

MODUL 120 - PROJEKTARBEIT

Userlogbuch



09. JANUAR 2021
RICHARD LEIST

1 Inhaltsverzeichnis

Inhalt

1	Inhaltsverzeichnis	2
2	Umsetzungsziel.....	3
2.1	Formelles Umsetzungsziel.....	3
2.2	An das Projekt gerichtete Umsetzungsziele	3
3	User Stories.....	3
3.1	Prioritäten Tabelle.....	3
3.2	Informationen.....	3
3.3	Einzelne User Stories	3
4	Arbeitspakete und Zeitplanung.....	4
4.1	Einleitung.....	4
4.2	Visuelle Darstellung als Diagramm	4
5	Containerentwurf	5
6	GUI Entwurf.....	5
7	GUI-Umsetzung	6
7.1	Auffindbarkeit.....	6
7.2	Zusätzlicher Kommentar	6
8	MVC Dokumentation und Klassendiagramm	6
8.1	Textuelle Beschreibung der MVC Umsetzung.....	6
8.2	Klassendiagramm	6
8.3	Quellen und Erkenntnisse.....	6
9	Umsetzung der Logik.....	7
9.1	Auffindbarkeit.....	7
9.2	Angaben zur Entwicklung	7
9.2.1	Tools	7
9.2.2	Quellen	7
10	Reflexion / Erkenntnisse	7

2 Umsetzungsziel

2.1 Formelles Umsetzungsziel

Das Projekt "Userlogbuch" wird im Rahmen des Unterrichts im Modul 120 implementiert und wird mit einer Note bewertet. Das formelle Ziel ist es, den Kriterien des Moduls gerecht zu werden und ein entsprechend, mit guter Qualität, realisiertes Projekt auf die Beine zu stellen und fristgerecht abgeben zu können.

2.2 An das Projekt gerichtete Umsetzungsziele

Das Projekt "Userlogbuch" soll ein Management-Tool werden. Es besitzt die Eigenschaft Kontakte einzutragen und zu löschen. Das Programm wird nach MVVM umgesetzt. Die genauen Anforderungen an das Projekt werden in den User-Stories erläutert.

3 User Stories

3.1 Prioritäten Tabelle

Priorität 1: Must-Have
Priorität 2: Nice-to-have
Priorität 3: Bonus

3.2 Informationen

Die Akzeptanzkriterien sind jeweils in den "Satz mit den drei Elementen" integriert, deswegen gibt es für diese keine extra aufgeführte Zeile.

3.3 Einzelne User Stories

US-01: Kontakt hinzufügen

Als Nutzer will ich die Möglichkeit haben, einen Kontakt hinzuzufügen, um mein Kontaktbuch zu erweitern.

Priorität: 1

Grobaufwand: 20min

US-02: Kontakt hinzufügen (Attribute)

Als Nutzer will ich die Möglichkeit haben, einem Kontakt die Attribute Vorname, Name, Adresse usw. zuzuweisen, um die wichtigsten Daten zu speichern. (Weitere Attribute wären ein "Bonus" und wären der Priorität 3 zuzuordnen)

Priorität: 1

Grobaufwand: 30min

US-03: Kontaktdaten verändern

Als Nutzer will ich die Möglichkeit haben, einen Kontakt zu bearbeiten, um nicht den Kontakt löschen und wieder neu erstellen zu müssen.

Priorität: 1

Grobaufwand: 20min

US-04: Kontakt löschen

Als Nutzer will ich die Möglichkeit haben, einen Kontakt zu löschen, um nicht mehr benötigte Daten zu löschen und für Übersicht zu sorgen.

Priorität: 1

Grobaufwand: 20min

US-05: Kontaktbuch einsehen

Als Nutzer will ich die Möglichkeit haben, in einem extra Reiter alle eingetragenen Kontakte inklusiver deren Daten einsehen zu können, um eine Gesamtübersicht zu haben.

Priorität: 2

Grobaufwand: 40min

US-06: Komplettes Kontaktbuch löschen

Als Nutzer will ich die Möglichkeit haben, das komplette Kontaktbuch zu löschen, um mir einen Mehraufwand, welcher durch einzelnes Löschen entstehen würde, zu sparen.

Priorität: 2

Grobaufwand: 20min

US-07: Listview / Listbox mit allen Kontakten

Als Nutzer will ich, dass alle Kontakte in einer Listview oder Listbox abgelagert sind, um eine bessere Übersicht zu haben.

