

Projektskizze

UniChat

The universal Messaging Client



Team: Aathavan Sivasothilingam, Burak Ünlü, Colin Vlahkinov, Flurin Meile, Jonas Briguët, Mattia Losa, Noe Javet, Philipp Marock, Philipp Tobler, Zulqarnain Khan

ZHAW School of Engineering

Lecturer: Andreas Meier

Class: IT22a_ZH

Submission Date: 17.03.2024

Module: Software-Project 4
(FS24)

Inhaltsverzeichnis

1. Ausgangslage & Idee	2
2. Stand der Technik / Konkurrenzanalyse	2
3. Kontextszenario	3
3.1. Registrierung.....	3
3.2. Chat-Übersicht	3
3.3. Kontakt auswählen	4
3.4. Kontaktdaten anzeigen	5
4. Weitere Anforderung.....	6
5. Ressourcen und Zeitplanung.....	6
6. Risiken	8
7. Wirtschaftlichkeit.....	8
8. Ausblick	8
9. Bibliografie	9

1. Ausgangslage & Idee

In verschiedenen Regionen der Welt benutzen Konsumenten unterschiedliche Kommunikations-Applikationen. In West-Europa dominiert WhatsApp, in Ost-Europa ist Telegram die bevorzugte Plattform und in Amerika kämpfen SMS und Facebook Messenger um den ersten Platz. Es gibt nicht nur Unterschiede zwischen Regionen, sondern auch Unterschiede im sozialen Umfeld. Personen, welche Sicherheitsbedenken haben, verwenden eher Threema oder Signal, während Andere sich keine Mühe machen wollen eine andere Messaging App zu verwenden. Das kommt vor allem davon, dass die bereits etablierten Messenger einen Grossteil der Bevölkerung als User haben.

Die Messenger unterscheiden sich jedoch nicht nur in ihren Benutzergruppen, aber auch in den Besonderheiten. Zum Beispiel unterscheiden sich WhatsApp und Telegram in der maximalen Dateigrösse, welche versendet werden kann. WhatsApp erlaubt nur 16MB, während Telegram grosszügiger ist und Dateien mit einer maximalen Grösse von 2GB erlaubt. So muss ein Benutzer, welcher grössere Dateien als Fotos oder kurze Videos versenden möchte, in jedem Fall eine weitere Plattform als WhatsApp benutzen.

Das Ziel von UniChat ist es verschiedene Messenger Apps zu vereinen, um es dem Nutzer zu ermöglichen mit seinen Kontakten auf verschiedenen Plattformen zu kommunizieren. Es ist unser Ziel Verbindungen zwischen Menschen zu vereinfachen und es dem End-Nutzer zu erlauben über künstlich geschaffenen Grenzen zwischen Plattformen zu springen.

Als weiteres Ziel möchten wir den Benutzern eine Möglichkeit geben, den Überblick zu behalten, welche Kontakte noch offene Nachrichten haben. Unsere Implementation wird uns auch erlauben in Zukunft eigene Features hinzuzufügen, welche nicht unbedingt von den Plattformen gewünscht sind. Zum Beispiel kann unsere Implementierung es dem Benutzer erlauben, Nachrichten zu lesen, ohne den Sender darüber zu benachrichtigen.

Unsere Idee ist vergleichbar mit einem Mail-Client, auf welchem verschiedene Mail-Anbieter in einer Plattform zusammen kommen. Uni Chat erlaubt es dem Benutzer verschiedene Accounts in einem einfachen UI zu benutzen und laufend zu sehen wer ihm auf welcher Plattform geschrieben hat.

2. Stand der Technik / Konkurrenzanalyse

Momentan gibt es einige Anbieter, welche ähnliche Features anbieten. Diese Plattformen haben jedoch Limitationen und oder sind nicht benutzerfreundlich. Als Hauptkonkurrenten zu unserer Umsetzung haben wir Beeper. Beeper bietet Verbindungen zu 15 verschiedenen Chat Networks als Desktop App an. Es hat jedoch mehrere Limitationen: Nicht alle können die App herunterladen. Es gibt eine Warteliste. Um die App zu benutzen monatliche Kosten hat (Beeper, 2024). Die gleichen Probleme bestehen für die App (Texts.com, 2024).

