Entwicklung einer Webanwendung zur automatisierten Veröffentlichung von Instagram-Beiträgen

vorgelegt am 14. Januar 2025

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WWI2022F

von

Leonard Eckert, David Kreismann, Tobias Schnarr und Nico Wagner

Inhaltsverzeichnis

Abkürzungsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Zielsetzung und Konzeption

1.1 Zielsetzung

Das Ziel dieses Projekts ist die Entwicklung einer Webanwendung, die das tägliche Posten von Beiträgen auf Instagram automatisiert. Dabei sollen drei verschiedene Beitragstypen unterstützt werden:

- Einzelbild-Posts
- Video-Posts
- Textbild-Posts

Jeder Beitrag soll mit passenden Hashtags versehen werden, um die Sichtbarkeit in sozialen Netzwerken zu erhöhen. Die Automatisierung soll über eine Weboberfläche gesteuert werden, sodass Nutzer die Beiträge zentral verwalten und planen können.

1.2 Konzeption

Um die Zielsetzung zu erreichen, sind verschiedene Funktionalitäten erforderlich. Diese werden als Anforderungen definiert und in drei Kategorien unterteilt: **MUSS**, **SOLL** und **KANN**. Dadurch wird eine klare Priorisierung der Anforderungen ermöglicht. Zur Planung kommen Jira und verschiedene UML-Diagramme zum Einsatz. Die Anforderungen werden in Jira als Stories erfasst und einer Kategorie zugeordnet. Im Laufe der Entwicklung werden die Stories den jeweiligen Sprints zugewiesen. Diese Zuweisung erfolgt im wöchentlichen Rhythmus, um flexibel auf unerwartete Herausforderungen oder Hindernisse reagieren zu können.

Anforderung		
MUSS	**SOLL**	**KANN**
Bild/Video hochladen	Instagram-Login	Berechtigungsverwaltung
Hinzufügen von Hashtags	Planung der Veröffentlichung	Automatische Anpassung von Medien
Hinzufügen von Text	Responsive Design	Erinnerung an ausstehende Beiträge

Tab. 1: Kategoriesierung der Anforderungen

Tabelle 1 zeigt wie die Anforderungen in Kategorien unterteilt werden. Es handelt sich hierbei um besonders relevante Anforderungen aus allen drei Kategorien. Eine vollständige Liste der Anforderung kann in Jira eingesehen werden.

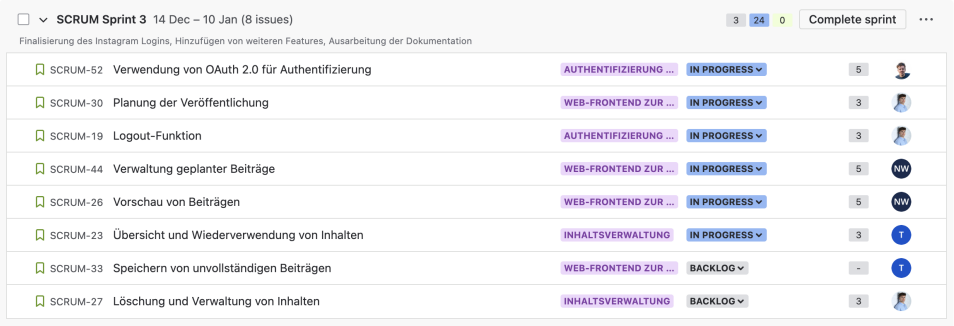

Abb. 1: Sprint 3

In Abbildung 1 ist eine Übersicht von Sprint 3 zu sehen. Die Anzahl der Stories pro Sprint variiert je nach Komplexität und Umfang. Die Entscheidung wie viele Stories in einen Sprint aufgenommen werden, wird im Team getroffen. Dabei wird darauf geachtet, dass die Stories in einem Sprint realistisch umsetzbar und die Storypoints von Sprint zu Sprint annähernd gleich sind. Wird eine Storie in einem Sprint nicht fertiggestellt, wird sie in den nächsten Sprint übernommen.

