SENTISCAPE

Functional Specification

Jason Boylan - 18342986

Kelan Smyth - 18342973

Table of contents

1. **Introduction**
   1.1 Overview
   1.2 Business Context
   1.3 Glossary

2. **General Description**
   2.1 System Functions
   2.2 User Characteristics and Objectives
   2.3 Operational Scenarios
   2.4 Constraints

3. **Functional Requirements**
   3.1 Create Account
   3.2 Login System
   3.3 Search System
   3.4 Network
   3.5 Message Screen
   3.6 Conversations Screen
   3.7 Sending a Message
   3.8 Receiving a Message
   3.9 Attaching an Emoticon to a Message
   3.10 Displaying a Conversation Sidebar
   3.11 Saving Past Conversations
   3.12 Generating Conversation Summaries
   3.13 Viewing Past Conversations
   3.14 Creating Group Chats
   3.15 Group Chats: Adding and Removing
   3.16 Sentiment Analysis to Remove Trolls
   3.17 Vivid and Muted Colours
   3.18 Light and Dark Mode
   3.19 Logout System

3.20 Delete Account

4. **System Architecture**
   4.1 System Architecture Diagram
   4.2 System Architecture Diagram Description

5. **High-Level Design**
   5.1 High-Level Design Context Diagram
   5.1 High-Level Design Diagram Description

6. **Preliminary Schedule**
   6.1 Schedule Overview
   6.2 GANTT Chart

7. **Appendices**
   7.1 References

Title

Sentiscape: A Next-Generation Messaging Application to Support Affectively-Enriched Communication

1. Introduction

1.1 Overview

The purpose of the system is to create a mobile messaging application for users looking for an application that can convey the affective side of messages as well as the informative side. It will also make it easy for users to categorise and find past conversations, using natural language processing and sentiment analysis to create emotive summaries of conversations. It is entitled Sentiscape, thanks to its association with sentimentalism.

Sentiscape will allow users to perform the basic functions of any messaging app, such as sending messages, attaching multimedia and creating group chats.

Also with *Sentiscape*, the app will re-think how text-messaging interaction can be done and subsequently design novel user-interfaces by bringing in up-to-date computational technologies at the back-end of the app.

This will be displayed with primarily two major changes. Firstly, the application will allow users to categorise their conversations with friends which will make it easier to look for past conversations. Conversations will be broken down by day and a summary of the conversation will be created to allow users to easily remember what conversation took place on a particular date.

Secondly, it will track the mood of the conversation which will allow for different emoticons, as well as softer and brighter backgrounds, depending on the mood. This will be done using sentiment analysis to analyse the mood of the conversation. If the most prominent emotion being displayed is sad, there will be more muted colours and suggested emoticons will relate to the conversation.

The background of this application came from a few different places. Firstly, as part of our project last year, we worked on a system that incorporated some machine learning algorithms and we wanted to work in the same field again. This was the reasoning behind our push for sentiment analysis to be a crucial part of the application.

Secondly, we noticed various shortcomings of existing messaging applications, such as the difficulty in accessing past conversations and the challenge that existed when trying to get across the affective side of a message.

Thirdly, we were inspired by Kansei engineering, a Japanese term that translates to “emotional” or “affective” engineering. This involves translating the consumer’s feelings into parameters that can be used as part of a product / service. We tried to incorporate some of these ideas into our application.

1.2 Business Context

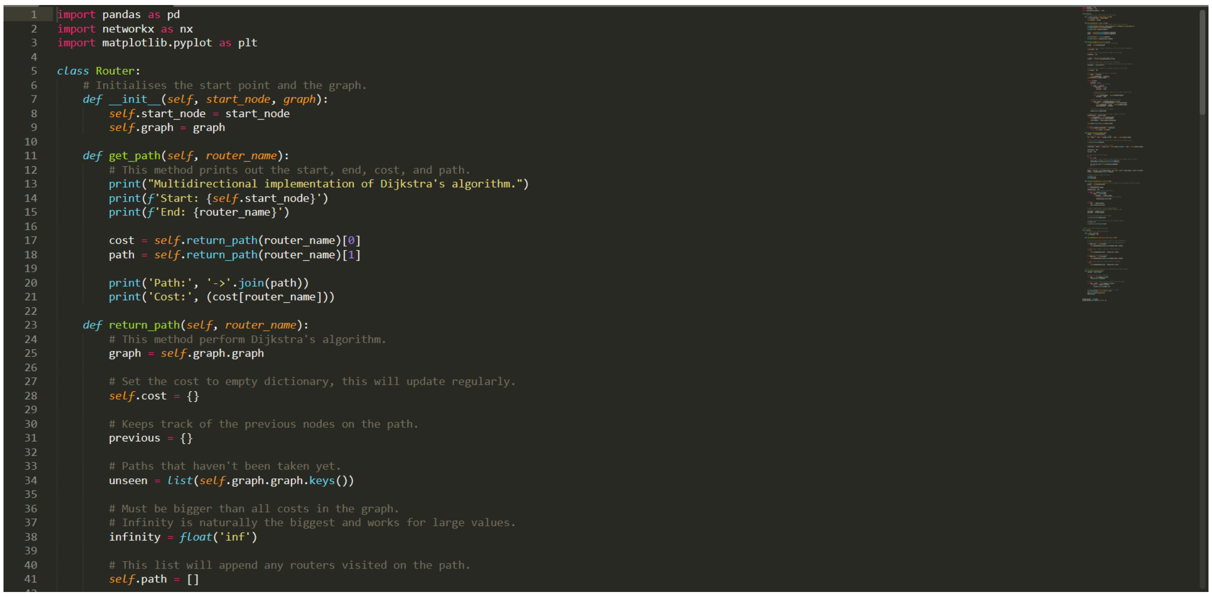
The mobile application will be used by a wide demographic of people in order to message each other (anyone with an Android smartphone). The application will not be sponsored by any business but will be freely available for anyone to use.