Wir denken, dass daher ein grosser Markt besteht für eine Kostengünstigere Implementation unserer Idee. Als grösstes Problem für unser Produkt eine grosse Benutzerakzeptanz zu finden, sehen wir die Unwissenheit der Benutzer, dass es eine Lösung gibt. Vor allem, da bereits bestehende Implementationen durch ihre Warteliste eine Marktlücke geschaffen haben, welche sie nicht selbst füllen.

3. Kontextszenario

Alice hat viele Freunde, mit denen sie auf verschiedenen Plattformen kommuniziert. Ihre Eltern und ihr engeres Umfeld bevorzugen Facebook Messenger und WhatsApp, ihre Studienkollegen hingegen benutzen Telegram und ihr Professor bevorzugt Signal. Da sie vier verschiedenen Plattformen benutzt, kann eine Nachricht oft lange unbeantwortet bleiben. Nachdem sie eine wichtige Nachricht von ihrem Professor auf Signal verpasst, entscheidet sich Alice, dass es eine bessere Lösung geben muss. Sie durchsucht das Internet nach einer Chat-Applikation, welche all ihre Chat-Plattformen zusammenfassen. Sie stösst auf UniChat und lädt die Applikation auf ihren Computer.

3.1. Registrierung

Nach dem Download startet sie das Programm und wird mit der Auswahl begrüsst, welche Plattformen sie benutzen möchte. Sie loggt sich mit ihrem WhatsApp-Account durch den angezeigten QR-Code ein. Nach dem der Login abgeschlossen ist, macht sie dasselbe für Telegram, Signal, Instagram und Facebook Messenger.

3.2. Chat-Übersicht

Nach dem sie sich mit all ihren Accounts eingeloggt hat sieht sie eine Benutzeroberfläche, welche sie von anderen Chat-Applikationen kennt (siehe Abbildung 1).

