

Projektdokumentation

Schul.Cloud-Call

Datum der Abgabe:

04.01.2024

1. Anforderungsanalyse	3
2. Beschreibung des Projektes	4
3. Front-End	5
4. Benutzerhandbuch.....	5
Einleitung	5
Button-Platzierung und -Aussehen	5
Anruf tätigen	5
Problemlösung	6
5. Projektablauf.....	6
6. Testverfahren.....	6
7. Reflektion	7
Positive Aspekte:.....	7
Herausforderungen und weniger erfolgreiche Aspekte:	7
Schlussfolgerung	8
Anhang	9

1. Anforderungsanalyse

1. Ziel des Projekts:

- Hinzufügen eines Call Buttons in eine bestehende Benutzeroberfläche einer All-In-One Plattform namens "BBZ-Cloud".
- Integration mit der schul.cloud-API, um Nachrichten zu senden und mit einem Chatbot zu interagieren.

2. Funktionalitäten des Call Buttons:

- Initiierung einer API-Anfrage an Schul.Cloud beim Drücken des Buttons.
- Kommunikation mit dem Chatbot über die API, um einen Meetings-Link zu generieren.
- Versenden des Meetings-Links an den anfragenden Benutzer.

3. Benutzeroberfläche:

- Integration des Call Buttons in die vorhandene Benutzeroberfläche der Plattform.
- Visuelle Darstellung des Buttons für eine benutzerfreundliche Bedienung.

4. Kommunikation mit Schul.Cloud-API:

- Umsetzung in JavaScript, basierend auf einem vorhandenen Python-Skript.
- Verwendung von Authentifizierungsmechanismen, falls erforderlich.

5. Chatbot-Interaktion:

- Integration mit dem Chatbot zur Generierung des Meetings-Links.
- Berücksichtigung von Fehlerbehandlungen und Rückmeldungen für den Benutzer.

6. Anruf-Funktionalität:

- Öffnen eines neuen Fensters mit dem Meetings-Link beim Klicken des Call Buttons.
- Implementierung der Schnittstelle für die Anrufdurchführung.

7. Dokumentation:

- Erstellung einer umfassenden Dokumentation für Entwickler und Benutzer.
- Erläuterung der Implementierung, Schnittstellen, und Anwendungsfälle.

8. Testing:

- Durchführung von umfassenden Tests, um die Funktionalitäten und Sicherheit zu gewährleisten.
- Identifizierung und Behebung von etwaigen Bugs oder Fehlfunktionen.

2. Beschreibung des Projektes

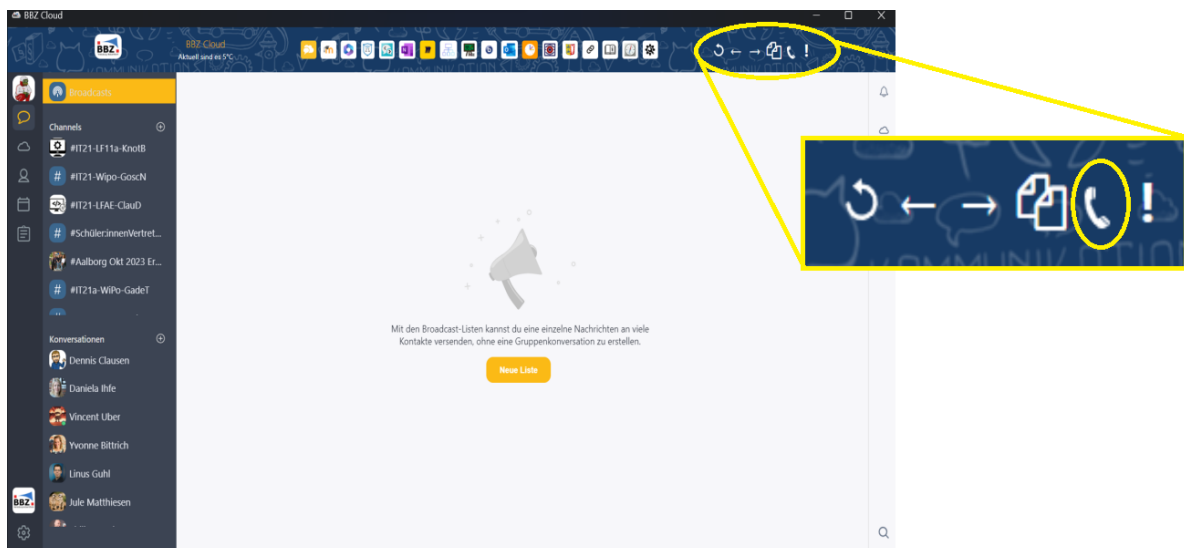
Das Ziel dieses Projekts ist die nahtlose Integration eines Call Buttons in die existierende Benutzeroberfläche der All-In-One Plattform "BBZ-Cloud", die vorwiegend für die Kommunikation über Schul.Cloud genutzt wird. Der Call Button ermöglicht es Benutzern, über die Schul.Cloud-API mit einem Chatbot zu interagieren. Ziel ist die Generierung eines Meetings-Link und die Simulation eines Anrufs.