1.3 Glossary

GUI
Graphical User Interface: A form of user interface that allows users to interact with devices through graphical icons and audio indicators such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

Trolls
Trolls are people who leave intentionally provocative or offensive messages on the internet in order to get attention, cause trouble or upset someone.

Sublime Text Editor
Sublime is a code editor software program for programming and markup languages. It features an overview of the code written on the right-hand side, which is seen below. ⇓⇓⇓



```python
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

class Router:
    # Initialises the start point and the graph.
    def __init__(self, start_node, graph):
        self.start_node = start_node
        self.graph = graph

    def get_path(self, router_name):
        # This method prints out the start, end, cost, and path.
        print("Multidirectional implementation of Dijkstra's algorithm.")
        print(f'Start: {self.start_node}')
        print(f'End: {router_name}')

        cost = self.return_path(router_name)[0]
        path = self.return_path(router_name)[1]

        print('Path:', '->'.join(path))
        print('Cost:', (cost[router_name]))

    def return_path(self, router_name):
        # This method perform Dijkstra's algorithm.
        graph = self.graph.graph

        # Set the cost to empty dictionary, this will update regularly.
        self.cost = {}

        # Keeps track of the previous nodes on the path.
        previous = {}

        # Paths that haven't been taken yet.
        unseen = list(self.graph.graph.keys())

        # Must be bigger than all costs in the graph.
        # Infinity is naturally the biggest and works for large values.
        infinity = float('inf')

        # This list will append any routers visited on the path.
        self.path = []
```

Natural language processing
An area of machine learning that studies the application of computational techniques to the analysis and synthesis of natural language.

Sentiment analysis
Sentiment analysis is the process of detecting positive or negative sentiment in text.

Tokenisation
Tokenisation breaks raw text into words, sentences called tokens.

Conversational summarisation
Natural Language Processing technique that summarizes the user conversation.

2. General Description

2.1 System Functions

User Functionality System

The user will download the application on their android device to access Sentiscape. Users will then log in to their unique account to access their conversations. This system allows any of the existing users to interact with one another if they search for each other by username. Conversations are moderated using sentiment analysis while also allowing users the ability to block certain accounts.

GUI Functionality

Sentiscape will be a mobile application that users will be able to use on any android device. The user will interact with a log-in system either following prompts to create a new Sentiscape account or logging in to an existing account. The user will be able to search for other Sentiscape accounts to communicate and begin a conversation. While communicating with another account users utilise the Sentiscape keyboard and text prediction functionality in the chatroom. Users can also browse previous conversations using the sidebar and once selected the user can also view summarizations of past conversations. Users will utilise the log out section to exit the application.

Conversation Summarization

One of Sentiscape's main functionalities will be using natural language processing to generate summaries of conversations between users to allow the ability to quickly recap on what has been said making it easier for users to jump back in right where they left off. Tokenization will be implemented on the user text to allow the system to process conversations and convert them into concise summaries.

Sentiment Analysis in Conversation

One of Sentiscape's main functionalities will be using sentiment analysis to create a more emotion driven conversation between users by reacting in turn with the detected emotion of the conversation. This will include background colour changes to convey the mood of the conversation as well as an emoticon feature that will change a users profile image depending on their detected mood. After the mood of each message has been detected it is then cataloged by the detected sentiment in the database to allow users to search by mood through their previous conversations, this will allow our users to quickly find funny jokes they sent or positive messages of encouragement they received.

Login / Account System

The application will support an account system where users can log in and out of the system. Upon opening the application for the first time, users will be able to sign up and create a new account, or log in to an existing account. The user will have a unique username and an encrypted password. They can also delete their account.

Starting a Conversation

Users will be able to search for other users to engage in conversation with. The user can search for others by username and request to converse. Upon accepting an invitation to converse, users will be able to message back and forth.

Creating Group Chats

The application will support group chats where multiple users can converse with each other. The group chat will have an admin (the creator of the group chat will be admin) who can add and remove users.