Die Definition der Anforderungen und die Planung der Sprints in Jira sind wesentliche Bestandteile des Projekts. Sie gewährleisten einen klaren Überblick über die Anforderungen und den Fortschritt während des gesamten Entwicklungsprozesses. Im Folgenden wird die technische Konzeption dargelegt.

Technische Konzeption Im Rahmen der Entwicklung der Webanwendung wurden verschiedene Frameworks und Tools eingesetzt, um eine effiziente, flexible und skalierbare Lösung zu gewährleisten. Diese werden nachfolgend erläutert.

Next.js[1] wurde als zentrales Framework für die Entwicklung der Webanwendung verwendet. Es bietet leistungsstarke Funktionen wie Server-Side Rendering (SSR) und eine hohe Performance, wodurch die Ladezeiten optimiert werden. Zudem ermöglicht die Flexibilität von Next.js eine reibungslose Integration weiterer Technologien.

Für die Automatisierung der Instagram-Beiträge wird die offizielle API von **Meta**[2] genutzt. Diese erlaubt den Zugriff auf Beitragsdaten sowie die Planung von Posts, was die zentrale Funktion der Anwendung unterstützt.

Zur Implementierung einer modernen und modularen Benutzeroberfläche wurde **shadcn/ui**[3] eingesetzt. Dieses UI-Framework zeichnet sich durch modulare Komponenten aus, die einfach erweiterbar sind. Zudem profitiert das Projekt von einer aktiven Community-Driven-Entwicklung, die regelmäßige Updates und Verbesserungen ermöglicht.

Für das Styling der Anwendung wurde **Tailwind CSS**[4] verwendet. Der Utility-First-Ansatz erlaubt eine schnelle und konsistente Gestaltung der Benutzeroberfläche. Dank Responsive Design

[1] Vgl. Vercel 2025
[2] Vgl. Meta Platforms, Inc. 2025
[3] Vgl. shadcn 2025
[4] Vgl. Tailwind Labs 2025

passt sich die Anwendung an verschiedene Bildschirmgrößen an. Zudem steigert Tailwind die Produktivität, da wiederverwendbare Klassen eine effiziente Entwicklung ermöglichen.

Zur sicheren und skalierbaren Verwaltung von Medieninhalten wurde **uploadthing** integriert. Dieses Tool bietet eine hohe Sicherheit für den Datei-Upload, eine nahtlose Next.js-Integration und eine gute Skalierbarkeit, um den wachsenden Anforderungen der Anwendung gerecht zu werden.

Diese beschriebenen Technologien bilden das Fundament der Webanwendung. Im nächsten Abschnitt wird die verwendete Architektur näher betrachtet.

Architektur Um eine durchdachte und strukturierte Architektur zu gewährleisten, wurden zunächst verschiedene UML-Diagramme erstellt. Diese dienen als Grundlage für die Implementierung und ermöglichen eine klare Strukturierung, aus der die einzelnen Komponenten der Anwendung abgeleitet werden können. Durch die Zusammenführung dieser Komponenten entsteht die Gesamtarchitektur der Anwendung.

