Max Projekttagebuch

Datum	Aktivität
28.04.2025	Projektstart, Repository angelegt, erste Dateien erstellt
05.05.2025	Klassendiagramm und Designplanung für Finance und Timestamp erstellt
08.05.2025	Das WPF Projekt erstellt (Soloution)
09.05.2025	Erster Timestamp Entwuf programmiert
26.05.2025	Checkout-Funktion erstellt und generelle Funktionen des Timestamp
28.05.2025	Timestamp funktioniert komplett und Das editWindow wurde hinzugefügt mit Funktionalität und Merge Konflikt gelöst
03.06.2025	Merge branch 'main', Timestamp funktioniert, Editing in ObservableCollection umgesetzt
04.06.2025	Änderungen an Timestamp
06.06.2025	Costs Livechart funktioniert, YAML für Timestamp erstellt, Leere Endpoints angelegt, gestageter Commit YAML und Controller
07.06.2025	Diagramm-Änderungen aufgrund von Fehlern
09.06.2025	Merge branch 'main', Timestamp über API mit Datenbank verknüpft, Design für Timestamp Window verbessert, Merge durchgeführt
11.06.2025	TimeStamp UI überarbeitet, Costs UI erstellt, Funktionalität Rechnung (suchen, öffnen, abspeichern), UI Costs erstellt + Funktionalität File öffnen und abspeichern
12.06.2025	Costs Get Funktionen implementiert (Frontend & Backend), Merge branch 'main'
13.06.2025	Farbe der Timestamps und EditTime angepasst, Merge branch 'main'
14.06.2025	Post Methode für Abrechnung erstellt (Backend und Frontend), Design angepasst Costs, Windows zu Pages geändert, Merge branch 'main'
15.06.2025	Methodenname geändert, Detail Window Costs - Design Änderung und Responsive, Responsive in der Cost Page verbessert, Merge branch 'main', Backend Abrechnungsfehler ausgebessert im Endpunkt pid uid, Costs_Detail Window erstellt und PDF eingebunden für LiveAnsicht, Merge branch 'main', Timestamp ist jetzt unabhängig vom Projekt
17.06.2025	Strukturiert Costs und Timestamp, Logging im Frontend und Dokumentiert
18.06.2025	UniTests für Backend und Frontend erstellt, Logging im Backend, Doxygen Dokumentation erstellt, Pflichtenheft, Readme, Lastenheft, Betriebsanleitung

Valentin Projekttagebuch

Datum	Beschreibung
25-04-2025	Heute haben wir mit dem Projekt angefangen. Nachdem uns erklärt wurde, was zu machen ist, haben wir direkt gestartet mit einem Kickoff-Meeting für die Projektidee.
28-04-2025	Heute haben wir uns Gedanken zur Projektidee gemacht und die alte verworfen und eine neue gesucht. Danach haben wir uns Gedanken über die Systemarchitektur gemacht.
29-04-2025	Heute haben wir ein neues Git-Repository angelegt. Danach haben wir dazu ein Projekt hinzugefügt, um mithilfe von einem Kanban (Ticketsystem) unsere Aufgaben im Projekt zu managen. Als Letztes haben wir dann noch die gewünschte Dateistruktur erstellt.
02-05-2025	Heute haben wir uns intensiv Gedanken zur Systemarchitektur gemacht.
05-05-2025	Heute habe ich angefangen mit dem Zeichnen von Klassendiagrammen.
06-05-2025	Heute habe ich es geschafft, fertig zu werden mit dem Klassendiagramm-Design.
09-05-2025	Heute habe ich ein ERM und ein ER-Diagramm designt.
19-05-2025	Heute habe ich für das Projekt ein grobes Sequenzdiagramm erstellt.
20-05-2025	Heute habe ich die aktuell benötigten Aufgaben in Tickets im Projekt-Kanban erstellt und mir überlegt, was ich wann mache.
23-05-2025	Heute habe ich mich erkundigt, welche die ideale Datenbank für unser Projekt ist. Ich hatte die Wahl zwischen Supabase und Neon. Da ich Supabase schon kannte und einfach etwas Neues ausprobieren wollte, habe ich mich für Neon entschieden. Ich habe mir dann anschließend einen Account auf Neon erstellt und eine neue Datenbank mit einer Testtabelle angelegt und einen Dateneintrag eingefügt. Dann habe ich probiert, eine einfache Verbindung zwischen Python (mithilfe von psycopg2) und der Datenbank auf Neon aufzubauen. (Nicht geklappt.)
26-05-2025	Heute habe ich es endlich geschafft, die Verbindung zwischen Python und der Datenbank aufzubauen. Problem: Die Server hatten an diesem Tag Probleme.
27-05-2025	Heute habe ich ein Python-Skript geschrieben, mit dem ich die Tabellen der Datenbank erstellen kann und diese gleich mit ein paar Testdaten befülle.
02-06-2025	Heute habe ich die Ordnerstruktur erstellt und wichtige Ressourcen hinzugefügt, die ich momentan benötige. Danach habe ich die Anmelden- und die Registrieren-Seiten erstellt und den XAML-Code dafür geschrieben. Ich habe es geschafft, mit der Anmelden-Seite fertig zu werden.