Searching Conversations

Another main functionality of Sentiscape will be the unique way users will be able to search through previous conversations. As mentioned above, messages will be cataloged by their detected sentiment which will allow users to quickly find messages of a happy, sad, or funny theme and so forth. This will be implemented through a sidebar graphic of the entire scaled down conversation, similar to that of a text editor, as this feature has never been implemented in a messaging application. Messages will be colour coded for each detected mood allowing users to quickly jump between messages and search previous conversations.

2.2 User Characteristics and Objectives

The community of users we expect to use Sentiscape will vary greatly in age and technical capability. Sentiscape will have a simple and intuitive user interface that will cater for the needs of every user. Ease of use is imperative for a positive user experience so Sentiscape will ensure easy navigation for users with each of the application's functionalities laid out strategically for ease of use. We can also assume that the majority of our users will have previous knowledge of messaging applications if they own an android device, this experience will help users as they familiarise themselves with the Sentiscape application.

2.3 Operational Scenarios

Use Case 1 - User creates account

The user will open the application for the first time and be presented with a log-in system where they will be asked to either log in or sign up. The user selects the “Sign up” button. They will then enter an email address, create a username and a password.

Use Case 2 - User logs in successfully

The user will open the application and be presented with a log-in system. They will be asked to enter their log-in credentials. There will be two fields, one where the user enters their username and one where they enter their password. Once this is done and they enter the correct credentials, the user taps on the “Login” button and successfully logs into the system.

Use Case 3 - User logs in unsuccessfully

When asked to enter their log-in credentials, the user enters their password incorrectly. When they tap on the “Login” button, a message appears that says “The username or password was entered incorrectly” and the user won't be able to log in.

Use Case 4 - User searches for a another user to message

The user taps on the search bar to find someone to message. They enter the person’s username and the results will display on screen. The user will click on the user they want to message and this will bring them to a screen where they can message that user.

Use Case 5 - User sends a message

The user is on the message screen with another user where they can send messages back and forth. At the bottom of the screen the user clicks on the message tab. This will bring up the keyboard and the user types in the message they want to send. The user clicks the “Send” button. This sends the message to the other user and it will display the message on the message screen.

Use Case 6 - User sends a positive message

The user sends a message that contains a number of words that are flagged as positive. When the message is sent, a smiling emoticon will appear alongside the message.

Use Case 7 - User sends a negative message

The user sends a message that contains a number of words that are flagged as negative. When the message is sent, an upset emoticon will appear alongside the message.

Use Case 8 - User sends a neutral message

The user sends a message that can’t be distinctly decided if it’s positive or negative. When the message is sent, a neutral emoticon will appear alongside the message.

Use Case 9 - User accesses previous conversations

The user clicks on the sidebar that displays the conversation being had. This brings up the previous conversations screen where the user can view a visual summary of the conversations had on different dates.

Use Case 10 - User selects conversation from a particular date

The user views the various conversation summaries, which includes a visual summary of the conversation as well as a text summary of the conversation. They select a summary from a particular date. This will bring them to the conversation from that date on the message screen.

Use Case 11 - User logs out

The user selects the logout button on the corner of the screen. They will be prompted with a pop-up that asks “Are you sure you want to log out?” There are two buttons to select from. “Yes” and “No”. The user selects “Yes”. They are logged out and are presented with the log-in screen.

2.4 Constraints

Time Constraints

The project deadline is 15th April 2022 and as such a functioning messaging app, a working network system and a successful implementation of the aforementioned NLP techniques must be incorporated by that date.

Language Constraints

The primary languages that will be used during the development of the application will be Java and Python. Both members of the team are proficient in Python, as well as some of its machine learning libraries. Neither has worked in this particular area of machine learning before, so gaining familiarity with new libraries may pose challenges.

Also, both members are proficient in Java thanks to coursework and INTRA internships. However, neither has worked on an Android app before so this may pose a variety of new challenges.

Hardware Constraints

The system must be tested using an Android smartphone. Android emulators can be used while testing using Android Studio but ultimately the application must be able to run on an Android device.

Dataset Constraints

Creating a dataset based only on previous conversations would pose too challenging for any machine learning algorithms to work on, so there will have to be an existing dataset used to base this on.

User Requirements

Users will require an Android smartphone as there are no plans for a web version of the application nor are there plans to make the application run on iOS/Windows devices or any other operating systems.

3. Functional Requirements

3.1 Create Account

Description
The system must allow users to create an account that can store a username and password in order to message other users and access features.

Criticality
This feature is critical to the overall application as without an account the user cannot use the application.

Technical issues
Passwords must be encrypted and accounts must be secure from potential breaches.

Dependencies with other requirements
None.

3.2 Login System

Description
The application must have a login system in order to allow users to log in using unique credentials. This will allow them to access the application's features.

Criticality
This system is critical to the application because users must be able to log in and have a unique account.

Technical issues
Email addresses, usernames and passwords must be stored safely and securely in order to prevent potential breaches.