Die Funktionalitäten des Call Buttons umfassen die Initiierung einer API-Anfrage an Schul.Cloud, die Kommunikation mit dem Chatbot für die Generierung des Meetings-Links und die Zustellung des Links an den anfragenden Benutzer. Die Benutzeroberfläche wird um diesen Button erweitert, um eine benutzerfreundliche Bedienung zu gewährleisten.

Die Umsetzung erfolgt in JavaScript, basierend auf einem vorhandenen Python-Skript für die Schul.Cloud-API. Hierbei werden Authentifizierungsmechanismen berücksichtigt, um die sichere Durchführung der API-Anfragen zu gewährleisten.

Die Interaktion mit dem Chatbot beinhaltet klare Rückmeldungen und Fehlerbehandlungen, um eine reibungslose Kommunikation sicherzustellen. Beim Klicken des Call Buttons öffnet sich ein neues Fenster mit dem generierten Meetings-Link, wodurch Benutzer einen simulierten Anruf durchführen und dem Meeting beitreten können.

3. Front-End



Hier ist Oberfläche der Schul.Cloud Applikation zu sehen. Der Anruf Button befindet sich im Header, oben, am rechten Rand der Applikation / des Browsers. Es handelt sich um das Telefon-Symbol (Gelb Markiert).

4. Benutzerhandbuch

Einleitung

Herzlich willkommen zum Benutzerhandbuch für den Anruf-Button der Schul.Cloud! Dieser Button ermöglicht es Ihnen, nahtlos von einem Chat aus, einen Anruf zu tätigen und direkt mit Ihren Kontakten zu sprechen. Dieses Handbuch soll Ihnen eine einfache Anleitung bieten, wie Sie diesen Button nutzen können.

Button-Platzierung und -Aussehen

Der Anruf-Button befindet sich in der Schul.Cloud-Oberfläche im Header an der rechten Seite. Sie erkennen ihn an einem charakteristischen Telefonsymbol. Der Button ist darauf ausgelegt, leicht zugänglich zu sein, damit Sie spontan und unkompliziert Anrufe starten können.

Anruf tätigen

Um einen Anruf zu tätigen, gehen Sie folgendermaßen vor: 1.) Öffnen Sie den Chat mit der Person, die Sie anrufen möchten. 2.) Klicken Sie auf den Anruf-Button oben im Header. Dies sendet einen Besprechungslink zu ihrem Gesprächspartner. Dieser muss Ihrem Meeting nun

beitreten. *Stellen Sie sicher, dass Ihr Gerät über eine aktive Internetverbindung und die erforderlichen Berechtigungen für Mikrofon und Kamera verfügt.*

Problemlösung

Falls Sie auf Probleme stoßen oder Hilfe benötigen, überprüfen Sie bitte: 1.) Internetverbindung: Stellen Sie sicher, dass Ihre Internetverbindung stabil ist. 2.) Berechtigungen: Überprüfen Sie, ob der App / Browser die erforderlichen Berechtigungen für Mikrofon und Kamera hat.

Bei weiteren Fragen oder Problemen steht Ihnen die Schuleigene IT / bzw. Ihr Administrator zur Verfügung.

5. Projektablauf

- Einarbeitung in das Projekt
- Aufsetzen IDE (JS)
- Aufsetzen IDE (Python)
- Button in Oberfläche einbauen
- Testen d. API in Python
- Übersetzen d. API in JS
- Einbau API (Backend)
- Finale Testdurchläufe

6. Testverfahren

Das Testverfahren für das Projekt erfolgte kontinuierlich während der Implementierung, um die Funktionalität und Integrität des Systems sicherzustellen. Insbesondere wurden Tests durchgeführt, während neue Funktionen eingeführt wurden, wie beispielsweise die Integration des Call Buttons. Bei der Implementierung des Call Buttons wurde insbesondere darauf geachtet, dass das Klicken auf den Button wie erwartet funktioniert.

