



OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

Angewandte Informatik an der
Fakultät Wirtschaftsinformatik und Angewandte Informatik

MASTERARBEIT

Entwicklung eines interaktiven Werkzeugs zur Kuratierung von Umweltdaten einer bürgerinitiierten Crowdsensing-Kampagne

VON

Samet Murat Akcabay

Betreuerinnen:

Prof. Dr. Daniela Nicklas

Lehrstuhl für Informatik, insbesondere Mobile Softwaresysteme/Mobilität

15. November 2023

Zusammenfassung

Danksagung

Prof. Dr. Thomas Foken Bürgerverein Bamberg Mitte e.V.

Inhaltsverzeichnis

KAPITEL 1	Einleitung	3
KAPITEL 2	Hintergrund	5
2.1	Crowdsensing vs. Crowdsourcing	5
2.2	Verwandte Arbeiten	7
KAPITEL 3	Konzept	11
3.1	Grundgedanke der Arbeit	11
3.2	Methodologie zum Aufstellen der Anforderungen	12
3.3	Erfassung der notwendigen Komponenten der Software	12
3.4	Umsetzung	13
KAPITEL 4	Anforderungsanalyse	17
4.1	Ausgangslage in Bamberg	17
4.2	Die Netatmo API — eine Schnittstelle zwischen Sensordaten und Stakeholder	21
4.3	Definition der Anforderungen	21
4.4	Use Cases	21
KAPITEL 5	User Stories	23
KAPITEL 6	Implementierung und Design	25
6.1	Containervirtualisierung mit Docker	25
6.2	Architektur für das Front-End	26
6.3	Architektur für das Back-End	28
KAPITEL 7	Diskussion und Evaluation	31
7.1	Analyse der Anforderungen	31
7.2	Evaluation auf Grundlage der Literatur	31
7.3	Evaluation auf Grundlage der Stakeholder	31
7.4	Limitationen der Arbeit	31
KAPITEL 8	Fazit und Ausblick	33
	Literaturverzeichnis	35
	Eidesstattliche Erklärung	37

Abkürzungsverzeichnis

BVM Bürgerverein Bamberg Mitte e.V.

IEE Institute of Electrical and Electronics Engineers

PIW Post-Iteration Workshop

LCZ Local Climate Zones

DWD Deutscher Wetterdienst

MOBI-Lehrstuhl Lehrstuhl für Informatik, insbesondere Mobile Softwaresysteme/Mobilität der Universität Bamberg

1 | Einleitung

Umwelt immer wichtiger, Klimawandel, Erderwärmung... Politik, aber auch Unternehmen setzt immer mehr darauf (Statistik hier zB), umso wichtiger ist es, das Thema Umwelt/Nachhaltigkeit in immer mehr Bereiche zu integrieren, unter anderem in die Softwareentwicklung. Allerdings fühlen sich immer mehr Bürger von Politik abgehängt, oder nicht respektiert/berücksichtigt, also braucht man Hard Facts/Statistik, um vorweisen zu können, was abgeht und welche Maßnahmen erforderlich sind. Was muss beim Städtebau berücksichtigt werden, wo sind die wärmsten Bereiche in einer Stadt und warum? Um dies aufzuzeigen, mein Tool als Lösung. Zwei Fliegen mit einer Klappe, da der Aspekt des Crowdsensings in die Softwareentwicklung miteinfließt und so nachgewiesen werden kann, wie sich das auf die Softwareentwicklung auswirkt und ob das überhaupt sinnvoll ist, das zu berücksichtigen. Wenn ja, eventuell in der Zukunft viel größerer Fokus darauf?

acro! (acro!) Bürgerverein Mitte mit dem Klimamessnetz in Bamberg, initiiert durch Prof. Foken. Regelmäßige Analysen und Auswertungen, z.T. sogar veröffentlicht, um aufzuzeigen, was falsch läuft in der Stadt. Allgemeines Ziel des BVM ist das Fördern von mehr Grün in der Innenstadt, als auch die Reduzierung des Verkehrs. Analysen und Auswertungen erfolgen dabei aber manuell, und es kommt regelmäßig zu Anomalien/Fehlern. Hier kann man natürlich irgendwelche qualitätssteigernde Algorithmen drüber laufen lassen, aber das erwischt natürlich auch nicht alles (Verweis auf Reem) - also Einbinden des Crowdsensing-Aspekts, als zusätzliche Qualitätsschicht.

Mehr grün in der Innenstadt, Reduzierung von Verkehr sind Ziele des BVM. Mein Ziel ist das Schaffen einer Schnittstelle zwischen Stakeholder und Sensordaten, um das Erreichen der Ziele der Stakeholder zu unterstützen. Am Ende möchte ich noch zeigen, ob der Aspekt des Crowdsensings irgendwas bringt in der Softwareentwicklung oder ob das zu vernachlässigen ist, vor allem auf die Langzeit betrachtet. Tool soll Auswertungen unterstützen und bei deren Treffen regelmäßig zum Einsatz kommen.

2 | Hintergrund

In diesem Abschnitt der Arbeit geht es darum, eine Grundlinie des theoretischen Hintergrunds zu schaffen. Dieser Aspekt wird dadurch umgesetzt, indem zunächst der Begriff des „Crowdsensings“ mit dem in der Literatur weitverbreiteten Begriff des „Crowdsourcings“ abgegrenzt wird. Infolgedessen ist es somit in den folgenden Kapiteln möglich, den Begriff des Crowdsensings in den Kontext der Arbeit zu setzen. Im Anschluss werden verwandte Arbeiten vorgestellt, welche sich mit dem Thema des Crowdsensings und der Umsetzung von Plattformen mit diesem Hintergrund beschäftigen.

2.1 Crowdsensing vs. Crowdsourcing

In der Literatur wird der Begriff des (mobilen) Crowdsensings durch das Vorhandensein einer großen Anzahl an Teilnehmenden für eine großflächige Überwachung der Umwelt beschrieben, welche rohe Daten mithilfe von in smarte Geräte eingebettete Sensoren messen [Ray+22]. Um an den Begriff des Crowdsensings heranzutreten, wird aber zunächst zwischen zwei Arten des Messens unterschieden: dem sogenannten „personal sensing“, in dessen Anwendung Individuen persönliche Informationen aus eigenem Interesse oder Bedarf (z.B. Messungen zum Überwachen der eigenen Gesundheit, aber auch zum Nachverfolgen von persönlichen Rekorden oder dem ökologischen Fußabdruck) messen und dem „community sensing“, bei welchem großflächige Phänomene, welche durch einzelne Individuen nicht gemessen werden können, untersucht werden [GYL11].

Als Beispiele können hierfür sämtliche Anwendungsfälle genannt werden, in denen die Teilnahme von mehreren Individuen (unter Umständen zur selben Zeit) unabdingbar ist, wie beim Messen der Temperatur, der Luftfeuchtigkeit oder der Luftqualität an verschiedenen Orten. Der Begriff des „community sensing“ kann hierbei weiterhin in zwei Kategorien unterteilt werden, dem sogenannten „participatory sensing“ [Bur+06] und dem „opportunistic sensing“ [Lan+10]:

Während bei Ersterem eine aktive Teilnahme, verbunden mit eigenmotiviertem Aufwand (z.B. Aufnahme von Fotos, Eingabe und Übermittlung von Information), notwendig ist, wird bei Letzterem eher ein minimaler Aufwand durch eine selbstständige Messung der Sensoren (z.B. kontinuierliche Temperaturmessung der Sensoren in Abhängigkeit vom Standort, ohne Eingaben der Nutzenden) betrieben [GYL11]. Aus diesem Grund

bildet der Begriff des Crowdsensing in der Literatur keine eindeutige Art der Messung, sondern vielmehr eine Domäne der genannten Arten an Messungen durch eine Gruppe von Individuen [GYL11].

Das mobile Crowdsensing kann in drei Schritte unterteilt werden: der Datenerhebung, der Datensammlung und dem Datenupload [Ray+22]. Die Datenerhebung erfolgt sowohl durch die User, als auch durch „mobile sensing devices“ (z.B. Thermometer, Smartphones etc.), welche auf einem Server gesammelt werden um im Anschluss hochgeladen werden können, um sich z.B. einer Qualitätskontrolle zu unterziehen [Ray+22]. Die Vorteile des mobilen Crowdsensing erstrecken sich dabei von schneller Erhebung von vielfältigeren Daten mit erhöhter Qualität der Ergebnisse, bis hin zu akzeptableren Ergebnissen und das Auslagern von Rechenleistung von einer zentralen Plattform zu bspw. den mobilen Geräten [Ray+22]. Auf der anderen Seite kann man jedoch beobachten, dass die erhobenen Daten geprägt von Redundanzen und Duplikaten sind, was zur Folge hat, dass eine anschließende Qualitätskontrolle und die Notwendigkeit von mehr Speicherplatz unabdingbar sind [Ray+22]. Es lassen sich häufig Smartphones als Messgeräte des mobilen Crowdsensing in der Literatur finden, da diese mit einer Vielzahl an Sensoren (Temperatur-, Gyroskop-, Umgebungslichtsensoren etc.) ausgestattet sind - das mobile Crowdsensing ist aber nicht auf diese limitiert und beinhaltet auch Geräte wie mobile Wetterstationen, Bewegungssensoren, Kameras o.Ä.

