

Messaging-Applikation Lastenheft

INF202-Projekt

Eren Naci Odabaşı

180501038

TMS-4

e180501038@stud.tau.edu.tr

Nihat Akın

180501008

TMS-4

e180501008@stud.tau.edu.tr



Inhaltsverzeichnis

1. Customer Story	2
2. Kundenanforderungen	2
3. Use Cases	3
4. Systemverhalten	3
5. Systemarchitektur	4
6. GUI Prototyp	5
7. Physikalische Architektur	6

1. Customer Story

- Der Kunde möchte eine vollständig skalierbare Messaging-/Organisationsanwendung sowohl für den persönlichen als auch für den organisatorischen Gebrauch.
- Die Anwendung sollte es ihren Nutzern ermöglichen, über persönliche Chats und Gruppenchats miteinander zu kommunizieren.
 - Außerdem sollte sie es dem Nutzer ermöglichen, seine Veranstaltungen zu organisieren.
- Die Anwendungsumgebung ist sowohl der Desktop als auch der Server.

2. Kundenanforderungen

Anforderungen des Kunden für das echtseitig-Messaging Applikation sind wie folgendes:

1. Sofortige Nachrichtenübermittlung: Die Messaging-Anwendung muss den Benutzern Instant-Messaging-Funktionen zur Verfügung stellen, damit sie Nachrichten in Echtzeit senden und empfangen können.
2. Gruppen-Nachrichtenübermittlung: Die Benutzer müssen die Möglichkeit haben, Gruppenchats zu erstellen und daran teilzunehmen, in denen mehrere Benutzer Nachrichten senden und empfangen können.
3. Verschlüsselung: Die Messaging-Anwendung muss eine Ende-zu-Ende-Verschlüsselung bieten, um sicherzustellen, dass Nachrichten sicher sind und nicht von unbefugten Benutzern abgefangen oder eingesehen werden können.
4. Nachrichtenverlauf: Benutzer müssen auf den Nachrichtenverlauf zugreifen können, einschließlich früherer Unterhaltungen und gesendeter/empfangener Nachrichten.
5. Benachrichtigungen: Die Messaging-Anwendung muss Push-Benachrichtigungen für Benutzer bereitstellen, wenn sie neue Nachrichten erhalten.

3. Use Cases

Verschiedene Anwendungsfälle für eine Echtzeit-Messaging-Anwendung, die für den internen Gebrauch entwickelt wurde, sind im folgenden Diagramm dargestellt:

