

Fachwerk-Management-System

Inhaltsverzeichnis:

- **Seite 1:**
 - Projektbeschreibung
 - Projektmotivation
- **Seite 2:**
 - Projektverlauf
- **Seite 3 – 6:**
 - Projektfortschritt
- **Seite 7 – 9:**
 - Seite 7: Nikolas
 - Seite 8: Leon
 - Seite 9: Yassin
 - Seite 10: Luis
 - Seite 11: Alex

Projektbeschreibung

Das Projekt mit dem Namen Fachwerk-Management-System (kurz FMS) ist eine Desktop-Applikation für eine Varietät von Betriebssystemen.

Das Programm soll Bibliothekaren/innen und Lehrkräften eine einfache und unkomplizierte UI bereitstellen, die das Anlegen und Ausleihen von Fachwerken / Büchern unterstützt und die Zusammenhänge von Buch und Ausleihendem in einer Datenbank speichert.

Es soll an jeglichen Schulen als Standard zur Verfügung stehen und als Open Source Projekt jederzeit für jede Person einsehbar und nutzbar sein.

Allgemeine Projektinformationen:

Derzeitige Version des Programms: 1.1.2

Erscheinungsdatum: 08.06.2022

Letztes Update: 08.06.2022

Derzeitiger Stand des Projektes:

Verwaltung von Büchern, Nutzern und deren Zuordnung.

MySQL-Server Support

Folgt in Zukunft:

Nutzersystem über Datenbank

Projektmotivation

Ursprünglich war das Projekt nur zur Nutzung und Präsentation im Schulunterricht vorgesehen, weshalb die Motivation für dieses Projekt sich zunächst in Grenzen hielt, da man anfangs nicht von einer zukünftigen Nutzung ausging.

Allerdings änderte sich das etwa zur Hälfte des Projektfortschritts hin, als die View gerade fertig mit dem Main-Window war. Das Design und die Dinge, die bereits so gut funktionierten, sorgten dafür, dass die Motivation im Team stark anstieg, da man nun davon ausging, man könne das Programm ja doch öffentlich verwenden oder wenigstens einen kleinen Nutzen daraus ziehen.

Je höher der Projektfortschritt, desto höher stieg die Motivation, durch endlose Entdeckung von Möglichkeiten und zukünftigen Nutzungsmöglichkeiten.

Deshalb beschlossen manche von uns sogar das Programm, auch nach der Präsentation und Benotung, noch weiter zu entwickeln, um es tatsächlich betriebsfähig in Bibliotheken und allen schulischen Einrichtungen einsetzen zu können.

Projektverlauf

Zeitplan:

Das Problem im Zeitplan bestand darin, dass so vieles des Codes auf der View aufbaute und man ab einem gewissen Punkt auf Teile des Programms warten musste, um es weiter zu entwickeln oder zu testen.

Dazu entwickelten wir die Strategie des „On-Time Programming“; Wir haben uns jede Stunde darauf geeinigt, was von den Tasks heute fertig werden muss, wohl beachtend die Priorität und Dauer der Aufgaben. Waren die Tasks von Controller und Model fertig, so begannen diese Gruppen eine Konsolenversion des Programms weiter zu programmieren. So konnten wir unsere Programme im Kern testen und weiterentwickeln. War die View fertig mit dem Auftrag, wurde der Prozess wiederholt.

Änderungen:

Die größten Änderungen wurden am Callback-System verübt. Am Ende einigte sich der Controller mit der View auf eine sehr effiziente und schnelle Methode „Passing Callbacks“:

Dazu wird nur EINE Methode im Controller benötigt, die Callbacks verwertet. Diese nannten wir `handleCallback(callbackType: Callback, *values)`. Sie nimmt als erstes Argument ein Enum vom Type `Callback`. Über dieses Argument kann der Controller ein Switch-Case Szenario erschaffen, indem er simpel den Typ des Callbacks überprüft und dann die dazu definierte Funktion ausführt. Dadurch benötigen wir nicht mehr für jedes Fenster andere Callback-Methoden, denn wir können einfach den Callback überprüfen und wissen dann, was der values-Tuple enthalten muss.