Priorität: 1

Grobaufwand: 30min

US-08: Darkmode

Als Nutzer will ich, dass ich zwischen einem Darkmode und einem Lightmode wechseln kann, um beispielsweise den Akkuverbrauch zu verringern oder die Augen zu schonen.

Priorität: 3

Grobaufwand: 30min

US-09: Suchfunktion im Listview / Listbox

Als Nutzer will ich, dass es eine Funktion gibt, mit welcher ich die Listview oder Listbox nach Daten auslesen kann, um mehr Übersicht zu haben.

Priorität: 3

Grobaufwand: 45min

US-10: Sortierung im Listview / Listbox

Als Nutzer will ich, dass es eine Funktion gibt, mit welcher ich die Listview oder Listbox nach Daten sortieren kann, um mehr Übersicht zu haben. (Zum Beispiel Vorname, alphabetisch, asc)

Priorität: 3

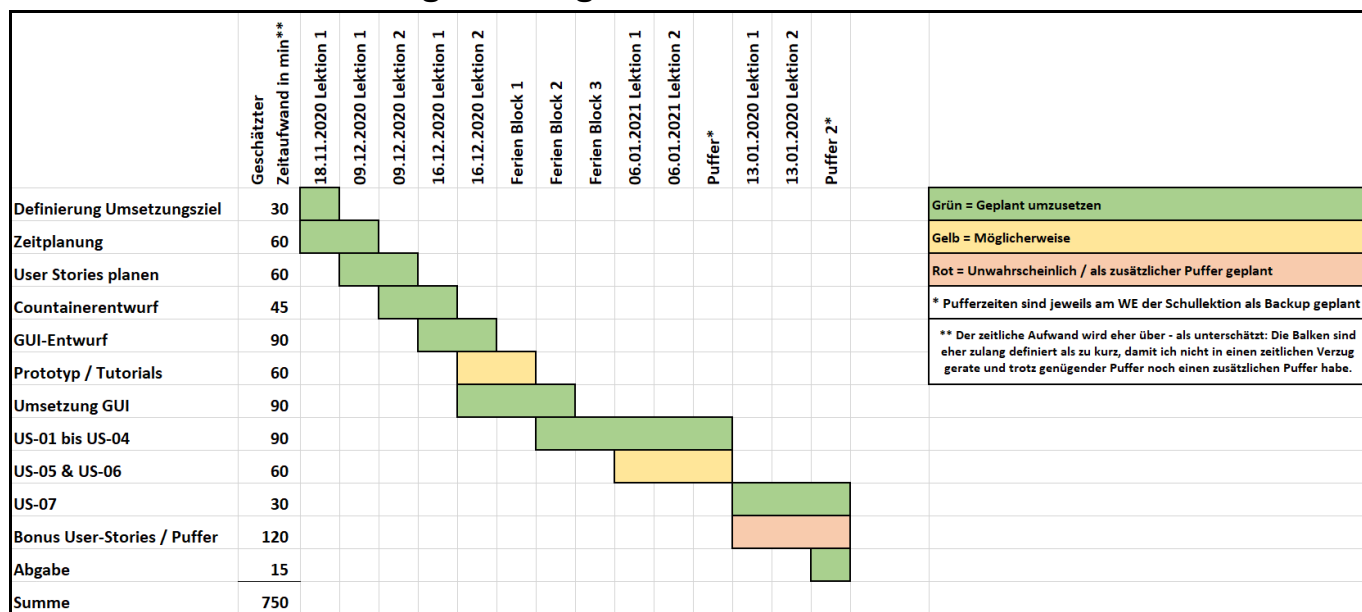
Grobaufwand: 45min

4 Arbeitspakete und Zeitplanung

4.1 Einleitung

Ich habe meine Arbeitspakete in einem Excel-Sheet definiert, zeitliche Einschätzungen aufgeschrieben und meine Zeitplanung in einem Diagramm dargestellt. Das Excel-Sheet ist auf GitHub als zusätzliche, einzelne Datei einzusehen.