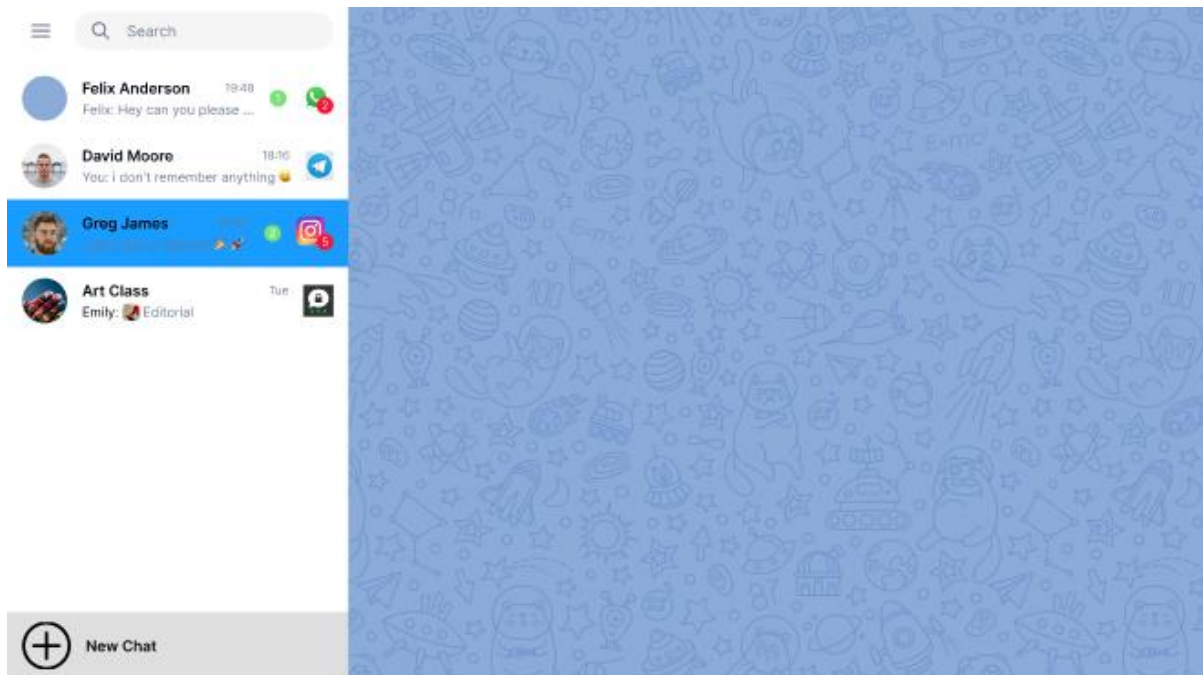


Abbildung 1 UniChat kein Chat ausgewählt

3.3. Kontakt auswählen

Alice kann auf der linken Seite ihre Kontakte sehen, die dazu gehören letzten Nachrichten, ob diese bereits gelesen sind und auf welcher Plattform sie sich befindet. Sie möchte nun die Nachrichten von Greg lesen. Dazu klickt sie auf den Kontakt von Greg. Als nächstes sieht sie das Chat-Fenster von Greg auf Instagram. (siehe Abbildung 2)

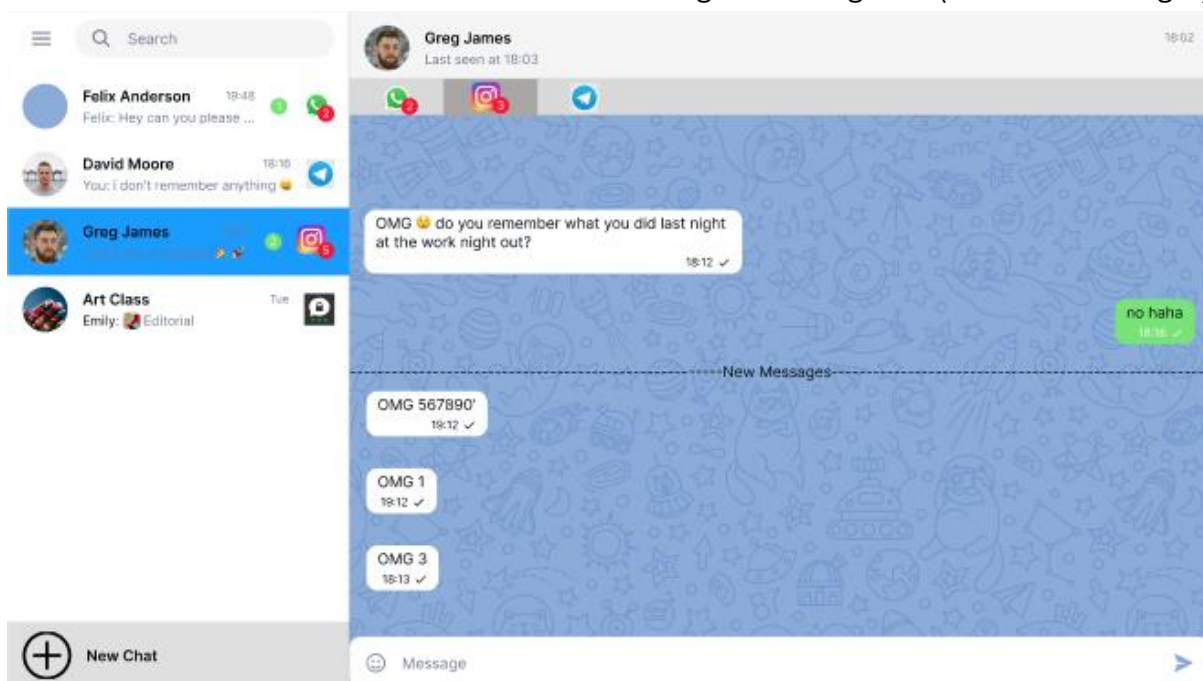


Abbildung 2 Chat mit Kontakt ist ausgewählt

Sie erkennt, dass sie mit Greg drei verschiedene gemeinsame Plattformen hat. Nämlich WhatsApp, Instagram und Telegram. Da sie momentan Instagram ausgewählt hat sieht

sie das Instagram selektiert ist. Sie kann nun auf die Nachrichten von Greg antworten oder zum Beispiel WhatsApp auswählen und die Nachrichten auf dieser Plattform lesen.

