

Herausgeber Prof. Dr. Arno Hitzges

Schriftreihe Bachelor-Resümee

Forschungsbereich Low-Code-Technologie

Entwicklung einer mobilen Zeiterfassungsapplikation mittels Microsoft Power Apps

Anwendungsentwicklung zur Unterstützung einer Hochschulvorlesung

Apostolos Chatzopoulos

Studieren. Wissen. Machen.

Impressum

Hochschule der Medien

Nobelstrasse 10

70569 Stuttgart

www.hdm-stuttgart.de

0711 8923-0

Autor

Apostolos Chatzopoulos

Betreuer

Prof. Dr. Arno Hitzges

Datum

August 2023

Wirtschaftsingenieurwesen Medien

www.hdm-stuttgart.de/wing

hitzges@hdm-stuttgart.de

0711/8923-2634

Layout

Jochen Riegg

Fotos und Illustrationen

Innenteil: Apostolos Chatzopoulos

Bachelor-Resümee

Entwicklung einer mobilen Zeiterfassungsapplikation mittels Microsoft Power Apps

Anwendungsentwicklung zur Unterstützung einer Hochschulvorlesung

Apostolos Chatzopoulos

August 2023

Apostolos Chatzopoulos

Apostolos Chatzopoulos begann im Wintersemester 2018 sein Studium im Bereich Wirtschaftsingenieurwesen Medien an der Hochschule der Medien Stuttgart. Sein Schwerpunkt lag auf dem Media Design Management. Im Verlauf seiner Bachelorarbeit konzipierte er eine mobile Anwendung, die dazu diente, die Vorlesung "Testcompany" zu unterstützen. Hierfür nutzte er die Low-Code-Plattform Microsoft Power Apps. Im Rahmen dieser Arbeit analysierte er die Vor- und Nachteile dieser Plattform.

1. Einleitung

Der wachsende Bedarf nach technischen Lösungen und Software-Systemen erfordert von IT-Dienstleistern effiziente und sichere Lösungen in begrenzter Zeit, ohne dabei die höchste Qualität zu vernachlässigen. Angesichts der fortlaufenden Interaktion der Nutzer mit digitalen Anwendungen, sei es im beruflichen oder persönlichen Kontext, entsteht die Erwartung, dass geschäftliche Prozesse sofortig, unkompliziert und in hoher Geschwindigkeit abgewickelt werden sollten. Diese Anforderung hat zur Folge, dass die prompte Entwicklung von Unternehmensanwendungen für IT-Dienstleister als bedeutendste Herausforderung betrachtet wird [1]. Des Weiterem zeigt sich ein wachsender Mangel an Fachkräften für die Digitalisierung von Unternehmen. Die Anzahl der offenen Stellen für IT-Experten erhöhte sich im Jahr 2021 branchenübergreifend auf 96.000, was einem Anstieg von 12 Prozent im Vergleich zum Vorjahr entspricht. [2]. Dies führt zu weniger Zeit für eine steigende Anzahl von Projekten und somit zu einer verlängerten Zeitspanne bis zur Markteinführung neuer Softwaresysteme [3].

Um dem Mangel an Fachkräften entgegenzuwirken und die erforderlichen Softwarelösungen bereitzustellen, besteht eine Lösung darin, auch Nicht-Entwickler in den Prozess der Anwendungserstellung zu integrieren. Daher gewinnt das Low-Code-Programmierparadigma an Bedeutung. Dieses Ansatz reflektiert die Möglichkeit, Software mit minimalem manuellen Programmieraufwand zu entwickeln. Durch die reduzierte Menge an benötigtem Quellcode verringert sich sowohl der Gesamtaufwand für die Anwendungsentwicklung als auch die Vorabinvestitionen für Aspekte wie Einrichtung, Schulung und Bereitstellung. [4]. Low-Code-Entwicklungsplattformen nutzen Techniken aus dem Bereich der modellbasierten und visuellen Softwareentwicklung, um diese Technologie umzusetzen [5].

Info

Mithilfe von **Low-Code-Entwicklungsplattformen** können Business-Anwendungen schnell und mit minimalem Programmieraufwand erstellt und bereitgestellt werden.

Citizen Developer sind Personen außerhalb der IT-Abteilung, die über geringe bis keine Programmierkenntnisse verfügen und sich dennoch am Softwareentwicklungsprozess beteiligen [6].

