

1. Team

1.1 Mitglieder

Marc Sommer, Student M.Sc. Wirtschaftsinformatik @TU München

Techstack: Webanwendungen, App-Entwicklung mit Typescript (Ionic), viel Wordpress/CMS-Erfahrungen, Java, PHP, SQL, Python, Firebase

Alexander Hodes: Software Entwickler und Student M.Sc. Verteilte und mobile Anwendungen

Techstack: Webanwendungen (Angular, TypeScript, NestJS, Front-/ Backend), App-Entwicklung (NativeScript, Android), Java, SQL, UNIX, Scrum

Larissa Greth, Innovation Consultant/ Idea Development, Fahrenheit 212 (Branding, Storytelling, Innovationsstrategie, Kommunikation, UX)

Dennis Beste, Tech Founder & Angel, Strategie, Konzept, Marketing, Innovation Management, Company Building, Projektmanagement, Generic Skills AI, Generic Skills Blockchain, Generic Skills IoT

Kerstin Limberger: Projektmanagerin, UX, Frontend, HTML, Design

Benedikt Bleuel: dualer Wirtschaftsinformatik-Student FOM Düsseldorf
(Entwickler-Kenntnisse: Backend: Java, PHP, Frontend: Angular, Typescript, Datenbanken: SQL)

Kilian Schneider: Android Entwicklung(Java), ggf Native Script(hybrid App), Javascript, HTML, CSS

André Rosendorff: dualer Student M.Sc. Data Science, Consultant für IoT & Industry 4.0, Python, Machine Learning, Java, Prozessmodellierung

Christian Klüh: abgeschlossenes B.Sc Studium der Informatik mit Entwickler Erfahrung & Fähigkeiten(u.a. Java und React)

Asha Olol: General Management Absolventin @Zeppelin Universität, Projektmanagerin und Start-up Gründerin, Übersetzerin für Somali & Deutsch für deutsche Behörden (meist für unbegleitete Jugendliche Geflüchtete)

Felix Schlegel: Python, Node.js, PHP und Web Development im Allgemeinen, iOS Entwicklung in Swift

Jannik Rulitschka: Full Stack mit Angular, Vue.js, Node.js, HTML/CSS, Java Spring Boot sowie UX, ggf. auch native Apps für iOS und Android

Sarah Mordelt: IT Consultant im Financial Services; Wirtschaftsinformatik-Bachelor (DHBW Stuttgart), basic skills / Erfahrungen: chatbots, Microsoft Azure Bot Framework, Dialogflow, web design, agile, scrum, c#, wordpress/elementor

David Schmidt, Student B.Sc. Informatik an der TU Dortmund
Tech: C++, Java, Python, diverses bzgl. Web, C#, SQL uvm., einige Erfahrungen mit IoT/Smart City, Linux, großes Interesse an der Verarbeitung natürlicher Sprache

Katharina Köth, Freelance Experience Director, Berlin
Informationsarchitektur, Wireframes, Prozesse & Flows, Digitalstrategie

Felix Beil, Informatik Student
Tech: Java, Android, PHP, Python, Web basics (JS/Angular)

Robert Ecker, Teamlead Engineering at HolidayCheck during the day, iOS developer at night

1.2 Ansprechpartner

Projektmanagerin: Kerstin Limberger
IT-Leitung: Marc Sommer

Android-Entwicklung: Kilian Schneider
iOS-Entwicklung: Felix Schlegel
Wireframing & Flow Chart : Katharina
Design: Hanna Burgstaller
Content: Asha und Kerstin Limberger
Marketing/Video/Präsentation/Webseite (Konzept): Larissa
Twilio: Jannik Rulitschka
Webseite (Umsetzung): MENT_Lukas_H
Audio Aufnahmen: Katja Egler

1.3 Abwesenheiten

Samstag

Andre: ab heute Nachmittag verfügbar

Alexander: halben Tag heute nicht verfügbar ab 17:30

Dennis: zw 23-08

Larissa: 15-18 Uhr abwesend

Katharina: ab 18uhr

Kerstin: 16:45 - 18:30 circa abwesend

Christian: Samstagabend ab ca. 21:00 Uhr

Asha: Samstag ab 21:15 Uhr

Sonntag

Felix B: 14 Uhr - 15:30 Uhr

Jannik: 15 - ca. 17 Uhr

David: 16:30-ca.18:30 Uhr

Asha: ca. 19:30-20:30 Uhr

Organisationstools

Ticketsystem: Trello

Kommunikation: Slack

Repository: Git

Abgabe Plattform: Devpost

Website: HTML, CSS, Bootstrap

Verwendete Technologien

App: Android (Java), iOS (Swift)

