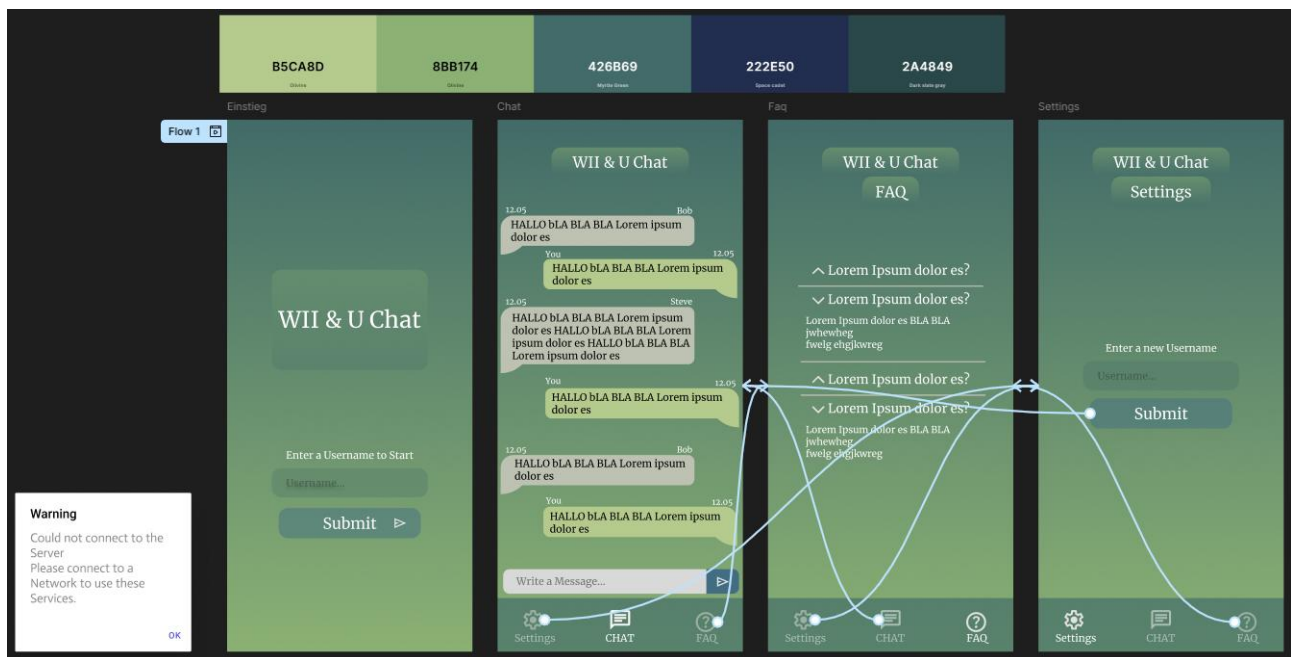


MEINE APP

Wii & U Chat

Version 1.0.0, Datum: 02. Juli 2025

Printscreen unserer App «Wii & U Chat» [Link zum interaktiven Mockup](#)



ABSTRACT (KURZBESCHREIBUNG)

Unsere App fokussiert auf alle Funktionen, die für eine nötige Chat App gebraucht werden. In unserem üK haben wir mit React Native Expo gearbeitet und verschiedene Funktionen wie Navigationen oder Komponente kennengelernt. Unsere App ist eine kleine Chat App, die das gelernte umgesetzt wird.

Die App beinhaltet folgende Seiten:

