

Technische Spezifikation Softwareentwicklung

Meet me

Entwicklung einer Open Source Applikation zur
Optimierung von Kommunikation und Terminfindung

Autor des Dokuments

1. Maged Yakan
2. Rayen Bedoui
3. Elia Ariana

Ort, Datum

Berlin, 30.04.2022

Inhalt

1	Analyse	3
1.1	Kontext und Überblick	3
1.1.1	Externe Schnittstellen	4
1.2	Anwendungsfälle	5
1.3	Qualitätsanforderungen	9
1.4	Abnahme Szenarien und -kriterien	9
2	Entwurf	12
2.1	Fachliche Strukturen und Modelle	12
2.2	Benutzeroberfläche	13
2.3	Systemarchitektur	14
2.4	Technologiestack und Lösungsansätze für die Umsetzung der Qualitätsanforderungen	15
2.5	Lizenz Modell	15
3	Aufteilung der Kapitel	15

1 . Analyse

1.1 Kontext und Überblick

Es soll eine App erschaffen werden, die Terminabsprachen zwischen zwei oder mehreren Nutzern ermöglicht und vereinfacht.

- /K10/ Dazu sollen sich Nutzer mit einem Profil registrieren können, in dem sie sich mit E-Mail und Passwort anmelden können.
- /K20/ Ein Nutzer soll andere Nutzer suchen und mittels E-Mail oder QR-Code in eine Kontaktliste hinzufügen können.
- /K30/ Benutzer sollen Termine im App-Kalender erstellen können, in dem Tag, Zeit und Ort festgelegt und gegebenenfalls ein Infotext festgelegt werden. Die Termine werden im iCAL Format gespeichert und können im Anschluss eingesehen, editiert und gelöscht werden können.
- /K40/ Andere Nutzer können zu erstellten Terminen über einen gespeicherten Kontakt oder eine E-Mail-Adresse eingeladen werden. Jeder eingeladene Nutzer hat die Möglichkeit Termine zu bestätigen oder abzulehnen.
- /K50/ Über die Termine können sich alle dort eingeladenen Nutzer mittels eines Terminbezogenen Chatrooms über Nachrichten austauschen.
- /K60/ Darüber hinaus sollen sich die Nutzer an Termine erinnern lassen können.

Für dieses Vorhaben benötigt es mehrere Schnittstellen. Es benötigt Schnittstellen zum Internet, einer Database sowie einem Kalenderformat. Da die App die Terminabsprache für die Nutzer vereinfachen soll, ist es sehr wichtig, dass sowohl die Darstellung als auch Nutzung benutzerfreundlich und einfach gestaltet ist.