Das mobile Crowdsourcing ermöglicht das Lösen einer komplexen Aufgabe durch die Auf- und Verteilung von Aufgaben an eine Gruppe von freiwilligen Nutzenden, hier über das Internet [Wan+19]. Die Komposition „Crowdsourcing“ besteht dabei aus den beiden einzelnen Begriffen *crowd* für die (Menschen-)Menge und *sourcing* für die Beschaffung (hier: von Informationen). Erstmals wurde der Begriff 2006 vom Journalisten Jeff Howe in einem Artikel verwendet, welcher das Crowdsourcing als kostengünstigere Alternative des *Outsourcing*¹ beschreibt, um Akteure aus unterschiedlichen Wissensständen und Domänen in die Softwareentwicklung miteinzubinden [How+06]. Im weiteren Verlauf der Verwendung des Begriffs spielen die Charakteristiken ebenfalls eine Rolle: Die Lösung eines Problems wird insofern erreicht, dass die Aufgaben (und unter Umständen auch Unteraufgaben, abhängig von Problemstellung) an eine (Menschen-)Menge übertragen werden, welche interessiert an der Lösung des Problems sind [Ray+22]. Weiterhin wird die Verteilung der Menge und die Existenz von variierender Menschenlogik genutzt, um Probleme an jenen Computer scheitern, zu lösen [Ray+22]. Der Unterschied zum Crowdsensing ist hierbei, dass die verrichtete Arbeit nicht auf die Interaktion mit Geräten und den entsprechenden Sensoren ist, sondern dass Arbeit im Allgemeinen verrichtet wird (vergleiche Parallele zwischen Crowdsourcing und Outsourcing). Beim Einsatz vom mobilen Crowdsensing ist vorteilhaft, dass Kosten durch nicht-benötigte Arbeitskräfte basierend auf einer expliziten Arbeitszeit verringert sind, der Einsatz einer Menge als Folge davon eine vielfältigere und qualitativ hochwertigere Datensammlung nach sich zieht und die Verfügbarkeit von vielfältigen Parametern zum Testen in Summe in eine hohe Zeitersparnis resultieren [Ray+22].

1: zu deutsch: Auslagerung, „mittel- bis langfristige Übertragung von Aufgaben der Informationsverarbeitung eines Unternehmens an ein spezialisiertes Unternehmen“ [HHR04]

Aus den Definitionen der beiden Begriffe wird deutlich, dass diese viele Gemeinsamkeiten aufweisen, was der Grund dafür ist, dass in der Umgangssprache, aber auch in der Literatur eine Verwechslung beider Begriffe auftreten kann. Aufgrund dessen ist es erforderlich, die Abgrenzung der Begriffe voneinander hervorzuheben, welche sich in der Bearbeitung der Aufgaben und der Notwendigkeit von menschlicher Intelligenz findet: Während das mobile Crowdsensing die Aufgaben auf das Erfassen von Rohdaten begrenzt, erstrecken sich Aufgaben beim Crowdsourcing über die Erfassung von Rohdaten bis zu anderen, von der Plattform zugewiesenen (allgemeinen) Aufgaben hinaus [Ray+22]. Für eben diese Aufgaben wird dementsprechend auch eine bestimmte menschliche Intelligenz vorausgesetzt, die mit der Rechenleistung zusammenarbeiten soll, um eine Lösung zu finden [Ray+22].

2.2 Verwandte Arbeiten

Der Aspekt des Crowdsensing lässt sich sowohl in der Literatur, als auch in der Praxis durch bereits existierende Projekte und Anwendungen vorfinden. Um ein grundlegendes Verständnis über das Thema und zur Definition eines theoretischen Rahmens zu schaffen, wurden diese verwandten Arbeiten herangezogen, welche in diesem Kapitel vorgestellt werden.

Bei der ersten Arbeit handelt es sich um „Crowdsensing für Bodensee Online“, ein von der ISB AG, vom Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung und der Ingenieurgesellschaft Prof. Kobus und Partner GmbH im Februar 2020 initiiertes Projekt zur Messung der Wassertemperatur des Bodensees durch Bootsbesitzer*innen mithilfe von mobilen Sensoren und Karten [Min21]. Der Fokus des Projekts liegt darin, die bereits vorhandenen Kenntnisse über die Verhältnisse am Bodensee durch einen citizen-science Ansatz zu erweitern und auf die Durchführbarkeit zu messen [IFI21]. Dabei sollte zunächst in einer Pilotphase die Funktionsweise und die Montagemöglichkeit der Sensoren vor Ort erprobt werden, um im Anschluss mit ausgereiften Sensoren in die Feldphase zu starten, in welcher die Probanden mithilfe der Sensoren mit den Messungen des Bodensees beginnen konnten [IFI21]. Aufgrund der COVID-19 Pandemie und den damit vorherrschenden Einschränkungen verliefen beide Phasen überlappend zueinander, sodass eine kontinuierliche Weiterentwicklung der Sensoren und der Nutzeroberfläche basierend auf den Erfahrungen der Nutzenden möglich gewesen wäre [IFI21]. Im Allgemeinen beschreiben die Durchführenden des Projekts die Ergebnisse als sehr positiv.

In Bezug auf die technischen Aspekte soll der Betrieb, aber auch die Verfügbarkeit der Sensoren keinen Aufwand dargestellt haben. Auch die zuvor aufgestellten Anforderungen seien im breiten Maße erfüllt worden. Lediglich der Aufwand der Standortsuche der Sensoren und die damit verbundene Auswirkung auf die Batterielaufzeit haben eine wesentliche Limitierung dargestellt - eine erste Lösung sei aber dadurch umzusetzen, dass die Sensoren als stationäre Messpunkte einzusetzen wären, um sowohl auf der einen Seite die Messdaten am Bodensee zu erhöhen, aber auf

der anderen Seite die Abtastrate des Standortes zu minimieren, um die bereits erwähnten Limitierungen aufzuheben, sodass diese eine plausible Ergänzung zu den stationären Messstationen am Bodensee darstellen [IFI21]. Der Crowdsensing-Ansatz des Projekts wird als geeignet für die Verbesserung und Verfeinerung einer bestehenden (Daten-)Struktur angesehen, da unter der Voraussetzung einer ausreichenden Qualitätssicherung und Datenmenge eine hohe Datenqualität und -dichte auf engem Raum erzielt werden kann [IFI21]. Prinzipiell ließe sich das Projekt laut Angaben der Durchführenden auf weitere (Naturschutz-)Regionen unter Zuhilfenahme weiterer Parameter (z.B. Lufttemperatur, -feuchtigkeit, -qualität) übertragen [IFI21]. Die zugrundeliegenden Charakteristiken schließen darauf, dass es sich bei „Crowdsensing für Bodensee Online“ um „opportunistic sensing“ handelt, da die Sensoren lediglich an die Boote angebracht werden müssen und keine aktive Teilnahme der Bootsbesitzer*innen notwendig ist.

2: <https://www.trackyourtinnitus.org/de/>

Aber auch in anderen Domänen ist das Crowdsensing schon zum Einsatz gekommen: Das Projekt „Track Your Tinnitus“² wurde von der Tinnitus Research Initiative (TRI) und dem Institut für Datenbanken und Informationssysteme (DBIS) der Universität Ulm für medizinische Einsatzzwecke ins Leben gerufen [Pry+17]. Etwa 10-15% der Weltbevölkerung leidet unter Tinnitus, einer Krankheit, die durch ein ständiges Ohrgeräusch gekennzeichnet ist und welche bis heute schwer behandelbar ist, da verfügbare Behandlungen nur bei einem Teil der Betroffenen anschlagen [Lan11]. Da klinische Tests und Versuche bisher den Standard repräsentiert haben, und die damit verbundene Informationsgewinnung sehr begrenzt gewesen ist, wurde das Projekt ins Leben gerufen, um eine komplementäre Vorgehensweise vorzuschlagen, die durch den Einsatz von Crowdsensing-Services die Datenerhebung vereinfacht, aber auch vervielfacht [Pry+15]. Die Nutzenden sollen dabei eine Smartphone-App installieren und nach jedem Wahrnehmen des Tinnitus Fragen zur Intensität und Belastung und daraus resultierend, die Auswirkungen auf die persönliche Stimmungslage beantworten, damit individuelle Schwankungen der Tinnituswahrnehmung erfasst werden können [Pry+15]. Einerseits soll es auf diese Weise ermöglicht werden, die behandelnden ärztlichen Fachpersonen mit den notwendigen Daten zu versorgen, um die Behandlung zu verbessern, aber auch das allgemeine Forschungsprojekt zu unterstützen und ein Verständnis für Ursachen und Wirkungen für den Tinnitus zu schaffen [Pry+15]. Die Voraussetzung der Eingabe der persönlichen Tinnituswahrnehmung charakterisiert das Projekt „Track Your Tinnitus“ als „participatory sensing“, da die Nutzenden aktiv an der Datenerhebung beteiligt sind.