Zusätzlich zur begrenzten Nutzung von Quellcode und der Anwendungsentwicklung durch Citizen Developer verspricht die Low-Code-Technologie eine Beschleunigung des Softwareentwicklungsprozesses. Durch die Anwendung des Low-Code-Prinzips sowie den Einsatz von Drag-and-Drop-Funktionen, vorgefertigten Benutzeroberflächen und Modellen für Geschäftsprozesse wird die erforderliche Entwicklungszeit reduziert. Laut der PEAK Matrix 2022 der Everest Group geben 76% der Low-Code-Kunden an, dass sie in weniger als sechs Monaten die Anwendung in die Produktionsumgebung integrieren und somit erfolgreich in den Betrieb überführen können [7]. Dennoch birgt die Technologie auch Herausforderungen: Die Unabhängigkeit der Fachabteilungen und die schnelle Bereitstellung von Anwendungen für verschiedene Zwecke könnten dazu führen, dass die IT-Abteilungen den Überblick über die im Unternehmen genutzten Anwendungen verlieren. Dies kann wiederum dazu führen, dass im Unternehmen Anwendungen verwendet werden, die nicht von der IT abgesegnet wurden. Außerdem können Probleme auftreten, wenn Citizen Developer das Unternehmen verlassen, die aktuell genutzte Geschäftsanwendungen erstellt haben. Es besteht die Möglichkeit, dass der Zugriff auf die Anwendung verloren geht, was Wartungen oder Änderungen unmöglich machen kann [8].

2. Anwendungsfall

Der Studiengang Wirtschaftsingenieurwesen Medien an der Hochschule der Medien (HdM) in Stuttgart bereitet Studierende auf eine Karriere im Zeitalter der Medien und Information vor. Nach Abschluss des Studiums sollen die Absolventen als Bindeglied zwischen der kreativen Tätigkeit von Designern und der technischen Arbeit von Entwicklern fungieren. [9].

Im Verlauf des Studiengangs Wirtschaftsingenieurwesen Medien nehmen Studierende am Modul Content Management Systeme (CMS) teil, das von Professor Dr. Arno Hitzges unterrichtet wird. Im Zuge dieser Veranstaltung erwerben die Studierenden grundlegendes Wissen über CMS und gewinnen Einblicke in Plattformen wie Wordpress, Microsoft SharePoint oder Microsoft Power Automate. Daher sollte den Studierenden über die Low-Code-Plattform Power Apps eine mobile Applikation zur Verfügung gestellt werden, die die Inhalte der Vorlesung zu CMS parallel zur neuen Vorlesung Testcompany, die ebenfalls von Prof. Dr. Hitzges angeboten wird, begleitet. Die Applikation sollte dabei folgende Handlungsszenarien enthalten: Zeiterfassung, Zeitnachtragung, Einsicht des eigenen Journals sowie Projektjournals und Projektabschluss. Neben der Entwicklung der Anwendung sollte auch analysiert werden, inwieweit es möglich war, die identifizierten Anforderungen an die Software mithilfe der Low-Code-Entwicklungsplattform umzusetzen, ohne dabei auf unzulässige Kopien zurückzugreifen. Des Weiteren erfolgte eine kritische Untersuchung darüber, inwiefern die Entwicklung mittels Power Apps den Prinzipien des Low-Code-Programmierparadigmas entspricht. Dabei wurden auch die Vorzüge und Limitationen im Vergleich zur konventionellen Softwareentwicklung analysiert.

Info

Power Apps repräsentiert eine Low-Code-Entwicklungsplattform, die von Microsoft entwickelt wurde. Sie ermöglicht das Erstellen von mobilen und webbasierten Anwendungen, die gezielt auf die Lösung spezifischer Probleme in Organisationen ausgerichtet sind. Diese Plattform ist ein integraler Bestandteil der Microsoft Power Platform.

Durch die **Microsoft Power Platform** erhalten Unternehmen die Gelegenheit, mithilfe von Workflows, Datenvisualisierung und anderen Werkzeugen Lösungen zu gestalten. Diese Lösungen sind darauf ausgerichtet, den kontinuierlich wechselnden Anforderungen gerecht zu werden und die unternehmerische Agilität zu erhöhen. [10].