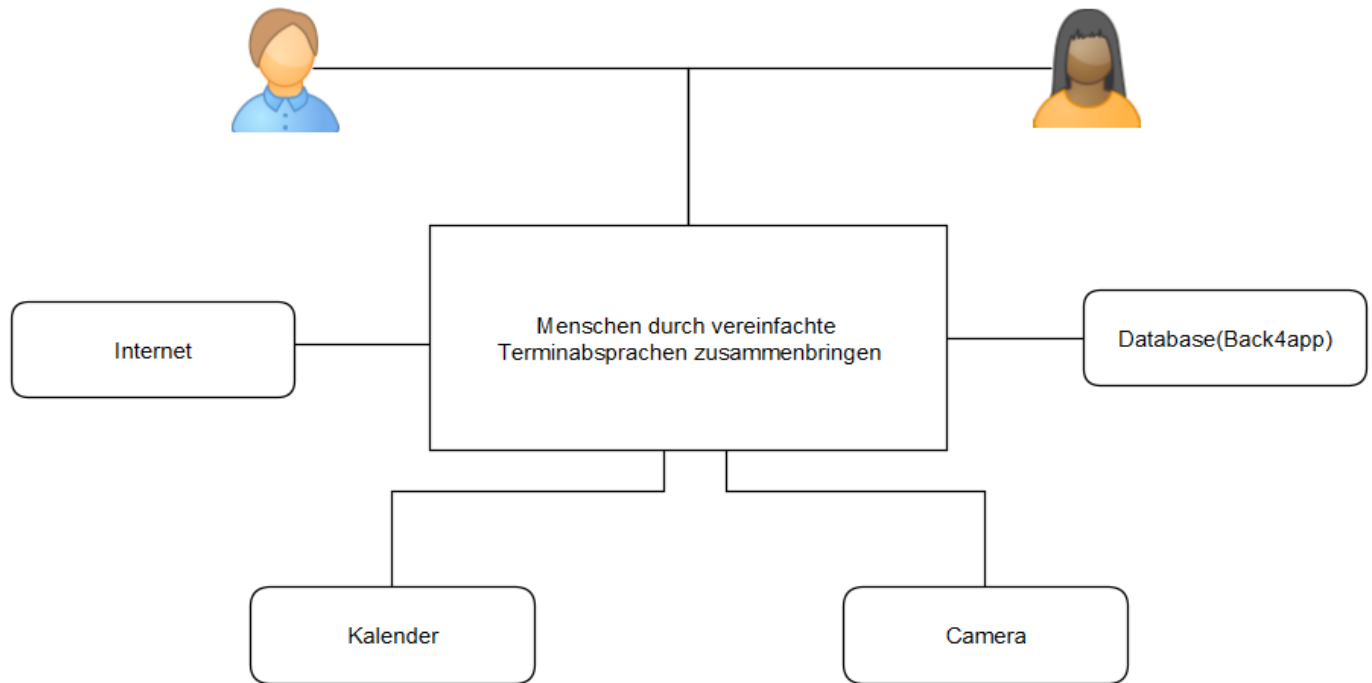


Abbildung 1 : Diagramm zum Systemkontext mit Erfassung der Schnittstellen

1.1.1 Externe Schnittstellen

<i>Name / Bezeichnung der Schnittstelle</i>	<i><Name der Schnittstelle></i>	Syntax: Daten und Formate
/LES10/	Internet	unsere Applikation benötigt Internet (Wlan oder Mobile Daten) Zugang um zu funktionieren.
/LES20/	Database(Back4app)	Datenformat : JSON (GraphQL)
/LES30/	Kalender	Die Termine werden im iCAL Format gespeichert und können im Anschluss eingesehen
/LES40/	Kamera	Zugriff auf die rückseitige Kamera (um Qr-Code zu lesen)

1.2 Anwendungsfälle

Das Use Case Diagramm erklärt, welche Funktionalitäten unsere App bietet.