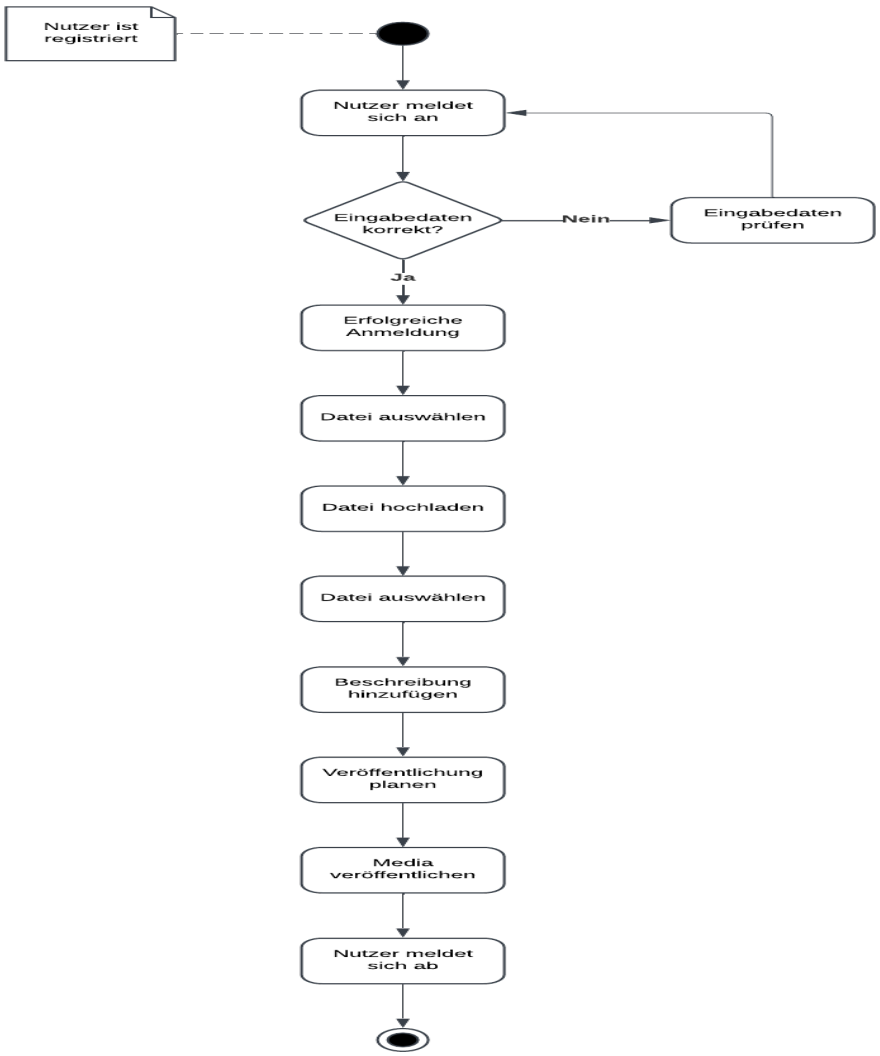

Abb. 2: Aktivitätsdiagramm

In Abbildung 2 ist das **Aktivitätsdiagramm** dargestellt, das den typischen Ablauf der Benutzerinteraktion mit der Webanwendung zur automatisierten Veröffentlichung von Instagram-Beiträgen veranschaulicht. Es zeigt die einzelnen Schritte von der Anmeldung bis zur Veröffentlichung eines Beitrags und der anschließenden Abmeldung des Nutzers. Der Prozess beginnt mit einem bereits registrierten Nutzer, der sich in die Anwendung einloggt. Anschließend erfolgt eine Überprüfung der Eingabedaten. Falls diese nicht korrekt sind, wird der Nutzer zur erneuten Eingabe aufgefordert. Andernfalls erfolgt die erfolgreiche Anmeldung, und der Nutzer gelangt zur Hauptfunktionalität der Anwendung. Nach der Anmeldung kann der Nutzer eine Datei auswählen und diese anschließend hochladen. Dieser Schritt kann sich wiederholen, falls der Nutzer mehrere Dateien hochladen möchte. Anschließend wird eine Beschreibung hinzugefügt, um den Beitrag zu vervollständigen. Danach erfolgt die Planung der Veröffentlichung, bei der der Nutzer den gewünschten Veröffentlichungszeitpunkt festlegt. Sobald die Planung abgeschlossen ist, wird die Medienveröffentlichung durchgeführt. Abschließend meldet sich der Nutzer von der Anwendung ab, womit der Ablauf endet. Das Diagramm bietet eine strukturierte Darstellung der Benutzerinteraktion und stellt alle relevanten Schritte für eine erfolgreiche Beitragserstellung dar.