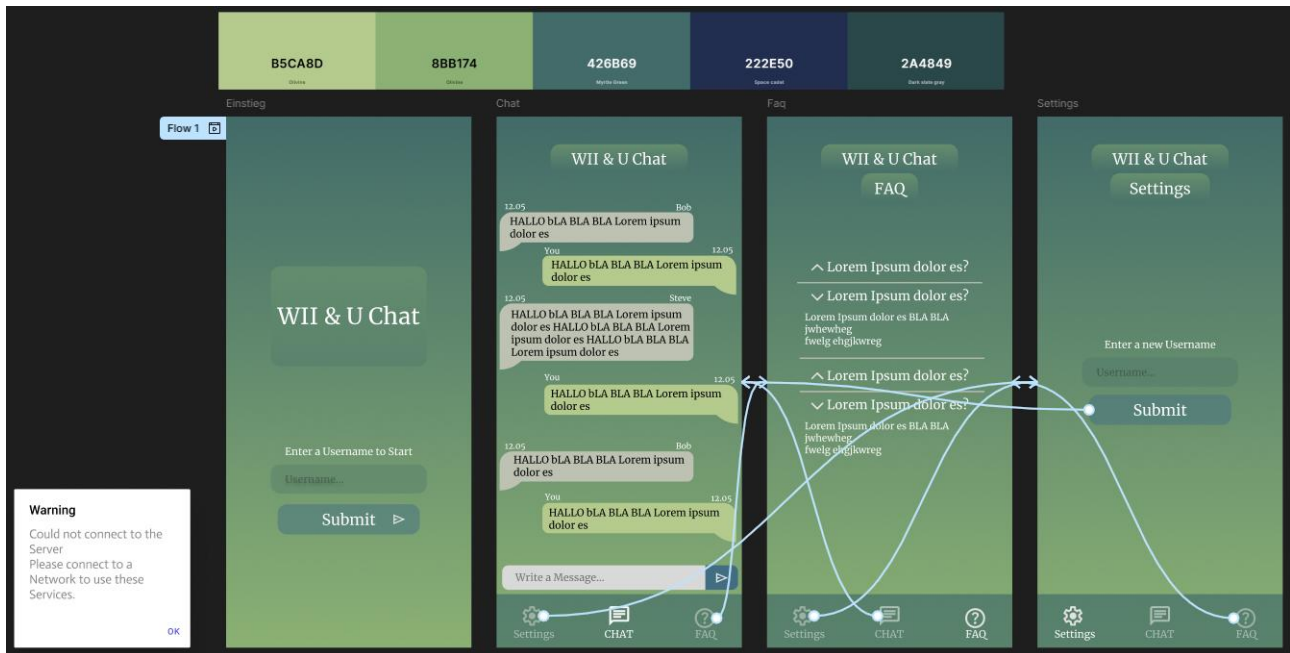
- Einstiegsseite: Verantwortlich für die initiale Einrichtung und die Eingabe des Benutzernamens, bevor der Nutzer in den Chatraum gelangt.
- Chat-Seite: Ermöglicht den Austausch von Nachrichten in Echtzeit.
- FAQ-Seite: Bietet Hilfestellungen und beantwortet häufig gestellte Fragen.
- Settings-Seite: Ermöglicht das nachträgliche Ändern des Benutzernamens und weitere Anpassungen.

USER STORIES

An dieser Stelle alle Anforderungen als User Stories auflisten:

- Als **Nutzer** möchte ich einen Benutzernamen auswählen, damit ich mich im Chatroom identifizieren kann.
- Als **Nutzer** möchte ich direkt nach der Anmeldung in den gemeinsamen Chatraum weitergeleitet werden, damit ich sofort mit anderen kommunizieren kann.
- Als **Nutzer** möchte ich die letzten Nachrichten im Chatroom sehen können, damit ich den aktuellen Gesprächsverlauf verfolgen kann.
- Als **Nutzer** möchte ich neue Nachrichten senden können, damit ich mich aktiv am Chat beteiligen kann.
- Als **Nutzer** möchte ich über eine Bottom Navigation einfach zwischen Chat, Einstellungen und FAQ wechseln können, damit ich schnell auf die wichtigsten Bereiche zugreifen kann.
- Als **Nutzer** möchte ich meinen Benutzernamen jederzeit in den Einstellungen ändern können, damit ich flexibel meinen Namen anpassen kann.
- Als **Nutzer** möchte ich eine FAQ-Seite aufrufen können, damit ich Antworten auf häufige Fragen und Hilfestellungen bekomme.
- Als **Nutzer** möchte ich eine Einstiegsseite zur ersten Einrichtung sehen, damit der Anmeldeprozess klar und einfach ist.

MOCKUPS



Unsere App besteht aus vier verschiedenen Seiten. Für die Farben haben wir eine Grün- bläuliche Farbpalette verwendet, da wir die Farben am besten passend gefunden haben. Beim Start der App kommt man direkt zur «Einstieg» Seite. Bei fehlender Internetverbindung poppt ein Alert automatisch auf. Nachdem man seinen Benutzernamen eingetragen hat im Textfeld kommt man mit dem Button «Submit» weiter zur «Chat» Seite. Das ist die Hauptseite der App, da alles darauf basierend ist. Im Chat kann man Nachrichten von anderen Gruppenmitglieder lesen und selbst Nachrichten schreiben. Bei der Navigation unten (Bottom Navigation) kann man zu den Settings rübergehen. Dort kann man seinen Benutzernamen ändern, falls er jemanden nicht gefällt oder wenn man gelangweilt ist. Mit «Submit» kann man die Änderung speichern. Wenn man bei der Navigation auf «FAQ» (Frequently asked questions) drückt, kommt man zu einer Seite die häufig gestellten Fragen beantwortet. Da stehen die meisten Antworten drin.