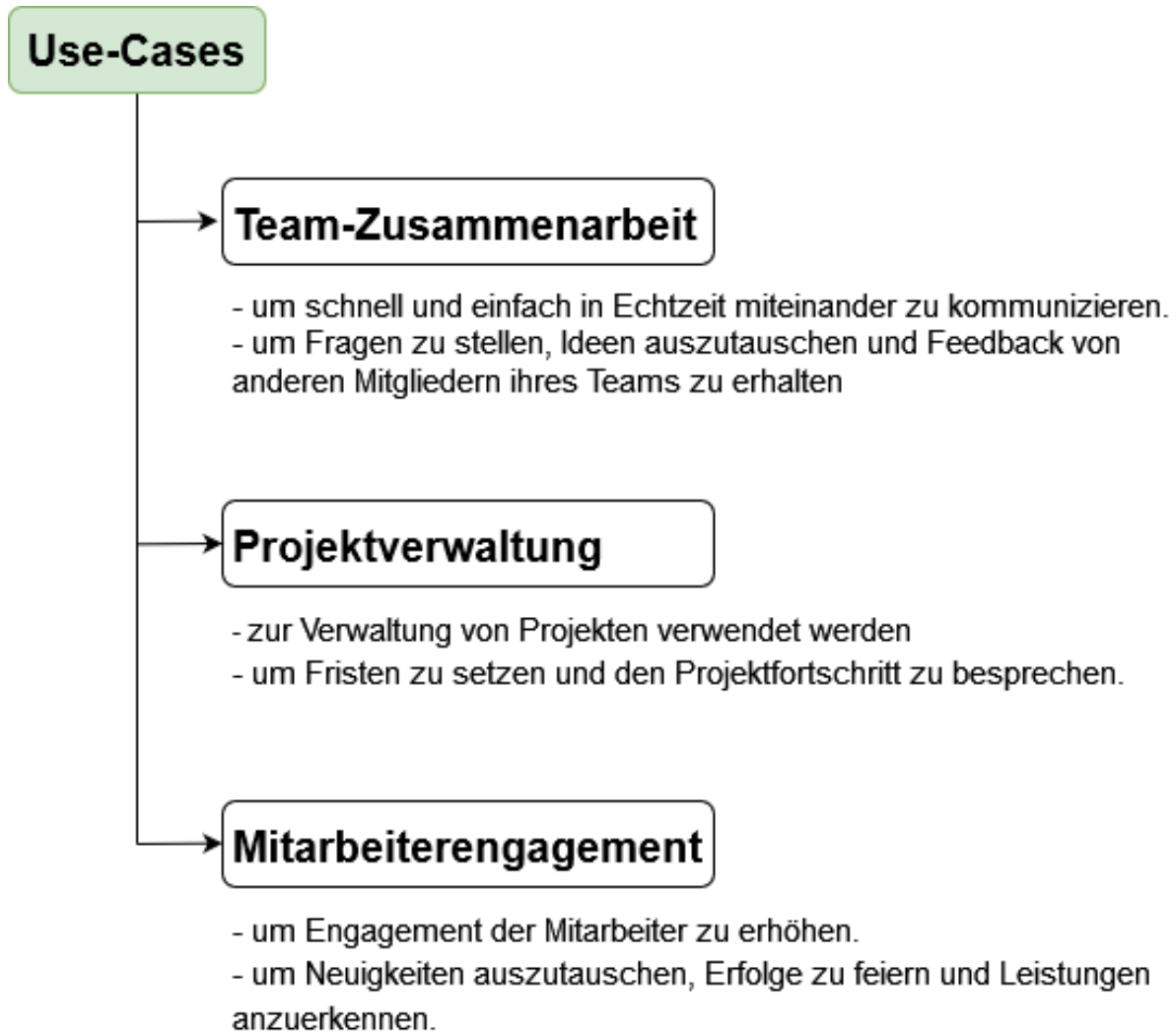


Abbildung 1: Use-Cases

4. Systemverhalten

Eine Echtzeit-Messaging-Anwendung besteht in der Regel aus Client-Anwendungen, einem Messaging-Dienst, einer Datenbank und APIs. Der Zustandsfluss einer Nachricht in einer Messaging-Anwendung besteht in der Regel aus den Zuständen Leerlauf, Senden, Warteschlange, Zustellung, Lesen und Löschen.

In folgende Zustandsfluss kann man sehen, dass das System 5 verschiedene Zustand hast. Nachdem Initial-Zustand wird das System zu Idle-Zustand gehen durch Server-Connection zu starten. Wenn das System nicht ausgeschaltet ist, kann anschließend je nach Wunsch des Benutzers eine Nachricht übermittelt oder der Zugang gewährt werden.