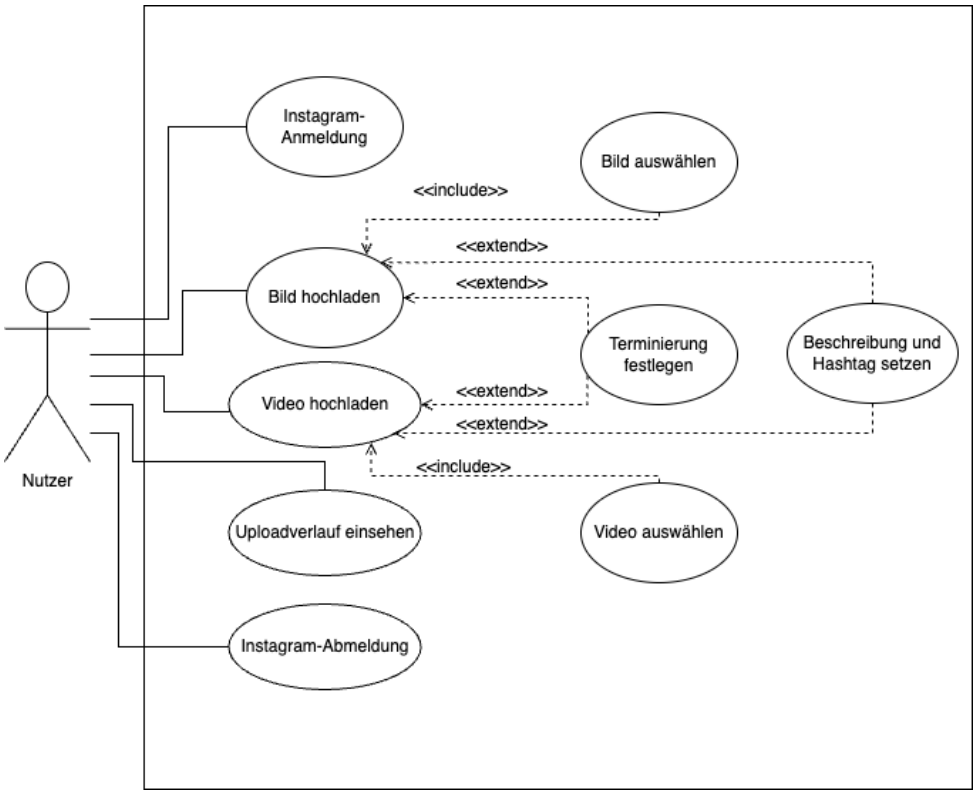

Abb. 3: Use Case Diagramm

Abbildung 3 stellt das **Use Case Diagramm** dar, das die wesentlichen Use Cases der Webanwendung beschreibt. Es zeigt die Interaktionen des Nutzers mit dem System und verdeutlicht die Beziehungen zwischen den einzelnen Anwendungsfällen. Der Nutzer kann sich zunächst über die Instagram-Anmeldung in die Anwendung einloggen.

Nach erfolgreicher Anmeldung stehen ihm mehrere Hauptfunktionen zur Verfügung:

- Bild hochladen
- Video hochladen
- Uploadverlauf einsehen
- Instagram-Abmeldung

Die beiden Anwendungsfälle Bild hochladen und Video hochladen beinhalten jeweils optionale Erweiterungen (*«extend»*):

- Terminierung festlegen: Der Nutzer kann den Zeitpunkt der Veröffentlichung bestimmen.
- Beschreibung und Hashtags setzen: Zusätzlich kann er eine Bildbeschreibung und relevante Hashtags hinzufügen.

Zudem gibt es *«include»*-Beziehungen, die darauf hinweisen, dass bestimmte Aktionen zwingend erforderlich sind:

- Bild hochladen und Video hochladen beinhalten jeweils das Auswählen einer Datei (Bild oder Video), bevor der Upload erfolgen kann.
- Uploadverlauf einsehen ist ein separater Use Case, der dem Nutzer ermöglicht, eine Übersicht über bereits hochgeladene Inhalte zu erhalten.

Das Diagramm visualisiert die Modularität und Erweiterbarkeit der Anwendung, indem es optionale (*«extend»*) und zwingend notwendige (*«include»*) Abhängigkeiten zwischen den Use Cases darstellt. Dadurch wird eine strukturierte und skalierbare Architektur unterstützt.