3: <https://budburst.org>

Eine weitere Verwandtschaft lässt sich der App „Budburst“³ zuordnen, welche Nutzende dazu animiert, mithilfe des eigenen Mobiltelefons Pflanzenarten zu identifizieren und in einer Gemeinschaft lokale Pflanzen(-arten) kontinuierlich zu erfassen. Die Nutzenden werden durch die kontinuierliche Erfassung der Pflanzen dazu animiert, den Zustand des lokalen Ökosystems zu überwachen um eventuelle Rückschlüsse auf den Klimawandel ziehen zu können. Durch das Fotografieren von Pflanzen innerhalb der App soll ein Diskurs über die Gesundheit des Ökosystems

angeregt werden, welcher durch die Nutzenden und die daraus geformte Gemeinschaft geführt wird. Die App „Budburst“ ist ein Beispiel für „participatory sensing“, da die Nutzenden ebenfalls aktiv an der Datenerhebung beteiligt sind.

Ausgehend von den genannten Arbeiten und Projekten ist es möglich, weitere verwandte Quellen zu finden, welche sich mit dem Thema des Crowdsensings beschäftigen. Die genannten Arbeiten haben dabei gezeigt, dass das Crowdsensing in unterschiedlichen Domänen zum Einsatz kommen kann, um Problemstellungen zu lösen, welche durch traditionelle Methoden nicht oder nur schwer zu lösen sind.

3 | Konzept

In diesem Abschnitt der Arbeit geht es darum, das grundlegende Konzept hinter der Entwicklung eines interaktiven Werkzeuges zur Kuratierung von Umweltdaten aufzubauen und zu erläutern. Dabei wird als erster Aspekt der Grundgedanke dieser Arbeit geschildert, um im Anschluss die Methodologie zu erläutern, welche zum Aufstellen der Anforderungen verwendet wird, um eine Nachvollziehbarkeit und Wiederholung des Projekts gewährleisten zu können.

Im nächsten Schritt ist es dann erforderlich, die notwendigen Komponenten der Software zu erfassen, um eine Übersicht über die benötigten Technologien zu erhalten. Im Anschluss wird dann die allgemeine Vorgehensweise für die gesamte Arbeit erläutert.

3.1 Grundgedanke der Arbeit

Der Grundgedanke eines interaktiven Werkzeuges zur Kuratierung von Umweltdaten ist es, ein bereits vorhandenes System (vgl. Kapitel 4.1) insofern zu erweitern, dass die Erfahrung und Expertise von Menschen als zusätzlicher Qualitätsfilter für die vorliegenden Daten genutzt werden kann, um Anomalien und Fehler, aber auch Kausalitäten für auftretende Phänomene zu erkennen. Die Kuratierung soll dabei durch die Nutzenden des Werkzeuges auf die erhobenen Daten durch das bestehende Klimamessnetz in Bamberg erfolgen. Zusätzlich soll es möglich sein, die Sensordaten visuell darzustellen, um Verläufe und Trends zu erkennen, Vergleiche zu ziehen und Ausblicke auf zukünftige Entwicklungen zu erhalten. Die Frage, inwieweit eine solche Kuratierung umgesetzt werden kann (vgl. Kapitel 3.4), welche Auswirkungen dieser Aspekt auf die Softwareentwicklung hat und ob darauf basierend eine Sinnhaftigkeit gegeben ist (vgl. Kapitel 7), soll in dieser Arbeit beantwortet werden. Weiterhin soll diese Arbeit ein Fundament für die Entwicklung eines Crowdsensing-Werkzeuges darstellen, welches in zukünftigen Arbeiten und Projekten aufgenommen und weiterentwickelt werden kann.

3.2 Methodologie zum Aufstellen der Anforderungen

Das Institute of Electrical and Electronics Engineers (IEEE) gliedert die Anforderungsanalyse in die Anforderungserhebung (*requirements elicitation*), die Anforderungsanalyse (*requirements analysis*), die Anforderungsspezifikation (*requirements specification*) und Anforderungsbewertung (*requirements validation*) [Com+04]. An dieser Vorgehensweise orientiert sich auch die Methodologie, welche in dieser Arbeit verwendet wird.

Zum Erheben der Anforderungen ist es zunächst erforderlich, die Stakeholder zu identifizieren, da diese die Nutzenden der Software sind und somit die Anforderungen stellen. Im Falle dieser Arbeit wurden die Stakeholder bereits vor Beginn des Projekts festgelegt (siehe Kapitel 4.1.1). Nachdem die Stakeholder identifiziert worden sind, ist es möglich gewesen, diese direkt nach den Anforderungen zu befragen. Vom Bürgerverein Bamberg Mitte e.V. (BVM)¹ finden in regelmäßigen Abständen Treffen statt, in welchen zusammen mit Domänenexperte Prof. Dr. Thomas Foken Analysen über die aktuellen Auswertungen der Wetterstationen durchgeführt werden. Dabei wird unter anderem auch darüber gesprochen, wie diese Werte zustande kommen, wie sie sich entwickeln könnten und was sie bedeuten. Da an diesen Treffen in der Regel auch die anderen Stakeholder teilgenommen haben, konnten diese dazu genutzt werden, Fragen zu Anforderungen an ein interaktives Werkzeug zur Kuratierung von Umweltdaten zu erheben. Insbesondere ist es wichtig gewesen, das Ziel und den Zweck des BVM zu erfragen, da sich aus dieser Frage bereits Anforderungen an das Werkzeug ergeben und wie die Priorisierung bei der darauffolgenden Analyse zu gestalten werden muss. Zusätzlich wurden die Stakeholder dazu befragt, welche Funktionalitäten sie sich für ein solches Tool wünschen und auf welche Art und Weise sie sich vorstellen könnten, ein solches zu verwenden. Die Gespräche wurden protokolliert und im Anschluss in der Analyse (siehe Kapitel 4) ausgewertet. Der Schritt der *requirements specification* wird in 7.1 näher betrachtet.

Hier werden die aufgestellten Anforderungen in einer Spezifikation gesammelt, die idealerweise Charakteristika wie Korrektheit, Vollständigkeit, Eindeutigkeit oder einer Verifizierbarkeit aufweist [EE84]. Im Anschluss muss eine Bewertung der Anforderungen erfolgen, um sicherzustellen, dass diese auch tatsächlich die Bedürfnisse der Stakeholder erfüllen. Dazu wird ein Prototyp übergeben, mit dessen Hilfe ein erstes Hands-on stattfinden kann. Die Stakeholder werden dann nach ihrem Feedback befragt und die Anforderungen werden entsprechend angepasst. Dieser Schritt wird in Kapitel 7.3 ausführlicher beschrieben.

3.3 Erfassung der notwendigen Komponenten der Software

Dieses Unterkapitel handelt davon, die erforderlichen Komponenten zu erfassen, um im Anschluss die Architektur in Kapitel 6 aufstellen zu können.

Aufgrund der Tatsache, dass das Werkzeug im fertigen Zustand eine Webanwendung darstellen soll, welche jederzeit unabhängig vom Standort

¹: <https://www.bvm-bamberg.de/de/projektseiten/klima/>

zugänglich ist, bietet es sich an, die Grundlage der Software mit einer Programmiersprache umzusetzen, die auf eine einfache Art und Weise sowohl eine Webanwendung erstellen, als auch durch das Hinzufügen weiterer Komponenten diese Anwendung erweitern kann.

Da grundsätzlich mit Daten gearbeitet werden soll (in Form von Sensordaten, aber auch notwendige Daten zur Funktionsweise einer Applikation, wie z.B. Benutzerdaten) spielt die Verwaltung dieser Daten eine große Rolle. Dieser Aspekt ist insofern wichtig, da die Daten sowohl von der Webanwendung angelegt (z.B. Login-Daten, Feedback der Nutzenden), als auch visualisiert werden müssen (z.B. Historie der Sensordaten einer Wetterstation). Dementsprechend ist eine Konnektivität zwischen diesen beiden Komponenten essenziell für die Funktionsweise der Software.

Zur grundlegenden Verwaltung der Komponenten, der Umgebung, der Daten und der Projektplanung hat sich GitHub² als hilfreich erwiesen: Das Anlegen eines GitHub-Repositories ist kostenlos und kann ohne Einschränkungen geteilt werden. Durch das Anlegen eines Projekts³ können Tickets angelegt werden, die als Aufgaben fungieren und in einem Issue-Board in verschiedene Abschnitte unterteilt werden können, basierend auf dem aktuellen Status des Tickets (hier: *TO-DO*, *In Progress*, *Needs Review*, *Done*, *Future Work*). In den Tickets selbst wird die Problembeschreibung bzw. Aufgabenstellung, die Lösung und Verantwortliche für die Bearbeitung festgehalten. Zusätzlich können durch Kommentare weitere Informationen festgehalten werden, die für die Bearbeitung des Tickets relevant sind (z.B. Bearbeitungsdauer, Lösungen oder Hinweise bei Wechsel der Verantwortlichen).

