

Projektdokumentation



Dokumentation

im Studiengang
Softwaretechnik und Medieninformatik

vorgelegt von

Levin Kerschberger	763080
Moritz Ruider	762806
Henning Weise	762511
Paul Kolbe	762405

am 18. Juni 2021
an der Hochschule Esslingen

Kurzfassung

Gegenstand der hier vorgestellten Arbeit ist die Dokumentation der Software, die im Zuge des Moduls Projekt Medieninformatik entstanden ist. Neben der Dokumentation der Software wird auf den Entwicklungsprozess sowie das Projektmanagement eingegangen.

Inhaltsverzeichnis

Kurzfassung	1
Inhaltsverzeichnis	2
Abbildungsverzeichnis	4
1 Warum Unibay?	5
2 Technologien	6
2.1 MERN-Stack	6
2.2 Frameworks, Libraries	6
2.3 Tools	7
3 Requirements Specification	8
3.1 Product Vision	8
3.2 Personas	9
3.3 User Stories	13
3.3.1 Anmeldung bei Unibay	13
3.3.2 Erstellen einer Kleinanzeige	14
3.3.3 Löschen einer Kleinanzeige	14
3.3.4 Anzeigen von Kleinanzeigen	15
3.3.5 Suchen nach Kleinanzeigen	15
3.3.6 Anzeigen des Profils	16
3.3.7 Merklste	16
3.3.8 Chat	17
4 Projektmanagement Planung	18
4.1 Aufgabenverteilung	18
4.2 Projektmanagement	18
4.3 Definition of Done	19
4.4 Definition of Ready	19
5 Funktionsumfang	20
5.1 Home-Seite	20
5.2 Neue Kleinanzeige-Seite	20
5.3 Profil-Seite	21
5.4 Messenger-Seite	21
6 Alleinstellungsmerkmale	22
7 Architektur	23
7.1 Softwarearchitektur	23
7.1.1 Verteilungsdiagramm des Gesamtsystems	23
7.1.2 Komponentendiagramm des Clients	25
7.1.3 Komponentendiagramm des Servers	27
7.2 Datenbankarchitektur	29
8 Installations- und Administrationshandbuch	31

9 Reflexion	33
9.1 Aufteilung der Teams	33
9.2 Projektmanagement	33
9.3 Lernfortschritt	34
10 Lizenzen	35
11 Ausblick	36

Abbildungsverzeichnis

Abbildung 1: Product Vision Canvas	8
Abbildung 2: Persona Astrid	9
Abbildung 3: Persona Julia	10
Abbildung 4: Persona Martin	11
Abbildung 5: Persona Steffen	12
Abbildung 6: Login-Screen	13
Abbildung 7: Screen Neue Anzeige	14
Abbildung 8: Home-Screen	15
Abbildung 9: Profil-Screen	16
Abbildung 10: Messenger-Screen	17
Abbildung 11: Gesamtsystem	24
Abbildung 12: Client - Komponentendiagramm	26
Abbildung 13: Server - Komponentendiagramm	28
Abbildung 14: ERM Diagramm.	30

1 Warum Unibay?

Wie im Namen unserer App schon erkennbar ist, wollen wir dem Nutzer das Gefühl einer eigenen, sicheren “Bucht” geben. Das bedeutet, dass man nicht wie bei anderen Kleinanzeigenanbietern mit unbekannten Nutzern global handelt, sondern sich im vertrauten Rahmen der Hochschule bewegt. Dies sorgt für ein sehr hohes Maß an Vertrauen und damit auch für ein gutes Gefühl beim Nutzer. Dieser muss sich keine Gedanken machen, dass er einen defekten oder beschädigten Artikel erhält oder sein Geld nicht bekommt. Sollte dies dennoch passieren, so besitzt das Rechenzentrum Daten der entsprechenden Person und kann mit dieser Kontakt aufnehmen. Von anderen Anbietern wie Ebay oder Ebay-Kleinanzeigen ist man im Browser ein Design gewöhnt, das eher veraltet ist und nicht zeitgemäß wirkt. Wir wollen unseren Nutzern das Gefühl geben, nicht mit Informationen überladen zu werden, aber trotzdem alle nötigen Informationen auf einen Blick erfassen zu können. Außerdem wollen wir durch unser modernes, intuitives Design auch in der Desktopansicht dem User das Gefühl geben, dass er sich in einer App befindet und nicht auf einer klassischen Website. Durch die Implementation mit Material-UI/Bootstrap sorgen wir dafür, dass unser Design sich auch bei kleineren Displays wie Laptops und Tablets responsiv anpasst. Als Nutzer kann man Kleinanzeigen in zwei Kategorien anlegen: “Ich suche” oder “Ich biete”. In beiden Kategorien können sowohl Produkte, als auch Dienstleistungen integriert werden. Dies unterscheidet uns von bisherigen Anbietern. Während man auf vielen anderen Plattformen nur materielle Dinge zum Verkauf anbieten kann, ist es mit unserer App auch möglich, Dienstleistungen anzubieten. So kann man Werbung für Seminare und Vorträge machen oder auch Nachhilfe anbieten. Ein weiteres Merkmal, das uns von anderen Anbietern abhebt, ist die “Ich suche” Kategorie. Hier wird es möglich sein, wie klassisch in einer Zeitung, nach Dingen zu suchen. So kann man schreiben, man suche eine Wohnung, einen Nachhilfelehrer, Helfer beim Umzug oder auch Hilfe bei der Einrichtung des Laptops.

2 Technologien

2.1 MERN-Stack

Für uns war relativ schnell klar, dass wir unsere WebApp auf Basis des MERN-Stack aufbauen. Dieser besteht aus der MongoDB, Express, React und NodeJS. Der MERN-Stack ist “State of the Art” und bietet einen tollen Workflow zwischen den einzelnen Elementen. Der Vorteil einer MongoDB liegt darin, dass diese sehr einfach einzurichten ist und wir uns nicht mit aufwändigen Mappings beschäftigen müssen. Außerdem lässt sich die MongoDB sehr gut skalieren und die Verbindung zwischen Backend-Server und Datenbank-Server ist, im Vergleich zu anderen Datenbanken, relativ einfach. Express ist ein Framework für NodeJs, welches als Server-seitiges backend JavaScript Framework funktioniert. Unser Frontend wurde mit React erstellt. Wir haben uns für React und gegen Angluar entschieden, da Levin und Paul React schon kannten und wir so zumindest in einer Sache schon etwas Vorwissen hatten. React ist eine Open-Source JavaScript Bibliothek, die uns das Arbeiten in Komponenten ermöglicht. Open-Source ist an dieser Stelle sehr wichtig für uns, denn bei Open-Source Bibliotheken ist die Menge an Internetseiten darüber sehr groß. NodeJS ist ein Open-Source JavaScript Runtime Environment und ermöglicht es uns, JavaScript auch außerhalb des Browsers, also auf unserem Server, ausführen zu können.

2.2 Frameworks, Libraries

Mongoose ist ein sehr hilfreiches Framework für uns, denn es ermöglicht uns, zwischen den Objekten im Code und der Respräsentation dieser Objekte in der MongoDB zu übersetzen. Für den Echtzeit-Chat benutzen wir Socket.io, welches uns bidirektionale und event-basierte Kommunikation ermöglicht. Das WebSocketprotokoll ist dabei ein Netzwerkprotokoll, das auf TCP aufsetzt. Außerdem benutzen wir statt “fetch” Axios. Axios hat den Vorteil, dass es auch mit älteren Browsern kompatibel ist, was bei fetch nicht gegeben ist. Außerdem hat Axios den Vorteil, dass es automatische JSON Datentransformation durchführt, was bei fetch hingegen manuell gemacht werden muss. Auch bietet Axios uns die Möglichkeit, ein “response timeout” zu setzen und eine Anfrage abzubereiten.

Weitere benutzte Frameworks siehe Kapitel 10: Lizenzen.

2.3 Tools

Zu Beginn war wir uns uneinig darüber, welche IDE wir benutzen sollen, denn jeder hat hier seinen eigenen Favoriten. Jedoch habe wir schnell gemerkt, dass wir uns auf eine IDE einigen müssen, damit wir unnötige Merge-Konflikte verhindern und jeder die gleichen Einstellungen benutzt für Formatierung etc. So haben wir uns gemeinsam auf die IDE Visual Studio Code geeinigt. Um das Problem der unterschiedlichen Einstellungen zu beheben, benutzen wir die Erweiterung “Settings Sync”. Mit Hilfe dieser können wir Einstellungen wie Formatierung, Maximale Zeilenlänge, Plugins etc. global speichern und alle benutzen. Für die teaminterne Kommunikation benutzen wir TeamSpeak und WhatsApp sowie WebEx, falls wir PairProgramming betreiben.