3. Anforderungsanalyse

Um die relevanten Anforderungen zu erfassen, wurden zunächst die beteiligten Akteure ermittelt. Dadurch konnten die Ziele und Erwartungen an die Applikation definiert werden. Die geplante Anwendung wies vier unterschiedliche Stakeholder-Kategorien auf: Betreiber, Benutzer, Entwickler und IT-Administratoren. Die Benutzergruppe umfasst verschiedene Mitglieder, darunter Prof. Dr. Arno Hitzges als Dozenten, aktuelle Studierende sowie zukünftige Dozenten und Studierende der Testcompanyvorlesung. Das Entwicklungsteam setzt sich aus dem Auftraggeber Prof. Dr. Hitzges und dem Auftragnehmer zusammen. Gemeinsam stellen sie die erforderlichen Ressourcen bereit, um eine Lösung zu entwickeln und das Projekt umzusetzen. Die Gruppe der IT-Administratoren umfasst sämtliche Verantwortliche, die zukünftig die Anwendung verwalten, erweitern und warten werden.

Info

Stakeholder repräsentieren Anspruchsberechtigte. Sie sind integraler Bestandteil des Systemumfelds und somit maßgeblich für die Anforderungen an die zu entwickelnde Anwendung verantwortlich. [11].

Um die Anforderungen weiter zu analysieren, wurde ein Use-Case-Modell verwendet. Dieses Modell bot die Gelegenheit, die Funktionen der Anwendung aus der Perspektive der Benutzer in sinnvoll zusammenhängende und überschaubare funktionale Einheiten zu strukturieren. [12]. Hierdurch wurde eine sowohl abstrakte als auch detaillierte Betrachtung des Systems, seiner Umgebung und der Interaktionen der Nutzer ermöglicht. Dieses Modell setzt sich aus Use-Case-Spezifikationen sowie einem Use-Case-Diagramm zusammen. Während die Use-Case-Spezifikation jeden Anwendungsfall (Use Case) im Detail beschreibt, stellt das Use-Case-Diagramm eine visuelle Darstellung der Gesamtheit aller Use Cases dar, die von einem Akteur ausgelöst werden können. [12]. Ausgehend von den gewünschten Szenarien der Anwendung wurden insgesamt elf Use Cases modelliert. Diese repräsentieren Anwendungsfälle und Handlungen, die im Kontext der Zeiterfassung, Zeitnachtragung, des eigenen Journals, Projektjournals und Projektabschlusses durchgeführt werden können. Das entsprechende Use-Case-Diagramm wurde in Abbildung 1 präsentiert. Es vermittelt einen visuellen Überblick über die Gesamtheit der zwölf dokumentierten Use-Case-Spezifikationen und zeigt deren Zusammenhänge zueinander auf.