Hosting und Datenbank: Firebase

Website/ Landing-Page: HTML, CSS, Bootstrap

Cloud-Kommunikationsplattform: Twilio

Personal Identification: Passbase

Audio Bearbeitung: Ardour

Video Schnitt: iMovie

Design: Adobe Illustrator, Adobe Photoshop, Keynote

Zugangsdaten:

Youtube:

E-mail: machbarschaft.de@web.de

PW: machbarschaft2020

2. IT-Architektur

- zentrale Datenbank (zB Firebase)
- Applikation (iOS/Android/Hybrid?)
 - **App**
 - Telefoniefunktion
 - Auflistung der Aufträge in der Umgebung
 - Registrierung
 - **Frontend**
 -
 - **Funktionen**
 - Registrierung (Personalausweis wie bei AirBnB?)
 -
 - **Backend:** Aufbereitung der Daten aus Trint (Abfrage per REST) und Speicherung in Datenbank, Bereitstellung von REST-Schnittstelle
 - Datenbank-Anbindung
 - REST-Schnittstelle
 - Schnittstelle zu Trint
 - Service für Ermittlung Aufträge im Umkreis
 - **Sprachansage:** Trint in Kombination mit Twilio?
 - **Landingpage:** Wordpress oder beliebiges Framework je nach verfügbarer Zeit (Host: Github, oder falls Wordpress genutzt, Google Cloud mit Credits)

Tech-Stack

App: Android (Java), iOS (Swift)

Hosting und Datenbank: Firebase

Website/ Landing-Page: Wordpress

Ideen:

Twilio, Trint für Voice to text/Telefon -> Kommunikation zu Server, Datenbank

REST Schnittstelle zum Abrufen der Daten

App Entwicklung

iOS-App-Entwicklung

Android-App-Entwicklung

Anmeldung (Authentifizierung)

Login (z.B. JWT Token)

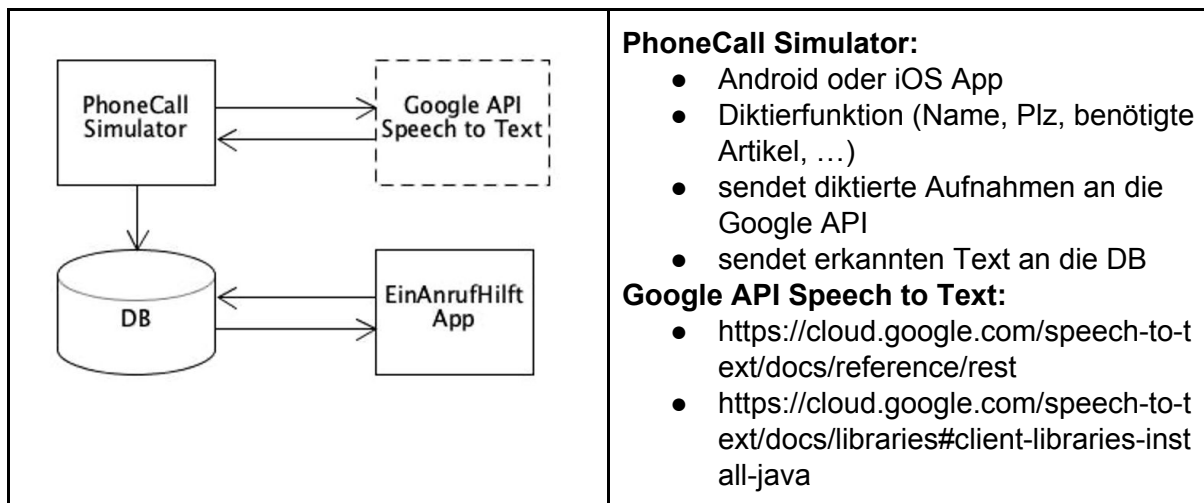
REST Aufruf

Visualisierung der Hilfe-Gesuche

Webpräsenz/Projektpräsentation

Kategorien: Einkaufen, Apotheke, Sonstige Angelegenheiten

Ohne Twilio und Trint:



Analyse Annahme des Hilfesgesuchs

Datenstruktur:

Entität: Order

Feld	Datentyp	Beispiel
phone_number	String	+49 160 1234567
address	String	Musterstraße 123, 12345 Musterstadt
first_name	String	Max
category	String, Array	Einkauf

id	String	UUID
last_name	String	Mustermann

Account (Helfer)

Feld	Datentyp	Beispiel
id	String	
phone_number	String	49 160 1234567
name	String	Maxi Musterfrau
credits	Integer	10
radius	Float	4,6 (km)

credits: Anzahl erfolgreicher Aufträge

radius: manuell gesetzter Radius vom User, in dem er angezeigt werden will

Order_Account

Feld	Datentyp	Beispiel
account_id	String	
order_id	String	
status	String (Enum)	open, confirmed, closed
created_timestamp	Date	21.03.2020 09:56
id	String	UUID

Offene Punkte

-

App

- Telefonnummer Authentifizierung
- Eingabe des Namens
- Stadt eingabe / automatische Standortermittlung (GPS) => Nutzer bestimmt Radius
- Liste von Aufträgen in der Umgebung
- Auftragsreservierung => Nummer des Kunden wird Sichtbar => Nutzer kann Kunden anrufen, um genauere Details zu bekommen
- Verifizierung eines erledigten Auftrags?
- (Optional) Push Notification sobald ein neuer Auftrag in Reichweite ist

Rest Schnittstelle:

POST Register

POST AUTH manNumber+manPassword -> get Token

GET List<AuftragsDTO>

Verifizierung?

Zwei Zielgruppen

Primäre Zielgruppe für Hilfe:

Menschen, die...

- nicht digital sind
- zur Risikogruppe gehören (älter/ vorerkrankt/ gefährdet)
- nicht mobil/ selbstständig sind
- sonst niemanden haben
- *dürfen sie krank sein?*

Primäre Zielgruppe für Helfer:

Menschen, die...

- gesund sind und nicht zur Risikogruppe gehören
- digital sind
- gemeinnützig + freiwillig helfen wollen
- (niemanden haben, der sonst "analog" unterstützt werden kann)
- *evtl. ausweitbar auf gemeinnützige Initiativen oder Unternehmen deren sonstige Tätigkeiten gerade gestoppt sind*

Was wird angeboten:

Unterstützende Tätigkeiten: Einkäufe, Erledigungen (Apotheke), Sonstiges

Helferhandbuch/Leitfaden: Kassenbon aufbewahren

Bewertungssystem für Nummern, die anrufen

Gemeinnützig, keine Vergütung für Helfer

Authentifizierung? Wie?

Bezahlung -> ausschließlich Bargeld

Ablauf Bedürftiger:

Anruf mit Angaben -> Wartet -> Wird vom Helfer angerufen -> restliches Besprechen

Automatischer Rückanruf, falls keiner
Manuell aufgenommen

Basic Requirements - Deadline Sonntag Abend

- Chatbot - Transkribierung
- Übertragung der Daten in App
- Darstellung des "Warenkorbs" + Auswahl und Darstellung für Helfer
- Account für Helfer
- Rückruffunktion
- Push Benachrichtigung
 - wo wird Hilfe benötigt
- Handbuch/FAQ

Ablauf Helfer:

Verifizierung -> Anmeldung -> Hilfe-Gesuche anschauen -> Hilfe-Gesuche auswählen/reservieren -> Telefonische Rückfrage mit Bedürftigem

Registrierung für

#EinAnrufrhilft

#DeutschlandhilftDir

Ausgangssituation: Nachbarschaftshilfe ist in einer Pandemie vor allem gegenüber den gefährdeten (älteren) Zielgruppen ein wichtiger Baustein für das Fortbestehen unserer Gesellschaft.

Problem: Viele Nachbarschaftshilfen organisieren sich digital. Ältere Menschen sind oft nicht digital und können daher Ihre Bedürfnisse nicht gerecht platzieren.

Lösung: Eine KI-gestützte Telefonnummer/Hotline, zB 0800-20002000, auf der ältere Menschen einfach anrufen und Ihre Bedürfnisse aussprechen können. Ein Chatbot nimmt den Wunsch/Bedürfnis entgegen. Der Chatbot bildet daraus ein Transkript (Künstliche Intelligenz via Speech-to-Text). Parallel wird durch die Rufnummer (Location Based Service) die Adresse ermittelt.