TECHNISCHE REALISIERUNG

Die App wird mit **React Native** entwickelt. Mit der Bottom Navigation, Schriftgrösse, Farben & Kontrast gewährleisten wir mit dieser App eine sehr benutzerfreundliche und schnelle App.

Komponenten und Architektur

1. **Startseite (Einstieg):** Diese Seite ist die erste Interaktionsmöglichkeit, bei der der Benutzer seinen Benutzernamen eingeben kann. Wird ein Benutzername eingegeben und der "Submit"-Button betätigt, wird der Benutzer zur Chat-Seite weitergeleitet. Bei fehlender Internetverbindung wird ein Alert angezeigt, um den Benutzer über das Verbindungsproblem zu informieren.
2. **Chat-Seite:** Hier können Benutzer Nachrichten in Echtzeit sehen und selbst verfassen. Nachrichten werden über die **Realtime Database von Firebase** synchronisiert, sodass alle Gruppenmitglieder gleichzeitig die Nachrichten sehen können. Firebase bietet hier eine effiziente Lösung für Echtzeit-Daten-Updates, ohne dass die App manuell aktualisiert werden muss.
3. **Bottom Navigation:** Die Navigation besteht aus drei Schaltflächen: Chat, Settings und FAQ. Die aktuelle Seite wird durch die Navigation gewechselt, was durch **React (Bottom) Navigation** realisiert wird. Beim Wechsel zu den **Settings** können Benutzer ihren Benutzernamen ändern und mit dem «Submit»-Button die Änderung in der Firebase-Datenbank speichern. Bei Auswahl der **FAQ-Seite** werden häufig gestellte Fragen aus einer statischen JSON-Datei oder einer Firebase-Datenbank geladen und angezeigt.

Datenverwaltung und Speicherung

Die Benutzerdaten (wie z.B. der Benutzernamen) und Nachrichten werden in der **Realtime Database von Firebase** gespeichert. Dies ermöglicht eine nahtlose Synchronisation zwischen allen Geräten der Benutzer und gewährleistet, dass die Daten in Echtzeit aktualisiert werden.

Datenfluss:

- Bei der Eingabe eines Benutzernamens wird dieser an die Realtime Database gesendet und dort gespeichert.
- Nachrichten werden von der Realtime Database abgerufen und live auf der Chat-Seite angezeigt.
- Bei der Änderung des Benutzernamens wird die Firebase-Datenbank mit den neuen Informationen aktualisiert.

Fehlerbehandlung und Internetverbindung

- Bei einer fehlenden Internetverbindung zeigt ein Alert eine Warnung an, was durch Firebase's Realtime Database-Fehlerbehandlung ermöglicht wird. In diesem Fall wird die Verbindung zur Datenbank geprüft, und ein entsprechender Fehlerdialog wird angezeigt.

Die App nutzt Firebase's Echtzeit-Datenbank (Realtime Database), um eine effiziente und skalierbare Lösung für die Kommunikation zwischen den Benutzern zu bieten. Dadurch wird eine schnelle und zuverlässige Synchronisation von Nachrichten sowie Benutzerdaten gewährleistet.

TESTING

Abschnitt	Inhalt
ID	ST-001
Anforderungen	<ul style="list-style-type: none">- Der Benutzer muss sich mit einem Benutzernamen anmelden können- Nach der Anmeldung soll der Benutzer zur Chat-Seite weitergeleitet werden
Vorbedingungen	<ul style="list-style-type: none">- Die App ist geöffnet und der Benutzer befindet sich auf der «Einstieg» Seite- Es ist keine Internetverbindung erforderlich, da es kein Verbindungsproblem gibt
Ablauf	<ol style="list-style-type: none">1. Der Benutzer gibt einen Benutzernamen in das Textfeld auf der Einstieg-Seite ein2. Der Benutzer klickt auf den «Submit» Button3. Die App sollte den Benutzernamen validieren und zur Chat-Seite weiterleiten
Erwartetes Resultat	Der Benutzer wird erfolgreich zur Chat-Seite weitergeleitet, und der eingegebene Benutzername wird gespeichert

Abschnitt	Inhalt
ID	ST-002
Anforderungen	- Bei fehlender Internetverbindung wird ein Alert angezeigt, der den Benutzer informiert
Vorbedingungen	- Die App ist geöffnet und der Benutzer befindet sich auf der «Einstieg» Seite. - Die Internetverbindung wird absichtlich deaktiviert (z.B. Flugmodus aktivieren)
Ablauf	<ol style="list-style-type: none"> 1. Der Benutzer öffnet die App und befindet sich auf der Einstieg-Seite 2. Beim Versuch, den «Submit» Button zu betätigen und den Benutzernamen einzugeben, stellt die App eine Verbindung zur Realtime-Datenbank her 3. Da keine Internetverbindung vorhanden ist, sollte ein Alert angezeigt werden, der besagt, dass keine Internetverbindung besteht 4. Der Benutzer kommt nicht zur «Chat» Seite
Erwartetes Resultat	Ein Alert erscheint, der den Benutzer über die fehlende Internetverbindung informiert