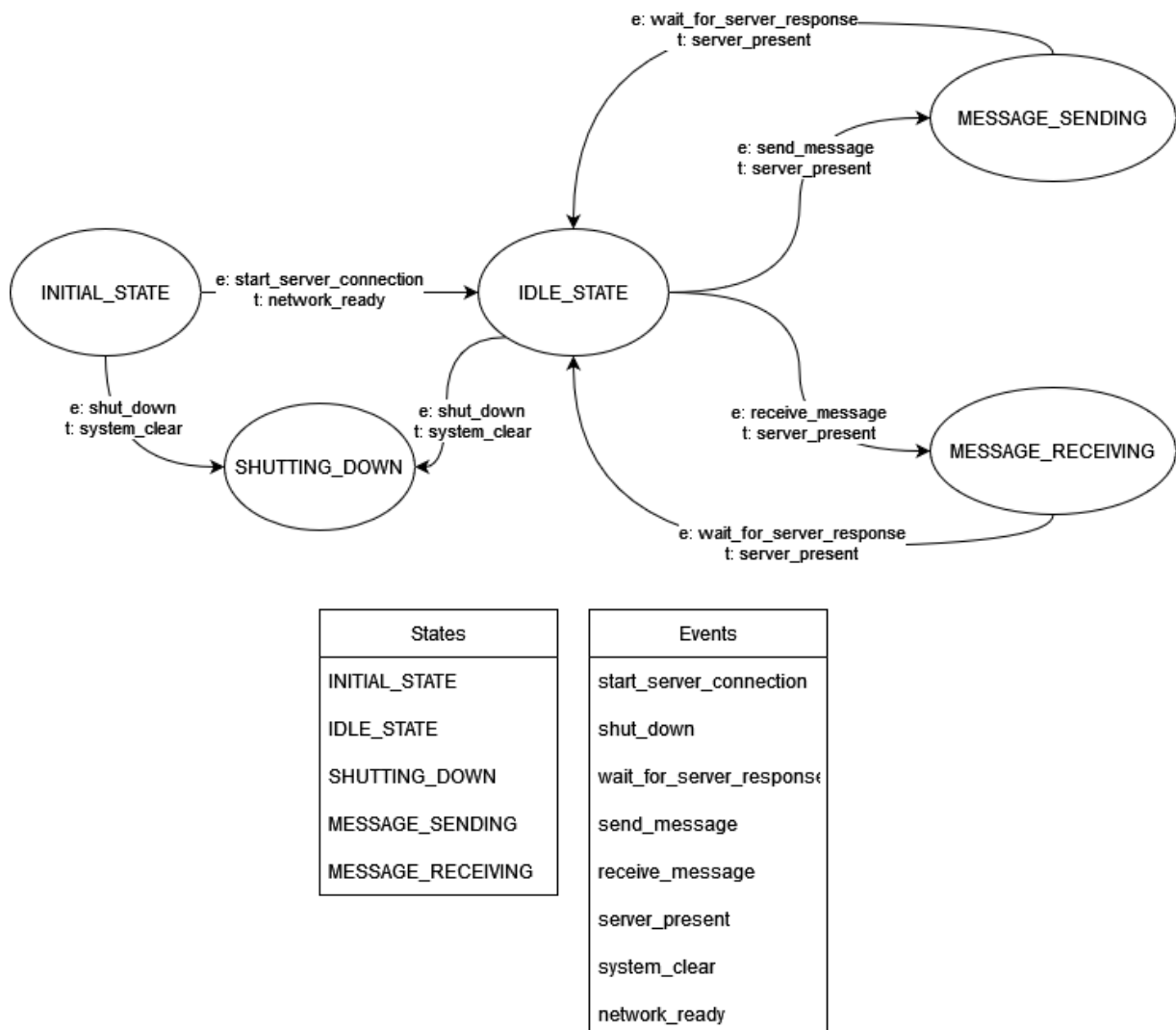


Abbildung 2: Systemverhalten

5. Systemarchitektur

Clients: Dies sind die Anwendungen, die die Endnutzer zum Senden und Empfangen von Nachrichten verwenden. Es kann sich um webbasierte oder native Anwendungen handeln, die auf Desktops, Laptops, Smartphones und Tablets laufen.

Server: Der Server ist die zentrale Komponente des Systems, die die Zustellung von Nachrichten verwaltet und den Zustand des Nachrichtensystems aufrechterhält.

Protokoll: Das Protokoll ist eine Reihe von Regeln, die festlegen, wie Nachrichten zwischen den Clients und dem Server übertragen werden.

Datenbank: Die Datenbank dient zur Speicherung von Benutzerdaten und Nachrichtenverläufen.

APIs: APIs bieten einen Kommunikationskanal zwischen den Clients und dem Server, über den sie Nachrichten senden und empfangen und den Nachrichtenverlauf abrufen können.

Lastausgleicher: Der Load Balancer verteilt den eingehenden Netzwerkverkehr auf mehrere Server, um sicherzustellen, dass der Messaging-Dienst eine große Anzahl von Benutzern und Nachrichten verarbeiten kann.