Procedere:

- 1) Ältere Person (ab XX Jahre) ruft bei automatisierter Hotline an. Folgende Dinge werden gesagt: XXXX. Die Person muss nicht registriert sein.
- 2)

Vorhandene Informationen: Name, Adresse, Bedürfnis

Diese Informationen werden in einer APP: #Deutschland hilft Dir (Arbeitstitel), angezeigt.

Nutzer, welche sich in der Nähe des Bedürftigen aufhalten und die App installiert haben und öffnen, können helfen.

Helfer: Menschen, die sich über die APP als Helfer anbieten wollen, müssen sich einmalig mit Ihrer Personalausweisnummer/Reisepass registrieren. Damit soll Missbrauch vorgebeugt werden.

Kommunikation: Postwurfsendung in jeden deutschen Briefkasten

Case: Zielgruppengerechter (Telefon) Service zur unmittelbaren Hilfe. Skalierbar, KI gestützt und schnell zu realisieren

Vorgehen

1. Lösungsansatz finden und Idee strukturieren
2. Umsetzung planen
3. Umsetzung durchführen

Backlog:

- Appzugang auch für Geholfene, nicht nur Telefon
- Bonuspunkte für Helfer, ggf. Perks mit Partnerunternehmen

3. Roadmap MACHBARSCHAFT

MACHBARSCHAFT

EIN ANRUF HILFT

Hilfe ohne Medienbruch für ältere Menschen

Wir entwickeln einen Telefonservice, den ältere Menschen einfach anrufen können, um ihre Bedürfnisse auszusprechen. Ein Bot nimmt die Anfrage entgegen und bildet daraus mit Hilfe von Künstlicher Intelligenz eine Anfrage, die wir in eine App einspielen. In der App finden sich freiwillige Nachbarn und Nachbarinnen, die Anfragen annehmen und Erledigungen übernehmen – vom kleinen Einkauf bis zur Abholung von Medikamenten oder anderen Erledigungen für das tägliche Leben.

So machen wir Nachbar:innen zu Machbar:innen.

Phase 1: BUILD + LAUNCH

Use Cases:

- Einkäufe und Erledigungen für den Alltag im Ausnahmezustand

Tech/ Plattform:

- Machbarschafts-Telefonnummer
- Bot/ KI
- Machbarschafts-App

Kommunikation:

- Zielgruppe ältere Menschen (Nachbar:innen): Postwurfsendung jeder Haushalt, Flyer im Supermarkt, Radio / Lokale Tages-/Zeitungen
- Zielgruppe jüngere Menschen (Machbar:innen): Social Media Marketing, OOH und TV über Partnerschaften, Flyer im Supermarkt

Phase 2: SCALE

Use Cases:

- Neue Services, z.B. Hunde ausführen oder Dinge leihen und tauschen

Tech/ Plattform:

- Ausbau der App und Ausweitung auf digitale Risikogruppen: Anfragen stellen und annehmen per App
- Entwicklung von Tech Back-end Lösung für Voice-/Telefon-Bestell-Service für Supermärkte (Abholung über Machbar:innen, Supermarkt-Lieferanten oder andere Lieferdienste)

Partnerschaften:

- mit anderen Nachbarschafts-Diensten

Kommunikation:

- Weiterführung der Kommunikation, ggf. Einbindung von geeigneten Influencern für die junge Zielgruppe

Phase 3: EXPAND (Post-Corona)**Use Cases und Features:**

- Handwerker-Services
- Nachbarschafts-Event-Planung und Community Gardening/ Farming
- Vermittlung von Pflegekräften und Haushalts-Services
- Hilfe für chronisch Kranke
 - Arzttermine buchen
 - Friseur/ Kosmetik/ Reha buchen
 - Transportdienste

Tech/Plattform:

- VideoChat (wenn Internet bei Zielgruppe gegeben wäre)
- Digitale Arzt Chat/ Arzt Visite (wenn Internet bei Zielgruppe gegeben wäre)
- Push Notifications Location Based Service für Hilfe an Helfer (Mehr Push vs Pull Prinzip für schnellere Hilfen)
- Seamless Payment wenn ältere Zielgruppe App und Wallet o.ä.

Partnerschaften

- Mobility Services
- Lebensmitteleinzelhandel
- Krankenkassen
- Apotheken/DocMorris
- Banken/Bankgeschäft
- Telefonische Behördendienste/Gänge

- MyHammer o.ä für Hauswerksarbeiten/Handwerkerdienste.