3 Requirements Specification

3.1 Product Vision

<u>Vision Statement</u>	Entwickeln einer Plattform/Web-App für Kleinanzeigen
-------------------------	--

<u>Zielgruppe</u>	<u>Bedürfnisse</u>	<u>Produkt</u>	<u>Nutzen</u>
-Student*innen und Professor*innen bzw. Mitarbeiter der Hochschule Esslingen	<ul style="list-style-type: none"> -Zeitgemäßes Design und intuitive Bedienung auf der Benutzeroberfläche -Sicheres Umfeld durch hochschul-internes Login-System -Der User kann Artikel zum Verkauf und Tausch anbieten und Suchanzeigen erstellen -Austausch zwischen allen Angehörigen der Hochschule -Geld verdienen -alte Gegenstände sinnvoll verwerten -gebrauchte Gegenstände ersteigern -Dienstleistungen in Anspruch nehmen -Kontakte knüpfen 	-Kleinanzeigen, z.B. suche/biete Bücher/Geräte/etc., Freizeitaktivitäten, Mitfahrgelegenheiten und auch Tätigkeiten	<ul style="list-style-type: none"> -Kenntnisse in der Webentwicklung vertiefen -positiver Beitrag zum Hochschulklima

Abbildung 1: Product Vision Canvas

3.2 Personas

Astrid Böttner



"Nur zur Übung, nicht zur Strafe"

Alter: 53

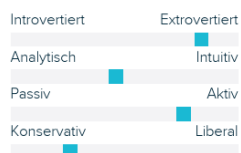
Beruf: Dozentin

Familienstand: verheiratet, 2 Kinder

Wohnort: Göppingen

Fachgebiet: Kindheitspädagogik

Personality



Bio

Astrid ist an der Hochschule Esslingen Dozentin für Kindheitspädagogik. Sie lehrt bereits seit 20 Jahren an der Hochschule und wohnt mit ihrem Mann und ihren 2 Kindern in Göppingen. Sie setzt sich ehrenamtlich für Jugendarbeit ein und unterstützt hilfsbedürftige Kinder mit Kleiderspenden. Außerdem sitzt Astrid abends gerne auf ihrer Terasse mit einem Glas Gin.

Ziele

- Will Materialien bekommen, die sie bei ihrer Arbeit mit Kindern einsetzen kann
- Sie will gebrauchte Klamotten oder Möbel ertauschen um diese zu spenden

Bisherige Erfahrungen

- Astrid hat bisher nicht sehr viel Erfahrung mit anderen Kleinanzeigen-Apps, da sie ihre Sachen meistens auf Flohmärkten kauft oder von Bekannten geschenkt bekommt.

Meine Erwartungen

- Aufgrund ihrer fehlenden Erfahrungen erwartet Astrid eine sehr leicht zu bedienende App
- Kein nerviger, langwieriger Anmeldeprozess
- möchte direkt loshandeln und nicht erst sich in der App zurecht finden müssen
- Sie erwartet eine gut funktionierende Chat-Funktion, da sie damit auch andere motivieren kann, sich für hilfsbedürftige Kinder einzusetzen.

Bevorzugte Geräte

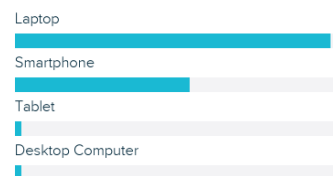


Abbildung 2: Persona Astrid

Julia Neumann



"You glow differently when you're actually happy"

Alter: 19

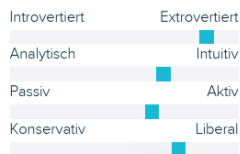
Beruf: Studentin

Familienstand: ledig

Wohnort: Kirchheim unter Teck

Studiengang: Softwaretechnik

Personality



Bio

Julia ist eine Informatik-Studentin im 1. Semester an der Hochschule Esslingen. Zur Zeit wohnt Julia noch bei ihren Eltern, da das Studieren durch Corona momentan von zuhause erfolgt. Julia ist ein sehr aufgeschlossener Mensch und vermisst es neue Leute kennen zu lernen. Zu Kommilitonen konnte sie leider bisher nur via Whatsapp Kontakt aufnehmen. Auch auf Instagram ist sie sehr aktiv und hat schon die ein oder andere bezahlte Werbung auf ihrem Konto veröffentlicht. Da Julia sehr Modeinteressiert ist nutzt sie Apps wie Kleiderkreisel um Kleidung zu tauschen. Das Studium ist für sie bisher ganz gut stemmbar, sie würde sich aber wünschen, dass es bald mit Präsenzunterricht weiter geht.

Ziele

- Guten Nebenverdienst aufbauen mit Instagram
- Das Studium in Regelstudienzeit abschließen
- Alte Klamotten loswerden
- sucht nach alten Formelsammlungen oder Lehrbüchern
- Eine nette Lerngruppe finden
- Sie möchte ohne großen Aufwand Dinge loswerden, heißt ohne das nervige Verpacken und Verschicken

Bisherige Erfahrungen

- Julia hat oft schon Probleme gehabt bei der Kommunikation über Kleiderkreisel. Auch das Verschicken von Paketen nervte sie öfters. Da Julia kein Kontakt zu ihren Mitstudierenden hat, ist sie nicht in der Lage, sich über den Lernstoff und andere Interessen auszutauschen.

Meine Erwartungen

- Julia möchte die App auf ihrem Handy mit einer Hand bedienen können
- Sie möchte neue Leute kennen lernen und mit alten Dingen noch etwas Geld verdienen
- Sie will auch selber Such-Anzeigen veröffentlichen, da sie gerade aktuell nach einem Nachhilfe-Lehrer für Java sucht

Bevorzugte Geräte

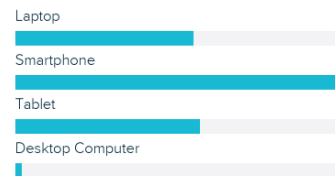


Abbildung 3: Persona Julia

Martin Ross



"Ein gutes Pferd springt nur so hoch wie es muss"

Alter: 23

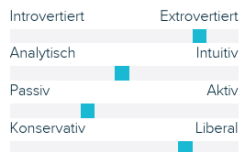
Beruf: Student

Familienstand: ledig

Wohnort: Esslingen am Neckar

Studiengang: Maschinenbau

Personality



Bio

Martin ist Maschinenbau-Student im 4. Semester an der Hochschule Esslingen. Er weiß die Hochschule mit allem was dazu gehört sehr zu schätzen. Außerdem wohnt er in Esslingen in einem Studentenwohnheim mit drei Kollegen zusammen, die ebenfalls Maschinenbau studieren. Zu viert genießen sie das Studentenleben mit all ihren Vorzügen. Martin geht das Studium entspannt an und schaut, dass seine Freizeit nicht zu kurz kommt. Zu seinen Hobbies gehören Basketball und er legt gerne auch auf Parties mal auf.

Ziele

- Studium erfolgreich abschließen
- Mit Unibay alten Krempel loswerden
- Will mehr coole Student*innen kennenlernen
- sucht neues Equipment für seine Tätigkeit als DJ
- Neue Einrichtung für Studentenbude finden

Bisherige Erfahrungen

- Ebay und Ebay Kleinanzeigen sieht aus wie von 1900
- In der Vergangenheit hat er schlechte Erfahrungen mit anderen Kleinanzeigen-Apps gemacht. So hat er über Ebay, Ebay-Kleinanzeigen, Kleiderkreisel, Gebraucht-Kaufen.de und andere Handel-Portale Dinge gekauft, die beschädigt oder auch gar nicht bei ihm ankamen. Auch hat er einige Sachen verschickt und nie das Geld dafür erhalten.
- Der Support der entsprechenden Anbieter hat auf seine Anfragen nichts unternommen, worüber er sehr enttäuscht ist.

Meine Erwartungen

- Moderne intuitive App
- Er will die App am PC benutzen
- leicht zu bedienen
- keine Abzocke mehr
- Support der bei Problemen aktiv wird

Bevorzugte Geräte

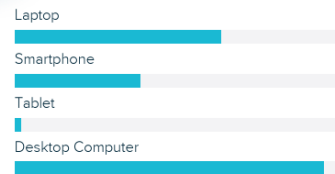


Abbildung 4: Persona Martin

Steffen Stahler



"Es ist keine Schande nichts zu wissen, wohl aber, nichts lernen zu wollen"

Alter: 42

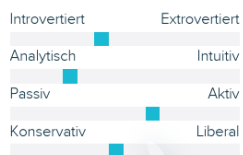
Beruf: Dozent

Familienstand: verheiratet, 2 Kinder

Wohnort: Stuttgart

Fachgebiet: BWL

Personality



Bio

Steffen Stahler hat eine turbulente Karriere in der Automobilindustrie hinter sich. Von Daimler, über Bosch bis Porsche war alles dabei. Daher hat er langjährige Erfahrung im Consulting und Managementbereich. Auf Anfrage der Hochschule Esslingen hat er 2019 als Dozent angefangen. Steffen hat gerne viel Kontakt mit seinen Studierenden und bietet auch individuelle Sprechstunden an. Er engagiert sich für die Erstsemester, damit sie in den schweren Zeiten von Corona schnell Anschluss finden können. Wenn er gerade nicht am Arbeiten ist geht er gerne ins Fußballstadion oder kocht gerne mit seiner Frau und seinen Kindern. Nebenbei geht er einer ehrenamtlichen Tätigkeit nach.

Ziele

- Viel Zeit mit seiner Familie verbringen
- Arbeiten ohne Stress, den hatte er schon genug bei früheren Tätigkeiten
- Karriere machen steht nicht an oberster Stelle
- Möchte sein Wissen und seine Erfahrungen gerne an andere weitergeben

Bisherige Erfahrungen

- Steffen findet die Bedienung von anderen Kleinanzeigen-Anbietern umständlich
- Er tat sich schwer, Leuten mit denen er verhandelt, zu vertrauen
- Ansonsten hat er keine negativen Erfahrungen bisher gemacht

Meine Erwartungen

- Eine sichere Plattform
- Er will gerne Workshops, Vorträge oder Nachhilfestunden anbieten und mit der App mehr Leute erreichen
- Er will interessierte Leute finden, die ihn bei seiner ehrenamtlichen Tätigkeit unterstützen

Bevorzugte Geräte

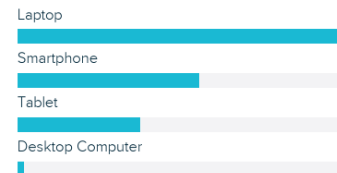


Abbildung 5: Persona Steffen

3.3 User Stories

3.3.1 Anmeldung bei Unibay

Als Unibay-Nutzer möchte ich mich bei der Plattform anmelden können, um alle meine Angebote und Suchanfragen oder relevante, nutzerbezogenen Daten sehen, verwalten und bearbeiten zu können.