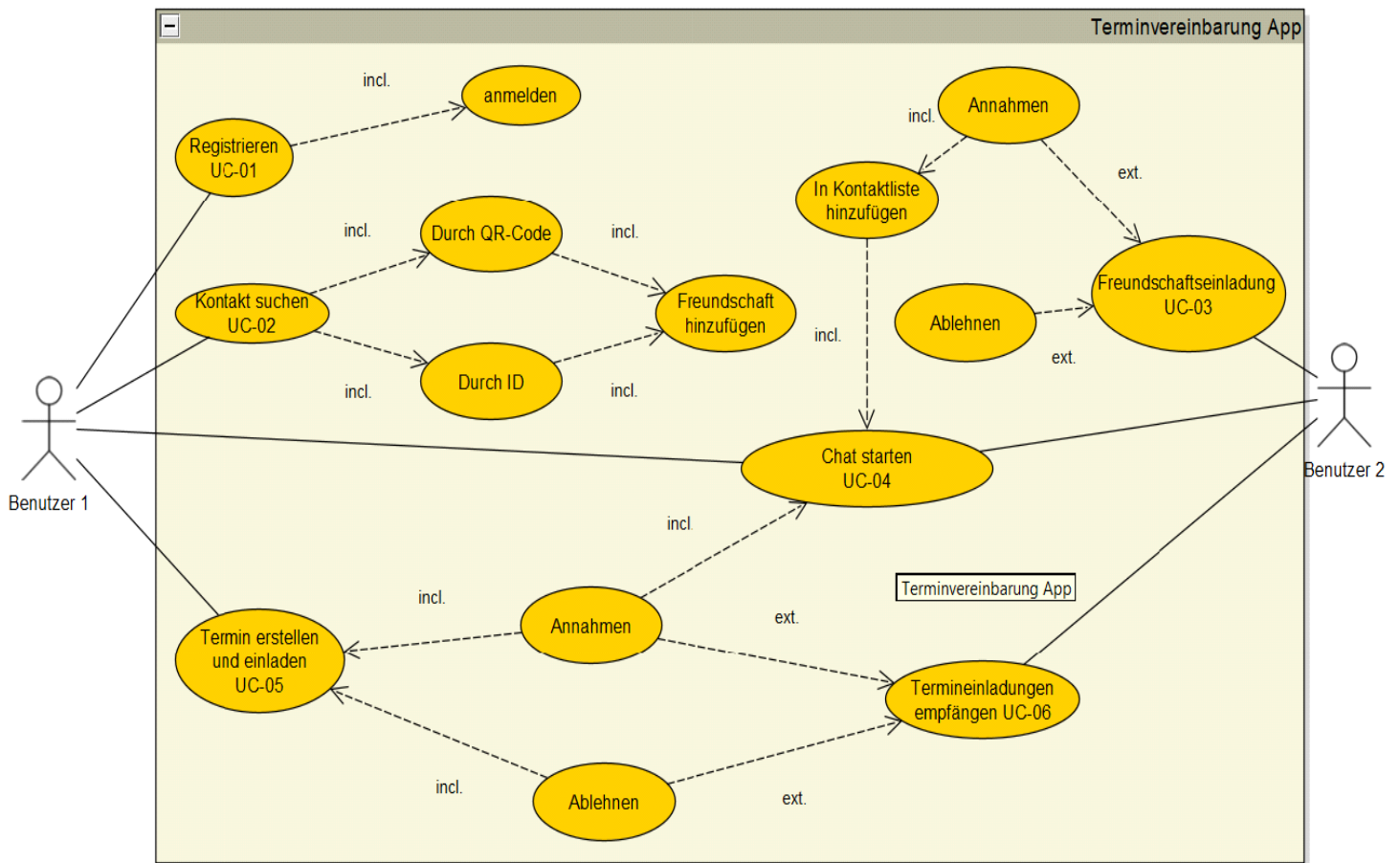


Abbildung 2 : Use Case Diagram

- Die folgende Tabellen beschreiben das Standardablauf der Use case Komponenten:

Informationsart	Beschreibung
Identifikation	UC-01
Beschreibung	Registrieren
Akteure	Benutzer
Vorbedingungen	Benutzer besitzt eine gültige E-Mail-Adresse und verfügt über einen Zugang zum Internet.
Standardablauf	<ul style="list-style-type: none"> • Registrierens Maske erscheint. • Benutzer gibt seine E-Mail-Adresse und sein Passwort ein. • System überprüft, ob die E-Mail-Adresse schon registriert ist und ob sie gültig ist und dann sendet einen Bestätigungslink zu dem Benutzer. • Der Benutzer soll durch einen Link, seine Registrierung bestätigen.
Alternative Abläufe	Benutzer kann sich durch social-Media Account registrieren.
Nachbedingungen	Bei erfolgter Bestätigung werden die Benutzerdaten dauerhaft serverseitig und persistent clientseitig gespeichert.
Bemerkung	Vorhandene E-Mail-Adresse ist Voraussetzung zur Registrierung.