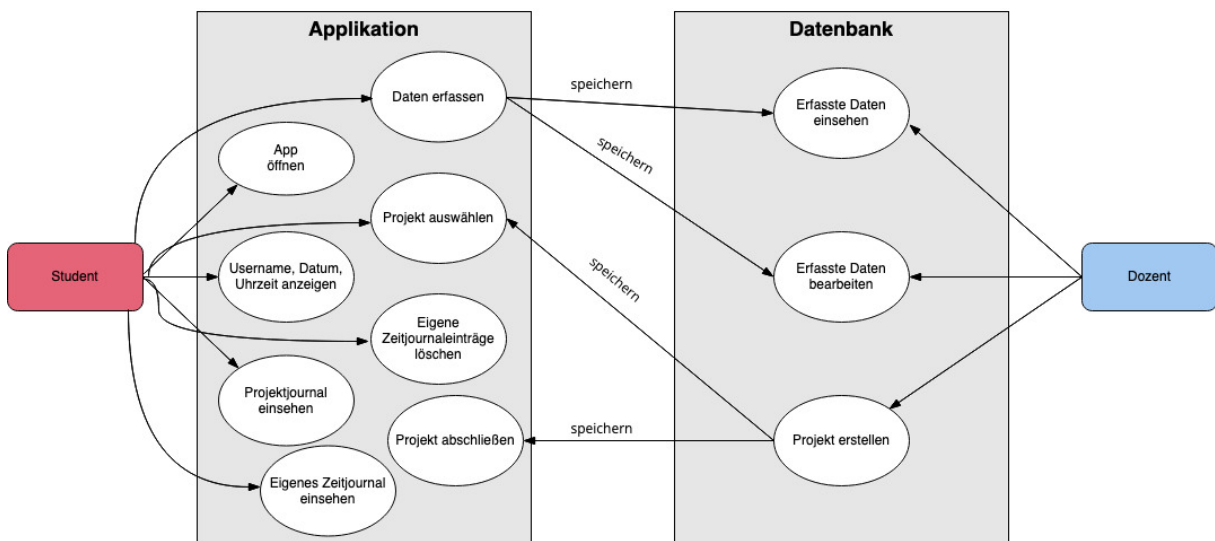


Abbildung 1: Use Case Diagramm aus der Menge aller spezifizierten Anwendungsfälle. (Links): Interaktionen der Studierenden mit der App, (Rechts): Interaktion des Dozenten mit der Datenbankumgebung.

Mithilfe der Ziele und Erwartungen der Stakeholder sowie der elf Use Cases konnten insgesamt 34 konkrete Anforderungen an die Applikation abgeleitet werden. Diese Anforderungen wurden gemäß dem IEEE-Standard in einer Anforderungsdokumentation formuliert und unter Berücksichtigung der Interessen der Stakeholder angemessen zusammengestellt [13]. Im abschließenden Schritt der Analyse wurden die Anforderungen nach ihrer Priorität sortiert. Von den insgesamt 34 Anforderungen erhielten achtzehn die hohe Priorität, neun wurden als mittelpriorisiert eingestuft, während sieben mit einer niedrigen Priorität versehen wurden. Die Realisierung der Anforderungen mit hoher Priorität hatte

oberste Priorität. Anforderungen mittlerer Priorität wurden im Projekt als sogenannte Wunschanforderungen betrachtet. Die Umsetzung dieser Anforderungen war gewünscht, jedoch nicht zwingend notwendig. Die Umsetzung dieser Anforderungen wurde zwar gewünscht, war jedoch nicht zwingend erforderlich. Niedrig priorisierte Anforderungen hatten keinen direkten Einfluss auf den Erfolg des Projekts.

4. Entwurfsphase

Der App-Entwurf bildete den Abschnitt des Projekts, in dem näher erläutert wurde, was in der prototypischen Umsetzung basierend auf den Informationen aus der Anforderungsanalyse entwickelt werden sollte. In dieser Phase wurden das Design der Benutzeroberfläche (User Interface, UI) sowie die Architektur des Datenflusses vorbereitet. Das konzeptionelle Design diente als umfassende Gestaltung der Anwendung. Dies umfasste den Navigationsfluss der Applikation sowie die Festlegung der Benutzeroberflächen mithilfe von ersten Entwurfsskizzen und Wireframes. Für die Nutzernavigation wurde eine hierarchische Struktur gewählt, bestehend aus einer Hauptmenüseite und zusätzlichen Unterseiten. Durch das Anklicken von Icons oder Schaltflächen bewegt sich der Nutzer innerhalb eines Funktionsbereichs kontinuierlich voran. Der Vorteil dieser Navigationsstruktur liegt darin, dass der Nutzer seine gewünschten Ziele schnell erreichen kann. Zudem ist diese Form der Navigation bereits aus Ordnerstrukturen auf Desktops oder anderen Mobilapplikationen vertraut [14]. Die Wireframes wurden unter Verwendung von ersten Entwurfsskizzen erstellt und decken jede individuelle Ansicht in der App sowie die detaillierte Funktionsbeschreibung ab. Die erstellten Wireframes dienten als Grundlage für die Realisierung des UI-Designs in Power Apps. Abbildung 2 zeigt die schematische Darstellung der Funktion *Zeiterfassung* in Form von Wireframes.