Als Unibay-Nutzer möchte ich, dass ich mich, falls ich mich während der Browser-Session schon einmal über das KeyCloak SSO angemeldet habe, nicht nochmals erneut auf UniBay anmelden muss, da ich so direkt Zugriff auf alle Funktionen habe und mir die erneute Anmeldung erspare.

Als HS-Esslingen Rechenzentrum Mitarbeiter möchte ich die Nutzer verwalten können ohne etwas an der Software Unibay zu verändern, da dies nicht im Bereich meiner administrativen Tätigkeit liegt und viel Zeit in Anspruch nehmen würde. Insbesondere aufgrund der nötigen Kommunikation mit dem Entwickler-Team (Non-Function).

Der Login für Unibay wird komplett ohne Login Forms sein, da die Userverwaltung über Keycloak implementiert wird. Somit können sich die User mit ihrem Kürzel von der Hochschule und dem entsprechenden Passwort anmelden. Nach erfolgreicher Anmeldung wird man auf die Startseite weitergeleitet.

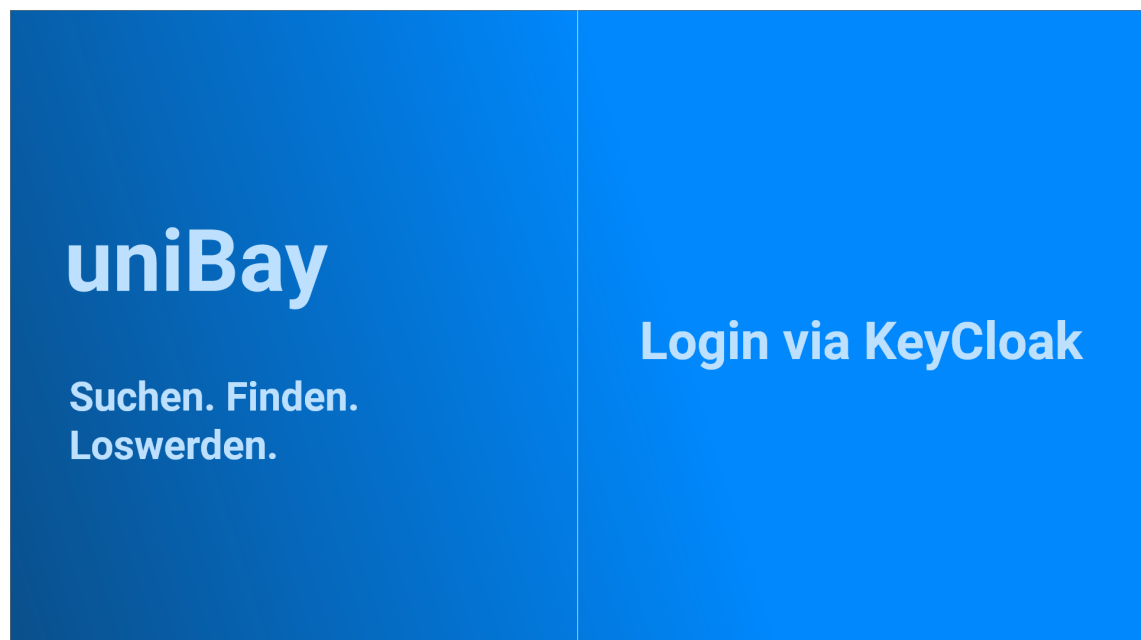


Abbildung 6: Login-Screen

3.3.2 Erstellen einer Kleinanzeige

Als Unibay-Nutzer möchte ich Kleinanzeigen erstellen können, welche ich mit einem Titel, einer Beschreibung, verschiedenen Tags und Kategorien und optionalen Bildern versehen kann, um so die Dinge die ich zu verkaufen/verschenken/tauschen habe strukturiert auf den Markt bringen zu können. Dabei habe ich die Wahl zwischen einer Ich-Suche oder einer Ich-Biete Anzeige.

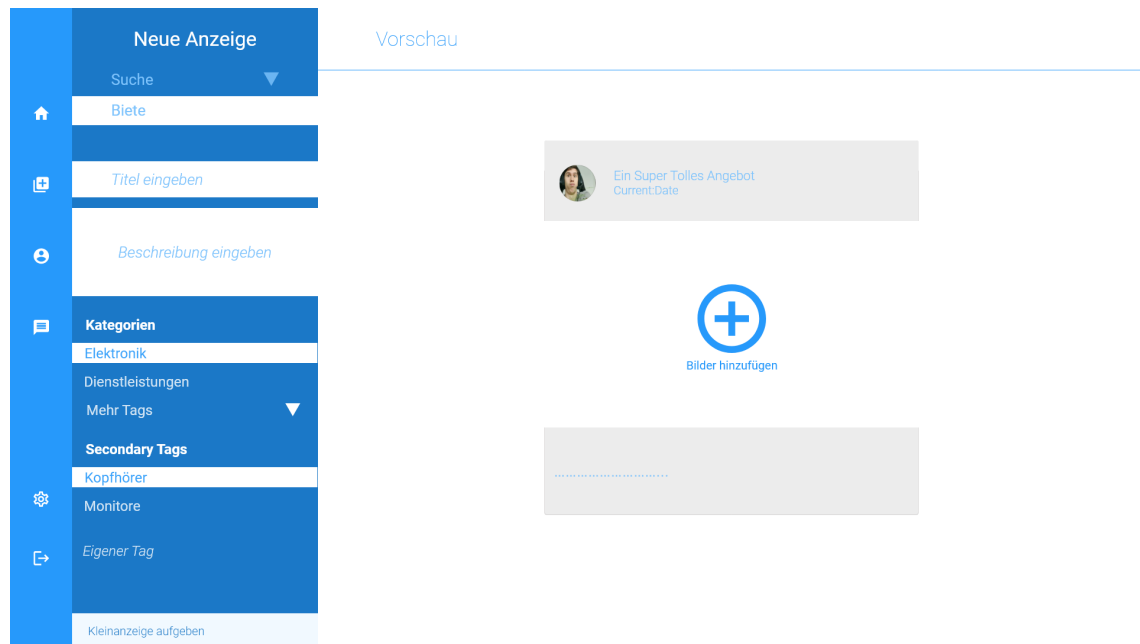


Abbildung 7: Screen Neue Anzeige

3.3.3 Löschen einer Kleinanzeige

Als Unibay-Nutzer möchte ich die Möglichkeit haben, eine Kleinanzeige zu löschen, wenn ich diese nicht mehr online haben will. Dies ermöglicht es mir, die Plattform für mich selbst und andere übersichtlich zu halten.

Als Unibay-Nutzer wünsche ich mir, dass ich Kleinanzeigen mit einem Ablaufdatum versehen kann. So muss ich mich nicht manuell um das Löschen kümmern, falls meine Anzeige niemanden anspricht.

3.3.4 Anzeigen von Kleinanzeigen

Als Unibay-Nutzer möchte ich die Möglichkeit haben, mir Kleinanzeigen anzeigen zu lassen um einen Überblick zu bekommen, was gerade angeboten wird.

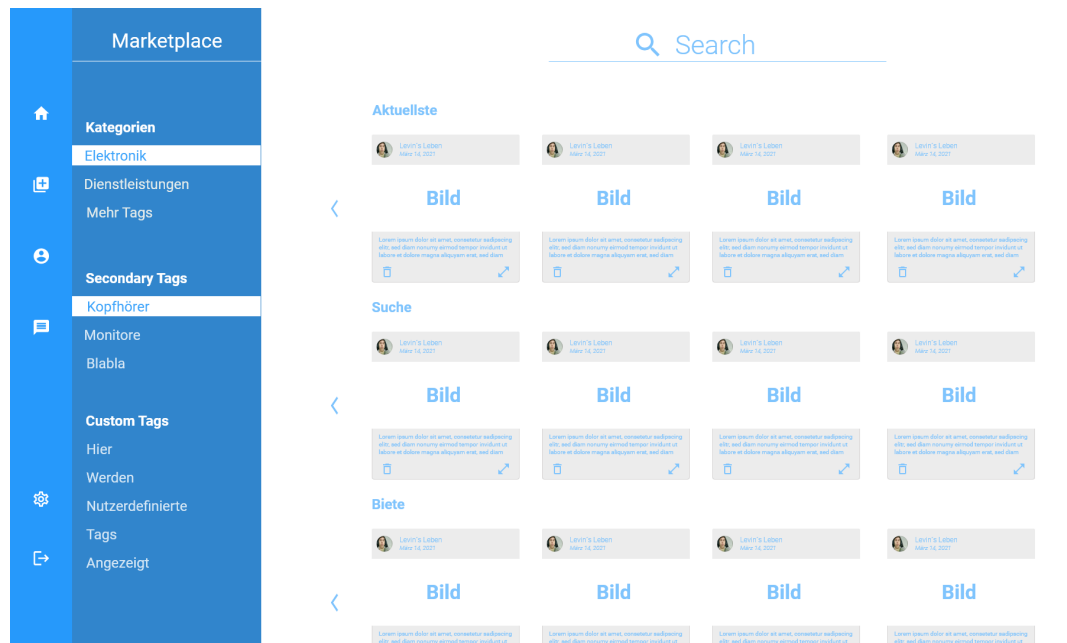


Abbildung 8: Home-Screen

3.3.5 Suchen nach Kleinanzeigen

Als Unibay-Nutzer möchte ich Kleinanzeigen nach Schlagwörtern durchsuchen, um so nur die Kleinanzeigen zu sehen, die mich interessieren.