Datum	Beschreibung
03-06-2025	Heute habe ich die Registrieren-Seite fertig gemacht.
04-06-2025	Heute habe ich mir von ChatGPT eine Beispiel-YAML-Datei generieren lassen für unser Projekt. Das Ergebnis war nicht sehr gut. Ich habe sie dann leicht angepasst und mithilfe einer Python-Datei, die ich noch von der Kaffeemaschine-Aufgabe aus DBI hatte, in WSL zu einem Swagger-Projekt gemacht. Danach habe ich ein paar Fehler im Code-Behind in der Registrieren-Seite gelöst. Anschließend habe ich die Passwort-zurücksetzen-Seite erstellt. Danach habe ich ein ansprechendes Design für die Registrieren-Seite und die Anmelden-Seite hinzugefügt. Bezüglich des Designs muss ich sagen, dass ich mich hier sehr stark an Microsoft Teams und der OpenAI-Website inspiriert habe.
05-06-2025	Heute habe ich den ersten Swagger-Endpunkt erstellt, der überprüft, ob eine bestimmte E-Mail in der Neon-Datenbank existiert. Dann habe ich einen Endpunkt erstellt für das Anmelden, sprich es wird überprüft, ob eine bestimmte E-Mail und ein Passwort existieren. Als Letztes habe ich noch einen Endpunkt für die Registrierung erstellt. Dabei wird ein Benutzer in der Benutzertabelle angelegt mit Vorname, Nachname, Geburtsdatum, E-Mail und Passwort. Was ich zu Swagger noch sagen muss: Ich hatte zuerst Probleme mit der Datenübermittlung. Um das zu lösen, habe ich Models verwendet – das hat mir ChatGPT empfohlen – und ich muss sagen, das ist wirklich angenehm, allerdings zu Beginn ein bisschen komplex.
06-06-2025	Heute habe ich das XAML für das Hauptfenster erstellt und danach schön designed. Danach habe ich die Backend-Logik für das Hauptfenster und für die Passwort-zurücksetzen-Seite erstellt – damit meine ich die Datenvalidierung und was passiert, wenn man auf Buttons klickt.
07-06-2025	Heute habe ich die Backend-Logik für die Anmelden-Seite erstellt. Danach habe ich eine Ladeanimation gemacht für die Seiten Anmelden, Registrieren und Passwort zurücksetzen. Danach habe ich die Backend-Logik für die Registrieren-Seite gemacht. Dann habe ich ein paar Bugs gelöst – unter anderem einen der dazu geführt hat, dass die Passwort-Auge-Animation nicht funktioniert hat. Dann habe ich einen Timer für das Hauptfenster gemacht und als Letztes mein Python-Skript für das Erstellen und Befüllen der Datenbanktabellen hinzugefügt.