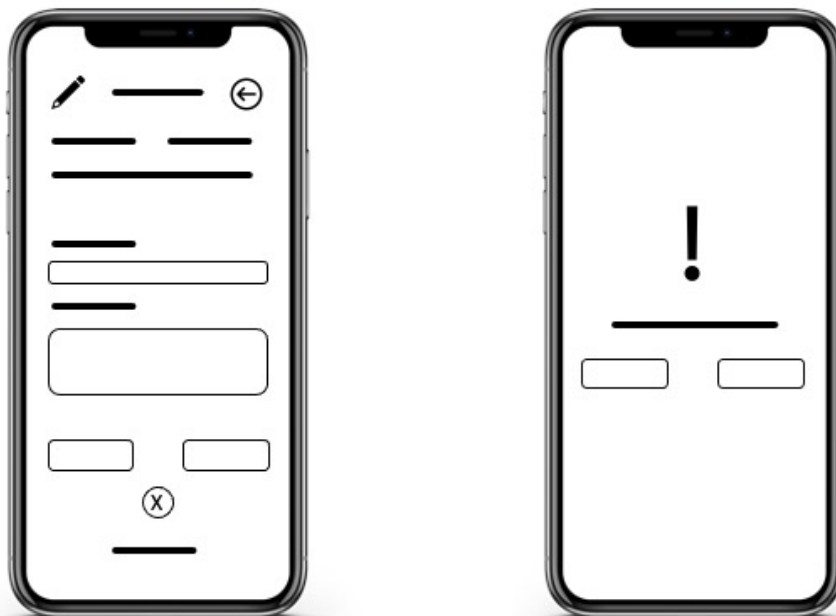


Abbildung 2: Schematische Darstellung der Funktion Zeiterfassung und Verify in Form von Wireframes.

Das Resultat des anschließenden architektonischen Entwurfs umfasste die Datenlogik und -struktur der Anwendung. Die Entscheidung, sich ausschließlich auf SharePoint als Datenfluss-Konnektor zu beschränken, trägt dazu bei, die Anwendung benutzerfreundlich zu gestalten und die benötigte Rechenleistung auf dem ausführenden Mobilgerät so gering wie möglich zu halten. Neben der Datenspeicherung fungiert SharePoint auch als Datenbank für den Vorlesungsdozenten. Die eigentliche App dient lediglich als Benutzeroberfläche (Interface). Alle App-Nutzer haben innerhalb der Anwendung dieselben Berechtigungsrollen. Mit der Ausnahme der *Projektabschluss* Funktion. Diese Funktion ist nur für den Dozenten oder IT-Administrator nutzbar.

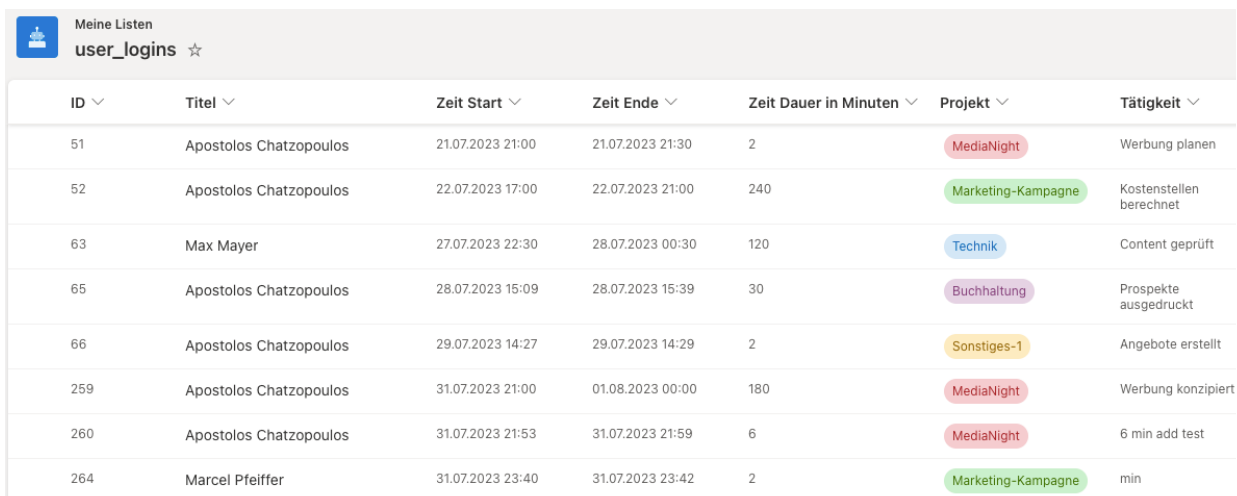
5. Entwicklungsphase

In diesem Abschnitt der Arbeit wurde die Anwendung mithilfe der Low-Code-Plattform Power Apps auf Basis der zuvor erzielten Ergebnisse entwickelt. Das Resultat der prototypischen Umsetzung war eine mobile Applikation mit vier Funktionsszenarien: Zeiterfassung, Zeitrnachtrag, Journalansichten und Projektabschluss. Im weiteren Verlauf wird spezifisch auf die Realisierung der Funktion *Start der Zeiterfassung* und die Anwendung von **Power Fx**-Formeln in der Power Apps-Entwicklungsumgebung eingegangen.