Die Versionsverwaltung mit Git⁴ bietet dabei die Möglichkeit, geschriebenen Code in einem Repository zu speichern und bei Bedarf auf einen früheren Stand zurückzugreifen. Dies ist insbesondere dann hilfreich, wenn Änderungen am Code vorgenommen werden, die sich als fehlerhaft herausstellen und somit die Funktionsweise der Software beeinträchtigen. Durch die Versionsverwaltung ist es möglich, den Code auf einen früheren Stand zurückzusetzen und die Änderungen zu verwerfen.

2: <https://github.com>

3: <https://github.com/users/SamAkc/projects/2>

4: <https://git-scm.com/>

3.4 Umsetzung

Um eine Nachvollziehbarkeit dieser Arbeit gewährleisten zu können, soll in diesem Abschnitt erläutert werden, wie die grundsätzliche Arbeit umgesetzt wurde.

Aufgrund der zeitlichen Limitierung dieser Arbeit ist die Anwendung von Ansätzen aus der agilen Softwareentwicklung notwendig gewesen: Diese beschreiben einen Prozess, der durch die Flexibilität der Software, was durch das Aufteilen der Hauptaufgabe in einfacher zu lösende Teilprozesse erreicht wird und den kontinuierlichen Austausch mit den Stakeholdern einen schnelleren Einsatz der Software ermöglicht und die Minimierung von Risiken im Entwicklungsprozess erlaubt [Sie18]. Eine sehr beliebte Methode ist hier die iterative Entwicklung, die sich insbesondere dann anbietet, wenn die Projektziele und der Zeitplan klar definiert sind, die Vorgehensweise, diese zu erreichen, aber noch unklar ist [SA07]. Durch den iterativen Ansatz ist es zudem auch möglich, die Software

schnell und adaptiv auf die Stakeholder ausrichten zu können, sowohl die höhere Kundenzufriedenheit, als auch eine geringere Defekt-Rate zu erzielen, da eine kontinuierliche face-to-face-Kommunikation zwischen Entwicklung und Stakeholder als primärer Wissensaustausch erfolgt [SAo7]. Der Verbesserungsprozess mithilfe von Iterationen wird in der Literatur in folgende Schritte unterteilt, auf welche in den nachfolgenden Unterkapiteln im Detail eingegangen wird: Vorbereitung, Sammeln von Erfahrungen, Planung von Verbesserungsschritten, Piloting, Follow-up und Validierung und Aufbewahrung [SAo7].

3.4.1 Vorbereitung

In der Vorbereitungsphase geht es darum, die Frequenz und den Zeitplan der sogenannten Post-Iteration Workshop (PIW)s⁵ festzulegen [SAo7]. Da diese Arbeit von einem einzigen Entwickler für die Stakeholder erfolgt ist, wurden die PIWs im zweiwöchigen Takt durchgeführt. Diese haben sich so gestaltet, dass aufgetretene Probleme auf einem Issue-Board in einem GitHub-Repository⁶ gesammelt wurden und nach Ende jeder Iteration explizit Lösungen für jene gesucht wurden, statt diese direkt zu beheben. So ist es möglich gewesen, die Arbeitszeit in eine effektive Arbeitsphase und Evaluierungsphase zu unterteilen. Sollte ein Problem nicht gelöst werden können, so wurden Ansprechpartner*innen gesucht, die bei der Lösung behilflich sein konnten — in der Regel waren dies die Stakeholder.

5: Eine PIW entspricht einer Review im Team nach jeder erfolgten Iteration um über aufgetretene Hindernisse und Probleme zu diskutieren und Lösungsansätze zu schaffen [SAo7]

6: <https://github.com/SamAkc/MasterThesis>

3.4.2 Sammeln von Erfahrungen

Diese Phase dient dazu, zum ersten PIW die Prozessverbesserungen auf die vom Softwareentwicklungsteam identifizierten Hindernisse und Probleme aufzubauen [SAo7]. Hier werden positive oder negative Erfahrungen, die bei der Softwareentwicklung aufgetreten sind, gesammelt und im Team analysiert, um in späteren Iterationen diese Erfahrungen zu berücksichtigen und einfachere Problemlösungen zu schaffen. Da diese Phase ein Team voraussetzt, konnte dieser Teil der Arbeit nur limitiert durchgeführt werden: Die positiven Erfahrungen in Form von Lösungen für aufgetretene Probleme wurden festgehalten, sodass diese für eventuell auftretende negative Erfahrungen in zukünftigen Iterationen als Lösungsansatz verwendet werden konnten. Negative Erfahrungen hingegen wurden nach erfolgter Identifikation festgehalten, um diese im nächsten Schritt weiter verarbeiten zu können.

Für negative Erfahrungen hingegen wurden zusammen mit Ansprechpartnern (in Form von Betreuenden, aber auch Domänenexperten mit Hintergrundwissen, wie z.B. Softwareentwickler in Unternehmen, welche mit identischen/ähnlichen Komponenten arbeiten) Lösungsansätze gesucht, um diese zu lösen oder gar in zukünftigen Iterationen zu vermeiden.

3.4.3 Planung von Verbesserungsschritten

Für die im vorherigen Abschnitt der Arbeit genannten negativen Erfahrungen sollen in diesem Schritt Ansätze zur Verbesserung abgeleitet werden. Dabei werden die Hindernisse und Probleme aus der vorherigen Iteration in einer sogenannten „Problem Area“ gesammelt und aus dem Hintergrundwissen von den Betreuenden dieser Arbeit, aber auch von Domänenexperten (z.B. Softwareentwickler*innen in einem Unternehmen, welches mit identischen/ähnlichen Komponenten arbeitet) nützliche Verbesserungsmöglichkeiten formuliert [SA07].

3.4.4 Piloting, Follow-up und Validierung der Verbesserungsschritte

Die vorher abgeleiteten Ansätze zur Verbesserung werden in der Phase des Pilotings umgesetzt. Neben der Umsetzung ist die Sammlung von Messdaten erforderlich, welche zu Validierungszwecken in den nächsten Phasen verwendet werden [SA07].

In dieser Arbeit wurde die Umsetzung der Verbesserungsschritte in der Regel in der nächsten Iteration umgesetzt, da die Umsetzung der Verbesserungsschritte in der Regel nicht viel Zeit in Anspruch genommen hat. Zum Zeitpunkt dieser Arbeit ist es nicht möglich gewesen, Messdaten neben der Umsetzung zu sammeln, da diese erst nach der Umsetzung der Verbesserungsschritte generiert werden konnten und zum jetzigen Zeitpunkt noch nicht ausreichend Daten vorhanden sind, um diese für Validierungszwecke zu verwenden. Aus diesem Grund hat auch kein Follow-up der Verbesserungsschritte stattgefunden, da es für diese erforderlich ist, die Messdaten als Grundlage zur Evaluierung der erfolgreichen Anwendung der Verbesserungsschritte zu verwenden [SA07].

3.4.5 Aufbewahrung der Verbesserungsschritte

In der Literatur wird empfohlen, die Ergebnisse der PIWs systematisch in Dokumenten oder Formularen zu sammeln, um diese in zukünftigen Iterationen (für Follow-ups und Validierungen) verwenden zu können [SA07]. Da diese Vorgehensweise aber den Rahmen dieser Arbeit zeitlich gesprengt hätte, wurden die Ergebnisse der PIWs im GitHub-Repository⁷ gesammelt. Eine beispielhafte negative Erfahrung in dieser Arbeit hat sich in der Schnittstelle zweier Komponenten gefunden (vgl. Kapitel 6): Diese haben isoliert voneinander ordnungsgemäß funktioniert, sind aber nicht in der Lage gewesen, miteinander zu kommunizieren um den Austausch von Daten zu ermöglichen. Die Lösung für dieses Problem wurde von einem Domänenexperten vorgeschlagen, der bereits Erfahrung mit der Verwendung dieser Komponenten hatte und somit eine funktionierende Lösung für dieses Problem anbieten konnte — diese Lösung, zusammen mit anderen Lösungen für aufgetretene Probleme und damit, negative Erfahrungen, wurde in einer „Solution Area“ (vgl. Abschnitt „Troubleshooting“ in README-Datei in GitHub-Repository) festgehalten, um diese in zukünftigen Iterationen verwenden zu können.

7: <https://github.com/SamAkc/MasterThesis>

4 | Anforderungsanalyse

Bevor mit der eigentlichen Implementierung begonnen werden kann, ist es notwendig, die Anforderungsanalyse. Der erste Schritt hierzu besteht daraus, die Ausgangslage in Bamberg zu untersuchen, um darauf aufbauend die Stakeholder zu identifizieren, da diese den Ursprung der Anforderungen darstellen. Nachdem das Aufstellen der Anforderungen erfolgreich gewesen ist, können diese im Anschluss analysiert, also priorisiert, auf ihre Validität und im Anschluss auf ihre Realisierbarkeit überprüft werden. Basierend auf den Anforderungen werden dann Use Cases formuliert, die die Anforderungen in einer konkreten Situation beschreiben und damit die Grundlage für die Implementierung bilden.