4.2 Visuelle Darstellung als Diagramm

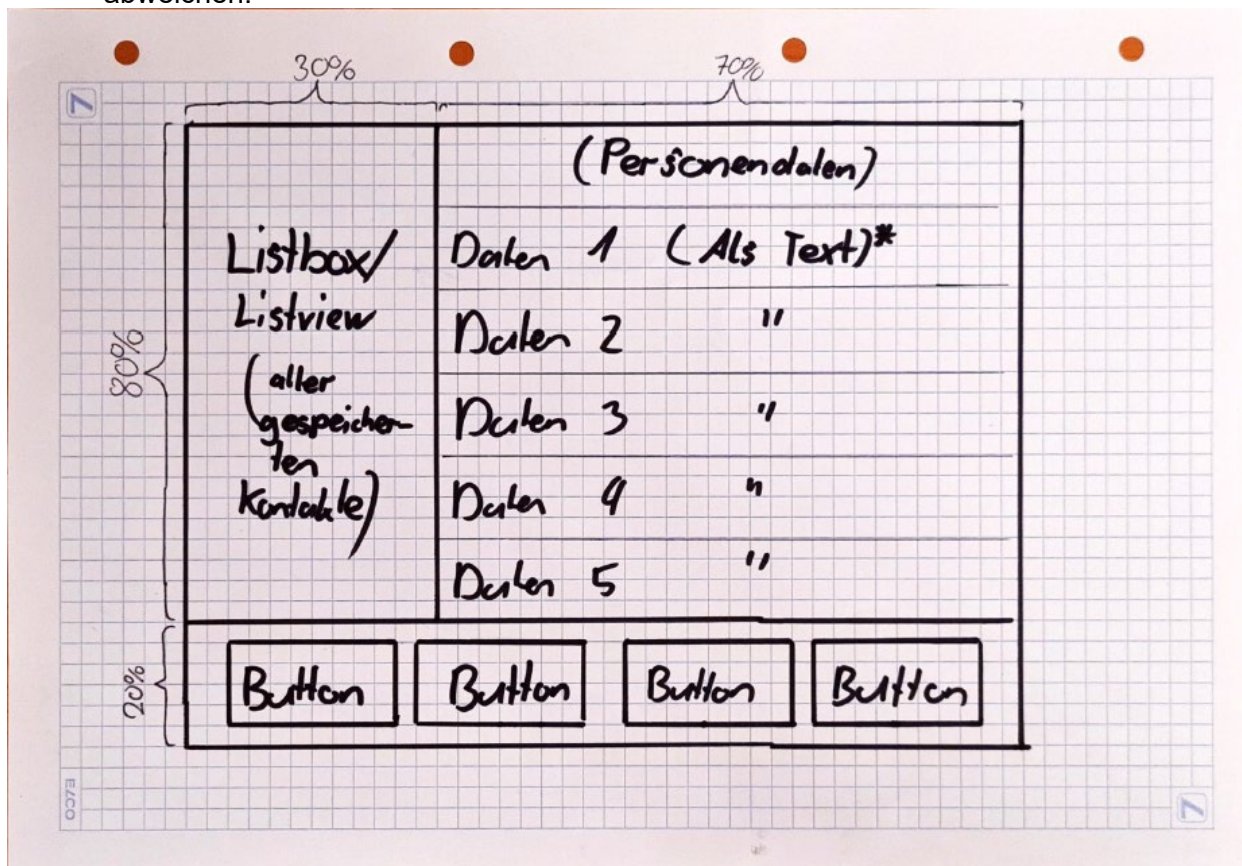


5 Containerentwurf

Nachfolgend ist der Containerentwurf zu betrachten. Dieser zeigt zur Vereinfachung ausschliesslich den Hauptbildschirm. Alle weiteren Darstellungen sind im GUI Entwurf ersichtlich. Diese sind aber, wie in den User Stories definiert, optional und ich bin mir nicht sicher, ob ich diese umsetzen werde.

Wichtige Informationen zum Containerentwurf:

- Der gesamte Containerentwurf ist als normales Grid geplant.
- Ob es eine Listbox oder Listview wird, werde ich bei der Umsetzung des GUIs herausfinden. Ich lasse mir beide Möglichkeiten offen.
- Die Personendaten werde ich entweder mit Grids umsetzen, in welchen dann Texte gespeichert sind oder tabellarisch darstellen. Falls ich bei der Umsetzung eine noch bessere Idee bekomme, werde ich diese umsetzen.
- Die zwei «Häuschen» Abstand zwischen den mittleren Buttons ist ausschliesslich deswegen gemacht worden, damit die Skizze besser aussieht. In der finalen Applikation werden alle vier Buttons den gleichen Abstand zueinander haben.
- Bei dem Containerentwurf handelt es sich um eine provisorische Umsetzung. Das Endprodukt kann abweichen.



6 GUI Entwurf

Der GUI Entwurf wurde in Adobe Xd umgesetzt. Dabei möchte ich mich bei Nadine Inglin bedanken, welche mir den Einstieg in das Programm erleichterte und bei Problemen geholfen hat. Ausschliesslich User Stories der Priorität 1 & 2 wurden umgesetzt. User Stories der Priorität 3 wurden zur Vereinfachung und zur Zeiterparnis aussenvorgelassen und werden, wenn, dann im Programmiereteil des Projektes umgesetzt. Bei dem GUI Entwurf handelt es sich um eine provisorische Umsetzung. Das Endprodukt kann abweichen. Das File ist interaktiv und ersetzt somit das Bedienkonzept.