Dependencies with other requirements
The user must have created an account in order to be able to log in.

3.3 Search system

Description

The application must provide a system where users can search for other users to message.

Criticality
The criticality of this system is vital as users must be able to find others that they want to message.

Technical issues
It is important that the search system is swift and accurate and that it provides the correct result.

Dependencies with other requirements
The search system relies on the create account system working correctly. It must ensure that each username created is unique so that the search system returns the correct user being searched for.

3.4 Network

Description
The application must have a working network so that users can send messages back and forth.

Criticality
This requirement is necessary because the entire application hinges on the network working correctly. If users can't send and receive messages, the system has failed on a basic level.

Technical issues
The network must be able to work both with one-on-one conversations and with group chats.

Dependencies with other requirements
None.

3.5 Message Screen

Description
Users must be able to send messages to each other on the message screen and the application must have a clean and simple GUI in order to do so. The message screen should display the user that the message is being sent to, the previous conversations had, a keyboard and a "Send" button.

Criticality
This element is not essential in order for the application to work properly. As long as the network is working, users should be able to send and receive messages.

However, having a visually pleasing interface that allows users to type and send messages makes the application easier for a wide demographic to use.

Technical issues
The GUI must be easy to use and aesthetically pleasing so good front-end design will be important.

Dependencies with other requirements
The network must be working to guarantee that the messages are being delivered and received. It must also be able to interact with the conversations screen.

3.6 Conversations Screen

Description
The conversations screen displays a list of conversations that the user is currently having. These will be ordered by recency, where the person / group the user most recently messaged will be displayed at the top. Selecting a conversation will bring the user to the message screen for that particular conversation. This screen will only display conversations that have been initiated already as well as the predominant emotion(s) that were detected within each conversation.

Criticality
This is a critical element of the application because the user must have the ability to select a user / group to converse with.

Technical issues
The screen must be able to sort conversations in order of recency.

Dependencies with other requirements
Must be able to interact with the message screen.

3.7 Sending a Message

Description
The system should provide a keyboard for users at the bottom of the message screen. This will be accompanied by a “Send” button that sends the message once the user taps on it. The user will also be prompted to attach an emoticon that the system has detected (elaborated further in 3.9).

Criticality
The button and keyboard are a necessary element of the GUI in order for the user to type and send their message.

Technical issues
Must be integrated with the Android keyboard.

Dependencies with other requirements
Must be linked with the network so that the message is sent upon tapping on the “Send” button.

3.8 Receiving a Message

Description
The system must notify the user that a message has been received. This will be done by highlighting the conversation with the user on the conversation screen, as well as displaying a notification outside of the application’s user interface.

Criticality
This feature isn’t critical. The application will still function without notifying the user that they have received a message. However, it would be difficult for users to know that they have received messages without having this feature.

Technical issues
Must be integrated with the Android user interface using `NotificationCompat`.

Dependencies with other requirements
Relies on the network working correctly and syncing the messages that come in with notifications.

3.9 Attaching an Emoticon to a Message

Description
The system must implement conversational sentiment analysis in order to attach an emoticon beside the message sent. These emoticons will be in place of a traditional user profile picture beside their message. Instead, an emoticon displaying the emotion of the message will be displayed so that the receiving user can instantly infer the tone of the message. The system will prompt the user with a question asking if they want the emoticon attached / if it’s the appropriate one.

Criticality
It is essential in order to succeed in making *Sentiscape* achieve its goals of being an informative and affectionate messaging service, the central aspect of the application.

Technical issues
The functionality of this requires using various conversational sentiment analysis methods, such as the split method and the neighbourhood method in order to correctly break up sentences that may contain positive and negative sentiments.

Dependencies with other requirements
The functionality of this feature requires that messages can be sent and received.

3.10 Displaying a Conversation Sidebar

Description
This functionality will display a sidebar along the side of the message screen that will provide an overview of the conversation had on a single date. Similar to Sublime Text Editor, this sidebar will display a miniature version of various shapes which represent emoticons, single-line and multi-line messages.

Criticality
This is a necessary step in creating a novel messaging application that will differentiate itself from its competitors.

Technical issues
This will require use of Android Studio's *Navigation* component by using the `NavigationUI` class.

Dependencies with other requirements
The message screen must be working correctly in order to incorporate this requirement.

3.11 Saving Past Conversations

Description
The system should auto-save the conversation sidebar summary from a 24-hour period.

Criticality
This is critical to the conversation summary system, as there needs to be a saved list of sidebar summaries in order to view past conversations.

Technical issues
The system should automatically do this at 23:59 at the end of every day.

Dependencies with other requirements
Relies on the conversation sidebar function working correctly.

3.12 Generating Conversation Summaries

Description
The system should generate a conversation summary for each conversation, along with tagging the result of sentiment analysis performed on each segment of the conversation. This will involve text summarisation using natural language processing and a dataset for training.

Criticality
This is critical to the functionality of the conversation summaries.