Abschnitt	Inhalt
ID	ST-003
Anforderungen	<ul style="list-style-type: none"> - Der Benutzer muss in der Lage sein, Nachrichten zu senden - Die gesendeten Nachrichten müssen in Echtzeit angezeigt werden
Vorbedingungen	<ul style="list-style-type: none"> - Der Benutzer hat sich erfolgreich angemeldet und ist auf der Chat Seite - Mindestens ein anderes Mitglied ist ebenfalls in der Gruppe aktiv
Ablauf	<ol style="list-style-type: none"> 1. Der Benutzer gibt eine Nachricht in das Textfeld auf der Chat-Seite ein. 2. Der Benutzer klickt auf den «Senden» Button 3. Die gesendete Nachricht wird sofort auf der Chat-Seite angezeigt 4. Andere Gruppenmitglieder sollten die Nachricht ebenfalls in Echtzeit erhalten
Erwartetes Resultat	Die Nachricht wird sofort im Chat angezeigt und auch den anderen Benutzern in der Gruppe zugestellt

Abschnitt	Inhalt
ID	ST-004
Anforderungen	<ul style="list-style-type: none"> - Der Benutzer muss in der Lage sein, seinen Benutzernamen in den Einstellungen zu ändern - Die Änderung sollte erfolgreich in der Datenbank gespeichert und in der App angezeigt werden
Vorbedingungen	<ul style="list-style-type: none"> - Der Benutzer ist erfolgreich auf der Chat-Seite und hat bereits einen Benutzernamen eingegeben - Der Benutzer hat die Einstellungen-Seite über die Bottom Navigation aufgerufen
Ablauf	<ol style="list-style-type: none"> 1. Der Benutzer navigiert zur Seite «Settings» und gibt einen neuen Benutzernamen im entsprechenden Textfeld ein 2. Der Benutzer klickt auf den «Submit» Button 3. Der neue Benutzername wird in der Firebase Datenbank gespeichert und sofort im Benutzerinterface angezeigt 4. Der Benutzer kehrt zur «Chat» Seite zurück, um zu prüfen, ob der neue Benutzername korrekt übernommen wurde
Erwartetes Resultat	Der neue Benutzername wird sowohl in den Einstellungen als auch auf der Chat-Seite korrekt angezeigt

TESTPROTOKOLL

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

ID	Person / Datum	Erfolgreich	Bemerkungen
ST-001	D. Borne, 04.07.2025	Ja	Der Testfall war erfolgreich. Anmeldung funktioniert reibungslos.
ST-002	N. Ammeter 04.07.2025	Ja	Der Testfall war erfolgreich. Warnung bei fehlendem Internet wird reibungslos angezeigt.
ST-003	S. Catéchis 04.07.2025	Ja	Der Testfall war erfolgreich. Nachrichten werden reibungslos gesendet und angezeigt. Ausserdem kann man auch scrollen, was ein gutes feature ist.
ST-004	D. Borne, 04.07.2025	Ja	Der Testfall war erfolgreich. Den Benutzernamen kann man reibungslos ändern und man kann sogar das Konto löschen, was ein gutes feature ist.

FAZIT

Hier kommt eure Reflexion zum Projekt.

- Was lief gut/schlecht?
 - Am Mittwochmorgen mussten wir zu zweit mit dem Projekt anfangen, da Nicolas Ammeter (bammen(inhong)) krank war und nach Hause ging. Wir kamen trotzdem sehr gut voran und konnten alles und sogar mehr machen als geplant war. Das Mockup war schnell fertig und die Applikation konnte in geplanter Zeit fertig gemacht werden. Wir sind selber erstaunt, wie dass das ganze Projekt gut lief da fast keine Probleme auftauchten.
- Wie seid ihr mit dem Endergebnis zufrieden?
 - Wir sind alle sehr zufrieden mit der fertigen Applikation. Die Applikation sieht genau gleich und sogar besser aus als auf dem Mockup.
- Was habt ihr gelernt?
 - Vieles. Wir haben gelernt, wie dass man Navigationen erstellt und implementiert, wie dass eine gute Ordnerstruktur aussehen soll und wie man eine Datenbank von Firebase implementiert.
- War alles vorhanden oder was fehlte noch?
 - Alles war vorhanden, ausser die Firebase Datenbank. Diese hat Spyros aufgestellt und implementiert.