Als Unibay-Nutzer ist es wichtig für mich, beim Stöbern durch die Anwendung verschiedene Filter wie Suchradius, Kategorien und Tags einstellen zu können, damit ich auch genau die Dinge angezeigt bekomme die mich interessieren.

3.3.6 Anzeigen des Profils

Als Unibay-Nutzer möchte ich eine Übersicht über mein Profil bekommen, um zu sehen, welche Anzeigen ich selber anbiete und um diese gegebenenfalls löschen zu können. Außerdem möchte ich hier einen Überblick über Anzeigen bekommen, die auf meiner Merkliste sind.

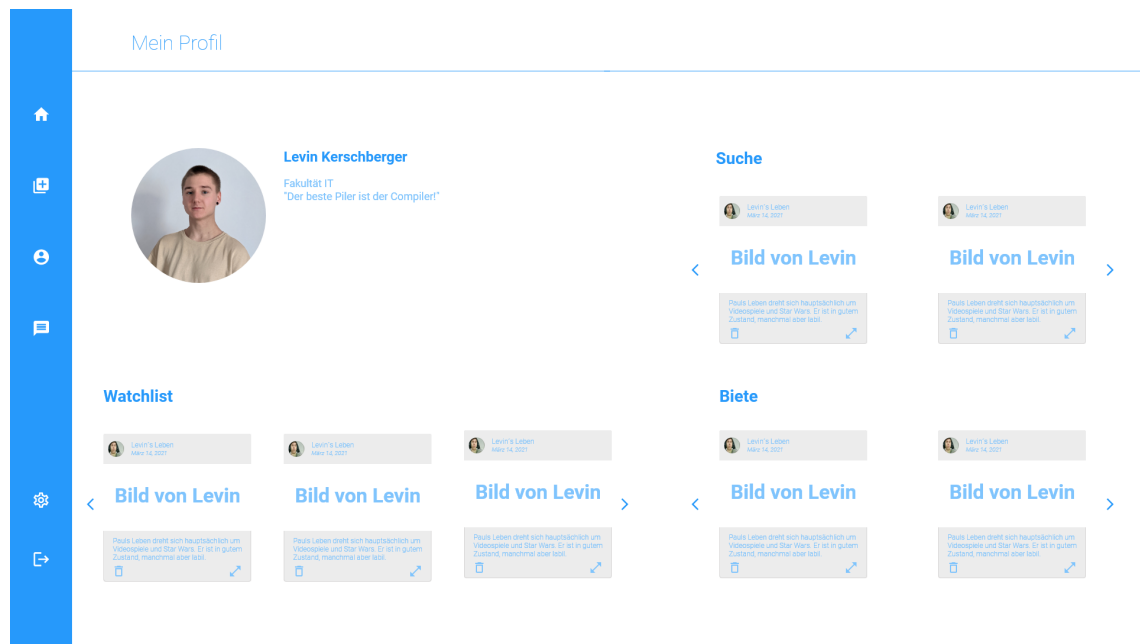


Abbildung 9: Profil-Screen

3.3.7 Merkliste

Als Unibay-Nutzer möchte ich verschiedene Kleinanzeigen für später speichern können, um etwas anderes zu tun und später wieder schnell darauf zurück zu kommen. Diese soll schnell erreichbar sein ohne umständlich viel hin- und her navigieren zu müssen.

3.3.8 Chat

Als Unibay-Nutzer möchte ich mit anderen Nutzern kommunizieren, um über den Preis eines Produkts zu verhandeln und Details austauschen zu können.

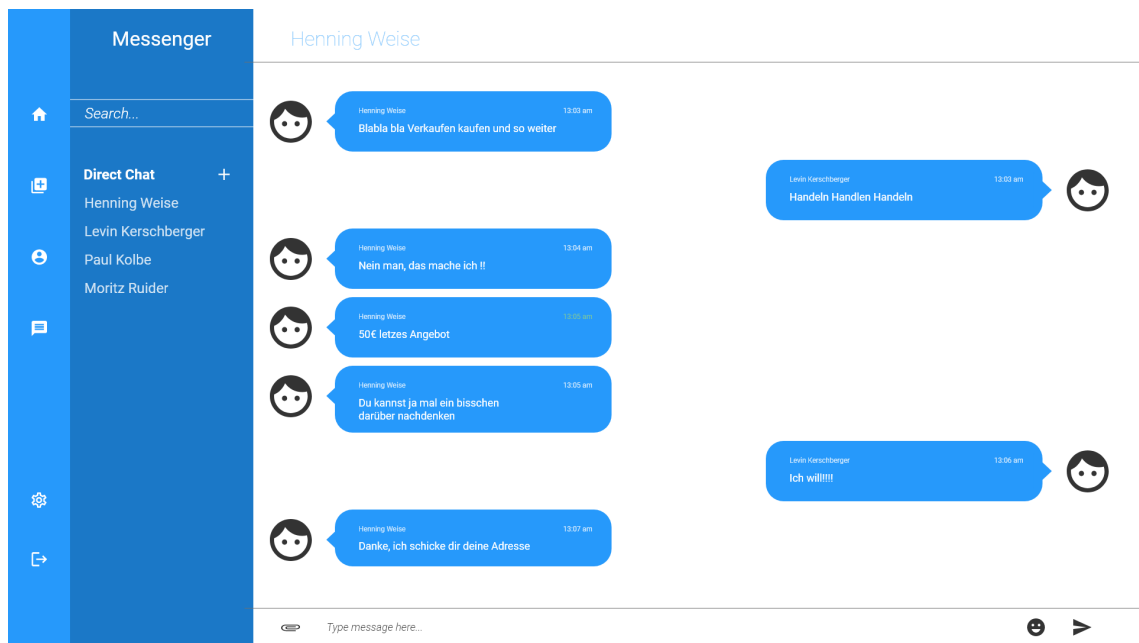


Abbildung 10: Messenger-Screen

4 Projektmanagement Planung

4.1 Aufgabenverteilung

Wir haben uns dazu entschieden, nicht wie oft üblich nach Backend und Frontend aufzuteilen. Da wir alle die Vertiefung Medieninformatik gewählt haben, liegt der Schwerpunkt von uns allen im Frontend und somit haben wir alle den gleichen Kenntnisstand im Backend. Folglich macht es keinen Sinn, zwei Leute dem Backend zuzuteilen. Wir sind der Meinung, dass wir mehr aus dem Projekt lernen können, wenn jeder sich mit Backend beschäftigt. Außerdem erzeugt das Trennen nach Back/Frontend eine gewisse Abhängigkeit. So kann an vielen Stellen erst mit dem Frontend begonnen werden, wenn die entsprechenden Schnittstellen im Backend geliefert wurden. Da wir jedoch sehr individuelle Arbeitsgeschwindigkeiten haben, könnte dies zu gewissen Engpässen führen. Das Aufteilen in Features ermöglicht es uns in Git, die Branches nach Features aufzuteilen und für jedes Feature einen einzelnen Branch anzulegen. Das Aufteilen nach Komponenten führt dazu, dass eine Person sich im Detail mit einem Feature auskennt, von Frontend bis Backend. Dies könnte auch später beim Testen und fixen von Bugs ein Vorteil sein. So kann ein Fehler in einer Funktion genau die Person betreuen, die dieses Feature geschrieben hat. So ist kein unnötiges „herumreichen“ nötig. Sollte es dennoch zu Problemen oder Sackgassen kommen, lassen wir uns einander nicht im Stich und werden uns versuchen zu helfen. Alle anderen Schritte und Aufgaben, die nichts mit der Implementierung direkt zu tun haben, haben wir bisher gemeinsam als Team erledigt und das werden wir auch so beibehalten.

4.2 Projektmanagement

Dienstags haben wir wöchentliche Meetings mit Herr Renz. Hier besprechen wir gemeinsam, wie wir voran kommen und wo wir Probleme oder Fragen haben. Des Weiteren nutzen wir diesen Termin um konstruktive Kritik zu erhalten und um diese dann in unser Projekt mit einfließen lassen können. Den Freitag haben wir als teaminternen Tag geblockt. Hier treffen wir uns jeden Morgen und besprechen Punkte, die gemeinsam diskutiert werden müssen und besprechen die Planung für die darauffolgende Woche. Die restliche Zeit teilt sich jeder individuell so ein, dass er zu den Stichterminen mit seiner Arbeit fertig ist. Um miteinander kommunizieren zu können, benutzen wir TeamSpeak3 und WhatsApp. Für Pair-Programming und Screen-Sharing benutzen wir WebEx und Google Meet. Wenn wir spontane Gedanken, Anregungen oder auch Fragen aufschreiben wollen, benutzen wir Google Docs.