Technical issues
Incorporating the conversation into the natural language processing algorithm will be difficult and will involve closely interlinking it with an existing dataset of conversation dialogue.

Dependencies with other requirements
The auto-save conversation feature must work as described.

3.13 Viewing Past Conversations

Description
The application should allow users to access a menu that will display a list of previous conversations' summaries. This will consist of the sidebar summary and an auto-generated text summary of the conversation.

Criticality
This feature is important as it is a large part of what will make *Sentiscape* stand apart from its competitors.

Technical issues
Pairing the sidebar summary with the conversation summary will be important to accomplish correctly.

Dependencies with other requirements
The system must be able to save past conversations as well as generate text summaries.

3.14 Creating Group Chats

Description
The user must be able to create a group chat where they can converse with multiple other people.

Criticality
This system is not essential but would provide a positive impact for the overall user experience.

Technical issues
Creating a network that will allow multiple users to participate.

Dependencies with other requirements
The network must work for a conversation greater than two people.

3.15 Group Chats: Adding and Removing

Description
The system must provide users with the ability to add and remove users to and from group chats that they've created.

Criticality
This requirement is not essential overall for the application but in terms of group chats it is important, otherwise the group chats serve no purpose.

Technical issues
The user that creates the group chat must be the only one that can add and remove other users.

Dependencies with other requirements
The user must be able to create a group chat in order to observe this functionality.

3.16 Sentiment Analysis to Remove Trolls

Description
If a member of a group chat uses consistently negative language, they will be highlighted and the group creator will be notified and provided with the opportunity to remove them.

Criticality
Not critical to the messaging functionality of the app but important to the machine learning element.

Technical issues
The sentiment analysis algorithm will have to be able to detect overtly negative language from a particular user, and it must be of a level greater than a certain threshold.

Dependencies with other requirements
Group chats must exist and the group creator must be able to remove other members of the group.

3.17 Vivid and Muted Colours

Description
Depending on the user's mood observed by the system, the colours of the app may appear more vivid or muted to create a more emotionally-rich conversation rather than an information exchange.

Criticality
This is not critical to the basic functionality of the messaging application, but it will involve sentiment analysis which is at the heart of the application.

Technical issues
The system will have to have a nuanced understanding of the user's prevailing emotion.

Dependencies with other requirements
None.

3.18 Light and Dark Mode

Description
The application will provide a functionality where the user can choose between a light and dark mode.

Criticality
This function is not critical to the application.

Technical issues
Must remember the user's most recent choice of either light or dark mode using local storage.

Dependencies with other requirements
None.

3.19 Logout System

Description
The application must have a logout system in order to allow users to log out from their account.

Criticality
This system is critical as a login system must be accompanied by the ability to log out.

Technical issues
Must make sure that the user is logged out even with application refresh.

Dependencies with other requirements
The login system must be able to function in order to then log out.

3.20 Delete Account

Description
The system must allow users to delete their account if they so wish. This should also delete all relevant information attached to their account, including their email address, username and password.

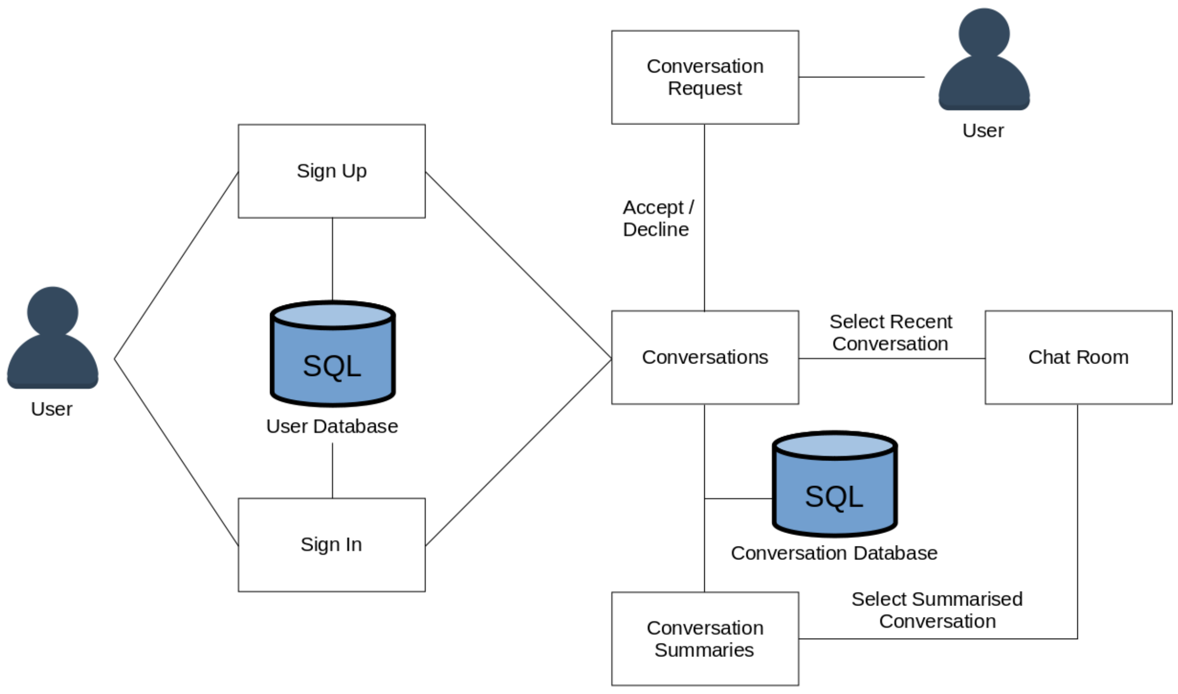
Criticality
This functionality is essential.