Bei der Implementierung der Kommunikation mit der StachCat API stießen wir auf mehrere Herausforderungen und Fehler. Aufgrund der fehlenden Dokumentation für die StachCat API haben wir intensiv Postman genutzt, um die API ausgiebig zu testen. Dies ermöglichte es uns,

mehr Informationen über die API-Funktionalität zu erhalten und die Interaktion mit den Endpunkten zu verstehen.

Die Testphasen mit Postman dienten so als Informationsquelle für die API-Funktionen. Durch das Testen der API in verschiedenen Szenarien konnten wir ein tieferes Verständnis für die Anforderungen und Möglichkeiten der StachCat API entwickeln.

7. Reflektion

Das Projekt war eine anspruchsvolle Herausforderung, die trotz vieler positiver Aspekte letztendlich nicht den gewünschten Erfolg erzielen konnte. In der Reflexion sollen sowohl die positiven als auch die weniger erfolgreichen Aspekte des Projekts beleuchtet werden.

Positive Aspekte:

- **Implementierung der Benutzeroberfläche und des Call Buttons:** Die Integration des Call Buttons in die bestehende Benutzeroberfläche wurde erfolgreich umgesetzt. Die visuelle Darstellung und die Benutzerinteraktion entsprachen den Erwartungen.
- **Kommunikation mit StachCat API in JavaScript:** Die Implementierung der Kommunikation zur StachCat API in JavaScript verlief erfolgreich. Die Umsetzung ermöglichte grundlegende Interaktionen mit der API.
- **Kontinuierliche Tests während der Implementierung:** Die Entscheidung, Tests kontinuierlich während der Implementierung durchzuführen, war positiv. Frühzeitige Fehlererkennung und Behebung trugen zur Entwicklung einer stabilen Anwendung bei.

Herausforderungen und weniger erfolgreiche Aspekte:

- **Fehlende Dokumentation für StachCat API:** Die größte Hürde im Projekt war das Fehlen einer ausreichenden Dokumentation für die StachCat API. Dies führte zu Schwierigkeiten bei der genauen Funktionsweise der API und wie sie mit der schul.cloud kommuniziert.
- **Erfolglose Tests mit Postman und anderen Projekten¹:** Trotz ausgiebiger Tests mit Postman und anderen Projekten, die die gleiche API nutzen, konnten wir keine

¹ A.1: Bild der Fehlermeldung - "Postman Fehlermeldung"

zufriedenstellenden Ergebnisse erzielen. Das Fehlen von klaren Informationen über die API-Funktionen beeinträchtigte den Fortschritt des Projekts erheblich.

- **Zeitliche Beschränkungen und Informationsmangel:** Die begrenzte Zeit und der Mangel an detaillierten Informationen zur StachCat API waren entscheidend für das Scheitern des Projekts. Ein tieferes Verständnis für die API-Funktionalität und die genaue Ansteuerung fehlten, um das gewünschte Ergebnis zu erreichen.

Schlussfolgerung:

Insgesamt war das Projekt in einem guten Zustand, was die implementierten Funktionen betrifft. Die Unfähigkeit, die StachCat API effektiv für die Kommunikation mit schul.cloud zu nutzen, war jedoch entscheidend für das Scheitern. Zukünftige Projekte sollten eine sorgfältige Vorabrecherche und Klärung von Schnittstellenanforderungen beinhalten, um möglichen Herausforderungen frühzeitig entgegenzuwirken. Der Lerneffekt aus diesem Projekt liegt in der Erkenntnis, dass eine klare Dokumentation von externen APIs von entscheidender Bedeutung ist und ein tieferes Verständnis für deren Funktionalität unerlässlich ist, um erfolgreich zu sein.

Anhang

A.1 "Postman Fehlermeldung"

```
1  {}  
2  "status": {  
3    "value": "FAILED",  
4    "short_message": "missing_values",  
5    "message": "missing values, credentials and app data required"  
6  },  
7  "payload": {  
8    "client_key": false
```

Quellcode: <https://github.com/faabiii/bbz-cloud>