Unser Arbeiten wird in Sprints erfolgen. Das Ziel ist es, einen Sprint mit circa zwei Wochen anzusetzen. Hier werden dann von jedem Teammitglied die vorher besprochenen Features entwickelt beziehungsweise implementiert. Je nachdem, wie uns das gelingt, werden wir die zwei Wochen dementsprechend verkürzen oder verlängern. Im Sprint

Planning entscheiden wir, welche Features wir in den Sprint nehmen und implementieren werden. Hierbei unterstützt uns auch die Versionsverwaltungssoftware Git, da wir hier die einzelnen Schritte markieren können, in dem sich ein Feature gerade befindet (“Ready”, “In Progress”, “In Testing”, “In Documentation”). Außerdem helfen uns die Milestones dabei, den Überblick zu behalten, wann ein Feature fertig sein muss. Wir haben eine Definition of Done erstellt, die uns vorgibt, wann ein Feature fertig ist. Hier haben wir uns darauf geeinigt, dass das Feature erfolgreich in den Master-Branch integriert sein muss, erfolgreich getestet worden sein muss und dokumentiert sein muss. Hiermit wollen wir dafür sorgen, dass die Qualität am Ende unseren Ansprüchen entspricht und wir nicht noch am Ende nachträglich testen oder dokumentieren müssen. Eine Definition of Ready haben wir erstellt, da wir nicht wussten, dass wir noch Vorgaben zu User-Stories, Szenarien etc. bekommen. Mit der DoR wollten wir gewährleisten, dass wir auch diese Dinge erledigen.

4.3 Definition of Done

Ein Feature ist dann fertig, wenn es sowohl im Frontend als auch im Backend mit dazugehöriger Datenbankfunktionen implementiert wurde. Außerdem sollte das Feature ausreichend getestet sein und keine direkt ersichtlichen Fehler mehr enthalten. Um die Zusammenarbeit zu erleichtern, sollte zusätzlich der Code ausreichend und verständlich kommentiert werden. Des Weiteren sollte die Vorgehensweise und die Codestruktur in der Dokumentation beschrieben sein. Auch Bugfixes/ Hotfixes fallen unter diese Definition of Done. Außerdem muss der Feature-Branch / Bugfix-Branch erfolgreich mit dem Master-Branch gemerged worden sein.

4.4 Definition of Ready

Da wir agil vorgehen, sollte das Product Backlog Item in Form einer User Story vorliegen und passende Szenarien formuliert werden. Ein Product Backlog Item ist dann bereit für den Sprint, wenn alle benötigte Abhängigkeiten klar definiert wurden und alle Teammitglieder die selbe Vorstellung davon haben, wie dieses umgesetzt wird.

5 Funktionsumfang

5.1 Home-Seite

Auf der Home-Seite unserer Web-App bekommt man einen Überblick über alle derzeit aktiven Kleinanzeigen. Um nicht von der Menge erschlagen zu werden, ist die Filterfunktion hier sehr hilfreich. Über den linken Drawer kann der Nutzer nach bestimmten Merkmalen suchen, wie nach einer der beiden Typen “Ich Suche/Ich Biete”. Außerdem kann er den Suchradius einstellen, um so nur Ergebnisse zu sehen, die in seiner Nähe sind. Das Filtern nach Kategorien ist sehr hilfreich für den Nutzer, falls er nur stöbern möchte. Will der Nutzer nach neuen Möbeln stöbern, so kann er in der Filterleiste die entsprechende Kategorie aktivieren. Neben den Filtern gibt es oben noch eine Suchleiste. Hier kann man nach passenden Schlagwörtern in den Titeln der Kleinanzeigen suchen, wie zum Beispiel “Fahrrad”. Alle Suchen funktionieren in Echtzeit und sobald eine Änderung getätigt wird, werden die Kleinanzeigen neu geladen. Eigene Kleinanzeigen werden hier selbstverständlich nicht angezeigt. Jede Kleinanzeige hat, je nach Daten, einen Preis, einen Ort, eine Beschreibung und einen Titel. Über den Button “Zur Merkliste” lässt sich die Kleinanzeige zu seiner Merkliste hinzufügen. Ist diese dort bereits vorhanden, ist der Button ausgegraut und ist ohne Funktion beim Klicken. Über den Button “Anfragen” lässt sich ein Echtzeitchat mit dem Besitzer starten. Existiert bereits ein Chat mit diesem, gibt es statt diesem Button einen “Zum Chat” Button.

5.2 Neue Kleinanzeige-Seite

Über die Navbar erreicht der Nutzer die Seite, um eine neue Kleinanzeige zu erstellen. Hier wird er durch die verschiedenen Schritte geleitet, um eine Kleinanzeige zu erstellen. Der Nutzer wird gefragt, ob er optional einen Preis, ein Bild und seinen Standort hinzufügen möchte. Je nach Wahl des Nutzers werden entsprechende Felder angezeigt. Klickt der User ausversehen auf “Weiter ohne Standort”, so kann er jederzeit beliebig viele Schritt zurück gehen um Änderungen zu tätigen. Gibt der Nutzer beim Titelfeld und bei der Beschreibung keine Daten ein, so wird er durch ein rotes Hervorheben des Textfeldes darauf hingewiesen, dass seine Eingabe nicht vollständig ist. Bei der Eingabe eines Preises wird geprüft, ob es sich bei der Eingabe um eine ganzzahlige Zahl handelt. Am Ende des Prozesses bekommt der Benutzer noch eine Vorschau, wie seine Kleinanzeige später aussehen wird.

5.3 Profil-Seite

Auf der Profilseite hat der Nutzer die Möglichkeit, seine eigenen Anzeigen sortiert nach “Ich Suche/Ich Biete” einzusehen. Hier hat er außerdem die Möglichkeit, seine eigenen Kleinanzeigen zu löschen. Tätigt er den “Löschen” Button, so wird er über eine zusätzliche Bestätigungsform gefragt, ob er diese wirklich löschen möchte. Dies haben wir hier eingebaut, da es für den Nutzer fatal ist, wenn er ausversehen auf den Löschen-Button kommt und seine Anzeige direkt gelöscht wird. Auch seine Merkliste sieht der Nutzer auf der Profilseite. Hier hat er eine Übersicht über alle seine Anzeigen auf der Merkliste und kann auch hier direkt einen Chat mit dem Anbieter starten oder auch die Kleinanzeige aus seiner Merkliste wieder entfernen.

5.4 Messenger-Seite

Zu jeder Kleinanzeige kann ein Nutzer eine Anfrage stellen. Diese Anfrage erstellt in der Datenbank einen Eintrag im Feld “conversations” und leitet den Nutzer automatisch auf den Messenger, mit dem neuen Chat offen, weiter. Schickt der Nutzer eine Nachricht in diesem Chat, dann wird die Nachricht in der Datenbank mit zugehöriger Konversation-ID gespeichert. Durch dieses Datenmodell ist es möglich, dem Nutzer alle seine offenen Anfragen auf der linken Seite mit Titel der Kleinanzeige und zugehörigem Nutzernamen anzuzeigen. Wechselt der Nutzer den Chat, werden wie zu erwarten auch direkt die zugehörigen Nachrichten geladen. Für die Chat-Funktion wurde zusätzlich ein Socket implementiert. Dieser ist dafür zuständig, dass Nachrichten in Echtzeit bei anderen Nutzern ankommen und ermöglicht es einen Online-Status für gerade aktive Nutzer anzuzeigen. Bestmöglich wird darauf geachtet, über das Frontend zu verhindern, dass Nutzer Anfragen mehrmals stellen oder sich selbst Anfragen schicken. So ist ein sauberes Nutzererlebnis gewährleistet.

6 Alleinstellungsmerkmale

Ziel unserer Web-App ist es, sich von anderen Kleinanzeigenanbietern abzuheben. Dies haben wir schon in unserem innovativen, modernen Design umgesetzt. Der Anwender soll das Gefühl bekommen, sich in einer App zu bewegen und nicht auf einer klassischen Webseite. Dies soll durch die seitliche Menüleiste zusätzlich gestärkt werden. Außerdem ist unsere WebApp mit Icons ausgestattet, die sie selbsterklärend machen und dem User eine intuitive Bedienung ermöglicht.

Ein Feature, welches wir an dieser Stelle besonders hervorheben können, ist das sogenannte “Reverse Geocoding”. In unserer MongoDB werden optional für Kleinanzeigen die longitude und latitude gespeichert. Dies erfolgt direkt über die Abfrage des Gerätestandorts. Aktiviert der Benutzer dieses Feature, so wird er durch ein Popup gefragt, ob er zulassen möchte, dass die App auf den Gerätestandort zugreift. Dies geschieht über die JavaScript API navigator.geolocation. Besitzt nun eine Kleinanzeige longitude und latitude, so wird das Reverse Geocoding benutzt, um die Koordinaten in einen Ort umzuwandeln. Dazu benutzen wir Geocode und in diesem Zusammenhang die Google Maps Geocoding API. So sieht man bei diesen Kleinanzeigen den Ort, in welchem diese erstellt wurden. Dies ermöglicht es uns auf der Startseite, Kleinanzeigen nach ihrem Standort zu filtern. Über einen Slider kann der User den Suchradius selbst festlegen oder auch bei Bedarf deaktivieren.