Sie realisiert, dass sie mit Greg drei gemeinsame Kommunikationsplattformen teilt: WhatsApp, Instagram und Telegram. Sie hat gerade Instagram geöffnet und der jeweilige Tab ist selektiert. Sie hat nun die Möglichkeit, auf Gregs Nachrichten zu antworten oder beispielsweise zu WhatsApp zu wechseln, um seine Nachrichten auf dieser Plattform zu lesen.

3.4. Kontaktdaten anzeigen

Alice kann durch einen Klick auf das Profilfoto von Greg auf seine Kontaktdaten gelangen. (siehe Abbildung 3)

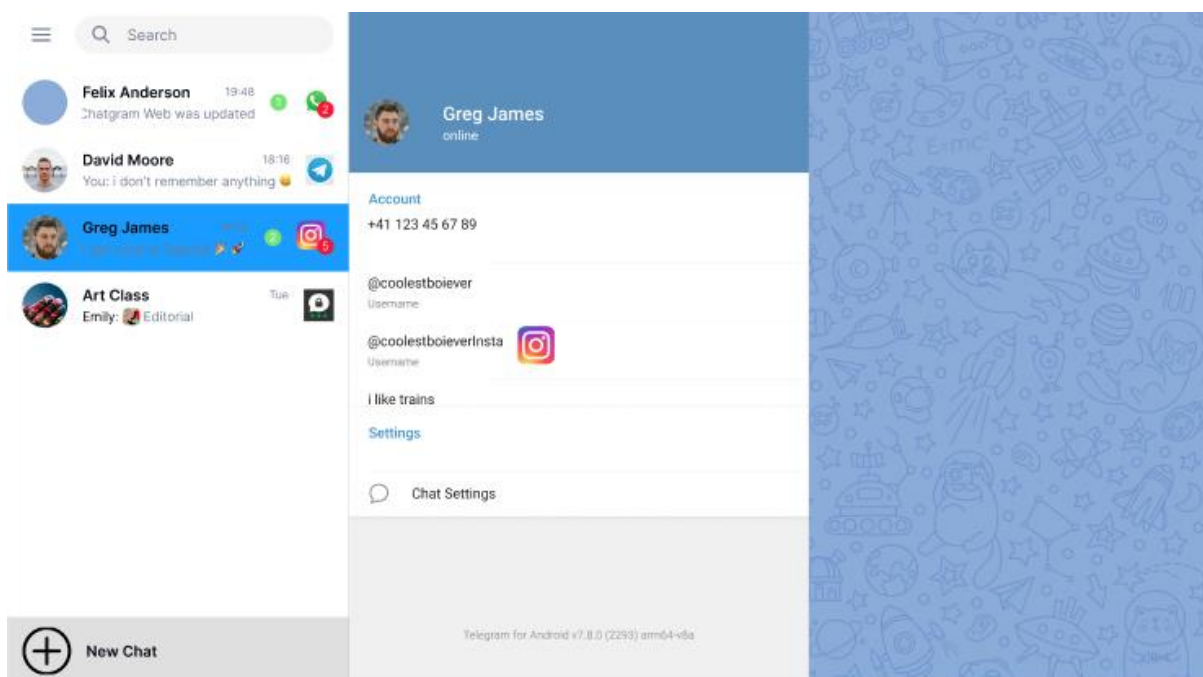


Abbildung 3 Kontakt ist geöffnet

Hier kann sie nun sehen, welche Kontakt-Daten UniChat von Greg zusammengeführt hat. Nämlich die Telefonnummer, der Telegram-Username und der Instagram-Username. Sie kann dann unter den Einstellungen noch weitere Plattformen zu diesem Kontakt hinzufügen. Das erlaubt Alice ihre Kontakte auf den verschiedenen Plattformen in einen Kontakt zusammenzuführen.