Info

Power Fx ist die deklarative und funktionale Programmiersprache der Microsoft Power Plattform, die in menschenlesbarem Text ausgedrückt wird. Durch ihre klare und einfache Struktur soll Power Fx auch sogenannten Citizen Developern die Möglichkeit bieten, Anwendungen zu entwickeln [15].

Um die relevanten Daten innerhalb der App zu speichern, wurde zunächst eine SharePoint-Liste erstellt. Diese ermöglichte es dem Nutzer, Informationen wie Startzeit, Projekt und Tätigkeit zu erfassen und zu speichern (siehe Abbildung 3).



ID	Titel	Zeit Start	Zeit Ende	Zeit Dauer in Minuten	Projekt	Tätigkeit
51	Apostolos Chatzopoulos	21.07.2023 21:00	21.07.2023 21:30	2	MediaNight	Werbung planen
52	Apostolos Chatzopoulos	22.07.2023 17:00	22.07.2023 21:00	240	Marketing-Kampagne	Kostenstellen berechnet
63	Max Mayer	27.07.2023 22:30	28.07.2023 00:30	120	Technik	Content geprüft
65	Apostolos Chatzopoulos	28.07.2023 15:09	28.07.2023 15:39	30	Buchhaltung	Prospekte ausgedruckt
66	Apostolos Chatzopoulos	29.07.2023 14:27	29.07.2023 14:29	2	Sonstiges-1	Angebote erstellt
259	Apostolos Chatzopoulos	31.07.2023 21:00	01.08.2023 00:00	180	MediaNight	Werbung konzipiert
260	Apostolos Chatzopoulos	31.07.2023 21:53	31.07.2023 21:59	6	MediaNight	6 min add test
264	Marcel Pfeiffer	31.07.2023 23:40	31.07.2023 23:42	2	Marketing-Kampagne	min

Abbildung 3: Beispielhafte Darstellung der SharePoint-Liste zur Speicherung und Abrufung von Daten der Zeiterfassungsapplikation.

Um das Erfassen der Zeit zu starten, wird zunächst eine Trigger-Schaltfläche implementiert, um einen in der Applikation für Benutzer unsichtbar formatierten *Timer* zu starten und die Daten wie Projekt, Tätigkeit, Benutzername, Zeit Start und Active Session an die SharePoint-Liste zu senden. Der gegebene Code veranschaulicht die Verwendung mehrerer Funktionen und Aktionen, um Daten in einer App zu steuern und zu aktualisieren (siehe Abbildung 4).

```
If ( . . . ) :
```

If ist eine bedingte Anweisung. Sie überprüft eine Bedingung und führt den darin enthaltenen Code aus, wenn die Bedingung erfüllt ist.

```
!IsBlank(projekt_choose.Selected.Value)  
&& !IsBlank(taetigkeit_txt.Text) :
```

Dies ist die Bedingung der ersten *If*-Anweisung. Sie prüft, ob sowohl ein ausgewähltes Projekt (`projekt_choose.Selected.Value`) als auch ein Text im `taetigkeit_txt`-Feld vorhanden ist. Somit wird verhindert, dass der Benutzer lückenhafte Daten an die SharePoint-Liste sendet.

```
Set (varNewItem; . . .) :
```

Nun wird eine Variable *varNewItem* erstellt und mit Werten befüllt. Der *Patch*-Befehl wird verwendet, um einen neuen Datensatz in der Datenquelle *user_logins* zu erstellen. Dabei werden die verschiedenen Spalten, wie *Titel*, *Projekt*, *Tätigkeit*, *Zeit Start -9h* und *Active Session*, in der SharePoint-Liste mit Werten belegt.

```
&& :
```

&& ist ein logischer Operator für „und“. In diesem Fall wird überprüft, ob der vorherige Ausdruck (`Set (varNewItem; . . .)`) erfolgreich ist, und falls ja, wird der folgende Code fortgesetzt.

```
UpdateContext ({varTimerActive:true}) :
```

Hier wird der Wert der Kontextvariable *varTimerActive* auf „true“ aktualisiert, um den unsichtbaren Timer zu aktivieren.

```
UpdateContext ({varTimerReset:false}) :
```

Anschließend wird der Wert der Kontextvariable *varTimerReset* auf „false“ aktualisiert. Um den Timer von seinem zurückgesetzten Zustand zu lösen.