Das Xd-File ist im GitHub Repo unter folgendem Link aufzufinden:

<https://github.com/richardleist/projekt120>

Sollte es Probleme beim Auffinden / Aufrufen geben bitte per MS Teams oder per Mail melden. An dem Upload-Zeitstempel wird erkennbar sein, dass nach Abgabe keine zusätzlichen Veränderungen gemacht wurden.

7 GUI-Umsetzung

7.1 Auffindbarkeit

Die GUI-Umsetzung ist auf dem bereits zuvor erwähnten GitHub Repo zu finden.
Die ursprüngliche Planung wurde durch ein Asp.net Projekt ersetzt.

7.2 Zusätzlicher Kommentar

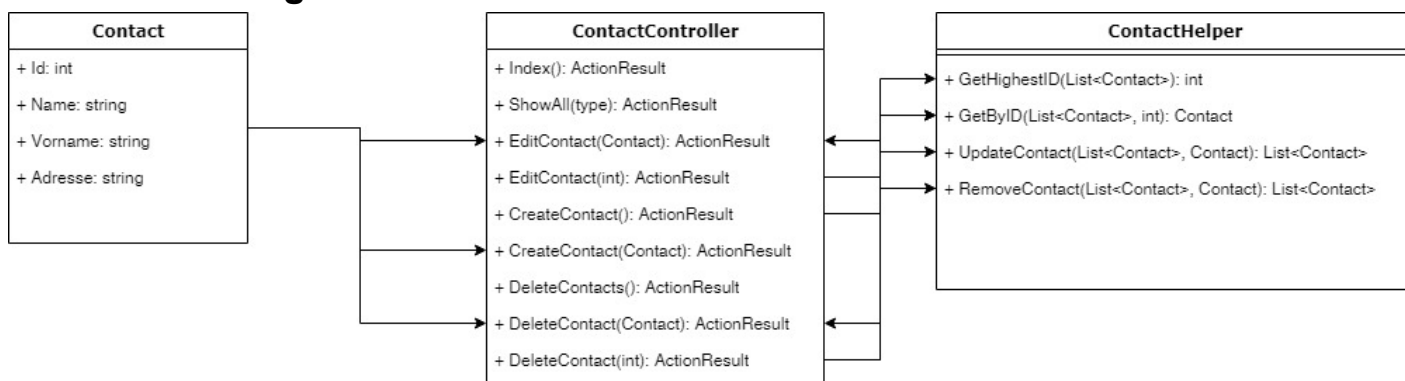
Da ich heute, Samstag, der 09.01.2021 mich dazu entschieden habe, bereits die Logik zu implementieren, gebe ich bereits heute das GUI inklusive Logik ab. Der Entscheid fiel vor allem deshalb, da wir nächsten Freitag sehr viele Projektabgaben haben werden und ich mich grade schon sehr gut auf das MVC-Projekt konzentriert habe und bereits im «Workflow» war.

8 MVC Dokumentation und Klassendiagramm

8.1 Textuelle Beschreibung der MVC Umsetzung

Die MVC Umsetzung in meinem Projekt wird beim Erstellen, Editieren und Löschen eines Kontaktes korrekt ausgeführt. Beim Erstellen übergibt der Controller der View eine neue, erstellte Instanz des Models, welche danach vom Benutzer ausgefüllt wird. Danach wird diese bei Bestätigung an den Controller übergeben, welcher den betreffenden Kontakt abspeichert. Beim Editieren eines Kontaktes wird zunächst die ID des ausgewählten Kontaktes an den Controller übergeben, welcher den dazugehörigen Kontakt an die View weitergibt. Bei welcher der Kontakt vom Benutzer auf der Weboberfläche modifiziert werden kann. Beim Bestätigen der Aktualisierung sendet die View den abgeänderten Kontakt an den Controller, welcher den alten Stand des ausgewählten Kontaktes löscht und die abgeänderten Kontaktinformationen neu hinzufügt. Das Löschen eines Kontaktes funktioniert so, dass von dem ausgewählten Kontakt die ID an den Controller übergeben wird, welcher die vollständigen Kontaktinformationen an die View weitergibt, welche diese anzeigt und im Falle einer Bestätigung die Kontaktdaten an den Controller zurücksendet, welcher diese im Anschluss dieser Prozedur löscht. Das Kontaktbuch ist ein View Element, welches alle abgespeicherten Daten aus der Nutzersession bezieht. Dies geschieht per Anfrage an den Controller, welcher der View die Daten der Session des Nutzers liefert. Das endgültige Löschen des kompletten Nutzer-Kontaktbuches funktioniert, in dem eine Anfrage seitens Nutzer per Oberfläche (also View) an den Controller gesendet wird. Daraufhin muss der anwendende Benutzer die Löschung des kompletten Kontaktbuches bestätigen. Im Falle einer bestätigen Löschungen, wird eine Anfrage an den Controller gesendet, welcher die Nutzersession bereinigt und somit alle eingetragenen Kontaktdaten löscht.