Informationsart	Beschreibung
Identifikation	UC-02
Beschreibung	Kontakt suchen
Akteure	Benutzer
Vorbedingungen	Benutzer und die gesuchte Person müssen einen gültigen Account in der App besitzen.
Standardablauf	<ul style="list-style-type: none"> • Kontakt suchen Symbole anklicken. • Durch QR-Code Kontakt suchen. • Gesuchte Kontakt hinzufügen.
Alternative Abläufe	Durch ID (Nickname) /E-mail kontakt suchen
Nachbedingungen	<ul style="list-style-type: none"> • gesuchte Kontakte werden in der Suche erscheinen <u>wenn sie schon registriert sind.</u>
Bemerkung	Der App hat Rechts auf die Kamera zugreifen, damit der Benutzer den QR-Code durch Kamera scannen kann.

Informationsart	Beschreibung
Identifikation	UC-03
Beschreibung	Freundschaftseinladung
Akteure	Benutzer
Vorbedingungen	Benutzer und die gesuchte Person müssen einen gültigen Account in der App besitzen.
Standardablauf	<ul style="list-style-type: none"> • standardablauf von UC-02 • Freundschaftseinladung anklicken
Alternative Abläufe	Freundschaftseinladung an den anderen Meeting-Teilnehmern senden (also man braucht kein Kontakt(User) suchen weil die beide in eine gemeinsame Meeting sind)
Nachbedingungen	<ul style="list-style-type: none"> • Freundschaftseinladung wurde geschickt. • Solange der user die Freundschaftseinladung nicht akzeptiert/abgelehnt hat , kann man keine neue Einladung nochmal schicken.
Bemerkung	

Informationsart	Beschreibung
Identifikation	UC-04
Beschreibung	Chat starten
Akteure	Benutzer
Vorbedingungen	Benutzer muss befreundet mit dem anderen Benutzer sein.
Standardablauf	<ul style="list-style-type: none"> • Im kontaktliste gewünschte Freund auswählen. • Chat symbole anklicken. • Chat Starten.
Alternative Abläufe	Man kann auch im Chats liste Chats starten.
Nachbedingungen	termin-chat : Nachrichten werden allen Meeting Teilnehmer angezeigt user-chat : Nachrichten werden den beabsichtigen user angezeigt
Bemerkung	Dateien und Links können auch im chat geteilt werden.

Informationsart	Beschreibung
Identifikation	UC-05
Beschreibung	Termin erstellen und einladen
Akteure	Benutzer
Vorbedingungen	Benutzer muss befreundet mit dem anderen Benutzer sein
Standardablauf	<ul style="list-style-type: none"> • Termin erstellen symbole anklicken. • Datum und Uhrzeit auswählen und Termin erstellen. • Freunde zum Termin einladen.
Alternative Abläufe	Users können auch durch ein QR-Code zu dem Termin eingeladen werden .
Nachbedingungen	Nachdem der Termin erstellt wurde , werden die Teilnehmer ein Termin Einladung bekommen
Bemerkung	Datum wird durch einen Kalender ausgewählt.

Informationsart	Beschreibung
Identifikation	UC-06
Beschreibung	Termineinladungen Empfangen
Akteure	Benutzer
Vorbedingungen	Benutzer muss befreundet mit dem anderen Benutzer sein.
Standardablauf	<ul style="list-style-type: none"> • Termineinladungen Liste erscheint. • Benutzer kann die Einladung annahmen oder ablehnen.
Alternative Abläufe	User kann den Termin QR-Code scannen .
Nachbedingungen	<ul style="list-style-type: none"> • Einladung annehmen : user wurde als Termin Teilnehmer hinzugefügt • Einladung ablehnen : user wurde nicht an der Termin teilnehmer und die Einladung wurde gelöscht
Bemerkung	kein

1.3 Qualitätsanforderungen

In Tabelle sind die wichtigste sechs Qualitätsanforderungen für das App:

Ziel	Beschreibung
Verfügbarkeit	Die App soll für den Nutzer zu jeder Tageszeit und ohne geografische Einschränkungen zugänglich sein.
Erlernbarkeit	Die GUI wird relativ simpel zu verstehen sein, sodass neue Benutzer alle Funktionalitäten die die App bietet, schnell erfassen können.
Skalierbarkeit	Die Leistung der App ändert sich nicht, selbst wenn eine große Anzahl von Benutzern die App gleichzeitig verwendet.
Sicherheit	Die App ist sicher und respektiert die Privatsphäre jedes Benutzers.
Betreibbarkeit	Der App wurde als IOS sowie Android verfügbar sein werden.
Attraktivität	Die GUI muss für alle Altersgruppen modern und attraktiv sein.