The screenshot shows a mobile application interface for time tracking. At the top, there is a header bar with a red border containing a pencil icon, the title "Zeiterfassung", and a back arrow. Below the header, a grey box displays the date "Datum: 12.08.2023", the time "Uhrzeit: 17:04:11", and the user "User: Apostolos Chatzopoulos". The main area has a "Projekt" section with a dropdown menu showing "Marketing-Kampagne". Below that is a "Tätigkeit" section with a text input field containing "Werbung konzipiert". At the bottom, there are two buttons: "kommen" (grey) and "gehen" (blue). A red "X" icon is positioned above the "Aktuelle Zeit:" label, which shows "00:01:20".

Abbildung 4: Beispielhafte Darstellung der Zeiterfassungsfunktion zur Speicherung von Daten der Zeiterfassungsapplikation.

6. Auswertung

Nach den ersten Tests der Zeiterfassung fiel auf, dass die erfassten Uhrzeiten in der Spalte „Zeit Start -9h“ und „Zeit Ende -9h“ fehlerhaft abgespeichert wurden. Eine Gegenprüfung konnte durch das eigene Zeitjournal abgeglichen werden. Hier wurde der Zeitstempel des Starts beispielsweise um 15:00 Uhr mit einer Eintragung von 15:00 Uhr richtig erfasst. Der finale Lösungsansatz wurde durch das Einsetzen einer Formel in die bestimmte Spalte der Zeiterfassung eingetragen. Über die Einstellungen der SharePoint-Liste werden in den neu erstellten Spalten, in denen die richtige Zeit ausgegeben werden soll, folgende Formeln eingegeben.

Insgesamt konnten von 33 Anforderungen erfüllt werden. Davon sind 18 mit hoher Priorität. Lediglich eine Anforderung mit mittlerer Priorität konnte nur zum Teil oder mit Einschränkungen erfüllt werden.

7. Ergebnis

Die Resultate der Arbeit bestätigen, dass die Entwicklung einer App inklusive Datenverbindung zu SharePoint ohne umfassende IT-Kenntnisse und Programmierfähigkeiten umsetzbar ist. Dennoch kann es von Vorteil sein, über ein solides Verständnis von Informationstechnologie zu verfügen, wenn man mithilfe von Power Apps eine Anwendung entwickeln möchte. Bei komplexeren Anwendungen sind zwar grundlegende IT-Kenntnisse und Programmierfähigkeiten von Vorteil, jedoch ist es nicht zwingend notwendig, vollständige Programmiersprachen zu beherrschen. Im Vergleich zur konventionellen Softwareentwicklung erfordert die Erstellung einer Low-Code-Anwendung trotz der genannten Aspekte deutlich weniger umfassende IT-Kenntnisse und Programmierfähigkeiten. Die Low-Code-Entwicklungstechnologie gilt aufgrund ihrer Anforderungen an IT-Erfahrung als leichter erlernbar im Vergleich zu einer herkömmlichen Programmiersprache, die zusätzlich spezifische Frameworks erfordert.

Die Arbeit hat außerdem gezeigt, dass die Anwendungsentwicklung über eine Low-Code-Plattform zu einer gewissen Zeitersparnis führt. Die Plattform stellt alle notwendigen Grundlagen zur Verfügung, was die schnelle Entwicklung der verschiedenen Anwendungsseiten sowie kleinerer Funktionen ermöglichte. Die prototypische Umsetzung hat jedoch auch verdeutlicht, dass Low-Code-Anwendungen im Vergleich zu Anwendungen der herkömmlichen Softwareentwicklung weniger Flexibilität aufweisen können. Zusätzlich zur vollständigen Kontrolle über alle Elemente und der Möglichkeit, diese individuell und flexibel anzupassen, hat ein Entwickler in der herkömmlichen Softwareentwicklung die Freiheit, die Programmiersprache seiner Wahl zu verwenden. In der Low-Code-Entwicklung ist der Entwickler auf die vorhandenen Komponenten und die spezifische Plattform-Syntax beschränkt. Eigenständiger Code kann mittels Power Fx-Formeln lediglich für bereits existierende Eigenschaften von vorgefertigten Komponenten verfasst werden. Das impliziert, dass falls eine Funktion in der Low-Code-Umgebung nicht mithilfe einer vorhandenen UI-Komponente realisierbar ist oder ein Objekt nicht über die erforderliche Eigenschaft verfügt, wird diese Funktion nicht umsetzbar sein.