Datum	Beschreibung
08-06-2025	Heute habe ich die Klassen erstellt, die ich im Moment und in naher Zukunft brauche. Danach habe ich Interfaces hinzugefügt, die ich für meine Verbindung zwischen Frontend und Backend brauche. Danach habe ich Klassen hinzugefügt für Anmelden, Registrieren und Passwort zurücksetzen – die ich jetzt brauche für eine leichte und angenehme Kommunikation zwischen Frontend und Backend. Dann habe ich Dienstklassen hinzugefügt, die von den Interfaces erben – diese machen unter anderem die HTTP-Requests. Danach musste ich ein bisschen die Swagger-Endpunkte abändern, da diese nicht ganz richtig waren, sprich nicht so wie ich sie wollte (ich habe falsch geplant). Dann habe ich den C#-Code geschrieben für die Verbindung zwischen den Swagger-Endpunkten und der Passwort-zurücksetzen-, Registrieren- und Anmelden-Seite. Danach habe ich die Projekt-Seite erstellt und den XAML-Code bearbeitet. Danach habe ich ein neues UserControl erstellt und den nötigen Code dafür geschrieben, um die Projekte einzeln als UI-Elemente anzuzeigen. Danach habe ich angefangen, Daten zwischen den Seiten korrekt auszutauschen. Dann musste ich ein paar Bugfixes machen, damit die Verbindung zwischen Swagger und C# wieder korrekt funktioniert. Als Nächstes habe ich eine neue Klasse erstellt, um eine normale Farbe zu einer Brush-Farbe zu machen und zusätzlich habe ich noch Testdaten für mein Projekt hinzugefügt. Danach habe ich es geschafft, dass die Projektkarten – also diese UserControls – richtig auf der Projekt-Seite angezeigt werden. Dann habe ich einen leichten Suchalgorithmus programmiert, welcher eine Live-Suche ermöglicht. Als Nächstes habe ich eine neue Seite erstellt, um die Eigenschaften eines jeden Projekts anzuzeigen – Projekt-Detail-Seite genannt. Nachdem ich diese Seite mit der nötigen Logik und Styling versehen habe, habe ich dann noch eine Methode hinzugefügt, welche es erlaubt, dass man, wenn man auf eine Projektkarte klickt, sich die Detail-Seite öffnet.
09-06-2025	Heute musste ich zuerst einen Bug lösen, der das Problem zwischen dem Datenaustausch zwischen Projekt-Seite und Projekt-Detail-Seite behoben hat. Als Nächstes habe ich dann die Kanban-Seite erstellt mit einem dazugehörigen UserControl für die Tickets. Danach habe ich die Kanban-Seite mit sehr viel Arbeit designed. Dann hatte ich ein Problem, die einzelnen Spalten im Kanban korrekt anzuzeigen – dafür musste ich aber zuerst eine neue Klasse quasi eine Hilfsklasse hinzufügen. Als Nächstes habe ich die Logik für die Kanban-Seite hinzugefügt. Danach habe ich ein neues Fenster erstellt, um neue Tickets hinzuzufügen. Anschließend habe ich natürlich das Fenster schön designed. Danach musste ich dieses Fenster nur noch mit der Kanban-Seite verbinden.

Datum	Beschreibung
11-06-2025	Heute habe ich das UserControl für die Tickets ausprogrammiert und auch designed. Anschließend habe ich ein paar kleine Bugs gefixt. Als Nächstes stand Drag and Drop von diesen Tickets an. Danach habe ich ein neues Fenster erstellt, um neue Projekte anzulegen – dafür habe ich auch gleich die Backend-Logik geschrieben. Danach habe ich die Projekt-Seite und das neue Fenster schön designed. Danach habe ich noch ein neues Fenster hinzugefügt, um ein Ticket zu updaten. Als Letztes habe ich dieses Fenster noch schön gemacht.
12-06-2025	Heute habe ich als Erstes die Backend-Logik für das Fenster geschrieben, wo man Tickets hinzufügen kann. Danach habe ich mit einem neuen Kapitel gestartet – und zwar eine neue Seite für Ordner im Projekt. Dann habe ich eine Verbindung zwischen der Projekt-Detail-Seite und der Projekt-Ordner-Seite gemacht. Als Nächstes habe ich ein neues UserControl hinzugefügt, um einen Ordner darzustellen und die Ordner-Seite nochmals überarbeitet. Danach wurden die Projekt-Ordner-Seite und das Ordner-Custom-Control schön gestylt. Danach habe ich die Dateien-Seite sowie ein UserControl für das Darstellen einer Datei erstellt und programmiert. Danach habe ich ein paar Bugfixes gemacht und dann ein neues Fenster gemacht, um eine Datei besser darzustellen (hat nicht ganz so gut geklappt). Dann habe ich für das Custom-Control für Dateien neue Knöpfe hinzugefügt, um die jeweilige Datei zu öffnen und zu löschen. Dann habe ich noch die Datei-Seite schön gemacht.
13-06-2025	Heute habe ich das Datei-Custom-Control gestylt. Im Anschluss habe ich dann noch ein paar Bugs gefixt. Dann habe ich Testdaten für die Dateien-Seite hinzugefügt. Dann habe ich den Suchalgorithmus von der Projekt-Seite auf die Ordner-Seite und auf die Dateien-Seite hinzugefügt. Dann habe ich bei Swagger einen neuen Endpunkt hinzugefügt, durch den man neue Projekte erstellen kann. Dann habe ich eine Verbindung zwischen dem neuen Swagger-Endpunkt und der Projekt-Seite gemacht. Dann habe ich drei neue Swagger-Endpunkte hinzugefügt – einmal um Tickets zu löschen, einmal um Tickets zu ändern und einmal um Tickets zu erstellen – dann jeweils die Verbindung zwischen den neuen Endpunkten und der Kanban-Seite sowie dem Custom-Control für die Tickets.