Technical issues
Must guarantee that the user's information is deleted.

Dependencies with other requirements
The user must be able to create an account initially.

4. System Architecture

4.1 System Architecture Diagram

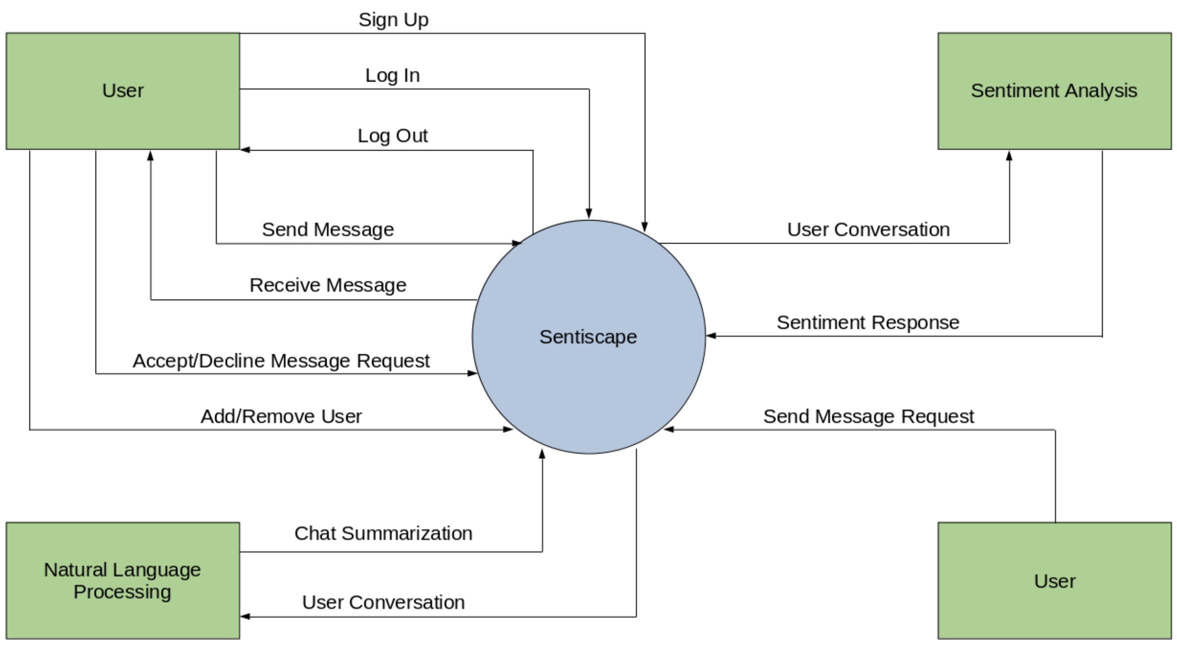


4.2 System Architecture Diagram Description

The diagram shown above shows the architecture intended for Sentiscape. Utilising an android device users will be able to log in or create an account for Sentiscape with usernames and passwords stored in a database. Each username and password stored in the database will be encrypted for security purposes. Once signed in, users will be brought to the conversations page where recent chats will be displayed and available for selection, as well as the option to view the summarised conversations page. Each chat session will be stored in a database to be used for natural language processing and sentiment analysis.

5. High-Level Design

5.1 Context Diagram



5.2 Context Diagram Description

Log in/Sign Up: Once the user opens the application they will be met with the login screen. If the user has an account they can follow the sign in procedure or if they do not have an existing account they can follow the prompts to create a new account.