Auf der Messenger-Seite war uns sehr wichtig, auf gute Bedienbarkeit und UX-Erfahrung zu achten, da wir in Zeiten von Whatsapp und Co. große Erwartungen an solche Features haben. Beispielsweise erkennt unsere Applikation, wenn neue Nachrichten hinzukommen und wenn man viele Nachrichten austauscht, scrollt das Chatfenster automatisch nach unten, um immer die aktuellsten Informationen anzuzeigen. Um dem Nutzer eine sehr angenehme und moderne Kommunikation zu ermöglichen, haben wir die Möglichkeiten von React genutzt, um Zeichenkombinationen wie “:)”, “:D”, “:/” als Emoji darzustellen.

7 Architektur

7.1 Softwarearchitektur

7.1.1 Verteilungsdiagramm des Gesamtsystems

Das Front- und Backend des Gesamtsystems ist in verschiedene Abschnitte unterteilt. Für das Backend nutzen wir einen NodeJS-Server, während das Frontend auf dem JavaScript Framework React basiert. Die Kommunikation wird über eine REST-API mittels Express realisiert und die Nutzerauthentifizierung erfolgt über eine SSO-Technologie namens Keycloak. Keycloak ist ein Server, welcher basierend auf dem SAML-Protokoll Authentifizierungsdaten empfangen kann. Um diese Daten an den Server zu schicken wird Passport.js verwendet. Mittels JWT-Tokens kann so Authentifikation durchgeführt werden, oder dem Backend mitgeteilt werden, dass ein Nutzer authentifiziert ist. Falls die Authentifizierung erfolgreich ist, wird im Backend der eingeloggte User als authentifiziert gespeichert und falls der Nutzer noch nicht in der Datenbank vorhanden ist, wird dieser automatisch angelegt. Sämtliche Daten der Application werden in einer MongoDB gespeichert. Diese cloudbasierte Lösung erfüllt Datensicherheitsstandards wie HIPPA (Health Insurance Portability and Accountability Act of 1996) oder SOC und stellt eine NoSQL-Datenbank dar.

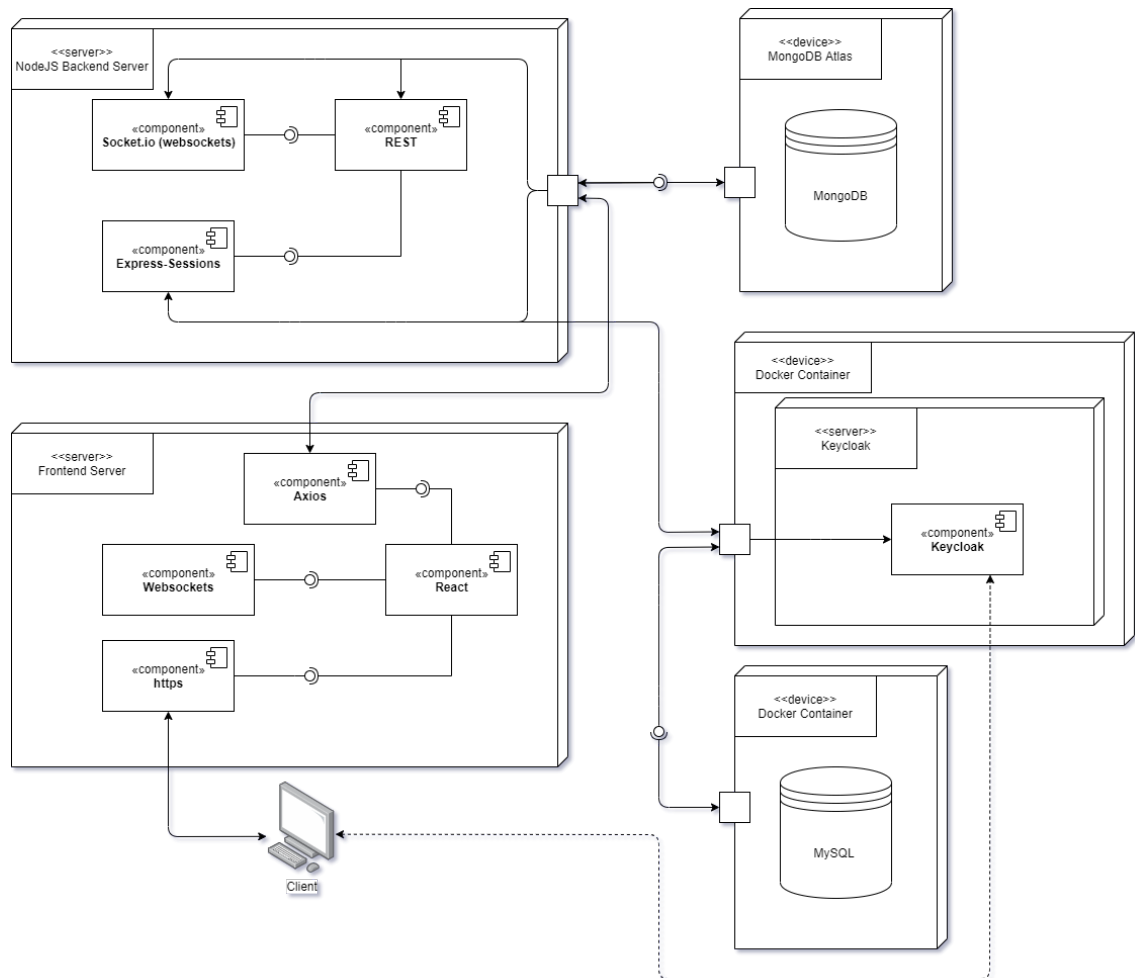


Abbildung 11: Gesamtsystem

7.1.2 Komponentendiagramm des Clients

Das Frontend wurde mittels React realisiert und ist in verschiedene Abschnitte, bzw. Komponenten unterteilt. Der React-Router wird dazu verwendet, um zwischen den einzelnen Komponenten zu navigieren. So lässt sich eine Single-Page-Application erstellen, die den Nutzer denken lässt, er rufe die einzelnen Unterseiten auf. Sämtliche Komponenten wurden mit Bootstrap erstellt. Die Kommunikation mit den Backend erfolgt mit dem Modul Axios. Darüber lassen sich Anfragen an das Backend schicken und Antworten verarbeiten. Man kann das Frontend in die Komponenten Home, Add, Profile, Messenger, Navbar und Search unterteilen. Navbar und Search sind dabei Komponenten, die ständig angezeigt werden. Über die Navbar kann man mittels dem React-Router zu den anderen Abschnitten navigieren. Die Search-Komponente dient dazu, gezielt nach Kleinanzeigen suchen zu können. Man landet bei der Suche immer auf der Home-Seite. Die Home-Komponente besteht aus der Filter- und Offercard-Komponente. In der Offercard-Komponente werden die einzelnen Kleinanzeigen dargestellt. Über die Filter-Komponente kann man über verschiedene Kriterien nach bestimmten Kleinanzeigen filtern. Kleinanzeigen können in der Add-Komponente erstellt werden. Diese besteht aus der AddForm-Komponente, welche wiederum in weitere kleine Abschnitte unterteilt ist. Diese Abschnitte dienen dazu, die verschiedenen Attribute einer Kleinanzeige zu erfassen. Ein Nutzer kann über die Profile-Komponente seine eigenen Kleinanzeigen und gespeicherte Kleinanzeigen verwalten. Die Komponenten OwnOffercard und Watchlistcard sind dabei Abwandlungen von der Offercard-Komponente. Die Messenger-Komponente ermöglicht einen Echtzeit-Chat zwischen verschiedenen Nutzern. Dazu wird das Modul socket.io benutzt. Das Frontend des Chats wird in die Komponenten Conversation, ChatOnline, Messages und Message unterteilt.



7.1.3 Komponentendiagramm des Servers

Das Backend lässt sich in verschiedene funktionale Abschnitte unterteilen. Die Daten der Application werden in einer MongoDB gespeichert. Die Struktur dieser Daten wird über verschiedene Models festgelegt. Die Modelle werden über das Modul mongoose erstellt. Unsere Rest-API ist in verschiedene Routen unterteilt, welche dem Frontend die Möglichkeit geben, über HTTP-Anfragen verschiedene Operationen in der Datenbank auszuführen. Zu jedem Model, bzw. jeder MongoDB-Collection, gibt es eine spezifische Route, welche Funktionen zum Erstellen, Verändern oder Abrufen von Daten aus der Datenbank bereitstellt. In der Hauptdatei werden per Middleware die Routen an die Application gebunden. Auch die Authentifizierung mittels Saml und Passport findet im Backend statt und wenn diese erfolgt ist, wird man wieder auf eine Frontend Route zurückgeleitet.