Probleme:

Tkinter hatte einige sehr merkwürdige Bugs, die so komplex waren, dass Leon am Ende in den Source Code von Tkinter gegangen ist und ein paar Sachen so gefixt hat.

Interessante Probleme / Bugs:

Bug #2: Programm beendet sich nach Schließen des Fensters nicht.

Event für das Main-Fenster: Wenn es sich schließt, wird das Programm mit `sys.exit(0)` beendet.

Bug #4: (Tkinter) Animation after Frame Callback:

Oft werden Aktionen nicht richtig ausgeführt, da das Fenster geschlossen wird, bevor die Daten daraus ausgelesen werden können. Lösung: Methode `after_call()` kann einen Callback ausführen und speichert dabei den derzeitigen Inhalt der Input Felder.

Projektfortschritt

Iteration 2:

CimeyClust / FMS > Projects > Priorität 10 | Übersicht Bestand

2 To do	4 In Progress	6 Done
<div>View M Button/Checkbox für Selektion erstellen Added by Yapollon</div>	<div>Model S Methoden zum Erhalten aller IDs als Liste für jede Entität. Added by Veron001</div>	<div>Model L Erstellen der Datenbank aus Python heraus. Added by Yapollon</div>
<div>View XL Eine Übersicht über jedes Exemplar. Dort sollen die Daten, sowie der derzeitige Ausleihstatus des Exemplars angezeigt werden. Buttons: QR-Code drucken Buttons: ausleihen / abgeben Beim Klicken von "ausleihen" auf die Ausleihsseite weiterleiten. Added by Yapollon</div>	<div>View S Erstellen des Layouts. Added by Yapollon</div>	<div>Model S Anlegen von Testdatensätzen Added by Yapollon</div>
	<div>Model L Erstellen der notwendigen Datenbankabfragen. – Auslesen aller Exemplare – Auslesen nur entliehenen Exemplare – Auslesen nur entleihbare Exemplare Added by Yapollon</div>	<div>Controller SS Initialisieren von Model & View. Programmstart. Added by Yapollon</div>
	<div>View L Visualisierung einer Tabelle aus allen Büchern. Die zurückgegebene Date ist ein Dictionary bestehend aus den buch-ids als Keys und einem weiteren Dictionary als Value, welches die Informationen des Buches enthält. Ein Beispiel: <pre>{ 1: { "notice": str-object "title": str-object "isbn": str-object "author": str-object "subject": str-object "usedBy": str-object # The student first name and last name, that uses the book currently } }</pre> Added by Yapollon</div>	<div>Controller M Festlegen der Namenskonventionen Added by Yapollon</div>
		<div>Model S Möglichkeit die Datenbank im %appdata% Ordner des Computers zu erstellen, um den Zugriff zu erschweren. Added by Yapollon</div>
		<div>Controller L Erstellen der notwendigen Callback-Funktionen. – Auslesen/Bereitstellen aller Exemplare – Auslesen/Bereitstellen nur entliehenen Exemplare – Auslesen/Bereitstellen nur entleihbare Exemplare Added by Yapollon</div>

CimeyClust / FMS > Projects > Priorität 20 | Titel anlegen

3 To do	1 In progress	1 Done
<div>View L Formularmaske zum Anlegen von Titeln in die View integrieren Added by Yapollon</div>	<div>Model M Methoden zum einfügen von Titeln Added by Yapollon</div>	<div>Controller M Callbacks für das anlegen von Titeln erstellen Added by Yapollon</div>
<div>View S Layout für die Formularmaske zum Anlegen von Titeln Added by Yapollon</div>		
<div>View S Entsprechenden Knopf zum Öffnen der Formularmaske ins Hauptfenster integrieren Added by Yapollon</div>		

Iteration 3:

CimeyClust / FMS > Projects > Priorität 10 | Übersicht Bestand

0 To do	2 In Progress	10 Done
	<div>Model L ... Erstellen der notwendigen Datenbankabfragen. – Auslesen aller Exemplare – Auslesen nur entliehenen Exemplare – Auslesen nur entlehbare Exemplare Added by Yapollon</div> <div>View XL ... Eine Übersicht über jedes Exemplar. Dort sollen die Daten, sowie der derzeitige Ausleihstatus des Exemplars angezeigt werden. Button: QR-Code drucken Button: ausleihen / abgeben Beim Klicken von "ausleihen" auf die Ausleihsseite weiterleiten. Added by Yapollon</div>	<div>Model L ... Erstellen der Datenbank aus Python heraus. Added by Yapollon</div> <div>View M ... Button/Checkbox für Selektion erstellen Added by Yapollon</div> <div>Model S ... Methoden zum Erhalten aller IDs als Liste für jede Entität. Added by Verax001</div> <div>Model S ... Anlegen von Testdatensätzen Added by Yapollon</div> <div>Controller SS ... Initialisieren von Model & View. Programmstart. Added by Yapollon</div> <div>Controller M ... Festlegen der Namenskonventionen Added by Yapollon</div> <div>Model S ... Möglichkeit die Datenbank im %appdata% Ordner des Computers zu erstellen, um den Zugriff zu erschweren. Added by Yapollon</div> <div>Controller L ... Erstellen der notwendigen Callback-Funktionen. – Auslesen/Bereitstellen aller Exemplare – Auslesen/Bereitstellen nur entliehenen Exemplare – Auslesen/Bereitstellen nur entlehbare</div>

CimeyClust / FMS > Projects > Priorität 20 | Titel anlegen

0 To do	0 In progress	5 Done
		<div>Controller M ... Callbacks für das anlegen von Titeln erstellen Added by Yapollon</div> <div>Model M ... Methoden zum einfügen von Titeln Added by Yapollon</div> <div>View L ... Formularmaske zum Anlegen von Titeln in die View integrieren Added by Yapollon</div> <div>View S ... Layout für die Formularmaske zum Anlegen von Titeln Added by Yapollon</div> <div>View S ... Entsprechenden Knopf zum Öffnen der Formularmaske ins Hauptfenster integrieren Added by Yapollon</div>

CimeyClust / FMS > Projects > Priorität 30 | Exemplar anlegen

0 To do

1 In progress

View | S |
Entsprechenden Knopf zum Öffnen der Formularmaske ins Hauptfenster integrieren
Added by Yapollon

3 Done

Model | M |
Methoden zum einfügen von Exemplaren
Added by Yapollon

Controller | M |
Callbacks für das anlegen von Exemplaren erstellen
Added by Yapollon

View | L |
Formularmaske zum Anlegen von Exemplaren in die View integrieren
Added by Yapollon

CimeyClust / FMS > Projects > Priorität 40 | Exemplar entleihen

1 To Do

View | S |
Implementierung der Form-View als eigene Window-Instanz.
Added by Verox001

2 In Progress

Model | S |
Methode zum Einfügen eines Datensatzes in Tabelle Ausleihe.
Added by Verox001

Controller | S |
Methode zum Erstellen des Datenbankeintrags aufrufen und mit der ID eine neue Exemplarinstanz (Book) kreieren. (Callbacks verwerten.)
Added by Verox001

1 Done

View | M |
Layout der Formularmaske entwerfen. Felder:
1. Buch-ID
2. Schülername
3. Klasse
Added by Verox001

Iteration 4:

CimeyClust / FMS > Projects > Priorität 50 | Exemplar zurückgeben

0 To Do

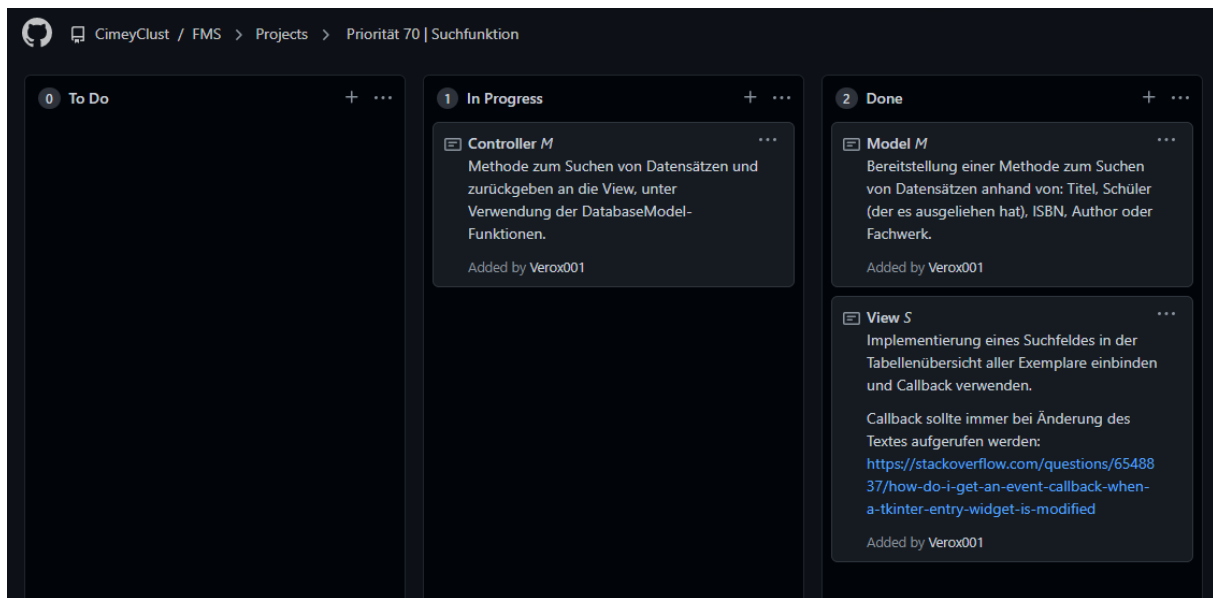
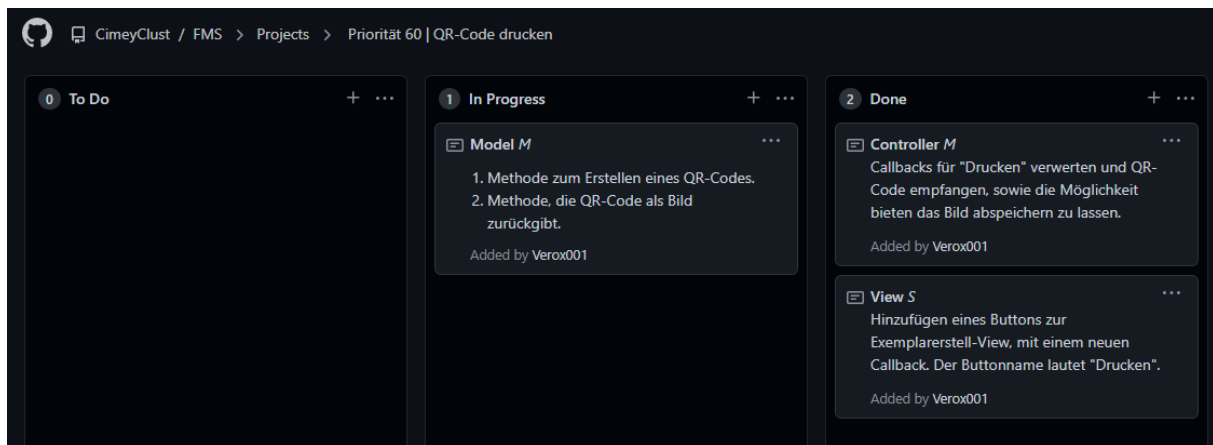
1 In Progress

View S
Implementierung des gleichen Fensters, wie in "Exemplar ausleihen", nur ohne die Felder "Schülername" und "Klasse".
Added by Verox001

2 Done

Model S
Methode zum Entfernen des Eintrags in der Ausleihetabelle.
Added by Verox001

Controller S
isBorrowed der Instanz des betreffenden Exemplars (Book) false setzen.
Added by Verox001



Iteration 5:

Der Rest der verbleibenden Tasks wurde in Iteration 5 und 6 (Pfingsten) beendet.

Während Pfingsten wurden auch die Major Bugs gefixt.

Userstories, die komplett abgeschlossen waren, wurden nicht mehr als Screenshots eingefügt, um unnötige Platzverschwendung zu verhindern.

Persönliche Einschätzung (Nikolas)

Das Projekt hat mir sehr gefallen, da meine mit-Programmierer sehr gut mitgearbeitet haben, Und ich viel über das generelle Programmieren mit mehreren Personen gelernt habe.

Wie lief die Arbeit innerhalb der Gruppe?

