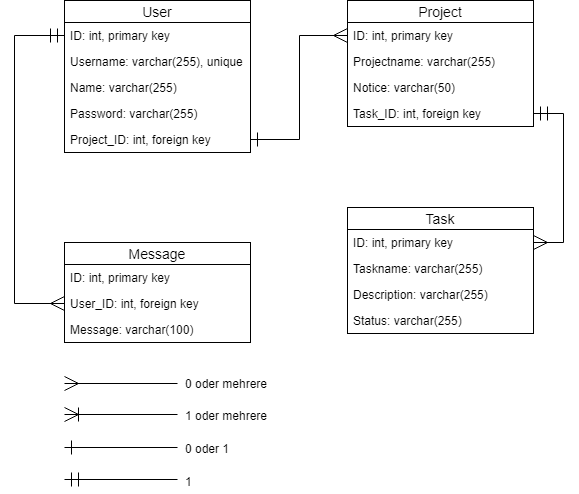
Wertebereich: 1 – 3 Punkte

* 1 Punkt: Dieser Wert wird für eine Variante gesetzt, wenn ein Kriterium nicht oder ungenügend erfüllt oder negative Auswirkungen mit sich bringen könnte.
* 2 Punkte: Dieser Wert wird verwendet um Lösungen die sich neutral verhalten zu markieren.
* 3 Punkte: Dieser Wert wird für eine Variante gesetzt, wenn ein Kriterium positiv / gut erfüllt und es sich lohnt die Variante für das entsprechende Kriterium zu benützen.

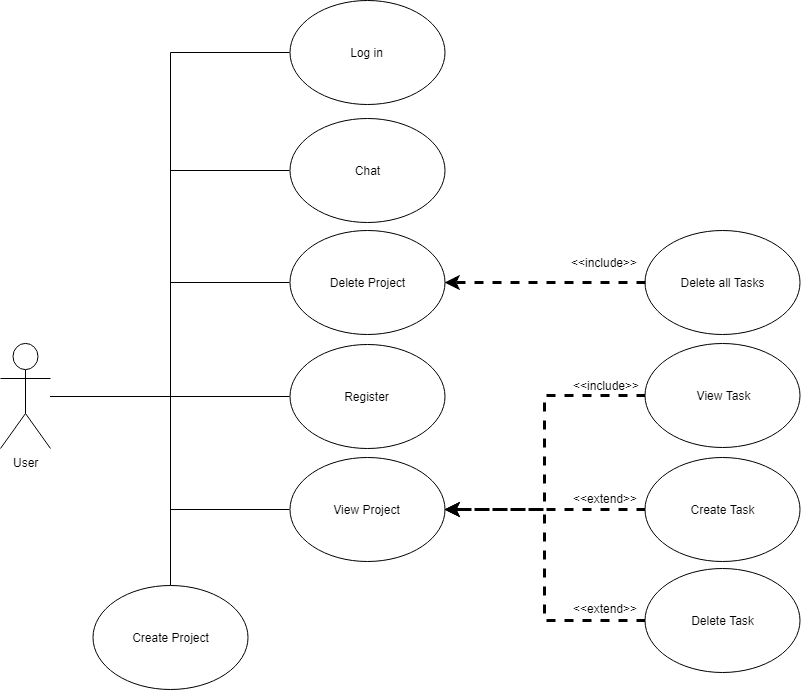
**Variantenentscheid**

Ich habe mich für das Pattern MVC entschieden, da es Wartbarer ist als MVVM.

Realisieren



----Mockups im Ordner Entwurf----



Kontrollieren

**Bugfixing**

Hier folgt eine Beschreibung was am Produkt angepasst und geändert werden muss, um allfällige Fehler zu beheben.

Manchmal kommt es vor, dass man einen Fehler findet, den man nicht beheben kann. Dann ist es wichtig hier festzuhalten, warum dies nicht geht und was für einen Einfluss dies hat.

Falls man Fehler gefunden und behoben hat, muss ein zweiter (oder dritter) Testdurchlauf mit den entsprechenden Testprotokollen erfolgen.

Auswerten

Fazit

War das Projekt erfolgreich? Konnte alles wie gewünscht umgesetzt werden? Wie gut war die eigene Zeitplanung? Benötigt das Projekt eine Weiterentwicklung? Was hat man gelernt? Welche Probleme musste man lösen? Ist man mit seiner Arbeit zufrieden? Offene Punkte? Was fehlt noch?