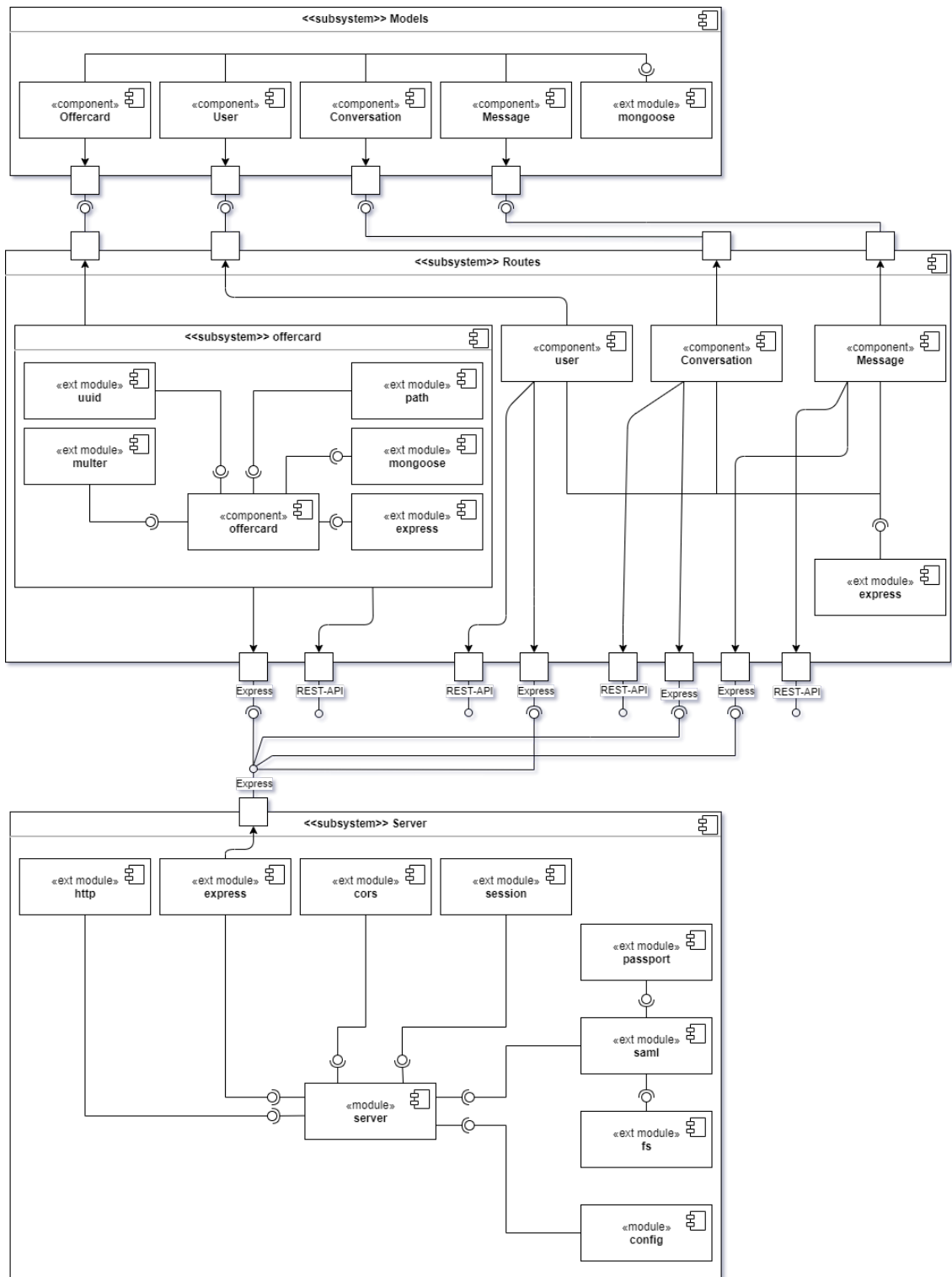


Abbildung 13: Server - Komponentendiagramm

7.2 Datenbankarchitektur

Das Diagramm zeigt die Datenmodelle, die in der MongoDB gespeichert sind und ihre Beziehungen zueinander. Die Modelle wurden mit dem Module mongoose erstellt. In einer MongoDB bekommt jedes Objekt eine einzigartige ObjectID. Über diese kann das Objekt gefunden werden. Das erste Modell bildet den User ab. Jeder User hat einen einzigartigen Username des Typs String. Dieser Username wird zudem für die Authentifizierung verwendet. Das Attribut photo verweist auf ein Foto. Dieses wird im Profil des Users als Profilbild angezeigt. In dem Attribut watchList speichert das Usermodell die ObjectIDs der Offercards ab, die sich auf seiner Merkliste befinden. So können über die watchList die gespeicherten Offercards gefunden werden. Die Attribute firstName, lastName und mail speichern den Vor- und Nachnamen und die E-Mail-Adresse des Users ab. An dieser Stelle sei gesagt, dass diese Informationen noch nicht in der Anwendung genutzt werden, allerdings für zukünftige Features bereits im Modell stehen. In dem Modell Offercard werden die Daten einer Kleinanzeige gespeichert. Jede Offercard hat einen Typ, der im Attribut type gespeichert wird. Dieser kann entweder „Suche“ oder „Biete“ sein. In den optionalen Attributen longitude und latitude werden, falls gewünscht, die Koordinaten des Erstellungsorts der Kleinanzeige gespeichert. Über sie wird der Ort angezeigt und es kann innerhalb eines Suchradius gefiltert werden. Der Ersteller der Kleinanzeige wird im Attribut username gespeichert. Dabei wird nicht die ObjectID eines Users verwendet, sondern sein Username. Das hat den Grund, da dieser auch für die Authentifizierung genutzt wird und es im Frontend leichter ist auf diesen zuzugreifen. Da der Username aber ebenfalls einzigartig sein muss, besteht keine Fehlergefahr. Im Attribut title steht der Title der Kleinanzeige, in description die Beschreibung, in price der Preis und in category die Kategorie. Diese Informationen werden bei der Erstellung der Kleinanzeige vom User festgelegt. Das letzte Attribut tags wird derzeit noch nicht genutzt. Es sollte vom User erstellte Tags speichern, nach denen gesucht werden kann. Dieses Feature wurde noch nicht umgesetzt. Für den Messenger von Unibay werden die Modelle Conversation und Message benötigt. In Conversation wird eine aktuelle Unterhaltung bzw. ein Chat von zwei Usern zu einer Kleinanzeige gespeichert. Dazu werden in dem Attribut members die Usernames von zwei Usern gespeichert und im Attribut offercardId die ObjectID einer Offercard. Im Modell Message wird eine einzelne Nachricht, die in einem Chat geschrieben wird gespeichert. Dazu wird der Username des Senders in sender gespeichert und die ObjectID der dazugehörenden Conversation in conversationId. Zusätzlich werden das Erstellungsdatum in createdAt und der Inhalt der Nachricht in text gespeichert.

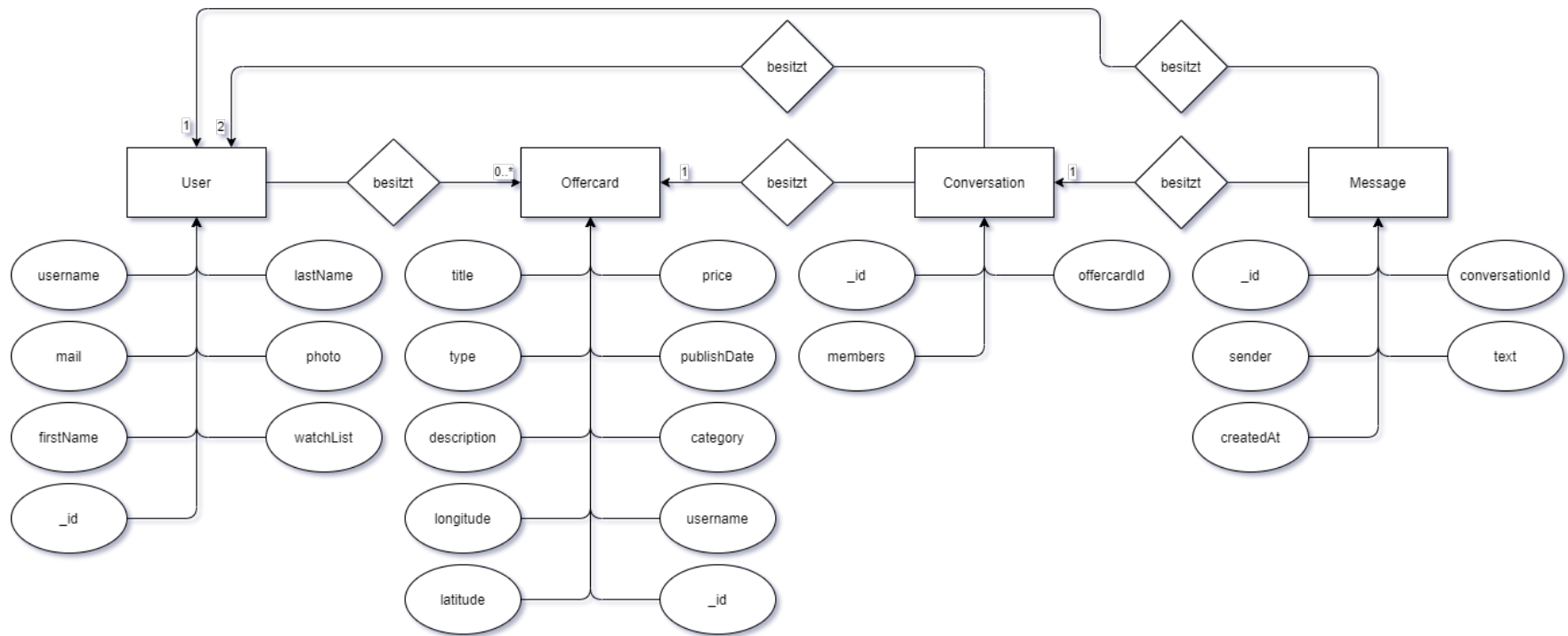


Abbildung 14: ERM Diagramm.