Die Zusammenarbeit während des Projektes fand ich sehr gut. Anfangs lief die Zusammenarbeit schleppend jedoch änderte sich dies mit der ersten Iteration, welche schon teilweise funktionierte. Mit dieser steigerte sich die abschlussrate der einzelnen To Do's enorm und wir kamen sehr schnell voran. Während der Schulstunden trugen wir die Ergebnisse der Arbeit von zuhause zusammen und integrierten diese in das Hauptprogramm. Wenn es zu einem Problem kam, wendeten sich nach meiner Erkenntnis Alle auch Ich an Leon womit ich ihn nochmal eindeutig für den Einsatz während des gesamten Projektes loben möchte. Auch meine Zusammenarbeit mit Alex innerhalb der View war sehr gut, Er half mir dabei im Internet eine gute Lösung für unser Tabellenproblem in Tkinter zu finden und erstellte auch 40% des Adobe XD Prototypen.

Was habe ich gelernt?

Ich habe während der Gruppenarbeit einiges über Tkinter und seine Nachteile gelernt, so besaß dieses viele Bugs und Unregelmäßigkeiten, die man umgehen oder Auswendiglernen musste. Ich lernte auch das Plugin CustomTkinter kennen welches Ich durch Zufall während online Recherchen im Internet fand und darauf hin zur Grundlage unseres UIs wurde. Mittels CustomTkinter hatten wir dann die Möglichkeit schöne funktionale UIs für unser Datenbank Management zu erstellen. Außerdem konnte ich lernen wie man gemeinsam innerhalb einer Gruppe mit mehreren Leuten gleichzeitig programmiert und diesen Code dann in GitHub austauscht

Reflektion

Ich persönlich fand das Projekt sehr gelungen und bin auch mit den Ergebnissen meiner View und auch dem Restlichen Team sehr zufrieden, da dieses Projekt derzeit mein definitiv komplexestes war und die View auch definitiv meiner derzeit beste.

Persönliche Einschätzung und Reflektion (Leon)

Das Projekt hat mir sehr viel Spaß gemacht, denn es war sehr angenehm mit anderen Leuten zu planen, sich darüber auszutauschen und nehme schon vorweg, dass ich dieses Projekt auch noch nach der Präsentation und der Benotung vorhabe weiterzuentwickeln und öffentlich für Schulen und Bibliotheken bereitzustellen.

Wie lief die Gruppenarbeit?

Die Gruppenarbeit lief so, wie ich mir sie tatsächlich von Anfang vorgestellt habe: Es gab Personen, die sehr stark und interessiert am Projekt gearbeitet haben, aber einige die sich eher mitschleifen ließen. Ich konnte Sinan sehr für das Projekt begeistern und er wirkte beim Pair-Programming sehr interessiert, trotz dessen, dass das Programmieren als Coder ihm einige Probleme bereitete. Mit Jan habe ich nicht sehr viel zusammen programmiert, da oft immer einer von uns herum gehen musste, um Fragen zu klären und generell zu helfen. Von dem Model habe ich aus meiner Sicht nur Erfolge gesehen; Yassin und Luis kamen gut voran, weswegen auch wir bald alle Callbacks und Verknüpfungen auf Seite des Models implementieren konnten. Am Ende mussten wir dann nur noch auf die View warten und Bugs fixen.

Als die View fertig war verbanden wir noch schnell die Callbacks und fertig war die erste BETA-Version!

Was konnte ich lernen?

Ich konnte tatsächlich nicht so viel neues dazu lernen, aber es ist immer wieder lustig seine Informatikpraxiskenntnisse in Python umzusetzen und zu sehen, wie intuitiv Python deinen Code umsetzt.

Eine wirkliche Errungenschaft in diesem Projekt war, dass ich sonst immer Applikationen mit Frameworks und vielen APIs erstellt habe, was sich in diesem Projekt unterschied:

Ich musste viel selber programmieren und konnte mich nicht auf ein Framework verlassen, was sonst so viel für mich übernimmt. Dazu musste ich auch immer wieder in tiefere Code Ebenen abtauchen und mir oft exotische Lösungen ausdenken.