8. Referenzen

- [1] J. Rymer und R. Koplowitz, „Low-Code Development Platforms for AD&D Professionals - The 13 Providers That Matter Most And How They Stack Up,“ The Forrester Wave, 2019.
- [2] „Bitkom Research,“ 3 Januar 2022. [Online]. Available: <https://www.bitkom-research.de/de/pressemitteilung/it-fachkraefteluecke-wird-groesser-96000-offene-jobs>. [Zugriff am 26 Oktober 2022].
- [3] JobRouter, „Whitepaper | Low-Code für digitale Prozesse in Unternehmen,“ 20 Februar 2020. [Online]. Available: <https://www.jobrouter.com/de/download/whitepaper-low-code-fuer-digitale-prozesse-in-unternehmen/>.
- [4] D. Di Ruscio, „Low-code development and model driven engineering: Two sides of the same coin?,“ SpringerLink, 2022.
- [5] R. Waszkowski, „Low-code platform for automating business processes in manufacturing. IFAC-PapersOnLine,“ 2019. [Online]. Available: <https://doi.org/10.1016/j.ifacol.2019.10.060>.
- [6] C. Rentrop und S. Augsten, „Low-Code Entwicklung - Ansatz und Tools. Dev-Insider,“ 22 Oktober 2018. [Online]. Available: <https://www.dev-insider.de/low-code-entwicklung-ansatz-und-tools-a-763889/>. [Zugriff am 26 Oktober 2022].
- [7] Everest Group, „Low-code Application Development Platforms PEAK Matrix Assessment 2022,“ Everest Group, 2022.
- [8] M. Koller, „Low Code - Wie sinnvoll ist das Konzept wirklich?,“ 21 Januar 2021. [Online]. Available: <https://www.upgreat.ch/blog/low-code-wie-sinnvoll-ist-das-konzept-wirklich>. [Zugriff am 4 November 2022].
- [9] H. Stuttgart, „Studienschwerpunkt Informationstechnologie,“ o.D.. [Online]. Available: https://www.hdm-stuttgart.de/wing/studieninteressierte/studieninhalte/studieninhalte_schwerpunkt_dpt.html. [Zugriff am 8 November 2022].
- [10] M. Learn, „Was ist Microsoft Power Platform?,“ o.D.. [Online]. Available: <https://learn.microsoft.com/de-de/training/modules/introduction-power-platform/2-what-is-power-platform>. [Zugriff am 19 Dezember 2022].
- [11] H. Tresp, „Agile objektorientierte Anforderungsanalyse: Planen - Ermitteln - Analysieren - Modellieren - Dokumentieren - Prüfen (erfolgreich studieren),“ Springer Vieweg, 2022, p. 48.

[12]

T. Cziharz, Juli 2022. [Online]. Available: https://www.ireb.org/redirect-download/?lang=de&file=ireb_cpre_handbuch_requirements_modeling_advanced_level_de_v2.0&redirect=https://www.ireb.org/content/downloads/19-handbook-cpre-advanced-level-requirements-modeling/ireb_cpre_handbuch_requirements_mod. [Zugriff am 8 Dezember 2022].

[13] IEEE, IEEE Recommended Practice for Software Requirements Specifications, IEEE Standard 830-1998, 1998.

[14] J. Semler und K. Tschierschke, App-Design: Das umfassende Handbuch., Rheinwerk Design, 2019.

[15] G. Lindhorst, „Microsoft Learn. Microsoft Power Fx Überblick - Power Platform.,“ 7 Juli 2022. [Online]. Available: <https://learn.microsoft.com/de-de/power-platform/power-fx/overview>. [Zugriff am 22 Dezember 2022].