4.1 Ausgangslage in Bamberg

Bamberg¹ ist eine kreisfreie Stadt in Bayern im Regierungsbezirk Oberfranken mit einer Einwohnerzahl von knapp 80.000 Einwohnern (Stand: Dezember 2022). Die Stadt selbst kann dabei in sogenannte Local Climate Zones (LCZ), also Klassifikationen, welche sich durch die logische Aufteilung einer Landschaft ergeben [SO11], unterteilt werden. Zusammen mit dem Domänenwissen des Meteorologen Prof. Dr. Thomas Foken (vgl. Kapitel 4.1.1) hat eine rudimentäre Unterteilung in sechs LCZ stattgefunden: Die Innenstadt, die ERBA, Gartenstadt, Bamberg-Ost, der Hain, am Laubanger und das Berggebiet (siehe Abbildung 4.1). Die Entscheidung über die Grenzen und der Gebiete erfolgt dabei durch die in der Literatur gegebenen Charakteristika, welche zum Definieren von LCZ notwendig sind [Okeo4; SO12]:

1: <https://www.stadt.bamberg.de/Unsere-Stadt/Stadtinfo/>

Innenstadt (2) – Die Innenstadt kann als „Compact midrise“ klassifiziert werden. Charakteristisch für diese LCZ ist der dichte Mix aus mittelhohen Gebäuden, das Vorhandensein von wenigen bis keinen Bäumen und hauptsächlich asphaltierten Landflächen. Überwiegend lassen sich Stein, Ziegel, Kacheln und Beton als Baumaterialien vorfinden.

ERBA (6) – Die ERBA kann als „Open low-rise“ klassifiziert werden. In einem solchen befinden sich in der Regel flache Gebäude, die in einer offeneren Struktur angeordnet sind. Die Landflächen sind asphaltiert, weisen aber auch Grünflächen in Form von kleinen Pflanzen oder verstreuten Bäumen auf. Die Baumaterialien sind ähnlich wie in der Innenstadt, jedoch kann man an der ERBA häufig Holz als Baumaterial vorfinden.

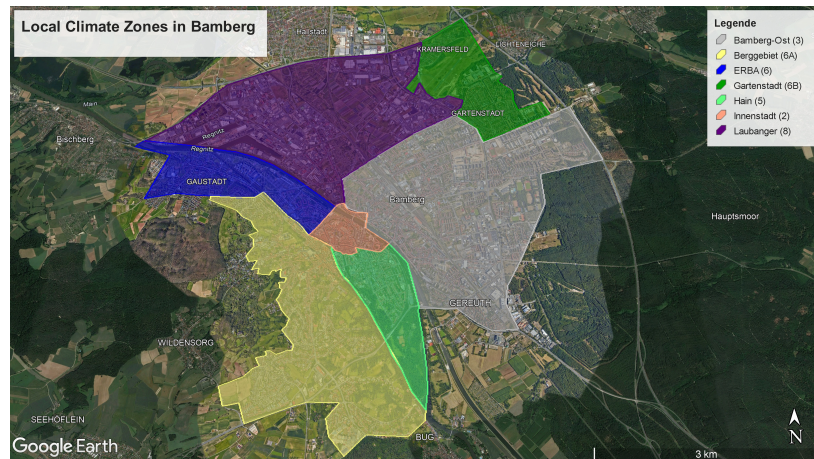


Abbildung 4.1: Die sechs LCZ in Bamberg (eigene Darstellung, basierend auf [Okeo4; SO12])

2: Nach [SO12]: Klassifikationen zu saisonalen Eigenschaften, wie z.B. schneebedeckter Boden, trockener/feuchter Boden etc.

Gartenstadt (6B) – Die Gartenstadt ist wie die ERBA ein „Open low-rise“. Allerdings kann hier der Zusatz „B“ als „Land cover type“² zur Klassifikation hinzugefügt werden, welche nach [SO12] bedeutet, dass die Landschaft rege mit immergrünen Bäumen bestückt ist. Die Landflächen weisen zu einem Großteil Grünflächen mit kleinen Pflanzen auf. Diese Arten von LCZ werden in der Regel für Wälder oder Stadtparks verwendet.

Bamberg-Ost (3) – Der Osten Bambergs wird als „Compact low-rise“ klassifiziert. Dieser weist die identischen Charakteristika wie die Bamberger Innenstadt auf, mit dem einzigen Unterschied, dass es sich bei den Gebäuden um niedrige Gebäude handelt.

Hain (5) – Der Hain befindet sich in der Klasse des „Open midrise“. Dieser weist die identischen Charakteristika wie die ERBA auf, mit dem einzigen Unterschied, dass es sich bei den Gebäuden um mittelhohe Gebäude handelt und statt Holz überwiegend Beton und Stahl als Baumaterialien eingesetzt werden.

Laubanger (8) – Der Laubanger wird in Bamberg als „Large low-rise“ eingestuft. In dieser Klasse lassen sich in der Regel große, aber flache Gebäude in einer offeneren Struktur vorfinden. Bäume oder Grünflächen sind meistens nicht bis kaum vorhanden, da die Straßen und Wege überwiegend asphaltiert sind. Als Baumaterialien kommen häufig Stahl, Beton und Stein zum Einsatz.

Berggebiet (6A) – Das Bamberger Berggebiet ähnlich wie die Gartenstadt und ERBA als „Open low-rise“ eingestuft und weist die identischen Charakteristika auf. Der Zusatz „A“ besagt hierbei, dass eine große bzw. dichte Menge an Bäumen in der Landschaft vorzufinden ist, was in der Regel auch für Wälder oder Stadtparks zutrifft.

Durch die Unterteilung der Stadt Bamberg in LCZ ist es möglich, die Auslesungen der Wetterstationen und die Unterschiede der Messwerte in den einzelnen Gebieten nachvollziehen und damit besser analysieren zu können.

In der Stadt selbst sind Wetterstationen³ verteilt, die jeweils die Temperatur und Luftfeuchtigkeit am entsprechenden Standort messen. Auf der Wetterkarte von Netatmo⁴ sind diese Auslesungen von öffentlichen Netatmo-Wetterstationen einsehbar.

3: Die genaue Anzahl der Sensoren ist unbekannt, da nicht alle öffentlich einsehbar sind (Stand Oktober 2023: 42 Wetterstationen einsehbar)

4: <https://weathermap.netatmo.com/>

4.1.1 Identifikation der Stakeholder

In diesem Abschnitt der Arbeit werden die Stakeholder identifiziert, dessen Bedürfnisse und Ansprüche durch die Entwicklung des interaktiven Werkzeuges befriedigt werden sollen. Diese werden im Folgenden vorgestellt:

Der Bürgerverein Bamberg Mitte e.V. Bei dem BVM handelt es sich um einen Bürgerverein in der Stadt Bamberg, welcher 1905 gegründet wurde und mit zahlreichen Projekten⁵ und durch das Praktizieren von Bürgerbeteiligung die Stadt Bamberg, vor allem in Hinblick auf die Stadtplanung und des Umwelt- und Denkmalschutzes, mitgestaltet⁶.

5: <https://bvm-bamberg.de/de/projektseiten/projekte-und-aktivitaeten/>

6: <https://bvm-bamberg.de/de/buergerverein/ueser-verein/>

Eines dieser Projekte ist das „Klimamessnetz auf der Bamberger Inselstadt“. Hier wird seit Anfang 2022 mithilfe von insgesamt neun smarten Netatmo-Wetterstationen⁷ (Stand: 21. Oktober 2023) ein Klimamessnetz im Bamberger Zentrum zum Messen der Temperatur und Luftfeuchtigkeit an den entsprechenden Standorten⁸ aufgebaut.

7: <https://shop.netatmo.com/de-de/weather/smart-weather-station/weatherstation>

Aus den Messungen der Wetterstationen ergibt sich, dass sich die Bamberger Innenstadt im Vergleich zu anderen LCZ kaum abkühlt und die Temperaturen sich häufig von der offiziellen Station des Deutschen Wetterdienst (DWD) unterscheiden (vgl. Webauftritt⁹ des BVM). Dies ist auch darauf zurückzuführen, dass das Fehlen von Bäumen und Grünflächen als Charakteristika eines „Compact mid-rise“ ausschlaggebend für das Aufwärmen der Innenstadt ist: Ein Team von Geoökologen der Universität ETH Zürich hat in einer Studie gezeigt, dass Grünanlagen mit Bäumen in Städten zu einem Kühlungseffekt führen [Bas+19]. Dieser Effekt entsteht durch die Tatsache, dass Bäume durch ihre Wurzeln mehr Wasser aufnehmen können, welches im Umkehrschluss an trockenen und heißen Tagen verdunsten und so zu einer Abkühlung führen kann [Bas+19].