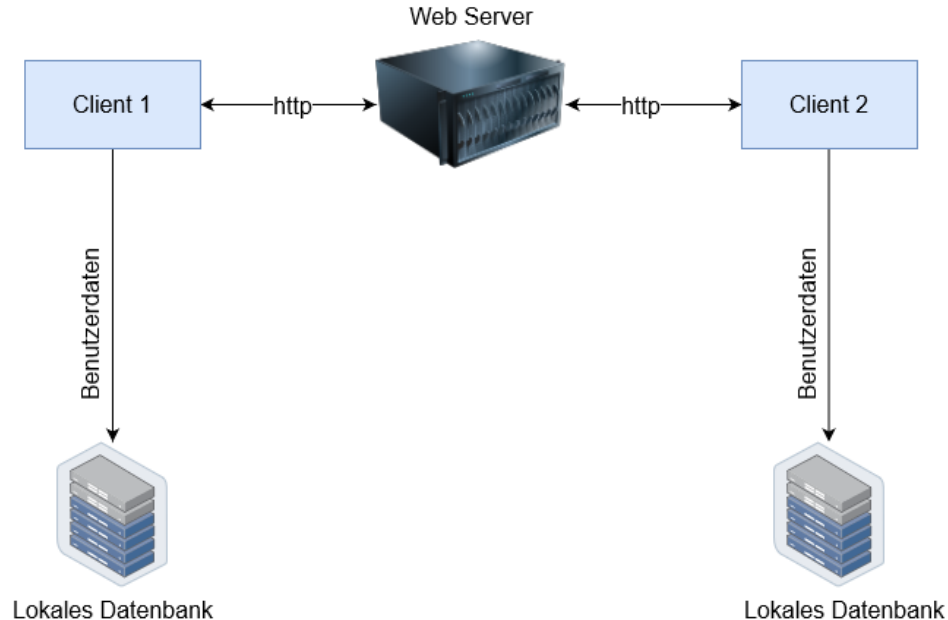


Abbildung 3: Systemarchitektur

6. GUI Prototyp

People

Groups

Tab 1
Tab 2

Event 1

Event Status/Time

Event 2

Event Status/Time

Event 3

Event Status/Time

Martı 2023						
	Pzt	Sal	Çar	Per	Cum	Cmt
9	27	28	1	2	3	4
10	6	7	8	9	10	11
11	13	14	15	16	17	18
12	20	21	22	23	24	25
13	27	28	29	30	31	1
14	3	4	5	6	7	8

Abbildung 4: GUI Prototyp

7. Physikalische Architektur

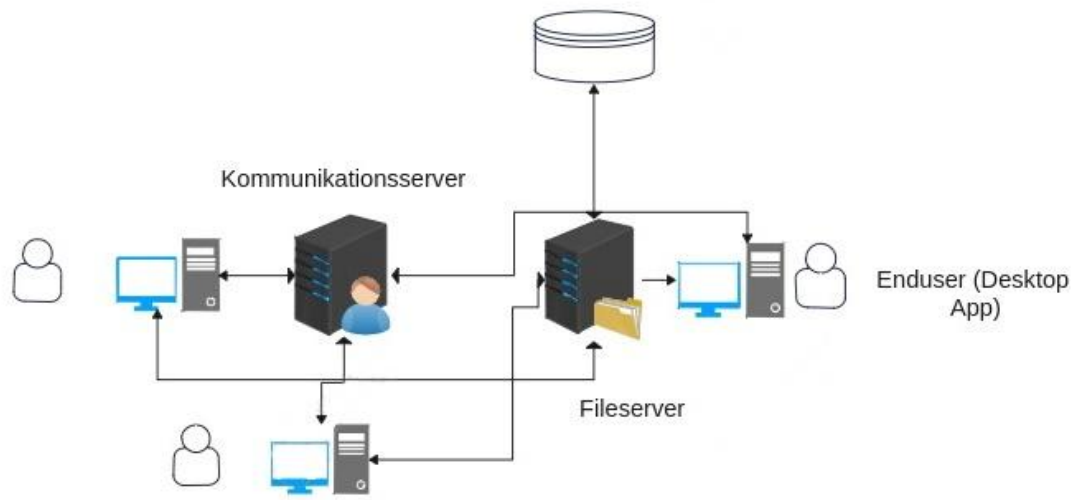


Abbildung 5: Physikalische Systemarchitektur