1.4 Testszenarien

Hier sind alle Test-Szenarien, die wir durchführen sollen, damit wir überprüfen, ob das App funktionsfähig ist.

Test-Szenario	Überprüfen Sie, ob der Nutzer die Sign-In-Funktionalität verwenden kann, um sich in seinem Konto einzuloggen.
Testfall 1	Anmeldung mit gültigen Daten.
Testfall 2	Anmeldung ohne gültige Daten.
Testfall 3	Anmelden Session Timeout.
Testfall 4	Melden Sie sich an, um das Passwort zurückzusetzen.
Testfall 5	Anmeldung obwohl der User noch angemeldet ist.

Test-Szenario	Termine verwalten
Testfall 1	Termine mit gültige/ungültige Daten hinzufügen.
Testfall 2	Termine mit abgelaufenem Datum hinzufügen.
Testfall 3	zwei Termine mit Duplikate Daten hinzufügen.
Testfall 4	Termin Löschen.
Testfall 5	Der Termin Vereinbarer ändert sein Termin Daten (Tag, Zeit, Ort, Datum und Infotext).
Testfall 6	ein Termin Teilnehmer ändert der Termin Daten.

Test-Szenario	Chat System
Testfall 1	Nachrichten Lassen
Testfall 2	Dateien im Chat Hochladen (Fotos, Videos und Pfd Files).
Testfall 3	Dateien im Chat Herunterladen.
Testfall 4	Markieren Sie ein Mitglied des Meetings im Chat.
Testfall 5	Markiere ein Nichtmitglied im Chat.
Testfall 6	Links im Chat hinzufügen (Links müssen gefiltert werden).

Test-Szenario	Notifikation: wir müssen sicherstellen, dass der User eine Benachrichtigung bekommt, wenn folgenden Aktionen erfolgen.
Testfall 1	eine Einladung/Nachricht bekommen.
Testfall 3	Jemand hat im Chat geschrieben.
Testfall 4	ein Termin Erinnerung.
Testfall 5	ein Termin ist abgelaufen.
Testfall 6	Ein Freundschaft Einladung bekommen.
Testfall 7	Termin eingenommen/abgelehnt.
Testfall 9	Freundschaft eingenommen/abgelehnt.

Test-Szenario	Kontakte verwalten
Testfall 1	Kontakte von Handy hinzufügen (wenn registriert und wenn nicht).
Testfall 2	Kontakte von social Media hinzufügen (wenn registriert und wenn nicht).
Testfall 3	Kontakte mit E-Mail-Adresse hinzufügen (wenn registriert und wenn nicht).
Testfall 4	User zweimal hinzufügen.

Test-Szenario	Kontakt suchen und hinzufügen
Testfall 1	mit QR-Code.
Testfall 2	Ungültiger QR-Code.
Testfall 3	mit ID-Nickname.
Testfall 4	Ungültiger ID-Nickname.
Testfall 5	mit gültiger E-Mail/ social media Account.
Testfall 6	Ungültiger E-Mail.