8: Standorte der smarten Wetterstationen (LCZ): Steinertstraße (Innenstadt), Lange Straße (Innenstadt), Am Weidenufer (ERBA), Holzmarkt (Innenstadt), Promenadestraße (Innenstadt), Wetzelsstraße (ERBA), Fischerei (Innenstadt), Frauenstraße (Innenstadt) und Hainstraße (Hain)

9: <https://bvm-bamberg.de/de/projektseiten/klima/>

Wie in Kapitel 3.2 bereits erläutert, trifft sich der BVM in regelmäßigen Abständen zum Analysieren der Auslesungen der Wetterstationen und Diskutieren von Maßnahmen, die zur Abkühlung der Innenstadt beitragen können. Durch die Teilnahme an diesen Treffen und die daraus resultierenden Gespräche mit den Mitgliedern des BVM wurde in Erfahrung gebracht, dass das grundlegende Ziel des Vereins ist, Führungspersonen (z.B. in der Politik, Stadtplanung, Ämtern etc.) durch die Auslesungen der Wetterstationen zu sensibilisieren und damit zu einer Verbesserung der Lebensqualität durch das Erreichen des Kühlungseffektes in der Stadt Bamberg beizutragen. Dies soll ihrer Ansicht nach durch das Reduzieren von Parkplätzen (zum Minimieren von stehen gelassenen Autos) und Straßen für Autos erreicht werden, sodass für die neu geschaffenen Flächen Bäume und Grünflächen gepflanzt werden können.

Die Domänenexpert*innen Als Domänenexpert*innen werden in dieser Arbeit Personen definiert, die durch ihre Expertise und Erfahrung

10: <https://www.micrometeorology.de/>

11: Nächte, in denen die minimale Temperatur zwischen 18 und 6 Uhr UTC mindestens 20°C betragen hat [Deu]

in einer Domäne wichtige Einblicke und Informationen, die über die Literatur hinausgehen, zu einem Projekt beitragen können. Sowohl für dieses Projekt, als auch für das Projekt des BVM fungiert hier Prof. Dr. Thomas Foken¹⁰, der durch seine Professur für Mikrometeorologie an der Universität Bayreuth langjährige Erfahrung in der (Mikro-)Meteorologie und der Klimaforschung besitzt. Durch die Teilnahme Prof. Dr. Fokens an den regelmäßigen Treffen mit dem BVM ist es möglich gewesen, Erläuterungen und Definition der (Mikro-)Meteorologie zu erhalten, sodass erste Fokusse der Arbeit gesetzt werden konnten:

Einer dieser Punkte sind die sogenannten „Tropennächte“¹¹ in Deutschland gewesen, die den Analysen des Professors nach eine deutliche Steigung in den letzten Jahren aufweisen. Dieser Aspekt ist insofern gefährlich für den Menschen, da der Körper des Menschen durch eine solche Hitzebelastung auf eine besondere Weise beansprucht wird und Probleme, die sich auf das Herz-Kreislaufsystem des Menschen auswirken, auftreten können [Umw23] Tropennächte bieten sich insofern als guter Messwert an, da diese Kennzeichen für Hitzewellen in relativ kühleren Regionen der Welt (*hier*: Deutschland) und für ihre geringe Frequenz eigentlich eine Ausnahmeerscheinung repräsentieren sollten TODO HIER ZITAT. [...]

12: <https://www.uni-bamberg.de/mobi/>

Der Lehrstuhl für Informatik, insbesondere Mobile Softwaresysteme/Mobilität der Universität Bamberg (MOBI-Lehrstuhl)

Der MOBI-Lehrstuhl¹² ist ein Lehrstuhl, welcher 2014 in der Fakultät Wirtschaftsinformatik und Angewandte Informatik an der Otto-Friedrich-Universität Bamberg angelegt wurde. Die Leitung übernimmt Prof. Dr. Daniela Nicklas, welche gleichzeitig die Betreuerin dieser Arbeit darstellt. Der Lehrstuhl beschäftigt sich insbesondere mit den „Fragen des Datenmanagements für mobile Systeme, Datenstrommanagement/komplexe Ereignisverarbeitung und der Unterstützung sensorbasierter Anwendungen, unter anderem im Bereich Smart Cities“. Die Hauptansprechpartner*innen im Rahmen dieser Arbeit sind Prof. Dr. Daniela Nicklas, Leonie Ackermann und Aboubakr El Hacen Benabbas gewesen.

Der MOBI-Lehrstuhl hat ebenfalls an den regelmäßigen Treffen des BVM zusammen mit Prof. Dr. Thomas Foken teilgenommen, sodass hier zusätzlich die Möglichkeit bestanden hat, Erfahrungen und Wissen aus dem Bereich der Informatik, der Datenverarbeitung und notwendige Schritte zum Entwickeln eines interaktiven Werkzeuges zu erhalten. Im gemeinsamen Austausch wurden Anforderungen ausgearbeitet, die sowohl für das BVM und die Domänenexpert*innen in Hinblick auf die Auslesungen der Wetterstationen und die damit verbundenen Analysen der Sensordaten, als auch für den MOBI-Lehrstuhl in Hinblick auf die technische Umsetzung relevant, aber auch realistisch sind.

Weitere Stakeholder Da eine Stadt selbst nicht als Stakeholder identifiziert werden kann, weil eine solche keine eigene Entscheidungs- bzw. Handlungseinheit darstellt, sollte Bamberg selbst trotzdem in diesem Abschnitt erwähnt werden. Dies ist darauf zurückzuführen, dass im Idealfall die eingesetzte Software Regierungsbehörden, Verwaltungen oder ähnliche Organisationen der Stadt erreichen kann und im Idealfall die

Auswirkungen dieser Sache Bürger*innen, Unternehmen oder andere Interessensgruppen innerhalb der Stadt betreffen. Allerdings hat zum Zeitpunkt der Anforderungsanalyse weder mit der Stadt Bamberg, noch mit den Bürger*innen ein Austausch stattgefunden, sodass diese als indirekter Stakeholder erwähnt, aber weiterführend in dieser Arbeit nicht betrachtet werden.

Die in diesem Kapitel erläuterten Stakeholder wurden zum Zeitpunkt der Anforderungsanalyse identifiziert. Eine Limitierung auf diese Stakeholder ist aber nicht gegeben, sodass im Laufe der Arbeit oder in der Zukunft weitere Stakeholder identifiziert und in die Anforderungsanalyse aufgenommen werden können.

4.2 Die Netatmo API — eine Schnittstelle zwischen Sensordaten und Stakeholder

4.3 Definition der Anforderungen

4.4 Use Cases

5 | User Stories

6 Implementierung und Design

In diesem Kapitel der Arbeit wird die Implementierung auf Basis der aufgestellten Anforderungen in Kapitel 4 betrachtet. Die Verzeichnisstruktur, auf welche in den folgenden Kapiteln eingegangen wird, ist als Baumstruktur mit Abbildung 6.2 gegeben. Die gesamte Applikation befindet sich auf dem Repository im Verzeichnis „app“. Da die allgemeine Architektur (vgl. Abbildung ??) mithilfe der Containervirtualisierung über Docker aufgebaut ist, wird zunächst dieser Aspekt erläutert. Im Anschluss findet jeweils eine Unterscheidung zwischen der Architektur für das Front-End und das Back-End statt.

6.1 Containervirtualisierung mit Docker

Aufgrund der Tatsache, dass ein privater Entwickler ein Produktivsystem mit mehreren Komponenten simulieren muss, hat sich der Einsatz Dockers als sinnvoll erwiesen. Docker ist eine Open-Source-Software, die es ermöglicht, Anwendungen mithilfe von Containervirtualisierung zu isolieren. Dabei wird die Anwendung in einem Container ausgeführt, der alle Abhängigkeiten enthält, die für die Ausführung der Anwendung erforderlich sind. Die Container sind dabei leichtgewichtiger als virtuelle Maschinen, da sie den Kernel des Host-Betriebssystems nutzen¹. Für diese Arbeit sind drei Anwendungen notwendig: eine Python-Flask Instanz für einen Webserver (Port 5000), eine PostgreSQL-Instanz für die Datenbank (Port 5432) und eine Grafana-Instanz (Port 3000) für die Verarbeitung und Visualisierung der Daten aus der Datenbank. Diese drei Anwendungen werden jeweils in einem eigenen Container ausgeführt und einem Netzwerk zugewiesen, um eine Kommunikation zwischen den Containern zu ermöglichen (vgl. Abbildung 6.3). Diese notwendigen Eigenschaften werden in einer `docker-compose.yml`-Datei definiert. Diese Datei wird von Docker verwendet, um die Container zu erstellen und zu starten. Die `docker-compose.yml`-Datei befindet sich im Hauptverzeichnis der Applikation.