8 Installations- und Administrationshandbuch

1. Installieren von Docker und Docker-Compose

- (a) Installieren der Docker Desktop Installer.exe und “Install required Windows components for WSL2” aktivieren

2. Im Terminal folgenden Befehl eingeben:

```
docker run -p 8080:8080 -e KEYCLOAK_USER=admin -e KEYCLOAK_PASSWORD=admin quay.io/keycloak/keycloak:14.0.0
```

3. In der Keycloak Admin Console mit Username und Passwort anmelden

4. Realm erstellen mit Namen: hse

5. Neuen Client erstellen

- (a) Name: UniBuy
- (b) Client Protocol: saml
- (c) Valid Redirect URIs: `http://localhost:5000/*`
- (d) Base URL: `http://localhost:3000/`
- (e) Master SAML Processing URL: `http://localhost:5000/login/callback`
- (f) Speichern

6. Öffne <http://localhost:8080/auth/realms/hse/protocol/saml/descriptor> und speicher den Inhalt von `<ds:X509Certificate>` in `<projectdir>/certs/idp_cert.pem`

7. Navigiere zu SAML-Keys, Export

- (a) Archive Format: PKCS12
- (b) Key Alias: uniBuy
- (c) Key Password: "1234"
- (d) Realm Certificate Alias: hse
- (e) Store Password: "1234"
- (f) Download speicher in `<projectdir>/keystore.p12`

8. Key und Cert extrahieren

- (a) `cd <projectdir>`
- (b) `openssl pkcs12 -in keystore.p12 -nocerts -out certs/privateKey.pem -passin pass:"1234`
- (c) `openssl rsa -in certs/privateKey.pem -out certs/key.pem -passin pass:"1234`
- (d) `openssl pkcs12 -in keystore.p12 -clcerts -nokeys -out certs/server.crt -passin pass:"1234`

9. Clone GitLab Repository

10. Eben erstellte Dateien in `/server/certs` verschieben, ggf. vorhandene Dateien überschreiben
11. Drei Konsolen öffnen in Project Directory
12. `cd server` bzw. `cd client` bzw. `cd socket`
13. jeweils `npm i`
14. jeweils `npm start`
15. <http://localhost:3000> and Enjoy!!

9 Reflexion

9.1 Aufteilung der Teams

Da wir unser Team nicht explizit nach Frontend/Backend sondern nach Features aufgeteilt haben, ist es manchmal dazu gekommen, dass ein Mitglied mehr Wissen in bestimmten Bereichen hatte, als die anderen Teammitglieder. Unsere Befürchtung, dass es zu Engpässen beziehungsweise Sackgassen kommen könnte, weil sich manche Teammitglieder nicht mit bestimmten Features befassen haben, hat sich nicht bewahrheitet. Durch gute Kommunikation und auszureichende Hilfestellung konnten wir diese Wissenslücken jedoch füllen und immer unser Ziel erreichen. Im Nachhinein lässt sich zur unserer Teamaufteilung sagen, dass es besser war, nach Features aufzuteilen. Dadurch konnte ein sehr großer Lernfaktor in unserem Team erreicht werden, der uns für unsere berufliche Karriere sicherlich hilfreich sein wird.

9.2 Projektmanagement

Grundsätzlich können wir sagen, dass wir unsere Planung unseres Projektmanagements nahezu 1:1 umgesetzt haben. Die angesetzten Termine wurden von allen Teammitgliedern wahrgenommen und die angestrebten Zwischenziele wurden erreicht. Natürlich ist es das ein oder andere mal dazugekommen, dass ein Zwischenziel zu einer angesetzten Deadline nicht erreicht wurde, aber selbst dann haben wir einander geholfen und uns gegenseitig unterstützt, um an unser Ziel anzukommen. Daraus können wir für uns schlussfolgern, dass manche Aufwandplanungen eventuell unterschätzt worden sind und besser, beziehungsweise detaillierter geplant werden hätten müssen. Selbstverständlich lag die Arbeitsproduktivität nicht immer bei 100%. Aufgrund von anderen Modulen oder sonstigen anstehenden Terminen kam es auch mal zu etwas unproduktiveren Arbeitstagen. Durch gegenseitige Motivation und sehr gute Organisation konnten wir unsere Ziele dennoch erreichen. Das Versionsverwaltungstool Git, als auch das Meeting mit unserem Betreuer, hat uns dabei sehr geholfen, um einschätzen zu können, in welchem Stand sich unser Projekt befindet und wie wir uns zeitlich organisieren müssen. Die Wahl unserer Kommunikationsmittel hat sich auch als sehr gut erwiesen. Durch ständigen Austausch in schriftlicher als auch in mündlicher Form konnten wichtige Informationen ausgetauscht werden, die zu unserem Ergebnis beigetragen haben. Durch die Teamspeak Funktion, dass man mehrere Audio-Chaträume erstellen kann, konnten wir uns in Zweiergruppen oder auch in Einzelarbeit aufteilen, um möglichst produktiv zu arbeiten. Die Pair-Programming Einheiten per WebEx haben sich auch als sehr effektiv herausgestellt. Bei Auftreten einiger Unklarheiten oder Problemstellungen konnte man sich gemeinsam an die Sache setzen und nach einer Lösung suchen. Zusammenfassend lässt sich sagen, dass die Planung unseres Projektmanagements sich als sehr positiv bestätigt hat und wir die Tools immer wieder für zukünftige Projekte benutzen würden.

9.3 Lernfortschritt

Durch dieses Projekt konnte nicht nur vorhandenes Wissen gefestigt, sondern auch neue Themengebiete als auch Spaß an der Projektarbeit vermittelt werden. Aufgrund unserer geringen Vorkenntnisse in der Fullstack-Entwicklung war der Anfang beziehungsweise die Einarbeitungsphase relativ schwierig. Doch durch Anschauen einiger Tutorials, gegenseitige Unterstützung sowie (gegenseitige) Motivation konnten wir einen Einstieg als Team finden und können rückblickend sehr stolz auf das Erbrachte sein. Auch in schwierigeren Phasen des Projekts, beziehungsweise Phasen, in denen Probleme nahezu unlösbar erschienen, haben wir nicht aufgegeben und als Team die Probleme bewältigt. Des Weiteren ist nicht nur das entstandene Ergebnis des Projekts für uns wichtig, sondern auch die Entwicklung der Kompetenzen der einzelnen Mitglieder. Für einige von uns hat sich nämlich das Interesse entwickelt, später mal im Berufsleben einer ähnlichen Tätigkeit, also als Webentwickler zu arbeiten. Letztendlich lässt sich sagen, dass durch das Projekt jeder von uns einen unheimlichen Lernfortschritt gemacht hat, da zu Beginn unsere Erfahrung in diesem Bereich nahe Null lagen. Wir haben bereits zuvor bei anderen Projekten als Team gearbeitet, aber dieses Projekt hat uns als Team noch einmal mehr zusammengeschweißt. Die Teamarbeit lief einwandfrei und wir würden jederzeit wieder in der selben Gruppe Projekte durchführen.

10 Lizenzen

Framework / Library	Lizenz	Version	Verwendung
Socket.io	MIT	4.1.2	Client und Server
Socket.io-Client	MIT	4.1.2	Client
Material-UI	MIT	4.11.4	Client
Axios	MIT	0.21.1	Client
Bootstrap	MIT	5.0.1	Client
Popper.js	MIT	1.16.1	Client
React	MIT	17.0.2	Client
React Date Picker	MIT	4.1.1	Client
React-Dom	MIT	17.0.2	Client
React-Geocode	MIT	0.2.3	Client
React-Router-Dom	MIT	5.2.0	Client
React-Scripts	MIT	4.0.3	Client
Timeago.js	MIT	4.0.2	Client
Web-Vitals	Apache-2.0	2.0.1	Client
React-Emoji	MIT	0.5.0	Client
React-Scroll-To-Bottom	MIT	4.1.2	Client
Body-Parser	MIT	1.19.0	Server
Cors	MIT	2.8.5	Server
Dotenv	BSD-2-Clause	10.0.0	Server
Express	MIT	5.17.1	Server
Express-Session	MIT	1.17.2	Server
Mongoose	MIT	5.12.14	Server
Nodemon	MIT	2.0.7	Server
Passport	MIT	0.4.1	Server
Passport-Saml	MIT	3.1.0	Server
Multer	MIT	1.4.2	Server
UUID	MIT	8.3.2	Server

11 Ausblick

Das Hauptziel unseres Projektes bestand darin, eine Plattform/Web-App zu entwickeln, die zum Erstellen, Anzeigen und Verwalten von Kleinanzeigen für geschlossene Benutzergruppen fungieren soll. Nach dem Abschluss unseres Projektes kann man sagen, dass die grundlegenden Funktionen, sowie zusätzliche Features implementiert wurden und auch funktionieren. Doch unsere App ist noch lang nicht ausgereift. Während der Entwicklung sind uns noch ein paar Punkte aufgefallen, die verbessert beziehungsweise ausgebaut werden können. Als ein weiteres Feature haben wir an eine Settingsseite gedacht, in der der User persönliche Einstellungen vornehmen kann. Mögliche Einstellungsmöglichkeiten könnten beispielsweise ein Profilbild sein oder das Einschalten des Darkmodes. Des Weiteren sollen Kleinanzeigen mit Tags versehen werden können, die quasi als "Kategorien" dienen. Der User kann dann nach vorhandenen Tags suchen oder neue erstellen. Eine optionale Logout-Funktion würden wir auch für sinnvoll halten. Um unserer App noch transparenter und vertraulicher wirken zu lassen, würde die Implementierung einer Bewertungsfunktion und die Möglichkeit, andere Profile und deren Durchschnittsbewertung zu sehen, ideal sein. Ansonsten soll noch die Option bestehen, dass Chats nach Titel durchsucht werden können, um den Benutzer eine strukturierte Übersicht der Chats bieten zu können.