2 Entwurf

2.1 Fachliche Strukturen und Modelle

2.1.1 ER Diagram :

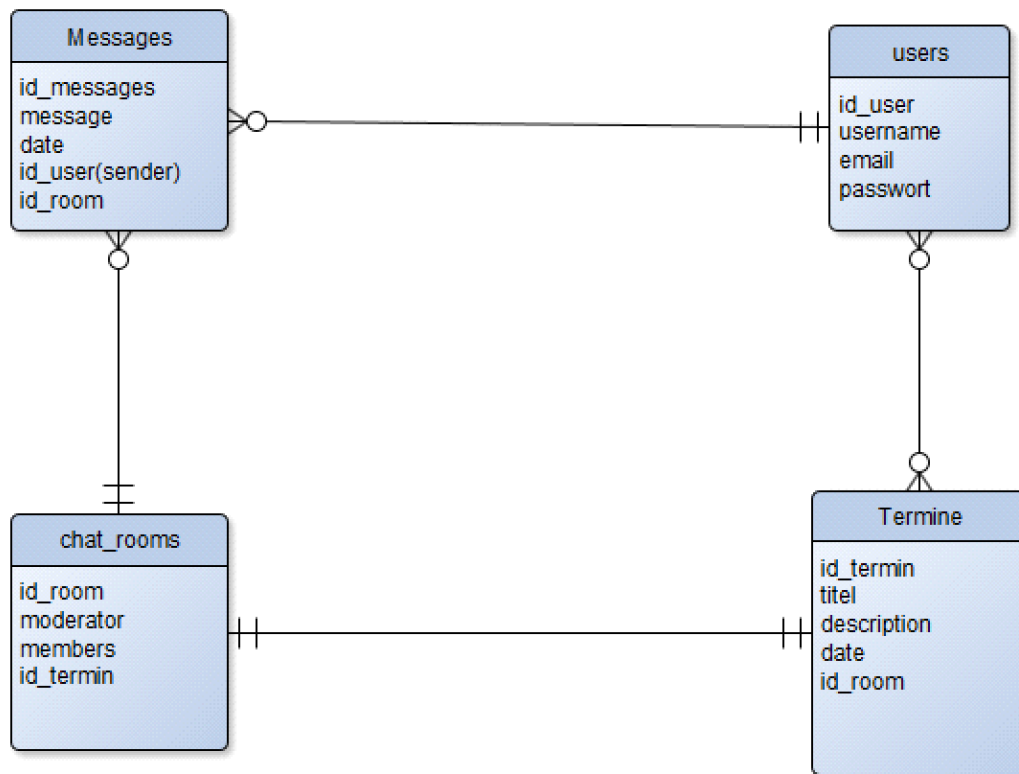


Abbildung 3 : Entity Relationship Diagramm

2.1.2 Beschreibung von Beziehungen :

→ **chat_rooms - Termine** : (1 to 1) jede Termin hat ein einzigartig chat room, wo der Termin teilnehmer miteinander kommunizieren können.

→ **chat_rooms - messages** : (1 to m) Ein Chatroom kann mehr als eine Nachricht enthalten, aber eine Nachricht mit einer einzigartigen ID kann nur in einem Chatroom existieren.

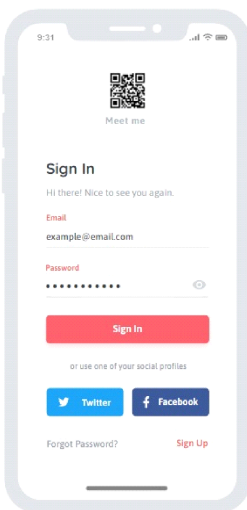
→ **users - messages** : (1 to m) Ein Benutzer kann eine unbegrenzte Anzahl von Nachrichten senden, aber eine Nachricht mit einer eindeutigen ID kann nur von einem Benutzer gesendet werden

→ **users - termine** : (n to m) Ein Termin kann einen oder mehrere Benutzer enthalten (+ Moderator) und ein Benutzer kann an mehr als einem Termin teilnehmen .

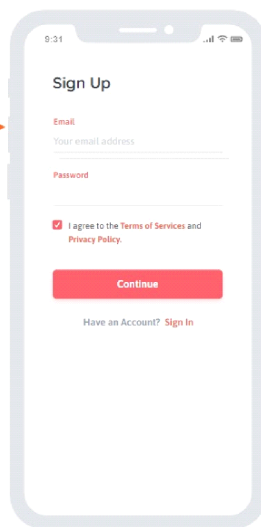
Für diese Beziehung sollten wir eine Tabelle mit folgenden Attributen erstellen: id, id_user, id_termin