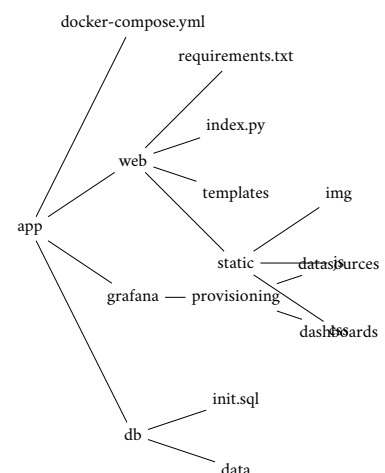


Abbildung 6.2: Die Verzeichnisstruktur der Applikation

1: <https://www.docker.com/resources/what-container/>

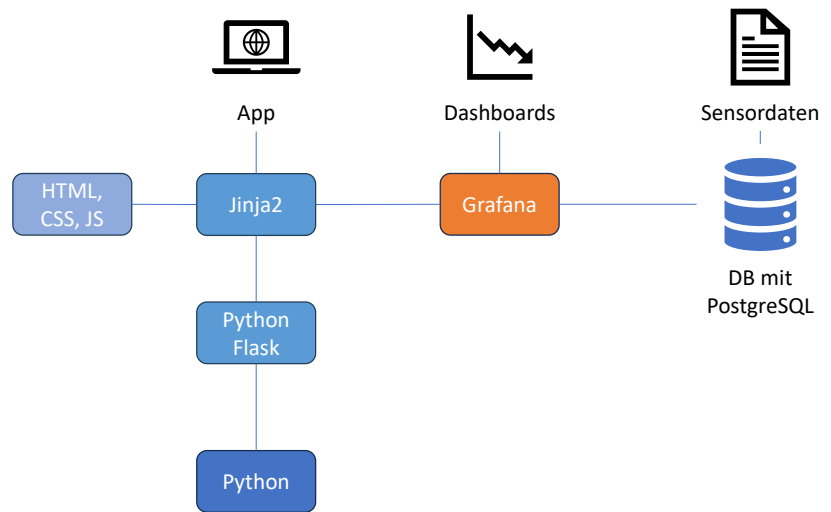


Abbildung 6.1: Die vorliegende Architektur für die Applikation

6.2 Architektur für das Front-End

Das Front-End der Applikation ist mit dem Web-Framework Python-Flask² und den daraus resultierenden Komponenten umgesetzt. Die Wahl, welche Programmiersprache zur Implementierung des Werkzeugs zum Einsatz kommen soll, ist auf Python³ gefallen, da es sich um eine leichtgewichtige und einfach zu erlernende Sprache handelt. Zudem ist Python eine der beliebtesten Programmiersprachen⁴ und bietet daher eine umfangreiche Dokumentation zum Umsetzen der individuellen Anforderungen. Flask verwendet als Template-Engine Jinja2⁵, mit dessen Hilfe HTML- und JavaScript-Code, als auch CSS-Stylesheets eingebunden werden können. Alle Webserver-Komponenten befinden sich im Verzeichnis „web“. Innerhalb dieses Verzeichnisses ist folgende Unterteilung vorzufinden:

- ▶ **templates** für alle HTML-Dateien, die als Templates für die Jinja2-Engine dienen. In diesem Verzeichnis befindet sich die `base.html`, welche die Grundlage für alle anderen HTML-Dateien darstellt. In dieser Datei werden alle notwendigen Komponenten wie Stylesheets, JavaScript-Code und die Navigation eingebunden. Die anderen HTML-Dateien erben von dieser Datei und können somit auf die Komponenten zugreifen. TODO: VERWEIS AUF ANHANG
- ▶ **static** für alle statischen Dateien, wie CSS-Stylesheets und Bilder (z.B. Favicons für Logos). Auch JavaScript-Dateien werden hier abgelegt um Unterverzeichnisse simpler zu gestalten, obwohl es sich hier um eine dynamische Programmiersprache handelt.
- ▶ **index.py** für die Python-Flask-Instanz, die den Webserver darstellt.

2: <https://flask.palletsprojects.com/>

3: <https://www.python.org/>

4: <https://spectrum.ieee.org/top-programming-languages-2022>

5: <https://jinja.palletsprojects.com/>

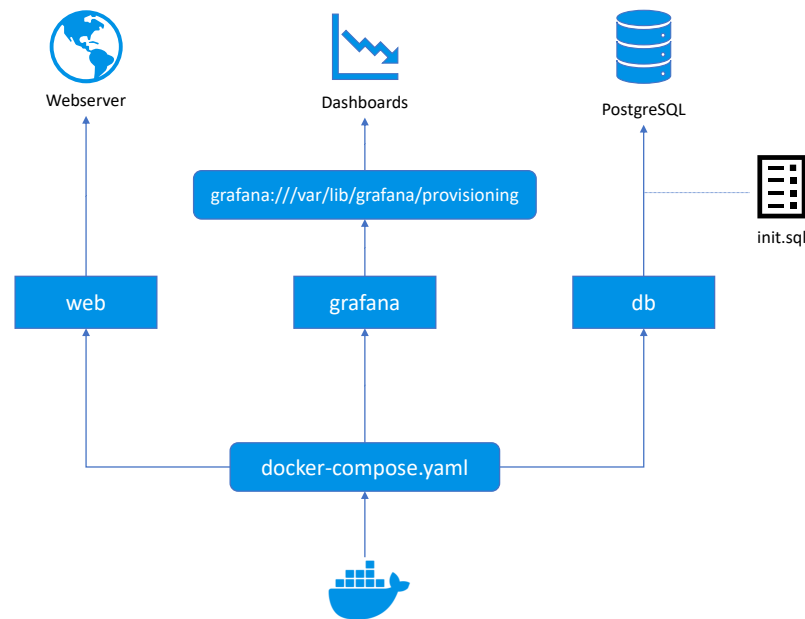


Abbildung 6.3: Die vorliegende Docker-Architektur für die Applikation

In dieser Datei werden alle Routen definiert, die der Webserver bereitstellen soll. Das Session- und Cookie-Handling, HTTP-Requests und -Responses und das Logging können ebenfalls in dieser Datei definiert werden.

- **requirements.txt** für alle Python-Abhängigkeiten, die für die Ausführung der Applikation notwendig sind. Diese Datei wird von Docker verwendet, um die Abhängigkeiten zu installieren.

Zum Visualisieren und Interagieren der Daten kommt die Open-Source-Software Grafana⁶ zum Einsatz. Mithilfe von Grafana lassen sich Daten aus einer Datenbank abrufen (vgl. 13), individuell visualisieren und in Dashboards einpflegen (vgl. Abbildung 6.4). Zu Visualisierungszwecken befinden sich acht Dashboards in der Sammlung der Bamberger Wetterstationen, wobei es sich bei einem Dashboard um die Zusammenlegung der anderen sieben Wetterstationen handelt, um einen Vergleich zwischen diesen zu ermöglichen. Der Fokus liegt dabei auf der Visualisierung der Temperatur zu einer gegebenen Uhrzeit für jede Wetterstation. Mithilfe von Grenzwerten, die in den Dashboards definiert werden können, lassen sich Temperaturbereiche farblich⁷ hervorheben. Der Aspekt des Crowdsensings wird nativ von Grafana durch die Möglichkeit, einen Punkt auf dem Graphen zu markieren und eine Annotation an diese Markierung hinzuzufügen, unterstützt. Diese Annotationen sollen dann durch andere Nutzer*innen des Werkzeuges eingesehen werden können.

Das Front-End bietet mehrere Seiten an, zwischen jenen mithilfe des Routings navigiert werden kann:

- **Home** für die Startseite der Applikation. Hier werden alle Wetterstationen und die definierten LCZ auf einer OpenStreetMap⁸-Karte

6: <https://grafana.com/>

7: Die Farbe Blau soll repräsentativ für kalte Temperaturen sein, während die Farbe Rot hohe Temperaturen symbolisieren soll

8: <https://www.openstreetmap.org/>

9: HTML-Tag zum Einbetten von anderen Dokumenten in eine HTML-Datei

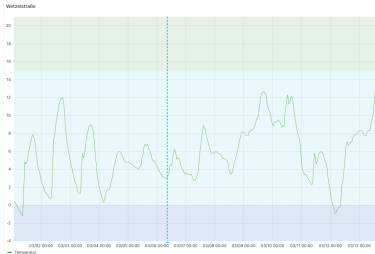


Abbildung 6.4: Repräsentatives Grafana-Dashboard mit den Sensordaten aus der Wetzelsstraße in Bamberg

10: <https://www.postgresql.org/>

11: Zwischen dem 01. März 2023 und 10. August 2023

12: Comma-separated values, häufiges Dateiformat zum Austausch von Daten

eingebildet. Auf der rechten Seite haben die Nutzer*innen mithilfe eines Chatfensters die Möglichkeit, sich mit anderen Nutzer*innen zu den Daten auf der Karte auszutauschen.

- ▶ **Sensorinspektor** für die Analyse der Daten aus den Bamberger Wetterstationen. Auf dieser Seite werden die Daten aus Grafana mithilfe von iFrames⁹ eingebunden und durch Filtermöglichkeiten (zu Wetterstationen, Datum und Uhrzeit) mithilfe von Dropdown-Menüs eingeschränkt.
- ▶ **Feedback** für die Möglichkeit, Feedback zu der Applikation zu geben. Hier können die Nutzer*innen eine Nachricht hinterlassen und dieser eine Kategorie zuweisen (Fehlermeldung, Änderungsvorschlag oder neues Feature).
- ▶ **User** für die Benutzerverwaltung. Hier ist es möglich, sich zu registrieren und anzumelden. Nach der Anmeldung können die Nutzer*innen ihr Profil, ihre Chats und ihr Feedback einsehen.

6.3 Architektur für das Back-End

Das Back-End der Applikation ist mit PostgreSQL¹⁰ als relationales Datenbankmanagementsystem umgesetzt. Die Wahl, welche Datenbank zum Einsatz kommen soll, ist auf PostgreSQL gefallen, da es sich um Open-Source-Software handelt, eine einfache Einbindung in Grafana möglich ist und PostgreSQL eine umfangreiche Dokumentation zur Verfügung stellt. Zusätzlich entspricht PostgreSQL weitestgehend dem SQL-Standard.