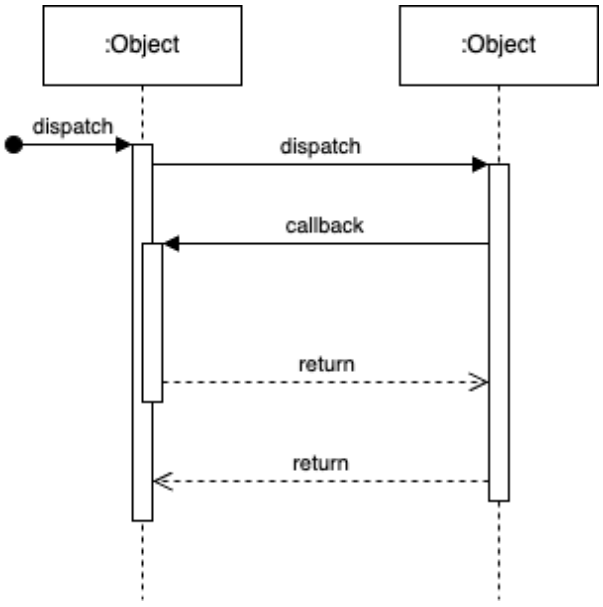

Abb. 4: Sequenzdiagramm

In Abbildung 4 ist das **Sequenzdiagramm** dargestellt, das die Interaktion zwischen zwei Objekten im System beschreibt. Es zeigt den Ablauf einer synchronen Kommunikation zwischen den Objekten, einschließlich Nachrichtenfluss, Rückgabewerten und Lebenslinien. Der Prozess beginnt mit einer Dispatch-Nachricht, die von einem Objekt an ein zweites Objekt gesendet wird. Nach dem Empfang der Nachricht verarbeitet das zweite Objekt die Anfrage und sendet eine Callback-Nachricht an das erste Objekt zurück. Anschließend werden zwei Rückgabenachrichten (Return) ausgetauscht, um die Kommunikation abzuschließen. Die vertikalen gestrichelten Linien (Lebenslinien) zeigen die Existenzdauer der Objekte während der Interaktion. Die horizontalen Pfeile repräsentieren den Nachrichtenfluss zwischen den Objekten, wobei durchgezogene Pfeile direkte Methodenaufrufe und gestrichelte Pfeile Rückgaben oder asynchrone Antworten darstellen.

2 Umsetzung

Erklärung zur Verwendung generativer KI-Systeme

Bei der Erstellung der eingereichten Arbeit habe ich die nachfolgend aufgeführten auf künstlicher Intelligenz (KI) basierten Systeme benutzt:

1. ChatGPT

Ich erkläre, dass ich

- mich aktiv über die Leistungsfähigkeit und Beschränkungen der oben genannten KI-Systeme informiert habe,[5]
- die aus den oben angegebenen KI-Systemen direkt oder sinngemäß übernommenen Passagen gekennzeichnet habe,
- überprüft habe, dass die mithilfe der oben genannten KI-Systeme generierten und von mir übernommenen Inhalte faktisch richtig sind,
- mir bewusst bin, dass ich als Autorin bzw. Autor dieser Arbeit die Verantwortung für die in ihr gemachten Angaben und Aussagen trage.

Die oben genannten KI-Systeme habe ich wie im Folgenden dargestellt eingesetzt:

Arbeitsschritt in der wissenschaftlichen Arbeit	**Eingesetzte(s) KI-System(e)**	**Beschreibung der Verwendungsweise**
Korrektur der Arbeit	ChatGPT	Einzelne Kapitel ChatGPT zum Korrigieren gegeben. Erfolg: geringfügig, nach der Korrektur wurden noch einige Fehler gefunden.

[5] U.a. gilt es hierbei zu beachten, dass an KI weitergegebene Inhalte ggf. als Trainingsdaten genutzt und wiederverwendet werden. Dies ist insb. für betriebliche Aspekte als kritisch einzustufen.

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Entwicklung einer Webanwendung zur automatisierten Veröffentlichung von Instagram-Beiträgen* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

(Ort, Datum) (Unterschrift)