4. Weitere Anforderung

Eine Anforderung nach Beenden des Prototyps wird die Unterstützung weiterer Messaging-Plattformen sein. Im UniChat Prototyp werden die dominierenden Plattformen integriert, beim weiteren entwickeln können dann kleinere spezialisierte Plattformen integriert werden, um eine noch breitere Nutzerbasis anzusprechen.

Die Multimedia Erfahrung zu optimieren, wird eine weitere Anforderung sein. Das heisst, dass nicht nur Fotos und Videos versendet werden können, sondern auch Sprachnachrichten aufzunehmen, Standorte zu verschicken oder GIFs zu integrieren.

Da UniChat auf verschiedene Messenger-Plattformen Zugriff hat und somit auch auf die persönlichen Daten der Benutzer zugreifen kann, wird es nach Beenden des Prototyps erforderlich sein, starke Verschlüsselungsstandards einzuführen, um die Sicherheit und den Datenschutz zu gewährleisten.

5. Ressourcen und Zeitplanung

Das Team besteht aus zehn Teammitgliedern. Alle Teammitglieder bringen ein gewisses Mass an Vorwissen im Programmieren. 3-4 Teammitglieder sind schon etwas mehr erfahren, da diese auch schon vor dem Studium programmiert haben, während die restlichen Mitglieder mit dem Studium an ihre Programmiererfahrungen gekommen sind. Das User Interface wird mit Qt für Python implementiert, damit haben bereits zwei Teammitglieder gearbeitet und können so das restliche Team bei der Erarbeitung des User Interface unterstützen. Es wird im UniChat Projekt auch mit APIs gearbeitet und auch hier gibt es bereits einige Teammitglieder, die das Know-How haben, wie man APIs richtig einbindet und benutzt. Plattformen, welche keine API uns anbieten, werden wir mit Hilfe von Selenium-Driver die Daten aus der Web-Applikation holen und unsere eigene Schnittstelle implementieren.

KW	Sprint	Start	Ende	Meilensteine
9	1	26.2.	3.3.	<ul style="list-style-type: none"> • Programmiersprache und Framework definieren • Wireframe von GUI
10	1	4.3.	10.3.	<ul style="list-style-type: none"> • Einarbeitung in verschiedenen APIs • GitHub Projekt erstellen
11	1	11.3.	17.3.	<ul style="list-style-type: none"> • Projektskizze erstellen (Abgabe 17.3.) • Implementation WhatsApp • Implementation Instagram • Starten mit GUI
12-14	2	19.3.	1.4.	<ul style="list-style-type: none"> • Weiterführung / Abschluss GUI • Implementation Telegram
14-16	3	2.4.	15.4.	<ul style="list-style-type: none"> • Implementation Facebook Messenger • Implementation Discord
16-18	4	16.4.	29.4.	<ul style="list-style-type: none"> • Implementation zusätzlicher Features, welche spezifisch sind zu verschiedenen Messenger
18-20	5	30.4.	13.5.	<ul style="list-style-type: none"> • Testing • Fehlerbehebungen
20-	6	14.5.	21.5.	<ul style="list-style-type: none"> • Individuelles Endprodukt NoTech (Abgabe 24.5.) • Testing • Fehlerbehebungen

Tabelle 4 Überblick Projekt Planung

Die Aufgabenverteilung erfolgt basierend auf den Fähigkeiten und auch den Präferenzen der verschiedenen Teammitgliedern und werden während der Sprint-Planung verteilt, welche jeden zweiten Dienstag durchgeführt werden. Durch regelmässige Sprint-Reviews

zwei bis drei Mal in der Woche wird der Fortschritt der jeweiligen Programmiergruppen oder Teammitglieder überprüft und etwaige Anpassungen diskutiert und vorgenommen.