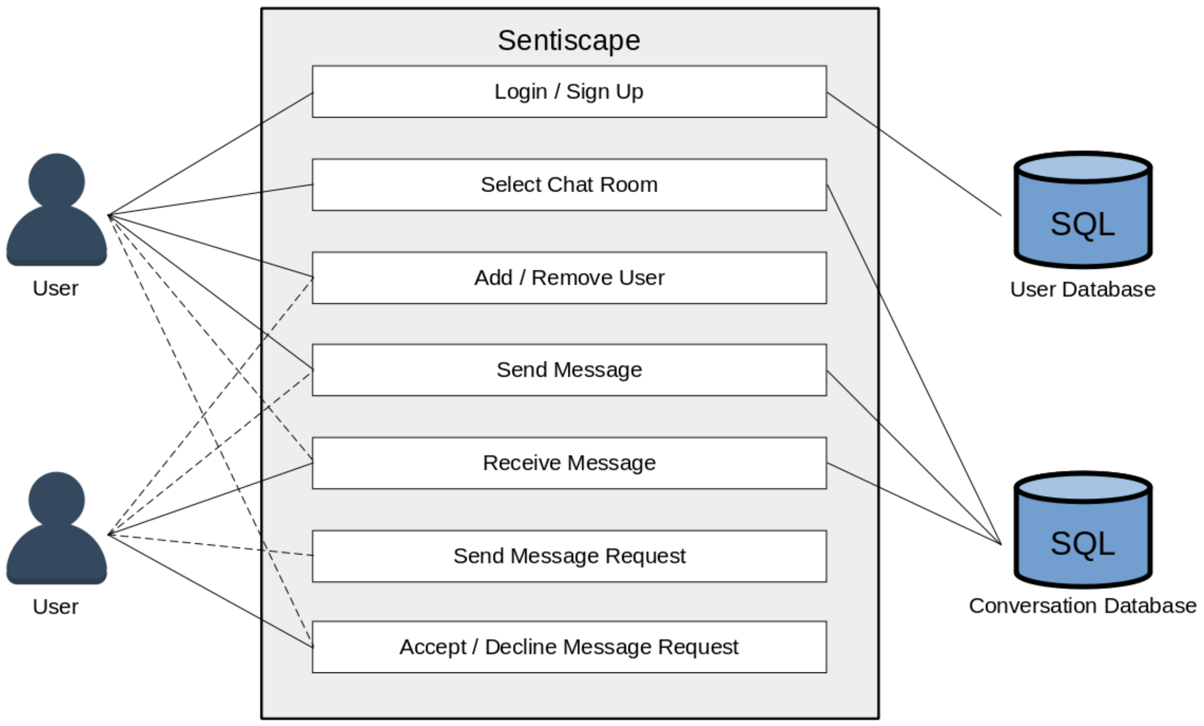
Connecting Users: Once the user has logged in they will be able to choose a recent chat or search a new user to chat with. Message requests are sent to start a chat room between two or more users which can be accepted or declined.

Chat rooms: Once users have joined a chat room with one or more people they will be able to send and receive messages. Chat rooms also have an Add and Remove functionality that will allow users to add or remove members of the chat room.

Sentiment Analysis: User conversations stored in the database will be processed by Sentiscapes sentiment analysis system which will determine what kind of sentiment response the system returns through Sentiscapes user interface.

Summarization: User conversations stored in the database will be processed by Sentiscapes natural language processing system to generate summaries of conversations to allow users to quickly browse and refamiliarise themselves with what was previously discussed in each of their chat rooms.