8.2 Klassendiagramm



Die View konnte nicht auf dem Klassendiagramm dargestellt werden, da es sich um eine Razor Page (cshtml) handelt und diese keinen Konstruktor, Felder oder Methoden besitzt. Wurde hierbei etwas falsch verstanden, bin ich sehr gerne bereit dies nachträglich zu korrigieren.

8.3 Quellen und Erkenntnisse

Die Angabe zu den Quellen und zu den Erkenntnissen befindet sich im Kapitel 9.2 (Quellen) und Kapitel 10 Reflexion (Erkenntnisse).

9 Umsetzung der Logik

9.1 Auffindbarkeit

Das fertige Projekt ist im bereits zuvor beschriebenen GitHub Repo zu finden und auszuführen.

9.2 Angaben zur Entwicklung

9.2.1 Tools

Genutzte Technologien / Bibliotheken

jQuery, Bootstrap, .NET Framework 4.7.2, Asp.net

Entwicklungsumgebung

Visual Studio 2019

UML-Klassendiagramm

Drawio

9.2.2 Quellen

<https://www.youtube.com/watch?v=wTLC9OFxD6k>

[https://de.wikipedia.org/wiki/ASP.NET MVC](https://de.wikipedia.org/wiki/ASP.NET_MVC)

<https://docs.microsoft.com/de-de/aspnet/mvc/overview/getting-started/introduction/getting-started>

<https://docs.microsoft.com/de-de/aspnet/mvc/overview/getting-started/introduction/adding-a-controller>

<https://docs.microsoft.com/de-de/aspnet/mvc/overview/getting-started/introduction/adding-a-model>

<https://docs.microsoft.com/de-de/aspnet/mvc/overview/getting-started/introduction/adding-a-view>

Im Modul 120 zur Verfügung gestellte Unterlagen.

10 Reflexion / Erkenntnisse

Durch das Projekt konnte ich Asp.net kennenlernen und die Regeln des MV*, in meinem Fall MVC erlernen und in der Praxis umsetzen und implementieren. Mir persönlich fiel allen voran der Einstieg schwer, konnte mich dann aber durch das verlinkte YouTube Tutorial und die Dokumentationen seitens Microsofts doch in das Projekt einfinden und in einem, meiner Meinung nach, gutem Zustand abgeben. Mein Wechsel war recht spontan und plötzlich, als ich mit einem Kollegen über MVC und MVVM sprach und mir dieser geraten hatte, MVC zu nutzen, da dieses simpler und einfacher wäre. Zusätzlich könne ich mit Asp.net etwas neues entdecken, was ich zuvor nicht kannte. Dies bewahrheitete sich dann auch und alles in allem bin ich sehr zufrieden mit dem Gelernten.

Ich habe definitiv erkannt, dass die optische Umsetzung eines Programms, also das GUI, mehr Aufwand ist, als man als Softwarenutzer glaubt.

Meine Zeitplanung, welche ich detailliert aufgestellt hatte, erwies sich als unnötig, da ich mich mal wieder nicht wirklich an diese gehalten habe. Inzwischen habe ich einfach das Gefühl, dass eine Zeitplanung bei mir fast nie etwas bringt, da es mir dann erst recht besonders schwerfällt, etwas fertig zu stellen.

Neben dem konnte ich bereits durch die zurückbekommene Bewertung des ersten Teils anhand meiner, seitens Roland Buchers korrigierten, Fehler mein Wissen zum Schreiben von User-Stories vertiefen.

Alles in allem bin ich dennoch zufrieden mit dem Projekt, vor allem da ich es eine Woche vor der eigentlichen Deadline abgebe und somit meinen Fehler der im 4. Semester leicht verspäteten Abgabe, nicht wiederholt habe und daraus definitiv etwas gelernt habe. Zusätzlich dazu bin ich mit dem Endprodukt des Projektes zufrieden.