Datum	Beschreibung
14-06-2025	Heute habe ich neue Swagger-Endpunkte hinzugefügt – einmal zum Erstellen und einmal zum Ändern von Spalten im Kanban. Dann habe ich die Verbindung zwischen dem Endpunkt in Swagger zum Erstellen einer neuen Spalte und der Kanban-Seite verbunden. Danach habe ich die Verbindung zwischen dem Endpunkt, welcher eine Spalte updated und der Kanban-Seite hergestellt. Danach habe ich die Verbindung zwischen meinem App-Teil mit Max' App-Teil verbunden (also seine Seiten eingebunden). Danach habe ich eine bessere Datenvalidierung für die Registrieren-Seite, die Anmelden-Seite und die Passwort-zurücksetzen-Seite gemacht. Danach musste ich einen Bug fixen, weil der Zurück-Button für die Projekt-Detail-Seite nicht richtig funktioniert hat. Als Letztes habe ich die Drag-and-Drop-Animation für die Tickets hinzugefügt – mithilfe eines Geisterfensters.
15-06-2025	Verbesserungen und Integration
Heute habe ich eine Ladeanimation für WPF-Seiten hinzugefügt. Zudem wurde die Kosten-Seite responsive gemacht und ein Backend-Fehler bei `pid` und `uid` behoben. Ich habe die Verbindung zwischen dem Swagger-UpdateProject-Endpunkt und der Projekt-Einstellungs-Seite hergestellt. Neue Endpunkte für das Aktualisieren und Löschen von Projekten wurden hinzugefügt und die Backend-Logik für die Projekt-Einstellungs-Seite erstellt. Auch neue Felder und Schaltflächen wurden auf der Projekt-Einstellungs-Seite eingebaut.	
16-06-2025	Anpassungen und Fehlerbehebungen

Datum	Beschreibung
Heute habe ich die Scroll-Funktion für das Ziehen von `ProjectIssueControl`-Elementen hinzugefügt. Das Datenproblem zwischen meinem Teil und dem anderen Teil wurde behoben. Das `ProjectIssueControl` hat jetzt einen 2D-Renderer, um die Leistung zu verbessern. Auch das Problem mit den Timestamps wurde gelöst, sodass er nun unabhängig vom Projekt funktioniert.	
17-06-2025	Logging und Dokumentation
Heute habe ich das Logging für alle Seiten und Services hinzugefügt. Ich habe die Doxygen-Dokumentation abgeschlossen und die Backend-Dokumentation fertiggestellt. Unit-Tests für den `ColorToBrushConverter` und den `ApiService` wurden geschrieben. Ich habe auch Tests für den `AuthService` erstellt und ein neues Testprojekt eingerichtet.	
18-06-2025	Dokumentation und Tests

Heute habe ich das Logging für alle User Controls und das `MainWindow` abgeschlossen. Die Doxygen-Dokumentation für das gesamte Projekt wurde fertiggestellt. Ich habe Unit-Tests für den `ApiService` und den `ColorToBrushConverter` geschrieben. Außerdem habe ich die `Readme`-Datei erstellt, die alle wichtigen Infos zum Projekt enthält.