Eine Stichprobe¹¹ der Sensordaten der Wetterstationen in Bamberg wird durch eine parallel laufende Abschlussarbeit am MOBI-Lehrstuhl in Form von CSV¹²-Dateien zur Verfügung gestellt. Diese Daten werden von Docker Compose beim Starten der Container aus dem Verzeichnis db/data in die Datenbank importiert und durch SQL-Befehle in der Datei db/init.sql in die Datenbank eingefügt (vgl. Abbildung 6.1).

Im Anschluss liegt die Tabelle stations in der Datenbank mit folgenden Spalten vor:

- ▶ **id** ist notwendig als Primary Key, um die Daten eindeutig zu identifizieren.
- ▶ **p_id** als ID der Wetterstation.
- ▶ **time** als Zeitstempel.
- ▶ **station** als Name der Wetterstation.
- ▶ **ta** als Temperatur in Grad Celsius.
- ▶ **lon** als Längengrad.
- ▶ **lat** als Breitengrad.
- ▶ **z** als Höhe über dem Meeresspiegel.
- ▶ **humidity** als Luftfeuchtigkeit in %.

Listing 6.1: SQL-Befehle zum Erstellen der Tabelle stations in der Datenbank

```

1  CREATE TABLE stations (
2      id SERIAL PRIMARY KEY,
3      p_id INT,
4      time TIMESTAMP,
5      station VARCHAR(50),
6      ta FLOAT,
7      lon FLOAT,
8      lat FLOAT,
9      z INT,
10     humidity FLOAT
11 );
12
13 COPY stations(p_id, time, station, ta, lon, lat, z, humidity) FROM '/data↵
    /Bamberg_Stations 01.03.2023to31.03.2023.csv' DELIMITER ';' CSV ↵
    HEADER;
14 COPY stations(p_id, time, station, ta, lon, lat, z, humidity) FROM '/data↵
    /Bamberg_Stations 01.04.2023to30.04.2023.csv' DELIMITER ';' CSV ↵
    HEADER;
15 COPY stations(p_id, time, station, ta, lon, lat, z, humidity) FROM '/data↵
    /Bamberg_Stations 01.06.2023to30.06.2023.csv' DELIMITER ';' CSV ↵
    HEADER;
16 COPY stations(p_id, time, station, ta, lon, lat, z, humidity) FROM '/data↵
    /Bamberg_Stations 01.07.2023to31.07.2023.csv' DELIMITER ';' CSV ↵
    HEADER;
17 COPY stations(p_id, time, station, ta, lon, lat, z, humidity) FROM '/data↵
    /Bamberg_Stations 01.08.2023to10.08.2023.csv' DELIMITER ';' CSV ↵
    HEADER;

```

Mit dieser Struktur ist es nun möglich, die Auslesungen der Wetterstationen für einen Zeitpunkt abzufragen. Außerdem bietet die Spalte station die Möglichkeit, nach einer bestimmten Wetterstation zu filtern. Auf diese Weise können die Daten aus der Datenbank in Grafana eingespeist werden.

7 | **Diskussion und Evaluation**

7.1 Analyse der Anforderungen

7.2 Evaluation auf Grundlage der Literatur

7.3 Evaluation auf Grundlage der Stakeholder

7.4 Limitationen der Arbeit

8 | Fazit und Ausblick

Literaturverzeichnis

- [Bas+19] JF Bastin u. a. The global tree restoration potential. In: *Science* 365(6448):(2019), 76–79. eprint: <https://www.science.org/doi/pdf/10.1126/science.aax0848>. <https://www.science.org/doi/abs/10.1126/science.aax0848> (siehe S. 19).
- [Bur+06] JA Burke u. a. Participatory sensing. In:(2006) (siehe S. 5).
- [Com+04] ICSP Committee u. a. SWEBOOK: Guide to the Software Engineering Body of Knowledge, 2004 version. In: *IEEE Computer Society*:(2004) (siehe S. 12).
- [Deu] Deutscher Wetterdienst. *Wetter- und Klimalexikon*. Tropennacht. <https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?nn=103346&lv2=102672&lv3=102802> (besucht am 04. 10. 2023) (siehe S. 20).
- [EE84] I of Electrical und E Engineers. IEEE guide to software requirements specifications: approved September 22, 1983, IEEE Standards Board; approved July 20, 1984, American National Standards Institute. In: Inst. of Electrical und Electronics Engineers. 1984 (siehe S. 12).
- [GYL11] RK Ganti, F Ye und H Lei. Mobile Crowdsensing: Current State and Future Challenges. In: *The Internet of Things*:(2011) (siehe S. 5, 6).
- [HHR04] L Heinrich, A Heinzl und F Roithmayr. *Wirtschaftsinformatik-Lexikon*. Oldenbourg Wissenschaftsverlag GmbH, 2004 (siehe S. 6).
- [How+06] J Howe u. a. The rise of crowdsourcing. In: *Wired magazine* 14(6):(2006), 176–183 (siehe S. 6).
- [IFI21] ISB AG, Fraunhofer-Institut für Optik, Systemtechnik und Bildauswertung und Ingenieurgesellschaft Prof. Kobus und Partner GmbH. *Das Projekt „CrowdSensing für Bodensee Online“*. Erfahrungsbericht der Projektbeteiligten. Juli 2021. https://um.baden-wuerttemberg.de/fileadmin/redaktion/m-um/intern/Dateien/Dokumente/3_Umwelt/Nachhaltigkeit_und_Umweltbildung/Digitalisierung/CrowdSensing-fuer-Bodensee-Online-Zusammenfassung.pdf (besucht am 04. 10. 2023) (siehe S. 7, 8).
- [Lan+10] ND Lane u. a. A survey of mobile phone sensing. In: *IEEE Communications magazine* 48(9):(2010), 140–150 (siehe S. 5).
- [Lan11] B Langguth. A review of tinnitus symptoms beyond 'ringing in the ears': a call to action. In: *Current medical research and opinion* 27(8):(2011), 1635–1643 (siehe S. 8).
- [Min21] Ministerium für Umwelt, Klima und Energiewirtschaft Baden-Württemberg. *Projekt „Crowdsensing“ für Bodensee Online*. Nov. 2021. <https://um.baden-wuerttemberg.de/de/umwelt-natur/nachhaltigkeit/nachhaltige-digitalisierung/projekte/crowdsensing-bodensee/> (besucht am 04. 10. 2023) (siehe S. 7).
- [Oke04] TR Oke. *Initial guidance to obtain representative meteorological observations at urban sites*. Bd. 81. World Meteorological Organization Geneva, 2004 (siehe S. 17, 18).
- [Pry+15] R Pryss u. a. Mobile crowd sensing services for tinnitus assessment, therapy, and research. In: *2015 IEEE international conference on mobile services*. IEEE. 2015, S. 352–359 (siehe S. 8).
- [Pry+17] R Pryss u. a. Mobile crowdsensing for the juxtaposition of realtime assessments and retrospective reporting for neuropsychiatric symptoms. In: *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE. 2017, S. 642–647 (siehe S. 8).
- [Ray+22] A Ray u. a. A survey of mobile crowdsensing and crowdsourcing strategies for smart mobile device users. In: *CCF Transactions on Pervasive Computing and Interaction*:(2022) (siehe S. 5–7).
- [SA07] O Salo und P Abrahamsson. An iterative improvement process for agile software development. In: *Software Process: Improvement and Practice* 12(1):(2007), 81–100 (siehe S. 13–15).

[Sie18] M Siepermann. *Agile Softwareentwicklung*. Feb. 2018. <https://wirtschaftslexikon.gabler.de/definition/agile-softwareentwicklung-53460/version-276549> (besucht am 04. 10. 2023) (siehe S. 13).

[SO12] ID Stewart und TR Oke. Local climate zones for urban temperature studies. In: *Bulletin of the American Meteorological Society* 93(12):(2012), 1879–1900 (siehe S. 17, 18).

[SO11] I Stewart und T Oke. Local climate zones: Origins, development, and application to urban heat island studies. In: *Proceedings of the Annual Meeting*

of the American Association of Geographers, Seattle, WA, USA. 2011, S. 12–16 (siehe S. 17).

[Umw23] Umweltbundesamt Deutschland. *Gesundheitsrisiken durch Hitze*. Juli 2023. <https://www.umweltbundesamt.de/daten/umwelt-gesundheit/gesundheitsrisiken-durch-hitze#indikatoren-der-lufttemperatur-heisse-tage-und-tropennachte> (besucht am 04. 10. 2023) (siehe S. 20).

[Wan+19] Y Wang u. a. An Optimization and Auction-Based Incentive Mechanism to Maximize Social Welfare for Mobile Crowdsourcing. In: *IEEE Transactions on Computational Social Systems* 6(3):(2019), 414–429 (siehe S. 6).

Eidesstattliche Erklärung

Ich erkläre hiermit gemäß § 9 Abs. 12 APO, dass ich die vorstehende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Masterarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Bamberg, den _____

Samet Murat Akcabay