Ein anderer sehr wichtiger Punkt war die Gruppenarbeit:

Ich erstellte Klassenschemata, Vorgaben für andere und besprach das Projekt sehr sachlich mit den anderen, was nicht nur meine soziale Kommunikation förderte, sondern mich auch lehrte effektiv im Team zu arbeiten. Ich hatte zwar schon mehrere größere Projekte mit Teams, aber bei diesem konnte ich mit Abstand am Meisten für eine effektive Projektarbeit sorgen, da ich im Controller saß und mir immer einen etwas besseren Überblick verschaffen konnte, anstatt nur als Backend-Developer stupide getter- und setter-Methoden zu schreiben.

Reflektion

Ich finde, dass das Projekt sehr sauber ablief und ich nur einige Schwierigkeiten hatte. Was mich etwas traurig stimmt, ist das vielen Leuten nicht bewusst ist, dass man Informatik auch hobbymäßig wie einen Sport oder ein Werkzeug, hobbymäßig und zu Hause ausüben muss, um wirklich Fortschritte vorzuzeigen.

Persönliche Einschätzung (Yassin)

Wie lief die Gruppenarbeit?

Die Gruppenarbeit lief trotz erster kleiner Probleme am Ende recht gut. Wie ich fand, konnte jeder sich gut auf seinen Teil der Arbeit konzentrieren. In der Model Gruppe mit Luis war unsere Arbeit recht unkompliziert und konnten uns deswegen gut austauschen über die Methoden, die wir schrieben. Meinerseits habe ich mich mehr auf die Struktur der Klasse und der Datenbankerstellung gekümmert und Luis sich auf die einzelnen Abfragen, die die Datenbank brauchte. Häufig kam es vor, dass wir uns mit der Controller Gruppe austauschten, meistens benötigten sie neue Abfragen oder etwas musste an ihnen geändert werden. Dies konnte immer problemlos geregelt werden.

Was habe ich gelernt?

Allgemein konnte ich einiges über die Programmierung in der Gruppe lernen, darüber wie man sich gegenseitig austauscht und Aufgaben aufteilt. Was ich aber vor allem durch die Arbeit als Model lernen konnte war der Umgang mit der Sqlite3 Bibliothek und später auch mit der MySQL Bibliothek. Dabei konnte ich vieles über Sqlite's Methoden und Verhalten bei der Datenbankerstellung sowie Abfragen herausfinden oder auch wie man MySQL benutzt. Zudem habe ich auch erstmals von Context Managern erfahren womit es möglich ist eine Klasse mit dem "with ... as" Statement zu benutzen. Was sich als sehr praktisch herausgestellt hat. Ich kann mir gut vorstellen diese Methode in zukünftigen Projekten auch zu nutzen. Zuletzt kann ich auch sagen das ich im Verlauf des Projekts dadurch das wir Github benutzt haben ziemlich versiert darin geworden bin.

Reflektion

Zurückblickend kann ich sagen, dass das Projekt ganz gut auslief und wir ein tolles Programm auch am Ende kreieren konnten. Mit dem Model von Luis und mir war ich am Ende auch recht stolz.

Persönliche Einschätzung (Luis)

Wie lief die Gruppenarbeit?

Die Gruppenarbeit lief überragend. Im Model war die Kommunikation dauerhaft gegeben, wenn Fragen zum weiteren Verlauf oder zum jeweils anderen Code geäußert wurden, waren sie innerhalb weniger Minuten auch online beantwortet. Auch mit dem Controller liefen die Absprachen trotz anfänglicher Schwierigkeiten vom Verlauf nach der ersten Woche hervorragend. Meinerseits kann ich mich deshalb nicht beschweren und bin über die Gruppenarbeit sehr stolz.

Reflektion

Zurückblickend kann ich sagen, dass das Projekt ganz gut auslief und wir ein tolles Programm auch am Ende kreieren konnten. Mit dem Model von Yassin und mir war ich am Ende auch recht zufrieden.

Persönliche Einschätzung (Alex)

Das Projekt hat mir sehr gut gefallen, da die Planung, als auch die Zusammenarbeit von allen wahrlich gut verlief. Es war auch ein cooles Projekt wo jeder etwas dazu beigetragen hat und die Gemeinsame Programmierung war insgesamt angenehm.

Wie lief die Gruppenarbeit?