2.2 Benutzeroberfläche

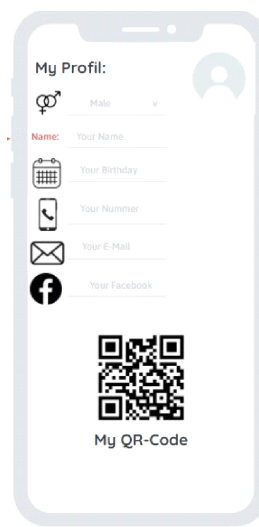
Anmeldung:



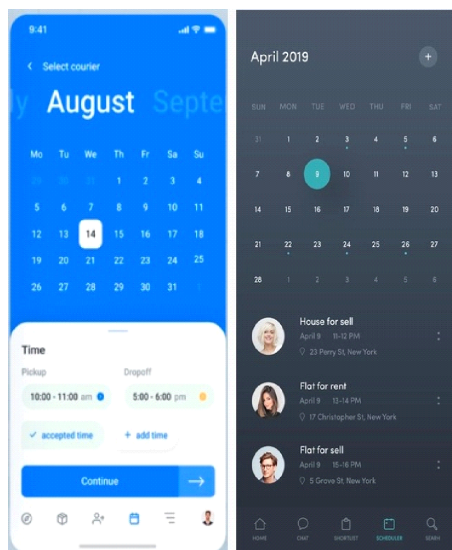
Registrieren:



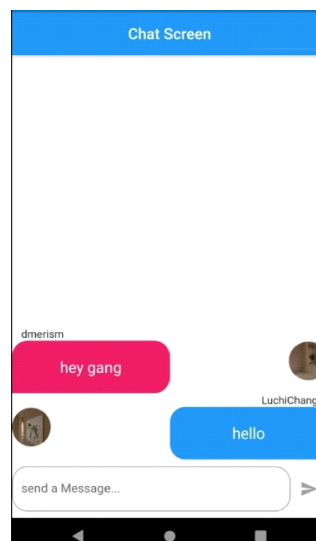
Profil:



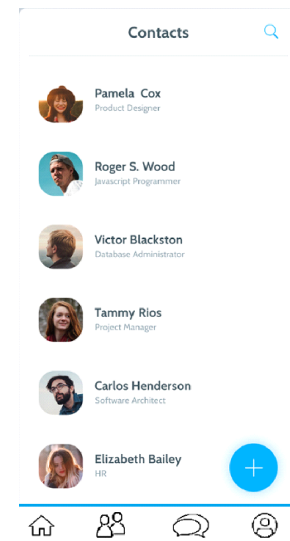
Startseite:



Chat:



Kontakt liste:



2.3 Systemarchitektur

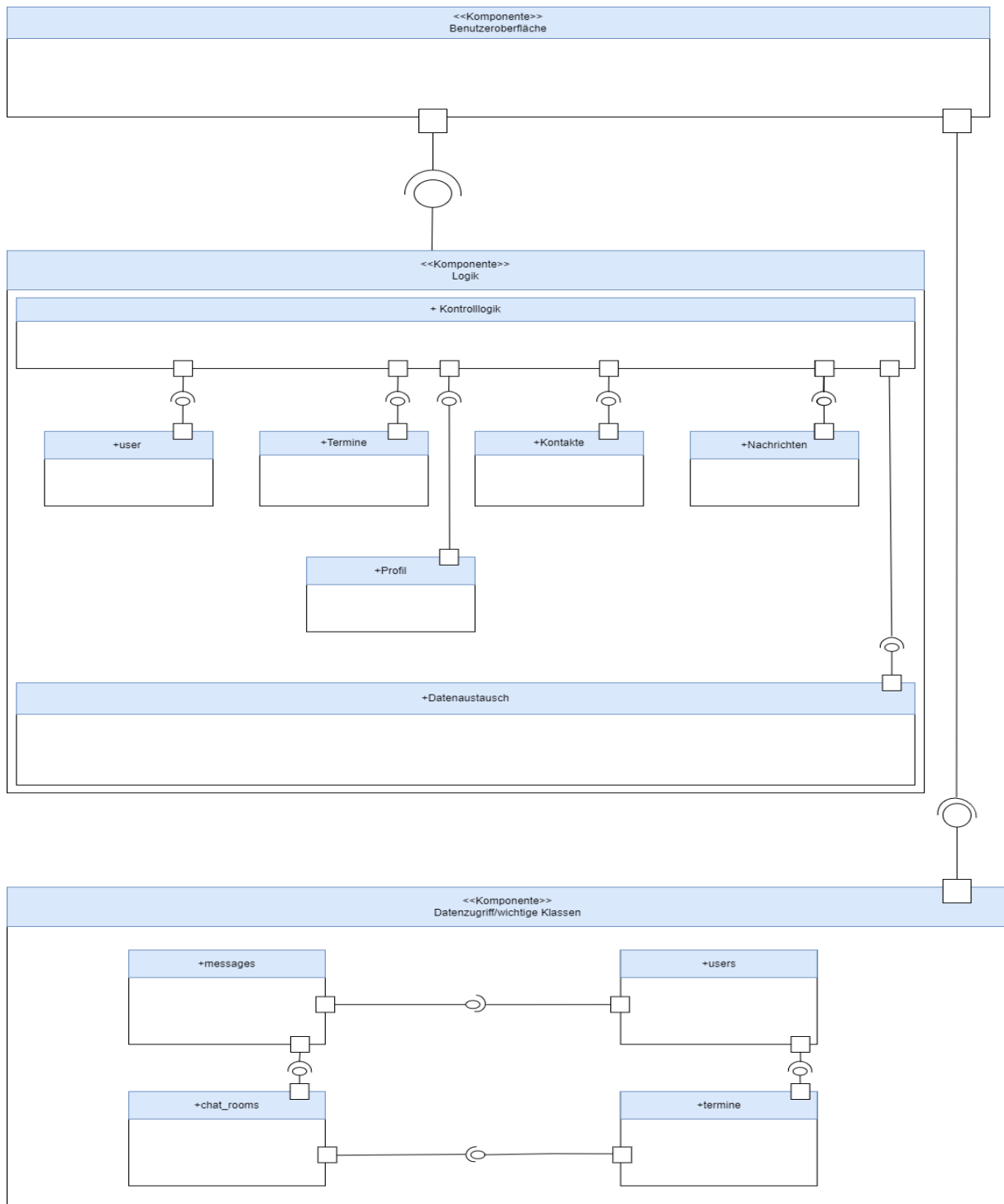


Abbildung 4 : Komponenten Diagramm

2.4 Technologiestack und Lösungsansätze für die Umsetzung der Qualitätsanforderungen

Ziel	Passende Lösungsansätze
Verfügbarkeit	<ul style="list-style-type: none">• Back4App
Sicherheit	<ul style="list-style-type: none">• Passwords werden in Datenbank encrypted• Benutzerrechte werden eingeschränkt
Skalierbarkeit	<ul style="list-style-type: none">• Implementieren von Datenbanken die die möglichkeit bietet eine grosse menge von Daten zu speichern => MongoDB im Back4App (Parse server)
Erlernbarkeit	<ul style="list-style-type: none">• benutzen von vereinfachte Widgets• weniger Text und weniger Informationen pro Seite
Betreibbarkeit	<ul style="list-style-type: none">• API26• IOS13
Attraktivität	<ul style="list-style-type: none">• Flutter und ein Paar dazugehöriger Packages (Flutter Calendar, Flutter Chat ...)

2.5 Lizenz Modell

Alle benutzten Komponenten sind mit **OPEN Source** Lizenz, weil :

- Das App kostenlos ist und immer verfügbar ist
- Open Sourcing bringt der gesamten Community ein nützliches Ergebnis.
- Alle benutzten bei der App-Entwicklung mitwirken können.

3. Aufteilung der Kapitel

Das Technische Spezifikation wurde als Teamarbeit erledigt.