5.3 Use Case Diagram

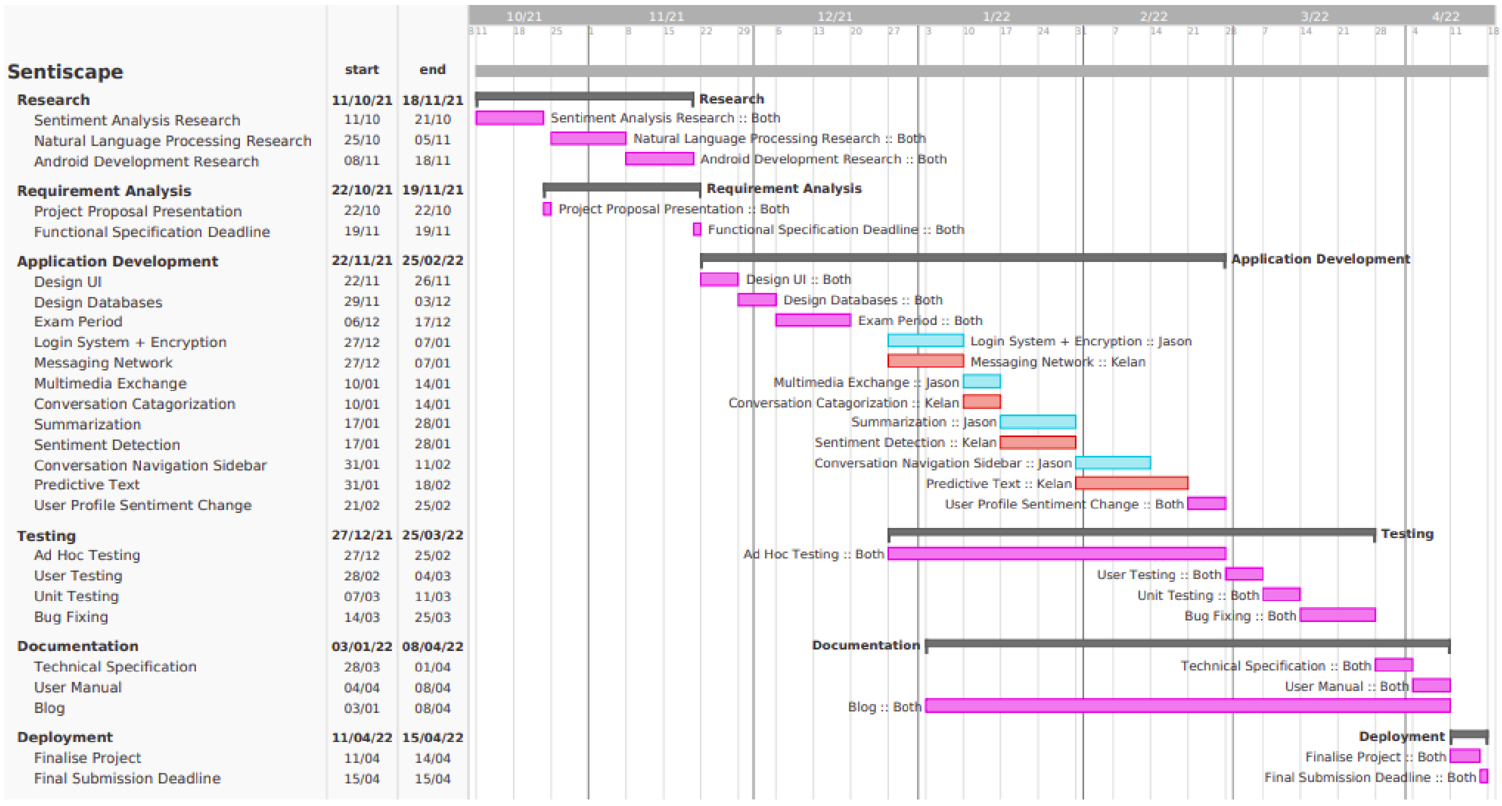


6. Preliminary Schedule

6.1 Schedule Overview

This section shows our project plan through the use of our GANTT chart. We want to show our progress and the path we wish to take as we complete the project. Times and dates may be subject to change if the need occurs but the plan is to reach all of our goals on schedule.

6.2 GANTT Chart



Sentiscape	start	end
Research	**11/10/21**	**18/11/21**
Sentiment Analysis Research	11/10	21/10
Natural Language Processing Research	25/10	05/11
Android Development Research	08/11	18/11
Requirement Analysis	**22/10/21**	**19/11/21**
Project Proposal Presentation	22/10	22/10
Functional Specification Deadline	19/11	19/11
Application Development	**22/11/21**	**25/02/22**
Design UI	22/11	26/11
Design Databases	29/11	03/12
Exam Period	06/12	17/12
Login System + Encryption	27/12	07/01
Messaging Network	27/12	07/01
Multimedia Exchange	10/01	14/01
Conversation Catagorization	10/01	14/01
Summarization	17/01	28/01
Sentiment Detection	17/01	28/01
Conversation Navigation Sidebar	31/01	11/02
Predictive Text	31/01	18/02
User Profile Sentiment Change	21/02	25/02
Testing	**27/12/21**	**25/03/22**
Ad Hoc Testing	27/12	25/02
User Testing	28/02	04/03
Unit Testing	07/03	11/03
Bug Fixing	14/03	25/03
Documentation	**03/01/22**	**08/04/22**
Technical Specification	28/03	01/04
User Manual	04/04	08/04
Blog	03/01	08/04
Deployment	**11/04/22**	**15/04/22**
Finalise Project	11/04	14/04
Final Submission Deadline	15/04	15/04

10/21
11/21
12/21
1/22
2/22
3/22
4/22
Research
Sentiment Analysis Research :: Both
Natural Language Processing Research :: Both
Android Development Research :: Both
Requirement Analysis
Project Proposal Presentation :: Both
Functional Specification Deadline :: Both
Application Development
Design UI :: Both
Design Databases :: Both
Exam Period :: Both
Login System + Encryption :: Jason
Messaging Network :: Kelan
Multimedia Exchange :: Jason
Conversation Catagorization :: Kelan
Summarization :: Jason
Sentiment Detection :: Kelan
Conversation Navigation Sidebar :: Jason
Predictive Text :: Kelan
User Profile Sentiment Change :: Both
Testing
Ad Hoc Testing :: Both
User Testing :: Both
Unit Testing :: Both
Bug Fixing :: Both
Documentation
Technical Specification :: Both
User Manual :: Both
Blog :: Both
Deployment
Finalise Project :: Both
Final Submission Deadline :: Both

7. Appendices

Kansei Engineering:
https://www.researchgate.net/publication/233349460_Concepts_methods_and_tools_in_Kansei_Engineering

TeamGantt: https://app.teamgantt.com/

Conversational Sentiment Analysis:
https://towardsdatascience.com/conversational-sentiment-analysis-52eabd20155b

Natural Language Processing:
https://medium.com/rocket-mortgage-technology-blog/conversational-summarization-with-natural-language-processing-c073a6bcaa3a

Notifications: https://developer.android.com/training/notify-user/build-notification/