6. Risiken

Eines der grössten Risiken besteht in der Abhängigkeit von Drittanbietern. Aufgrund der Verwendung von Drittanbieter-APIs im UniChat und damit das Risiko von Einschränkungen oder Ausfällen eines Drittanbieters. Änderungen in den APIs oder Geschäftsbedingungen können die Funktionalität der Anwendung beeinträchtigen.

Ein weiteres Risiko besteht in der Erfahrung des Teams, da beim User Interface mit Qt für Python gearbeitet wird und nicht alle Teammitglieder ausreichend Kenntnis verfügen, um ein solches Interface zu implementieren. Dadurch könnten unerwartet Verzögerungen oder nicht eingeplante Zeit für Recherche entstehen.

7. Wirtschaftlichkeit

UniChat ist eine Softwareapplikation, welche im Eigenauftrag entwickelt wird. Die Monetarisierung ist als einmaliges Erwerben der Software eines jeden Nutzers. Der genaue Kostenpunkt wird noch besprochen.

Es wird bewusst auf ein Subskription-Modell verzichtet. Dies aus zweierlei Gründen, zum einen setzt die bestehende Konkurrenz auf dieses Finanzierungsmodell (siehe Abschnitt Konkurrenzanalyse) und zum anderen deswegen, weil die laufenden Kosten sehr überschaubar sind. Im Gegensatz zu herkömmlichen Kommunikations-applikationen braucht UniChat keine online Server und Datenbanken, um Chat Historien zu sichern oder Nachrichten zu verschicken/empfangen. Denn UniChat bietet einzig ein Vereintes Interface über verschiedenste Messenger-Applikationen. Das Versenden und Zwischenspeichern der Nachrichten geschieht nach wie vor über Konkrete Messenger Apps, wie zum Beispiel WhatsApp oder Telegram.

Der Unterhalt der Software selbst wird somit der einzige laufende Kostenpunkt nach abgeschlossener Entwicklung sein. Im Ganzen ergibt sich nach [Ressourcen und Zeitplanung](#) bei einem Zeitaufwand von 1200 Arbeitsstunden und theoretischem Durchschnittslohn von CHF 30/h prognostizierte Projektkosten von CHF 36'000 bis zu einem Lauffähigen Prototyp.

8. Ausblick

Gegen Ende von PM4 wollen wir eine Lauffähige Applikation ab GitHub mit einem 'README' mit präzisen Installationsinstruktionen bereitstellen. Auf eine Bedienungsanleitung der bereits installierten Applikation wird voraussichtlich verzichtet, denn UniChat soll intuitiv wie jede andere Messenger Applikation spielerisch bedienbar und erlernbar sein und keiner Anleitung bedürfen.

Mögliche Erweiterungen bestehen wie bereits erwähnt hauptsächlich aus der Integration weiterer Messenger Apps und dem Hinzufügen zusätzlicher Features inkl. des Verschicken von Sprachnachrichten und des Standortes.

Um marktfähig zu sein, müssten einige dieser Features schon implementiert sein, wie auch einiges an zusätzlicher Arbeit im Rahmen des Deployments und bestenfalls einer Portierung zu Android/iOS als mobile App.

Eine direkte Kommerzialisierung ist nach PM4 unter anderem deswegen nicht geplant, die Applikation soll hauptsächlich für den Eigengebrauch verwendbar sein.

9. Bibliografie

9.1 Abbildungsverzeichnis

Abbildung 1 UniChat kein Chat ausgewählt.....	4
Abbildung 2 Chat mit Kontakt ist ausgewählt.....	4
Abbildung 3 Kontakt ist geöffnet.....	5

9.2 Quellenverzeichnis

Beeper. (2024, 3 16). *Beeper FAQ*. From Beeper FAQ: <https://www.beeper.com/faq>

Texts.com. (2024, 3 16). *Texts.com FAQ*. From Tests.com: <https://texts.com/faq>